

DAFTAR LAMPIRAN

Admin akun adapter

```
package
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin
```

```
import android.graphics.Color
import android.graphics.Typeface
import android.view.LayoutInflater
import android.view.ViewGroup
import
androidx.recyclerview.widget.RecyclerView
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.UsersModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListAdminAkunBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
```

```
class AdminAkunAdapter(
    private var listSemuaUser:
    ArrayList<UsersModel>,
    private var onClick: OnClickItem.ClickAkun
):
    RecyclerView.Adapter<AdminAkunAdapter.ViewHolder>() {
```

```
    class ViewHolder(val binding:
    ListAdminAkunBinding):
    RecyclerView.ViewHolder(binding.root)
```

```
    override fun onCreateViewHolder(parent:
    ViewGroup, viewType: Int): ViewHolder {
        val binding = ListAdminAkunBinding.inflate(
            LayoutInflater.from(parent.context), parent,
            false
        )
        return ViewHolder(binding)
    }
```

```
    override fun getItemCount(): Int {
        return (listSemuaUser.size+1)
    }
```

```
    override fun onBindViewHolder(holder:
    ViewHolder, position: Int) {
        holder.binding.apply {
            if(position==0){
                tvNo.text = "NO"
                tvNama.text = "Nama User"
                tvNomorHp.text = "Nomor Hp"
                tvUsername.text = "Username"
```

```
tvPassword.text = "Password"
tvSetting.text = ""
```

```
tvNo.setBackgroundResource(R.drawable.bg_table_
title)
```

```
tvNama.setBackgroundResource(R.drawable.bg_tab
le_title)
```

```
tvNomorHp.setBackgroundResource(R.drawable.bg_
table_title)
```

```
tvUsername.setBackgroundResource(R.drawable.bg_
table_title)
```

```
tvPassword.setBackgroundResource(R.drawable.bg_
table_title)
```

```
tvSetting.setBackgroundResource(R.drawable.bg_ta
ble_title)
```

```
tvNo.setTextColor(Color.parseColor("#ffffff"))
```

```
tvNama.setTextColor(Color.parseColor("#ffffff"))
```

```
tvNomorHp.setTextColor(Color.parseColor("#ffffff"
))
```

```
tvUsername.setTextColor(Color.parseColor("#ffffff"
))
```

```
tvPassword.setTextColor(Color.parseColor("#ffffff"
))
```

```
tvSetting.setTextColor(Color.parseColor("#ffffff"))
```

```
tvNo.setTypeface(null, Typeface.BOLD)
tvNama.setTypeface(null,
```

```
Typeface.BOLD)
```

```
tvNomorHp.setTypeface(null,
Typeface.BOLD)
```

```
tvUsername.setTypeface(null,
Typeface.BOLD)
```

```
tvPassword.setTypeface(null,
Typeface.BOLD)
```

```
tvSetting.setTypeface(null,
Typeface.BOLD)
```

```
}
```

```
else{
```

```
val semuaUser = listSemuaUser[(position-
1)]
```

```
tvNo.text = "$position"
```

```
tvNama.text = semuaUser.nama
```

```
tvNomorHp.text = semuaUser.nomorHp
```

```

        tvUsername.text = semuaUser.username
        tvPassword.text = semuaUser.password
        tvSetting.text = "..."

tvNo.setBackgroundResource(R.drawable.bg_table)

tvNama.setBackgroundResource(R.drawable.bg_table)

tvNomorHp.setBackgroundResource(R.drawable.bg_table)

tvUsername.setBackgroundResource(R.drawable.bg_table)

tvPassword.setBackgroundResource(R.drawable.bg_table)

tvSetting.setBackgroundResource(R.drawable.bg_table)

tvNo.setTextColor(Color.parseColor("#000000"))

tvNama.setTextColor(Color.parseColor("#000000"))

tvNomorHp.setTextColor(Color.parseColor("#000000"))

tvUsername.setTextColor(Color.parseColor("#000000"))

tvPassword.setTextColor(Color.parseColor("#000000"))

tvSetting.setTextColor(Color.parseColor("#000000"))

        tvNo.setTypeface(null,
        Typeface.NORMAL)
        tvNama.setTypeface(null,
        Typeface.NORMAL)
        tvNomorHp.setTypeface(null,
        Typeface.NORMAL)
        tvUsername.setTypeface(null,
        Typeface.NORMAL)
        tvPassword.setTypeface(null,
        Typeface.NORMAL)
        tvSetting.setTypeface(null,
        Typeface.NORMAL)

        tvSetting.setOnClickListener {
            onClick.clickItemSetting(semuaUser, it)
        }
    }
}
}
}

```

Admin halaman galeri herbal adapter

```

package
com.example.edukasipengobatanpadapenderitastrok

```

```

e.adapter.admin

```

```

import android.view.LayoutInflater
import android.view.ViewGroup
import android.widget.Toast
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalMainModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListAdminGaleriHerbalHalamanBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListTentangStrokeBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem

```

```

class AdminHalamanGaleriHerbalAdapter(
    private val listHalGaleriHerbal:
    ArrayList<GaleriHerbalMainModel>,
    private val click:
    OnClickItem.ClickAdminHalamanGaleriHerbal
):
    RecyclerView.Adapter<AdminHalamanGaleriHerbalAdapter.ViewHolder>() {
        class ViewHolder(val binding:
        ListAdminGaleriHerbalHalamanBinding):
        RecyclerView.ViewHolder(binding.root)

```

```

        override fun onCreateViewHolder(parent:
        ViewGroup, viewType: Int): ViewHolder {
            val view =
            ListAdminGaleriHerbalHalamanBinding.inflate(Lay
            outInflater.from(parent.context), parent, false)
            return ViewHolder(view)
        }

```

```

        override fun getItemCount(): Int {
            return listHalGaleriHerbal.size
        }

```

```

        override fun onBindViewHolder(holder:
        ViewHolder, position: Int) {
            val galeriHerbal =
            listHalGaleriHerbal[position]
            holder.binding.apply {
                tvJudulObatHerbal.text = galeriHerbal.judul
                tvDeskripsiObatHerbal.text =
                galeriHerbal.deskripsi

```

```

                Glide.with(holder.itemView)

```

```

                .load("${Constant.BASE_URL}${Constant.LOCAT
                ION_GAMBAR}${galeriHerbal.gambar!!}") //

```

```
URL Gambar
        .error(R.drawable.gambar_error_image)
        .into(ivObatHerbal) // imageView mana
yang akan diterapkan
```

```
        ivObatHerbal.setOnClickListener {
            click.clickItemGambar(galeriHerbal.gambar!!,
            galeriHerbal.judul!!, it)
        }
        holder.itemView.setOnClickListener {
            //
            Toast.makeText(holder.itemView.context, "klik
            semua", Toast.LENGTH_SHORT).show()
            click.clickItemNextPage(galeriHerbal, it)
        }
        ivSetting.setOnClickListener {
            click.clickItemSetting(galeriHerbal, it)
        }
    }
}
}
```

Admin halamanTentang stroke adapter

```
package
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TentangStrokeListModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListAdminTentangStrokeHalamanBin
ding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem

class AdminHalamanTentangStrokeAdapter(
    private val listTentangStroke:
    ArrayList<TentangStrokeListModel>,
    private val click:
    OnClickItem.ClickAdminTentangStroke
):
    RecyclerView.Adapter<AdminHalamanTentangStro
    keAdapter.ViewHolder>() {
        class ViewHolder(val binding:
        ListAdminTentangStrokeHalamanBinding):
        RecyclerView.ViewHolder(binding.root)

        override fun onCreateViewHolder(parent:
        ViewGroup, viewType: Int): ViewHolder {
            val view =
            ListAdminTentangStrokeHalamanBinding.inflate(La
            youtInflater.from(parent.context), parent, false)
            return ViewHolder(view)
        }
    }
}
```

```
override fun getItemCount(): Int {
    return listTentangStroke.size
}
```

```
override fun onBindViewHolder(holder:
ViewHolder, position: Int) {
    val tentangStroke = listTentangStroke[position]
    holder.binding.apply {
        tvJudulTentangStroke.text =
tentangStroke.halaman
```

```
        btnTentangStroke.setOnClickListener {
            click.clickItem(tentangStroke,
tentangStroke.id_hal_tentang_stroke!!,
tentangStroke.halaman!!, it)
        }
```

```
        ivSetting.setOnClickListener {
            click.clickItem(tentangStroke,
tentangStroke.id_hal_tentang_stroke!!,
tentangStroke.halaman!!, it)
        }
    }
}
```

admin meu sehat adapter

```
package
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin
```

```
import android.graphics.Color
import android.graphics.Typeface
import android.view.Gravity
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.MenuSehatModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListAdminGaleriHerbalValueBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListAdminTentangStrokeValueBind
ing
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListAdminMenuSehatBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Youtube
```

```

class AdminMenuSehatAdapter(
    private var listMenuSehat:
    ArrayList<MenuSehatModel>,
    private var onClick:
    OnClickItem.ClickAdminMenuSehat
):
    RecyclerView.Adapter<AdminMenuSehatAdapter.
    ViewHolder>() {

```

```

    class ViewHolder(val binding:
    ListAdminMenuSehatBinding):
    RecyclerView.ViewHolder(binding.root)

```

```

    override fun onCreateViewHolder(parent:
    ViewGroup, viewType: Int): ViewHolder {
        val binding =
        ListAdminMenuSehatBinding.inflate(
        LayoutInflater.from(parent.context), parent,
        false
        )
        return ViewHolder(binding)
    }

```

```

    override fun getItemCount(): Int {
        return (listMenuSehat.size+1)
    }

```

```

    override fun onBindViewHolder(holder:
    ViewHolder, position: Int) {
        holder.binding.apply {
            if(position==0){
                tvNo.text = "NO"
                tvJudul.text = "Nama Menu Sehat"
                tvDeskripsi.text = "Isi Penjelasan"
                tvSetting.text = ""
            }

```

```

            tvNo.setBackgroundResource(R.drawable.bg_table_
            title)

```

```

            tvJudul.setBackgroundResource(R.drawable.bg_tabl
            e_title)

```

```

            tvDeskripsi.setBackgroundResource(R.drawable.bg_
            table_title)

```

```

            tvSetting.setBackgroundResource(R.drawable.bg_ta
            ble_title)

```

```

            tvNo.setTextColor(Color.parseColor("#ffffff"))

```

```

            tvJudul.setTextColor(Color.parseColor("#ffffff"))

```

```

            tvDeskripsi.setTextColor(Color.parseColor("#ffffff"
            ))

```

```

            tvSetting.setTextColor(Color.parseColor("#ffffff"))

```

```

                tvNo.setTypeface(null, Typeface.BOLD)
                tvJudul.setTypeface(null,
                Typeface.BOLD)
                tvDeskripsi.setTypeface(null,
                Typeface.BOLD)

```

```

                tvSetting.setTypeface(null,
                Typeface.BOLD)
            }
            else{
                val terapi = listMenuSehat[(position-1)]

                tvNo.text = "$position"
                tvJudul.text = terapi.judul
                tvDeskripsi.text = terapi.deskripsi
                tvSetting.text = ":::"

```

```

            tvNo.setBackgroundResource(R.drawable.bg_table)

```

```

            tvJudul.setBackgroundResource(R.drawable.bg_tabl
            e)

```

```

            tvDeskripsi.setBackgroundResource(R.drawable.bg_
            table)

```

```

            tvSetting.setBackgroundResource(R.drawable.bg_ta
            ble)

```

```

            tvNo.setTextColor(Color.parseColor("#000000"))

```

```

            tvJudul.setTextColor(Color.parseColor("#000000"))

```

```

            tvDeskripsi.setTextColor(Color.parseColor("#00000
            0"))

```

```

            tvSetting.setTextColor(Color.parseColor("#000000"
            ))

```

```

                tvNo.setTypeface(null,
                Typeface.NORMAL)
                tvJudul.setTypeface(null,
                Typeface.NORMAL)
                tvDeskripsi.setTypeface(null,
                Typeface.NORMAL)
                tvSetting.setTypeface(null,
                Typeface.NORMAL)

```

```

                tvJudul.setOnClickListener{
                    onClick.clickItemJudul(terapi.judul!!,
                    it)
                }

```

```

                tvDeskripsi.setOnClickListener {

```

```

                    onClick.clickItemDeskripsi(terapi.deskripsi!!, it)
                }

```

```

                tvSetting.setOnClickListener {
                    onClick.clickItemSetting(terapi, it)
                }

```

```

            }
        }
    }
}

```

adminTerapi adapter

```

package
com.example.edukasipengobatanpadapenderitaastrok
e.adapter.admin

```

```

import android.graphics.Color
import android.graphics.Typeface
import android.view.Gravity
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TerapiModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListAdminGaleriHerbalValueBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListAdminTentangStrokeValueBindin
g
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListAdminTerapiBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Youtube

class AdminTerapiAdapter(
    private var listTerapi: ArrayList<TerapiModel>,
    private var onClick:
OnClickItem.ClickAdminTerapi
):
RecyclerView.Adapter<AdminTerapiAdapter.View
Holder>() {

    private var youtube = Youtube()

    class ViewHolder(val binding:
ListAdminTerapiBinding):
RecyclerView.ViewHolder(binding.root)

    override fun onCreateViewHolder(parent:
ViewGroup, viewType: Int): ViewHolder {
        val binding = ListAdminTerapiBinding.inflate(
            LayoutInflater.from(parent.context), parent,
            false
        )
        return ViewHolder(binding)
    }

    override fun getItemCount(): Int {
        return (listTerapi.size+1)
    }

    private fun setShowTextImage(holder:
ViewHolder){
        holder.binding.apply {
            tvGambar.visibility = View.VISIBLE

```

```

            ivGambar.visibility = View.GONE
        }
    }

    private fun setShowImage(holder: ViewHolder){
        holder.binding.apply {
            ivGambar.visibility = View.VISIBLE
            tvGambar.visibility = View.GONE
        }
    }

    override fun onBindViewHolder(holder:
ViewHolder, position: Int) {
        holder.binding.apply {
            if(position==0){
                tvNo.text = "NO"
                tvNamaTerapi.text = "Nama Terapi"
                tvDeskripsi.text = "Isi Penjelasan"
                tvGambar.text = "Gambar"
                tvSetting.text = ""

                setShowTextImage(holder)

                tvNo.setBackgroundResource(R.drawable.bg_table_
                title)

                tvNamaTerapi.setBackgroundResource(R.drawable.
                bg_table_title)

                tvDeskripsi.setBackgroundResource(R.drawable.bg_
                table_title)

                tvGambar.setBackgroundResource(R.drawable.bg_t
                able_title)

                tvSetting.setBackgroundResource(R.drawable.bg_ta
                ble_title)

                tvNo.setTextColor(Color.parseColor("#ffffff"))

                tvNamaTerapi.setTextColor(Color.parseColor("#fff
                ff"))

                tvDeskripsi.setTextColor(Color.parseColor("#ffffff"
                ))

                tvGambar.setTextColor(Color.parseColor("#ffffff"))

                tvSetting.setTextColor(Color.parseColor("#ffffff"))

                tvNo.setTypeface(null, Typeface.BOLD)
                tvNamaTerapi.setTypeface(null,
                Typeface.BOLD)
                tvDeskripsi.setTypeface(null,
                Typeface.BOLD)
                tvGambar.setTypeface(null,
                Typeface.BOLD)
                tvSetting.setTypeface(null,
                Typeface.BOLD)
            }
            else{
                val terapi = listTerapi[(position-1)]

```



```

        override fun getItemCount(): Int {
            return listTestimoni.size
        }

        override fun onBindViewHolder(holder:
ViewHolder, position: Int) {
            val testimoni = listTestimoni[position]
            holder.binding.apply {
                val nama = testimoni.nama!!
                val tanggal =
tanggalDanWaktu.konversiBulan(testimoni.tanggal!
!)
                val valueTestimoni = testimoni.testimoni!!
                val bintang =
testimoni.bintang!!.trim().toInt()
                val firstLetter = nama.substring(0, 1)

                tvNama.text = nama
                tvTanggal.text = tanggal
                tvTestimoni.text = valueTestimoni
                tvInisial.text = firstLetter

                val listBgCircle = ArrayList<Int>()
                listBgCircle.add(R.drawable.bg_circle_1)
                listBgCircle.add(R.drawable.bg_circle_2)
                listBgCircle.add(R.drawable.bg_circle_3)
                listBgCircle.add(R.drawable.bg_circle_4)
                listBgCircle.add(R.drawable.bg_circle_5)

                when (bintang) {
                    1 -> {

ivBintang1.setImageResource(R.drawable.icon_star
_aktif)
                }
                    2 -> {

ivBintang1.setImageResource(R.drawable.icon_star
_aktif)

ivBintang2.setImageResource(R.drawable.icon_star
_aktif)
                }
                    3 -> {

ivBintang1.setImageResource(R.drawable.icon_star
_aktif)

ivBintang2.setImageResource(R.drawable.icon_star
_aktif)

ivBintang3.setImageResource(R.drawable.icon_star
_aktif)
                }
                    4 -> {

ivBintang1.setImageResource(R.drawable.icon_star
_aktif)

ivBintang2.setImageResource(R.drawable.icon_star
_aktif)

ivBintang3.setImageResource(R.drawable.icon_star
_aktif)

```

```

ivBintang4.setImageResource(R.drawable.icon_star
_aktif)
                }
                    5 -> {

ivBintang1.setImageResource(R.drawable.icon_star
_aktif)

ivBintang2.setImageResource(R.drawable.icon_star
_aktif)

ivBintang3.setImageResource(R.drawable.icon_star
_aktif)

ivBintang4.setImageResource(R.drawable.icon_star
_aktif)

ivBintang5.setImageResource(R.drawable.icon_star
_aktif)
                }
            }

            val mathRandom =
(Math.random()*5).toInt()

tvInisial.setBackgroundResource(listBgCircle[math
Random])

            if(testimoni.gambar!!.trim().isEmpty()){
                ivBukti.visibility = View.GONE
            } else{
                ivBukti.visibility = View.VISIBLE
                Glide.with(holder.itemView)

.load("${Constant.BASE_URL}${Constant.LOCAT
ION_GAMBAR}${testimoni.gambar!!}") // URL
Gambar

                .error(R.drawable.gambar_error_image)
                .into(ivBukti) // imageView mana yang
akan diterapkan
            }

            ivBukti.setOnClickListener {
                click.clickGambar(testimoni.gambar!!,
testimoni.nama!!, it)
            }
        }
    }
}

```

Admin value galeri gerbal adapter

```

package
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin

import android.graphics.Color
import android.graphics.Typeface
import android.view.Gravity
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.bumptechnology.glide.Glide

```

```

import
com.example.edukasipengobatanpadapenderitastroke.R
import
com.example.edukasipengobatanpadapenderitastroke.utils.Constant
import
com.example.edukasipengobatanpadapenderitastroke.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastroke.data.model.GaleriHerbalListModel
import
com.example.edukasipengobatanpadapenderitastroke.databinding.ListAdminGaleriHerbalValueBinding
import
com.example.edukasipengobatanpadapenderitastroke.databinding.ListAdminTentangStrokeValueBinding
import
com.example.edukasipengobatanpadapenderitastroke.utils.Youtube

class AdminValueGaleriHerbalAdapter(
    private var listValGaleriHerbal:
    ArrayList<GaleriHerbalListModel>,
    private var onClick:
    OnClickItem.ClickAdminValueGaleriHerbal
):
    RecyclerView.Adapter<AdminValueGaleriHerbalAdapter.ViewHolder>() {

        private var youtube = Youtube()

        class ViewHolder(val binding:
        ListAdminGaleriHerbalValueBinding):
        RecyclerView.ViewHolder(binding.root)

            override fun onCreateViewHolder(parent:
            ViewGroup, viewType: Int): ViewHolder {
                val binding =
                ListAdminGaleriHerbalValueBinding.inflate(
                    LayoutInflater.from(parent.context), parent,
                    false
                )
                return ViewHolder(binding)
            }

            override fun getItemCount(): Int {
                return (listValGaleriHerbal.size+1)
            }

            private fun setShowTextImage(holder:
            ViewHolder){
                holder.binding.apply {
                    tvGambar.visibility = View.VISIBLE
                    tvYoutube.visibility = View.VISIBLE

                    ivGambar.visibility = View.GONE
                    ivYoutube.visibility = View.GONE
                }
            }

            private fun setShowImage(holder: ViewHolder){
                holder.binding.apply {

```

```

                    ivGambar.visibility = View.VISIBLE
                    ivYoutube.visibility = View.VISIBLE

                    tvGambar.visibility = View.GONE
                    tvYoutube.visibility = View.GONE
                }
            }

            override fun onBindViewHolder(holder:
            ViewHolder, position: Int) {
                holder.binding.apply {
                    if(position==0){
                        tvNo.text = "NO"
                        tvJudul.text = "Judul Isi"
                        tvDeskripsi.text = "Isi Penjelasan"
                        tvTataCaraPengolahan.text = "Tata Cara
                        Pengolahan"
                        tvGambar.text = "Gambar"
                        tvYoutube.text = "Youtube"
                        tvSetting.text = ""

                        setShowTextImage(holder)

                        tvNo.setBackgroundResource(R.drawable.bg_table_
                        title)

                        tvJudul.setBackgroundResource(R.drawable.bg_tabl
                        e_title)

                        tvDeskripsi.setBackgroundResource(R.drawable.bg_
                        table_title)

                        tvTataCaraPengolahan.setBackgroundResource(R.dr
                        awable.bg_table_title)

                        tvGambar.setBackgroundResource(R.drawable.bg_t
                        able_title)

                        tvYoutube.setBackgroundResource(R.drawable.bg_t
                        able_title)

                        tvSetting.setBackgroundResource(R.drawable.bg_ta
                        ble_title)

                        tvNo.setTextColor(Color.parseColor("#ffffff"))

                        tvJudul.setTextColor(Color.parseColor("#ffffff"))

                        tvDeskripsi.setTextColor(Color.parseColor("#ffffff"
                        ))

                        tvTataCaraPengolahan.setTextColor(Color.parseCol
                        or("#ffffff"))

                        tvGambar.setTextColor(Color.parseColor("#ffffff"))

                        tvYoutube.setTextColor(Color.parseColor("#ffffff"))

                        tvSetting.setTextColor(Color.parseColor("#ffffff"))

                        tvNo.setTypeface(null, Typeface.BOLD)
                        tvJudul.setTypeface(null,
                        Typeface.BOLD)

```

```

                tvDeskripsi.setTypeface(null,
Typeface.BOLD)
                tvTataCaraPengolahan.setTypeface(null,
Typeface.BOLD)
                tvGambar.setTypeface(null,
Typeface.BOLD)
                tvYoutube.setTypeface(null,
Typeface.BOLD)
                tvSetting.setTypeface(null,
Typeface.BOLD)
            }
            else{
                val galeriHerbal =
listValGaleriHerbal[(position-1)]

                tvNo.text = "$position"
                tvJudul.text = galeriHerbal.nama
                tvDeskripsi.text = galeriHerbal.deskripsi
                tvTataCaraPengolahan.text =
galeriHerbal.deskripsi
                tvSetting.text = "..."

                setShowImage(holder)

                Glide.with(holder.itemView)

.load("${Constant.BASE_URL}${Constant.LOCATION_
GAMBAR}${galeriHerbal.gambar}")

.error(R.drawable.gambar_error_image)
                .into(ivGambar)

                // set Image Youtube

youtube.setImageFromYoutube(holder.itemView.co
ntext, galeriHerbal.youtube!!, ivYoutube)

tvNo.setBackgroundResource(R.drawable.bg_table)

tvJudul.setBackgroundResource(R.drawable.bg_tabl
e)

tvDeskripsi.setBackgroundResource(R.drawable.bg_
table)

tvTataCaraPengolahan.setBackgroundResource(R.dr
awable.bg_table)

tvGambar.setBackgroundResource(R.drawable.bg_t
able)

tvYoutube.setBackgroundResource(R.drawable.bg_t
able)

tvSetting.setBackgroundResource(R.drawable.bg_ta
ble)

tvNo.setTextColor(Color.parseColor("#000000"))

tvJudul.setTextColor(Color.parseColor("#000000"))

tvDeskripsi.setTextColor(Color.parseColor("#00000
0"))

```

```

tvTataCaraPengolahan.setTextColor(Color.parseCol
or("#000000"))

tvGambar.setTextColor(Color.parseColor("#000000
"))

tvYoutube.setTextColor(Color.parseColor("#000000
"))

tvSetting.setTextColor(Color.parseColor("#000000"
))

                tvNo.setTypeface(null,
Typeface.NORMAL)
                tvJudul.setTypeface(null,
Typeface.NORMAL)
                tvDeskripsi.setTypeface(null,
Typeface.NORMAL)
                tvTataCaraPengolahan.setTypeface(null,
Typeface.NORMAL)
                tvGambar.setTypeface(null,
Typeface.NORMAL)
                tvYoutube.setTypeface(null,
Typeface.NORMAL)
                tvSetting.setTypeface(null,
Typeface.NORMAL)

                tvJudul.setOnClickListener {

onClick.clickItemJudul(galeriHerbal.nama!!, it)
                }
                tvDeskripsi.setOnClickListener {

onClick.clickItemDeskripsi(galeriHerbal.deskripsi!!,
it)
                }
                tvTataCaraPengolahan.setOnClickListener

{

onClick.clickItemTataCaraPengolahan(galeriHerbal.
deskripsi!!, it)
                }
                ivGambar.setOnClickListener {

onClick.clickItemGambar(galeriHerbal.gambar!!,
galeriHerbal.nama!!, it)
                }
                ivYoutube.setOnClickListener {

//
youtube.setToYoutubeVideo(holder.itemView.conte
xt, galeriHerbal.youtube!!)

onClick.clickItemYoutube(galeriHerbal.youtube!!,
it)
                }
                tvSetting.setOnClickListener {
                    onClick.clickItemSetting(galeriHerbal,
it)
                }
            }
        }
    }
}

```

Admin value halaman tentang stroke adaadapter

```
package
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin

import android.graphics.Color
import android.graphics.Typeface
import android.view.Gravity
import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TentangStrokeDetailModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListAdminTentangStrokeValueBindin
g

class AdminValueTentangStrokeAdapter(
    private val listValTentangStroke:
    ArrayList<TentangStrokeDetailModel>,
    private var onClick:
    OnClickItem.ClickAdminValueTentangStroke
):
    RecyclerView.Adapter<AdminValueTentangStroke
    Adapter.ViewHolder>() {

    class ViewHolder(val binding:
    ListAdminTentangStrokeValueBinding):
    RecyclerView.ViewHolder(binding.root)

    override fun onCreateViewHolder(parent:
    ViewGroup, viewType: Int): ViewHolder {
        val binding =
        ListAdminTentangStrokeValueBinding.inflate(
            LayoutInflater.from(parent.context), parent,
            false
        )
        return ViewHolder(binding)
    }

    override fun getItemCount(): Int {
        return (listValTentangStroke.size+1)
    }

    override fun onBindViewHolder(holder:
    ViewHolder, position: Int) {
        holder.binding.apply {
            if(position==0){
                tvNo.text = "NO"
                tvJudul.text = "Judul Isi"
                tvDeskripsi.text = "Isi Penjelasan"
                tvSetting.text = ""
```

```
tvNo.setBackgroundResource(R.drawable.bg_table_
title)

tvJudul.setBackgroundResource(R.drawable.bg_tabl
e_title)

tvDeskripsi.setBackgroundResource(R.drawable.bg_
table_title)

tvSetting.setBackgroundResource(R.drawable.bg_ta
ble_title)

tvNo.setTextColor(Color.parseColor("#ffffff"))

tvJudul.setTextColor(Color.parseColor("#ffffff"))

tvDeskripsi.setTextColor(Color.parseColor("#ffffff"
))

tvSetting.setTextColor(Color.parseColor("#ffffff"))

        tvNo.setTypeface(null, Typeface.BOLD)
        tvJudul.setTypeface(null,
        Typeface.BOLD)
        tvDeskripsi.setTypeface(null,
        Typeface.BOLD)
        tvSetting.setTypeface(null,
        Typeface.BOLD)
    }
    else{
        val valTentangStroke =
        listValTentangStroke[(position-1)]

        tvNo.text = "$position"
        tvJudul.text = valTentangStroke.judul
        tvDeskripsi.text =
        valTentangStroke.deskripsi
        tvSetting.text = "..."

tvNo.setBackgroundResource(R.drawable.bg_table)

tvJudul.setBackgroundResource(R.drawable.bg_tabl
e)

tvDeskripsi.setBackgroundResource(R.drawable.bg_
table)

tvSetting.setBackgroundResource(R.drawable.bg_ta
ble)

tvNo.setTextColor(Color.parseColor("#000000"))

tvJudul.setTextColor(Color.parseColor("#000000"))

tvDeskripsi.setTextColor(Color.parseColor("#00000
0"))

tvSetting.setTextColor(Color.parseColor("#000000"
))
```

```

        tvNo.setTypeface(null,
Typeface.NORMAL)
        tvJudul.setTypeface(null,
Typeface.NORMAL)
        tvDeskripsi.setTypeface(null,
Typeface.NORMAL)
        tvSetting.setTypeface(null,
Typeface.NORMAL)

        tvJudul.gravity =
Gravity.CENTER_VERTICAL

        tvJudul.setOnClickListener{

onClick.clickItemJudul(valTentangStroke.judul!!, it)
        }
        tvDeskripsi.setOnClickListener {

onClick.clickItemDeskripsi(valTentangStroke.deskri
psi!!, it)
        }
        tvSetting.setOnClickListener {

onClick.clickItemSetting(valTentangStroke, it)
        }
    }
}
}
}

```

Galeri herbal list adapter

```

package
com.example.edukasipengobatanpadapenderitastrok
e.adapter.user

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalListModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListGaleriHerbalListBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem

class GaleriHerbalListAdapter(
    private val listGaleriHerbalMain:
ArrayList<GaleriHerbalListModel>,
    private val click:
OnClickItem.ClickGaleriHerbalList
):
RecyclerView.Adapter<GaleriHerbalListAdapter.Vi
ewHolder>() {
    class ViewHolder(val binding:
ListGaleriHerbalListBinding):

```

```

RecyclerView.ViewHolder(binding.root)

        override fun onCreateViewHolder(parent:
ViewGroup, viewType: Int): ViewHolder {
            val view =
ListGaleriHerbalListBinding.inflate(LayoutInflater.f
rom(parent.context), parent, false)
            return ViewHolder(view)
        }

        override fun getItemCount(): Int {
            return listGaleriHerbalMain.size
        }

        override fun onBindViewHolder(holder:
ViewHolder, position: Int) {
            val galeriHerbalMain =
listGaleriHerbalMain[position]
            holder.binding.apply {
                tvJudulObatHerbal.text =
galeriHerbalMain.nama
            }
            //r
            Glide.with(holder.itemView)

            .load("${Constant.BASE_URL}${Constant.LOCAT
ION_GAMBAR}${galeriHerbalMain.gambar}")
            .error(R.drawable.gambar_error_image)
            .into(ivGaleriHerbalList)

            //
            holder.itemView.setOnClickListener {
                click.clickItem(galeriHerbalMain, it)
            }
        }
    }
}
}

```

Galeri herbal main adapter

```

package
com.example.edukasipengobatanpadapenderitastrok
e.adapter.user

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalMainModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListGaleriHerbalMainBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem

class GaleriHerbalMainAdapter(
    private val listGaleriHerbalMain:
ArrayList<GaleriHerbalMainModel>,

```

```

        private val click: OnClickItem.ClickObatHerbal
    ):
    RecyclerView.Adapter<GaleriHerbalMainAdapter.
    ViewHolder>() {
        class ViewHolder(val binding:
    ListGaleriHerbalMainBinding):
    RecyclerView.ViewHolder(binding.root)

        override fun onCreateViewHolder(parent:
    ViewGroup, viewType: Int): ViewHolder {
            val view =
    ListGaleriHerbalMainBinding.inflate(LayoutInflater
    .from(parent.context), parent, false)
            return ViewHolder(view)
        }

        override fun getItemCount(): Int {
            return listGaleriHerbalMain.size
        }

        override fun onBindViewHolder(holder:
    ViewHolder, position: Int) {
            val galeriHerbalMain =
    listGaleriHerbalMain[position]
            holder.binding.apply {
                tvJudulObatHerbal.text =
    galeriHerbalMain.judul
                tvDeskripsiObatHerbal.text =
    galeriHerbalMain.deskripsi

                Glide.with(holder.itemView)

                .load("${Constant.BASE_URL}${Constant.LOCATION_GAMBAR}${galeriHerbalMain.gambar}")
                .error(R.drawable.gambar_error_image)
                .into(ivObatHerbal)

                holder.itemView.setOnClickListener {

                    click.clickItem(galeriHerbalMain.id_hal_galeri_herbal!!, galeriHerbalMain.judul!!, it)
                }
            }
        }
    }
}

```

Menu sehat adapter

```

package
com.example.edukasipengobatanpadapenderitastroke.adapter.user

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.example.edukasipengobatanpadapenderitastroke.data.model.MenuSehatModel
import com.example.edukasipengobatanpadapenderitastroke.databinding.ListTentangStrokeDetailBinding

class MenuSehatAdapter(

```

```

    private val listMenuSehat:
    ArrayList<MenuSehatModel>
    ):
    RecyclerView.Adapter<MenuSehatAdapter.ViewHolder>() {
        class ViewHolder(val binding:
    ListTentangStrokeDetailBinding):
    RecyclerView.ViewHolder(binding.root)

        override fun onCreateViewHolder(parent:
    ViewGroup, viewType: Int): ViewHolder {
            val view =
    ListTentangStrokeDetailBinding.inflate(LayoutInflater
    .from(parent.context), parent, false)
            return ViewHolder(view)
        }

        override fun getItemCount(): Int {
            return listMenuSehat.size
        }

        override fun onBindViewHolder(holder:
    ViewHolder, position: Int) {
            val menuSehat = listMenuSehat[position]
            holder.binding.apply {
                tvJudulTentangStroke.text =
    menuSehat.judul
                tvDeskripsiTentangStroke.text =
    menuSehat.deskripsi
            }
        }
    }
}

```

Tentang stroke adapter

```

package
com.example.edukasipengobatanpadapenderitastroke.adapter.user

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.example.edukasipengobatanpadapenderitastroke.data.model.TentangStrokeListModel
import com.example.edukasipengobatanpadapenderitastroke.databinding.ListTentangStrokeBinding
import com.example.edukasipengobatanpadapenderitastroke.utils.OnClickItem

class TentangStrokeAdapter(
    private val listTentangStroke:
    ArrayList<TentangStrokeListModel>,
    private val click:
    OnClickItem.ClickTentangStroke
    ):
    RecyclerView.Adapter<TentangStrokeAdapter.ViewHolder>() {
        class ViewHolder(val binding:
    ListTentangStrokeBinding):
    RecyclerView.ViewHolder(binding.root)

```

```

        override fun onCreateView(parent:
ViewGroup, viewType: Int): ViewHolder {
            val view =
ListTentangStrokeBinding.inflate(LayoutInflater.from
(parent.context), parent, false)
            return ViewHolder(view)
        }

        override fun getItemCount(): Int {
            return listTentangStroke.size
        }

        override fun onBindViewHolder(holder:
ViewHolder, position: Int) {
            val tentangStroke = listTentangStroke[position]
            holder.binding.apply {
                tvJudulTentangStroke.text =
tentangStroke.halaman

                btnTentangStroke.setOnClickListener {

                    click.clickItem(tentangStroke.id_hal_tentang_stroke
!!, tentangStroke.halaman!!, it)
                }
            }
        }
    }
}

```

Tentang stroke detail adaapter

```

package
com.example.edukasipengobatanpadapenderitastrok
e.adapter.user

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TentangStrokeDetailModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListTentangStrokeDetailBinding

class TentangStrokeDetailAdapter(
    private val listTentangStrokeDetail:
ArrayList<TentangStrokeDetailModel>
):
RecyclerView.Adapter<TentangStrokeDetailAdapte
r.ViewHolder>() {
    class ViewHolder(val binding:
ListTentangStrokeDetailBinding):
RecyclerView.ViewHolder(binding.root)

        override fun onCreateView(parent:
ViewGroup, viewType: Int): ViewHolder {
            val view =
ListTentangStrokeDetailBinding.inflate(LayoutInflat
er.from(parent.context), parent, false)
            return ViewHolder(view)
        }

        override fun getItemCount(): Int {
            return listTentangStrokeDetail.size
        }
    }
}

```

```

    }

    override fun onBindViewHolder(holder:
ViewHolder, position: Int) {
        val tentangStroke =
listTentangStrokeDetail[position]
        holder.binding.apply {
            tvJudulTentangStroke.text =
tentangStroke.judul
            tvDeskripsiTentangStroke.text =
tentangStroke.deskripsi
        }
    }
}

```

Terapi adapter

```

package
com.example.edukasipengobatanpadapenderitastrok
e.adapter.user

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TerapiModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ListGaleriHerbalMainBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem

class TerapiAdapter(
    private val listTerapi: ArrayList<TerapiModel>,
    private val click: OnClickItem.ClickTerapi
):
RecyclerView.Adapter<TerapiAdapter.ViewHolder
>() {
    class ViewHolder(val binding:
ListGaleriHerbalMainBinding):
RecyclerView.ViewHolder(binding.root)

        override fun onCreateView(parent:
ViewGroup, viewType: Int): ViewHolder {
            val view =
ListGaleriHerbalMainBinding.inflate(LayoutInflater
.from(parent.context), parent, false)
            return ViewHolder(view)
        }

        override fun getItemCount(): Int {
            return listTerapi.size
        }

        override fun onBindViewHolder(holder:
ViewHolder, position: Int) {

```

```

        val terapi = listTerapi[position]
        holder.binding.apply {
            tvJudulObatHerbal.text = terapi.nama_terapi
            tvDeskripsiObatHerbal.text = terapi.deskripsi

            Glide.with(holder.itemView)

                .load("${Constant.BASE_URL}${Constant.LOCATION_GAMBAR}${terapi.gambar}")
                .error(R.drawable.gambar_error_image)
                .into(ivObatHerbal)

            holder.itemView.setOnClickListener {
                click.clickItem(terapi, it)
            }
        }
    }
}

```

Testimoni adapter

```

package
com.example.edukasipengobatanpadapenderitastroke.adapter.user

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import com.example.edukasipengobatanpadapenderitastroke.R
import com.example.edukasipengobatanpadapenderitastroke.data.model.TestimoniModel
import com.example.edukasipengobatanpadapenderitastroke.databinding.ListTestimoniBinding
import com.example.edukasipengobatanpadapenderitastroke.utils.Constant
import com.example.edukasipengobatanpadapenderitastroke.utils.OnClickItem
import com.example.edukasipengobatanpadapenderitastroke.utils.TanggalDanWaktu
import java.lang.Exception

class TestimoniAdapter(
    private val listTestimoni:
        ArrayList<TestimoniModel>,
    private var checkTestimoni: Boolean, // Kalau
        true maka testimoni halaman home
    private val onClick: OnClickItem.ClickTestimoni
):
    RecyclerView.Adapter<TestimoniAdapter.ViewHolder>() {

        private val tanggalDanWaktu =
            TanggalDanWaktu()

        class ViewHolder(val binding:
            ListTestimoniBinding)

```

```

        RecyclerView.ViewHolder(binding.root)

        override fun onCreateViewHolder(parent:
            ViewGroup, viewType: Int): ViewHolder {
            val view =
                ListTestimoniBinding.inflate(LayoutInflater.from(pa
                    rent.context), parent, false)
            return ViewHolder(view)
        }

        override fun getItemCount(): Int {
            var size = listTestimoni.size
            if(checkTestimoni){
                size = if(listTestimoni.size<3){
                    listTestimoni.size
                } else{
                    3
                }
            } else{
                listTestimoni.size
            }

            return size
        }

        override fun onBindViewHolder(holder:
            ViewHolder, position: Int) {
            val testimoni = listTestimoni[position]
            holder.binding.apply {
                val nama = testimoni.nama!!
                val tanggal =
                    tanggalDanWaktu.konversiBulan(testimoni.tanggal!
                        !)

                val valueTestimoni = testimoni.testimoni!!
                val bintang =
                    testimoni.bintang!!?.trim().toInt()

                val firstLetter = try {
                    nama.substring(0, 1)
                } catch (ex: Exception){
                    "A"
                }

                tvNama.text = nama
                tvTanggal.text = tanggal
                tvTestimoni.text = valueTestimoni
                tvInisial.text = firstLetter

                val listBgCircle = ArrayList<Int>()
                listBgCircle.add(R.drawable.bg_circle)
                listBgCircle.add(R.drawable.bg_circle_1)
                listBgCircle.add(R.drawable.bg_circle_2)
                listBgCircle.add(R.drawable.bg_circle_3)
                listBgCircle.add(R.drawable.bg_circle_4)
                listBgCircle.add(R.drawable.bg_circle_5)

                when (bintang) {
                    1 -> {

                    ivBintang1.setImageResource(R.drawable.icon_star
                        _aktif)
                    }
                    2 -> {

```

```

ivBintang1.setImageResource(R.drawable.icon_star_aktif)

ivBintang2.setImageResource(R.drawable.icon_star_aktif)
    }
    3 -> {

ivBintang1.setImageResource(R.drawable.icon_star_aktif)

ivBintang2.setImageResource(R.drawable.icon_star_aktif)

ivBintang3.setImageResource(R.drawable.icon_star_aktif)
    }
    4 -> {

ivBintang1.setImageResource(R.drawable.icon_star_aktif)

ivBintang2.setImageResource(R.drawable.icon_star_aktif)

ivBintang3.setImageResource(R.drawable.icon_star_aktif)

ivBintang4.setImageResource(R.drawable.icon_star_aktif)
    }
    5 -> {

ivBintang1.setImageResource(R.drawable.icon_star_aktif)

ivBintang2.setImageResource(R.drawable.icon_star_aktif)

ivBintang3.setImageResource(R.drawable.icon_star_aktif)

ivBintang4.setImageResource(R.drawable.icon_star_aktif)

ivBintang5.setImageResource(R.drawable.icon_star_aktif)
    }
    }

    val mathRandom =
    (Math.random()*5).toInt()

    tvInisial.setBackgroundResource(listBgCircle[mathRandom])

    if(testimoni.gambar!!.trim().isEmpty()){
        ivBukti.visibility = View.GONE
    } else{
        ivBukti.visibility = View.VISIBLE
        Glide.with(holder.itemView)

    }.load("${Constant.BASE_URL}${Constant.LOCATION_GAMBAR}${testimoni.gambar!!}") // URL
    Gambar

```

```

.error(R.drawable.gambar_error_image)
        .into(ivBukti) // imageView mana yang
        akan diterapkan
    }
    ivBukti.setOnClickListener {
        onClick.clickGambar(testimoni.gambar!!,
        testimoni.nama!!, it)
    }
    }
    }
}

```

Galeri herbal listmodel

```

package
com.example.edukasipengobatanpadapenderitastroke.data.model

import android.os.Parcel
import android.os.Parcelable
import com.google.gson.annotations.SerializedName

class GaleriHerbalListModel (
    @SerializedName("id_val_galeri_herbal")
    var id_val_galeri_herbal: String? = null,

    @SerializedName("id_hal_galeri_herbal")
    var id_hal_galeri_herbal: String? = null,

    @SerializedName("nama")
    var nama: String? = null,

    @SerializedName("deskripsi")
    var deskripsi: String? = null,

    @SerializedName("tata_cara_pengolahan")
    var tata_cara_pengolahan: String? = null,

    @SerializedName("gambar")
    var gambar: String? = null,

    @SerializedName("youtube")
    var youtube: String? = null
): Parcelable {
    constructor(parcel: Parcel) : this(
        parcel.readString(),
        parcel.readString(),
        parcel.readString(),
        parcel.readString(),
        parcel.readString(),
        parcel.readString(),
        parcel.readString()
    ) {
    }

    override fun writeToParcel(parcel: Parcel, flags:
    Int) {
        parcel.writeString(id_val_galeri_herbal)
        parcel.writeString(id_hal_galeri_herbal)
        parcel.writeString(nama)
        parcel.writeString(deskripsi)
        parcel.writeString(tata_cara_pengolahan)
        parcel.writeString(gambar)
    }
}

```

```

        parcel.writeString(youtube)
    }

    override fun describeContents(): Int {
        return 0
    }

    companion object CREATOR :
Parcelable.Creator<GaleriHerbalListModel> {
        override fun createFromParcel(parcel: Parcel):
GaleriHerbalListModel {
            return GaleriHerbalListModel(parcel)
        }

        override fun newArray(size: Int):
Array<GaleriHerbalListModel?> {
            return arrayOfNulls(size)
        }
    }
}

```

Galeri herbal mainmodel

```

package
com.example.edukasipengobatanpadapenderitastrok
e.data.model

import android.os.Parcel
import android.os.Parcelable
import com.google.gson.annotations.SerializedName

class GaleriHerbalMainModel (
    @SerializedName("id_hal_galeri_herbal")
    var id_hal_galeri_herbal: String? = null,

    @SerializedName("judul")
    var judul: String? = null,

    @SerializedName("deskripsi")
    var deskripsi: String? = null,

    @SerializedName("gambar")
    var gambar: String? = null
): Parcelable {
    constructor(parcel: Parcel) : this(
        parcel.readString(),
        parcel.readString(),
        parcel.readString(),
        parcel.readString()
    ) {
    }

    override fun writeToParcel(parcel: Parcel, flags:
Int) {
        parcel.writeString(id_hal_galeri_herbal)
        parcel.writeString(judul)
        parcel.writeString(deskripsi)
        parcel.writeString(gambar)
    }

    override fun describeContents(): Int {
        return 0
    }

    companion object CREATOR :

```

```

Parcelable.Creator<GaleriHerbalMainMod
override fun createFromParcel(parcel:
GaleriHerbalMainModel {
    return GaleriHerbalMainModel(parcel)
}

    override fun newArray(size: Int):
Array<GaleriHerbalMainModel?> {
    return arrayOfNulls(size)
}
}
}

```

Menusehat model

```

package
com.example.edukasipengobatanpadapenderitastrok
e.data.model

import android.os.Parcel
import android.os.Parcelable
import com.google.gson.annotations.SerializedName

class MenuSehatModel (
    @SerializedName("id_menu_sehat")
    var id_menu_sehat: String? = null,

    @SerializedName("judul")
    var judul: String? = null,

    @SerializedName("deskripsi")
    var deskripsi: String? = null
)

```

reponse model

```

package
com.example.edukasipengobatanpadapenderitastrok
e.data.model

import com.google.gson.annotations.SerializedName

class ResponseModel(
    @SerializedName("response")
    var response: String? = null,

    @SerializedName("status")
    var status: String? = null,

    @SerializedName("message_response")
    var message_response: String? = null
)

```

Tentang stroke detailmodel

```

package
com.example.edukasipengobatanpadapenderitastrok
e.data.model

import android.os.Parcel
import android.os.Parcelable
import com.google.gson.annotations.SerializedName

class TentangStrokeDetailModel (
    @SerializedName("id_val_tentang_stroke")

```

```

var id_val_tentang_stroke: String? = null,

@SerializedName("id_hal_tentang_stroke")
var id_hal_tentang_stroke: String? = null,

@SerializedName("hal_tentang_stroke")
var hal_tentang_stroke: String? = null,

@SerializedName("judul")
var judul: String? = null,

@SerializedName("deskripsi")
var deskripsi: String? = null
)

```

terapi strokedetailmodel

```

package
com.example.edukasipengobatanpadapenderitastrok
e.data.model

import android.os.Parcel
import android.os.Parcelable
import com.google.gson.annotations.SerializedName

class TentangStrokeListModel (
    @SerializedName("id_hal_tentang_stroke")
    var id_hal_tentang_stroke: String? = null,

    @SerializedName("halaman")
    var halaman: String? = null
): Parcelable {
    constructor(parcel: Parcel) : this(
        parcel.readString(),
        parcel.readString()
    ) {
    }

    override fun writeToParcel(parcel: Parcel, flags:
Int) {
        parcel.writeString(id_hal_tentang_stroke)
        parcel.writeString(halaman)
    }

    override fun describeContents(): Int {
        return 0
    }

    companion object CREATOR :
Parcelable.Creator<TentangStrokeListModel> {
        override fun createFromParcel(parcel: Parcel):
TentangStrokeListModel {
            return TentangStrokeListModel(parcel)
        }

        override fun newArray(size: Int):
Array<TentangStrokeListModel?> {
            return arrayOfNulls(size)
        }
    }
}

```

Terapimodel

```

package

```

```

com.example.edukasipengobatanpadapenderitastrok
e.data.model

```

```

import android.os.Parcel
import android.os.Parcelable
import com.google.gson.annotations.SerializedName

```

```

class TerapiModel (
    @SerializedName("id_terapi")
    var id_terapi: String? = null,

    @SerializedName("nama_terapi")
    var nama_terapi: String? = null,

    @SerializedName("deskripsi")
    var deskripsi: String? = null,

    @SerializedName("gambar")
    var gambar: String? = null
): Parcelable {
    constructor(parcel: Parcel) : this(
        parcel.readString(),
        parcel.readString(),
        parcel.readString(),
        parcel.readString()
    ) {
    }

    override fun writeToParcel(parcel: Parcel, flags:
Int) {
        parcel.writeString(id_terapi)
        parcel.writeString(nama_terapi)
        parcel.writeString(deskripsi)
        parcel.writeString(gambar)
    }

    override fun describeContents(): Int {
        return 0
    }

    companion object CREATOR :
Parcelable.Creator<TerapiModel> {
        override fun createFromParcel(parcel: Parcel):
TerapiModel {
            return TerapiModel(parcel)
        }

        override fun newArray(size: Int):
Array<TerapiModel?> {
            return arrayOfNulls(size)
        }
    }
}

```

Testimonimodel

```

package
com.example.edukasipengobatanpadapenderitastrok
e.data.model

```

```

import com.google.gson.annotations.SerializedName

```

```

class TestimoniModel (
    @SerializedName("id_testimoni")
    var id_testimoni: String? = null,

```

```

        @SerializedName("id_user")
        var id_user: String? = null,

        @SerializedName("nama")
        var nama: String? = null,

        @SerializedName("email")
        var email: String? = null,

        @SerializedName("testimoni")
        var testimoni: String? = null,

        @SerializedName("bintang")
        var bintang: String? = null,

        @SerializedName("gambar")
        var gambar: String? = null,

        @SerializedName("tanggal")
        var tanggal: String? = null
    )

```

usermodel

```

package
com.example.edukasipengobatanpadapenderitastrok
e.data.model

import com.google.gson.annotations.SerializedName

class UserModel (
    @SerializedName("id_user")
    var idUser: String? = null,

    @SerializedName("nama")
    var nama: String? = null,

    @SerializedName("nomor_hp")
    var nomorHp: String? = null,

    @SerializedName("username")
    var username: String? = null,

    @SerializedName("password")
    var password: String? = null,

    @SerializedName("sebagai")
    var sebagai: String? = null
)

```

Adminakun activity

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.akun

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.MenuItem
import android.view.View
import android.widget.PopupMenu
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog

```

```

import
androidx.recyclerview.widget.LinearLayoutManager
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin.AdminAkunAdapter
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.UsersModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityAdminAkunBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogAkunBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKeteranganBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKonfirmasiBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KontrolNavigationDrawer
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

```

```

@AndroidEntryPoint
class AdminAkunActivity : AppCompatActivity() {
    private lateinit var binding:
    ActivityAdminAkunBinding
    private val viewModel: AdminAkunViewModel
    by viewModels()
    @Inject
    lateinit var loading: LoadingAlertDialog
    private lateinit var kontrolNavigationDrawer:
    KontrolNavigationDrawer
    private lateinit var adapter: AdminAkunAdapter
    private var tempAlertDialog: AlertDialog? = null
    override fun onCreate(savedInstanceState:
    Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
        ActivityAdminAkunBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setKontrolNavigationDrawer()
        setButton()
        fetchSemuaAkun()
        getSemuaAkun()
        getTambahUser()
    }
}

```

```

        getUpdateUser()
        getHapusUser()
    }

    private fun setKontrolNavigationDrawer() {
        binding.apply {
            kontrolNavigationDrawer =
            KontrolNavigationDrawer(this@AdminAkunActivit
            y)

            kontrolNavigationDrawer.cekSebagai(navView)

            kontrolNavigationDrawer.onClickItemNavigationDr
            awer(navView, drawerLayoutMain, ivDrawerView,
            this@AdminAkunActivity)
        }
    }

    // tambah
    private fun setButton() {
        binding.apply {
            btnTambah.setOnClickListener {
                showDialogTambahData()
            }
        }
    }

    // memunculkan tambah data
    private fun showDialogTambahData() {
        val view =
        AlertDialogAkunBinding.inflate(layoutInflater)

        val alertDialog =
        AlertDialog.Builder(this@AdminAkunActivity)
        alertDialog.setView(view.root)
        .setCancelable(true)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        tempAlertDialog = dialogInputan

        view.apply {

            btnSimpan.setOnClickListener {
                val nama =
                etEditNama.text.toString().trim()
                val nomorHp =
                etEditNomorHp.text.toString().trim()
                val username =
                etEditUsername.text.toString().trim()
                val password =
                etEditPassword.text.toString().trim()
                postTambahUser(nama, nomorHp,
                username, password)
            }
            btnBatal.setOnClickListener {
                dialogInputan.dismiss()
                tempAlertDialog = null
            }
        }
    }

    private fun postTambahUser(
        nama: String, nomorHp: String,
        username: String, password: String

```

```

    ) {
        viewModel.postTambahAkun(nama, nomorHp,
        username, password, "user")
    }

    private fun getTambahUser(){
        viewModel.getTambahAkun().observe(this@Admin
        AkunActivity){result->
            when(result){
                is UIState.Loading->
                loading.alertDialogLoading(this@AdminAkunActiv
                ity)
                is UIState.Failure->
                setFailureTambahAkun(result.message)
                is UIState.Success->
                setSuccessTambahAkun(result.data)
            }
        }

        private fun setSuccessTambahAkun(data:
        ArrayList<ResponseModel>) {
            loading.alertDialogCancel()
            if(data.isNotEmpty()){
                if(data[0].status=="0"){

                    Toast.makeText(this@AdminAkunActivity,
                    "Berhasil Tambah",
                    Toast.LENGTH_SHORT).show()
                    tempAlertDialog!!.dismiss()
                    tempAlertDialog = null
                    fetchSemuaAkun()
                } else{

                    Toast.makeText(this@AdminAkunActivity,
                    data[0].message_response,
                    Toast.LENGTH_SHORT).show()
                }
            } else{
                Toast.makeText(this@AdminAkunActivity,
                "Gagal di web", Toast.LENGTH_SHORT).show()
            }
        }

        private fun setFailureTambahAkun(message:
        String) {
            Toast.makeText(this@AdminAkunActivity,
            message, Toast.LENGTH_SHORT).show()
            loading.alertDialogCancel()
        }

        private fun fetchSemuaAkun() {
            viewModel.fetchAkun()
        }

        private fun getSemuaAkun(){

        viewModel.getAkun().observe(this@AdminAkunAc
        tivity){result->
            when(result){
                is UIState.Loading-> setStartShimmer()
                is UIState.Failure->
                setFailureFetchAkun(result.message)
                is UIState.Success->

```

```

setSuccessFetchAkun(result.data)
    }
}
}

private fun setFailureFetchAkun(message: String)
{
    setStopShimmer()
    Toast.makeText(this@AdminAkunActivity,
message, Toast.LENGTH_SHORT).show()
}

private fun setSuccessFetchAkun(data:
ArrayList<UsersModel>) {
    setStopShimmer()
    if(data.isEmpty()){
        setAdapter(data)
    } else{
        Toast.makeText(this@AdminAkunActivity,
"tidak ada data", Toast.LENGTH_SHORT).show()
    }
}

private fun setAdapter(data:
ArrayList<UsersModel>) {
    binding.apply {
        adapter = AdminAkunAdapter(data, object:
OnClickItem.ClickAkun{
            override fun clickItemSetting(akun:
UsersModel, it: View) {
                val popupMenu =
PopupMenu(this@AdminAkunActivity, it)

                popupMenu.inflate(R.menu.popup_edit_hapus)

                popupMenu.setOnMenuItemClickListener(object :
PopupMenu.OnMenuItemClickListener {
                    override fun
onMenuItemClick(menuItem: MenuItem?): Boolean
{
                        when (menuItem!!.itemId) {
                            R.id.edit -> {
                                setShowDialogEdit(akun)
                                return true
                            }
                            R.id.hapus -> {
                                setShowDialogHapus(akun)
                                return true
                            }
                        }
                        return true
                    }
                })
                popupMenu.show()
            }
        })
    }

    binding.apply {
        rvAkun.layoutManager =
LinearLayoutManager(this@AdminAkunActivity,
LinearLayoutManager.VERTICAL, false)
        rvAkun.adapter = adapter
    }
}

```

```

    }
}

private fun setShowDialogEdit(akun:
UsersModel) {
    val view =
AlertDialogAkunBinding.inflate(layoutInflater)

    val alertDialog =
AlertDialog.Builder(this@AdminAkunActivity)
        alertDialog.setView(view.root)
        .setCancelable(true)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

    tempAlertDialog = dialogInputan

    view.apply {
        etEditNama.setText(akun.nama)
        etEditNomorHp.setText(akun.nomorHp)
        etEditUsername.setText(akun.username)
        etEditPassword.setText(akun.password)

        btnSimpan.setOnClickListener {
            val idUser = akun.idUser!!
            val nama =
etEditNama.text.toString().trim()
            val nomorHp =
etEditNomorHp.text.toString().trim()
            val username =
etEditUsername.text.toString().trim()
            val password =
etEditPassword.text.toString().trim()
            val usernameLama = akun.username!!
            postUpdateUser(idUser, nama, nomorHp,
username, password, usernameLama)
        }
        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
            tempAlertDialog = null
        }
    }

    private fun postUpdateUser(
        idUser: String, nama: String, nomorHp: String,
        username: String, password: String,
        usernameLama: String
    ) {
        viewModel.postUpdateAkun(idUser, nama,
        nomorHp, username, password, usernameLama)
    }

    private fun getUpdateUser(){

    viewModel.getUpdateAkun().observe(this@Admin
AkunActivity){result->
        when(result){
            is UIState.Loading->
loading.alertDialogLoading(this@AdminAkunActiv
ity)
            is UIState.Failure->
setFailureUpdateAkun(result.message)
        }
    }
}

```

```

        is UIState.Success->
        setSuccessUpdateAkun(result.data)
    }
}

private fun setSuccessUpdateAkun(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

            Toast.makeText(this@AdminAkunActivity,
            "Berhasil Update", Toast.LENGTH_SHORT).show()
            tempAlertDialog!!.dismiss()
            tempAlertDialog = null
            fetchSemuaAkun()
        } else{

            Toast.makeText(this@AdminAkunActivity,
            data[0].message_response,
            Toast.LENGTH_SHORT).show()
        }
    } else{
        Toast.makeText(this@AdminAkunActivity,
        "Gagal di web", Toast.LENGTH_SHORT).show()
    }
}

private fun setFailureUpdateAkun(message:
String) {
    Toast.makeText(this@AdminAkunActivity,
    message, Toast.LENGTH_SHORT).show()
    loading.alertDialogCancel()
}

private fun setShowDialogHapus(akun:
UsersModel) {
    val view =
AlertDialogKonfirmasiBinding.inflate(layoutInflater
)

    val alertDialog =
AlertDialog.Builder(this@AdminAkunActivity)
        alertDialog.setView(view.root)
        .setCancelable(true)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

    view.apply {
        tvTitleKonfirmasi.text = "Hapus
${akun.nama!!} ?"
        tvBodyKonfirmasi.text = "Akun ini akan
hapus dan data tidak dapat dipulihkan"

        btnKonfirmasi.setOnClickListener {
            val idUser = akun.idUser!!
            postHapusUser(idUser)
            dialogInputan.dismiss()
        }
        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

```

```

private fun postHapusUser(idUser: String){
    viewModel.postDeleteAkun(idUser!!)
}

private fun getHapusUser(){

viewModel.getDeleteAkun().observe(this@AdminA
kunActivity){result->
    when(result){
        is UIState.Loading->
        loading.alertDialogLoading(this@AdminAkunActiv
ity)
        is UIState.Failure->
        setFailureHapusAkun(result.message)
        is UIState.Success->
        setSuccessHapusAkun(result.data)
    }
}

private fun setSuccessHapusAkun(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

            Toast.makeText(this@AdminAkunActivity,
            "Berhasil Hapus", Toast.LENGTH_SHORT).show()
            fetchSemuaAkun()
        } else{

            Toast.makeText(this@AdminAkunActivity,
            data[0].message_response,
            Toast.LENGTH_SHORT).show()
        }
    } else{
        Toast.makeText(this@AdminAkunActivity,
        "Gagal di web", Toast.LENGTH_SHORT).show()
    }
}

private fun setFailureHapusAkun(message:
String) {
    Toast.makeText(this@AdminAkunActivity,
    message, Toast.LENGTH_SHORT).show()
    loading.alertDialogCancel()
}

private fun setStopShimmer(){
    binding.apply {
        rvAkun.visibility = View.VISIBLE

        smAkun.stopShimmer()
        smAkun.visibility = View.GONE
    }
}

private fun setStartShimmer(){
    binding.apply {
        smAkun.startShimmer()
        smAkun.visibility = View.VISIBLE

        rvAkun.visibility = View.GONE
    }
}

```

```
}
```

Admin halamangaleriherbal activity

```
package  
com.example.edukasipengobatanpadapenderitastrok  
e.ui.activity.admin.galeri_herbal.halaman_galeri_herbal
```

```
import android.Manifest  
import android.annotation.SuppressLint  
import android.app.Activity  
import android.content.Intent  
import android.content.pm.PackageManager  
import android.net.Uri  
import android.os.Build  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.os.Environment  
import android.provider.OpenableColumns  
import android.provider.Settings  
import android.util.Log  
import android.view.MenuItem  
import android.view.View  
import android.widget.PopupMenu  
import android.widget.Toast  
import androidx.activity.viewModels  
import androidx.appcompat.app.AlertDialog  
import androidx.core.app.ActivityCompat  
import androidx.core.content.ContextCompat  
import  
import androidx.recyclerview.widget.LinearLayoutManager  
import com.bumptech.glide.Glide  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.R  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.adapter.admin.AdminHalamanGaleriHerbalAdapte  
r  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.data.model.ResponseModel  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.data.model.GaleriHerbalMainModel  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.databinding.ActivityAdminHalamanGaleriHerbalB  
inding  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.databinding.AlertDialogHalGaleriHerbalBinding  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.databinding.AlertDialogKonfirmasiBinding  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.databinding.AlertDialogShowImageBinding  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.ui.activity.admin.galeri_herbal.value_galeri_herbal  
.AdminValueGaleriHerbalActivity  
import
```

```
com.example.edukasipengobatanpadapenderitastrok  
e.ui.activity.admin.main.AdminMainActivity  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.utils.Constant  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.utils.KataAcak  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.utils.KontrolNavigationDrawer  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.utils.LoadingAlertDialog  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.utils.OnClickItem  
import  
com.example.edukasipengobatanpadapenderitastrok  
e.utils.network.UISState  
import dagger.hilt.android.AndroidEntryPoint  
import  
okhttp3.MediaType.Companion.toMediaTypeOrNull  
import okhttp3.MultipartBody  
import okhttp3.RequestBody  
import  
okhttp3.RequestBody.Companion.toRequestBody  
import javax.inject.Inject
```

```
@AndroidEntryPoint  
class AdminHalamanGaleriHerbalActivity :  
AppCompatActivity() {  
    private lateinit var binding:  
ActivityAdminHalamanGaleriHerbalBinding  
    private lateinit var kontrolNavigationDrawer:  
KontrolNavigationDrawer  
    private val viewModel:  
AdminHalamanGaleriHerbalViewModel by  
viewModels()  
    private lateinit var adapter:  
AdminHalamanGaleriHerbalAdapter  
    private var tempAlertDialog: AlertDialog? = null  
    @Inject  
lateinit var loading: LoadingAlertDialog  
    private var tempView:  
AlertDialogHalGaleriHerbalBinding? = null  
    private var fileImage: MultipartBody.Part? = null  
    private val kataAcak = KataAcak()  
    override fun onCreate(savedInstanceState:  
Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding =  
ActivityAdminHalamanGaleriHerbalBinding.inflate(  
layoutInflater)  
        setContentView(binding.root)  
  
        setKontrolNavigationDrawer()  
        setButton()  
        fetchGaleriHerbal()  
        getGaleriHerbal()  
        getTambahHalamanGaleriHerbal()  
        getUpdateHalamanGaleriHerbal()  
        getHapusHalamanGaleriHerbal()  
    }  
}
```

```

        private fun setKontrolNavigationDrawer() {
            kontrolNavigationDrawer =
            KontrolNavigationDrawer(this@AdminHalamanGaleriHerbalActivity)
            binding.apply {

            kontrolNavigationDrawer.cekSebagai(navView)

            kontrolNavigationDrawer.onClickItemNavigationDrawer(navView, drawerLayoutMain, ivDrawerView, this@AdminHalamanGaleriHerbalActivity)
            }
        }

        // tambah
        private fun setButton() {
            binding.btnTambah.setOnClickListener {
                dialogTambahHalamanGaleriHerbal()
            }
        }

        // menampilkan halaman tambah data
        private fun dialogTambahHalamanGaleriHerbal() {
            {
                val view =
                AlertDialogHalGaleriHerbalBinding.inflate(layoutInflater)

                val alertDialog =
                AlertDialog.Builder(this@AdminHalamanGaleriHerbalActivity)
                alertDialog.setView(view.root)
                .setCancelable(false)
                val dialogInputan = alertDialog.create()
                dialogInputan.show()

                tempAlertDialog = dialogInputan
                tempView = view

                view.apply {
                    // button mengganti gambar
                    etEditGambar.setOnClickListener {
                        if(checkPermission()){
                            pickImageFile()
                        } else{
                            requestPermission()
                        }
                    }
                    // button simpan
                    btnSimpan.setOnClickListener {
                        var check = true

                        if(etEditGambar.text.toString().trim() ==
                        "Masukkan Gambar".trim()){
                            etEditGambar.error = "Tidak Boleh
                        Kosong"
                            check = false
                        }
                        if(etEditJudul.text.isEmpty()){
                            etEditJudul.error = "Tidak Boleh
                        Kosong"
                            check = false
                        }
                        if(etEditDeskripsi.text.isEmpty()){

```

```

                            etEditDeskripsi.error = "Tidak Boleh
                        Kosong"
                            check = false
                        }
                    }
                    if(check){
                        val kata =
                        kataAcak.getHurufDanAngka()
                        val judul = etEditJudul.text.toString()
                        val deskripsi =
                        etEditDeskripsi.text.toString()
                        postTambahHalamanGaleriHerbal(kata,
                        judul, deskripsi, fileImage!!)
                    }
                }
                // button batal
                btnBatal.setOnClickListener {
                    dialogInputan.dismiss()
                    tempAlertDialog = null
                }
            }
        }

        // mengirim data ke database
        private fun postTambahHalamanGaleriHerbal(
            kataAcak: String, judul: String, deskripsi:
            String, fileImage: MultipartBody.Part
        ) {
            viewModel.postTambahHalamanGaleriHerbal(
            convertStringToMultipartBody("post_tambah_hal_g
            aleri_herbal"),
            convertStringToMultipartBody(kataAcak),
            convertStringToMultipartBody(judul),
            convertStringToMultipartBody(deskripsi),
            fileImage
        )
        }

        private fun getTambahHalamanGaleriHerbal(){
            viewModel.getResponseTambahHalamanGaleriHerbal().observe(this@AdminHalamanGaleriHerbalActivity){result->
                when(result){
                    is UIState.Loading->
                    loading.alertDialogLoading(this@AdminHalamanGaleriHerbalActivity)
                    is UIState.Failure->
                    setFailureTambahHalamanGaleriHerbal(result.message)
                    is UIState.Success->
                    setSuccessHalamanTambahHalamanGaleriHerbal(result.data)
                }
            }
        }

        private fun
        setFailureTambahHalamanGaleriHerbal(message:
        String) {
            loading.alertDialogCancel()

            Toast.makeText(this@AdminHalamanGaleriHerbalActivity, message, Toast.LENGTH_SHORT).show()
        }
    }
}

```

```

        private fun
        setSuccessHalamTambahanGaleriHerbal(data:
        ArrayList<ResponseModel>) {
            loading.alertDialogCancel()
            if(data.isNotEmpty()){
                if(data[0].status=="0"){

                    Toast.makeText(this@AdminHalamanGaleriHerbal
                    Activity, "Berhasil Tambah",
                    Toast.LENGTH_SHORT).show()
                    tempAlertDialog!!.dismiss()
                    tempAlertDialog = null
                    tempView = null
                    fetchGaleriHerbal()
                } else{

                    Toast.makeText(this@AdminHalamanGaleriHerbal
                    Activity, data[0].message_response,
                    Toast.LENGTH_SHORT).show()
                }
            } else{

                Toast.makeText(this@AdminHalamanGaleriHerbal
                Activity, "ada error di aplikasi",
                Toast.LENGTH_SHORT).show()
            }
        }

        private fun fetchGaleriHerbal() {
            viewModel.fetchGaleriHerbal()
        }

        private fun getGaleriHerbal(){

            viewModel.getGaleriHerbal().observe(this@Admin
            HalamanGaleriHerbalActivity){result->
                when(result){
                    is UIState.Loading-> setStartShimmer()
                    is UIState.Failure->
                    setFailureGaleriHerbal(result.message)
                    is UIState.Success->
                    setSuccessGaleriHerbal(result.data)
                }
            }
        }

        private fun setFailureGaleriHerbal(message:
        String) {
            setStopShimmer()

            Toast.makeText(this@AdminHalamanGaleriHerbal
            Activity, message, Toast.LENGTH_SHORT).show()
        }

        private fun setSuccessGaleriHerbal(data:
        ArrayList<GaleriHerbalMainModel>) {
            setStopShimmer()
            if(data.isNotEmpty()){
                setAdapter(data)
            } else{

                Toast.makeText(this@AdminHalamanGaleriHerbal
                Activity, "Data Kosong",
                Toast.LENGTH_SHORT).show()
            }
        }

```

```

        }

        // menampilkan semua data
        private fun setAdapter(data:
        ArrayList<GaleriHerbalMainModel>) {
            adapter =
            AdminHalamanGaleriHerbalAdapter(data, object :
            OnClickItem.ClickAdminHalamanGaleriHerbal{
                override fun clickItemGambar(gambar:
                String, title: String, it: View) {
                    setShowImage(gambar, title)
                }

                override fun
                clickItemNextPage(galeriHerbal:
                GaleriHerbalMainModel, it: View) {
                    val i =
                    Intent(this@AdminHalamanGaleriHerbalActivity,
                    AdminValueGaleriHerbalActivity::class.java)
                    i.putExtra("galeri_herbal_main", data)
                    i.putExtra("id_hal_galeri_herbal",
                    galeriHerbal.id_hal_galeri_herbal)
                    i.putExtra("judul", galeriHerbal.judul)
                    startActivity(i)
                }

                override fun clickItemSetting(galeriHerbal:
                GaleriHerbalMainModel, it: View) {
                    val popupMenu =
                    PopupMenu(this@AdminHalamanGaleriHerbalActi
                    vity, it)

                    popupMenu.inflate(R.menu.popup_rincian_edit_hap
                    us)

                    popupMenu.setOnMenuItemClickListener(object :
                    PopupMenu.OnMenuItemClickListener
                    {
                        override fun
                        onMenuItemClick(menuItem: MenuItem?): Boolean
                        {
                            when (menuItem!!.itemId) {
                                R.id.rincian -> {
                                    val i =
                                    Intent(this@AdminHalamanGaleriHerbalActivity,
                                    AdminValueGaleriHerbalActivity::class.java)
                                    i.putExtra("galeri_herbal_main",
                                    data)

                                    i.putExtra("id_hal_galeri_herbal",
                                    galeriHerbal.id_hal_galeri_herbal)
                                    i.putExtra("judul",
                                    galeriHerbal.judul)
                                    startActivity(i)
                                    return true
                                }

                                // button edit data
                                R.id.edit -> {

                                    dialogEditHalamanGaleriHerbal(galeriHerbal)
                                    return true
                                }

                                // button hapus data
                                R.id.hapus -> {

```

```

        dialogHapusHalamanGaleriHerbal(galeriHerbal.id_h
al_galeri_herbal!!, galeriHerbal.judul!!)
            return true
        }
    }
    return true
}

})
popupMenu.show()
}

})

binding.apply {
    rvHalGaleriHerbal.layoutManager =
LinearLayoutManager(this@AdminHalamanGaleri
HerbalActivity, LinearLayoutManager.VERTICAL,
false)
    rvHalGaleriHerbal.adapter = adapter
}

// menampilkan gambar
private fun setShowImage(gambar: String,
title:String) {
    val view =
AlertDialogShowImageBinding.inflate(layoutInflate
r)

    val alertDialog =
AlertDialog.Builder(this@AdminHalamanGaleriHer
balActivity)
        alertDialog.setView(view.root)
        .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

    view.apply {
        tvTitle.text = "Gambar $title"
        btnClose.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

Glide.with(this@AdminHalamanGaleriHerbalActivi
ty)

.load("${Constant.BASE_URL}${Constant.LOCAT
ION_GAMBAR}$gambar") // URL Gambar
    .error(R.drawable.gambar_error_image)
    .into(view.ivShowImage) // imageView
mana yang akan diterapkan

}

// menampilkan edit data
private fun
dialogEditHalamanGaleriHerbal(galeriHerbal:
GaleriHerbalMainModel) {
    val view =
AlertDialogHalGaleriHerbalBinding.inflate(layoutIn
flater)

```

```

        val alertDialog =
AlertDialog.Builder(this@AdminHalamanGaleriHer
balActivity)
        alertDialog.setView(view.root)
        .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        view.tvTitle.text = "Update Halaman Galeri
Herbal"

        tempAlertDialog = dialogInputan
        tempView = view

        view.apply {
            etEditJudul.setText(galeriHerbal.judul)
        }
        etEditDeskripsi.setText(galeriHerbal.deskripsi)

        etEditGambar.setOnClickListener {
            if(checkPermission()){
                pickImageFile()
            } else{
                requestPermission()
            }
        }

        // button simpan
        btnSimpan.setOnClickListener {
            var check = true

            if(etEditJudul.text.isEmpty()){
                etEditJudul.error = "Tidak Boleh
Kosong"
                check = false
            }
            if(etEditDeskripsi.text.isEmpty()){
                etEditDeskripsi.error = "Tidak Boleh
Kosong"
                check = false
            }
            if(check){
                if(fileImage != null){
                    postEditHalamanGaleriHerbal(
galeriHerbal.id_hal_galeri_herbal!!,
kataAcak.getHurufDanAngka(),
                    etEditJudul.text.toString(),
                    etEditDeskripsi.text.toString(),
                    fileImage!!
                )
            } else{
                postEditHalamanGaleriHerbalNoHaveImage(
galeriHerbal.id_hal_galeri_herbal!!,
                    etEditJudul.text.toString(),
                    etEditDeskripsi.text.toString()
                )
            }
        }
        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

```

```

        tempAlertDialog = null
    }
}

private fun postEditHalamanGaleriHerbal(
    idHalGaleriHerbal: String, kataAcak: String,
    judul: String, deskripsi: String, fileImage:
    MultipartBody.Part
) {
    viewModel.postUpdateHalamanGaleriHerbal(
        convertStringToMultipartBody("post_update_hal_ga
        leri_herbal"),
        convertStringToMultipartBody(idHalGaleriHerbal),
        convertStringToMultipartBody(kataAcak),
        convertStringToMultipartBody(judul),
        convertStringToMultipartBody(deskripsi),
        fileImage
    )
}

private fun
postEditHalamanGaleriHerbalNoHaveImage(
    idHalGaleriHerbal: String, judul: String,
    deskripsi: String
) {
    viewModel.postUpdateHalamanGaleriHerbalNoHaveImage(
        "post_update_hal_galeri_herbal_no_have_image",
        idHalGaleriHerbal, judul, deskripsi
    )
}

private fun getUpdateHalamanGaleriHerbal(){
    viewModel.getResponseUpdateHalamanGaleriHerbal().observe(this@AdminHalamanGaleriHerbalActivity){result->
        when(result){
            is UIState.Loading->
            loading.alertDialogLoading(this@AdminHalamanGaleriHerbalActivity)
            is UIState.Failure->
            setFailureUpdateHalamanGaleriHerbal(result.message)
            is UIState.Success->
            setSuccessHalamUpdateanGaleriHerbal(result.data)
        }
    }
}

private fun
setFailureUpdateHalamanGaleriHerbal(message:
String) {
    loading.alertDialogCancel()

    Toast.makeText(this@AdminHalamanGaleriHerbalActivity, message, Toast.LENGTH_SHORT).show()
}

private fun

```

```

setSuccessHalamUpdateanGaleriHerbal(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

            Toast.makeText(this@AdminHalamanGaleriHerbalActivity, "Berhasil Update",
            Toast.LENGTH_SHORT).show()
            tempAlertDialog!!.dismiss()
            tempAlertDialog = null
            fetchGaleriHerbal()
        } else{

            Toast.makeText(this@AdminHalamanGaleriHerbalActivity, data[0].message_response,
            Toast.LENGTH_SHORT).show()
        }
    } else{

        Toast.makeText(this@AdminHalamanGaleriHerbalActivity, "ada error di aplikasi",
        Toast.LENGTH_SHORT).show()
    }
}

// menampilkan hapus data
private fun
dialogHapusHalamanGaleriHerbal(idHalGaleriHerbal: String, halaman: String) {
    val view =
    AlertDialogKonfirmasiBinding.inflate(layoutInflater
    )

    val alertDialog =
    AlertDialog.Builder(this@AdminHalamanGaleriHerbalActivity)
        alertDialog.setView(view.root)
        .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

    tempAlertDialog = dialogInputan

    view.apply {
        tvTitleKonfirmasi.text = "Yakin Hapus halaman ini?"
        tvBodyKonfirmasi.text = "Halaman $halaman akan dihapus dan data yang ada di dalamnya akan ikut terhapus. data yang telah terhapus tidak dapat di kembalikan"

        btnKonfirmasi.setOnClickListener {

            postHapusHalamanGaleriHerbal(idHalGaleriHerbal!)
        }
        btnBatal.setOnClickListener {
            tempAlertDialog = null
            dialogInputan.dismiss()
        }
    }
}

private fun

```

```

postHapusHalamanGaleriHerbal(idHalGaleriHerbal:
String) {

viewModel.postHapusHalamanGaleriHerbal(idHalG
aleriHerbal)
}

private fun getHapusHalamanGaleriHerbal(){

viewModel.getResponseHapusHalamanGaleriHerbal
().observe(this@AdminHalamanGaleriHerbalActivit
y){result->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@AdminHalamanG
aleriHerbalActivity)
        is UIState.Failure->
setFailureHapusHalamanGaleriHerbal(result.messag
e)
        is UIState.Success->
setSuccessHapusHalamanGaleriHerbal(result.data)
    }
}

private fun
setFailureHapusHalamanGaleriHerbal(message:
String) {
    loading.alertDialogCancel()

Toast.makeText(this@AdminHalamanGaleriHerbal
Activity, message, Toast.LENGTH_SHORT).show()
}

private fun
setSuccessHapusHalamanGaleriHerbal(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

Toast.makeText(this@AdminHalamanGaleriHerbal
Activity, "Berhasil Hapus",
Toast.LENGTH_SHORT).show()
            tempAlertDialog!!.dismiss()
            tempAlertDialog = null
            fetchGaleriHerbal()
        } else{

Toast.makeText(this@AdminHalamanGaleriHerbal
Activity, data[0].message_response,
Toast.LENGTH_SHORT).show()
        }
    } else{

Toast.makeText(this@AdminHalamanGaleriHerbal
Activity, "ada error di aplikasi",
Toast.LENGTH_SHORT).show()
    }
}

private fun setStopShimmer(){
    binding.apply {
        rvHalGaleriHerbal.visibility = View.VISIBLE

```

```

        smHalGaleriHerbal.stopShimmer()
        smHalGaleriHerbal.visibility = View.GONE
    }
}

private fun setStartShimmer(){
    binding.apply {
        smHalGaleriHerbal.startShimmer()
        smHalGaleriHerbal.visibility =
View.VISIBLE

        rvHalGaleriHerbal.visibility = View.GONE
    }
}

// melakukan mengizinkan sistem untuk
mengambil gambar
private fun requestPermission(){
    if(Build.VERSION.SDK_INT >=
Build.VERSION_CODES.R){
        if
(Environment.isExternalStorageManager()) {
            startActivity(Intent(this,
AdminHalamanGaleriHerbalAdapter::class.java))
        } else { //request for the permission
            val intent =
Intent(Settings.ACTION_MANAGE_APP_ALL_FIL
ES_ACCESS_PERMISSION)
            val uri = Uri.fromParts("package",
packageName, null)
            intent.data = uri
            startActivity(intent)
        }
    } else{
        ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.WRITE_EXTERNAL_
STORAGE,
Manifest.permission.READ_EXTERNAL_STORAGE
),
Constant.STORAGE_PERMISSION_CODE
        )
    }
}

// mengambil gambar
private fun pickImageFile() {
    val intent =
Intent(Intent.ACTION_OPEN_DOCUMENT).apply
{
        type = "image/*"
    }

    startActivityForResult(intent,
Constant.IMAGE_CODE)
}

override fun onActivityResult(requestCode: Int,
resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode,
resultCode, data)

    if (requestCode == Constant.IMAGE_CODE

```

```

&& resultCode == Activity.RESULT_OK && data
!= null) {
    // Mendapatkan URI file PDF yang dipilih
    val fileUri = data.data!!

    val nameImage = getNameFile(fileUri)

    tempView?.let {
        it.etEditGambar.text = nameImage
    }

    // Mengirim file PDF ke website
    menggunakan Retrofit
    fileImage = uploadImageToStorage(fileUri,
    nameImage, "galeri_herbal_image")
    }
}

private fun getNameFile(uri: Uri): String {
    val cursor = contentResolver.query(uri, null,
    null, null, null)
    val nameIndex =
    cursor?.getColumnIndex(OpenableColumns.DISPL
    AY_NAME)
    cursor?.moveToFirst()
    val name = cursor?.getString(nameIndex!!)
    cursor?.close()
    return name!!
}

@SuppressLint("Recycle")
private fun uploadImageToStorage(imageUri:
Uri?, fileName: String, nameAPI:String):
MultipartBody.Part? {
    var pdfPart : MultipartBody.Part? = null
    imageUri?.let {
        val file =
        contentResolver.openInputStream(imageUri)?.readB
        ytes()

        if (file != null) {
            // Membuat objek RequestBody dari file
            PDF
            val requestFile =
            file.toRequestBody("application/pdf".toMediaType
            OrNull())
            // Membuat objek MultipartBody.Part
            untuk file PDF
            pdfPart =
            MultipartBody.Part.createFormData("materi_pdf",
            fileName, requestFile)

            pdfPart =
            convertFileToMultipartBody(file, fileName,
            nameAPI)
        }
    }
    return pdfPart
}

private fun convertFileToMultipartBody(file:
ByteArray, fileName: String, nameAPI:String):
MultipartBody.Part?{
    // val requestFile =
    file.toRequestBody("application/pdf".toMediaType

```

```

OrNull())
    val requestFile =
    file.toRequestBody("application/image".toMediaType
    OrNull())
    val filePart =
    MultipartBody.Part.createFormData(nameAPI,
    fileName, requestFile)

    return filePart
}

private fun convertStringToMultipartBody(data:
String): RequestBody {
    return RequestBody.create("multipart/form-
    data".toMediaTypeOrNull(), data)
}

private fun checkPermission(): Boolean{
    return if (Build.VERSION.SDK_INT >=
    Build.VERSION_CODES.R){
        //Android is 11(R) or above
        Environment.isExternalStorageManager()
    }
    else{
        //Android is below 11(R)
        val write =
        ContextCompat.checkSelfPermission(this,
        Manifest.permission.WRITE_EXTERNAL_STORAG
        E)

        val read =
        ContextCompat.checkSelfPermission(this,
        Manifest.permission.READ_EXTERNAL_STORAGE
        )

        write ==
        PackageManager.PERMISSION_GRANTED &&
        read == PackageManager.PERMISSION_GRANTED
    }
}

override fun onBackPressed() {
    super.onBackPressed()

    startActivity(Intent(this@AdminHalamanGaleriHerb
    alActivity, AdminMainActivity::class.java))
    finish()
}
}

```

Admin halamangaleriherbal viewmodel

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.galeri_herbal.halaman_galeri_her
bal

import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import
com.example.edukasipengobatanpadapenderitastrok
e.data.database.api.ApiService
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalMainModel

```

```

import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.delay
import kotlinx.coroutines.launch
import okhttp3.MultipartBody
import okhttp3.RequestBody
import javax.inject.Inject

@HiltViewModel
class AdminHalamanGaleriHerbalViewModel
@Inject constructor(
    private val api: ApiService
): ViewModel() {
    private val _galeriHerbal =
MutableLiveData<UISState<ArrayList<GaleriHerbal
MainModel>>>()
    private val _postTambahHalamanGaleriHerbal =
MutableLiveData<UISState<ArrayList<ResponseMo
del>>>()
    private val _postUpdateHalamanGaleriHerbal =
MutableLiveData<UISState<ArrayList<ResponseMo
del>>>()
    private val _postHapusHalamanGaleriHerbal =
MutableLiveData<UISState<ArrayList<ResponseMo
del>>>()

    fun fetchGaleriHerbal() {
        viewModelScope.launch {
            _galeriHerbal.postValue(UISState.Loading)
            delay(1_000)
            try {
                val data = api.getGaleriHerbalMain("")
            } catch (ex: Exception) {
                _galeriHerbal.postValue(UISState.Success(data))
            }

            _galeriHerbal.postValue(UISState.Failure("Error
${ex.message}"))
        }
    }

    fun postTambahHalamanGaleriHerbal(
        post: RequestBody, kataAcak: RequestBody,
        judul: RequestBody, deskripsi: RequestBody,
        gambar: MultipartBody.Part
    ){
        viewModelScope.launch(Dispatchers.IO) {

            _postTambahHalamanGaleriHerbal.postValue(UISSta
te.Loading)
            delay(1_000)
            try {
                val postTambahHalamanGaleriHerbal =
api.addHalamanGaleriHerbal(post, kataAcak, judul,
deskripsi, gambar)

                _postTambahHalamanGaleriHerbal.postValue(UISSta
te.Success(postTambahHalamanGaleriHerbal))

```

```

            } catch (ex: Exception){

            _postTambahHalamanGaleriHerbal.postValue(UISSta
te.Failure("Error: ${ex.message}"))
        }
    }

    fun postUpdateHalamanGaleriHerbal(
        post: RequestBody, id_hal_galeri_herbal:
RequestBody, kataAcak: RequestBody, judul:
RequestBody, deskripsi: RequestBody, gambar:
MultipartBody.Part
    ){
        viewModelScope.launch(Dispatchers.IO) {

            _postUpdateHalamanGaleriHerbal.postValue(UISStat
e.Loading)
            delay(1_000)
            try {
                val postTambahHalamanGaleriHerbal =
api.postUpdateHalamanGaleriHerbal(
                    post, id_hal_galeri_herbal, kataAcak,
                    judul, deskripsi, gambar
                )

                _postUpdateHalamanGaleriHerbal.postValue(UISStat
e.Success(postTambahHalamanGaleriHerbal))
            } catch (ex: Exception){

            _postUpdateHalamanGaleriHerbal.postValue(UISStat
e.Failure("Error: ${ex.message}"))
        }
    }

    fun
postUpdateHalamanGaleriHerbalNoHaveImage(
        post: String, id_hal_galeri_herbal: String, judul:
String, deskripsi: String
    ){
        viewModelScope.launch(Dispatchers.IO) {

            _postUpdateHalamanGaleriHerbal.postValue(UISStat
e.Loading)
            delay(1_000)
            try {
                val postTambahHalamanGaleriHerbal =
api.postUpdateHalamanGaleriHerbalNoHaveImage(
                    post, id_hal_galeri_herbal, judul,
                    deskripsi
                )

                _postUpdateHalamanGaleriHerbal.postValue(UISStat
e.Success(postTambahHalamanGaleriHerbal))
            } catch (ex: Exception){

            _postUpdateHalamanGaleriHerbal.postValue(UISStat
e.Failure("Error: ${ex.message}"))
        }
    }

    fun
postHapusHalamanGaleriHerbal(id_hal_galeri_herb

```

```

al: String){
    viewModelScope.launch(Dispatchers.IO) {

        _postHapusHalamanGaleriHerbal.postValue(UIState
.Loading)
            delay(1_000)
            try {
                val postTambahHalamanGaleriHerbal =
api.postHapusHalamanGaleriHerbal("",
id_hal_galeri_herbal)

                _postHapusHalamanGaleriHerbal.postValue(UIState
.Success(postTambahHalamanGaleriHerbal))
            } catch (ex: Exception){

                _postHapusHalamanGaleriHerbal.postValue(UIState
.Failure("Error: ${ex.message}"))
            }
        }

        fun getGaleriHerbal():
LiveData<UIState<ArrayList<GaleriHerbalMainMo
del>>> = _galeriHerbal
            fun getResponseTambahHalamanGaleriHerbal():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postTambahHalamanGaleriHerbal
            fun getResponseUpdateHalamanGaleriHerbal():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postUpdateHalamanGaleriHerbal
            fun getResponseHapusHalamanGaleriHerbal():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postHapusHalamanGaleriHerbal
        }
    }
}

```

adminvaluegaleriherbalactivity

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.galeri_herbal.value_galeri_herbal

import android.Manifest
import android.annotation.SuppressLint
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Environment
import android.provider.OpenableColumns
import android.provider.Settings
import android.util.Log
import android.view.MenuItem
import android.view.View
import android.widget.AdapterView
import android.widget.AdapterView
import android.widget.ArrayAdapter
import android.widget.PopupMenu
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import

```

```

androidx.recyclerview.widget.LinearLayoutManager
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin.AdminHalamanGaleriHerbalAdapte
r
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin.AdminValueGaleriHerbalAdapter
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalListModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalMainModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityAdminValueGaleriHerbalBin
ding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogHalGaleriHerbalBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKeteranganBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKonfirmasiBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogShowImageBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogValGaleriHerbalBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.galeri_herbal.halaman_galeri_her
bal.AdminHalamanGaleriHerbalActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KataAcak
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Youtube
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISate
import dagger.hilt.android.AndroidEntryPoint
import
okhttp3.MediaType.Companion.toMediaTypeOrNull

```

```

1
import okhttp3.MultipartBody
import okhttp3.RequestBody
import
import okhttp3.RequestBody.Companion.toRequestBody
import java.util.ArrayList
import javax.inject.Inject

@AndroidEntryPoint
class AdminValueGaleriHerbalActivity :
AppCompatActivity() {
    private lateinit var binding:
ActivityAdminValueGaleriHerbalBinding
    private val viewModel:
AdminValueGaleriHerbalViewModel by
viewModelS()
    private lateinit var adapter:
AdminValueGaleriHerbalAdapter
    @Inject
    lateinit var loading: LoadingAlertDialog
    private var idHalGaleriHerbal = "1"
    private lateinit var listHalamanGaleriHerbal:
ArrayList<String>
    private lateinit var listIdHalamanGaleriHerbal:
ArrayList<String>
    private var youtube = Youtube()
    private var tempAlertDialog: AlertDialog? = null
    private var tempView:
AlertDialogValGaleriHerbalBinding? = null
    private var fileImage: MultipartBody.Part? = null
    private var kataAcak = KataAcak()
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityAdminValueGaleriHerbalBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setDataSebelumnya()
        setButton()
        fetchData()
        getValueGaleriHerbal()
        getTambahValueGaleriHerbal()
        getUpdateValueGaleriHerbal()
        getHapusValueGaleriHerbal()
    }

    private fun setDataSebelumnya() {
        val extras = intent.extras
        if (extras != null) {
            var halGaleriHerbal =
extras.getParcelableArrayList<GaleriHerbalMainModel>("galeri_herbal_main")!!
            idHalGaleriHerbal =
extras.getString("id_hal_galeri_herbal")!!
            listHalamanGaleriHerbal =
searchDataArrayHalaman(halGaleriHerbal)
            listIdHalamanGaleriHerbal =
searchDataArrayId(halGaleriHerbal)
            val judul = extras.getString("judul")!!
            binding.titleHeader.text = judul
        }
    }

```

```

        private fun searchDataArrayHalaman(list:
ArrayList<GaleriHerbalMainModel>):
ArrayList<String> {
            val valueList: ArrayList<String> =
arrayListOf()
            for (value in list) {
                valueList.add(value.judul!!)
            }

            return valueList
        }

        private fun searchDataArrayId(list:
ArrayList<GaleriHerbalMainModel>):
ArrayList<String> {
            val valueList: ArrayList<String> =
arrayListOf()
            for (value in list) {
                valueList.add(value.id_hal_galeri_herbal!!)
            }

            return valueList
        }

        private fun setButton() {
            binding.apply {
                btnTambah.setOnClickListener {
                    setShowDialogTambah()
                }
                btnBack.setOnClickListener {
                    finish()
                }
            }
        }

        private fun setShowDialogTambah() {
            val view =
AlertDialogValGaleriHerbalBinding.inflate(layoutInflater)

            val alertDialog =
AlertDialog.Builder(this@AdminValueGaleriHerbalActivity)
                alertDialog.setView(view.root)
                    .setCancelable(false)
            val dialogInputan = alertDialog.create()
            dialogInputan.show()

            tempAlertDialog = dialogInputan
            tempView = view

            view.apply {

                var numberPosition = 0
                var selectedValue = ""

                val tambahDialogHalaman =
binding.titleHeader.text.toString().trim()
                val listTambahDialog : ArrayList<String> =
arrayListOf()

                listTambahDialog.add(tambahDialogHalaman)

```

```

        val arrayAdapter =
        ArrayAdapter(this@AdminValueGaleriHerbalActivi
        ty, android.R.layout.simple_spinner_item,
        listTambahDialog)

        arrayAdapter.setDropDownViewResource(android.
        R.layout.simple_spinner_dropdown_item)

        spHalamanGaleriHerbal.adapter =
        arrayAdapter

        spHalamanGaleriHerbal.onItemSelectedListener =
        object: AdapterView.OnItemSelectedListener{
            override fun onItemSelected(
                parent: AdapterView<*>?,
                view: View?,
                position: Int,
                id: Long
            ) {
                numberPosition =
                spHalamanGaleriHerbal.selectedItemPosition
                selectedValue =
                spHalamanGaleriHerbal.selectedItem.toString()
            }

            override fun onNothingSelected(parent:
            AdapterView<*>?) {

            }
        }

        spHalamanGaleriHerbal.adapter =
        arrayAdapter

        etEditGambar.setOnClickListener{
            if(checkPermission()){
                pickImageFile()
            } else{
                requestPermission()
            }
        }

        btnSimpan.setOnClickListener {
            var checkValue = true
            if(etEditGambar.text.toString() ==
            "Masukkan Gambar"){
                etEditGambar.error = "Masukkan
            gambar"
                checkValue = false
            }
            if(etEditJudul.text.isEmpty()){
                etEditJudul.error = "Masukkan gambar"
                checkValue = false
            }
            if(etEditDeskripsi.text.isEmpty()){
                etEditDeskripsi.error = "Masukkan
            gambar"
                checkValue = false
            }
        }

        if(etEditTataCaraPengolahan.text.isEmpty()){
            etEditTataCaraPengolahan.error =
            "Masukkan gambar"
            checkValue = false
        }

```

```

        }
        if(etEditLinkYoutube.text.isEmpty()){
            etEditLinkYoutube.error = "Masukkan
        gambar"
            checkValue = false
        }

        if(checkValue){
            val idHalGaleriHerbal =
            idHalGaleriHerbal
            val nama =
            etEditJudul.text.toString().trim()
            val judul =
            etEditJudul.text.toString().trim()
            val deskripsi =
            etEditDeskripsi.text.toString().trim()
            val tataCaraPengolahan =
            etEditTataCaraPengolahan.text.toString().trim()
            val linkYoutube =
            etEditLinkYoutube.text.toString().trim()

            postTambahValueGaleriHerbal(
                idHalGaleriHerbal, nama, judul,
                deskripsi, tataCaraPengolahan, fileImage!!,
                linkYoutube
            )
        }
        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
            tempAlertDialog = null
            tempView = null
        }
    }

    private fun postTambahValueGaleriHerbal(
        idHalGaleriHerbal: String, kataAcak: String,
        nama: String, deskripsi: String, tataCaraPengolahan:
        String,
        fileImage: MultipartBody.Part, linkYoutube:
        String,
    ) {
        viewModel.postTambahValueGaleriHerbal(
            convertStringToMultipartBody("post_tambah_val_g
            aleri_herbal"),
            convertStringToMultipartBody(idHalGaleriHerbal),
            convertStringToMultipartBody(kataAcak),
            convertStringToMultipartBody(nama),
            convertStringToMultipartBody(deskripsi),
            convertStringToMultipartBody(tataCaraPengolahan)
        ,
            fileImage,
            convertStringToMultipartBody(linkYoutube),
        )
    }

    private fun getTambahValueGaleriHerbal(){
        viewModel.getResponseTambahValueGaleriHerbal(
        ).observe(this@AdminValueGaleriHerbalActivity){r

```

```

result->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@AdminValueGaleriHerbalActivity)
        is UIState.Failure->
setFailureTambahValueGaleriHerbal(result.message)
    )
        is UIState.Success->
setSuccessTambahValueGaleriHerbal(result.data)
    }
}

private fun
setSuccessTambahValueGaleriHerbal(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

Toast.makeText(this@AdminValueGaleriHerbalActivity, "Berhasil Tambah",
Toast.LENGTH_SHORT).show()
            tempAlertDialog!!.dismiss()
            tempAlertDialog = null
            tempView = null
            fetchData()
        } else{

Toast.makeText(this@AdminValueGaleriHerbalActivity, data[0].message_response,
Toast.LENGTH_SHORT).show()
        }
    } else{

Toast.makeText(this@AdminValueGaleriHerbalActivity, "Gagal di web",
Toast.LENGTH_SHORT).show()
    }
}

private fun
setFailureTambahValueGaleriHerbal(message:
String) {

Toast.makeText(this@AdminValueGaleriHerbalActivity, message, Toast.LENGTH_SHORT).show()
    loading.alertDialogCancel()
}

private fun fetchData() {

viewModel.fetchGaleriHerbal(idHalGaleriHerbal)
}

private fun getValueGaleriHerbal(){

viewModel.getGaleriHerbal().observe(this@AdminValueGaleriHerbalActivity){result->
    when(result){
        is UIState.Loading-> setStartShimmer()
        is UIState.Failure->
setFailureFetchValueGaleriHerbal(result.message)
        is UIState.Success->
setSuccessFetchValueGaleriHerbal(result.data)
    }
}
}

```

```

    }
}

private fun
setFailureFetchValueGaleriHerbal(message: String)
{

Toast.makeText(this@AdminValueGaleriHerbalActivity, message, Toast.LENGTH_SHORT).show()
    setStopShimmer()
}

private fun
setSuccessFetchValueGaleriHerbal(data:
ArrayList<GaleriHerbalListModel>) {
    setStopShimmer()
    if(data.isNotEmpty()){
        setAdapter(data)
    } else{

Toast.makeText(this@AdminValueGaleriHerbalActivity, "Tidak ada data",
Toast.LENGTH_SHORT).show()
    }
}

private fun setAdapter(data:
ArrayList<GaleriHerbalListModel>) {
    adapter =
AdminValueGaleriHerbalAdapter(data, object:
OnClickItem.ClickAdminValueGaleriHerbal{
        override fun clickItemJudul(judul: String, it:
View) {
            showClickText("Judul Isi", judul)
        }

        override fun clickItemDeskripsi(deskripsi:
String, it: View) {
            showClickText("Isi Penjelasan", deskripsi)
        }

        override fun
clickItemTataCaraPengolahan(tataCaraPengolahan:
String, it: View) {
            showClickText("Tata Cara Pengolahan",
tataCaraPengolahan)
        }

        override fun clickItemGambar(gambar:
String, title: String, it: View) {
            setShowImage(gambar, title)
        }

        override fun clickItemYoutube(linkYoutube:
String, it: View) {

youtube.setToYoutubeVideo(this@AdminValueGaleriHerbalActivity, linkYoutube)
        }

        override fun clickItemSetting(galeriHerbal:
GaleriHerbalListModel, it: View) {
            val popupMenu =
PopupMenu(this@AdminValueGaleriHerbalActivity

```

```

, it)

popupMenu.inflate(R.menu.popup_edit_hapus)

popupMenu.setOnMenuItemClickListener(object :
    PopupMenu.OnMenuItemClickListener
{
    override fun
onMenuItemClick(menuItem: MenuItem?): Boolean
{
    when (menuItem!!.itemId) {
        R.id.edit -> {

setShowDialogEdit(galeriHerbal)
            return true
        }
        R.id.hapus -> {

setShowDialogHapus(galeriHerbal)
            return true
        }
    }
    return true
}

})
popupMenu.show()
}
})

binding.apply {
    rvValGaleriHerbal.layoutManager =
LinearLayoutManager(
    this@AdminValueGaleriHerbalActivity,
    LinearLayoutManager.VERTICAL,
    false
    )
    rvValGaleriHerbal.adapter = adapter
}

}

private fun showClickText(keterangan:String,
judul: String) {
    val view =
AlertDialogKeteranganBinding.inflate(layoutInflater
)

    val alertDialog =
AlertDialog.Builder(this@AdminValueGaleriHerbal
Activity)
        alertDialog.setView(view.root)
            .setCancelable(false)
        val dialogInputan = alertDialog.create()
            dialogInputan.show()

    view.apply {
        tvTitleKeterangan.text = keterangan
        tvBodyKeterangan.text = judul
        btnClose.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}
}

```

```

private fun setShowImage(gambar: String,
title:String) {
    val view =
AlertDialogShowImageBinding.inflate(layoutInflater
r)

    val alertDialog =
AlertDialog.Builder(this@AdminValueGaleriHerbal
Activity)
        alertDialog.setView(view.root)
            .setCancelable(false)
        val dialogInputan = alertDialog.create()
            dialogInputan.show()

    view.apply {
        tvTitle.text = "Gambar $title"
        btnClose.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

```

```

Glide.with(this@AdminValueGaleriHerbalActivity)

.load("${Constant.BASE_URL}${Constant.LOCAT
ION_GAMBAR}$gambar") // URL Gambar
    .error(R.drawable.gambar_error_image)
    .into(view.ivShowImage) // imageView
mana yang akan diterapkan

}

```

```

private fun setShowDialogEdit(tentangStroke:
GaleriHerbalListModel) {
    val view =
AlertDialogValGaleriHerbalBinding.inflate(layoutIn
flater)

    val alertDialog =
AlertDialog.Builder(this@AdminValueGaleriHerbal
Activity)
        alertDialog.setView(view.root)
            .setCancelable(false)
        val dialogInputan = alertDialog.create()
            dialogInputan.show()

    view.tvTitle.text = "Edit Value Galeri Herbal"

    tempAlertDialog = dialogInputan

    view.apply {
        etEditJudul.setText(tentangStroke.nama)

etEditDeskripsi.setText(tentangStroke.deskripsi)

etEditTataCaraPengolahan.setText(tentangStroke.tat
a_cara_pengolahan)

etEditLinkYoutube.setText(tentangStroke.youtube)

        var numberPosition = 0
        var selectedValue = ""

        val arrayAdapter =
ArrayAdapter(this@AdminValueGaleriHerbalActivi

```

```

ty, android.R.layout.simple_spinner_item,
listHalamanGaleriHerbal)

arrayAdapter.setDropDownViewResource(android.
R.layout.simple_spinner_dropdown_item)

        var id = 0
        for((no, value) in
listIdHalamanGaleriHerbal.withIndex()){
            if(value ==
tentangStroke.id_hal_galeri_herbal){
                id = no
                Log.d("DetailTAG", "val $no: $value")
            }
        }
        Log.d("DetailTAG", "val $id")

        spHalamanGaleriHerbal.setSelection(id)
        spHalamanGaleriHerbal.adapter =
arrayAdapter

spHalamanGaleriHerbal.onItemSelectedListener =
object: AdapterView.OnItemSelectedListener{
    override fun onItemSelected(
        parent: AdapterView<*>?,
        view: View?,
        position: Int,
        id: Long
    ) {
        numberPosition =
spHalamanGaleriHerbal.selectedItemPosition
        selectedValue =
spHalamanGaleriHerbal.selectedItem.toString()
    }

    override fun onNothingSelected(parent:
AdapterView<*>?) {

    }
}

spHalamanGaleriHerbal.setSelection(id)

etEditGambar.setOnClickListener{
    if(checkPermission()){
        pickImageFile()
    } else{
        requestPermission()
    }
}

btnSimpan.setOnClickListener {
    var checkValue = true

    if(etEditJudul.text.isEmpty()){
        etEditJudul.error = "Masukkan Judul"
        checkValue = false
    }
    if(etEditDeskripsi.text.isEmpty()){
        etEditJudul.error = "Masukkan
Deskripsi"
        checkValue = false
    }
}

```

```

if(etEditTataCaraPengolahan.text.isEmpty()){
    etEditJudul.error = "Masukkan Tata
Cara Pengolahan"
    checkValue = false
}
if(etEditLinkYoutube.text.isEmpty()){
    etEditJudul.error = "Masukkan Link
Youtube"
    checkValue = false
}

if(checkValue){
    val idValueGaleriHerbal =
tentangStroke.id_val_galeri_herbal!!
    val idHalGaleriHerbal =
listIdHalamanGaleriHerbal[numberPosition]
    val nama =
etEditJudul.text.toString().trim()
    val deskripsi =
etEditDeskripsi.text.toString().trim()
    val tataCaraPengolahan =
etEditTataCaraPengolahan.text.toString().trim()
    val linkYoutube =
etEditLinkYoutube.text.toString().trim()
    val valueKataAcak =
kataAcak.getHurufDanAngka()

    if(fileImage != null){
        postUpdateValueGaleriHerbal(
            idValueGaleriHerbal,
            idHalGaleriHerbal, valueKataAcak, nama, deskripsi,
            tataCaraPengolahan, fileImage!!, linkYoutube
        )
    } else{

postUpdateValueGaleriHerbalNoHaveImage(
            idValueGaleriHerbal,
            idHalGaleriHerbal, nama, deskripsi,
            tataCaraPengolahan, linkYoutube
        )
    }
}

btnBatal.setOnClickListener {
    dialogInputan.dismiss()
    tempAlertDialog!!.dismiss()
    tempAlertDialog = null
    tempView = null
}

private fun postUpdateValueGaleriHerbal(
    idValGaleriHerbal: String, idHalGaleriHerbal:
String, kataAcak: String, nama: String, deskripsi:
String,
    tataCaraPengolahan: String, fileImage:
MultipartBody.Part, linkYoutube: String,
    ){
    viewModel.postUpdateValueGaleriHerbal(
convertStringToMultipartBody("post_tambah_val_g
aleri_herbal"),

convertStringToMultipartBody(idValGaleriHerbal),

```

```

convertStringToMultipartBody(idHalGaleriHerbal),
    convertStringToMultipartBody(kataAcak),
    convertStringToMultipartBody(nama),
    convertStringToMultipartBody(deskripsi),

convertStringToMultipartBody(tataCaraPengolahan)
,
    fileImage,

convertStringToMultipartBody(linkYoutube),
)
}

private fun
postUpdateValueGaleriHerbalNoHaveImage(
    idValGaleriHerbal: String, idHalGaleriHerbal:
String, nama: String,
    deskripsi: String, tataCaraPengolahan: String,
linkYoutube: String,
) {

viewModel.postUpdateValueGaleriHerbalNoHaveI
mage(

"post_update_val_galeri_herbal_no_have_image",
idValGaleriHerbal, idHalGaleriHerbal,
    nama, deskripsi, tataCaraPengolahan,
linkYoutube
)
}

private fun getUpdateValueGaleriHerbal(){

viewModel.getResponseUpdateValueGaleriHerbal()
.observe(this@AdminValueGaleriHerbalActivity){re
sult->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@AdminValueGaler
iHerbalActivity)
        is UIState.Failure->
setFailureUpdateValueGaleriHerbal(result.message)
        is UIState.Success->
setSuccessUpdateValueGaleriHerbal(result.data)
    }
}

private fun
setSuccessUpdateValueGaleriHerbal(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

Toast.makeText(this@AdminValueGaleriHerbalAct
ivity, "Berhasil Update",
Toast.LENGTH_SHORT).show()
        tempAlertDialog!!.dismiss()
        tempAlertDialog = null
        tempView = null
        fetchData()
    } else{

```

```

Toast.makeText(this@AdminValueGaleriHerbalAct
ivity, data[0].message_response,
Toast.LENGTH_SHORT).show()
        }
    } else{

Toast.makeText(this@AdminValueGaleriHerbalAct
ivity, "Gagal di web",
Toast.LENGTH_SHORT).show()
    }
}

private fun
setFailureUpdateValueGaleriHerbal(message:
String) {

Toast.makeText(this@AdminValueGaleriHerbalAct
ivity, message, Toast.LENGTH_SHORT).show()
    loading.alertDialogCancel()
}

private fun setShowDialogHapus(pesanan:
GaleriHerbalListModel) {
    val view =
AlertDialogKonfirmasiBinding.inflate(layoutInflater
)

    val alertDialog =
AlertDialog.Builder(this@AdminValueGaleriHerbal
Activity)
        alertDialog.setView(view.root)
        .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

    view.apply {
        tvTitleKonfirmasi.text = "Hapus
${pesanan.nama!!} ?"
        tvBodyKonfirmasi.text = "Data ini akan
terhapus dan data tidak dapat dipulihkan"

        btnKonfirmasi.setOnClickListener {
            val idValueGaleriHerbal =
pesanan.id_val_galeri_herbal!!

postHapusValueGaleriHerbal(idValueGaleriHerbal)
            dialogInputan.dismiss()
        }
        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

private fun
postHapusValueGaleriHerbal(idValueGaleriHerbal:
String) {

viewModel.postHapusValueGaleriHerbal(idValueG
aleriHerbal)
}

private fun getHapusValueGaleriHerbal(){

viewModel.getResponseHapusValueGaleriHerbal().

```

```

observe(this@AdminValueGaleriHerbalActivity){result->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@AdminValueGaleriHerbalActivity)
        is UIState.Failure->
setFailureHapusValueGaleriHerbal(result.message)
        is UIState.Success->
setSuccessHapusValueGaleriHerbal(result.data)
    }
}

private fun
setSuccessHapusValueGaleriHerbal(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

Toast.makeText(this@AdminValueGaleriHerbalActivity, "Berhasil Hapus",
Toast.LENGTH_SHORT).show()
            fetchData()
        } else{

Toast.makeText(this@AdminValueGaleriHerbalActivity, data[0].message_response,
Toast.LENGTH_SHORT).show()
            }
        } else{

Toast.makeText(this@AdminValueGaleriHerbalActivity, "Gagal di web",
Toast.LENGTH_SHORT).show()
            }
        }

private fun
setFailureHapusValueGaleriHerbal(message: String)
{

Toast.makeText(this@AdminValueGaleriHerbalActivity, message, Toast.LENGTH_SHORT).show()
        loading.alertDialogCancel()
    }

private fun setStopShimmer(){
    binding.apply {
        rvValGaleriHerbal.visibility = View.VISIBLE

        smValGaleriHerbal.stopShimmer()
        smValGaleriHerbal.visibility = View.GONE
    }
}
private fun setStartShimmer(){
    binding.apply {
        smValGaleriHerbal.startShimmer()
        smValGaleriHerbal.visibility =
View.VISIBLE

        rvValGaleriHerbal.visibility = View.GONE
    }
}

```

```

// permission add image
private fun requestPermission(){
    if(Build.VERSION.SDK_INT >=
Build.VERSION_CODES.R){
        if
(Environment.isExternalStorageManager()) {
//            startActivity(Intent(this,
AdminHalamanGaleriHerbalAdapter::class.java))

startActivity(Intent(this@AdminValueGaleriHerbal
Activity,
AdminHalamanGaleriHerbalActivity::class.java))
        } else { //request for the permission
            val intent =
Intent(Settings.ACTION_MANAGE_APP_ALL_FILES_ACCESS_PERMISSION)
            val uri = Uri.fromParts("package",
packageName, null)
            intent.data = uri
            startActivity(intent)
        }
    } else{
        ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.WRITE_EXTERNAL_
STORAGE,
Manifest.permission.READ_EXTERNAL_STORAGE
),
Constant.STORAGE_PERMISSION_CODE
        )
    }
}

private fun pickImageFile() {
    val intent =
Intent(Intent.ACTION_OPEN_DOCUMENT).apply
{
        type = "image/*"
    }

    startActivityForResult(intent,
Constant.IMAGE_CODE)
}

override fun onActivityResult(requestCode: Int,
resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode,
resultCode, data)

    if (requestCode == Constant.IMAGE_CODE
&& resultCode == Activity.RESULT_OK && data
!= null) {
        // Mendapatkan URI file PDF yang dipilih
        val fileUri = data.data!!

        val nameImage = getNameFile(fileUri)

        tempView?.let {
            it.etEditGambar.text = nameImage
        }

        // Mengirim file PDF ke website

```

```

menggunakan Retrofit
        fileImage = uploadImageToStorage(fileUri,
nameImage, "galeri_herbal_image")
    }
}

private fun getNameFile(uri: Uri): String {
    val cursor = contentResolver.query(uri, null,
null, null, null)
    val nameIndex =
cursor?.getColumnIndex(OpenableColumns.DISPL
AY_NAME)
    cursor?.moveToFirst()
    val name = cursor?.getString(nameIndex!!)
    cursor?.close()
    return name!!
}

@SuppressLint("Recycle")
private fun uploadImageToStorage(imageUri:
Uri?, fileName: String, nameAPI:String):
MultipartBody.Part? {
    var pdfPart : MultipartBody.Part? = null
    imageUri?.let {
        val file =
contentResolver.openInputStream(imageUri)?.readB
ytes()

        if (file != null) {
            //      // Membuat objek RequestBody dari file
PDF
            //      val requestFile =
file.toRequestBody("application/pdf".toMediaType
OrNull())
            //      // Membuat objek MultipartBody.Part
untuk file PDF
            //      pdfPart =
MultipartBody.Part.createFormData("materi_pdf",
fileName, requestFile)

            pdfPart =
convertFileToMultipartBody(file, fileName,
nameAPI)
        }
    }
    return pdfPart
}

private fun convertFileToMultipartBody(file:
ByteArray, fileName: String, nameAPI:String):
MultipartBody.Part?{
    //      val requestFile =
file.toRequestBody("application/pdf".toMediaType
OrNull())
    val requestFile =
file.toRequestBody("application/image".toMediaTyp
eOrNull())
    val filePart =
MultipartBody.Part.createFormData(nameAPI,
fileName, requestFile)

    return filePart
}

private fun convertStringToMultipartBody(data:

```

```

String): RequestBody {
    return RequestBody.create("multipart/form-
data".toMediaTypeOrNull(), data)
}

private fun checkPermission(): Boolean{
    return if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.R){
        //Android is 11(R) or above
        Environment.isExternalStorageManager()
    }
    else{
        //Android is below 11(R)
        val write =
ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAG
E)
        val read =
ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE
)
        write ==
PackageManager.PERMISSION_GRANTED &&
read == PackageManager.PERMISSION_GRANTED
    }
}

```

Admin valuegaleriherbalviewmodel

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.galeri_herbal.value_galeri_herbal

import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import
com.example.edukasipengobatanpadapenderitastrok
e.data.database.api.ApiService
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalListModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalMainModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.delay
import kotlinx.coroutines.launch
import okhttp3.MultipartBody
import okhttp3.RequestBody
import javax.inject.Inject

@HiltViewModel
class AdminValueGaleriHerbalViewModel @Inject
constructor(
    private var api: ApiService
): ViewModel() {
    private val _valueGaleriHerbal =

```

```

MutableLiveData<UIState<ArrayList<GaleriHerbal
ListModel>>>()
    private val _postTambahValueGaleriHerbal =
MutableLiveData<UIState<ArrayList<ResponseMo
del>>>()
    private val _postUpdateValueGaleriHerbal =
MutableLiveData<UIState<ArrayList<ResponseMo
del>>>()
    private val _postHapusValueGaleriHerbal =
MutableLiveData<UIState<ArrayList<ResponseMo
del>>>()

    fun
fetchGaleriHerbal(id_hal_galeri_herbal:String) {
    viewModelScope.launch {

_valueGaleriHerbal.postValue(UIState.Loading)
    delay(1_000)
    try {
        val data = api.getGaleriHerbalList("",
id_hal_galeri_herbal)

_valueGaleriHerbal.postValue(UIState.Success(data
))
    } catch (ex: Exception) {

_valueGaleriHerbal.postValue(UIState.Failure("Erro
r ${ex.message}"))
    }
    }

    fun postTambahValueGaleriHerbal(
        post: RequestBody, idHalGaleriHerbal:
RequestBody, kataAcak: RequestBody, nama:
RequestBody,
        deskripsi: RequestBody, tataCaraPengolahan:
RequestBody, gambar: MultipartBody.Part, youtube:
RequestBody
    ){
        viewModelScope.launch(Dispatchers.IO) {

_postTambahValueGaleriHerbal.postValue(UIState.
Loading)
        delay(1_000)
        try {
            val postTambahValueGaleriHerbal =
api.addValueGaleriHerbal(
                post, idHalGaleriHerbal, kataAcak,
nama, deskripsi, tataCaraPengolahan, gambar,
youtube
            )

_postTambahValueGaleriHerbal.postValue(UIState.
Success(postTambahValueGaleriHerbal))
        } catch (ex: Exception){

_postTambahValueGaleriHerbal.postValue(UIState.
Failure("Error: ${ex.message}"))
        }
    }

    fun postUpdateValueGaleriHerbal(
        post: RequestBody, idValGaleriHerbal:

```

```

RequestBody, idHalGaleriHerbal: RequestBody,
kataAcak: RequestBody, nama: RequestBody,
        deskripsi: RequestBody, tataCaraPengolahan:
RequestBody, gambar: MultipartBody.Part, youtube:
RequestBody
    ){
        viewModelScope.launch(Dispatchers.IO) {

_postUpdateValueGaleriHerbal.postValue(UIState.L
oading)
        delay(1_000)
        try {
            val postTambahValueGaleriHerbal =
api.postUpdateValueGaleriHerbal(
                post, idValGaleriHerbal,
idHalGaleriHerbal, kataAcak, nama, deskripsi,
tataCaraPengolahan, gambar, youtube
            )

_postUpdateValueGaleriHerbal.postValue(UIState.S
uccess(postTambahValueGaleriHerbal))
        } catch (ex: Exception){

_postUpdateValueGaleriHerbal.postValue(UIState.F
ailure("Error: ${ex.message}"))
        }
    }

    fun postUpdateValueGaleriHerbalNoHaveImage(
        post: String, idValGaleriHerbal: String,
idHalGaleriHerbal: String, nama: String,
        deskripsi: String, tataCaraPengolahan: String,
youtube: String
    ){
        viewModelScope.launch(Dispatchers.IO) {

_postUpdateValueGaleriHerbal.postValue(UIState.L
oading)
        delay(1_000)
        try {
            val postTambahValueGaleriHerbal =
api.postUpdateValueGaleriHerbalNoHaveImage(
                "", idValGaleriHerbal,
idHalGaleriHerbal, nama, deskripsi,
tataCaraPengolahan, youtube
            )

_postUpdateValueGaleriHerbal.postValue(UIState.S
uccess(postTambahValueGaleriHerbal))
        } catch (ex: Exception){

_postUpdateValueGaleriHerbal.postValue(UIState.F
ailure("Error: ${ex.message}"))
        }
    }

    fun
postHapusValueGaleriHerbal(idValGaleriHerbal:
String){
        viewModelScope.launch(Dispatchers.IO) {

_postHapusValueGaleriHerbal.postValue(UIState.L
oading)

```

```

        delay(1_000)
        try {
            val postTambahValueGaleriHerbal =
                api.postHapusValueGaleriHerbal("",
                    idValGaleriHerbal)

            _postHapusValueGaleriHerbal.postValue(UIState.Success(
                postTambahValueGaleriHerbal))
        } catch (ex: Exception) {}

        _postHapusValueGaleriHerbal.postValue(UIState.Failure(
            "Error: ${ex.message}"))
    }
}

fun getGaleriHerbal():
    LiveData<UIState<ArrayList<GaleriHerbalListModel>>> = _valueGaleriHerbal
    fun getResponseTambahValueGaleriHerbal():
        LiveData<UIState<ArrayList<ResponseModel>>>
        = _postTambahValueGaleriHerbal
    fun getResponseUpdateValueGaleriHerbal():
        LiveData<UIState<ArrayList<ResponseModel>>>
        = _postUpdateValueGaleriHerbal
    fun getResponseHapusValueGaleriHerbal():
        LiveData<UIState<ArrayList<ResponseModel>>>
        = _postHapusValueGaleriHerbal
    }
}

```

Adminmainactivity

```

package
com.example.edukasipengobatanpadapenderitaastroke.ui.activity.admin.main

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.widget.Toast
import
com.example.edukasipengobatanpadapenderitaastroke.R
import
com.example.edukasipengobatanpadapenderitaastroke.databinding.ActivityAdminMainBinding
import
com.example.edukasipengobatanpadapenderitaastroke.ui.activity.admin.akun.AdminAkunActivity
import
com.example.edukasipengobatanpadapenderitaastroke.ui.activity.admin.galeri_herbal.halaman_galeri_herbal.AdminHalamanGaleriHerbalActivity
import
com.example.edukasipengobatanpadapenderitaastroke.ui.activity.admin.menu_sehat.AdminMenuSehatActivity
import
com.example.edukasipengobatanpadapenderitaastroke.ui.activity.admin.tentang_stroke.halaman_tentang_stroke.AdminHalamanTentangStrokeActivity
import
com.example.edukasipengobatanpadapenderitaastroke.ui.activity.admin.terapi.AdminTerapiActivity

```

```

import
com.example.edukasipengobatanpadapenderitaastroke.ui.activity.admin.testimoni.AdminTestimoniActivity
import
com.example.edukasipengobatanpadapenderitaastroke.utils.KontrolNavigationDrawer
import dagger.hilt.android.AndroidEntryPoint

```

```

@AndroidEntryPoint
class AdminMainActivity : AppCompatActivity() {
    private lateinit var binding:
        ActivityAdminMainBinding
    private lateinit var kontrolNavigationDrawer:
        KontrolNavigationDrawer
    override fun onCreate(savedInstanceState:
        Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
            ActivityAdminMainBinding.inflate(layoutInflater)
            setContentView(binding.root)

        setKontrolNavigationDrawer()
        setButton()
    }

    private fun setKontrolNavigationDrawer() {
        kontrolNavigationDrawer =
            KontrolNavigationDrawer(this@AdminMainActivity)
        binding.apply {

```

```

            kontrolNavigationDrawer.cekSebagai(navView)

```

```

            kontrolNavigationDrawer.onClickItemNavigationDrawer(navView, drawerLayoutMain, ivDrawerView,
                this@AdminMainActivity)
        }
    }

```

```

    private fun setButton() {
        binding.apply {
            btnTentangStroke.setOnClickListener {
                val i = Intent(this@AdminMainActivity,
                    AdminHalamanTentangStrokeActivity::class.java)
                startActivity(i)
                finish()
            }
            btnGaleriHerbal.setOnClickListener {
                val i = Intent(this@AdminMainActivity,
                    AdminHalamanGaleriHerbalActivity::class.java)
                startActivity(i)
                finish()
            }
            btnTerapi.setOnClickListener {
                val i = Intent(this@AdminMainActivity,
                    AdminTerapiActivity::class.java)
                startActivity(i)
                finish()
            }
            btnMenuSehat.setOnClickListener {
                val i = Intent(this@AdminMainActivity,
                    AdminMenuSehatActivity::class.java)
                startActivity(i)
                finish()
            }

```

```

    }
    btnTestimoni.setOnClickListener {
        val i = Intent(this@AdminMainActivity,
AdminTestimoniActivity::class.java)
        startActivity(i)
        finish()
    }
//    btnAkun.setOnClickListener {
//        val i = Intent(this@AdminMainActivity,
AdminAkunActivity::class.java)
//        startActivity(i)
//        finish()
//    }
}

private var tapDuaKali = false
override fun onBackPressed() {
    if (tapDuaKali){
        super.onBackPressed()
    }
    tapDuaKali = true
    Toast.makeText(this@AdminMainActivity,
    "Tekan Sekali Lagi untuk keluar",
    Toast.LENGTH_SHORT).show()

    Handler().postDelayed({
        tapDuaKali = false
    }, 2000)
}
}

```

Adminmenusehatactivity

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.menu_sehat

import android.Manifest
import android.annotation.SuppressLint
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Environment
import android.provider.OpenableColumns
import android.provider.Settings
import android.view.MenuItem
import android.view.View
import android.widget.PopupMenu
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.recyclerview.widget.LinearLayoutManager
import com.bumptech.glide.Glide
import com.example.edukasipengobatanpadapenderitastrok
e.R

```

```

import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin.AdminMenuSehatAdapter
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.MenuSehatModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityAdminMenuSehatBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKeteranganBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKonfirmasiBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogShowImageBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogMenuSehatBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.main.AdminMainActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KataAcak
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KontrolNavigationDrawer
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.AndroidEntryPoint
import
okhttp3.MediaType.Companion.toMediaTypeOrNull
import
okhttp3.MultipartBody
import okhttp3.RequestBody
import
okhttp3.RequestBody.Companion.toRequestBody
import javax.inject.Inject

```

```

@AndroidEntryPoint
class AdminMenuSehatActivity :
AppCompatActivity() {
    private lateinit var binding:
ActivityAdminMenuSehatBinding
    private lateinit var kontrolNavigationDrawer:
KontrolNavigationDrawer
    private val viewModel:
AdminMenuSehatViewModel by viewModels()
    private lateinit var adapter:

```

```

AdminMenuSehatAdapter
private var tempAlertDialog: AlertDialog? = null
@Inject
lateinit var loading: LoadingAlertDialog
private var tempView:
AlertDialogMenuSehatBinding? = null
override fun onCreate(savedInstanceState:
Bundle?) {
    super.onCreate(savedInstanceState)
    binding =
ActivityAdminMenuSehatBinding.inflate(layoutInfl
ater)
    setContentView(binding.root)

    setKontrolNavigationDrawer()
    setButton()
    fetchMenuSehat()
    getMenuSehat()
    getTambahMenuSehat()
    getUpdateMenuSehat()
    getHapusMenuSehat()
}

private fun setKontrolNavigationDrawer() {
    kontrolNavigationDrawer =
KontrolNavigationDrawer(this@AdminMenuSehat
Activity)
    binding.apply {

kontrolNavigationDrawer.cekSebagai(navView)

kontrolNavigationDrawer.onClickItemNavigationDr
awer(navView, drawerLayoutMain, ivDrawerView,
this@AdminMenuSehatActivity)
    }

    private fun setButton() {
        binding.btnTambah.setOnClickListener {
            dialogTambahMenuSehat()
        }
    }

    private fun dialogTambahMenuSehat() {
        val view =
AlertDialogMenuSehatBinding.inflate(layoutInflater
)

        val alertDialog =
AlertDialog.Builder(this@AdminMenuSehatActivit
y)
        alertDialog.setView(view.root)
        .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        tempAlertDialog = dialogInputan
        tempView = view

        view.apply {
            btnSimpan.setOnClickListener {
                var check = true

                if(etEditJudul.text.isEmpty()){
                    etEditJudul.error = "Tidak Boleh

```

```

Kosong"
                check = false
            }
            if(etEditDeskripsi.text.isEmpty()){
                etEditDeskripsi.error = "Tidak Boleh
Kosong"
                check = false
            }
            if(check){
                val judul = etEditJudul.text.toString()
                val deskripsi =
etEditDeskripsi.text.toString()

                postTambahMenuSehat(judul,
deskripsi)
            }
        }
        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
            tempAlertDialog = null
        }
    }

    private fun postTambahMenuSehat(
        judul: String, deskripsi: String
    ) {
        viewModel.postTambahMenuSehat(
            judul, deskripsi
        )
    }

    private fun getTambahMenuSehat(){
        viewModel.getResponseTambahMenuSehat().obser
ve(this@AdminMenuSehatActivity){result->
            when(result){
                is UIState.Loading->
                    loading.alertDialogLoading(this@AdminMenuSehat
Activity)
                is UIState.Failure->
                    setFailureTambahMenuSehat(result.message)
                is UIState.Success->
                    setSuccessHalamTambahanMenuSehat(result.data)
            }
        }
    }

    private fun
setFailureTambahMenuSehat(message: String) {
        loading.alertDialogCancel()

        Toast.makeText(this@AdminMenuSehatActivity,
message, Toast.LENGTH_SHORT).show()
    }

    private fun
setSuccessHalamTambahanMenuSehat(data:
ArrayList<ResponseModel>) {
        loading.alertDialogCancel()
        if(data.isNotEmpty()){
            if(data[0].status=="0"){

                Toast.makeText(this@AdminMenuSehatActivity,
"Berhasil Tambah",

```

```

Toast.LENGTH_SHORT).show()
        tempAlertDialog!!.dismiss()
        tempAlertDialog = null
        tempView = null
        fetchMenuSehat()
    } else{

Toast.makeText(this@AdminMenuSehatActivity,
data[0].message_response,
Toast.LENGTH_SHORT).show()
    }
    } else{

Toast.makeText(this@AdminMenuSehatActivity,
"ada error di API", Toast.LENGTH_SHORT).show()
    }
    }

    private fun fetchMenuSehat() {
        viewModel.fetchMenuSehat()
    }

    private fun getMenuSehat(){

viewModel.getMenuSehat().observe(this@AdminM
enuSehatActivity){result->
    when(result){
        is UIState.Loading-> setStartShimmer()
        is UIState.Failure->
setFailureMenuSehat(result.message)
        is UIState.Success->
setSuccessMenuSehat(result.data)
    }
}

    private fun setFailureMenuSehat(message: String)
{
    setStopShimmer()

Toast.makeText(this@AdminMenuSehatActivity,
message, Toast.LENGTH_SHORT).show()
    }

    private fun setSuccessMenuSehat(data:
ArrayList<MenuSehatModel>) {
        setStopShimmer()
        if(data.isNotEmpty()){
            setAdapter(data)
        } else{

Toast.makeText(this@AdminMenuSehatActivity,
"Data Kosong", Toast.LENGTH_SHORT).show()
    }
    }

    private fun setAdapter(data:
ArrayList<MenuSehatModel>) {
        adapter = AdminMenuSehatAdapter(data,
object : OnClickItem.ClickAdminMenuSehat{
            override fun
clickItemJudul(namaMenuSehat: String, it: View) {
                showClickText("Nama MenuSehatp",
namaMenuSehat)
            }
        }
    }

```

```

        override fun clickItemDeskripsi(deskripsi:
String, it: View) {
            showClickText("Deskripsi MenuSehat",
deskripsi)
        }

        override fun clickItemSetting(menuSehat:
MenuSehatModel, it: View) {
            val popupMenu =
PopupMenu(this@AdminMenuSehatActivity, it)

popupMenu.inflate(R.menu.popup_edit_hapus)

popupMenu.setOnMenuItemClickListener(object :
PopupMenu.OnMenuItemClickListener
{
            override fun
onMenuItemClick(menuItem: MenuItem?): Boolean
{
                when (menuItem!!.itemId) {
                    R.id.edit -> {

dialogEditMenuSehat(menuSehat)
                        return true
                    }
                    R.id.hapus -> {

dialogHapusMenuSehat(menuSehat.id_menu_sehat!
!, menuSehat.judul!!)
                        return true
                    }
                }
                return true
            }
        })
        popupMenu.show()
    }

    binding.apply {
        rvMenuSehat.layoutManager =
LinearLayoutManager(this@AdminMenuSehatActiv
ity, LinearLayoutManager.VERTICAL, false)
        rvMenuSehat.adapter = adapter
    }

    private fun showClickText(keterangan:String,
judul: String) {
        val view =
AlertDialogKeteranganBinding.inflate(layoutInflater
)

        val alertDialog =
AlertDialog.Builder(this@AdminMenuSehatActivit
y)
        alertDialog.setView(view.root)
            .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()
    }

```

```

        view.apply {
            tvTitleKeterangan.text = keterangan
            tvBodyKeterangan.text = judul
            btnClose.setOnClickListener {
                dialogInputan.dismiss()
            }
        }
    }

    private fun setShowImage(gambar: String,
        title:String) {
        val view =
        AlertDialogShowImageBinding.inflate(layoutInflater)

        val alertDialog =
        AlertDialog.Builder(this@AdminMenuSehatActivity)

        alertDialog.setView(view.root)
            .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        view.apply {
            tvTitle.text = "Gambar $title"
            btnClose.setOnClickListener {
                dialogInputan.dismiss()
            }
        }

        Glide.with(this@AdminMenuSehatActivity)

        .load("${Constant.BASE_URL}${Constant.LOCATION_GAMBAR}$gambar") // URL Gambar
            .error(R.drawable.gambar_error_image)
            .into(view.ivShowImage) // imageView
        mana yang akan diterapkan

    }

    private fun dialogEditMenuSehat(menuSehat:
        MenuSehatModel) {
        val view =
        AlertDialogMenuSehatBinding.inflate(layoutInflater)

        val alertDialog =
        AlertDialog.Builder(this@AdminMenuSehatActivity)

        alertDialog.setView(view.root)
            .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        view.tvTitle.text = "Update Halaman Menu
        Sehat"

        tempAlertDialog = dialogInputan
        tempView = view

        view.apply {
            etEditJudul.setText(menuSehat.judul)

            etEditDeskripsi.setText(menuSehat.deskripsi)

```

```

        btnSimpan.setOnClickListener {
            var check = true

            if(etEditJudul.text.isEmpty()){
                etEditJudul.error = "Tidak Boleh
                Kosong"
                check = false
            }
            if(etEditDeskripsi.text.isEmpty()){
                etEditDeskripsi.error = "Tidak Boleh
                Kosong"
                check = false
            }
            if(check){
                postEditMenuSehat(
                    menuSehat.id_menu_sehat!!,
                    etEditJudul.text.toString(),
                    etEditDeskripsi.text.toString(),
                )
            }
        }

        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
            tempAlertDialog = null
        }

        private fun postEditMenuSehat(
            idHalMenuSehat: String, judul: String,
            deskripsi: String
        ) {
            viewModel.postUpdateMenuSehat(
                idHalMenuSehat, judul, deskripsi
            )
        }

        private fun getUpdateMenuSehat(){

            viewModel.getResponseUpdateMenuSehat().observe(
                this@AdminMenuSehatActivity){result->
                    when(result){
                        is UIState.Loading->
                            loading.alertDialogLoading(this@AdminMenuSehat
                                Activity)
                        is UIState.Failure->
                            setFailureUpdateMenuSehat(result.message)
                        is UIState.Success->
                            setSuccessHalamanUpdateMenuSehat(result.data)
                    }
                }
            }

            private fun setFailureUpdateMenuSehat(message:
                String) {
                loading.alertDialogCancel()

                Toast.makeText(this@AdminMenuSehatActivity,
                    message, Toast.LENGTH_SHORT).show()
            }

            private fun
            setSuccessHalamanUpdateMenuSehat(data:
                ArrayList<ResponseModel>) {
                loading.alertDialogCancel()

```

```

        if(data.isNotEmpty()){
            if(data[0].status=="0"){

                Toast.makeText(this@AdminMenuSehatActivity,
                    "Berhasil Update", Toast.LENGTH_SHORT).show()
                tempAlertDialog!!.dismiss()
                tempAlertDialog = null
                fetchMenuSehat()
            } else{

                Toast.makeText(this@AdminMenuSehatActivity,
                    data[0].message_response,
                    Toast.LENGTH_SHORT).show()
            }
        } else{

            Toast.makeText(this@AdminMenuSehatActivity,
                "ada error di aplikasi",
                Toast.LENGTH_SHORT).show()
        }
    }

    private fun
    dialogHapusMenuSehat(idHalMenuSehat: String,
        namaMenuSehat: String) {
        val view =
        AlertDialogKonfirmasiBinding.inflate(layoutInflater
        )

        val alertDialog =
        AlertDialog.Builder(this@AdminMenuSehatActivity)
        alertDialog.setView(view.root)
            .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        tempAlertDialog = dialogInputan

        view.apply {
            tvTitleKonfirmasi.text = "Yakin Hapus
            Penjelasan ini?"
            tvBodyKonfirmasi.text = "Penjelasan ini
            akan dihapus dan data yang ada di dalamnya akan
            ikut terhapus. data yang telah terhapus tidak dapat di
            kembalikan"

            btnKonfirmasi.setOnClickListener {
                postHapusMenuSehat(idHalMenuSehat!!)
            }
            btnBatal.setOnClickListener {
                tempAlertDialog = null
                dialogInputan.dismiss()
            }
        }
    }

    private fun
    postHapusMenuSehat(idHalMenuSehat: String) {

        viewModel.postHapusMenuSehat(idHalMenuSehat)
    }

    private fun getHapusMenuSehat(){

```

```

        viewModel.getResponseHapusMenuSehat().observe
        (this@AdminMenuSehatActivity){result->
            when(result){
                is UIState.Loading->
                loading.alertDialogLoading(this@AdminMenuSehat
                Activity)
                is UIState.Failure->
                setFailureHapusMenuSehat(result.message)
                is UIState.Success->
                setSuccessHapusMenuSehat(result.data)
            }
        }

        private fun setFailureHapusMenuSehat(message:
        String) {
            loading.alertDialogCancel()

            Toast.makeText(this@AdminMenuSehatActivity,
                message, Toast.LENGTH_SHORT).show()
        }

        private fun setSuccessHapusMenuSehat(data:
        ArrayList<ResponseModel>) {
            loading.alertDialogCancel()
            if(data.isNotEmpty()){
                if(data[0].status=="0"){

                    Toast.makeText(this@AdminMenuSehatActivity,
                        "Berhasil Hapus", Toast.LENGTH_SHORT).show()
                    tempAlertDialog!!.dismiss()
                    tempAlertDialog = null
                    fetchMenuSehat()
                } else{

                    Toast.makeText(this@AdminMenuSehatActivity,
                        data[0].message_response,
                        Toast.LENGTH_SHORT).show()
                }
            } else{

                Toast.makeText(this@AdminMenuSehatActivity,
                    "ada error di aplikasi",
                    Toast.LENGTH_SHORT).show()
            }
        }

        private fun setStopShimmer(){
            binding.apply {
                rvMenuSehat.visibility = View.VISIBLE

                smMenuSehat.stopShimmer()
                smMenuSehat.visibility = View.GONE
            }
        }

        private fun setStartShimmer(){
            binding.apply {
                smMenuSehat.startShimmer()
                smMenuSehat.visibility = View.VISIBLE

                rvMenuSehat.visibility = View.GONE
            }
        }

        override fun onBackPressed() {

```

```

        super.onBackPressed()

        startActivity(Intent(this@AdminMenuSehatActivity,
            AdminMainActivity::class.java))
        finish()
    }
}

```

Admin menusehatviewmodel

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.menu_sehat

```

```

import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import
com.example.edukasipengobatanpadapenderitastrok
e.data.database.api.ApiService
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.MenuSehatModel
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.delay
import kotlinx.coroutines.launch
import okhttp3.MultipartBody
import okhttp3.RequestBody
import javax.inject.Inject

```

```

@HiltViewModel
class AdminMenuSehatViewModel @Inject
constructor(
    private val api: ApiService
): ViewModel() {
    private val _menuSehat =
MutableLiveData<UISState<ArrayList<MenuSehatModel>>>()
    private val _postTambahMenuSehat =
MutableLiveData<UISState<ArrayList<ResponseModel>>>()
    private val _postUpdateMenuSehat =
MutableLiveData<UISState<ArrayList<ResponseModel>>>()
    private val _postHapusMenuSehat =
MutableLiveData<UISState<ArrayList<ResponseModel>>>()

```

```

    fun fetchMenuSehat() {
        viewModelScope.launch {
            _menuSehat.postValue(UISState.Loading)
            delay(1_000)
            try {
                val data = api.getMenuSehat("")

```

```

            _menuSehat.postValue(UISState.Success(data))
        } catch (ex: Exception) {

```

```

        _menuSehat.postValue(UISState.Failure("Error
            ${ex.message}"))
        }
    }
}

```

```

    fun postTambahMenuSehat(
        judul: String, deskripsi: String
    ){
        viewModelScope.launch(Dispatchers.IO) {

```

```

            _postTambahMenuSehat.postValue(UISState.Loading)
            delay(1_000)
            try {
                val postTambahMenuSehat =
                api.addMenuSehat("", judul, deskripsi)

```

```

            _postTambahMenuSehat.postValue(UISState.Success(
                postTambahMenuSehat))
            } catch (ex: Exception){

```

```

            _postTambahMenuSehat.postValue(UISState.Failure(
                "Error: ${ex.message}"))
            }
        }
    }
}

```

```

    fun postUpdateMenuSehat(
        id_menu_sehat: String, judul: String, deskripsi:
String
    ){
        viewModelScope.launch(Dispatchers.IO) {

```

```

            _postUpdateMenuSehat.postValue(UISState.Loading)
            delay(1_000)
            try {
                val postTambahMenuSehat =
                api.postUpdateMenuSehat(
                    "", id_menu_sehat, judul, deskripsi
                )

```

```

            _postUpdateMenuSehat.postValue(UISState.Success(
                postTambahMenuSehat))
            } catch (ex: Exception){

```

```

            _postUpdateMenuSehat.postValue(UISState.Failure("
                Error: ${ex.message}"))
            }
        }
    }
}

```

```

    fun postHapusMenuSehat(id_menu_sehat:
String){
        viewModelScope.launch(Dispatchers.IO) {

```

```

            _postHapusMenuSehat.postValue(UISState.Loading)
            delay(1_000)
            try {
                val postTambahMenuSehat =
                api.postHapusMenuSehat("", id_menu_sehat)

```

```

            _postHapusMenuSehat.postValue(UISState.Success(p
                ostTambahMenuSehat))

```

```

        } catch (ex: Exception){

        }

        _postHapusMenuSehat.postValue(UIState.Failure("
        Error: ${ex.message}"))
    }
}

fun getMenuSehat():
LiveData<UIState<ArrayList<MenuSehatModel>>>
= _menuSehat
    fun getResponseTambahMenuSehat():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postTambahMenuSehat
    fun getResponseUpdateMenuSehat():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postUpdateMenuSehat
    fun getResponseHapusMenuSehat():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postHapusMenuSehat
}

```

admin halaman tentang stroke activity

```

package
com.example.edukasipengobatanpadapenderitaastrok
e.ui.activity.admin.tentang_stroke.halaman_tentang_
stroke

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.MenuItem
import android.view.View
import android.widget.PopupMenu
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import
androidx.recyclerview.widget.LinearLayoutManager
import
com.example.edukasipengobatanpadapenderitaastrok
e.R
import
com.example.edukasipengobatanpadapenderitaastrok
e.adapter.admin.AdminHalamanTentangStrokeAdap
ter
import
com.example.edukasipengobatanpadapenderitaastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitaastrok
e.data.model.TentangStrokeListModel
import
com.example.edukasipengobatanpadapenderitaastrok
e.databinding.ActivityAdminHalamanTentangStroke
Binding
import
com.example.edukasipengobatanpadapenderitaastrok
e.databinding.AlertDialogHalTentangStrokeBinding
import
com.example.edukasipengobatanpadapenderitaastrok
e.databinding.AlertDialogKonfirmasiBinding
import
com.example.edukasipengobatanpadapenderitaastrok

```

```

e.ui.activity.admin.main.AdminMainActivity
import
com.example.edukasipengobatanpadapenderitaastrok
e.ui.activity.admin.tentang_stroke.value_tentang_str
oke.AdminValueTentangStrokeActivity
import
com.example.edukasipengobatanpadapenderitaastrok
e.utils.KontrolNavigationDrawer
import
com.example.edukasipengobatanpadapenderitaastrok
e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitaastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitaastrok
e.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

```

```

@AndroidEntryPoint
class AdminHalamanTentangStrokeActivity :
AppCompatActivity() {
    private lateinit var binding:
ActivityAdminHalamanTentangStrokeBinding
    private lateinit var kontrolNavigationDrawer:
KontrolNavigationDrawer
    private val viewModel:
AdminHalamanTentangStrokeViewModel by
viewModels()
    private lateinit var adapter:
AdminHalamanTentangStrokeAdapter
    private var tempAlertDialog: AlertDialog? = null
    @Inject lateinit var loading: LoadingAlertDialog
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityAdminHalamanTentangStrokeBinding.inflate(
layoutInflater)
        setContentView(binding.root)

        setKontrolNavigationDrawer()
        setButton()
        fetchTentangStroke()
        getTentangStroke()
        getTambahHalamanTentangStroke()
        getUpdateHalamanTentangStroke()
        getHapusHalamanTentangStroke()
    }

    private fun setKontrolNavigationDrawer() {
        kontrolNavigationDrawer =
KontrolNavigationDrawer(this@AdminHalamanTen
tangStrokeActivity)
        binding.apply {

            kontrolNavigationDrawer.cekSebagai(navView)

            kontrolNavigationDrawer.onClickItemNavigationDr
awer(navView, drawerLayoutMain, ivDrawerView,
this@AdminHalamanTentangStrokeActivity)
        }
    }
}

```

```

private fun setButton() {
    binding.btnTambah.setOnClickListener {
        dialogTambahHalamanTentangStroke()
    }
}

private fun
dialogTambahHalamanTentangStroke() {
    val view =
AlertDialogHalTentangStrokeBinding.inflate(layout
Inflater)

    val alertDialog =
AlertDialog.Builder(this@AdminHalamanTentangSt
rokeActivity)
        alertDialog.setView(view.root)
        .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

    tempAlertDialog = dialogInputan

    view.apply {
        btnSimpan.setOnClickListener {
            var check = true

            if(etEditNamaHalaman.text.isEmpty()){
                etEditNamaHalaman.error = "Tidak
Boleh Kosong"
                check = false
            }
            if(check){

postTambahHalamanTentangStroke(etEditNamaHal
aman.text.toString())
            }
        }
        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
            tempAlertDialog = null
        }
    }
}

private fun
postTambahHalamanTentangStroke(namaHalaman:
String) {

viewModel.postTambahHalamanTentangStroke(na
maHalaman)
}

private fun getTambahHalamanTentangStroke(){

viewModel.getResponseTambahHalamanTentangStr
oke().observe(this@AdminHalamanTentangStrokeA
ctivity){result->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@AdminHalamanTe
ntangStrokeActivity)
        is UIState.Failure->
setFailureTambahHalamanTentangStroke(result.mes
sage)
        is UIState.Success->

```

```

setSuccessHalamTambahanTentangStroke(result.dat
a)
    }
}

private fun
setFailureTambahHalamanTentangStroke(message:
String) {

Toast.makeText(this@AdminHalamanTentangStrok
eActivity, message,
Toast.LENGTH_SHORT).show()
}

private fun
setSuccessHalamTambahanTentangStroke(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

Toast.makeText(this@AdminHalamanTentangStrok
eActivity, "Berhasil Tambah",
Toast.LENGTH_SHORT).show()
            tempAlertDialog!!.dismiss()
            tempAlertDialog = null
            fetchTentangStroke()
        } else{

Toast.makeText(this@AdminHalamanTentangStrok
eActivity, data[0].message_response,
Toast.LENGTH_SHORT).show()
        }
    } else{

Toast.makeText(this@AdminHalamanTentangStrok
eActivity, "ada error di aplikasi",
Toast.LENGTH_SHORT).show()
    }
}

private fun fetchTentangStroke() {
    viewModel.fetchTentangStrokeList()
}

private fun getTentangStroke(){

viewModel.getTentangStrokeList().observe(this@A
dminHalamanTentangStrokeActivity){result->
    when(result){
        is UIState.Loading-> setStartShimmer()
        is UIState.Failure->
setFailureTentangStroke(result.message)
        is UIState.Success->
setSuccessTentangStroke(result.data)
    }
}

private fun setFailureTentangStroke(message:
String) {
    setStopShimmer()

Toast.makeText(this@AdminHalamanTentangStrok

```

```

eActivity, message,
Toast.LENGTH_SHORT).show()
    }

    private fun setSuccessTentangStroke(data:
    ArrayList<TentangStrokeListModel>) {
        setStopShimmer()
        if(data.isNotEmpty()){
            setAdapter(data)
        } else{

Toast.makeText(this@AdminHalamanTentangStrok
eActivity, "Data Kosong",
Toast.LENGTH_SHORT).show()
        }
    }

    private fun setAdapter(data:
    ArrayList<TentangStrokeListModel>) {
        adapter =
        AdminHalamanTentangStrokeAdapter(data, object :
        OnClickItem.ClickAdminTentangStroke{
            override fun clickItem(
                hal_tentang_stroke:
                TentangStrokeListModel,
                id_hal_tentang_stroke: String,
                halaman: String,
                it: View
            ) {
                val popupMenu =
                PopupMenu(this@AdminHalamanTentangStrokeAc
                tivity, it)

                popupMenu.inflate(R.menu.popup_rincian_edit_hap
                us)

                popupMenu.setOnMenuItemClickListener(object :
                PopupMenu.OnMenuItemClickListener
                {
                    override fun
                    onMenuItemClick(menuItem: MenuItem?): Boolean
                    {
                        when (menuItem!!.itemId) {
                            R.id.rincian -> {
                                val i =
                                Intent(this@AdminHalamanTentangStrokeActivity,
                                AdminValueTentangStrokeActivity::class.java)
                                i.putExtra("hal_tentang_stroke",
                                data)

                                i.putExtra("id_hal_tentang_stroke",
                                id_hal_tentang_stroke)
                                i.putExtra("halaman", halaman)
                                startActivity(i)
                                return true
                            }
                            R.id.edit -> {

                                dialogEditHalamanTentangStroke(id_hal_tentang_st
                                roke, halaman)

                                return true
                            }
                            R.id.hapus -> {

                                dialogHapusHalamanTentangStroke(id_hal_tentang

```

```

                                _stroke, halaman)

                                return true
                            }
                        }
                    }
                })
                popupMenu.show()
            }
        })

        binding.apply {
            rvTentangStroke.layoutManager =
            LinearLayoutManager(this@AdminHalamanTentan
            gStrokeActivity, LinearLayoutManager.VERTICAL,
            false)
            rvTentangStroke.adapter = adapter
        }

        private fun
        dialogEditHalamanTentangStroke(id_hal_tentang_st
        roke: String, halaman: String) {
            val view =
            AlertDialogHalTentangStrokeBinding.inflate(layout
            Inflater)

            val alertDialog =
            AlertDialog.Builder(this@AdminHalamanTentangSt
            rokeActivity)
            alertDialog.setView(view.root)
            .setCancelable(false)
            val dialogInputan = alertDialog.create()
            dialogInputan.show()

            view.tvTitle.text = "Update Halaman Tentang
            Stroke"

            tempAlertDialog = dialogInputan

            view.apply {
                etEditNamaHalaman.setText(halaman)

                btnSimpan.setOnClickListener {
                    var check = true

                    if(etEditNamaHalaman.text.isEmpty()){
                        etEditNamaHalaman.error = "Tidak
                        Boleh Kosong"
                        check = false
                    }
                    if(check){

                        postEditHalamanTentangStroke(id_hal_tentang_stro
                        ke, etEditNamaHalaman.text.toString())
                    }
                }
                btnBatal.setOnClickListener {
                    dialogInputan.dismiss()
                    tempAlertDialog = null
                }
            }
        }
    }
}

```

```

        private fun
        postEditHalamanTentangStroke(idHalTentangStroke: String, halaman: String) {

        viewModel.postUpdateHalamanTentangStroke(idHalTentangStroke, halaman )
        }

        private fun getUpdateHalamanTentangStroke(){

        viewModel.getResponseUpdateHalamanTentangStroke().observe(this@AdminHalamanTentangStrokeActivity){result->
            when(result){
                is UIState.Loading->
                loading.alertDialogLoading(this@AdminHalamanTentangStrokeActivity)
                is UIState.Failure->
                setFailureUpdateHalamanTentangStroke(result.message)
                is UIState.Success->
                setSuccessHalamanUpdateTentangStroke(result.data)
            }
        }
        }

        private fun
        setFailureUpdateHalamanTentangStroke(message: String) {
            loading.alertDialogCancel()

            Toast.makeText(this@AdminHalamanTentangStrokeActivity, message,
            Toast.LENGTH_SHORT).show()
        }

        private fun
        setSuccessHalamanUpdateTentangStroke(data: ArrayList<ResponseModel>) {
            loading.alertDialogCancel()
            if(data.isNotEmpty()){
                if(data[0].status=="0"){

                Toast.makeText(this@AdminHalamanTentangStrokeActivity, "Berhasil Update",
                Toast.LENGTH_SHORT).show()
                tempAlertDialog!!.dismiss()
                tempAlertDialog = null
                fetchTentangStroke()
            } else{

            Toast.makeText(this@AdminHalamanTentangStrokeActivity, data[0].message_response,
            Toast.LENGTH_SHORT).show()
            }
        } else{

        Toast.makeText(this@AdminHalamanTentangStrokeActivity, "ada error di aplikasi",
        Toast.LENGTH_SHORT).show()
        }
        }
    }

```

```

        private fun
        dialogHapusHalamanTentangStroke(idHalTentangStroke: String, halaman: String) {
            val view =
            AlertDialogKonfirmasiBinding.inflate(layoutInflater)
        }

        val alertDialog =
        AlertDialog.Builder(this@AdminHalamanTentangStrokeActivity)
        alertDialog.setView(view.root)
        alertDialog.setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        tempAlertDialog = dialogInputan

        view.apply {
            tvTitleKonfirmasi.text = "Yakin Hapus halaman tentang Stroke ini ini?"
            tvBodyKonfirmasi.text = "Halaman $halaman akan dihapus dan data yang ada di dalamnya akan ikut terhapus. data yang telah terhapus tidak dapat di kembalikan"

            btnKonfirmasi.setOnClickListener {

            postHapusHalamanTentangStroke(idHalTentangStroke!!)
            }
            btnBatal.setOnClickListener {
                tempAlertDialog = null
                dialogInputan.dismiss()
            }
        }
    }

    private fun
    postHapusHalamanTentangStroke(idHalTentangStroke: String) {

    viewModel.postHapusHalamanTentangStroke(idHalTentangStroke)
    }

    private fun getHapusHalamanTentangStroke(){

    viewModel.getResponseHapusHalamanTentangStroke().observe(this@AdminHalamanTentangStrokeActivity){result->
        when(result){
            is UIState.Loading->
            loading.alertDialogLoading(this@AdminHalamanTentangStrokeActivity)
            is UIState.Failure->
            setFailureHapusHalamanTentangStroke(result.message)
            is UIState.Success->
            setSuccessHapusHalamanTentangStroke(result.data)
        }
    }
    }

    private fun
    setFailureHapusHalamanTentangStroke(message:

```

```
String) {
    loading.alertDialogCancel()

    Toast.makeText(this@AdminHalamanTentangStrokeActivity, message,
        Toast.LENGTH_SHORT).show()
    }

    private fun
    setSuccessHapusHalamanTentangStroke(data:
    ArrayList<ResponseModel>) {
        loading.alertDialogCancel()
        if(data.isNotEmpty()){
            if(data[0].status=="0"){

                Toast.makeText(this@AdminHalamanTentangStrokeActivity, "Berhasil Hapus",
                    Toast.LENGTH_SHORT).show()
                    tempAlertDialog!!.dismiss()
                    tempAlertDialog = null
                    fetchTentangStroke()
                } else{

                Toast.makeText(this@AdminHalamanTentangStrokeActivity, data[0].message_response,
                    Toast.LENGTH_SHORT).show()
                }
            } else{

            Toast.makeText(this@AdminHalamanTentangStrokeActivity, "ada error di aplikasi",
                Toast.LENGTH_SHORT).show()
            }
        }

        private fun setStopShimmer(){
            binding.apply {
                rvTentangStroke.visibility = View.VISIBLE

                smTentangStroke.stopShimmer()
                smTentangStroke.visibility = View.GONE
            }
        }
        private fun setStartShimmer(){
            binding.apply {
                smTentangStroke.startShimmer()
                smTentangStroke.visibility = View.VISIBLE

                rvTentangStroke.visibility = View.GONE
            }
        }

        override fun onBackPressed() {
            super.onBackPressed()
        }

        startActivity(Intent(this@AdminHalamanTentangStrokeActivity, AdminMainActivity::class.java))
        finish()
    }
}
```

admin halaman tentangstrokeviewmodel

```
package
com.example.edukasipengobatanpadapenderitastrok
```

```
e.ui.activity.admin.tentang_stroke.halaman_tentang_stroke
```

```
import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import
com.example.edukasipengobatanpadapenderitastrok
e.data.database.api.ApiService
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TentangStrokeListModel
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.delay
import kotlinx.coroutines.launch
import javax.inject.Inject
```

```
@HiltViewModel
class AdminHalamanTentangStrokeViewModel
@Inject constructor(
    private val api: ApiService
): ViewModel() {
    private val _tentangStrokeList =
    MutableLiveData<UISState<ArrayList<TentangStrokeListModel>>>()
    private val _postTambahHalamanTentangStroke =
    MutableLiveData<UISState<ArrayList<ResponseModel>>>()
    private val _postUpdateHalamanTentangStroke =
    MutableLiveData<UISState<ArrayList<ResponseModel>>>()
    private val _postHapusHalamanTentangStroke =
    MutableLiveData<UISState<ArrayList<ResponseModel>>>()
}
```

```
fun fetchTentangStrokeList() {
    viewModelScope.launch {

        _tentangStrokeList.postValue(UISState.Loading)
        delay(1_000)
        try {
            val data = api.getTentangStrokeList("")

            _tentangStrokeList.postValue(UISState.Success(data)
        )
        } catch (ex: Exception) {

        }

        _tentangStrokeList.postValue(UISState.Failure("Error
        ${ex.message}"))
    }
}

fun postTambahHalamanTentangStroke(
    halaman:String
){
    viewModelScope.launch(Dispatchers.IO) {
```

```

        _postTambahHalamanTentangStroke.postValue(UIS
tate.Loading)
            delay(1_000)
            try {
                val postTambahHalamanTentangStroke =
api.addHalamanTentangStroke("", halaman)

                _postTambahHalamanTentangStroke.postValue(UIS
tate.Success(postTambahHalamanTentangStroke))
            } catch (ex: Exception){

                _postTambahHalamanTentangStroke.postValue(UIS
tate.Failure("Error: ${ex.message}"))
            }
        }

        fun postUpdateHalamanTentangStroke(
            idHalamanTentangStroke:String,
            halaman:String
        ){
            viewModelScope.launch(Dispatchers.IO) {

                _postUpdateHalamanTentangStroke.postValue(UIS
tate.Loading)
                    delay(1_000)
                    try {
                        val postTambahHalamanTentangStroke =
api.postUpdateHalamanTentangStroke("",
idHalamanTentangStroke, halaman)

                        _postUpdateHalamanTentangStroke.postValue(UIS
tate.Success(postTambahHalamanTentangStroke))
                    } catch (ex: Exception){

                        _postUpdateHalamanTentangStroke.postValue(UIS
tate.Failure("Error: ${ex.message}"))
                    }
                }

                fun
postHapusHalamanTentangStroke(id_hal_tentang_st
roke: String){
                    viewModelScope.launch(Dispatchers.IO) {

                        _postHapusHalamanTentangStroke.postValue(UIS
tate.Loading)
                            delay(1_000)
                            try {
                                val postTambahHalamanTentangStroke =
api.postHapusHalamanTentangStroke("",
id_hal_tentang_stroke)

                                _postHapusHalamanTentangStroke.postValue(UIS
tate.Success(postTambahHalamanTentangStroke))
                            } catch (ex: Exception){

                                _postHapusHalamanTentangStroke.postValue(UIS
tate.Failure("Error: ${ex.message}"))
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        fun getTentangStrokeList():
LiveData<UIState<ArrayList<TentangStrokeListMo
del>>> = _tentangStrokeList
        fun
getResponseTambahHalamanTentangStroke():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postTambahHalamanTentangStroke
        fun getResponseUpdateHalamanTentangStroke():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postUpdateHalamanTentangStroke
        fun getResponseHapusHalamanTentangStroke():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postHapusHalamanTentangStroke
    }
}

```

AdminValueTentangStrokeActivity

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.tentang_stroke.value_tentang_st
roke

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.MenuItem
import android.view.View
import android.widget.AdapterView
import android.widget.AdapterView.Adapter
import android.widget.PopupMenu
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import
androidx.recyclerview.widget.LinearLayoutManager
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin.AdminValueTentangStrokeAdapter
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TentangStrokeDetailModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TentangStrokeListModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityAdminValueTentangStrokeBi
nding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKeteranganBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKonfirmasiBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogValTentangStrokeBinding
import
com.example.edukasipengobatanpadapenderitastrok

```

```

e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.AndroidEntryPoint
import java.util.ArrayList
import javax.inject.Inject

@AndroidEntryPoint
class AdminValueTentangStrokeActivity :
AppCompatActivity() {
    private lateinit var binding:
ActivityAdminValueTentangStrokeBinding
    private val viewModel:
AdminValueTentangStrokeViewModel by
viewModel()
    private lateinit var adapter:
AdminValueTentangStrokeAdapter
    @Inject
    lateinit var loading: LoadingAlertDialog
    private var idHalTentangStroke = "1"
    private lateinit var listHalamanTentangStroke:
ArrayList<String>
    private lateinit var listIdHalamanTentangStroke:
ArrayList<String>
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityAdminValueTentangStrokeBinding.inflate(
layoutInflater)
        setContentView(binding.root)

        setDataSebelumnya()
        setButton()
        fetchData()
        getValueTentangStroke()
        getTambahValueTentangStroke()
        getUpdateValueTentangStroke()
        getHapusValueTentangStroke()
    }

    private fun setDataSebelumnya() {
        val extras = intent.extras
        if (extras != null) {
            val halTentangStroke =
extras.getParcelableArrayList<TentangStrokeListModel>("hal_tentang_stroke")!!
            idHalTentangStroke =
extras.getString("id_hal_tentang_stroke")!!
            listHalamanTentangStroke =
searchDataArrayHalaman(halTentangStroke)
            listIdHalamanTentangStroke =
searchDataArrayId(halTentangStroke)
            val halaman = extras.getString("halaman")!!
            binding.titleHeader.text = halaman
        }
    }

    private fun searchDataArrayId(list:
ArrayList<TentangStrokeListModel>):
ArrayList<String> {

```

```

        val valueList: ArrayList<String> =
arrayListOf()
        for (value in list) {
            valueList.add(value.id_hal_tentang_stroke!!)
        }

        return valueList
    }

    private fun searchDataArrayHalaman(list:
ArrayList<TentangStrokeListModel>):
ArrayList<String> {
        val valueList: ArrayList<String> =
arrayListOf()
        for (value in list) {
            valueList.add(value.halaman!!)
        }

        return valueList
    }

    private fun setButton() {
        binding.apply {
            btnTambah.setOnClickListener {
                setShowDialogTambah()
            }
            btnBack.setOnClickListener {
                finish()
            }
        }
    }

    private fun setShowDialogTambah() {
        val view =
AlertDialogValTentangStrokeBinding.inflate(layout
Inflater)

        val alertDialog =
AlertDialog.Builder(this@AdminValueTentangStrokeActivity)
        alertDialog.setView(view.root)
        .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        view.apply {

            var numberPosition = 0
            var selectedValue = ""

            val tambahDialogHalaman =
binding.titleHeader.text.toString().trim()
            val listTambahDialog : ArrayList<String> =
arrayListOf()

            listTambahDialog.add(tambahDialogHalaman)

            val arrayAdapter =
ArrayAdapter<String>(this@AdminValueTentangStrokeActivity, android.R.layout.simple_spinner_item,
listTambahDialog)

            arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)

```

```

        spHalamanTentangStroke.adapter =
        arrayAdapter

        spHalamanTentangStroke.onItemSelectedListener =
        object: AdapterView.OnItemSelectedListener{
            override fun onItemSelected(
                parent: AdapterView<*>?,
                view: View?,
                position: Int,
                id: Long
            ) {
                numberPosition =
                spHalamanTentangStroke.selectedItemPosition
                selectedValue =
                spHalamanTentangStroke.selectedItem.toString()
            }

            override fun onNothingSelected(parent:
            AdapterView<*>?) {

            }

        }

        spHalamanTentangStroke.adapter =
        arrayAdapter

        btnSimpan.setOnClickListener {
            val idHalTentangStroke =
            idHalTentangStroke
            val judul =
            etEditJudul.text.toString().trim()
            val deskripsi =
            etEditDeskripsi.text.toString().trim()

            postTambahValueTentangStroke(
                idHalTentangStroke, deskripsi, judul
            )
            dialogInputan.dismiss()
        }
        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
        }
    }

    private fun postTambahValueTentangStroke(
        idHalTentangStroke: String, judul: String,
        deskripsi: String
    ) {
        viewModel.postTambahValueTentangStroke(
            idHalTentangStroke, judul, deskripsi
        )
    }

    private fun getTambahValueTentangStroke(){
        viewModel.getTambahValueTentangStroke().observ
        e(this@AdminValueTentangStrokeActivity){result-
        >
            when(result){
                is UIState.Loading->
                loading.alertDialogLoading(this@AdminValueTenta
                ngStrokeActivity)

```

```

                is UIState.Failure->
                setFailureTambahValueTentangStroke(result.messag
                e)
                is UIState.Success->
                setSuccessTambahValueTentangStroke(result.data)
            }
        }

        private fun
        setSuccessTambahValueTentangStroke(data:
        ArrayList<ResponseModel>) {
            loading.alertDialogCancel()
            if(data.isNotEmpty()){
                if(data[0].status=="0"){

                    Toast.makeText(this@AdminValueTentangStrokeA
                    ctivity, "Berhasil Tambah",
                    Toast.LENGTH_SHORT).show()
                    fetchData()
                } else{

                    Toast.makeText(this@AdminValueTentangStrokeA
                    ctivity, data[0].message_response,
                    Toast.LENGTH_SHORT).show()
                }
            } else{

                Toast.makeText(this@AdminValueTentangStrokeA
                ctivity, "Gagal di web",
                Toast.LENGTH_SHORT).show()
            }
        }

        private fun
        setFailureTambahValueTentangStroke(message:
        String) {

            Toast.makeText(this@AdminValueTentangStrokeA
            ctivity, message, Toast.LENGTH_SHORT).show()
            loading.alertDialogCancel()
        }

        private fun fetchData() {

            viewModel.fetchValueTentangStroke(idHalTentang
            Stroke)
        }
        private fun getValueTentangStroke(){

            viewModel.getTentangStrokeList().observe(this@A
            dminValueTentangStrokeActivity){result->
                when(result){
                    is UIState.Loading-> setStartShimmer()
                    is UIState.Failure->
                    setFailureFetchValueTentangStroke(result.message)
                    is UIState.Success->
                    setSuccessFetchValueTentangStroke(result.data)
                }
            }
        }

        private fun
        setFailureFetchValueTentangStroke(message:
        String) {

```

```

Toast.makeText(this@AdminValueTentangStrokeActivity, message, Toast.LENGTH_SHORT).show()
    setStopShimmer()
}

private fun
setSuccessFetch ValueTentangStroke(data:
ArrayList<TentangStrokeDetailModel>) {
    setStopShimmer()
    if(data.isEmpty()){
        setAdapter(data)
    } else{

Toast.makeText(this@AdminValueTentangStrokeActivity, "Tidak ada data",
Toast.LENGTH_SHORT).show()
    }
}

private fun setAdapter(data:
ArrayList<TentangStrokeDetailModel>) {
    adapter =
AdminValueTentangStrokeAdapter(data, object:
OnClickItem.ClickAdminValueTentangStroke{
    override fun clickItemHalaman(halaman:
String, it: View) {
        showClickHalaman(halaman)
    }

    override fun clickItemJudul(judul: String, it:
View) {
        showClickJudul(judul)
    }

    override fun clickItemDeskripsi(deskripsi:
String, it: View) {
        showClickDeskripsi(deskripsi)
    }

    override fun clickItemSetting(pesanan:
TentangStrokeDetailModel, it: View) {
        val popupMenu =
PopupMenu(this@AdminValueTentangStrokeActivity, it)

popupMenu.inflate(R.menu.popup_edit_hapus)

popupMenu.setOnMenuItemClickListener(object :
PopupMenu.OnMenuItemClickListener
{
    override fun
onMenuItemClick(menuItem: MenuItem?): Boolean
{
        when (menuItem!!.itemId) {
            R.id.edit -> {
                setShowDialogEdit(pesanan)
                return true
            }
            R.id.hapus -> {
                setShowDialogHapus(pesanan)
                return true
            }
        }
    }
}
return true

```

```

    }
    })
    popupMenu.show()
}
})

binding.apply {
    rvValTentangStroke.layoutManager =
LinearLayoutManager(
        this@AdminValueTentangStrokeActivity,
        LinearLayoutManager.VERTICAL,
        false
    )
    rvValTentangStroke.adapter = adapter
}

}

private fun showClickHalaman(halaman: String) {
    val view =
AlertDialogKeteranganBinding.inflate(layoutInflater
)

    val alertDialog =
AlertDialog.Builder(this@AdminValueTentangStrokeActivity)
    alertDialog.setView(view.root)
        .setCancelable(false)
    val dialogInputan = alertDialog.create()
    dialogInputan.show()

    view.apply {
        tvTitleKeterangan.text = "Jenis Halaman"
        tvBodyKeterangan.text = halaman
        btnClose.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

private fun showClickJudul(judul: String) {
    val view =
AlertDialogKeteranganBinding.inflate(layoutInflater
)

    val alertDialog =
AlertDialog.Builder(this@AdminValueTentangStrokeActivity)
    alertDialog.setView(view.root)
        .setCancelable(false)
    val dialogInputan = alertDialog.create()
    dialogInputan.show()

    view.apply {
        tvTitleKeterangan.text = "Judul Isi"
        tvBodyKeterangan.text = judul
        btnClose.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

private fun showClickDeskripsi(deskripsi: String)
{

```

```

        val view =
AlertDialogKeteranganBinding.inflate(layoutInflater
)

        val alertDialog =
AlertDialog.Builder(this@AdminValueTentangStro
keActivity)
        alertDialog.setView(view.root)
            .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        view.apply {
            tvTitleKeterangan.text = "Isi Penjelasan"
            tvBodyKeterangan.text = deskripsi
            btnClose.setOnClickListener {
                dialogInputan.dismiss()
            }
        }
    }

    private fun setShowDialogEdit(tentangStroke:
TentangStrokeDetailModel) {
        val view =
AlertDialogValTentangStrokeBinding.inflate(layout
Inflater)

        val alertDialog =
AlertDialog.Builder(this@AdminValueTentangStro
keActivity)
        alertDialog.setView(view.root)
            .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        view.tvTitle.text = "Edit Value Tentang
Stroke"

        view.apply {
            etEditJudul.setText(tentangStroke.judul)
        }
        etEditDeskripsi.setText(tentangStroke.deskripsi)

        var numberPosition = 0
        var selectedValue = ""

        val arrayAdapter =
ArrayAdapter(this@AdminValueTentangStrokeActi
vity, android.R.layout.simple_spinner_item,
listHalamanTentangStroke)

        arrayAdapter.setDropDownViewResource(android.
R.layout.simple_spinner_dropdown_item)

        var id = 0
        for((no, value) in
listIdHalamanTentangStroke.withIndex()){
            if(value ==
tentangStroke.id_hal_tentang_stroke){
                id = no
                Log.d("DetailTAG", "val $no: $value")
            }
        }
        Log.d("DetailTAG", "val $id")
    }

```

```

        spHalamanTentangStroke.setSelection(id)
        spHalamanTentangStroke.adapter =
arrayAdapter

        spHalamanTentangStroke.onItemSelectedListener =
object: AdapterView.OnItemSelectedListener{
            override fun onItemSelected(
                parent: AdapterView<*>?,
                view: View?,
                position: Int,
                id: Long
            ) {
                numberPosition =
                spHalamanTentangStroke.selectedItemPosition
                selectedValue =
                spHalamanTentangStroke.selectedItem.toString()
            }

            override fun onNothingSelected(parent:
AdapterView<*>?) {
            }
        }

        spHalamanTentangStroke.setSelection(id)

        btnSimpan.setOnClickListener {
            val idValueTentangStroke =
tentangStroke.id_val_tentang_stroke!!
            val idHalTentangStroke =
listIdHalamanTentangStroke[numberPosition]
            val judul =
etEditJudul.text.toString().trim()
            val deskripsi =
etEditDeskripsi.text.toString().trim()

            postUpdateValueTentangStroke(
                idValueTentangStroke,
                idHalTentangStroke, judul, deskripsi
            )
            dialogInputan.dismiss()
        }
        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
        }
    }

    private fun postUpdateValueTentangStroke(
        idValueTentangStroke: String,
        idHalTentangStroke: String, judul: String, deskripsi:
String,
    ) {
        viewModel.postUpdateValueTentangStroke(
            idValueTentangStroke, idHalTentangStroke,
            judul, deskripsi
        )
    }

    private fun getUpdateValueTentangStroke(){
        viewModel.getUpdateValueTentangStroke().observe
(this@AdminValueTentangStrokeActivity){result->

```

```

        when(result){
            is UIState.Loading->
loading.alertDialogLoading(this@AdminValueTentangStrokeActivity)
            is UIState.Failure->
setFailureUpdateValueTentangStroke(result.message)
            is UIState.Success->
setSuccessUpdateValueTentangStroke(result.data)
        }
    }
}

private fun
setSuccessUpdateValueTentangStroke(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

Toast.makeText(this@AdminValueTentangStrokeActivity, "Berhasil Update",
Toast.LENGTH_SHORT).show()
            fetchData()
        } else{

Toast.makeText(this@AdminValueTentangStrokeActivity, data[0].message_response,
Toast.LENGTH_SHORT).show()
        }
    } else{

Toast.makeText(this@AdminValueTentangStrokeActivity, "Gagal di web",
Toast.LENGTH_SHORT).show()
    }
}

private fun
setFailureUpdateValueTentangStroke(message:
String) {

Toast.makeText(this@AdminValueTentangStrokeActivity, message, Toast.LENGTH_SHORT).show()
    loading.alertDialogCancel()
}

private fun setShowDialogHapus(pesanan:
TentangStrokeDetailModel) {
    val view =
AlertDialogKonfirmasiBinding.inflate(layoutInflater)
)

    val alertDialog =
AlertDialog.Builder(this@AdminValueTentangStrokeActivity)
        alertDialog.setView(view.root)
        .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

    view.apply {
        tvTitleKonfirmasi.text = "Hapus
${pesanan.judul!!} ?"
        tvBodyKonfirmasi.text = "Judul dan Isi ini

```

akan hapus dan data tidak dapat dipulihkan"

```

        btnKonfirmasi.setOnClickListener {
            val idValueTentangStroke =
pesanan.id_val_tentang_stroke!!

postHapusValueTentangStroke(idValueTentangStroke)

            dialogInputan.dismiss()
        }
        btnBatal.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

private fun
postHapusValueTentangStroke(idValueTentangStroke: String) {

viewModel.postHapusValueTentangStroke(idValueTentangStroke)
}

private fun getHapusValueTentangStroke(){

viewModel.getHapusValueTentangStroke().observe(
this@AdminValueTentangStrokeActivity){result->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@AdminValueTentangStrokeActivity)
        is UIState.Failure->
setFailureHapusValueTentangStroke(result.message)
    }
        is UIState.Success->
setSuccessHapusValueTentangStroke(result.data)
    }
}

private fun
setSuccessHapusValueTentangStroke(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

Toast.makeText(this@AdminValueTentangStrokeActivity, "Berhasil Hapus",
Toast.LENGTH_SHORT).show()
            fetchData()
        } else{

Toast.makeText(this@AdminValueTentangStrokeActivity, data[0].message_response,
Toast.LENGTH_SHORT).show()
        }
    } else{

Toast.makeText(this@AdminValueTentangStrokeActivity, "Gagal di web",
Toast.LENGTH_SHORT).show()
    }
}
}

```

```

        private fun
        setFailureHapusValueTentangStroke(message:
        String) {

        Toast.makeText(this@AdminValueTentangStrokeA
        ctivity, message, Toast.LENGTH_SHORT).show()
        loading.alertDialogCancel()
        }

        private fun setStopShimmer(){
        binding.apply {
        rvValTentangStroke.visibility =
        View.VISIBLE

        smValTentangStroke.stopShimmer()
        smValTentangStroke.visibility =
        View.GONE
        }
        }
        private fun setStartShimmer(){
        binding.apply {
        smValTentangStroke.startShimmer()
        smValTentangStroke.visibility =
        View.VISIBLE

        rvValTentangStroke.visibility = View.GONE
        }
        }
    }

```

Adminvaluetentangstrokeviewmodel

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.tentang_stroke.value_tentang_str
oke

```

```

import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import
com.example.edukasipengobatanpadapenderitastrok
e.data.database.api.ApiService
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TentangStrokeDetailModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TentangStrokeListModel
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.delay
import kotlinx.coroutines.launch
import javax.inject.Inject

```

```

@HiltViewModel
class AdminValueTentangStrokeViewModel
@Inject constructor(
    private val api: ApiService

```

```

): ViewModel() {
    private val _valueTentangStroke =
    MutableLiveData<UIState<ArrayList<TentangStrok
    eDetailModel>>>>()
    private val _postTambahValueTentangStroke =
    MutableLiveData<UIState<ArrayList<ResponseMo
    del>>>>()
    private val _postUpdateValueTentangStroke =
    MutableLiveData<UIState<ArrayList<ResponseMo
    del>>>>()
    private val _postHapusValueTentangStroke =
    MutableLiveData<UIState<ArrayList<ResponseMo
    del>>>>()

    fun
    fetchValueTentangStroke(id_val_tentang_stroke:
    String) {
        viewModelScope.launch {

        _valueTentangStroke.postValue(UIState.Loading)
        delay(1_000)
        try {
            val data = api.getTentangStrokeDetail("",
            id_val_tentang_stroke)

            _valueTentangStroke.postValue(UIState.Success(dat
            a))
        } catch (ex: Exception) {

        _valueTentangStroke.postValue(UIState.Failure("Er
        ror ${ex.message}")
        )
        }
        }

        fun postTambahValueTentangStroke(
        idHalaman:String, judul:String, deskripsi:
        String
        ){
            viewModelScope.launch(Dispatchers.IO) {

            _postTambahValueTentangStroke.postValue(UIStat
            e.Loading)
            delay(1_000)
            try {
                val postTambahValueTentangStroke =
                api.addValueTentangStroke("", idHalaman, judul,
                deskripsi)

            _postTambahValueTentangStroke.postValue(UIStat
            e.Success(postTambahValueTentangStroke))
            } catch (ex: Exception){

            _postTambahValueTentangStroke.postValue(UIStat
            e.Failure("Error: ${ex.message}")
            )
            }
            }

            fun postUpdateValueTentangStroke(
            idValueTentangStroke:String,
            idHalaman:String, judul:String, deskripsi: String
            ){
                viewModelScope.launch(Dispatchers.IO) {

```

```

        _postUpdateValueTentangStroke.postValue(UIState
.Loading)
            delay(1_000)
            try {
                val postTambahValueTentangStroke =
api.postUpdateValueTentangStroke("",
idValueTentangStroke, idHalaman, judul, deskripsi)

        _postUpdateValueTentangStroke.postValue(UIState
.Success(postTambahValueTentangStroke))
            } catch (ex: Exception){

        _postUpdateValueTentangStroke.postValue(UIState
.Failure("Error: ${ex.message}"))
            }
        }

        fun
postHapusValueTentangStroke(id_val_tentang_stro
ke: String){
            viewModelScope.launch(Dispatchers.IO) {

        _postHapusValueTentangStroke.postValue(UIState.
Loading)
            delay(1_000)
            try {
                val postTambahValueTentangStroke =
api.postHapusValueTentangStroke("",
id_val_tentang_stroke)

        _postHapusValueTentangStroke.postValue(UIState.
Success(postTambahValueTentangStroke))
            } catch (ex: Exception){

        _postHapusValueTentangStroke.postValue(UIState.
.Failure("Error: ${ex.message}"))
            }
        }

        fun getTentangStrokeList():
LiveData<UIState<ArrayList<TentangStrokeDetail
Model>>> = _valueTentangStroke
        fun getTambahValueTentangStroke():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postTambahValueTentangStroke
        fun getUpdateValueTentangStroke():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postUpdateValueTentangStroke
        fun getHapusValueTentangStroke():
LiveData<UIState<ArrayList<ResponseModel>>>
= _postHapusValueTentangStroke
    }

```

Adminterapiactivity

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.terapi

import android.Manifest
import android.annotation.SuppressLint
import android.app.Activity
import android.content.Intent

```

```

import android.content.pm.PackageManager
import android.net.Uri
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Environment
import android.provider.OpenableColumns
import android.provider.Settings
import android.view.MenuItem
import android.view.View
import android.widget.PopupMenu
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import
androidx.recyclerview.widget.LinearLayoutManager
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin.AdminTerapiAdapter
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TerapiModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityAdminTerapiBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKeteranganBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogTerapiBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKonfirmasiBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogShowImageBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.main.AdminMainActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KataAcak
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KontrolNavigationDrawer
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import

```

```

com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.AndroidEntryPoint
import
okhttp3.MediaType.Companion.toMediaTypeOrNull
import okhttp3.MultipartBody
import okhttp3.RequestBody
import
okhttp3.RequestBody.Companion.toRequestBody
import javax.inject.Inject

@AndroidEntryPoint
class AdminTerapiActivity : AppCompatActivity() {
    private lateinit var binding:
ActivityAdminTerapiBinding
    private lateinit var kontrolNavigationDrawer:
KontrolNavigationDrawer
    private val viewModel: AdminTerapiViewModel
by viewModels()
    private lateinit var adapter: AdminTerapiAdapter
    private var tempAlertDialog: AlertDialog? = null
    @Inject
    lateinit var loading: LoadingAlertDialog
    private var tempView:
AlertDialogTerapiBinding? = null
    private var fileImage: MultipartBody.Part? = null
    private val kataAcak = KataAcak()
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityAdminTerapiBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setKontrolNavigationDrawer()
        setButton()
        fetchTerapi()
        getTerapi()
        getTambahTerapi()
        getUpdateTerapi()
        getHapusTerapi()
    }

    private fun setKontrolNavigationDrawer() {
        kontrolNavigationDrawer =
KontrolNavigationDrawer(this@AdminTerapiActivi
ty)
        binding.apply {

            kontrolNavigationDrawer.cekSebagai(navView)

            kontrolNavigationDrawer.onClickItemNavigationDr
awer(navView, drawerLayoutMain, ivDrawerView,
this@AdminTerapiActivity)
        }
    }

    private fun setButton() {
        binding.btnTambah.setOnClickListener {
            dialogTambahTerapi()
        }
    }

    private fun dialogTambahTerapi() {

```

```

        val view =
AlertDialogTerapiBinding.inflate(layoutInflater)

        val alertDialog =
AlertDialog.Builder(this@AdminTerapiActivity)
        alertDialog.setView(view.root)
            .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        tempAlertDialog = dialogInputan
        tempView = view

        view.apply {
            etEditGambar.setOnClickListener {
                if(checkPermission()){
                    pickImageFile()
                } else{
                    requestPermission()
                }
            }
            btnSimpan.setOnClickListener {
                var check = true

                if(etEditGambar.text.toString().trim() ==
"Masukkan Gambar".trim()){
                    etEditGambar.error = "Tidak Boleh
Kosong"
                    check = false
                }
                if(etEditJudul.text.isEmpty()){
                    etEditJudul.error = "Tidak Boleh
Kosong"
                    check = false
                }
                if(etEditDeskripsi.text.isEmpty()){
                    etEditDeskripsi.error = "Tidak Boleh
Kosong"
                    check = false
                }
                if(check){
                    val kata =
kataAcak.getHurufDanAngka()
                    val judul = etEditJudul.text.toString()
                    val deskripsi =
etEditDeskripsi.text.toString()
                    postTambahTerapi(kata, judul,
deskripsi, fileImage!!)
                }
            }
            btnBatal.setOnClickListener {
                dialogInputan.dismiss()
                tempAlertDialog = null
            }
        }

        private fun postTambahTerapi(
            kataAcak: String, judul: String, deskripsi:
String, fileImage: MultipartBody.Part
        ) {
            viewModel.postTambahTerapi(
                convertStringToMultipartBody("post_tambah_terapi
"),

```

```

        convertStringToMultipartBody(kataAcak),
        convertStringToMultipartBody(judul),
        convertStringToMultipartBody(deskripsi),
        fileImage
    )
}

private fun getTambahTerapi(){

viewModel.getResponseTambahTerapi().observe(this@AdminTerapiActivity){result->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@AdminTerapiActivity)
        is UIState.Failure->
setFailureTambahTerapi(result.message)
        is UIState.Success->
setSuccessHalamTambahTerapi(result.data)
    }
}

private fun setFailureTambahTerapi(message: String) {
    loading.alertDialogCancel()
    Toast.makeText(this@AdminTerapiActivity, message, Toast.LENGTH_SHORT).show()
}

private fun
setSuccessHalamTambahTerapi(data: ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

Toast.makeText(this@AdminTerapiActivity, "Berhasil Tambah",
Toast.LENGTH_SHORT).show()
            tempAlertDialog!!.dismiss()
            tempAlertDialog = null
            tempView = null
            fetchTerapi()
        } else{

Toast.makeText(this@AdminTerapiActivity, data[0].message_response,
Toast.LENGTH_SHORT).show()
            }
        } else{
            Toast.makeText(this@AdminTerapiActivity, "ada error di API", Toast.LENGTH_SHORT).show()
        }
    }

private fun fetchTerapi() {
    viewModel.fetchTerapi()
}

private fun getTerapi(){

viewModel.getTerapi().observe(this@AdminTerapiActivity){result->
    when(result){

```

```

        is UIState.Loading-> setStartShimmer()
        is UIState.Failure->
setFailureTerapi(result.message)
        is UIState.Success->
setSuccessTerapi(result.data)
    }
}

private fun setFailureTerapi(message: String) {
    setStopShimmer()
    Toast.makeText(this@AdminTerapiActivity, message, Toast.LENGTH_SHORT).show()
}

private fun setSuccessTerapi(data: ArrayList<TerapiModel>) {
    setStopShimmer()
    if(data.isNotEmpty()){
        setAdapter(data)
    } else{
        Toast.makeText(this@AdminTerapiActivity, "Data Kosong", Toast.LENGTH_SHORT).show()
    }
}

private fun setAdapter(data: ArrayList<TerapiModel>) {
    adapter = AdminTerapiAdapter(data, object :
OnClickItem.ClickAdminTerapi{
        override fun clickItemJudul(namaTerapi: String, it: View) {
            showClickText("Nama Terapi", namaTerapi)
        }

        override fun clickItemDeskripsi(deskripsi: String, it: View) {
            showClickText("Deskripsi Terapi", deskripsi)
        }

        override fun clickItemGambar(gambar: String, title: String, it: View) {
            setShowImage(gambar, title)
        }

        override fun clickItemSetting(terapi: TerapiModel, it: View) {
            val popupMenu =
PopupMenu(this@AdminTerapiActivity, it)
            popupMenu.inflate(R.menu.popup_edit_hapus)

            popupMenu.setOnMenuItemClickListener(object :
PopupMenu.OnMenuItemClickListener
{
                override fun
onMenuItemClick(menuItem: MenuItem?): Boolean
{
                    when (menuItem!!.itemId) {
                        R.id.edit -> {
                            dialogEditTerapi(terapi)
                            return true
                        }

```

```

        R.id.hapus -> {
            dialogHapusTerapi(terapi.id_terapi!!,
            terapi.nama_terapi!!)
                return true
            }
        }
        return true
    }

    })
    popupMenu.show()
}

})

binding.apply {
    rvTerapi.layoutManager =
    LinearLayoutManager(this@AdminTerapiActivity,
    LinearLayoutManager.VERTICAL, false)
    rvTerapi.adapter = adapter
}

private fun showClickText(keterangan:String,
judul: String) {
    val view =
    AlertDialogKeteranganBinding.inflate(layoutInflater
    )

    val alertDialog =
    AlertDialog.Builder(this@AdminTerapiActivity)
    alertDialog.setView(view.root)
    .setCancelable(false)
    val dialogInputan = alertDialog.create()
    dialogInputan.show()

    view.apply {
        tvTitleKeterangan.text = keterangan
        tvBodyKeterangan.text = judul
        btnClose.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

private fun setShowImage(gambar: String,
title:String) {
    val view =
    AlertDialogShowImageBinding.inflate(layoutInflater
    )

    val alertDialog =
    AlertDialog.Builder(this@AdminTerapiActivity)
    alertDialog.setView(view.root)
    .setCancelable(false)
    val dialogInputan = alertDialog.create()
    dialogInputan.show()

    view.apply {
        tvTitle.text = "Gambar $title"
        btnClose.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

```

```

        Glide.with(this@AdminTerapiActivity)
        .load("${Constant.BASE_URL}${Constant.LOCATION_GAMBAR}$gambar") // URL Gambar
        .error(R.drawable.gambar_error_image)
        .into(view.ivShowImage) // imageView
        mana yang akan diterapkan
    }

    private fun dialogEditTerapi(terapi: TerapiModel)
    {
        val view =
        AlertDialogTerapiBinding.inflate(layoutInflater)

        val alertDialog =
        AlertDialog.Builder(this@AdminTerapiActivity)
        alertDialog.setView(view.root)
        .setCancelable(false)
        val dialogInputan = alertDialog.create()
        dialogInputan.show()

        view.tvTitle.text = "Update Halaman Galeri
        Herbal"

        tempAlertDialog = dialogInputan
        tempView = view

        view.apply {
            etEditJudul.setText(terapi.nama_terapi)
            etEditDeskripsi.setText(terapi.deskripsi)

            etEditGambar.setOnClickListener {
                if(checkPermission()){
                    pickImageFile()
                } else{
                    requestPermission()
                }
            }

            btnSimpan.setOnClickListener {
                var check = true

                if(etEditJudul.text.isEmpty()){
                    etEditJudul.error = "Tidak Boleh
                    Kosong"
                    check = false
                }
                if(etEditDeskripsi.text.isEmpty()){
                    etEditDeskripsi.error = "Tidak Boleh
                    Kosong"
                    check = false
                }
                if(check){
                    if(fileImage != null){
                        postEditTerapi(
                            terapi.id_terapi!!,
                            kataAcak.getHurufDanAngka(),
                            etEditJudul.text.toString(),
                            etEditDeskripsi.text.toString(),
                            fileImage!!
                        )
                    } else{
                        postEditTerapiNoHaveImage(

```

```

                terapi.id_terapi!!,
                etEditJudul.text.toString(),
                etEditDeskripsi.text.toString()
            )
        }
    }
    btnBatal.setOnClickListener {
        dialogInputan.dismiss()
        tempAlertDialog = null
    }
}

private fun postEditTerapi(
    idHalTerapi: String, kataAcak: String, judul:
    String, deskripsi: String, fileImage:
    MultipartBody.Part
) {
    viewModel.postUpdateTerapi(
        convertStringToMultipartBody("post_update_terapi"
        ),
        convertStringToMultipartBody(idHalTerapi),
        convertStringToMultipartBody(kataAcak),
        convertStringToMultipartBody(judul),
        convertStringToMultipartBody(deskripsi),
        fileImage
    )
}

private fun postEditTerapiNoHaveImage(
    idHalTerapi: String, judul: String, deskripsi:
    String
) {
    viewModel.postUpdateTerapiNoHaveImage(
        "post_update_terapi_no_have_image",
        idHalTerapi, judul, deskripsi
    )
}

private fun getUpdateTerapi(){

viewModel.getResponseUpdateTerapi().observe(this
@AdminTerapiActivity){result->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@AdminTerapiActi
vity)
        is UIState.Failure->
setFailureUpdateTerapi(result.message)
        is UIState.Success->
setSuccessHalamUpdateanTerapi(result.data)
    }
}

private fun setFailureUpdateTerapi(message:
String) {
    loading.alertDialogCancel()
    Toast.makeText(this@AdminTerapiActivity,
message, Toast.LENGTH_SHORT).show()
}

private fun

```

```

setSuccessHalamUpdateanTerapi(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

Toast.makeText(this@AdminTerapiActivity,
"Berhasil Update", Toast.LENGTH_SHORT).show()
        tempAlertDialog!!.dismiss()
        tempAlertDialog = null
        fetchTerapi()
    } else{

Toast.makeText(this@AdminTerapiActivity,
data[0].message_response,
Toast.LENGTH_SHORT).show()
    }
    } else{
        Toast.makeText(this@AdminTerapiActivity,
"ada error di aplikasi",
Toast.LENGTH_SHORT).show()
    }
}

private fun dialogHapusTerapi(idHalTerapi:
String, namaTerapi: String) {
    val view =
AlertDialogKonfirmasiBinding.inflate(layoutInflater
)

    val alertDialog =
AlertDialog.Builder(this@AdminTerapiActivity)
    alertDialog.setView(view.root)
    .setCancelable(false)
    val dialogInputan = alertDialog.create()
    dialogInputan.show()

    tempAlertDialog = dialogInputan

    view.apply {
        tvTitleKonfirmasi.text = "Yakin Hapus
Terapi ini?"
        tvBodyKonfirmasi.text = "$namaTerapi akan
dihapus dan data yang ada di dalamnya akan ikut
terhapus. data yang telah terhapus tidak dapat di
kembalikan"

        btnKonfirmasi.setOnClickListener {
            postHapusTerapi(idHalTerapi!!)
        }
        btnBatal.setOnClickListener {
            tempAlertDialog = null
            dialogInputan.dismiss()
        }
    }
}

private fun postHapusTerapi(idHalTerapi: String)
{
    viewModel.postHapusTerapi(idHalTerapi)
}

private fun getHapusTerapi(){

viewModel.getResponseHapusTerapi().observe(this

```

```

@AdminTerapiActivity){result->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@AdminTerapiActi
vity)
        is UIState.Failure->
setFailureHapusTerapi(result.message)
        is UIState.Success->
setSuccessHapusTerapi(result.data)
    }
}

private fun setFailureHapusTerapi(message:
String) {
    loading.alertDialogCancel()
    Toast.makeText(this@AdminTerapiActivity,
message, Toast.LENGTH_SHORT).show()
}

private fun setSuccessHapusTerapi(data:
ArrayList<ResponseModel>) {
    loading.alertDialogCancel()
    if(data.isNotEmpty()){
        if(data[0].status=="0"){

Toast.makeText(this@AdminTerapiActivity,
"Berhasil Hapus", Toast.LENGTH_SHORT).show()
            tempAlertDialog!!.dismiss()
            tempAlertDialog = null
            fetchTerapi()
        } else{

Toast.makeText(this@AdminTerapiActivity,
data[0].message_response,
Toast.LENGTH_SHORT).show()
        }
    } else{
        Toast.makeText(this@AdminTerapiActivity,
"ada error di aplikasi",
Toast.LENGTH_SHORT).show()
    }
}

private fun setStopShimmer(){
    binding.apply {
        rvTerapi.visibility = View.VISIBLE

        smTerapi.stopShimmer()
        smTerapi.visibility = View.GONE
    }
}
private fun setStartShimmer(){
    binding.apply {
        smTerapi.startShimmer()
        smTerapi.visibility = View.VISIBLE

        rvTerapi.visibility = View.GONE
    }
}

// permission add image
private fun requestPermission(){
    if(Build.VERSION.SDK_INT >=

```

```

Build.VERSION_CODES.R){
        if
(Environment.isExternalStorageManager()) {
            startActivity(Intent(this,
AdminTerapiAdapter::class.java))
        } else { //request for the permission
            val intent =
Intent(Settings.ACTION_MANAGE_APP_ALL_FIL
ES_ACCESS_PERMISSION)
            val uri = Uri.fromParts("package",
packageName, null)
            intent.data = uri
            startActivity(intent)
        }
    } else{
        ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.WRITE_EXTERNAL_
STORAGE,
Manifest.permission.READ_EXTERNAL_STORAGE
),
Constant.STORAGE_PERMISSION_CODE
    )
}

private fun pickImageFile() {
    val intent =
Intent(Intent.ACTION_OPEN_DOCUMENT).apply
{
        type = "image/*"
    }

    startActivityForResult(intent,
Constant.IMAGE_CODE)
}

override fun onActivityResult(requestCode: Int,
resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode,
resultCode, data)

    if (requestCode == Constant.IMAGE_CODE
&& resultCode == Activity.RESULT_OK && data
!= null) {
        // Mendapatkan URI file PDF yang dipilih
        val fileUri = data.data!!

        val nameImage = getNameFile(fileUri)

        tempView?.let {
            it.etEditGambar.text = nameImage
        }

        // Mengirim file PDF ke website
        menggunakan Retrofit
        fileImage = uploadImageToStorage(fileUri,
nameImage, "galeri_herbal_image")
    }
}

private fun getNameFile(uri: Uri): String {
    val cursor = contentResolver.query(uri, null,
null, null, null)

```

```

        val nameIndex =
        cursor?.getColumnIndex(OpenableColumns.DISPLAY_NAME)
        cursor?.moveToFirst()
        val name = cursor?.getString(nameIndex!!)
        cursor?.close()
        return name!!
    }

    @SuppressLint("Recycle")
    private fun uploadImageToStorage(imageUri: Uri?, fileName: String, nameAPI:String): MultipartBody.Part? {
        var pdfPart : MultipartBody.Part? = null
        imageUri?.let {
            val file =
            contentResolver.openInputStream(imageUri)?.readBytes()

            if (file != null) {
                // Membuat objek RequestBody dari file PDF
                val requestFile =
                file.toRequestBody("application/pdf".toMediaTypeOrNull())
                // Membuat objek MultipartBody.Part untuk file PDF
                pdfPart =
                MultipartBody.Part.createFormData("materi_pdf", fileName, requestFile)

                pdfPart =
                convertFileToMultipartBody(file, fileName, nameAPI)
            }
            return pdfPart
        }

        private fun convertFileToMultipartBody(file: ByteArray, fileName: String, nameAPI:String): MultipartBody.Part? {
            // val requestFile =
            file.toRequestBody("application/pdf".toMediaTypeOrNull())
            val requestFile =
            file.toRequestBody("application/image".toMediaTypeOrNull())
            val filePart =
            MultipartBody.Part.createFormData(nameAPI, fileName, requestFile)

            return filePart
        }

        private fun convertStringToMultipartBody(data: String): RequestBody {
            return RequestBody.create("multipart/form-data".toMediaTypeOrNull(), data)
        }

        private fun checkPermission(): Boolean {
            return if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
                //Android is 11(R) or above

```

```

                Environment.isExternalStorageManager()
            }
            else {
                //Android is below 11(R)
                val write =
                ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE_EXTERNAL_STORAGE)
                val read =
                ContextCompat.checkSelfPermission(this, Manifest.permission.READ_EXTERNAL_STORAGE)
                write ==
                PackageManager.PERMISSION_GRANTED && read == PackageManager.PERMISSION_GRANTED
            }
        }

        override fun onBackPressed() {
            super.onBackPressed()

            startActivity(Intent(this@AdminTerapiActivity, AdminMainActivity::class.java))
            finish()
        }
    }

    AdminTerapiViewModel

    package
    com.example.edukasipengobatanpadapenderitastroke.ui.activity.admin.terapi

    import androidx.lifecycle.LiveData
    import androidx.lifecycle.MutableLiveData
    import androidx.lifecycle.ViewModel
    import androidx.lifecycle.ViewModelScope
    import
    com.example.edukasipengobatanpadapenderitastroke.data.database.api.ApiService
    import
    com.example.edukasipengobatanpadapenderitastroke.data.model.TerapiModel
    import
    com.example.edukasipengobatanpadapenderitastroke.data.model.ResponseModel
    import
    com.example.edukasipengobatanpadapenderitastroke.utils.network.UISState
    import dagger.hilt.android.lifecycle.HiltViewModel
    import kotlinx.coroutines.Dispatchers
    import kotlinx.coroutines.delay
    import kotlinx.coroutines.launch
    import okhttp3.MultipartBody
    import okhttp3.RequestBody
    import javax.inject.Inject

    @HiltViewModel
    class AdminTerapiViewModel @Inject constructor(
        private val api: ApiService
    ): ViewModel() {
        private val _terapi =
        MutableLiveData<UISState<ArrayList<TerapiModel>>>()
        private val _postTambahTerapi =
        MutableLiveData<UISState<ArrayList<ResponseModel>>>()

```

```

        private val _postUpdateTerapi =
MutableLiveData<UIState<ArrayList<ResponseMo
del>>>()
        private val _postHapusTerapi =
MutableLiveData<UIState<ArrayList<ResponseMo
del>>>()

        fun fetchTerapi() {
            viewModelScope.launch {
                _terapi.postValue(UIState.Loading)
                delay(1_000)
                try {
                    val data = api.getTerapi("")
                    _terapi.postValue(UIState.Success(data))
                } catch (ex: Exception) {
                    _terapi.postValue(UIState.Failure("Error
${ex.message}"))
                }
            }

            fun postTambahTerapi(
                post: RequestBody, kataAcak: RequestBody,
                judul: RequestBody, deskripsi: RequestBody,
                gambar: MultipartBody.Part
            ){
                viewModelScope.launch(Dispatchers.IO) {

                    _postTambahTerapi.postValue(UIState.Loading)
                    delay(1_000)
                    try {
                        val postTambahTerapi =
api.addTerapi(post, kataAcak, judul, deskripsi,
gambar)

                        _postTambahTerapi.postValue(UIState.Success(post
TambahTerapi))
                    } catch (ex: Exception){

                        _postTambahTerapi.postValue(UIState.Failure("Err
or: ${ex.message}"))
                    }
                }

                fun postUpdateTerapi(
                    post: RequestBody, id_hal_galeri_herbal:
RequestBody, kataAcak: RequestBody, judul:
RequestBody, deskripsi: RequestBody, gambar:
MultipartBody.Part
                ){
                    viewModelScope.launch(Dispatchers.IO) {

                        _postUpdateTerapi.postValue(UIState.Loading)
                        delay(1_000)
                        try {
                            val postTambahTerapi =
api.postUpdateTerapi(
                                post, id_hal_galeri_herbal, kataAcak,
                                judul, deskripsi, gambar
                            )

                            _postUpdateTerapi.postValue(UIState.Success(post
TambahTerapi))
                        } catch (ex: Exception){

```

```

                            _postUpdateTerapi.postValue(UIState.Failure("Erro
r: ${ex.message}"))
                        }
                    }
                }

                fun postUpdateTerapiNoHaveImage(
                    post: String, id_hal_galeri_herbal: String, judul:
String, deskripsi: String
                ){
                    viewModelScope.launch(Dispatchers.IO) {

                        _postUpdateTerapi.postValue(UIState.Loading)
                        delay(1_000)
                        try {
                            val postTambahTerapi =
api.postUpdateTerapiNoHaveImage(
                                post, id_hal_galeri_herbal, judul,
                                deskripsi
                            )

                            _postUpdateTerapi.postValue(UIState.Success(post
TambahTerapi))
                        } catch (ex: Exception){

                            _postUpdateTerapi.postValue(UIState.Failure("Erro
r: ${ex.message}"))
                        }
                    }
                }

                fun postHapusTerapi(id_hal_galeri_herbal:
String){
                    viewModelScope.launch(Dispatchers.IO) {

                        _postHapusTerapi.postValue(UIState.Loading)
                        delay(1_000)
                        try {
                            val postTambahTerapi =
api.postHapusTerapi("", id_hal_galeri_herbal)

                            _postHapusTerapi.postValue(UIState.Success(postT
ambahTerapi))
                        } catch (ex: Exception){

                            _postHapusTerapi.postValue(UIState.Failure("Error:
${ex.message}"))
                        }
                    }
                }

                fun getTerapi():
LiveData<UIState<ArrayList<TerapiModel>>> =
                _terapi
                    fun getResponseTambahTerapi():
LiveData<UIState<ArrayList<ResponseModel>>>
                    = _postTambahTerapi
                    fun getResponseUpdateTerapi():
LiveData<UIState<ArrayList<ResponseModel>>>
                    = _postUpdateTerapi
                    fun getResponseHapusTerapi():
LiveData<UIState<ArrayList<ResponseModel>>>
                    = _postHapusTerapi
                }
            }

```

AdminTestimoniActivity

```
package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.testimoni
```

```
import android.annotation.SuppressLint
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import
androidx.recyclerview.widget.LinearLayoutManager
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin.AdminTestimoniAdapter
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TestimoniModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityAdminTestimoniBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogShowImageBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.main.MainActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KontrolNavigationDrawer
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.SharedPreferencesLogin
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject
```

```
@AndroidEntryPoint
class AdminTestimoniActivity :
AppCompatActivity() {
    private lateinit var binding:
ActivityAdminTestimoniBinding
    private lateinit var kontrolNavigationDrawer:
KontrolNavigationDrawer
    private val viewModel:
AdminTestimoniViewModel by viewModels()
    private lateinit var adapter:
```

```
AdminTestimoniAdapter
    private lateinit var sharedPreferencesLogin:
SharedPreferencesLogin
    @Inject
    lateinit var loading: LoadingAlertDialog
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityAdminTestimoniBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setSharedPreferencesLogin()
        setNavigationDrawer()
        fetchTestimoni()
        getTestimoni()
    }

    @SuppressLint("SetTextI18n")
    private fun setSharedPreferencesLogin() {
        sharedPreferencesLogin =
SharedPreferencesLogin(this@AdminTestimoniActi
vity)
    }

    private fun setNavigationDrawer() {
        kontrolNavigationDrawer =
KontrolNavigationDrawer(this@AdminTestimoniAc
tivity)
        binding.apply {

            kontrolNavigationDrawer.cekSebagai(navView)

            kontrolNavigationDrawer.onClickItemNavigationDr
awer(navView, drawerLayoutMain, ivDrawerView,
this@AdminTestimoniActivity)
        }
    }

    private fun fetchTestimoni() {
        viewModel.fetchTestimoni()
    }

    private fun getTestimoni() {
        viewModel.getTestimoni().observe(this@AdminTes
timoniActivity){result->
            when(result){
                is UISState.Loading-> setStartShimmer()
                is UISState.Failure->
setFailureTestimoni(result.message)
                is UISState.Success->
setSuccessTestimoni(result.data)
            }
        }
    }

    private fun setFailureTestimoni(message: String)
{
        setStopShimmer()

        Toast.makeText(this@AdminTestimoniActivity,
message, Toast.LENGTH_SHORT).show()
    }
}
```

```

        private fun setSuccessTestimoni(data:
        ArrayList<TestimoniModel>) {
            setAdapter(data)
        }

        private fun setAdapter(data:
        ArrayList<TestimoniModel>) {
            adapter = AdminTestimoniAdapter(data, object
            : OnClickItem.ClickTestimoni{
                override fun clickGambar(gambar: String,
                nama: String, it: View) {
                    setShowImage(gambar, nama)
                }
            })
            binding.apply {
                rvTestimoni.layoutManager =
                LinearLayoutManager(this@AdminTestimoniActivi
                ty, LinearLayoutManager.VERTICAL, false)
                rvTestimoni.adapter = adapter
            }
            setStopShimmer()
        }

        private fun setShowImage(gambar: String,
        title:String) {
            val view =
            AlertDialogShowImageBinding.inflate(layoutInflater)
            val alertDialog =
            AlertDialog.Builder(this@AdminTestimoniActivity)
            alertDialog.setView(view.root)
            .setCancelable(false)
            val dialogInputan = alertDialog.create()
            dialogInputan.show()

            view.apply {
                tvTitle.text = "Gambar $title"
                btnClose.setOnClickListener {
                    dialogInputan.dismiss()
                }
            }

            Glide.with(this@AdminTestimoniActivity)

            .load("${Constant.BASE_URL}${Constant.LOCAT
            ION_GAMBAR}$gambar") // URL Gambar
            .error(R.drawable.gambar_error_image)
            .into(view.ivShowImage) // imageView
            mana yang akan diterapkan

        }

        private fun setStopShimmer(){
            binding.apply {
                rvTestimoni.visibility = View.VISIBLE

                smTestimoni.stopShimmer()
                smTestimoni.visibility = View.GONE
            }
        }

        private fun setStartShimmer(){
            binding.apply {
                smTestimoni.startShimmer()
            }
        }

```

```

                smTestimoni.visibility = View.VISIBLE
            }
            rvTestimoni.visibility = View.GONE
        }

        override fun onBackPressed() {
            super.onBackPressed()

            startActivity(Intent(this@AdminTestimoniActivity,
            MainActivity::class.java))
            finish()
        }

        AdminTestimoniViewModel
        package
        com.example.edukasipengobatanpadapenderitastrok
        e.ui.activity.admin.testimoni

        import androidx.lifecycle.LiveData
        import androidx.lifecycle.MutableLiveData
        import androidx.lifecycle.ViewModel
        import androidx.lifecycle.viewModelScope
        import
        com.example.edukasipengobatanpadapenderitastrok
        e.data.database.api.ApiService
        import
        com.example.edukasipengobatanpadapenderitastrok
        e.data.model.ResponseModel
        import
        com.example.edukasipengobatanpadapenderitastrok
        e.data.model.TestimoniModel
        import
        com.example.edukasipengobatanpadapenderitastrok
        e.utils.network.UISState
        import dagger.hilt.android.lifecycle.HiltViewModel
        import kotlinx.coroutines.Dispatchers
        import kotlinx.coroutines.delay
        import kotlinx.coroutines.launch
        import javax.inject.Inject

        @HiltViewModel
        class AdminTestimoniViewModel @Inject
        constructor(
            private val api: ApiService
        ): ViewModel() {
            private val _testimoni =
            MutableLiveData<UISState<ArrayList<TestimoniMo
            del>>>()>()
            fun fetchTestimoni() {
                viewModelScope.launch {
                    _testimoni.postValue(UISState.Loading)
                    delay(1_000)
                    try {
                        val data = api.getTestimoni("")
                        _testimoni.postValue(UISState.Success(data))
                    } catch (ex: Exception) {
                        _testimoni.postValue(UISState.Failure("Error
                        ${ex.message}"))
                    }
                }
            }

            fun getTestimoni():

```

```

LiveData<UIState<ArrayList<TestimoniModel>>>
= _testimoni
}

```

LoginActivity

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.login

```

```

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.widget.Toast
import androidx.activity.viewModels
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.SharedPreferencesLogin
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.UsersModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityLoginBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.admin.main.AdminMainActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.registrasi.RegistrasiActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.main.MainActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

```

```

@AndroidEntryPoint
class LoginActivity : AppCompatActivity() {
    private lateinit var loginBinding:
ActivityLoginBinding
    private lateinit var sharedPreferencesLogin:
SharedPreferencesLogin
    @Inject
    lateinit var loading : LoadingAlertDialog
    private val loginViewModel: LoginViewModel
by viewModels()

```

```

    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        loginBinding =
ActivityLoginBinding.inflate(layoutInflater)
        setContentView(loginBinding.root)

```

```

        konfigurasiUtils()
        button()
        getData()
    }

```

```

    private fun konfigurasiUtils() {
        sharedPreferencesLogin =
SharedPreferencesLogin(this@LoginActivity)
    }

```

```

    private fun button(){
        btnLogin()
    }

```

```

    private fun btnLogin() {
        loginBinding.apply {
            btnLogin.setOnClickListener{
                if(etUsername.text.isNotEmpty() &&
etPassword.text.isNotEmpty()){
loading.alertDialogLoading(this@LoginActivity)
cekUsers(etUsername.text.toString(),
etPassword.text.toString())
                }
            } else{
                if(etUsername.text.isEmpty()){
                    etUsername.error = "Masukkan
Username"
                }
                if(etPassword.text.isEmpty()){
                    etPassword.error = "Masukkan
Password"
                }
            }
        }
    }

```

```

    private fun cekUsers(username: String, password:
String) {
        loginViewModel.fetchDataUser(username,
password)
    }

```

```

    private fun getData(){
loginViewModel.getDataUser().observe(this@Login
Activity){result ->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@LoginActivity)
        is UIState.Success->
setSuccessDataUser(result.data)
        is UIState.Failure ->
setFailureDataUser(result.message)
    }
}
}

```

```

    private fun setFailureDataUser(message: String) {
        Toast.makeText(this@LoginActivity, message,
Toast.LENGTH_SHORT).show()
        loading.alertDialogCancel()
    }

```

```

    private fun setSuccessDataUser(data:
ArrayList<UsersModel>) {
        loading.alertDialogCancel()
        if(data.isNotEmpty()){
            successFetchLogin(data)
        }
    }

```

```

    } else{
        Toast.makeText(this@LoginActivity, "Data
tidak ditemukan \nPastikan Username dan Password
ada", Toast.LENGTH_SHORT).show()
    }
}

private fun successFetchLogin(userModel:
ArrayList<UsersModel>){
    var valueIdUser = 0
    userModel[0].idUser?.let {
        valueIdUser = it.toInt()
    }
    val valueNama = userModel[0].nama.toString()
    val valueNomorHp =
userModel[0].nomorHp.toString()
    val valueUsername =
userModel[0].username.toString()
    val valuePassword =
userModel[0].password.toString()
    val valueSebagai=
userModel[0].sebagai.toString()

    try{
        Toast.makeText(this@LoginActivity, "Login
Berhasil", Toast.LENGTH_SHORT).show()

        sharedPreferencesLogin.setLogin(valueIdUser,
valueNama, valueNomorHp, valueUsername,
valuePassword, valueSebagai)
        if(valueSebagai=="admin"){
            startActivity(Intent(this@LoginActivity,
AdminMainActivity::class.java))
            finish()
        } else{
            Toast.makeText(this@LoginActivity,
"Tolong periksa username dan password",
Toast.LENGTH_SHORT).show()
        }
    } catch (ex: Exception){
        Toast.makeText(this@LoginActivity, "gagal:
$ex", Toast.LENGTH_SHORT).show()
    }
}

override fun onBackPressed() {
    super.onBackPressed()
    startActivity(Intent(this@LoginActivity,
MainActivity::class.java))
    finish()
}

Loginviewwmodel
package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.login

import android.util.Log
import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import
com.example.edukasipengobatanpadapenderitastrok
e.data.database.api.ApiService
import

```

```

com.example.edukasipengobatanpadapenderitastrok
e.data.model.UsersModel
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch
import javax.inject.Inject

```

```

@HiltViewModel
class LoginViewModel @Inject constructor(
    private val api: ApiService
): ViewModel() {
    private var _user =
MutableLiveData<UISState<ArrayList<UsersModel>
>>()

```

```

    fun fetchDataUser(username:String,
password:String){
        viewModelScope.launch(Dispatchers.IO) {
            try {
                val data = api.getUser("", username,
password)
                _user.postValue(UISState.Success(data))

                Log.d("TAG", "fetchDataUser: berhasil
$data ")
            } catch (ex: Exception){
                _user.postValue(UISState.Failure("Error
bang! ${ex.message}"))
                Log.d("TAG", "fetchDataUser:
${ex.message}")
            }
        }
    }

```

```

    fun getDataUser():
LiveData<UISState<ArrayList<UsersModel>>> =
_user
}

```

RegistrasiActivity

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.registrasi

```

```

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import androidx.activity.viewModels
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityRegistrasiBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.login.LoginActivity
import

```

```

com.example.edukasipengobatanpadapenderitastrok
e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

```

```

@AndroidEntryPoint
class RegistrasiActivity : AppCompatActivity() {
    private lateinit var binding:
    ActivityRegistrasiBinding
    @Inject
    lateinit var loading: LoadingAlertDialog
    private val viewModel : RegistrasiViewModel by
    viewModels()
    override fun onCreate(savedInstanceState:
    Bundle?) {
        super.onCreate(savedInstanceState)

        binding =
        ActivityRegistrasiBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setButton()
        postRegistrasiUser()
    }

    private fun setButton() {
        binding.apply {
            tvLogin.setOnClickListener{

startActivity(Intent(this@RegistrasiActivity,
LoginActivity::class.java))
                finish()
            }
            btnRegistrasi.setOnClickListener {
                buttonRegistrasi()
            }
        }
    }
}

```

```

    private fun buttonRegistrasi() {
        binding.apply {
            var cek = false
            if (etEditNama.toString().isEmpty()) {
                etEditNama.error = "Tidak Boleh
Kosong"
                cek = true
            }
            if (etEditNomorHp.toString().isEmpty()) {
                etEditNomorHp.error = "Tidak Boleh
Kosong"
                cek = true
            }
            if (etEditUsername.toString().isEmpty()) {
                etEditUsername.error = "Tidak Boleh
Kosong"
                cek = true
            }
            if (etEditPassword.toString().isEmpty()) {
                etEditPassword.error = "Tidak Boleh
Kosong"
                cek = true
            }
        }
    }
}

```

```

        if (!cek) {
            postTambahData(
                etEditNama.text.toString().trim(),
                etEditNomorHp.text.toString().trim(),
                etEditUsername.text.toString().trim(),
                etEditPassword.text.toString().trim()
            )
        }
    }

    private fun postTambahData(nama: String,
    nomorHp: String, username: String, password:
    String){
        viewModel.postRegistrasiUser(nama,
    nomorHp, username, password, "user")
    }

    private fun postRegistrasiUser(){

viewModel.getRegistrasiUser().observe(this@Regist
    rasiActivity){ result ->
        when(result){
            is UISState.Loading->
            loading.alertDialogLoading(this@RegistrasiActivity
            )
            is UISState.Failure->
            responseFailureRegistrasiUser(result.message)
            is UISState.Success->
            responseSuccessRegiserUser(result.data)
        }
    }

    private fun
    responseFailureRegistrasiUser(message: String) {
        Toast.makeText(this@RegistrasiActivity,
    message, Toast.LENGTH_SHORT).show()
        loading.alertDialogCancel()
    }

    private fun responseSuccessRegiserUser(data:
    ArrayList<ResponseModel>) {
        if (data.isNotEmpty()){
            if (data[0].status == "0"){
                Toast.makeText(this@RegistrasiActivity,
    "Berhasil melakukan registrasi",
    Toast.LENGTH_SHORT).show()
            } else{
                Toast.makeText(this@RegistrasiActivity,
    data[0].message_response,
    Toast.LENGTH_SHORT).show()
            }
        } else{
            Toast.makeText(this@RegistrasiActivity,
    "Maaf gagal", Toast.LENGTH_SHORT).show()
        }
        loading.alertDialogCancel()
    }
}

Registerviewwmodel
package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.registrasi

```

```

import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import
com.example.edukasipengobatanpadapenderitastrok
e.data.database.api.ApiService
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch
import javax.inject.Inject

@HiltViewModel
class RegistrasiViewModel @Inject constructor(
    private val api: ApiService
): ViewModel() {
    private var _registerUser =
MutableLiveData<UISState<ArrayList<ResponseMo
del>>>()

    fun postRegistrasiUser(nama:String,
nomorHp:String, username:String, password:String,
sebagai:String){
        viewModelScope.launch(Dispatchers.IO) {
            try {
                val registerUser = api.addUser("", nama,
nomorHp, username, password, sebagai)

                _registerUser.postValue(UISState.Success(registerUs
er))
            } catch (ex: Exception){

                _registerUser.postValue(UISState.Failure("Gagal :
${ex.message}"))
            }
        }

        fun getRegistrasiUser():
LiveData<UISState<ArrayList<ResponseModel>>>
= _registerUser
    }
AkunActivity
package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.akun

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import

```

```

com.example.edukasipengobatanpadapenderitastrok
e.data.model.UsersModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityAkunBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogAkunBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KontrolNavigationDrawer
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.SharedPreferencesLogin
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

@AndroidEntryPoint
class AkunActivity : AppCompatActivity() {
    private lateinit var binding : ActivityAkunBinding
    private lateinit var kontrolNavigationDrawer:
KontrolNavigationDrawer
    private lateinit var sharedPreferences:
SharedPreferencesLogin
    private val viewModel: AkunViewModel by
viewModels()
    @Inject
    lateinit var loading: LoadingAlertDialog
    private lateinit var tempUser: UsersModel
    private var tempDialog: AlertDialog? = null
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityAkunBinding.inflate(layoutInflater)
        setContentView(binding.root)

        tempUser = UsersModel()
        setSharedPreferences()
        setKontrolNavigationDrawer()
        setData()
        button()
        getPostUpdateData()
    }

    private fun setSharedPreferences() {
        sharedPreferences =
SharedPreferencesLogin(this@AkunActivity)
    }

    private fun setKontrolNavigationDrawer() {
        binding.apply {
            kontrolNavigationDrawer =
KontrolNavigationDrawer(this@AkunActivity)

            kontrolNavigationDrawer.cekSebagai(navView)

            kontrolNavigationDrawer.onClickItemNavigationDr
awer(navView, drawerLayoutMain, ivDrawerView,

```

```

this@AkunActivity)
    }
}

private fun setData(){
    binding.apply {
        etNama.text = sharedPreferences.getNama()
        etNomorHp.text =
sharedPreferences.getNomorHp()
        etUsername.text =
sharedPreferences.getUsername()
        etPassword.text =
sharedPreferences.getPassword()
    }
}

private fun button() {
    binding.btnUbahData.setOnClickListener {
        setDialogUpdateData()
    }
}

private fun setDialogUpdateData() {
    val view =
AlertDialogAkunBinding.inflate(layoutInflater)
    val alertDialog =
AlertDialog.Builder(this@AkunActivity)
    alertDialog.setView(view.root)
        .setCancelable(false)
    val dialogInputan = alertDialog.create()
    dialogInputan.show()

    view.apply {

etEditNama.setText(sharedPreferences.getNama())

etEditNomorHp.setText(sharedPreferences.getNomo
rHp())

etEditUsername.setText(sharedPreferences.getUsen
ame())

etEditPassword.setText(sharedPreferences.getPassw
ord())

        btnSimpan.setOnClickListener {
            var cek = false
            if(etEditNama.toString().isEmpty()){
                etEditNama.error = "Tidak Boleh
Kosong"
                cek = true
            }
            if(etEditNomorHp.toString().isEmpty()){
                etEditNomorHp.error = "Tidak Boleh
Kosong"
                cek = true
            }
            if(etEditUsername.toString().isEmpty()){
                etEditUsername.error = "Tidak Boleh
Kosong"
                cek = true
            }
            if(etEditPassword.toString().isEmpty()){
                etEditPassword.error = "Tidak Boleh
Kosong"

```

```

cek = true
        }

        if(!cek){
            tempDialog = dialogInputan
            tempUser = UsersModel(
sharedPreferences.getIdUser().toString(),
            etEditNama.text.toString(),
            etEditNomorHp.text.toString(),
            etEditUsername.text.toString(),
            etEditPassword.text.toString(),
            sharedPreferences.getUsername()
        )
            postUpdateData(
sharedPreferences.getIdUser().toString(),
            etEditNama.text.toString(),
            etEditNomorHp.text.toString(),
            etEditUsername.text.toString(),
            etEditPassword.text.toString(),
            sharedPreferences.getUsername()
        )
        }

        btnBatal.setOnClickListener {
            tempDialog = null
            dialogInputan.dismiss()
        }
    }
}

private fun postUpdateData(idUser: String, nama:
String, nomorHp: String, username: String,
password: String, usernameLama: String) {
    Log.d("AkunActivityTAG", "postUpdateData:
$idUser, $nama, $nomorHp, $username, $password,
$usernameLama")
    viewModel.postUpdateUser(idUser, nama,
nomorHp, username, password, usernameLama)
}

private fun getPostUpdateData(){
viewModel.getUpdateData().observe(this@AkunAct
ivity){result->
    when(result){
        is UIState.Loading->
loading.alertDialogLoading(this@AkunActivity)
        is UIState.Success->
setSuccessUpdateData(result.data)
        is UIState.Failure->
setFailureUpdateData(result.message)
    }
}

private fun setFailureUpdateData(message:
String) {
    Toast.makeText(this@AkunActivity, message,
Toast.LENGTH_SHORT).show()
    loading.alertDialogCancel()
}

```

```

        private fun setSuccessUpdateData(data:
        ArrayList<ResponseModel>) {
            if(data.isNotEmpty()){
                if(data[0].status == "0"){
                    Toast.makeText(this@AkunActivity,
                    "Berhasil Update Akun",
                    Toast.LENGTH_SHORT).show()
                    sharedPreferences.setLogin(
                        tempUser.idUser!!.trim().toInt(),
                        tempUser.nama!!,
                        tempUser.nomorHp!!,
                        tempUser.username!!,
                        tempUser.password!!,
                        "user"
                    )
                    tempUser = UsersModel()
                    tempDialog!!.dismiss()
                    tempDialog = null
                    setData()
                } else{
                    Toast.makeText(this@AkunActivity,
                    data[0].message_response,
                    Toast.LENGTH_SHORT).show()
                }
            } else{
                Toast.makeText(this@AkunActivity, "Gagal:
                Ada kesalahan di API",
                Toast.LENGTH_SHORT).show()
            }
            loading.alertDialogCancel()
        }
    }
}

```

Akunviewmodel

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.akun

```

```

import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import
com.example.edukasipengobatanpadapenderitastrok
e.data.database.api.ApiService
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.delay
import kotlinx.coroutines.launch
import javax.inject.Inject

```

```

@HiltViewModel
class AkunViewModel @Inject constructor(
    private val api: ApiService
): ViewModel() {
    var _responsePostUpdateUser =
    MutableLiveData<UISState<ArrayList<ResponseMo
    del>>>()

```

```

        fun postUpdateUser(
            idUser:String, nama: String, nomorHp: String,
            username: String, password: String,
            usernameLama:String
        ){
            viewModelScope.launch(Dispatchers.IO){
                _responsePostUpdateUser.postValue(UISState.Loading)
                delay(1_000)
                try {
                    val data = api.postUpdateUser("", idUser,
                    nama, nomorHp, username, password,
                    usernameLama)
                } catch (ex: Exception){
                    _responsePostUpdateUser.postValue(UISState.Success(data))
                }
            }
        }
    }
}

```

```

        fun getUpdateData():
        LiveData<UISState<ArrayList<ResponseModel>>>
        = _responsePostUpdateUser
    }
}

```

Galeriherbaldetailactivity

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.galeri_herbal.detail

```

```

import android.content.ActivityNotFoundException
import android.content.Intent
import android.net.Uri
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalListModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityGaleriHerbalDetailBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Youtube

```

```

class GaleriHerbalDetailActivity :
    AppCompatActivity() {
        private lateinit var binding:
        ActivityGaleriHerbalDetailBinding
        private var listGaleriHerbalDetail :
        GaleriHerbalListModel? = null
        private var youtube = Youtube()
    }
}

```

```

        override fun onCreate(savedInstanceState:
Bundle?) {
            super.onCreate(savedInstanceState)
            binding =
ActivityGaleriHerbalDetailBinding.inflate(layoutInfl
ater)
            setContentView(binding.root)

            fetchDataSebelumnya()
            setButton()
            setData()

            Log.d("GaleriHerbalDetailTAG", "onCreate:
${listGaleriHerbalDetail!!.id_val_galeri_herbal}")
        }

        private fun fetchDataSebelumnya() {
            val extras = intent.extras
            if(extras != null) {
                listGaleriHerbalDetail =
intent.getParcelableExtra("galeri_herbal")
            }
        }

        private fun setButton() {
            binding.apply {
                btnBack.setOnClickListener {
                    finish()
                }
                btnVideoYoutube.setOnClickListener {

youtube.setToYoutubeVideo(this@GaleriHerbalDet
ailActivity, listGaleriHerbalDetail!!.youtube!!)
                }
            }

            private fun setData() {
                binding.apply {
                    titleHeader.text =
listGaleriHerbalDetail!!.nama
                    tvNamaGaleriHerbalDetail.text =
listGaleriHerbalDetail!!.nama
                    tvDeskripsiGaleriHerbalDetail.text =
listGaleriHerbalDetail!!.deskripsi
                    tvTataCaraPengolahan.text =
listGaleriHerbalDetail!!.tata_cara_pengolahan
                    Glide.with(this@GaleriHerbalDetailActivity)

.load("${Constant.BASE_URL}${Constant.LOCAT
ION_GAMBAR}${listGaleriHerbalDetail!!.gambar
}")

                    .error(R.drawable.gambar_error_image)
                    .into(binding.ivGaleriHerbalDetail)

youtube.setImageFromYoutube(this@GaleriHerbal
DetailActivity, listGaleriHerbalDetail!!.youtube!!,
binding.ivYoutube)
                }
            }
        }
    }
}
Galeriherballistactivity

```

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.galeri_herbal.list

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import
androidx.recyclerview.widget.GridLayoutManager
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.user.GaleriHerbalListAdapter
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalListModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityGaleriHerbalListBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.galeri_herbal.detail.GaleriHerbalD
etailActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.main.MainActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.AndroidEntryPoint

@AndroidEntryPoint
class GalerihHerbalListActivity :
AppCompatActivity() {
    private lateinit var binding:
ActivityGaleriHerbalListBinding
    private val viewModel:
GaleriHerbalListViewModel by viewModels()
    private lateinit var adapter:
GaleriHerbalListAdapter
    private var judulGaleriHerbal: String? = null
    private var id_hal_galeri_herbal: String? = null

    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityGaleriHerbalListBinding.inflate(layoutInflat
er)
        setContentView(binding.root)

        setBack()
        fetchDataSebelumnya()
        fetchGaleriHerbal()
        getGaleriHerbal()
    }

    private fun setBack() {
        binding.btnBack.setOnClickListener {
            finish()
        }
    }
}

```

```

    }
}

private fun fetchDataSebelumnya() {
    val extras = intent.extras
    if(extras != null) {
        judulGaleriHerbal = extras.getString("judul")
        id_hal_galeri_herbal =
extras.getString("id_hal_galeri_herbal")

        binding.titleHeader.text = judulGaleriHerbal
    }
}

private fun fetchGaleriHerbal() {

viewModel.fetchGaleriHerbalList(id_hal_galeri_her
bal!!)
}

private fun getGaleriHerbal(){

viewModel.getGaleriHerbalList().observe(this@Gal
eriHerbalListActivity){result->
    when(result){
        is UIState.Loading-> setStartShimmer()
        is UIState.Failure->
setFailureGaleriHerbal(result.message)
        is UIState.Success->
setSuccessGaleriHerbal(result.data)
    }
}

private fun setFailureGaleriHerbal(message:
String) {
    setStopShimmer()

Toast.makeText(this@GaleriHerbalListActivity,
message, Toast.LENGTH_SHORT).show()
}

private fun setSuccessGaleriHerbal(data:
ArrayList<GaleriHerbalListModel>) {
    setStopShimmer()
    if(data.isEmpty()){
        setAdapter(data)
    } else{

Toast.makeText(this@GaleriHerbalListActivity,
"Data Kosong", Toast.LENGTH_SHORT).show()
    }

private fun setAdapter(data:
ArrayList<GaleriHerbalListModel>) {
    adapter = GalerHerbalListAdapter(data, object
: OnClickItem.ClickGaleriHerbalList{
        override fun clickItem(galeriHerbalList:
GaleriHerbalListModel, it: View) {
            var i =
Intent(this@GaleriHerbalListActivity,
GaleriHerbalDetailActivity::class.java)
            i.putExtra("galeri_herbal",
galeriHerbalList)

```

```

startActivity(i)
    }
})

binding.apply {
    rvGaleriHerbalList.layoutManager =
GridLayoutManager(this@GaleriHerbalListActivity,
2)
    rvGaleriHerbalList.adapter = adapter
}

private fun setStopShimmer(){
    binding.apply {
        rvGaleriHerbalList.visibility =
View.VISIBLE

        smGaleriHerbalList.stopShimmer()
        smGaleriHerbalList.visibility = View.GONE
    }
}

private fun setStartShimmer(){
    binding.apply {
        smGaleriHerbalList.startShimmer()
        smGaleriHerbalList.visibility =
View.VISIBLE

        rvGaleriHerbalList.visibility = View.GONE
    }
}

Galeriherballistviewmodel
package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.galeri_herbal.list

import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import
com.example.edukasipengobatanpadapenderitastrok
e.data.database.api.ApiService
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalListModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalMainModel
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UIState
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.delay
import kotlinx.coroutines.launch
import javax.inject.Inject

@HiltViewModel
class GalerHerbalListViewModel @Inject
constructor(
    private val api: ApiService
): ViewModel() {
    private val _galeriHerbalList =
MutableLiveData<UIState<ArrayList<GaleriHerbal
ListModel>>>()

```

```

        fun fetchGaleriHerbalList(id_hal_galeri_herbal:
String) {
            viewModelScope.launch {

                _galeriHerbalList.postValue(UIState.Loading)
                delay(1_000)
                try {
                    val data = api.getGaleriHerbalList("",
id_hal_galeri_herbal)

                    _galeriHerbalList.postValue(UIState.Success(data))
                } catch (ex: Exception) {

                    _galeriHerbalList.postValue(UIState.Failure("Error
${ex.message}"))
                }
            }
        }

        fun getGaleriHerbalList():
LiveData<UIState<ArrayList<GaleriHerbalListMod
el>>> = _galeriHerbalList
    }
}

Galeriherbalmainactivity
package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.galeri_herbal.main

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import
androidx.recyclerview.widget.LinearLayoutManager
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.user.GaleriHerbalMainAdapter
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalMainModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityGaleriHerbalMainBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.galeri_herbal.list.GaleriHerbalList
Activity
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.main.MainActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KontrolNavigationDrawer
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint

@AndroidEntryPoint
class GaleriHerbalMainActivity :

```

```

AppCompatActivity() {
    private lateinit var binding:
ActivityGaleriHerbalMainBinding
    private lateinit var kontrolNavigationDrawer:
KontrolNavigationDrawer
    private val viewModel:
GaleriHerbalMainViewModel by viewModels()
    private lateinit var adapter:
GaleriHerbalMainAdapter
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityGaleriHerbalMainBinding.inflate(layoutInfl
ater)
        setContentView(binding.root)

        setKontrolNavigationDrawer()
        fetchGaleriHerbal()
        getGaleriHerbal()
    }

    private fun setKontrolNavigationDrawer() {
        kontrolNavigationDrawer =
KontrolNavigationDrawer(this@GaleriHerbalMain
Activity)
        binding.apply {

            kontrolNavigationDrawer.cekSebagai(navView)

            kontrolNavigationDrawer.onClickItemNavigationDr
awer(navView, drawerLayoutMain, ivDrawerView,
this@GaleriHerbalMainActivity)
        }
    }

    private fun fetchGaleriHerbal() {
        viewModel.fetchGaleriHerbal()
    }

    private fun getGaleriHerbal(){
        viewModel.getGaleriHerbal().observe(this@GaleriH
erbalMainActivity){result->
            when(result){
                is UIState.Loading-> setStartShimmer()
                is UIState.Failure->
setFailureGaleriHerbal(result.message)
                is UIState.Success->
setSuccessGaleriHerbal(result.data)
            }
        }
    }

    private fun setFailureGaleriHerbal(message:
String) {
        setStopShimmer()

        Toast.makeText(this@GaleriHerbalMainActivity,
message, Toast.LENGTH_SHORT).show()
    }

    private fun setSuccessGaleriHerbal(data:
ArrayList<GaleriHerbalMainModel>) {
        setStopShimmer()
    }
}

```

```

        if(data.isEmpty()){
            setAdapter(data)
        } else{

Toast.makeText(this@GaleriHerbalMainActivity,
"Data Kosong", Toast.LENGTH_SHORT).show()
        }

        private fun setAdapter(data:
ArrayList<GaleriHerbalMainModel>) {
            adapter = GalerHerbalMainAdapter(data,
object : OnClickItem.ClickObatHerbal{
                override fun clickItem(id: String, judul:
String, it: View) {
                    val i =
Intent(this@GaleriHerbalMainActivity,
GalerHerbalListActivity::class.java)
                    i.putExtra("id_hal_galeri_herbal", id)
                    i.putExtra("judul", judul)
                    startActivity(i)
                }
            })

            binding.apply {
                rvGaleriHerbal.layoutManager =
LinearLayoutManager(this@GaleriHerbalMainActiv
ity, LinearLayoutManager.VERTICAL, false)
                rvGaleriHerbal.adapter = adapter
            }

            private fun setStopShimmer(){
                binding.apply {
                    rvGaleriHerbal.visibility = View.VISIBLE

                    smGaleriHerbal.stopShimmer()
                    smGaleriHerbal.visibility = View.GONE
                }
            }
            private fun setStartShimmer(){
                binding.apply {
                    smGaleriHerbal.startShimmer()
                    smGaleriHerbal.visibility = View.VISIBLE

                    rvGaleriHerbal.visibility = View.GONE
                }
            }

            override fun onBackPressed() {
                super.onBackPressed()

startActivity(Intent(this@GaleriHerbalMainActivity,
MainActivity::class.java))
                finish()
            }
        }
    }
}

galeriherbalmainviewmodel
package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.galeri_herbal.main

import androidx.lifecycle.LiveData

```

```

import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import
com.example.edukasipengobatanpadapenderitastrok
e.data.database.api.ApiService
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.GaleriHerbalMainModel
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.delay
import kotlinx.coroutines.launch
import javax.inject.Inject

```

```

@HiltViewModel
class GalerHerbalMainViewModel @Inject
constructor(
    private val api: ApiService
): ViewModel() {
    private val _galeriHerbalMain =
MutableLiveData<UISState<ArrayList<GaleriHerbal
MainModel>>>()

```

```

        fun fetchGaleriHerbal() {
            viewModelScope.launch {

_galeriHerbalMain.postValue(UISState.Loading)
                delay(1_000)
                try {
                    val data = api.getGaleriHerbalMain("")

_galeriHerbalMain.postValue(UISState.Success(data)
                )
            } catch (ex: Exception) {

_galeriHerbalMain.postValue(UISState.Failure("Error
${ex.message}"))
            }
        }
    }
}

```

```

        fun getGaleriHerbal():
LiveData<UISState<ArrayList<GaleriHerbalMainMo
del>>> = _galeriHerbalMain
    }
}

```

Maniactivity

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.main

```

```

import android.annotation.SuppressLint
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import
androidx.recyclerview.widget.LinearLayoutManager
import com.bumptech.glide.Glide

```

```

import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.user.TestimoniAdapter
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TestimoniModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityMainBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogShowImageBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.login.LoginActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.galeri_herbal.main.GaleriHerbalMa
inActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.menu_sehat.MenuSehatActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.tentang_stroke.list.TentangStrokeL
istActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.terapi.list.TerapiListActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.testimoni.TestimoniActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KontrolNavigationDrawer
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.SharedPreferencesLogin
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISate
import dagger.hilt.android.AndroidEntryPoint
import kotlin.collections.ArrayList

@AndroidEntryPoint
class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private lateinit var kontrolNavigationDrawer:
KontrolNavigationDrawer
    private val viewModel: MainViewModel by
viewModels()
    private lateinit var adapter: TestimoniAdapter
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        binding =

```

```

ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

//    fragmentHome()
//    setNavigationDrawerFragment()

        setNavigationDrawer()
        setButton()
        fetchTestimoni()
        getTestimoni()
    }

//    private fun setNavigationDrawerFragment() {
//        kontrolNavigationDrawerFragment=
KontrolNavigationDrawerFragment(this@MainActi
vity)
//        binding.apply {
//
//            kontrolNavigationDrawer.cekSebagai(navView)
//
//            kontrolNavigationDrawer.onClickItemNavigationDr
awer(navView, drawerLayoutMain, ivDrawerView,
this@MainActivity)
//        }
//    }

//    private fun fragmentHome() {
//        replaceFragment(GaleriHerbalFragment())
//    }

//    private fun replaceFragment(fragment:
Fragment){
//        var fragmentManager : FragmentManager =
supportFragmentManager
//        var fragmentTransaction =
fragmentManager.beginTransaction()
//        fragmentTransaction.replace(R.id.flMain,
fragment)
//        fragmentTransaction.commit()
//    }

    private fun setNavigationDrawer() {
        kontrolNavigationDrawer =
KontrolNavigationDrawer(this@MainActivity)
        binding.apply {

            kontrolNavigationDrawer.cekSebagai(navView)

            kontrolNavigationDrawer.onClickItemNavigationDr
awer(navView, drawerLayoutMain, ivDrawerView,
this@MainActivity)
        }
    }

    private fun setButton() {
        binding.apply {
            btnLogin.setOnClickListener {
                startActivity(Intent(this@MainActivity,
LoginActivity::class.java))
                finish()
            }

            btnTentangStroke.setOnClickListener {
                startActivity(Intent(this@MainActivity,

```

```

TentangStrokeListActivity::class.java))
        finish()
    }
    btnGaleriHerbal.setOnClickListener {
        startActivity(Intent(this@MainActivity,
GaleriHerbalMainActivity::class.java))
        finish()
    }
    btnTerapi.setOnClickListener {
        startActivity(Intent(this@MainActivity,
TerapiListActivity::class.java))
        finish()
    }
    btnMenuSehat.setOnClickListener {
        startActivity(Intent(this@MainActivity,
MenuSehatActivity::class.java))
        finish()
    }

    btnTestimoni.setOnClickListener {
        startActivity(Intent(this@MainActivity,
TestimoniActivity::class.java))
        finish()
    }
}

private fun fetchTestimoni() {
    viewModel.fetchTestimoni()
}

private fun getTestimoni() {
    viewModel.getTestimoni().observe(this@MainActiv
ity){result->
        when(result){
            is UIState.Loading-> setStartShimmer()
            is UIState.Failure->
setFailureTestimoni(result.message)
            is UIState.Success->
setSuccessTestimoni(result.data)
        }
    }
}

private fun setFailureTestimoni(message: String)
{
    setStopShimmer()
    Toast.makeText(this@MainActivity, message,
Toast.LENGTH_SHORT).show()
}

private fun setSuccessTestimoni(data:
ArrayList<TestimoniModel>) {
    val testimoniSendiri:
ArrayList<TestimoniModel> = arrayListOf()
    val testimoniOrangLain:
ArrayList<TestimoniModel> = arrayListOf()
    val testimoniSemua:
ArrayList<TestimoniModel> = arrayListOf()
    for (value in data){
        testimoniOrangLain.add(value)
    }
    testimoniSemua.addAll(testimoniSendiri)
    testimoniSemua.addAll(testimoniOrangLain)
}

```

```

setAdapter(testimoniSemua)

//    setAdapter(data)
//    }

private fun setAdapter(data:
ArrayList<TestimoniModel>) {
    adapter = TestimoniAdapter(data, true, object :
OnClickItem.ClickTestimoni{
        override fun clickGambar(gambar: String,
nama: String, it: View) {

//            val view =
AlertDialogShowImageBinding.inflate(layoutInflate
r)
//            val alertDialog =
AlertDialog.Builder(this@MainActivity)
//            alertDialog.setView(view.root)
//            .setCancelable(false)
//            val dialogInputan = alertDialog.create()
//            dialogInputan.show()
//            view.apply {
//                tvTitle.text = nama
//                btnClose.setOnClickListener {
//                    dialogInputan.dismiss()
//                }
//            }
//            Glide.with(this@MainActivity)
//            .load("${Constant.BASE_URL}${Constant.LOCAT
ION_GAMBAR}$gambar") // URL Gambar
//            .error(R.drawable.gambar_error_image)
//            .into(view.ivShowImage) //
imageView mana yang akan diterapkan

        }

    })
    binding.apply {
        rvTestimoni.layoutManager =
LinearLayoutManager(this@MainActivity,
LinearLayoutManager.VERTICAL, false)
        rvTestimoni.adapter = adapter
    }
    setStopShimmer()
}

private fun setStopShimmer(){
    binding.apply {
        rvTestimoni.visibility = View.VISIBLE

        smTestimoni.stopShimmer()
        smTestimoni.visibility = View.GONE
    }
}
private fun setStartShimmer(){
    binding.apply {
        smTestimoni.startShimmer()
        smTestimoni.visibility = View.VISIBLE

        rvTestimoni.visibility = View.GONE
    }
}

```

```

    }
}

private var tapDuaKali = false
override fun onBackPressed() {
    if (tapDuaKali){
        super.onBackPressed()
    }
    tapDuaKali = true
    Toast.makeText(this@MainActivity, "Tekan
    Sekali Lagi untuk keluar",
    Toast.LENGTH_SHORT).show()

    Handler().postDelayed({
        tapDuaKali = false
    }, 2000)
}

menusehatmainactivty
package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.menu_sehat

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import
androidx.recyclerview.widget.LinearLayoutManager
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.user.MenuSehatAdapter
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.MenuSehatModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityMenuSehatBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.main.MainActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KontrolNavigationDrawer
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.AndroidEntryPoint

@AndroidEntryPoint
class MenuSehatActivity : AppCompatActivity() {
    private lateinit var binding:
    ActivityMenuSehatBinding
    private lateinit var kontrolNavigationDrawer:
    KontrolNavigationDrawer
    private val viewModel: MenuSehatViewModel by
    viewModels()
    private lateinit var adapter: MenuSehatAdapter
    override fun onCreate(savedInstanceState:
    Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
        ActivityMenuSehatBinding.inflate(layoutInflater)
        setContentView(binding.root)

```

```

        setKontrolNavigationDrawer()
        fetchTentangStroke()
        getTentangStroke()
    }

    private fun setKontrolNavigationDrawer() {
        kontrolNavigationDrawer =
        KontrolNavigationDrawer(this@MenuSehatActivity
        )
        binding.apply {

            kontrolNavigationDrawer.cekSebagai(navView)

            kontrolNavigationDrawer.onClickItemNavigationDr
            awer(navView, drawerLayoutMain, ivDrawerView,
            this@MenuSehatActivity)
        }
    }

    private fun fetchTentangStroke() {
        viewModel.fetchMenuSehat()
    }

    private fun getTentangStroke(){
        viewModel.getMenuSehat().observe(this@MenuSeh
        atActivity){result->
            when(result){
                is UIState.Loading-> setStartShimmer()
                is UIState.Failure->
                setFailureTentangStroke(result.message)
                is UIState.Success->
                setSuccessTentangStroke(result.data)
            }
        }
    }

    private fun setFailureTentangStroke(message:
    String) {
        setStopShimmer()
        Toast.makeText(this@MenuSehatActivity,
        message, Toast.LENGTH_SHORT).show()
    }

    private fun setSuccessTentangStroke(data:
    ArrayList<MenuSehatModel>) {
        setStopShimmer()
        if(data.isNotEmpty()){
            setAdapter(data)
        } else{
            Toast.makeText(this@MenuSehatActivity,
            "Data Kosong", Toast.LENGTH_SHORT).show()
        }
    }

    private fun setAdapter(data:
    ArrayList<MenuSehatModel>) {
        adapter = MenuSehatAdapter(data)

        binding.apply {
            tvMenuSehat.layoutManager =
            LinearLayoutManager(this@MenuSehatActivity,
            LinearLayoutManager.VERTICAL, false)
            tvMenuSehat.adapter = adapter

```

```

    }
}

private fun setStopShimmer(){
    binding.apply {
        tvMenuSehat.visibility = View.VISIBLE

        smMenuSehat.stopShimmer()
        smMenuSehat.visibility = View.GONE
    }
}

private fun setStartShimmer(){
    binding.apply {
        smMenuSehat.startShimmer()
        smMenuSehat.visibility = View.VISIBLE

        tvMenuSehat.visibility = View.GONE
    }
}

override fun onBackPressed() {
    super.onBackPressed()
    startActivity(Intent(this@MenuSehatActivity,
MainActivity::class.java))
    finish()
}
}

```

Testimoni

```

package
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.testimoni

import android.Manifest
import android.annotation.SuppressLint
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Environment
import android.provider.OpenableColumns
import android.provider.Settings
import android.view.View
import android.widget.ImageView
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.recyclerview.widget.LinearLayoutManager
import import com.bumptech.glide.Glide
import
com.example.edukasipengobatanpadapenderitastrok
e.R
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.admin.AdminHalamanGaleriHerbalAdapte
r
import
com.example.edukasipengobatanpadapenderitastrok
e.adapter.user.TestimoniAdapter

```

```

import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.ResponseModel
import
com.example.edukasipengobatanpadapenderitastrok
e.data.model.TestimoniModel
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.ActivityTestimoniBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogKonfirmasiBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.databinding.AlertDialogShowImageBinding
import
com.example.edukasipengobatanpadapenderitastrok
e.ui.activity.user.main.MainActivity
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.Constant
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KataAcak
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.KontrolNavigationDrawer
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.LoadingAlertDialog
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.OnClickItem
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.SharedPreferencesLogin
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.TanggalDanWaktu
import
com.example.edukasipengobatanpadapenderitastrok
e.utils.network.UISState
import dagger.hilt.android.AndroidEntryPoint
import
okhttp3.MediaType.Companion.toMediaTypeOrNull
import okhttp3.MultipartBody
import okhttp3.RequestBody
import
okhttp3.RequestBody.Companion.toRequestBody
import javax.inject.Inject
import kotlin.collections.ArrayList

```

```

@AndroidEntryPoint
class TestimoniActivity : AppCompatActivity() {
    private lateinit var binding:
ActivityTestimoniBinding
    private lateinit var kontrolNavigationDrawer:
KontrolNavigationDrawer
    private val viewModel: TestimoniViewModel by
viewModels()
    private lateinit var adapter: TestimoniAdapter
    private var testimoniSendiri:
ArrayList<TestimoniModel> = arrayListOf()
    private var testimoniOrangLain:

```

```

ArrayList<TestimoniModel> = arrayListOf()
@Inject lateinit var tanggalDanWaktu:
TanggalDanWaktu
@Inject lateinit var loading: LoadingAlertDialog
private var tempAlertDialog: AlertDialog? = null
private var fileImage: MultipartBody.Part? = null
private var kataAcak = KataAcak()

override fun onCreate(savedInstanceState:
Bundle?) {
    super.onCreate(savedInstanceState)
    binding =
    ActivityTestimoniBinding.inflate(layoutInflater)
    setContentView(binding.root)

    setNavigationDrawer()
    fetchTestimoni()
    getTestimoni()
    setButton()
    getTambahTestimoni()
    // getHapusTestimoni()
}

// Tambah testimoni
private fun setButton() {
    binding.apply {
        btnTambahTestimoni.setOnClickListener {
            openAddTestimoni()
        }
    }
}

// buka input testimoni
private fun openAddTestimoni(){
    binding.apply {
        llPostTestimoniSendiri.visibility =
View.VISIBLE
        btnTambahTestimoni.visibility =
View.GONE

        var jumlahBintang = 0
        ivPostBintang1.setOnClickListener {
            setBintang1(ivPostBintang1,
            ivPostBintang2, ivPostBintang3, ivPostBintang4,
            ivPostBintang5)
            jumlahBintang = 1
        }
        ivPostBintang2.setOnClickListener {
            setBintang2(ivPostBintang1,
            ivPostBintang2, ivPostBintang3, ivPostBintang4,
            ivPostBintang5)
            jumlahBintang = 2
        }
        ivPostBintang3.setOnClickListener {
            setBintang3(ivPostBintang1,
            ivPostBintang2, ivPostBintang3, ivPostBintang4,
            ivPostBintang5)
            jumlahBintang = 3
        }
        ivPostBintang4.setOnClickListener {
            setBintang4(ivPostBintang1,
            ivPostBintang2, ivPostBintang3, ivPostBintang4,
            ivPostBintang5)
            jumlahBintang = 4
        }
    }
}

```

```

        ivPostBintang5.setOnClickListener {
            setBintang5(ivPostBintang1,
            ivPostBintang2, ivPostBintang3, ivPostBintang4,
            ivPostBintang5)
            jumlahBintang = 5
        }

        // foto
        etImageBukti.setOnClickListener {
            if(checkPermission()){
                pickImageFile()
            } else{
                requestPermission()
            }
        }

        // kirim testimoni
        btnKirimTestimoni.setOnClickListener {
            if(jumlahBintang==0){

                Toast.makeText(this@TestimoniActivity,
                "Masukkan Jumlah Bintang",
                Toast.LENGTH_SHORT).show()
            } else{
                var check = true
                if(etNama.text.isEmpty()){
                    etNama.error = "Masukkan Nama"
                    check = false
                }

                if(etEmail.text.isEmpty()){
                    etEmail.error = "Masukkan Email"
                    check = false
                }

                if(etTestimoni.text.isEmpty()){
                    etTestimoni.error = "Masukkan
                Testimoni"
                    check = false
                }

                if(tvTextImageBukti.text.toString() ==
                "Masukkan Gambar"){
                    tvTextImageBukti.error =
                "Masukkan Gambar"
                    check = false
                }

                if(check){
                    val nama = etNama.text.toString()
                    val email = etEmail.text.toString()
                    val testimoni =
                    etTestimoni.text.toString()
                    val kata =
                    kataAcak.getHurufDanAngka()

                    postTambahTestimoni(
                        kata, nama, email, testimoni,
                        jumlahBintang.toString(), fileImage!!
                    )
                }
            }
        }
    }
}

```

```

        // batal testimoni
        btnBatalTambah.setOnClickListener {
            btnTambahTestimoni.visibility =
                View.VISIBLE
            llPostTestimoniSendiri.visibility =
                View.GONE
        }
    }

    private fun setNavigationDrawer() {
        kontrolNavigationDrawer =
            KontrolNavigationDrawer(this@TestimoniActivity)
        binding.apply {

            kontrolNavigationDrawer.cekSebagai(navView)

            kontrolNavigationDrawer.onClickItemNavigationDrawer(
                navView, drawerLayoutMain, ivDrawerView,
                this@TestimoniActivity
            ) {
            }

            private fun fetchTestimoni() {
                viewModel.fetchTestimoni()
            }

            private fun getTestimoni() {
                viewModel.getTestimoni().observe(this@TestimoniActivity) { result ->
                    when(result) {
                        is UIState.Loading -> setStartShimmer()
                        is UIState.Failure ->
                            setFailureTestimoni(result.message)
                        is UIState.Success ->
                            setSuccessTestimoni(result.data)
                    }
                }
            }

            private fun setFailureTestimoni(message: String) {
                {
                    setStopShimmer()
                    Toast.makeText(this@TestimoniActivity,
                        message, Toast.LENGTH_SHORT).show()
                }

                private fun setSuccessTestimoni(data:
                    ArrayList<TestimoniModel>) {
                    testimoniSendiri = arrayListOf()
                    testimoniOrangLain = arrayListOf()
                    for (value in data) {
                        testimoniOrangLain.add(value)
                    }
                    setNoHaveDataTestimoni()
                }

                // setAdapter(data)
                setAdapter(testimoniOrangLain)
            }

            private fun setNoHaveDataTestimoni() {
                binding.apply {
                    btnTambahTestimoni.visibility =

```

```

                View.VISIBLE
                llPostTestimoniSendiri.visibility =
                    View.GONE
                // llTestimoniSendiri.visibility = View.GONE
            }
        }

        private fun setHiddenDataTestimoni() {
            binding.apply {
                btnTambahTestimoni.visibility =
                    View.GONE
                llPostTestimoniSendiri.visibility =
                    View.GONE
            }
        }

        private fun postTambahTestimoni(kata:String,
            nama:String, email:String, testimoni:String,
            bintang:String, file: MultipartBody.Part) {
            // viewModel.postTambahData(nama, testimoni,
            bintang)

            viewModel.postTambahData(
                convertStringToMultipartBody("post_tambah_testimoni"),
                convertStringToMultipartBody(kata),
                convertStringToMultipartBody("1"),
                convertStringToMultipartBody(nama),
                convertStringToMultipartBody(email),
                convertStringToMultipartBody(testimoni),
                convertStringToMultipartBody(bintang),
                file
            )
        }

        private fun getTambahTestimoni() {
            viewModel.getResponseTambahTestimoni().observe(
                this@TestimoniActivity) { result ->
                    when(result) {
                        is UIState.Loading ->
                            loading.alertDialogLoading(this@TestimoniActivity)
                        is UIState.Failure ->
                            setFailureTambahTestimoni(result.message)
                        is UIState.Success ->
                            setSuccessTambahTestimoni(result.data)
                    }
                }
            }

            private fun setFailureTambahTestimoni(message:
                String) {
                loading.alertDialogCancel()
                Toast.makeText(this@TestimoniActivity,
                    message, Toast.LENGTH_SHORT).show()
            }

            private fun setSuccessTambahTestimoni(data:
                java.util.ArrayList<ResponseModel>) {
                loading.alertDialogCancel()
                if(data.isNotEmpty()){

```

```

        if(data[0].status == "0"){
            Toast.makeText(this@TestimoniActivity,
"Berhasil Tambah",
Toast.LENGTH_SHORT).show()
            setHiddenDataTestimoni()
            fetchTestimoni()
        } else{
            Toast.makeText(this@TestimoniActivity,
data[0].message_response,
Toast.LENGTH_SHORT).show()
        }
    }
}

private fun openEditTestimoni(testimoniSendiri:
ArrayList<TestimoniModel>) {
    binding.apply {
        llPostTestimoniSendiri.visibility =
View.VISIBLE
        btnTambahTestimoni.visibility =
View.GONE

etTestimoni.setText(testimoniSendiri[0].testimoni)
        when
(testimoniSendiri[0].bintang!!.trim().toInt()) {
            1 -> {
                setBintang1(ivPostBintang1,
ivPostBintang2, ivPostBintang3, ivPostBintang4,
ivPostBintang5)
            }
            2 -> {
                setBintang2(ivPostBintang1,
ivPostBintang2, ivPostBintang3, ivPostBintang4,
ivPostBintang5)
            }
            3 -> {
                setBintang3(ivPostBintang1,
ivPostBintang2, ivPostBintang3, ivPostBintang4,
ivPostBintang5)
            }
            4 -> {
                setBintang4(ivPostBintang1,
ivPostBintang2, ivPostBintang3, ivPostBintang4,
ivPostBintang5)
            }
            5 -> {
                setBintang5(ivPostBintang1,
ivPostBintang2, ivPostBintang3, ivPostBintang4,
ivPostBintang5)
            }
        }

        var jumlahBintang =
testimoniSendiri[0].bintang!!.trim().toInt()
        ivPostBintang1.setOnClickListener {
            setBintang1(ivPostBintang1,
ivPostBintang2, ivPostBintang3, ivPostBintang4,
ivPostBintang5)
            jumlahBintang = 1
        }
        ivPostBintang2.setOnClickListener {
            setBintang2(ivPostBintang1,
ivPostBintang2, ivPostBintang3, ivPostBintang4,
ivPostBintang5)

```

```

            jumlahBintang = 2
        }
        ivPostBintang3.setOnClickListener {
            setBintang3(ivPostBintang1,
ivPostBintang2, ivPostBintang3, ivPostBintang4,
ivPostBintang5)
            jumlahBintang = 3
        }
        ivPostBintang4.setOnClickListener {
            setBintang4(ivPostBintang1,
ivPostBintang2, ivPostBintang3, ivPostBintang4,
ivPostBintang5)
            jumlahBintang = 4
        }
        ivPostBintang5.setOnClickListener {
            setBintang5(ivPostBintang1,
ivPostBintang2, ivPostBintang3, ivPostBintang4,
ivPostBintang5)
            jumlahBintang = 5
        }

etImageBukti.setOnClickListener {
    if(checkPermission()){
        pickImageFile()
    } else{
        requestPermission()
    }
}

btnKirimTestimoni.setOnClickListener {
    if(jumlahBintang==0){

Toast.makeText(this@TestimoniActivity,
"Masukkan Jumlah Bintang",
Toast.LENGTH_SHORT).show()
        } else{
            val idTestimoni =
testimoniSendiri[0].id_testimoni!!
            // val idUser =
sharedPreferencesLogin.getIdUser().toString()
            val testimoni =
etTestimoni.text.toString()
            val kata =
kataAcak.getHurufDanAngka()

            if(fileImage == null){
                //
                postEditTestimoniNoHaveImage(idTestimoni,
idUser, testimoni, jumlahBintang.toString())
            } else{
                //
                postEditTestimoni(kata,
idTestimoni, idUser, testimoni,
jumlahBintang.toString(), fileImage!!)
            }
        }
    }

    btnBatalTambah.setOnClickListener {
        llPostTestimoniSendiri.visibility =
View.GONE
        btnTambahTestimoni.visibility =
View.GONE

        tvTextImageBukti.text = ""
        fileImage = null
    }
}

```

```

    }
}
}
// private fun
postEditTestimoni(id_testimoni:String,
id_user:String, testimoni:String, bintang:String) {
//    viewModel.postUpdatehData(
//        id_testimoni, id_user, testimoni, bintang
//    )
// }

private fun setBintang0(
    iv1: ImageView,
    iv2: ImageView,
    iv3: ImageView,
    iv4: ImageView,
    iv5: ImageView,
){

iv1.setImageResource(R.drawable.icon_star_non_aktif)

iv2.setImageResource(R.drawable.icon_star_non_aktif)

iv3.setImageResource(R.drawable.icon_star_non_aktif)

iv4.setImageResource(R.drawable.icon_star_non_aktif)

iv5.setImageResource(R.drawable.icon_star_non_aktif)
}
private fun setBintang1(
    iv1: ImageView,
    iv2: ImageView,
    iv3: ImageView,
    iv4: ImageView,
    iv5: ImageView,
){

iv1.setImageResource(R.drawable.icon_star_aktif)

iv2.setImageResource(R.drawable.icon_star_non_aktif)

iv3.setImageResource(R.drawable.icon_star_non_aktif)

iv4.setImageResource(R.drawable.icon_star_non_aktif)

iv5.setImageResource(R.drawable.icon_star_non_aktif)
}
private fun setBintang2(
    iv1: ImageView,
    iv2: ImageView,
    iv3: ImageView,
    iv4: ImageView,
    iv5: ImageView,
){

iv1.setImageResource(R.drawable.icon_star_aktif)

```

```

iv2.setImageResource(R.drawable.icon_star_aktif)

iv3.setImageResource(R.drawable.icon_star_non_aktif)

iv4.setImageResource(R.drawable.icon_star_non_aktif)

iv5.setImageResource(R.drawable.icon_star_non_aktif)
}
private fun setBintang3(
    iv1: ImageView,
    iv2: ImageView,
    iv3: ImageView,
    iv4: ImageView,
    iv5: ImageView,
){

iv1.setImageResource(R.drawable.icon_star_aktif)

iv2.setImageResource(R.drawable.icon_star_aktif)

iv3.setImageResource(R.drawable.icon_star_aktif)

iv4.setImageResource(R.drawable.icon_star_non_aktif)

iv5.setImageResource(R.drawable.icon_star_non_aktif)
}
private fun setBintang4(
    iv1: ImageView,
    iv2: ImageView,
    iv3: ImageView,
    iv4: ImageView,
    iv5: ImageView,
){

iv1.setImageResource(R.drawable.icon_star_aktif)

iv2.setImageResource(R.drawable.icon_star_aktif)

iv3.setImageResource(R.drawable.icon_star_aktif)

iv4.setImageResource(R.drawable.icon_star_aktif)

iv5.setImageResource(R.drawable.icon_star_non_aktif)
}
private fun setBintang5(
    iv1: ImageView,
    iv2: ImageView,
    iv3: ImageView,
    iv4: ImageView,
    iv5: ImageView,
){

iv1.setImageResource(R.drawable.icon_star_aktif)

iv2.setImageResource(R.drawable.icon_star_aktif)

iv3.setImageResource(R.drawable.icon_star_aktif)

```

```

iv4.setImageResource(R.drawable.icon_star_aktif)

iv5.setImageResource(R.drawable.icon_star_aktif)
}

private fun setAdapter(data:
ArrayList<TestimoniModel>) {
    adapter = TestimoniAdapter(data,false, object:
OnClickItem.ClickTestimoni{
        override fun clickGambar(gambar: String,
nama: String, it: View) {
            setShowImage(gambar, nama)
        }
    })
    binding.apply {
        rvTestimoni.layoutManager =
LinearLayoutManager(this@TestimoniActivity,
LinearLayoutManager.VERTICAL, false)
        rvTestimoni.adapter = adapter
    }
    setStopShimmer()
}

private fun setShowImage(gambar: String, nama:
String){
    val view =
AlertDialogShowImageBinding.inflate(layoutInflate
r)

    val alertDialog =
AlertDialog.Builder(this@TestimoniActivity)
    alertDialog.setView(view.root)
        .setCancelable(false)
    val dialogInputan = alertDialog.create()
    dialogInputan.show()

    view.apply {
        tvTitle.text = nama
        btnClose.setOnClickListener {
            dialogInputan.dismiss()
        }
    }
}

Glide.with(this@TestimoniActivity)

.load("${Constant.BASE_URL}${Constant.LOCAT
ION_GAMBAR}$gambar") // URL Gambar
    .error(R.drawable.gambar_error_image)
    .into(view.ivShowImage) // imageView
mana yang akan diterapkan
}

private fun setStopShimmer(){
    binding.apply {
        rvTestimoni.visibility = View.VISIBLE

        smTestimoni.stopShimmer()
        smTestimoni.visibility = View.GONE
    }
}

private fun setStartShimmer(){
    binding.apply {
        smTestimoni.startShimmer()
        smTestimoni.visibility = View.VISIBLE

```

```

        rvTestimoni.visibility = View.GONE
    }
}

// permission add image
private fun requestPermission(){
    if(Build.VERSION.SDK_INT >=
Build.VERSION_CODES.R){
        if
(Environment.isExternalStorageManager()) {
            startActivity(Intent(this,
AdminHalamanGaleriHerbalAdapter::class.java))
        } else { //request for the permission
            val intent =
Intent(Settings.ACTION_MANAGE_APP_ALL_FIL
ES_ACCESS_PERMISSION)
            val uri = Uri.fromParts("package",
packageName, null)
            intent.data = uri
            startActivity(intent)
        }
    } else{
        ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.WRITE_EXTERNAL_
STORAGE,
Manifest.permission.READ_EXTERNAL_STORAGE
),
Constant.STORAGE_PERMISSION_CODE
        )
    }
}

private fun pickImageFile() {
    val intent =
Intent(Intent.ACTION_OPEN_DOCUMENT).apply
{
        type = "image/*"
    }

    startActivityForResult(intent,
Constant.IMAGE_CODE)
}

override fun onActivityResult(requestCode: Int,
resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode,
resultCode, data)

    if (requestCode == Constant.IMAGE_CODE
&& resultCode == Activity.RESULT_OK && data
!= null) {
        // Mendapatkan URI file PDF yang dipilih
        val fileUri = data.data!!

        val nameImage = getNameFile(fileUri)

        binding.tvTextImageBukti.text = nameImage

        // Mengirim file PDF ke website
        menggunakan Retrofit
        fileImage = uploadImageToStorage(fileUri,
nameImage, "gambar_testimoni")

```

```

    }
}

private fun getNameFile(uri: Uri): String {
    val cursor = contentResolver.query(uri, null,
    null, null, null)
    val nameIndex =
    cursor?.getColumnIndex(OpenableColumns.DISPLAY_NAME)
    cursor?.moveToFirst()
    val name = cursor?.getString(nameIndex!!)
    cursor?.close()
    return name!!
}

@SuppressLint("Recycle")
private fun uploadImageToStorage(imageUri:
Uri?, fileName: String, nameAPI:String):
MultipartBody.Part? {
    var pdfPart : MultipartBody.Part? = null
    imageUri?.let {
        val file =
        contentResolver.openInputStream(imageUri)?.readBytes()

        if (file != null) {
            // Membuat objek RequestBody dari file
            PDF
            val requestFile =
            file.toRequestBody("application/pdf".toMediaType
            OrNull())
            // Membuat objek MultipartBody.Part
            untuk file PDF
            pdfPart =
            MultipartBody.Part.createFormData("materi_pdf",
            fileName, requestFile)

            pdfPart =
            convertFileToMultipartBody(file, fileName,
            nameAPI)
        }
    }
    return pdfPart
}

private fun convertFileToMultipartBody(file:
ByteArray, fileName: String, nameAPI:String):
MultipartBody.Part?{
    // val requestFile =
    file.toRequestBody("application/pdf".toMediaType
    OrNull())
    val requestFile =
    file.toRequestBody("application/image".toMediaType
    OrNull())
    val filePart =
    MultipartBody.Part.createFormData(nameAPI,
    fileName, requestFile)

    return filePart
}

private fun convertStringToMultipartBody(data:
String): RequestBody {
    return RequestBody.create("multipart/form-
    data".toMediaTypeOrNull(), data)
}

```

```

}

private fun checkPermission(): Boolean{
    return if (Build.VERSION.SDK_INT >=
    Build.VERSION_CODES.R){
        //Android is 11(R) or above
        Environment.isExternalStorageManager()
    }
    else{
        //Android is below 11(R)
        val write =
        ContextCompat.checkSelfPermission(this,
        Manifest.permission.WRITE_EXTERNAL_STORAGE)
        val read =
        ContextCompat.checkSelfPermission(this,
        Manifest.permission.READ_EXTERNAL_STORAGE
        )
        write ==
        PackageManager.PERMISSION_GRANTED &&
        read == PackageManager.PERMISSION_GRANTED
    }
}

override fun onBackPressed() {
    super.onBackPressed()
    startActivity(Intent(this@TestimoniActivity,
    MainActivity::class.java))
    finish()
}
}

```