LAMPIRAN

Lampiran 1. Kode Program AktivitasFragment.kt

```
package com.abcd.pemantauankesehatananak.ui.fragment.user.aktivitas

import android.content.Intent
import android.os.Bundle
import android.text.Editable
import android.text.TextWatcher
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.fragment.app.viewModels
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.pemantauankesehatananak.adapter.AktivitasAdapter
import com.abcd.pemantauankesehatananak.adapter.KategoriAdapter
import com.abcd.pemantauankesehatananak.data.model.AktivitasModel
import com.abcd.pemantauankesehatananak.data.model.KategoriModel
import com.abcd.pemantauankesehatananak.databinding.FragmentAktivitasBinding
import
com.abcd.pemantauankesehatananak.ui.activity.user.detail_aktivitas.DetailAktivitas
Activity
import com.abcd.pemantauankesehatananak.utils.OnClickItem
import com.abcd.pemantauankesehatananak.utils.SharedPreferencesLogin
import com.abcd.pemantauankesehatananak.utils.TanggalDanWaktu
import com.abcd.pemantauankesehatananak.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint
import java.util.ArrayList

@AndroidEntryPoint
class AktivitasFragment : Fragment() {
    private lateinit var binding: FragmentAktivitasBinding
    private val viewModel: AktivitasViewModel by viewModels()
    private lateinit var sp: SharedPreferencesLogin
    private lateinit var adapterAktivitas: AktivitasAdapter
    private lateinit var adapterKategori: KategoriAdapter
    private var tanggalDanWaktu = TanggalDanWaktu()
    private lateinit var arrayListKategori: ArrayList<KategoriModel>

    private var checkKategoriFirst = true
    private var nameKategori = ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        sp = SharedPreferencesLogin(requireContext())
        val usia = tanggalDanWaktu.hitungUsiaDalamBulan(sp.getTanggalLahir())
        fetchAktivitas(sp.getIdUser(), usia)
    }

    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
savedInstanceState: Bundle?): View {
        binding = FragmentAktivitasBinding.inflate(inflater, container, false)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        setTopAppBar()
```

```
        getKategori()
        getAktivitas()
    }

    private fun setTopAppBar() {
        binding.myAppBar.apply {
//            etSearch.hint = "Cari judul, kategori, usia"
            etSearch.addTextChangedListener(object : TextWatcher{
                override fun beforeTextChanged(
                    s: CharSequence?,
                    start: Int,
                    count: Int,
                    after: Int
                ) {

                }

                override fun onTextChanged(s: CharSequence?, start: Int, before:
Int, count: Int) {

                }

                override fun afterTextChanged(s: Editable?) {
                    adapterAktivitas.searchData(s.toString())
                    if(!checkKategoriFirst){
                        adapterAktivitas.searchKategori(nameKategori)
                    }
                }

            })
        }
    }

    private fun fetchAktivitas(idUser: Int, usia: Int) {
        viewModel.loadData(idUser, usia)
    }

    private fun getKategori() {
        viewModel.getKategori.observe(viewLifecycleOwner){result->
            when(result){
                is UIState.Loading-> setStartShimmerKategori()
                is UIState.Success-> setSuccessFetchKategori(result.data)
                is UIState.Failure-> setFailureFetchKategori(result.message)
            }
        }
    }

    private fun setSuccessFetchKategori(data: ArrayList<KategoriModel>) {
        if(data.isNotEmpty()){
            arrayListKategori = arrayListOf()
            arrayListKategori.add(KategoriModel(0, "Semua", ""))
            arrayListKategori.addAll(data)
            setAdapterKategori(arrayListKategori)
//            setAdapterKategori(data)
        } else{
            Toast.makeText(requireContext(), "Tidak ada data",
Toast.LENGTH_SHORT).show()
        }
        setStoptShimmerKategori()
    }
```

```
    private fun setFailureFetchKategori(message: String) {
        Toast.makeText(requireContext(), message, Toast.LENGTH_SHORT).show()
        setStoptShimmerKategori()
    }

    private fun setAdapterKategori(data: ArrayList<KategoriModel>) {
        adapterKategori = KategoriAdapter(data, object: OnClickItem.ClickKategori{
            override fun clickKategori(kategori: KategoriModel) {
                checkKategoriFirst = kategori.id_kategori == 0
                if(checkKategoriFirst){
                    adapterAktivitas.searchKategori("")
                } else{
                    adapterAktivitas.searchKategori(kategori.kategori!!)
                }
                nameKategori = kategori.kategori!!
            }
        })

        binding.apply {
            rvKategori.layoutManager = LinearLayoutManager(requireContext(),
LinearLayoutManager.HORIZONTAL, false)
            rvKategori.adapter = adapterKategori
        }
    }

    private fun getAktivitas() {
        viewModel.getAktivitas.observe(viewLifecycleOwner){result->
            when(result){
                is UIState.Loading-> setStartShimmerAktivitas()
                is UIState.Success-> setSuccessFetchAktivitas(result.data)
                is UIState.Failure-> setFailureFetchAktivitas(result.message)
            }
        }
    }

    private fun setSuccessFetchAktivitas(data: ArrayList<AktivitasModel>) {
        if(data.isNotEmpty()){
            setAdapter(data)
        } else{
            Toast.makeText(requireContext(), "Tidak ada data",
Toast.LENGTH_SHORT).show()
        }
        setStoptShimmerAktivitas()
    }

    private fun setFailureFetchAktivitas(message: String) {
        Toast.makeText(requireContext(), message, Toast.LENGTH_SHORT).show()
        setStoptShimmerAktivitas()
    }

    private fun setAdapter(data: ArrayList<AktivitasModel>) {
        adapterAktivitas = AktivitasAdapter(data, object:
OnClickItem.ClickAktivitas{
            override fun clickAktivitas(aktivitas: AktivitasModel) {
                val i = Intent(requireContext(),
DetailAktivitasActivity::class.java)
                i.putExtra("aktivitas", aktivitas)
                startActivity(i)
            }
        }, false)
```

```
        binding.apply {
            rvAktivitas.layoutManager = LinearLayoutManager(requireContext(),
LinearLayoutManager.VERTICAL, false)
            rvAktivitas.adapter = adapterAktivitas
        }
    }

    private fun setStartShimmerKategori(){
        binding.apply {
            smKategori.startShimmer()
            smKategori.visibility = View.VISIBLE
            rvKategori.visibility = View.GONE
        }
    }
    private fun setStoptShimmerKategori(){
        binding.apply {
            smKategori.stopShimmer()
            smKategori.visibility = View.GONE
            rvKategori.visibility = View.VISIBLE
        }
    }

    private fun setStartShimmerAktivitas(){
        binding.apply {
            smAktivitas.startShimmer()
            smAktivitas.visibility = View.VISIBLE
            rvAktivitas.visibility = View.GONE
        }
    }
    private fun setStoptShimmerAktivitas(){
        binding.apply {
            smAktivitas.stopShimmer()
            smAktivitas.visibility = View.GONE
            rvAktivitas.visibility = View.VISIBLE
        }
    }
}
```

Lampiran 2. Kode Program DetailAktivitasActivity.kt

```
package com.abcd.pemantauankesehatananak.ui.activity.user.detail_aktivitas

import android.annotation.SuppressLint
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AppCompatActivity
import androidx.lifecycle.lifecycleScope
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.pemantauankesehatananak.adapter.RiwayatAktivitasAdapter
import com.abcd.pemantauankesehatananak.data.model.AktivitasModel
import com.abcd.pemantauankesehatananak.data.model.ResponseModel
import com.abcd.pemantauankesehatananak.data.model.RiwayatAktivitasModel
import com.abcd.pemantauankesehatananak.databinding.ActivityDetailAktivitasBinding
import com.abcd.pemantauankesehatananak.utils.SharedPreferencesLogin
import com.abcd.pemantauankesehatananak.utils.network.UIState
import com.pierfrancescosoffritti.androidyoutubeplayer.core.player.YouTubePlayer
import
com.pierfrancescosoffritti.androidyoutubeplayer.core.player.listeners.AbstractYouT
ubePlayerListener
```

```
import dagger.hilt.android.AndroidEntryPoint
import kotlinx.coroutines.launch
import java.util.ArrayList

@Suppress("DEPRECATION")
@AndroidEntryPoint
class DetailAktivitasActivity : AppCompatActivity() {
    private lateinit var binding: ActivityDetailAktivitasBinding
    private val viewModel: DetailAktivitasViewModel by viewModels()
    private lateinit var aktivitas: AktivitasModel
    private lateinit var sharedPreferences: SharedPreferencesLogin
    private var idAktivitas: Int = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityDetailAktivitasBinding.inflate(layoutInflater)
        setContentView(binding.root)

        sharedPreferences = SharedPreferencesLogin(this@DetailAktivitasActivity)
        setDataSebelumnya()
        setButton()
        fetchRiwayatAktivitas()
        getRiwayatAktivitas()
        getUpdateSelesai()
    }

    private fun setButton() {
        binding.apply {
            btnTandai.setOnClickListener {
                postUpdateSelesai()
            }
            myAppBar.ivBack.setOnClickListener {

                finish()
            }
        }
    }

    private fun setDataSebelumnya() {
        val i = intent
        if(i!=null){
            aktivitas = i.getParcelableExtra("aktivitas")!!
            idAktivitas = aktivitas.id_aktivitas!!
            loadDetailAktivitas(aktivitas)
            setYoutubVideo(aktivitas.video_url!!)
        }
    }

    @SuppressLint("SetTextI18n")
    private fun loadDetailAktivitas(aktivitas: AktivitasModel) {
        lifecycleScope.launch {
            binding.apply {
                setTopAppBar()
                tvJudul.text = aktivitas.judul
                tvKategori.text = "Kategori : "+aktivitas.kategori!!.kategori
                tvUsia.text = "Usia Minimal : ${aktivitas.usia_minimal} Bulan"
                tvDeskripsi.text = aktivitas.deskripsi
            }
        }
    }
```

```
    private fun setTopAppBar() {
        binding.myAppBar.apply {
            ivNavDrawer.visibility = View.GONE
            ivBack.visibility = View.VISIBLE

            tvTitle.text = aktivitas.judul
        }
    }

    private fun setYoutubVideo(videoUrl: String) {
        val videoId = searchIdUrlVideo(videoUrl)
        lifecycle.addObserver(binding.youtubePlayerView)

        binding.youtubePlayerView.addYouTubePlayerListener(object :
AbstractYouTubePlayerListener() {
            override fun onReady(youTubePlayer: YouTubePlayer) {
                youTubePlayer.loadVideo(videoId, 0f)
            }
        })
    }

    private fun searchIdUrlVideo(urlVideo: String): String {
        var url = ""
        try {
            val arrayUrlIdVideo = urlVideo.split("v=")
            url = arrayUrlIdVideo[1]

            try {
                val arraySearchUrlSymbol = url.split("&")
                url = arraySearchUrlSymbol[0]
            } catch (_: Exception){
                url = arrayUrlIdVideo[1]
            }
        } catch (ex: Exception){
            try {
                val arrayUrlIdVideo = urlVideo.split("si=")
                url = arrayUrlIdVideo[1]
            } catch (ex: Exception){
                url = "0"
            }
        }
        return url
    }

    private fun fetchRiwayatAktivitas() {
        viewModel.loadData(sharedPreferences.getIdUser(), idAktivitas)
    }

    private fun getRiwayatAktivitas() {

viewModel.getRiwayatAktivitas.observe(this@DetailAktivitasActivity){result->
            when(result){
                is UIState.Loading-> {}
                is UIState.Success-> setSuccessFetchRiwyatAktivitas(result.data)
                is UIState.Failure->
setFailureFetchRiwyatAktivitas(result.message)
            }
        }
    }

    private fun setSuccessFetchRiwyatAktivitas(data:
```

```
ArrayList<RiwayatAktivitasModel>) {
        if(data.isNotEmpty()){
            setAdapterRiwayatAktivitas(data)
        } else{
            binding.tvTitleRiwayatAktivitas.visibility = View.GONE

            Toast.makeText(this@DetailAktivitasActivity,
"${sharedPreferences.getIdUser()}-$idAktivitas", Toast.LENGTH_SHORT).show()
        }
    }

    private fun setAdapterRiwayatAktivitas(data: ArrayList<RiwayatAktivitasModel>)
{
        val adapterRiwayatAktivitas = RiwayatAktivitasAdapter(data)
        binding.rvRiwayatAktivitas.apply {
            layoutManager = LinearLayoutManager(this@DetailAktivitasActivity,
LinearLayoutManager.VERTICAL, false)
            adapter = adapterRiwayatAktivitas
        }
    }

    private fun setFailureFetchRiwyatAktivitas(message: String) {
        Toast.makeText(this@DetailAktivitasActivity, message,
Toast.LENGTH_SHORT).show()
        binding.tvTitleRiwayatAktivitas.visibility = View.GONE
    }


    private fun postUpdateSelesai() {
        viewModel.postUpdateSelesai(sharedPreferences.getIdUser(), idAktivitas)
    }

    private fun getUpdateSelesai() {
        viewModel.getUpdateSelesai.observe(this@DetailAktivitasActivity){result->
            when(result){
                is UIState.Loading-> {}
                is UIState.Success-> setSuccessPostUpdateSelesai(result.data)
                is UIState.Failure-> setFailurePostUpdateSelesai(result.message)
            }
        }
    }

    private fun setSuccessPostUpdateSelesai(data: ResponseModel) {
        if(data.status == "0"){
            Toast.makeText(this@DetailAktivitasActivity, "Ditandai selesai",
Toast.LENGTH_SHORT).show()
            fetchRiwayatAktivitas()
        } else{
            Toast.makeText(this@DetailAktivitasActivity, data.message_response,
Toast.LENGTH_SHORT).show()
        }
    }

    private fun setFailurePostUpdateSelesai(message: String) {
        Toast.makeText(this@DetailAktivitasActivity, message,
Toast.LENGTH_SHORT).show()
    }
}
```

Lampiran 3. Kode Program DetailPemeriksaanActivity.kt

```
package com.abcd.pemantauankesehatananak.ui.activity.pemeriksaan

import android.annotation.SuppressLint
import android.os.Bundle
import android.view.View
import androidx.appcompat.app.AppCompatActivity
import com.abcd.pemantauankesehatananak.data.model.PelayananModel
import
com.abcd.pemantauankesehatananak.databinding.ActivityDetailPemeriksaanBinding
import com.abcd.pemantauankesehatananak.utils.TanggalDanWaktu

@Suppress("DEPRECATION")
class DetailPemeriksaanActivity : AppCompatActivity() {
    private lateinit var binding: ActivityDetailPemeriksaanBinding
    private var pelayanan: PelayananModel? = null
    private val tanggalDanWaktu = TanggalDanWaktu()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityDetailPemeriksaanBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setDataSebelumnya()
        setTopBar()
    }

    private fun setDataSebelumnya() {
        val i = intent
        if(i != null){
            pelayanan = i.getParcelableExtra("pemeriksaan")
            setData(pelayanan)
        }
    }

    private fun setTopBar() {
        binding.myAppBar.apply {
            tvTitle.text = pelayanan?.pelayanan
            ivNavDrawer.visibility = View.GONE
            ivBack.visibility = View.VISIBLE

            ivBack.setOnClickListener {
                finish()
            }
        }
    }

    private fun setData(pelayanan: PelayananModel?) {
        pelayanan?.let {data->
            setDataKeterangan(data)
            setDataPemeriksaan(data)
        }
    }

    @SuppressLint("SetTextI18n")
    private fun setDataKeterangan(data: PelayananModel) {
        binding.apply {
            tvPenanggungJawab.text = data.penanggung_jawab
            tvJabatan.text = data.jabatan
            data.tanggal?.let {tanggal->
                tvTanggal.text = tanggalDanWaktu.konversiBulan(tanggal)
```

```
            }
            data.waktu?.let {waktu->
                tvWaktu.text = tanggalDanWaktu.waktuNoSecond(waktu)+" WITA"
            }
        }
    }

    private fun setDataPemeriksaan(data: PelayananModel) {
        binding.apply {
            tvHasil.text = data.hasil
            tvKeterangan.text = data.keterangan
            tvCatatan.text = data.catatan
        }
    }

}
```

Lampiran 4. Kode Program HomeFragment.kt

```
package com.abcd.pemantauankesehatananak.ui.fragment.user.home

import android.content.Intent
import androidx.fragment.app.viewModels
import android.os.Bundle
import android.util.Log
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.pemantauankesehatananak.R
import com.abcd.pemantauankesehatananak.adapter.AktivitasAdapter
import com.abcd.pemantauankesehatananak.adapter.MilestoneAdapter
import com.abcd.pemantauankesehatananak.data.model.AktivitasModel
import com.abcd.pemantauankesehatananak.data.model.MilestoneModel
import com.abcd.pemantauankesehatananak.databinding.FragmentHomeBinding
import
com.abcd.pemantauankesehatananak.ui.activity.user.detail_aktivitas.DetailAktivitas
Activity
import com.abcd.pemantauankesehatananak.ui.activity.user.main.MainActivity
import com.abcd.pemantauankesehatananak.utils.OnClickItem
import com.abcd.pemantauankesehatananak.utils.SharedPreferencesLogin
import com.abcd.pemantauankesehatananak.utils.TanggalDanWaktu
import com.abcd.pemantauankesehatananak.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint
import java.util.ArrayList

@AndroidEntryPoint
class HomeFragment : Fragment() {
    private lateinit var binding: FragmentHomeBinding
    private val viewModel: HomeViewModel by viewModels()
    private val tanggalDanWaktu = TanggalDanWaktu()
    private lateinit var sharedPreferences: SharedPreferencesLogin

    // Adapter
    private lateinit var adapterAktivitas: AktivitasAdapter
    private lateinit var adapterMilestone: MilestoneAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
```

```
        sharedPreferences = SharedPreferencesLogin(requireContext())
        fetchData()
    }

    private fun fetchData() {
        val umur =
tanggalDanWaktu.hitungUsiaDalamBulan(sharedPreferences.getTanggalLahir())
        viewModel.loadData(sharedPreferences.getIdUser(), umur)
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        binding = FragmentHomeBinding.inflate(layoutInflater)

        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        setButton()
        getAktivitas()
        getMilestone()
    }

    private fun setButton() {
        binding.apply {
            btnAktivitas.setOnClickListener {
                (activity as MainActivity).setAktivitasFragment()
                (activity as MainActivity).binding.bottomNavigation.selectedItemId
= R.id.menu_aktivitas
            }
            btnMilestone.setOnClickListener {
                (activity as MainActivity).setMilestoneFragment()
                (activity as MainActivity).binding.bottomNavigation.selectedItemId
= R.id.menu_milestone
            }
            llAktivitas.setOnClickListener {
                (activity as MainActivity).setAktivitasFragment()
                (activity as MainActivity).binding.bottomNavigation.selectedItemId
= R.id.menu_aktivitas
            }
            llMilestone.setOnClickListener {
                (activity as MainActivity).setMilestoneFragment()
                (activity as MainActivity).binding.bottomNavigation.selectedItemId
= R.id.menu_milestone
            }
            llProfile.setOnClickListener {
                (activity as MainActivity).setProfileFragment()
                (activity as MainActivity).binding.bottomNavigation.selectedItemId
= R.id.menu_profile
            }
        }
    }

    private fun getAktivitas() {
        viewModel.getAktivitas.observe(viewLifecycleOwner){result->
            when(result){
```

```
                is UIState.Loading-> setStartShimmerAktivitas()
                is UIState.Success-> setSuccessFetchAktivitas(result.data)
                is UIState.Failure-> setFailureFetchAktivitas(result.message)
            }
        }
    }

    private fun setSuccessFetchAktivitas(data: ArrayList<AktivitasModel>) {
        if(data.isNotEmpty()){
            setAdapterAktivitas(data)
        } else{
            Toast.makeText(requireContext(), "Tidak ada data",
Toast.LENGTH_SHORT).show()
        }
        setStoptShimmerAktivitas()
    }

    private fun setFailureFetchAktivitas(message: String) {
        Toast.makeText(requireContext(), message, Toast.LENGTH_SHORT).show()
        Log.e("HomeFragmentTAG", "$message ")
        setStoptShimmerAktivitas()
    }

    private fun getMilestone() {
        viewModel.getMilestone.observe(viewLifecycleOwner){result->
            when(result){
                is UIState.Loading-> setStartShimmerMilestone()
                is UIState.Success-> setSuccessFetchMilestone(result.data)
                is UIState.Failure-> setFailureFetchMilestone(result.message)
            }
        }
    }

    private fun setSuccessFetchMilestone(data: ArrayList<MilestoneModel>) {
        if(data.isNotEmpty()){
            setAdapterMilestone(data)
        } else{
            Toast.makeText(requireContext(), "Tidak ada data",
Toast.LENGTH_SHORT).show()
        }
        setStoptShimmerMilestone()
    }

    private fun setFailureFetchMilestone(message: String) {
        Toast.makeText(requireContext(), message, Toast.LENGTH_SHORT).show()
        Log.e("HomeFragmentTAG", "$message ")
        setStoptShimmerMilestone()
    }


    // Set adapter
    // Aktivitas
    private fun setAdapterAktivitas(data: ArrayList<AktivitasModel>) {
        adapterAktivitas = AktivitasAdapter(data, object :
OnClickItem.ClickAktivitas{
            override fun clickAktivitas(aktivitas: AktivitasModel) {
                val i = Intent(requireContext(),
DetailAktivitasActivity::class.java)
                i.putExtra("aktivitas", aktivitas)
                startActivity(i)
            }
```

```
        }, true)
        binding.rvAktivitas.apply {
            adapter = adapterAktivitas
            layoutManager = LinearLayoutManager(requireContext(),
LinearLayoutManager.VERTICAL, false)
        }
    }

    // Milestone
    private fun setAdapterMilestone(data: ArrayList<MilestoneModel>) {
        adapterMilestone = MilestoneAdapter(data, object:
OnClickItem.ClickMilestone{
            override fun clickMilestone(milestone: MilestoneModel) {

            }
        }, true)
        binding.rvMilestone.apply {
            adapter = adapterMilestone
            layoutManager = LinearLayoutManager(requireContext(),
LinearLayoutManager.VERTICAL, false)
        }
    }

    // Shimmer
    private fun setStartShimmerAktivitas(){
        binding.apply {
            smAktivitas.startShimmer()
            smAktivitas.visibility = View.VISIBLE
            rvAktivitas.visibility = View.GONE
        }
    }
    private fun setStoptShimmerAktivitas(){
        binding.apply {
            smAktivitas.stopShimmer()
            smAktivitas.visibility = View.GONE
            rvAktivitas.visibility = View.VISIBLE
        }
    }

    private fun setStartShimmerMilestone(){
        binding.apply {
            smMilestone.startShimmer()
            smMilestone.visibility = View.VISIBLE
            rvMilestone.visibility = View.GONE
        }
    }
    private fun setStoptShimmerMilestone(){
        binding.apply {
            smMilestone.stopShimmer()
            smMilestone.visibility = View.GONE
            rvMilestone.visibility = View.VISIBLE
        }
    }

}
```

Lampiran 5. Kode Program LoginActivity.kt

```
package com.abcd.pemantauankesehatananak.ui.activity.login

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.activity.viewModels
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.abcd.pemantauankesehatananak.R
import com.abcd.pemantauankesehatananak.data.model.UserModel
import com.abcd.pemantauankesehatananak.databinding.ActivityLoginBinding
import com.abcd.pemantauankesehatananak.ui.activity.register.RegisterActivity
import com.abcd.pemantauankesehatananak.ui.activity.user.main.MainActivity
import com.abcd.pemantauankesehatananak.utils.LoadingAlertDialog
import com.abcd.pemantauankesehatananak.utils.SharedPreferencesLogin
import com.abcd.pemantauankesehatananak.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

@AndroidEntryPoint
class LoginActivity : AppCompatActivity() {
    private lateinit var binding: ActivityLoginBinding
    private val viewModel: LoginViewModel by viewModels()
    @Inject
    lateinit var loading: LoadingAlertDialog
    private lateinit var sharedPreferencesLogin: SharedPreferencesLogin
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityLoginBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setupListeners()
        setSharedPreferencesLogin()
        getUser()
    }


    private fun setSharedPreferencesLogin() {
        sharedPreferencesLogin = SharedPreferencesLogin(this@LoginActivity)
    }

    private fun setupListeners() {
        binding.apply {
            btnLogin.setOnClickListener {
                if (validateInput()) {
                    performLogin()
                }
            }

            tvRegister.setOnClickListener {
                navigateToRegister()
            }
        }
    }

    private fun validateInput(): Boolean {
```

```
        binding.apply {
            if (etNomorKtp.text.toString().isEmpty()) {
                etNomorKtp.error = "Nomor No BPJS / Username harus diisi"
                return false
            }
            if (etPassword.text.toString().isEmpty()) {
                etPassword.error = "Password harus diisi"
                return false
            }
        }
        return true
    }

    private fun performLogin() {
        var noKtpUsername: String
        var password: String
        binding.apply {
            noKtpUsername = etNomorKtp.text.toString()
            password = etPassword.text.toString()
        }
        fetchUser(noKtpUsername, password)
    }

    private fun fetchUser(noBpjUsername: String, password: String){
        viewModel.fetchUser(noBpjUsername, password)
    }

    private fun getUser(){
        viewModel.getUser().observe(this@LoginActivity){result->
            when(result){
                is UIState.Loading->
loading.alertDialogLoading(this@LoginActivity)
                is UIState.Success-> setSuccessFetchUser(result.data)
                is UIState.Failure-> setFailureFetchUser(result.message)
            }
        }
    }

    private fun setSuccessFetchUser(data: UserModel) {
        if(checkFetchUser(data)){
            setDataSharedPreferences(data)
        } else{
            Toast.makeText(this@LoginActivity, "Data tidak ditemukan \n" +
                    "Pastikan Username dan Password Terdaftar ada",
Toast.LENGTH_SHORT).show()
        }
        loading.alertDialogCancel()
    }

    private fun setFailureFetchUser(message: String) {
//        Toast.makeText(this@LoginActivity, "Data tidak ditemukan \n" +
//                "Pastikan No Bpjs/Username dan Password Terdaftar ada",
Toast.LENGTH_SHORT).show()
        Toast.makeText(this@LoginActivity, "Data tidak ditemukan \n" +
                "Pastikan Username dan Password Terdaftar ada",
Toast.LENGTH_SHORT).show()
        Log.d("LoginTAG", "setFailureFetchUser: $message")
        loading.alertDialogCancel()
    }

    private fun checkFetchUser(data: UserModel): Boolean{
```

```
        return data.idUser != null
    }

    private fun setDataSharedPreferences(data: UserModel){
        try {
            data.apply {
                sharedPreferencesLogin.setLogin(
                    idUser!!, no_ktp!!, nama!!, nomorHp!!, alamat!!, nama_anak!!,
                    tanggal_lahir!!, jk!!, username!!, password!!, sebagai!!
                )
                Toast.makeText(this@LoginActivity, "Berhasil",
Toast.LENGTH_SHORT).show()
                moveToMainActivity(sebagai!!)
            }
        } catch (ex: Exception){
            Toast.makeText(this@LoginActivity, ex.message,
Toast.LENGTH_SHORT).show()
        }
    }

    private fun moveToMainActivity(sebagai: String){
        if(sebagai == "user"){
            startActivity(Intent(this@LoginActivity, MainActivity::class.java))
        } else{
            Toast.makeText(this@LoginActivity, "Data tidak ditemukan \n" +
                    "Pastikan No KTP dan Password Terdaftar ada",
Toast.LENGTH_SHORT).show()
        }
    }

    private fun navigateToRegister() {
        val intent = Intent(this, RegisterActivity::class.java)
        startActivity(intent)
    }
}
```

Lampiran 6. Kode Program MainActivity.kt

```
package com.abcd.pemantauankesehatananak.ui.activity.user.main

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.abcd.pemantauankesehatananak.R
import com.abcd.pemantauankesehatananak.databinding.ActivityMainBinding
import
com.abcd.pemantauankesehatananak.ui.fragment.user.aktivitas.AktivitasFragment
import com.abcd.pemantauankesehatananak.ui.fragment.user.home.HomeFragment
import
com.abcd.pemantauankesehatananak.ui.fragment.user.pemeriksaan.PemeriksaanFragment
import
com.abcd.pemantauankesehatananak.ui.fragment.user.perkembangan.MilestoneFragment
import com.abcd.pemantauankesehatananak.ui.fragment.user.profile.ProfileFragment
import dagger.hilt.android.AndroidEntryPoint

@AndroidEntryPoint
class MainActivity : AppCompatActivity() {
    lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
```

```
    setContentView(binding.root)

    supportFragmentManager.beginTransaction()
        .replace(R.id.fragment_container, HomeFragment())
        .commit()

    binding.bottomNavigation.setOnItemSelectedListener { item ->
        when (item.itemId) {
            R.id.menu_home -> {
                setHomeFragment()
                true
            }
            R.id.menu_aktivitas -> {
                setAktivitasFragment()
                true
            }
            R.id.menu_milestone -> {
                setMilestoneFragment()
                true
            }
            R.id.menu_milestone -> {
                setMilestoneFragment()
                true
            }
            R.id.menu_pemeriksaan->{
                setPemeriksaanFragment()
                true
            }
            R.id.menu_profile -> {
                setProfileFragment()
                true
            }
            else -> false
        }
    }
}

private fun setHomeFragment(){
    supportFragmentManager.beginTransaction()
        .replace(R.id.fragment_container, HomeFragment())
        .commit()
}

fun setAktivitasFragment(){
    supportFragmentManager.beginTransaction()
        .replace(R.id.fragment_container, AktivitasFragment())
        .commit()
}

fun setMilestoneFragment(){
    supportFragmentManager.beginTransaction()
        .replace(R.id.fragment_container, MilestoneFragment())
        .commit()
}

fun setPemeriksaanFragment(){
    supportFragmentManager.beginTransaction()
        .replace(R.id.fragment_container, PemeriksaanFragment())
        .commit()
}
```

```
    fun setProfileFragment(){
        supportFragmentManager.beginTransaction()
            .replace(R.id.fragment_container, ProfileFragment())
            .commit()
    }
}
```

Lampiran 7. Kode Program MilestoneFragment.kt

```
package com.abcd.pemantauankesehatananak.ui.fragment.user.perkembangan

import android.os.Bundle
import android.text.Editable
import android.text.TextWatcher
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.fragment.app.viewModels
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.pemantauankesehatananak.adapter.KategoriAdapter
import com.abcd.pemantauankesehatananak.adapter.MilestoneAdapter
import com.abcd.pemantauankesehatananak.data.model.KategoriModel
import com.abcd.pemantauankesehatananak.data.model.MilestoneModel
import com.abcd.pemantauankesehatananak.data.model.ResponseModel
import com.abcd.pemantauankesehatananak.databinding.FragmentMilestoneBinding
import com.abcd.pemantauankesehatananak.utils.LoadingAlertDialog
import com.abcd.pemantauankesehatananak.utils.OnClickItem
import com.abcd.pemantauankesehatananak.utils.SharedPreferencesLogin
import com.abcd.pemantauankesehatananak.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject
import kotlin.collections.ArrayList

@AndroidEntryPoint
class MilestoneFragment : Fragment() {
    private lateinit var binding: FragmentMilestoneBinding
    private val viewModel: MilestoneViewModel by viewModels()
    private lateinit var sp: SharedPreferencesLogin
    private lateinit var adapterMilestone: MilestoneAdapter
    private lateinit var adapterKategori: KategoriAdapter
    @Inject lateinit var loading: LoadingAlertDialog
    private lateinit var arrayListKategori: ArrayList<KategoriModel>

    private var checkKategoriFirst = true
    private var nameKategori = ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        sp = SharedPreferencesLogin(requireContext())
        fetchData()
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?, savedInstanceState:
Bundle?
    ): View {
        binding = FragmentMilestoneBinding.inflate(inflater, container, false)
```

```
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        setTopAppBar()
        getKategori()
        getMilestone()
        getUpdateMilestone()
    }

    private fun setTopAppBar() {
        binding.myAppBar.apply {
            etSearch.hint = "Cari judul, kategori, usia ideal"
            etSearch.addTextChangedListener(object : TextWatcher {
                override fun beforeTextChanged(
                    s: CharSequence?,
                    start: Int,
                    count: Int,
                    after: Int
                ) {

                }

                override fun onTextChanged(s: CharSequence?, start: Int, before:
Int, count: Int) {

                }

                override fun afterTextChanged(s: Editable?) {
                    adapterMilestone.searchData(s.toString())
                    if(!checkKategoriFirst){
                        adapterMilestone.searchKategori(nameKategori)
                    }
                }

            })
        }
    }

    private fun fetchData() {
        viewModel.loadData(sp.getIdUser())
    }

    private fun getKategori() {
        viewModel.getKategori.observe(viewLifecycleOwner){result->
            when(result){
                is UIState.Loading-> setStartShimmerKategori()
                is UIState.Success-> setSuccessFetchKategori(result.data)
                is UIState.Failure-> setFailureFetchKategori(result.message)
            }
        }
    }

    private fun setSuccessFetchKategori(data: ArrayList<KategoriModel>) {
        if(data.isNotEmpty()){
            arrayListKategori = arrayListOf()
            arrayListKategori.add(KategoriModel(0, "Semua", ""))
            arrayListKategori.addAll(data)
            setAdapterKategori(arrayListKategori)
```

```
//            setAdapterKategori(data)
        } else{
            Toast.makeText(requireContext(), "Tidak ada data",
Toast.LENGTH_SHORT).show()
        }
        setStoptShimmerKategori()
    }

    private fun setFailureFetchKategori(message: String) {
        Toast.makeText(requireContext(), message, Toast.LENGTH_SHORT).show()
        setStoptShimmerKategori()
    }

    private fun setAdapterKategori(data: ArrayList<KategoriModel>) {
        adapterKategori = KategoriAdapter(data, object: OnClickItem.ClickKategori{
            override fun clickKategori(kategori: KategoriModel) {
                checkKategoriFirst = kategori.id_kategori == 0
                if(checkKategoriFirst){
                    adapterMilestone.searchKategori("")
                } else{
                    adapterMilestone.searchKategori(kategori.kategori!!)
                }
                nameKategori = kategori.kategori!!
            }
        })

        binding.apply {
            rvKategori.layoutManager = LinearLayoutManager(requireContext(),
LinearLayoutManager.HORIZONTAL, false)
            rvKategori.adapter = adapterKategori
        }
    }

    private fun getMilestone() {
        viewModel.getMilestone.observe(viewLifecycleOwner){result->
            when(result){
                is UIState.Loading-> setStartShimmerMilestone()
                is UIState.Success-> setSuccessFetchMilestone(result.data)
                is UIState.Failure-> setFailureFetchMilestone(result.message)
            }
        }
    }

    private fun setSuccessFetchMilestone(data: ArrayList<MilestoneModel>) {
        if(data.isNotEmpty()){
            setAdapterMilestone(data)
        } else{
            Toast.makeText(requireContext(), "Tidak ada data",
Toast.LENGTH_SHORT).show()
        }
        setStoptShimmerMilestone()
    }

    private fun setFailureFetchMilestone(message: String) {
        Toast.makeText(requireContext(), message, Toast.LENGTH_SHORT).show()
        setStoptShimmerMilestone()
    }

    private fun setAdapterMilestone(data: ArrayList<MilestoneModel>) {
        adapterMilestone = MilestoneAdapter(data, object:
OnClickItem.ClickMilestone{
```

```
            override fun clickMilestone(milestone: MilestoneModel) {
                setUpdateMilestone(milestone.id_milestone!!)
            }
        }, false)

        binding.apply {
            rvMilestone.layoutManager = LinearLayoutManager(requireContext(),
LinearLayoutManager.VERTICAL, false)
            rvMilestone.adapter = adapterMilestone
        }
    }

    private fun setUpdateMilestone(idAktivitas: Int) {
        viewModel.postUpdateCheck(sp.getIdUser(), idAktivitas)
    }

    private fun getUpdateMilestone(){
        viewModel.getCheck.observe(viewLifecycleOwner){result->
            when(result){
                is UIState.Loading-> loading.alertDialogLoading(requireContext())
                is UIState.Success-> setSuccessUpdateCheck(result.data)
                is UIState.Failure-> setFailureUpdateCheck(result.message)
            }
        }
    }

    private fun setSuccessUpdateCheck(data: ResponseModel) {
        if(data.status=="0"){
            Toast.makeText(requireContext(), "Berhasil",
Toast.LENGTH_SHORT).show()
        } else{
            Toast.makeText(requireContext(), data.message_response,
Toast.LENGTH_SHORT).show()
        }
        loading.alertDialogCancel()
    }

    private fun setFailureUpdateCheck(message: String) {
        Toast.makeText(requireContext(), message, Toast.LENGTH_SHORT).show()
        loading.alertDialogCancel()
    }

    private fun setStartShimmerKategori(){
        binding.apply {
            smKategori.startShimmer()
            smKategori.visibility = View.VISIBLE
            rvKategori.visibility = View.GONE
        }
    }
    private fun setStoptShimmerKategori(){
        binding.apply {
            smKategori.stopShimmer()
            smKategori.visibility = View.GONE
            rvKategori.visibility = View.VISIBLE
        }
    }

    private fun setStartShimmerMilestone(){
        binding.apply {
            smMilestone.startShimmer()
            smMilestone.visibility = View.VISIBLE
```

```
            rvMilestone.visibility = View.GONE
        }
    }
    private fun setStoptShimmerMilestone(){
        binding.apply {
            smMilestone.stopShimmer()
            smMilestone.visibility = View.GONE
            rvMilestone.visibility = View.VISIBLE
        }
    }
}
```

Lampiran 8. Kode Program PemeriksaanFragment.kt

```
package com.abcd.pemantauankesehatananak.ui.fragment.user.pemeriksaan

import androidx.fragment.app.viewModels
import android.os.Bundle
import android.text.Editable
import android.text.TextWatcher
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.recyclerview.widget.LinearLayoutManager
import com.abcd.pemantauankesehatananak.adapter.PemeriksaanAdapter
import com.abcd.pemantauankesehatananak.data.model.PelayananModel
import com.abcd.pemantauankesehatananak.databinding.FragmentPemeriksaanBinding
import com.abcd.pemantauankesehatananak.utils.SharedPreferencesLogin
import com.abcd.pemantauankesehatananak.utils.network.UIState
import dagger.hilt.android.AndroidEntryPoint
import java.util.ArrayList

@AndroidEntryPoint
class PemeriksaanFragment : Fragment() {
    private lateinit var binding: FragmentPemeriksaanBinding
    private val viewModel: PemeriksaanViewModel by viewModels()
    private lateinit var sharedPreferences: SharedPreferencesLogin
    private lateinit var adapterPemeriksaan: PemeriksaanAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        sharedPreferences = SharedPreferencesLogin(requireContext())
        viewModel.fetchPemeriksaan(sharedPreferences.getIdUser())
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        binding = FragmentPemeriksaanBinding.inflate(layoutInflater)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        setTopBarSearch()
        getPemeriksaan()
    }
```

```
    private fun setTopBarSearch() {
        binding.myAppBar.apply {
            etSearch.hint = "Cari Pelayanan, Hasil, Tanggal dan Waktu"
            etSearch.addTextChangedListener(object: TextWatcher{
                override fun beforeTextChanged(
                    s: CharSequence?,
                    start: Int,
                    count: Int,
                    after: Int
                ) {

                }

                override fun onTextChanged(s: CharSequence?, start: Int, before:
Int, count: Int) {
                    adapterPemeriksaan.searchData(s?.trim().toString())
                }

                override fun afterTextChanged(s: Editable?) {

                }

            })
        }
    }

    private fun getPemeriksaan() {
        viewModel.getPemeriksaan.observe(viewLifecycleOwner){result->
            when(result){
                is UIState.Loading-> setStartShimmer()
                is UIState.Failure-> setFailurePemeriksaan(result.message)
                is UIState.Success-> setSuccessPemeriksaan(result.data)
            }
        }
    }

    private fun setFailurePemeriksaan(message: String) {
        setStopShimmer()
        Toast.makeText(requireContext(), message, Toast.LENGTH_SHORT).show()
    }

    private fun setSuccessPemeriksaan(data: ArrayList<PelayananModel>) {
        setStopShimmer()
        if(data.isNotEmpty()){
            setAdapterPemeriksaan(data)
        } else{
            Toast.makeText(requireContext(), "Tidak ada data",
Toast.LENGTH_SHORT).show()
        }
    }

    private fun setAdapterPemeriksaan(data: ArrayList<PelayananModel>) {
        adapterPemeriksaan = PemeriksaanAdapter(data)
        binding.rvPemeriksaan.apply {
            adapter = adapterPemeriksaan
            layoutManager = LinearLayoutManager(requireContext(),
LinearLayoutManager.VERTICAL, false)
        }
    }
```

```
    private fun setStartShimmer(){
        binding.apply {
            smPemeriksaan.startShimmer()
            smPemeriksaan.visibility = View.VISIBLE
            rvPemeriksaan.visibility = View.GONE
        }
    }

    private fun setStopShimmer(){
        binding.apply {
            smPemeriksaan.stopShimmer()
            smPemeriksaan.visibility = View.GONE
            rvPemeriksaan.visibility = View.VISIBLE
        }
    }
}
```

Lampiran 9. Kode Program ProfileFragment.kt

```
package com.abcd.pemantauankesehatananak.ui.fragment.user.profile

import android.annotation.SuppressLint
import android.content.Intent
import android.os.Bundle
import android.util.Log
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.fragment.app.viewModels
import com.abcd.pemantauankesehatananak.R
import com.abcd.pemantauankesehatananak.data.model.ResponseModel
import com.abcd.pemantauankesehatananak.data.model.UserModel
import com.abcd.pemantauankesehatananak.databinding.AlertDialogAkunBinding
import com.abcd.pemantauankesehatananak.databinding.FragmentProfileBinding
import com.abcd.pemantauankesehatananak.ui.activity.login.LoginActivity
import com.abcd.pemantauankesehatananak.utils.LoadingAlertDialog
import com.abcd.pemantauankesehatananak.utils.SharedPreferencesLogin
import com.abcd.pemantauankesehatananak.utils.TanggalDanWaktu
import com.abcd.pemantauankesehatananak.utils.network.UIState
import com.google.android.material.datepicker.MaterialDatePicker
import dagger.hilt.android.AndroidEntryPoint
import java.text.SimpleDateFormat
import java.util.Calendar
import java.util.Date
import java.util.Locale
import java.util.TimeZone
import javax.inject.Inject

@AndroidEntryPoint
class ProfileFragment : Fragment() {
    private lateinit var binding: FragmentProfileBinding
    private val viewModel : ProfileViewModel by viewModels()
    private lateinit var sharedPreferences: SharedPreferencesLogin
    private lateinit var tanggalDanWaktu: TanggalDanWaktu
    private var selectedDate: Date? = null
    private lateinit var tempUser: UserModel
    @Inject lateinit var loading: LoadingAlertDialog
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        sharedPreferences = SharedPreferencesLogin(requireContext())
        tanggalDanWaktu = TanggalDanWaktu()
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        binding = FragmentProfileBinding.inflate(layoutInflater)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        setTopAppNavDrawer()
        setViewData()
        setButton()
        getUpdateDiri()
    }

    @SuppressLint("SetTextI18n")
    private fun setTopAppNavDrawer() {
        binding.navTopBar.apply {
            ivBack.visibility = View.GONE
            ivNavDrawer.visibility = View.GONE
            tvTitle.text = "Profile Anda"
        }
    }

    @SuppressLint("SetTextI18n")
    private fun setViewData(){
        binding.apply {
            tvNoKtp.text = sharedPreferences.getNoKtp()
            tvNama.text = sharedPreferences.getNama()
            tvNomorHp.text = sharedPreferences.getNomorHp()
            tvAlamat.text = sharedPreferences.getAlamat()
            tvNamaAnak.text = sharedPreferences.getNamaAnak()
            tvTanggalLahir.text = sharedPreferences.getTanggalLahir()
            tvTanggalLahir.text = sharedPreferences.getTanggalLahir()
            tvUsia.text =
"${tanggalDanWaktu.hitungUsiaDalamBulan(sharedPreferences.getTanggalLahir())}
Bulan"
            tvJenisKelamin.text = sharedPreferences.getJenisKelamin()
            tvUsername.text = sharedPreferences.getUsername()
            tvPassword.text = sharedPreferences.getPassword()
        }
    }

    private fun setButton() {
        binding.apply {
            btnUbahData.setOnClickListener {
                setShowEditProfile()
            }
            btnLogout.setOnClickListener {
                sharedPreferences.setLogout()
                startActivity(Intent(requireActivity(),
LoginActivity::class.java))
```

```
            requireActivity().finish()
        }
    }
}

private fun setShowEditProfile() {
    val view = AlertDialogAkunBinding.inflate(layoutInflater)
    val alertDialog = AlertDialog.Builder(requireContext())
    alertDialog.setView(view.root)
        .setCancelable(false)
    val dialogInputan = alertDialog.create()
    dialogInputan.show()

    view.apply {
        etEditNoKtp.setText(sharedPreferences.getNoKtp())
        etEditNama.setText(sharedPreferences.getNama())
        etEditNomorHp.setText(sharedPreferences.getNomorHp())
        etEditAlamat.setText(sharedPreferences.getAlamat())
        etEditNamaAnak.setText(sharedPreferences.getNamaAnak())
        etEditTanggalLahir.text = sharedPreferences.getTanggalLahir()
        etEditUsername.setText(sharedPreferences.getUsername())
        etEditPassword.setText(sharedPreferences.getPassword())
        when(sharedPreferences.getJenisKelamin()){
            "Laki-laki"-> rbLakiLaki.isChecked = true
            "Perempuan"-> rbPerempuan.isChecked = true
        }

        etEditTanggalLahir.setOnClickListener {
            showDatePicker(view, sharedPreferences.getTanggalLahir())
        }

        var selectedGender = ""
        rgGender.setOnCheckedChangeListener { _, checkedId ->
            when (checkedId) {
                R.id.rbLakiLaki -> selectedGender = "Laki-laki"
                R.id.rbPerempuan -> selectedGender = "Perempuan"
            }
        }

        btnSimpan.setOnClickListener {
            var cek = false
            if(etEditNoKtp.toString().isEmpty()){
                etEditNoKtp.error = "Tidak Boleh Kosong"
                cek = true
            }
            if(etEditNama.toString().isEmpty()){
                etEditNama.error = "Tidak Boleh Kosong"
                cek = true
            }
            if(etEditAlamat.toString().isEmpty()){
                etEditAlamat.error = "Tidak Boleh Kosong"
                cek = true
            }
            if(etEditNomorHp.toString().isEmpty()){
                etEditNomorHp.error = "Tidak Boleh Kosong"
                cek = true
            }
            if(etEditNamaAnak.toString().isEmpty()){
                etEditNamaAnak.error = "Tidak Boleh Kosong"
                cek = true
            }
```

```
                if(etEditTanggalLahir.text.toString() ==
resources.getString(R.string.ket_klik_tanggal)){
                    etEditTanggalLahir.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){
                    val noKtp = etEditNoKtp.text.toString()
                    val nama = etEditNama.text.toString()
                    val nomorHp = etEditNomorHp.text.toString()
                    val alamat = etEditAlamat.text.toString()
                    val namaAnak = etEditNamaAnak.text.toString()
                    val tanggalLahir = etEditTanggalLahir.text.toString()
                    val jenisKelamin = selectedGender
                    val username = etEditUsername.text.toString()
                    val password = etEditPassword.text.toString()
                    val usernameLama = sharedPreferences.getUsername()

                    tempUser = UserModel(
                        sharedPreferences.getIdUser(), noKtp,
                        nama, nomorHp, alamat, namaAnak, tanggalLahir,
jenisKelamin,
                        username, password, usernameLama
                    )
                    postUpdateData(
                        sharedPreferences.getIdUser(), noKtp, nama, nomorHp,
alamat,
                        namaAnak, tanggalLahir, jenisKelamin, username, password,
usernameLama
                    )
                }
                dialogInputan.dismiss()
            }
            btnBatal.setOnClickListener {
                dialogInputan.dismiss()
            }
        }
    }

    private fun postUpdateData(
        idUser: Int,
        noKtp: String,
        nama: String,
        nomorHp: String,
        alamat: String,
        namaAnak: String,
        tanggalLahir: String,
        jenisKelamin: String,
        username: String,
        password: String,
        usernameLama: String
    ) {
        viewModel.postUpdateDataDiri(
```

```
            idUser, noKtp, nama, nomorHp, alamat, namaAnak, tanggalLahir,
            jenisKelamin, username, password, usernameLama
        )
    }

    private fun getUpdateDiri(){
        viewModel.getResponseUpdateDataDiri.observe(viewLifecycleOwner){result->
            when(result){
                is UIState.Loading-> loading.alertDialogLoading(requireContext())
                is UIState.Success-> setSuccessUpdateDiri(result.data)
                is UIState.Failure-> setFailureUpdateDiri(result.message)
            }
        }
    }

    private fun setSuccessUpdateDiri(data: ResponseModel) {
        if(data.status == "0"){
            Toast.makeText(requireContext(), "Berhasil Update",
Toast.LENGTH_SHORT).show()
            tempUser.apply {
                sharedPreferences.setLogin(
                    idUser!!, no_ktp!!, nama!!, nomorHp!!, alamat!!, nama_anak!!,
tanggal_lahir!!, jk!!, username!!, password!!, "user"
                )
                setViewData()
            }
        } else{
            Toast.makeText(requireContext(), data.message_response,
Toast.LENGTH_SHORT).show()
        }
        loading.alertDialogCancel()
        tempUser = UserModel()
    }

    private fun setFailureUpdateDiri(message: String) {
        Toast.makeText(requireContext(), message, Toast.LENGTH_SHORT).show()
        loading.alertDialogCancel()
        tempUser = UserModel()
    }

    private fun showDatePicker(view: AlertDialogAkunBinding, tanggal: String) {
        var vTahun = 2025
        var vBulan = 1
        var vTanggal = 1
        try{
            val listTanggal = tanggal.split("-")
            vTahun = listTanggal[0].trim().toInt()
            vBulan = listTanggal[1].trim().toInt()-1 // 0=jan, 1= feb, dst
            vTanggal = listTanggal[2].trim().toInt()
        } catch (ex: Exception){
            Log.d("TAG", "showDatePicker: ${ex.message}")
        }

        val calendar = Calendar.getInstance(TimeZone.getTimeZone("UTC"))
        calendar.set(vTahun, vBulan, vTanggal) // Tahun, Bulan, Tanggal

        val selectedDateInMillis = calendar.timeInMillis

        val datePicker = MaterialDatePicker.Builder.datePicker()
            .setTitleText("Pilih Tanggal Lahir")
            .setSelection(selectedDateInMillis)
```

```
            .build()

        datePicker.addOnPositiveButtonClickListener { selection ->
            selectedDate = Date(selection)
            val dateFormat = SimpleDateFormat("dd-MM-yyyy", Locale.getDefault())
            view.etEditTanggalLahir.text = dateFormat.format(selectedDate!!)
        }

        datePicker.show(childFragmentManager, "DATE_PICKER")
    }
}
```

Lampiran 10. Kode Program SplashScreenActivity.kt

```
package com.abcd.pemantauankesehatananak.ui.activity.splash_screen

import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.abcd.pemantauankesehatananak.R
import com.abcd.pemantauankesehatananak.ui.activity.login.LoginActivity
import com.abcd.pemantauankesehatananak.ui.activity.user.main.MainActivity
import com.abcd.pemantauankesehatananak.utils.SharedPreferencesLogin

class SplashScreenActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_splash_screen)

        val sharedPreferencesLogin =
SharedPreferencesLogin(this@SplashScreenActivity)

        Handler(Looper.getMainLooper()).postDelayed({
            if(sharedPreferencesLogin.getIdUser().toString() != ""){
                if(sharedPreferencesLogin.getSebagai() == "user"){
                    // To User
                    startActivity(Intent(this@SplashScreenActivity,
MainActivity::class.java))
                    finish()
                } else{
                    startActivity(Intent(this@SplashScreenActivity,
LoginActivity::class.java))
                    finish()
                }
            }
            else{
                startActivity(Intent(this@SplashScreenActivity,
LoginActivity::class.java))
                finish()
            }
        }, 3000)
    }
}
```

PEMERINTAH KOTA PAREPARE
DINAS KESEHATAN
UPTD PUSKESMAS LEMOE
Alamat: Jlan.Jend.M.Yusuf Kel.Lemoe Kec.Bacukiki
Email : puskesmaslemoe01@gmail.com Kode Pos : 91125

Parepare, 27 Agustus 2025

Nomor : 000 / /Puskesmas Lemoe
Lampiran : 1 (Satu)
Perihal : Hasil pemeriksaan

Yang bertanda tangan di bawah ini menerangkan bahwa pada hari ini telah dilakukan pemeriksaan kesehatan dan imunisasi terhadap anak:

Identitas Anak

1. Nama : **Cahaya Medna**
2. Umur : **7 bulan**
3. Jenis Kelamin : **Perempuan**
4. Alamat : Jl. Melati No. 12, Parepare

Hasil Pemeriksaan

1. Berat Badan : **8,0 kg**
2. Tinggi Badan : **68 cm**
3. Suhu tubuh : 36,8 °C
4. Keadaan umum : Baik, aktif, responsif
5. Tidak ada tanda infeksi akut

Jenis Imunisasi	Usia Pemberian	Tanggal Pemberian	Keterangan
Hepatitis B 0	0 bulan	10-11-2024	Lengkap
BCG	1 bulan	15-12-2024	Lengkap
Polio 1	2 bulan	15-01-2025	Lengkap
DPT-HB-Hib 1	2 bulan	15-01-2025	Lengkap
Polio 2	3 bulan	15-02-2025	Lengkap
DPT-HB-Hib 2	3 bulan	15-02-2025	Lengkap
Polio 3	4 bulan	15-03-2025	Lengkap
DPT-HB-Hib 3	4 bulan	15-03-2025	Lengkap

Kesimpulan Pemeriksaan

1. Status imunisasi : **Lengkap sesuai usia (7 bulan)**
2. Anak dalam keadaan sehat, pertumbuhan sesuai usia
3. Jadwal imunisasi berikutnya : **Polio 4 & Campak-Rubella (MR 1) pada usia 9 bulan**

Demikian surat hasil pemeriksaan ini dibuat untuk dapat dipergunakan sebagaimana mestinya atas perhatian dan kerjasama Bapak/Ibu diucapkan terima kasih.

Salam Sejawat
13 Agustus 2025

UNIT PUJI ASTUTI, S. Ked

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

HASISWA	: NURHIDAYAH	Pembimbing I : Marlina, S.Kom., M.Kom.
	: 218280067	Pembimbing II : Mughaffir Yunus, S.Kom., M.Kom
ul Skripsi	: APLIKASI PEMANTAUAN KESEHATAN ANAK PADA PUSKESMAS LEMOE BERBASIS ANDROID	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
nsultasi 1 ari Referensi yg 5 thn terakhir		Konsultasi 1 Ikuti Panduan Penulisan	
nsultasi 2 Perbaiki Daftar Isi Perbaiki Daftar pustaka		Konsultasi 2 Ukuran kerangka pikir Dan Daftar pustaka	
nsultasi 3 2		Konsultasi 3 Acc Seminar Proposal	
nsultasi 4 Acc Ujian proposal		Konsultasi 4	
nsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

HASIL

AHASISWA : Nurhidayah	Pembimbing I : Marlina, S.Kom.,M.Kom.
IM : 218 280 067	Pembimbing II : Mughaffir Yunus, S.T.,M.T
udul Skripsi : Aplikasi Pemantauan Kesehatan Anak Pada Puskesmas Lemoe Berbasis Android	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
nsultasi 1		Konsultasi 1	
		Perbaiki Pendahuluan Penulisan	
nsultasi 2		Konsultasi 2	
		Tambahkan Data pada	
nsultasi 3		Konsultasi 3	
Acc ujian hasil		Acc Seminar hasil	
nsultasi 4		Konsultasi 4	
		Perbaiki revisi penguji	
nsultasi 5		Konsultasi 5	
Acc Ujian Tutup		Acc Ujian Tutup	

Lanjut ke halaman sebelah...

hatian :

Mahasiswa wajib konsultasi minimal 5 kali
Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik