LAMPIRAN

- ***Controller***

```
package
com.example.kuisrambulalulintas.
ui.activities

import android.content.Intent
import android.os.Bundle
import android.widget.Toast
import
androidx.appcompat.app.AppCompat
Activity
import
com.example.kuisrambulalulintas.
R
import
com.example.kuisrambulalulintas.
databinding.ActivityGetNameBindi
ng
import
com.example.kuisrambulalulintas.
utils.Constants

class GetNameActivity :
AppCompatActivity() {

    private lateinit var
binding: ActivityGetNameBinding

    override fun
onCreate(savedInstanceState:
Bundle?) {

super.onCreate(savedInstanceStat
e)
        binding =
ActivityGetNameBinding.inflate(l
ayoutInflater)

setContentView(binding.root)

        val idLevel =
intent.getIntExtra("idLevel", 0)
        val question =
intent.getIntExtra("question",
0)
        val level =
intent.getStringExtra("level")
        val desc =
intent.getStringExtra("desc")

        binding.tvLevel.text =
level
        binding.tvDesLevel.text
= desc

        if (idLevel == 1) {

binding.clBackground.setBackgrou
ndResource(R.drawable.bg_button)

binding.btnStart.setBackgroundRe
source(R.drawable.bg_button)
        } else if (idLevel == 2)
{

binding.clBackground.setBackgrou
ndResource(R.drawable.bg_level2)

binding.btnStart.setBackgroundRe
source(R.drawable.bg_level2)
        } else if (idLevel == 3)
{

binding.clBackground.setBackgrou
ndResource(R.drawable.bg_level3)

binding.btnStart.setBackgroundRe
source(R.drawable.bg_level3)
        } else if (idLevel == 4)
{

binding.clBackground.setBackgrou
ndResource(R.drawable.bg_level4)

binding.btnStart.setBackgroundRe
source(R.drawable.bg_level4)
        } else {

binding.clBackground.setBackgrou
ndResource(R.drawable.bg_level5)

binding.btnStart.setBackgroundRe
source(R.drawable.bg_level5)
        }


binding.btnStart.setOnClickListe
ner {
            if
(binding.etName.text!!.isEmpty()
) {

Toast.makeText(this, "Please,
enter your name",
Toast.LENGTH_LONG).show()
            } else {
                val    intent    =
Intent(this,
QuestionActivity::class.java)

intent.putExtra(Constants.USER_N
AME,
binding.etName.text.toString())

intent.putExtra("question",
question)

startActivity(intent)
//                finish()
            }
        }
    }
}

package
com.example.kuisrambulalulintas.
ui.activities

import
android.annotation.SuppressLint
import android.content.Intent
import android.graphics.Color
import android.graphics.Typeface
import
androidx.appcompat.app.AppCompat
Activity
import android.os.Bundle
import android.os.CountDownTimer
import android.view.View
import android.widget.*
import
android.widget.SeekBar.OnSeekBar
ChangeListener
import
androidx.core.content.ContextCom
pat
import
com.example.kuisrambulalulintas.
R
import
com.example.kuisrambulalulintas.
databinding.ActivityKuisBinding
import
com.example.kuisrambulalulintas.
model.Soal
import
com.example.kuisrambulalulintas.
utils.Constants
```

```
import
com.google.firebase.firestore.Fi
rebaseFirestore

class       KuisActivity       :
AppCompatActivity(),OnSeekBarCha
ngeListener {

    private lateinit var binding
: ActivityKuisBinding
    private     var     userName:
String? = null
    private lateinit  var  timer:
CountDownTimer

    private   val   questionsList:
ArrayList<Soal>                 =
Constants.getSoals()
    private                    var
currentQuestionIndex = 0;
    private                    var
selectedAlternativeIndex = -1;
    private  var  isAnswerChecked
= false;
    private var totalScore = 0;
    private var kesalahan = 0;
    private  val  alternativesIds
= arrayOf(
        R.id.optionOne,
        R.id.optionTwo,
        R.id.optionThree,
        R.id.optionFour
    )

    private  var  tvAlternatives:
ArrayList<TextView>? = null

    override                   fun
onCreate(savedInstanceState:
Bundle?) {

super.onCreate(savedInstanceStat
e)
        binding                 =
ActivityKuisBinding.inflate(layo
utInflater)

setContentView(binding.root)

        binding.sbTime.max = 10
        binding.sbTime.progress
= 10
```

```
binding.sbTime.setOnSeekBarChang
eListener(this)


        timer = object :
CountDownTimer((binding.sbTime.p
rogress * 1000).toLong(),1000){
            override fun
onTick(millisUntilFinished:
Long) {

binding.tvTimer.text =
"00:0${millisUntilFinished /
1000}"
                val longValue =
millisUntilFinished / 1000

updateSeekbar(longValue.toInt())
            }

            override fun
onFinish() {
                if
(currentQuestionIndex <
questionsList.size - 1) {

currentQuestionIndex++

updateQuestion()
                } else {
                    val intent =
Intent(this@KuisActivity,
ResultActivity::class.java)

intent.putExtra(Constants.USER_N
AME, userName)

intent.putExtra(Constants.TOTAL_
SoalS, questionsList.size)

intent.putExtra(Constants.SCORE,
totalScore)

startActivity(intent)
                    finish()
                }
            }

        }
        timer.start()

        tvAlternatives =
arrayListOf(
```

```
            binding.optionOne,
            binding.optionTwo,
            binding.optionThree,
            binding.optionFour,
        )

        updateQuestion()


binding.btnSubmit.setOnClickList
ener {
            if
(!isAnswerChecked) {
                val
anyAnswerIsChecked =
selectedAlternativeIndex != -1
                if
(!anyAnswerIsChecked) {

Toast.makeText(this, "Please,
pilih jawaban anda",
Toast.LENGTH_SHORT).show()
                } else {
                    val
currentQuestion =
questionsList[currentQuestionInd
ex]
                    if (

selectedAlternativeIndex ==
currentQuestion.jawabanBenarInde
x
                    ) {

answerView(tvAlternatives!![sele
ctedAlternativeIndex],

R.drawable.correct_option_border
_bg
                        )

totalScore++

timer.cancel()
                    } else {

kesalahan++

timer.cancel()

answerView(tvAlternatives!![sele
ctedAlternativeIndex],
R.drawable.wrong_option_border_b
g
                        )

answerView(tvAlternatives!![curr
entQuestion.jawabanBenarIndex],

R.drawable.correct_option_border
_bg
                        )

                        if
(kesalahan == 1){

binding.ivKesempatan1.visibility
= View.GONE
                        }    else
if (kesalahan == 2){

binding.ivKesempatan2.visibility
= View.GONE
                        }    else
if (kesalahan == 3){

binding.ivKesempatan3.visibility
= View.GONE
                        }else {
                            val
intent       =       Intent(this,
ResultActivity::class.java)

intent.putExtra(Constants.USER_N
AME, userName)

intent.putExtra(Constants.TOTAL_
SoalS, questionsList.size)

intent.putExtra(Constants.SCORE,
totalScore)

startActivity(intent)

finish()
                        }
                    }


isAnswerChecked = true

binding.btnSubmit.text    =    if
(currentQuestionIndex         ==
questionsList.size      -      1)
```

```
"SELESAI"       else       "SOAL
SELANJUTNYA"

selectedAlternativeIndex = -1
                }
            } else {
                if
(currentQuestionIndex          <
questionsList.size - 1) {

currentQuestionIndex++

updateQuestion()
                } else {
                    val  intent  =
Intent(this,
ResultActivity::class.java)

intent.putExtra(Constants.USER_N
AME, userName)

intent.putExtra(Constants.TOTAL_
SoalS, questionsList.size)

intent.putExtra(Constants.SCORE,
totalScore)

startActivity(intent)
                    finish()
                }

                isAnswerChecked
= false
            }
        }

        tvAlternatives?.let {
            for  (optionIndex  in
it.indices) {

it[optionIndex].let {

it.setOnClickListener{
                        if
(!isAnswerChecked) {

selectedAlternativeView(it     as
TextView, optionIndex)
                        }
                    }
                }
            }
        }
    }
    override                  fun
onProgressChanged(p0:   SeekBar?,
progress: Int, p2: Boolean) {
        updateSeekbar(progress)
    }

    override                  fun
onStartTrackingTouch(p0:
SeekBar?) {

    }

    override                  fun
onStopTrackingTouch(p0:
SeekBar?) {

    }

    private  fun  updateQuestion()
{

defaultAlternativesView()
        timer.start()

        // Render Question Text
        binding.tvQuestion.text
=
questionsList[currentQuestionInd
ex].teks_Soal
        // Render Question Image

binding.ivImage.setImageResource
(questionsList[currentQuestionIn
dex].gambar)
        // progressBar

binding.progressBar.progress    =
currentQuestionIndex + 1
        // Text of progress bar
        binding.tvProgress.text
=    "${currentQuestionIndex    +
1}/${questionsList.size}"

        for  (alternativeIndex  in
questionsList[currentQuestionInd
ex].alternative.indices) {

tvAlternatives!![alternativeInde
x].text                         =
questionsList[currentQuestionInd
ex].alternative[alternativeIndex
]
        }
```

```
        binding.btnSubmit.text  =
if    (currentQuestionIndex    ==
questionsList.size      -      1)
"SELESAI" else "JAWAB"
    }

    private                   fun
defaultAlternativesView() {
        for   (alternativeTv   in
tvAlternatives!!) {

alternativeTv.typeface          =
Typeface.DEFAULT

alternativeTv.setTextColor(Color
.parseColor("#7A8089"))

alternativeTv.background        =
ContextCompat.getDrawable(

this@KuisActivity,

R.drawable.default_option_border
_bg
            )
        }
    }

    private                   fun
selectedAlternativeView(option:
TextView, index: Int) {

defaultAlternativesView()
        selectedAlternativeIndex
= index

        option.setTextColor(

Color.parseColor("#363A43")
        )

option.setTypeface(option.typefa
ce, Typeface.BOLD)
        option.background       =
ContextCompat.getDrawable(
            this@KuisActivity,

R.drawable.selected_option_borde
r_bg
        )
    }
    private fun answerView(view:
TextView, drawableId: Int) {
        view.background =
ContextCompat.getDrawable(
            this@KuisActivity,
            drawableId
        )

tvAlternatives!![selectedAlterna
tiveIndex].setTextColor(

Color.parseColor("#FFFFFF")
        )
    }

    @SuppressLint("SetTextI18n")
    private fun
updateSeekbar(progress: Int){
        val minute : Int =
progress / 60
        val seconds : Int =
progress % 60
        var secondsFinal = ""
        if (seconds <= 9){
            secondsFinal = "0"
+seconds
        } else {
            secondsFinal = "" +
seconds
        }

        binding.sbTime.progress
= progress
        binding.tvTimer.text =
"$minute:$secondsFinal"
    }


}
```

```
package
com.example.kuisrambulalulintas.
ui.activities

import android.content.Intent
import
androidx.appcompat.app.AppCompat
Activity
import android.os.Bundle
import
com.example.kuisrambulalulintas.
R
import
com.example.kuisrambulalulintas.
databinding.ActivityLevelBinding

class LevelActivity :
AppCompatActivity() {
    private lateinit var binding
: ActivityLevelBinding

    override fun
onCreate(savedInstanceState:
Bundle?) {

super.onCreate(savedInstanceStat
e)
        binding =
ActivityLevelBinding.inflate(lay
outInflater)

setContentView(binding.root)



binding.clLevel1.setOnClickListe
ner {
            val idLevel1 = 1
            val question1 = 3
            val tvLevel = "Level
1"
            val tvDescLevel =
"Newbie"
            val intent =
Intent(this,GetNameActivity::cla
ss.java)

intent.putExtra("idLevel",idLeve
l1)

intent.putExtra("question",quest
ion1)

intent.putExtra("level",tvLevel)

intent.putExtra("desc",tvDescLev
el)

startActivity(intent)
        }


binding.clLevel2.setOnClickListe
ner {
            val idLevel2 = 2
            val question2 = 5
            val tvLevel = "Level
2"
            val   tvDescLevel   =
"Continuing"
            val     intent     =
Intent(this,GetNameActivity::cla
ss.java)

intent.putExtra("idLevel",idLeve
l2)

intent.putExtra("question",quest
ion2)

intent.putExtra("level",tvLevel)

intent.putExtra("desc",tvDescLev
el)

startActivity(intent)
        }


binding.btnLevel3.setOnClickList
ener {
            val idLevel3 = 3
            val question3 = 7
            val tvLevel = "Level
3"
            val   tvDescLevel   =
"Medium"
            val     intent     =
Intent(this,GetNameActivity::cla
ss.java)

intent.putExtra("idLevel",idLeve
l3)

intent.putExtra("question",quest
ion3)

intent.putExtra("level",tvLevel)

intent.putExtra("desc",tvDescLev
el)

startActivity(intent)
        }


binding.btnLevel4.setOnClickList
ener {
            val idLevel4 = 4
            val question4 = 9
```

```
            val tvLevel = "Level
4"
            val   tvDescLevel   =
"Pro"
            val     intent     =
Intent(this,GetNameActivity::cla
ss.java)

intent.putExtra("idLevel",idLeve
l4)

intent.putExtra("question",quest
ion4)

intent.putExtra("level",tvLevel)

intent.putExtra("desc",tvDescLev
el)

startActivity(intent)
        }


binding.btnLevel5.setOnClickList
ener {
            val idLevel5 = 5
            val question5 = 11
            val tvLevel = "Level
5"
            val   tvDescLevel   =
"Legend"
            val     intent     =
Intent(this,GetNameActivity::cla
ss.java)

intent.putExtra("idLevel",idLeve
l5)

intent.putExtra("question",quest
ion5)

intent.putExtra("level",tvLevel)

intent.putExtra("desc",tvDescLev
el)

startActivity(intent)
        }

    }
}
```

```
package
com.example.kuisrambulalulintas.
ui.activities

import
androidx.appcompat.app.AppCompat
Activity
import android.os.Bundle
import
com.example.kuisrambulalulintas.
R

class MateriActivity :
AppCompatActivity() {
    override fun
onCreate(savedInstanceState:
Bundle?) {

super.onCreate(savedInstanceStat
e)

setContentView(R.layout.activity
_materi)
    }
}

package
com.example.kuisrambulalulintas.
ui.activities

import
android.annotation.SuppressLint
import
android.content.DialogInterface
import android.content.Intent
import android.graphics.Color
import android.graphics.Typeface
import android.os.Bundle
import android.os.CountDownTimer
import android.util.Log
import android.view.View
import android.widget.SeekBar
import android.widget.TextView
import android.widget.Toast
import
androidx.activity.viewModels
import
androidx.appcompat.app.AlertDial
og
import
androidx.appcompat.app.AppCompat
Activity
import
androidx.core.content.ContextCom
pat
import com.bumptech.glide.Glide
import
com.example.kuisrambulalulintas.
R
import
com.example.kuisrambulalulintas.
databinding.ActivityQuestionBind
ing
import
com.example.kuisrambulalulintas.
model.DataSoal
import
com.example.kuisrambulalulintas.
utils.Constants
import
com.example.kuisrambulalulintas.
utils.Resource
import
com.example.kuisrambulalulintas.
viewmodel.MainViewModel
import
com.google.firebase.firestore.Fi
rebaseFirestore
import
dagger.hilt.android.AndroidEntry
Point


@AndroidEntryPoint
class QuestionActivity :
AppCompatActivity(),
SeekBar.OnSeekBarChangeListener
{
    private lateinit var binding
: ActivityQuestionBinding

    private val viewModel :
MainViewModel by viewModels()

    private var userName:
String? = null
    private var question: Int? =
null
    private lateinit var timer:
CountDownTimer

    private var questionsList:
ArrayList<DataSoal> =
ArrayList()
```

```
    private var
currentQuestionIndex = 0
    private var
selectedAlternativeIndex = -1
    private var isAnswerChecked
= false;
    private var totalScore = 0;
    private var kesalahan = 0;
    private val alternativesIds
= arrayOf(
        R.id.optionOne,
        R.id.optionTwo,
        R.id.optionThree,
        R.id.optionFour
    )

    private var tvAlternatives:
ArrayList<TextView>? = null


    override fun
onCreate(savedInstanceState:
Bundle?) {

super.onCreate(savedInstanceStat
e)
        binding =
ActivityQuestionBinding.inflate(
layoutInflater)

setContentView(binding.root)

        userName =
intent.getStringExtra(Constants.
USER_NAME)
        question =
intent.getIntExtra("question",0)

        val db =
FirebaseFirestore.getInstance()
        val soalRef =
db.collection("kuis")

        getDataKuis()

        binding.sbTime.max = 10
        binding.sbTime.progress
= 10

binding.sbTime.setOnSeekBarChang
eListener(this)
```

```
        timer = object :
CountDownTimer((binding.sbTime.p
rogress * 1000).toLong(),1000){
            override fun
onTick(millisUntilFinished:
Long) {

binding.tvTimer.text =
"00:0${millisUntilFinished /
1000}"
                val longValue =
millisUntilFinished / 1000

updateSeekbar(longValue.toInt())
            }

            override fun
onFinish() {
                if
(currentQuestionIndex <
question!! - 1) {

currentQuestionIndex++

updateQuestion()
                } else {
                    val intent =
Intent(this@QuestionActivity,
ResultActivity::class.java)

intent.putExtra(Constants.USER_N
AME, userName)

intent.putExtra(Constants.TOTAL_
SoalS, question!!)

intent.putExtra(Constants.SCORE,
totalScore)

startActivity(intent)
                    finish()
                }
            }

        }
        timer.start()

        tvAlternatives =
arrayListOf(
            binding.optionOne,
            binding.optionTwo,
            binding.optionThree,
            binding.optionFour,
        )
        //updateQuestion()


binding.btnSubmit.setOnClickList
ener {
            if
(!isAnswerChecked) {
                val
anyAnswerIsChecked            =
selectedAlternativeIndex != -1
                if
(!anyAnswerIsChecked) {

Toast.makeText(this,     "Please,
pilih       jawaban       anda",
Toast.LENGTH_SHORT).show()
                } else {
                    val
currentQuestion               =
questionsList[currentQuestionInd
ex]
                    if (

selectedAlternativeIndex       ==
currentQuestion.jawaban
                    ) {

answerView(tvAlternatives!![sele
ctedAlternativeIndex],

R.drawable.correct_option_border
_bg
                        )

totalScore++

timer.cancel()
                    } else {

kesalahan++

timer.cancel()

answerView(tvAlternatives!![sele
ctedAlternativeIndex],

R.drawable.wrong_option_border_b
g
                        )

answerView(tvAlternatives!![curr
entQuestion.jawaban!!],
```

```
R.drawable.correct_option_border
_bg
                        )

                        if
(kesalahan == 1){

binding.ivKesempatan1.visibility
= View.GONE
                        }    else
if (kesalahan == 2){

binding.ivKesempatan2.visibility
= View.GONE
                        }    else
if (kesalahan == 3){

binding.ivKesempatan3.visibility
= View.GONE
                        }else {
                            val
intent       =       Intent(this,
ResultActivity::class.java)

intent.putExtra(Constants.USER_N
AME, userName)

intent.putExtra(Constants.TOTAL_
SoalS, question!!)

intent.putExtra(Constants.SCORE,
totalScore)

startActivity(intent)

finish()
                        }
                    }


isAnswerChecked = true

binding.btnSubmit.text    =    if
(currentQuestionIndex          ==
question!!  -  1)  "SELESAI"  else
"SOAL SELANJUTNYA"

selectedAlternativeIndex = -1
                }
            } else {
                if
(currentQuestionIndex          <
question!! - 1) {
```

```
currentQuestionIndex++

updateQuestion()
                } else {
                    val intent =
Intent(this,
ResultActivity::class.java)

intent.putExtra(Constants.USER_N
AME, userName)

intent.putExtra(Constants.TOTAL_
SoalS, question!!)

intent.putExtra(Constants.SCORE,
totalScore)

startActivity(intent)
                    finish()
                }

                isAnswerChecked
= false
            }
        }

        tvAlternatives?.let {
            for (optionIndex in
it.indices) {

it[optionIndex].let {

it.setOnClickListener{
                        if
(!isAnswerChecked) {

selectedAlternativeView(it as
TextView, optionIndex)
                        }
                    }
                }
            }
        }

    }

    private fun getDataKuis() {
        viewModel.getKuis()

viewModel.soal.observe(this){
response ->
            when(response){
```

```
                is
Resource.Success -> {

questionsList = response.data as
ArrayList<DataSoal>

questionsList.shuffle()

updateQuestion()


Log.d("DataResponse","$questions
List")
                }

                is
Resource.Loading -> {

                }
                is
Resource.Error -> {

                }
                else -> {

                }
            }

        }
    }


    private fun updateQuestion()
{

defaultAlternativesView()
        timer.start()

        // Render Question Text
        binding.tvQuestion.text
= "Apa arti gambar lalu lintas
di bawah ini"
        // Render Question Image

Glide.with(applicationContext)

.load(questionsList[currentQuest
ionIndex].image)

.into(binding.ivImage)
        // progressBar

//binding.ivImage.setIma(questio
nsList[currentQuestionIndex].ima
ge)
        binding.progressBar.max
= question!!

binding.progressBar.progress =
currentQuestionIndex + 1
        // Text of progress bar
        binding.tvProgress.text
= "${currentQuestionIndex +
1}/${question!!}"

        for (alternativeIndex in
questionsList[currentQuestionInd
ex].pilihan!!.indices) {

tvAlternatives!![alternativeInde
x].text =
questionsList[currentQuestionInd
ex].pilihan?.get(alternativeInde
x)!!
        }

        binding.btnSubmit.text =
if (currentQuestionIndex ==
question!! - 1) "SELESAI" else
"JAWAB"
    }

    private fun
defaultAlternativesView() {
        for (alternativeTv in
tvAlternatives!!) {

alternativeTv.typeface =
Typeface.DEFAULT

alternativeTv.setTextColor(Color
.parseColor("#7A8089"))

alternativeTv.background =
ContextCompat.getDrawable(

this@QuestionActivity,

R.drawable.default_option_border
_bg
            )
        }
    }

    private fun
selectedAlternativeView(option:
TextView, index: Int) {

defaultAlternativesView()
        selectedAlternativeIndex
= index

        option.setTextColor(

Color.parseColor("#363A43")
        )

option.setTypeface(option.typefa
ce, Typeface.BOLD)
        option.background =
ContextCompat.getDrawable(

this@QuestionActivity,

R.drawable.selected_option_borde
r_bg
        )
    }

    private fun answerView(view:
TextView, drawableId: Int) {
        view.background =
ContextCompat.getDrawable(

this@QuestionActivity,
            drawableId
        )

tvAlternatives!![selectedAlterna
tiveIndex].setTextColor(

Color.parseColor("#FFFFFF")
        )
    }

    @SuppressLint("SetTextI18n")
    private fun
updateSeekbar(progress: Int){
        val minute : Int =
progress / 60
        val seconds : Int =
progress % 60
        var secondsFinal = ""
        if (seconds <= 9){
            secondsFinal = "0"
+seconds
        } else {
            secondsFinal = "" +
seconds
        }
        binding.sbTime.progress
= progress
        binding.tvTimer.text =
"$minute:$secondsFinal"
    }

    override fun
onProgressChanged(p0: SeekBar?,
progress: Int, p2: Boolean) {
        updateSeekbar(progress)
    }

    override fun
onStartTrackingTouch(p0:
SeekBar?) {
    }

    override fun
onStopTrackingTouch(p0:
SeekBar?) {
    }

    override fun onBackPressed()
{
        showAlertDialog()

    }

    private fun
showAlertDialog() {
        val dialogBuilder =
AlertDialog.Builder(this)


dialogBuilder.setMessage("Apakah
anda yakin akan keluar dari
permainan ini ?")

.setCancelable(false)

.setPositiveButton("Ya",
DialogInterface.OnClickListener
{ _, _ ->
                finish()
            })

.setNegativeButton("keluar",
DialogInterface.OnClickListener
{ dialogInterface, i ->

dialogInterface.cancel()
            })
```

```
        val alert =
dialogBuilder.create()
        alert.setTitle("Keluar")
        alert.show()
    }
}

package
com.example.kuisrambulalulintas.
ui.activities

import android.content.Intent
import
androidx.appcompat.app.AppCompat
Activity
import android.os.Bundle
import
com.example.kuisrambulalulintas.
MainActivity
import
com.example.kuisrambulalulintas.
databinding.ActivityResultBindin
g
import
com.example.kuisrambulalulintas.
utils.Constants

class ResultActivity :
AppCompatActivity() {

    private lateinit var binding
: ActivityResultBinding
    override fun
onCreate(savedInstanceState:
Bundle?) {

super.onCreate(savedInstanceStat
e)
        binding =
ActivityResultBinding.inflate(la
youtInflater)

setContentView(binding.root)

        val userName =
intent.getStringExtra(Constants.
USER_NAME)
        val totalQuestions =
intent.getIntExtra(Constants.TOT
AL_SoalS, 0)
        val score =
intent.getIntExtra(Constants.SCO
RE, 0)
```

```
binding.congratulationsTv.text  =
"Hai, $userName!"
        binding.scoreTv.text    =
"Kamu  jawab  benar  $score  dari
$totalQuestions soal"
        binding.tvHasil.text    =
"$score/$totalQuestions"

binding.progressBar2.progress   =
score
        binding.progressBar2.max
= totalQuestions

binding.btnRestart.setOnClickLis
tener{
            val      intent      =
Intent(this,
QuestionActivity::class.java)

intent.putExtra("question",total
Questions)

intent.putExtra(Constants.USER_N
AME,userName)

startActivity(intent)
            finish()
        }


binding.btnMenu.setOnClickListen
er{
            val      intent      =
Intent(this,
MainActivity::class.java)

startActivity(intent)
            finish()
        }
    }
}

package
com.example.kuisrambulalulintas.
ui.fragments

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.navigation.NavArgs
import
androidx.navigation.fragment.nav
Args
import
com.example.kuisrambulalulintas.
R
import
com.example.kuisrambulalulintas.
databinding.FragmentMateriBindin
g
import
com.example.kuisrambulalulintas.
model.Gambar
import
com.example.kuisrambulalulintas.
utils.Constants

class      MateriFragment      :
Fragment() {

    private     lateinit     var
binding: FragmentMateriBinding

    private     val     args     :
MateriFragmentArgs by navArgs()

    private                  val
rambuLaranganList              =
Constants.getRambuLarangan()
    private                  val
rambuPeringatanList            =
Constants.getRambuPeringatan()
    private                  val
rambuPerintahList              =
Constants.getRambuPerintah()
    private                  val
rambuPetunjukList              =
Constants.getRambuLPetunjuk()
    private   var   gambarList   :
ArrayList<Gambar>? = null
    private var selectIndex = 0

    override fun onCreateView(
        inflater:
LayoutInflater,       container:
ViewGroup?,
        savedInstanceState:
Bundle?
    ): View? {
        // Inflate the layout
for this fragment
        binding =
FragmentMateriBinding.inflate(la
youtInflater, container, false)


        val currentGambar =
rambuLaranganList[selectIndex]

        if (args.rambu == 1){
            gambarList =
rambuLaranganList

binding.clBackground.setBackgrou
ndResource(R.drawable.bg_laranga
n)
        } else if (args.rambu ==
2){
            gambarList =
rambuPeringatanList

binding.clBackground.setBackgrou
ndResource(R.drawable.bg_peringa
tan)
        }
        else if (args.rambu ==
3){
            gambarList =
rambuPerintahList

binding.clBackground.setBackgrou
ndResource(R.drawable.bg_perinta
h)
        } else {
            gambarList =
rambuPetunjukList

binding.clBackground.setBackgrou
ndResource(R.drawable.bg_petunju
k)
        }



updateGambar(selectIndex)


binding.btnNext.setOnClickListen
er {
            if (selectIndex <
gambarList!!.size - 1) {
                selectIndex++

updateGambar(selectIndex)


Log.d("selectIndex",
"$selectIndex")

Log.d("selectIndex",
"${gambarList!!.size - 1}
${gambarList!!.size}")
            }

        }


binding.btnPrev.setOnClickListen
er {
            if (selectIndex <
gambarList!!.size) {
                    selectIndex-
-

updateGambar(selectIndex)
            }
        }

        return binding.root
    }

    private fun
updateGambar(selectIndex: Int) {

        if (selectIndex == 0) {

binding.btnPrev.visibility =
View.GONE
        } else if (selectIndex
== gambarList!!.size - 1) {

binding.btnNext.visibility =
View.GONE
        } else {

binding.btnNext.visibility =
View.VISIBLE

binding.btnPrev.visibility =
View.VISIBLE
        }

binding.ivRambu.setImageResource
(gambarList!![selectIndex].gamba
r)
```

```
binding.tvKeterangan.text       =
gambarList!![selectIndex].ketera
ngan
    }
}
package
com.example.kuisrambulalulintas.
ui.fragments

import android.os.Bundle
import
androidx.fragment.app.Fragment
import
android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import
androidx.navigation.fragment.fin
dNavController
import
com.example.kuisrambulalulintas.
R
import
com.example.kuisrambulalulintas.
databinding.FragmentMenuMateriBi
nding

class     MenuMateriFragment     :
Fragment() {

    private  lateinit  var  binding
: FragmentMenuMateriBinding

    override fun onCreateView(
        inflater:
LayoutInflater,        container:
ViewGroup?,
        savedInstanceState:
Bundle?
    ): View? {
        //  Inflate  the  layout
for this fragment
        binding                 =
FragmentMenuMateriBinding.inflat
e(layoutInflater,      container,
false)



binding.cvLarangan.setOnClickLis
tener {
            val      data      =
MenuMateriFragmentDirections.act
ionMenuMateriFragmentToMateriFra
gment(1)

findNavController().navigate(dat
a)
        }


binding.cvPeringatan.setOnClickL
istener {
            val      data      =
MenuMateriFragmentDirections.act
ionMenuMateriFragmentToMateriFra
gment(2)

findNavController().navigate(dat
a)
        }


binding.cvPerintah.setOnClickLis
tener {
            val      data      =
MenuMateriFragmentDirections.act
ionMenuMateriFragmentToMateriFra
gment(3)

findNavController().navigate(dat
a)
        }


binding.cvPetunjuk.setOnClickLis
tener {
            val      data      =
MenuMateriFragmentDirections.act
ionMenuMateriFragmentToMateriFra
gment(4)

findNavController().navigate(dat
a)
        }


        return binding.root
    }
}

package
com.example.kuisrambulalulintas.
utils
```

```
import
com.example.kuisrambulalulintas.
R
import
com.example.kuisrambulalulintas.
model.Gambar
import
com.example.kuisrambulalulintas.
model.Soal

object Constants {
    val   USER_NAME:   String   =
"user_name"
    val  TOTAL_SoalS:  String  =
"total_Soals"
    val SCORE: String = "score"

    fun              getSoals():
ArrayList<Soal> {
        val      SoalsList      =
ArrayList<Soal>()

        // 1
        val SoalOne = Soal(
            1,
            "Apa arti rambu lalu
lintas  dari  gambar  dibawah  ini
?",
            R.drawable.camera,

arrayListOf("Dilarang  berhenti",
"Dilarang      Putar      balik",
"Dilarang  memotret",  "Dilarang
merokok"),
            2,
        )
        SoalsList.add(SoalOne)

        // 2
        val SoalTwo = Soal(
            2,
            "Apa arti rambu lalu
lintas dari gambar dibawah ?",
            R.drawable.delapan,

arrayListOf("Dilarang  berhenti",
"Batas Maksimal kecepatan",
                "Dilarang
tidur", "Dilarang jalan"),
            1
        )
        SoalsList.add(SoalTwo)

        // 3
        val SoalThree = Soal(
            3,
            "Apa arti rambu lalu
lintas dari gambar dibawah ?",
            R.drawable.hatihati,
            arrayListOf("Hati-
hati", "Jalanan licin",
                "Dilarang
berhenti", "Dilarang parkir"),
            0
        )
        SoalsList.add(SoalThree)

        // 4
        val SoalFour = Soal(
            4,
            "Apa arti rambu lalu
lintas dari gambar dibawah ?",

R.drawable.larangan_motor,

arrayListOf("Dilarang   Kencang",
"Jalan licin",
                "Dilaran    Lewat
Kendaraaan",            "Dilarang
berhenti"),
            2
        )
        SoalsList.add(SoalFour)

        // 5
        val SoalFive = Soal(
            5,
            "Apa arti rambu lalu
lintas dari gambar dibawah ?",

R.drawable.larangan_parkir,

arrayListOf("Dilarang   Kencang",
"Jalan licin",
                "Dilaran    Lewat
Kendaraaan", "Dilarang Parkir"),
            3
        )
        SoalsList.add(SoalFive)

        // 6
        val SoalSix = Soal(
            6,
            "Apa arti rambu lalu
lintas dari gambar dibawah ?",
            R.drawable.licin,
arrayListOf("Dilarang Kencang",
"Jalan licin",
                "Dilaran Lewat
Kendaraaan", "Dilarang
berhenti"),
            1
        )

        SoalsList.add(SoalSix)

        // 7
        val SoalSeven = Soal(
            7,
            "Apa arti rambu lalu
lintas dari gambar dibawah ?",
            R.drawable.merokok,

arrayListOf("Dilarang Merokok",
"Jalan licin",
                "Dilaran Lewat
Kendaraaan", "Dilarang
berhenti"),
            0
        )
        SoalsList.add(SoalSeven)

        // 8
        val SoalEight = Soal(
            8,
            "Apa arti rambu lalu
lintas dari gambar dibawah ?",
            R.drawable.motor,

arrayListOf("Dilarang Kencang",
"Jalanan licin",
                "Dilarang
motor", "Dilarang berhenti"),
            2
        )
        SoalsList.add(SoalEight)

        // 9
        val SoalNine = Soal(
            9,
            "Apa arti rambu lalu
lintas dari gambar dibawah ?",
            R.drawable.pejalan,

arrayListOf("Dilarang Kencang",
"Jalanan licin",
                "Dilarang
motor", "Dilarang Jalan kaki"),
            3
        )
        SoalsList.add(SoalNine)

        // 10
        val SoalTen = Soal(
            10,
            "Apa arti rambu lalu
lintas dari gambar dibawah ?",

R.drawable.putarbalik,

arrayListOf("Dilarang Kencang",
"Dilarang Putar ke Kiri",
                "Dilarang
motor", "Dilarang berhenti"),
            1
        )
        SoalsList.add(SoalTen)

        return SoalsList
    }

    fun getRambuLarangan() :
ArrayList<Gambar>{
        val gambarList =
ArrayList<Gambar>()

        val g1 =
Gambar(1,"Dilarang
Parkir",R.drawable.larangan_park
ir)
        gambarList.add(g1)
        val g2 = Gambar(2,"Hati-
hati",R.drawable.hatihati)
        gambarList.add(g2)
        val g3 =
Gambar(3,"Jalanan
licin",R.drawable.licin)
        gambarList.add(g3)
        val g4 =
Gambar(4,"Dilarang Belok
Kiri",R.drawable.putarbalik)
        gambarList.add(g4)
        val g5 =
Gambar(5,"DilarangMerokok",R.dra
wable.merokok)
        gambarList.add(g5)
        val g6 =
Gambar(6,"Dilarang berjalan
terus, wajib berhenti sesaat dan
meneruskan perjalanan setelah
mendapat kepastian aman dari
lalu lintas arah
```

```
lainnya.",R.drawable.larangan6_b
erjalan_terus)
        gambarList.add(g6)
        val g7 =
Gambar(7,"Kendaraan dengan
muatan sumbu terberat (MST)
lebih besar dari 8 ton atau
ukuran lebar tidak melebihi
2.500 milimeter atau ukuran
panjang tidak melebihi 12.000
milimeter.",R.drawable.larangan7
_muatan_terberat)
        gambarList.add(g7)
        val g8 =
Gambar(8,"Dilarang berjalan
terus apabila mengakibatkan
rintangan, hambatan, gangguan
bagi lalulintas dari arah lain
yang wajib
didahulukan.",R.drawable.laranga
n8_berjalan_terus)
        gambarList.add(g8)
        val g9 =
Gambar(9,"Dilarang berjalan
seruas, wajib berhenti sesaat
sebelum bagian jalan tertentu
dan meneruskan perjalanan
setelah mendahulukan kendaraan
yang datang dari arah depan
secara
bersamaan.",R.drawable.larangan9
_berjalan_seruas)
        gambarList.add(g9)
        val g10 =
Gambar(10,"Batas akhir larangan
mendahului kendaraan
lain.",R.drawable.larangan10_bat
asakhir)
        gambarList.add(g10)
        val g11 =
Gambar(11,"Larangan berbalik
arah bagi kendaraan bermotor
maupun tidak
bermotor.",R.drawable.larangan11
_berbalik_arah)
        gambarList.add(g11)
        val g12 =
Gambar(12,"Larangan berbelok ke
kiri bagi kendaraan bermotor
maupun tidak bermotor untuk
masuk jalan simpangan atau
berpindah jalur yang searah
lalulintas.",R.drawable.larangan
12_belok_kiri)
        gambarList.add(g12)
        val g13 =
Gambar(13,"Larangan berhenti
sampai jarak 15 m dari tempat
pemasangan rambu menurut arah
lalulintas kecuali dinyatakan
lain dengan papan
tambahan.",R.drawable.larangan13
_berhenti15)
        gambarList.add(g13)
        val g14 =
Gambar(14,"Larangan berbelok ke
kanan bagi kendaraan bermotor
maupun tidak bermotor untuk
masuk jalan simpangan atau
berpindah jalur yang searah
lalulintas.",R.drawable.larangan
14_belok_kanan)
        gambarList.add(g14)
        val g15 =
Gambar(15,"Larangan kecepatan
kendaraan lebih dari 40 km
perjam (Rambu
Larangan).",R.drawable.larangan1
5_kecepatan40)
        gambarList.add(g15)
        val g16 =
Gambar(16,"Larangan masuk bagi
becak.",R.drawable.larangan16_be
cak)
        gambarList.add(g16)
        val g17 =
Gambar(17,"Larangan masuk bagi
bus.",R.drawable.larangan17_bus)
        gambarList.add(g17)
        val g18 =
Gambar(18,"Larangan masuk bagi
delman dan
sejenisnya.",R.drawable.larangan
18_delman)
        gambarList.add(g18)
        val g19 =
Gambar(19,"Larangan masuk bagi
gerobak dan
dokar.",R.drawable.larangan19_ge
robak)
        gambarList.add(g19)
        val g20 =
Gambar(20,"Larangan masuk bagi
gerobak dorong dan
sejenisnya.",R.drawable.larangan
20_gerobak_dorong)
        gambarList.add(g20)
        val g21 =
Gambar(21,"Larangan masuk bagi
gerobak pedati dan sejenisnya
(Rambu
Larangan).",R.drawable.larangan2
1_batasakhir2)
        gambarList.add(g21)
        val g22 =
Gambar(22,"Larangan masuk bagi
kendaraan bermotor dan
mobil.",R.drawable.larangan22_mo
bil_motor)
        gambarList.add(g22)
        val g23 =
Gambar(23,"Larangan masuk bagi
kendaraan bermotor dengan kereta
gandeng.",R.drawable.larangan23_
motor_kereta_gandeng)
        gambarList.add(g23)
        val g24 =
Gambar(24,"Larangan masuk bagi
kendaraan bermotor dengan kereta
tempel.",R.drawable.larangan24_k
ereta_tempel)
        gambarList.add(g24)
        val g25 =
Gambar(25,"Larangan masuk bagi
kendaraan bermotor roda empat
atau
lebih.",R.drawable.larangan25_ke
ndaraan_roda_empat)
        gambarList.add(g25)
        val g26 =
Gambar(26,"Larangan masuk bagi
kendaraan bermotor roda
tiga.",R.drawable.larangan26_rod
atiga)
        gambarList.add(g26)
        val g27 =
Gambar(27,"Larangan masuk bagi
kendaraan yang seluruh berat
termasuk muatannya lebih dari 7
ton.",R.drawable.larangan27_7ton
)
        gambarList.add(g27)
        val g28 =
Gambar(28,"Larangan masuk bagi
mobil
barang.",R.drawable.larangan28_m
obil_barang)
        gambarList.add(g28)
        val g29 =
Gambar(29,"Larangan masuk bagi
pejalan
kaki.",R.drawable.larangan29_pej
alan_kaki)
        gambarList.add(g29)
        val g30 =
Gambar(30,"Batas kecepatan
maksimum 40 km
jam.",R.drawable.larangan30_bata
s_kecepatan)
        gambarList.add(g30)
        val g31 =
Gambar(31,"Larangan masuk bagi
semua kendaraan bermotor maupun
tidak bermotor dari kedua
arah.",R.drawable.larangan31_sem
uakendaraan_semuarah)
        gambarList.add(g31)
        val g32 =
Gambar(32,"Larangan masuk bagi
semua kendaraan tidak
bermotor.",R.drawable.larangan32
_semua_tidak_bermotor)
        gambarList.add(g32)
        val g33 =
Gambar(33,"Larangan masuk bagi
sepeda.",R.drawable.larangan33_s
epeda)
        gambarList.add(g33)
        val g34 =
Gambar(34,"Larangan masuk bagi
kendaraan dengan lebar lebih
dari 2,7
m.",R.drawable.larangan34_lebar2
7m)
        gambarList.add(g34)
        val g35 =
Gambar(35,"Larangan masuk bagi
kendaraan dengan muatan sumbu
dari 8
ton.",R.drawable.larangan35_8ton
)
        gambarList.add(g35)
        val g36 =
Gambar(36,"Larangan masuk bagi
kendaraan dengan muatan sumbu
terberat (MST) lebih besar dari
10 ton atau ukuran lebar tidak
melebihi 2.500 milimeter atau
ukuran panjang tidak melebihi
18.000
milimeter.",R.drawable.larangan3
6_10ton)
        gambarList.add(g36)
        val g37 =
Gambar(37,"Larangan masuk bagi
kendaraan dengan muatan sumbu
terberat (MTS) lebih besar dari
8 ton atau ukuran lebar tidak
melebihi 2.500 milimeter atau
ukuran panjang tidak melebihi
18.000
milimeter.",R.drawable.larangan3
7_3a)
        gambarList.add(g37)
        val g38 =
Gambar(38,"Larangan masuk bagi
kendaraan dengan tinggi lebih
dari 4,5
m.",R.drawable.larangan38_45mete
r)
        gambarList.add(g38)
        val g39 =
Gambar(39,"Larangan masuk bagi
kendaraan tidak bermotor dengan
panjang lebih dari …
m.",R.drawable.larangan39_panjan
g)
        gambarList.add(g39)
        val g40 =
Gambar(40,"Larangan masuk bagi
sepeda dan
becak.",R.drawable.larangan40_be
cak_sepeda)
        gambarList.add(g40)
        val g41 =
Gambar(41,"Larangan mendahului
kendaraan lain yang berjalan
didepan.",R.drawable.larangan41_
mendahului)
        gambarList.add(g41)
        val g42 =
Gambar(42,"Larangan menggunakan
isyarat
suara",R.drawable.larangan42_sua
ra)
        gambarList.add(g42)
        val g43 =
Gambar(43,"Sumbu terberat (MST)
lebih besar dari 8 ton atau
ukuran lebar tidak melebihi
2.100 milimeter atau ukuran
panjang tidak melebihi 9.000
milimeter",R.drawable.larangan43
_9000milimeter)
        gambarList.add(g43)
        val g44 =
Gambar(44,"Larangan masuk bagi
gerobak pedati dan sejenisnya
(Rambu
Larangan)",R.drawable.larangan44
_gerobak_pedati)
        gambarList.add(g44)
        val g45 =
Gambar(45,"Larangan masuk bagi
semua kendaraan bermotor maupun
tidak bermotor (Rambu
Larangan).",R.drawable.larangan4
5_semua_kendaraan)
        gambarList.add(g45)

        return gambarList
    }

    fun getRambuPeringatan() :
ArrayList<Gambar>{
        val gambarList =
ArrayList<Gambar>()

        val g1 = Gambar(1,"angin
dari
samping",R.drawable.peringatan1_
angin_dari_samping)
        gambarList.add(g1)
        val g2 = Gambar(2,"area
banyak pejalan kaki (Rambu
Peringatan)",R.drawable.peringat
an2_banyak_pejalan)
        gambarList.add(g2)
        val g3 =
Gambar(3,"banyak anak-
anak",R.drawable.peringatan3_ban
yak_anak)
        gambarList.add(g3)
        val g4 =
Gambar(4,"banyak satwa jinak dan
sering
menyebrang",R.drawable.peringata
n4_satwa_jinak)
        gambarList.add(g4)
        val g5 =
Gambar(5,"banyak satwa liar dan
sering
menyebrang",R.drawable.peringata
n5_satwa_liar)
        gambarList.add(g5)
        val g6 =
Gambar(6,"banyak
tikungan.",R.drawable.peringatan
6_banyak_tikungan)
        gambarList.add(g6)
        val g7 =
Gambar(7,"bundaran.",R.drawable.
peringatan7_bundaran)
```

```
        gambarList.add(g7)
        val g8 = Gambar(8,"hati-
hati.",R.drawable.peringatan8_ha
tihati)
        gambarList.add(g8)
        val g9 = Gambar(9,"jalan
bergelombang.",R.drawable.pering
atan9_jalan_bergelombang)
        gambarList.add(g9)
        val g10 =
Gambar(10,"jalan
cekung.",R.drawable.peringatan10
_jalan_cekumg)
        gambarList.add(g10)
        val g11 =
Gambar(11,"jalan
cembung.",R.drawable.peringatan_
jalan_cembung)
        gambarList.add(g11)
        val g12 =
Gambar(12,"jalan
licin",R.drawable.peringatan12_j
alan_licin)
        gambarList.add(g12)
        val g13 =
Gambar(13,"jembatan.",R.drawable
.peringatan13_jembatan)
        gambarList.add(g13)
        val g14 =
Gambar(14,"jembatan
angkat.",R.drawable.peringatan14
_angkat)
        gambarList.add(g14)
        val g15 =
Gambar(15,"kerikil
lepas.",R.drawable.peringatan15_
kericil_lepas)
        gambarList.add(g15)
        val g16 =
Gambar(16,"lalu lintas dua
arah",R.drawable.peringatan15_la
luluntas_dua_arah)
        gambarList.add(g16)
        val g17 =
Gambar(17,"lampu lalu
lintas.",R.drawable.peringatan17
_lalulintas)
        gambarList.add(g17)
        val g18 =
Gambar(18,"lebar ruang bebas
(Rambu
Peringatan).",R.drawable.peringa
tan18_ruang_bebas)
        gambarList.add(g18)
        val g19 =
Gambar(19,"lintasan pesawat
terbang.",R.drawable.peringatan1
9_lintasan_pesawat)
        gambarList.add(g19)
        val g20 =
Gambar(20,"longsoran
tanah.",R.drawable.peringatan20_
longsoran_tanah)
        gambarList.add(g20)
        val g21 =
Gambar(21,"pekerjaan
jalan.",R.drawable.peringatan21_
pekerjaan_jalan)
        gambarList.add(g21)
        val g22 =
Gambar(22,"pendakian.",R.drawabl
e.peringatan22_pendakian)
        gambarList.add(g22)
        val g23 =
Gambar(23,"pendakian
curam.",R.drawable.peringatan23_
pendakian_curam)
        gambarList.add(g23)
        val g24 =
Gambar(24,"pengarah tikungan ke
kiri dan ke
kanan.",R.drawable.peringatan24_
pengarah_kirikanan)
        gambarList.add(g24)
        val g25 =
Gambar(25,"penyebrangan pejalan
kaki.",R.drawable.peringatan25_p
enyebrangan_pejalankaki)
        gambarList.add(g25)
        val g26 =
Gambar(26,"penyempitan jalan di
kiri dan
kanan.",R.drawable.peringatan26_
penyempitan_kirikanan)
        gambarList.add(g26)
        val g27 =
Gambar(27,"penyempitan jalan
sebelah
kanan.",R.drawable.peringatan26_
penyempitan_kanan)
        gambarList.add(g27)
        val g28 =
Gambar(28,"penyempitan jalan
sebelah kiri (Rambu
Peringatan).",R.drawable.peringa
tan28_penyimpatan_kiri)
        gambarList.add(g28)
        val g29 =
Gambar(29,"perisimpangan tiga
sisi
kanan.",R.drawable.peringatan29_
persimpangantiga_kanan)
        gambarList.add(g29)
        val g30 =
Gambar(30,"persilangan datar
dengan lintasan kereta api
berpintu.",R.drawable.peringatan
30_persilangan_datar_berpintu)
        gambarList.add(g30)
        val g31 =
Gambar(31,"persilangan datar
dengan lintasan kereta api tanpa
pintu.",R.drawable.peringatan31_
persilangan_datar_tanpa_pintu)
        gambarList.add(g31)
        val g32 =
Gambar(32,"persimpangan
empat.",R.drawable.peringatan32_
persimpangan_empat)
        gambarList.add(g32)
        val g33 =
Gambar(33,"persimpangan ganda
kiri
kanan.",R.drawable.peringatan33_
persimpangan_ganda_kirikanan)
        gambarList.add(g33)
        val g34 =
Gambar(34,"persimpangan tiga
serong ke
kanan.",R.drawable.peringatan34_
persimpangantiga_serong_kanan)
        gambarList.add(g34)
        val g35 =
Gambar(35,"persimpangan tiga
serong ke
kiri.",R.drawable.peringatan35_p
ersimpangantiga_serong_kiri)
        gambarList.add(g35)
        val g36 =
Gambar(36,"persimpangan tiga
sisi
kiri.",R.drawable.peringatan36_p
ersimpangantiga_kiri)
        gambarList.add(g36)
        val g37 =
Gambar(37,"persimpangan tiga
type
T.",R.drawable.peringatan37_pers
impangantiga_typet)
        gambarList.add(g37)
        val g38 =
Gambar(38,"persimpangan tiga
type
Y.",R.drawable.peringatan38_pers
impangantiga_typey)
        gambarList.add(g38)
        val g39 =
Gambar(39,"tikungan ganda
tikungan pertama ke
kanan.",R.drawable.peringatan39_
tikungan_ganda_kanan)
        gambarList.add(g39)
        val g40 =
Gambar(40,"tikungan ganda
tikungan pertama ke
kiri.",R.drawable.peringatan40_t
ikungan_ganda_kiri)
        gambarList.add(g40)
        val g41 =
Gambar(41,"tikungan ke
kanan.",R.drawable.peringatan41_
tikungan_kekanan)
        gambarList.add(g41)
        val g42 =
Gambar(42,"tikungan ke
kiri",R.drawable.peringatan42_ti
kungan_kekiri)
        gambarList.add(g42)
        val g43 =
Gambar(43,"tikungan tajam ke
kanan",R.drawable.peringatan43_t
ikungan_tajam_kanan)
        gambarList.add(g43)
        val g44 =
Gambar(44,"tinggi ruang
bebas",R.drawable.peringatan45_t
inggi_ruang_bebas)
        gambarList.add(g44)
        val g45 =
Gambar(45,"turunan
.",R.drawable.peringatan45_turun
an)
        gambarList.add(g45)
        val g46 =
Gambar(46,"turunan terjal
.",R.drawable.peringatan46_turun
an_terjal)
        gambarList.add(g46)

        return gambarList
    }

    fun getRambuPerintah() :
ArrayList<Gambar>{
        val gambarList =
ArrayList<Gambar>()

        val g1 = Gambar(1,"Batas
akhir kecepatan minimum yang
diwajibkan",R.drawable.perintah1
_batas_akhir_kecepatan)
        gambarList.add(g1)
        val g2 = Gambar(2,"Batas
akhir wajib memakai rantai pada
ban",R.drawable.perintah2_batas_
memakai_rantai)
        gambarList.add(g2)
        val g3 = Gambar(3,"jalan
satu arah ke
kanan",R.drawable.perintah3_jala
n_satuh_arah_kanan)
        gambarList.add(g3)
        val g4 = Gambar(4,"Lajur
atau bagian jalan yang wajib
dilewati",R.drawable.perintah4_j
alur_wajib_dilewati)
        gambarList.add(g4)
        val g5 =
Gambar(5,"Perintah kecepatan
minumum yang
diwajibkan",R.drawable.perintah5
_kecepatan_minimun)
        gambarList.add(g5)
        val g6 = Gambar(6,"Wajib
berjalan lurus
kedepan.",R.drawable.perintah6_w
ajib_berjalan_lurus)
        gambarList.add(g6)
        val g7 = Gambar(7,"Wajib
melewati salah satu lajur yang
ditunjuk.",R.drawable.perintah7_
wajib_melewati_salasatu_jalur)
        gambarList.add(g7)
        val g8 = Gambar(8,"Wajib
memakai rantai pada
ban.",R.drawable.perintah8_wajib
_memakai_ban_rantai)
        gambarList.add(g8)
        val g9 = Gambar(9,"Wajib
Mengikuti Arah Ke
Kanan.",R.drawable.perintah9_waj
ib_jalur_kanan)
        gambarList.add(g9)
        val g10 =
Gambar(10,"Wajib Mengikuti Arah
Ke
Kiri.",R.drawable.perintah10_waj
ib_jalur_kiri)
        gambarList.add(g10)
        val g11 =
Gambar(11,"Wajib mengikuti arah
yang
ditunjuk.",R.drawable.perintah11
_wajib_jalur_ditunjuk)
        gambarList.add(g11)
        val g12 =
Gambar(12,"Wajib mengikuti salah
satu arah yang
ditunjuk.",R.drawable.perintah12
_wajub_mengikuti_salahsatu_jalur
)
        gambarList.add(g12)
        val g13 =
Gambar(13,"Wajib mengikuti salah
satu arah yang
ditunjuki.",R.drawable.perintah1
3_wajub_mengikuti_salahsatu_jalu
r)
        gambarList.add(g13)
        val g14 =
Gambar(14,"Wajib untuk lalu
lintas
pedati.",R.drawable.perintah14_w
ajib_pedati)
        gambarList.add(g14)
        val g15 =
Gambar(15,"Wajib untuk
lalulintas
becak.",R.drawable.perintah15_wa
jib_becak)
        gambarList.add(g15)
        val g16 =
Gambar(16,"lWajib untuk
lalulintas
bersepeda",R.drawable.perintah16
_wajib_sepeda)
        gambarList.add(g16)
        val g17 =
Gambar(17,"Wajib untuk
lalulintas
dokar.",R.drawable.perintah17_wa
jib_dokar)
        gambarList.add(g17)
        val g18 =
Gambar(18,"Wajib untuk
lalulintas pedati, gerobak
dorong dan
dokar.",R.drawable.perintah18_wa
jib_gerobak)
        gambarList.add(g18)
        val g19 =
Gambar(19,"Wajib untuk
lalulintas pengendara
berkuda.",R.drawable.perintah19_
wajib_berkuda)
        gambarList.add(g19)
        val g20 =
Gambar(20,"Wajib untuk pejalan
kaki.",R.drawable.perintah20_waj
ib_pejalan)
        gambarList.add(g20)

        return gambarList
    }

    fun getRambuLPetunjuk() :
ArrayList<Gambar>{
        val gambarList =
ArrayList<Gambar>()

        val g1 = Gambar(1,"akhir
dari jalan
tol",R.drawable.petunjuk1_akhir_
jalan_tol)
        gambarList.add(g1)
        val g2 =
Gambar(2,"bandara
udara",R.drawable.petunjuk2_band
ara)
        gambarList.add(g2)
        val g3 =
Gambar(3,"bengkel",R.drawable.pe
tunjuk3_bengkel)
        gambarList.add(g3)
        val g4 =
Gambar(4,"gelanggang
olahraga",R.drawable.petunjuk4_o
lahraga)
        gambarList.add(g4)
        val g5 =
Gambar(5,"gereja",R.drawable.pet
unjuk5_gereja)
        gambarList.add(g5)
        val g6 = Gambar(6,"halte
bus.",R.drawable.petunjuk6_halte
_bus)
        gambarList.add(g6)
        val g7 = Gambar(7,"jalan
buntu",R.drawable.petunjuk7_jala
n_buntu)
        gambarList.add(g7)
        val g8 = Gambar(8,"jalan
satu arah ke
kiri",R.drawable.petunjuk8_jalan
_satu_arah_kiri)
        gambarList.add(g8)
        val g9 = Gambar(9,"jalan
satu arah
lurus",R.drawable.petunjuk9_jalu
r_satu_arah_lurus)
        gambarList.add(g9)
        val g10 =
Gambar(10,"jalan
tol",R.drawable.petunjuk10_jalan
_tol)
        gambarList.add(g10)
        val g11 =
Gambar(11,"jembatan
timbang",R.drawable.petunjuk11_j
embatan_timbang)
        gambarList.add(g11)
        val g12 =
Gambar(12,"kamp
pengunsian",R.drawable.petunjuk1
2_kamp_pengungsian)
        gambarList.add(g12)
        val g13 =
Gambar(13,"kantor
pos",R.drawable.petunjuk13_kanto
r_pos)
        gambarList.add(g13)
        val g14 =
Gambar(14,"kemah
karvan",R.drawable.petunjuk14_ke
mah_karvan)
        gambarList.add(g14)
        val g15 =
Gambar(15,"kemah
tenda",R.drawable.petunjuk15_kem
ah_tenda)
        gambarList.add(g15)
        val g16 =
Gambar(16,"lkendaraan umum
selain bus dan
taksi",R.drawable.petunjuk16_ken
daraan_umum_selain_taksi)
        gambarList.add(g16)
        val g17 =
Gambar(17,"khusus kendaraan
bermotor",R.drawable.petunjuk17_
kendaraan_bermotor)
        gambarList.add(g17)
        val g18 =
Gambar(18,"kolam
renang",R.drawable.petunjuk18_ko
lam_renang)
        gambarList.add(g18)
        val g19 =
Gambar(19,"lapangan olahraga
terbuka",R.drawable.petunjuk19_l
apangan_olahraga)
        gambarList.add(g19)
        val g20 =
Gambar(20,"lokasi berkumpul
darurat",R.drawable.petunjuk20_l
okasi_berkumpul_darurat)
        gambarList.add(g20)
        val g21 =
Gambar(21,"masjid",R.drawable.pe
tunjuk21_masjid)
        gambarList.add(g21)
        val g22 =
Gambar(22,"museum",R.drawable.pe
tunjuk22_museum)
        gambarList.add(g22)
        val g23 =
Gambar(23,"pantai",R.drawable.pe
tunjuk23_pantai)
        gambarList.add(g23)
        val g24 =
Gambar(24,"Pasar",R.drawable.pet
unjuk24_pasar)
        gambarList.add(g24)
        val g25 =
Gambar(25,"pelabuhan",R.drawable
.petunjuk25_pelabuhan)
        gambarList.add(g25)
        val g26 =
Gambar(26,"penginapan",R.drawabl
e.petunjuk26_penginapan)
        gambarList.add(g26)
        val g27 =
Gambar(27,"perkemahan",R.drawabl
e.petunjuk27_perkemahan)
        gambarList.add(g27)
        val g28 =
Gambar(28,"perpustakaan",R.drawa
ble.petunjuk28_perpustakaan)
        gambarList.add(g28)
        val g29 =
Gambar(29,"pompa bahan
bakar",R.drawable.petunjuk29_pom
pa_bahan_bakar)
        gambarList.add(g29)
        val g30 =
Gambar(30,"Pura",R.drawable.petu
njuk30_pura)
        gambarList.add(g30)
        val g31 =
Gambar(31,"pusat
pengunsian",R.drawable.petunjuk3
1_pusat_pengungsian)
        gambarList.add(g31)
        val g32 =
Gambar(32,"puskesmas",R.drawable
.petunjuk32_puskesmas)
        gambarList.add(g32)
        val g33 =
Gambar(33,"rumah
makan",R.drawable.petunjuk33_rum
ah_makan)
        gambarList.add(g33)
        val g34 =
Gambar(34,"rumah
sakit",R.drawable.petunjuk34_rum
ah_sakit)
        gambarList.add(g34)
        val g35 =
Gambar(35,"rute evakuasi gempa
bumi",R.drawable.petunjuk35_rute
_evakuasi_gempa)
        gambarList.add(g35)
        val g36 =
Gambar(36,"rute evakuasi letusan
gunung
merapi",R.drawable.petunjuk36_le
tusan_gunung)
        gambarList.add(g36)
        val g37 =
Gambar(37,"rute evakuasi
stsunami",R.drawable.petunjuk37_
tsunami)
        gambarList.add(g37)
        val g38 =
Gambar(38,"rute penjelajah
alam",R.drawable.petunjuk38_penj
elajah_alam)
        gambarList.add(g38)
        val g39 =
Gambar(39,"sekolah",R.drawable.p
etunjuk39_sekolah)
        gambarList.add(g39)
        val g40 =
Gambar(40,"stadion",R.drawable.p
etunjuk40_stadion)
        gambarList.add(g40)
        val g41 =
Gambar(41,"stand
taksi",R.drawable.petunjuk41_sta
nd_taksi)
        gambarList.add(g41)
        val g42 =
Gambar(42,"stasiun kereta
api",R.drawable.petunjuk42_stasi
un_keretaapi)
        gambarList.add(g42)
        val g43 =
Gambar(43,"stasiun uji
berkala",R.drawable.petunjuk43_s
tasiun_uji_berkala)
        gambarList.add(g43)
        val g44 =
Gambar(44,"stasiun uji
emisi",R.drawable.petunjuk44_uji
_emisi)
        gambarList.add(g44)
        val g45 =
Gambar(45,"taman",R.drawable.pet
unjuk45_taman)
        gambarList.add(g45)
        val g46 =
Gambar(46,"telpon
umum",R.drawable.petunjuk46_tele
pon_umum)
        gambarList.add(g46)
        val g47 =
Gambar(47,"tempat pembuangan
akhir",R.drawable.petunjuk47_pen
uangan_akhir)
        gambarList.add(g47)
        val g48 =
Gambar(48,"tempat pembuangan
sementara",R.drawable.petunjuk48
_pembuangan_sementara)
        gambarList.add(g48)
        val g49 =
Gambar(49,"tempat penyebrangan
orang",R.drawable.petunjuk49_tem
pat_penyebrangan)
        gambarList.add(g49)
        val g50 =
Gambar(50,"tempat putar balik
arah",R.drawable.petunjuk50_puta
r_balik_arah)
        gambarList.add(g50)
        val g51 =
Gambar(51,"terminal
bus",R.drawable.petunjuk51_termi
nal_bus)
        gambarList.add(g51)
        val g52 =
Gambar(52,"toko
obat",R.drawable.petunjuk52_toko
_obat)
        gambarList.add(g52)
        val g53 =
Gambar(53,"trowongan
berakhir",R.drawable.petunjuk53_
terowongan_berakhir)
        gambarList.add(g53)
        val g54 =
Gambar(54,"trowongan
bermula",R.drawable.petunjuk54_t
erowongan_bermula)
        gambarList.add(g41)
        val g55 =
Gambar(55,"vila",R.drawable.petu
njuk55_vila)
        gambarList.add(g55)
        val g56 =
Gambar(56,"warung
kopi",R.drawable.petunjuk56_waru
ng_kopi)
        gambarList.add(g56)
        val g57 =
Gambar(57,"Whara",R.drawable.pet
unjuk57_whara)
        gambarList.add(g57)
        val g58 =
Gambar(58,"Zona
Parkir",R.drawable.petunjuk58_zo
na_patkir)
        gambarList.add(g58)
        val g59 =
Gambar(59,"zona parkir untuk
penyadangan
cacat",R.drawable.petunjuk59_zon
a_penyandang_cacat)
        gambarList.add(g59)

        return gambarList
    }
}

package
com.example.kuisrambulalulintas.
utils

sealed class Resource<T>(val
data: T? = null, val message:
String? = null) {
    class Success<T>(data: T) :
Resource<T>(data)
    class Loading<T>(data: T? =
null) : Resource<T>(data)
    class Error<T>(message:
String, data: T? = null) :
Resource<T>(data, message)
    class Empty<Unit>():
Resource<Unit>()
}
```

```
package com.example.kuisrambulalulintas.viewmodel

import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel
import com.example.kuisrambulalulintas.data.MainRepository
import com.example.kuisrambulalulintas.model.DataSoal
import com.example.kuisrambulalulintas.utils.Resource
import dagger.hilt.android.lifecycle.HiltViewModel
import javax.inject.Inject

@HiltViewModel
class MainViewModel @Inject constructor(
    private val mainRepository: MainRepository
) : ViewModel() {


    private val _soal = MutableLiveData<Resource<List<DataSoal>>>()
    val soal : MutableLiveData<Resource<List<DataSoal>>>
    get() = _soal


    fun getKuis(){

        _soal.value = Resource.Loading()
        mainRepository.getKuis {
            _soal.value = it
        }
    }

}

package com.example.kuisrambulalulintas

import android.app.Application
import dagger.hilt.android.HiltAndroidApp

@HiltAndroidApp
class KuisApp : Application() {
}

package com.example.kuisrambulalulintas

import android.content.DialogInterface
import android.content.Intent
import android.content.SharedPreferences
import android.media.MediaPlayer
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.appcompat.app.AlertDialog
import com.example.kuisrambulalulintas.databinding.ActivityMainBinding
import com.example.kuisrambulalulintas.service.BackgroundSoundService
import com.example.kuisrambulalulintas.ui.activities.GetNameActivity
import com.example.kuisrambulalulintas.ui.activities.LevelActivity
import com.example.kuisrambulalulintas.ui.activities.MateriActivity

class MainActivity : AppCompatActivity() {

    var mediaPlayer: MediaPlayer? = null
    var status: Int? = 0

    lateinit var sharedPref : SharedPreferences

    var mPlay = false
    private      lateinit      var
binding: ActivityMainBinding
    override                  fun
onCreate(savedInstanceState:
Bundle?) {

super.onCreate(savedInstanceStat
e)
        binding                  =
ActivityMainBinding.inflate(layo
utInflater)

setContentView(binding.root)

        sharedPref               =
getSharedPreferences("myPref",
MODE_PRIVATE)
        val       editor         =
sharedPref.edit()


//startService(Intent(this,
BackgroundSoundService::class.ja
va))



binding.cvPlay.setOnClickListene
r {

startActivity(Intent(this,
LevelActivity::class.java))
        }


binding.cvMusic.setOnClickListen
er {
            if (mPlay){
                mPlay = false
                status = 1
                editor.apply {

putInt("status", status!!)
                }
                val  statusOFF  =
sharedPref.getInt("status",statu
s!!)
                status           =
statusOFF

binding.tvMusic.text  =  "MUSIC  :
OFF"

stopService(Intent(this,
BackgroundSoundService::class.ja
va))
            } else {
                mPlay = true
                status = 0
                editor.apply {

putInt("status", status!!)
                }
                val  statusOFF  =
sharedPref.getInt("status",statu
s!!)
                status           =
statusOFF

binding.tvMusic.text  =  "MUSIC  :
ON"

startService(Intent(this,
BackgroundSoundService::class.ja
va))
            }
        }


binding.cvMateri.setOnClickListe
ner {

startActivity(Intent(this,Materi
Activity::class.java))

        }


binding.cvExit.setOnClickListene
r {
            showAlertDialog()
        }

    }

    override  fun  onBackPressed()
{
       showAlertDialog()
    }

    private                   fun
showAlertDialog() {
        val    dialogBuilder     =
AlertDialog.Builder(this)


dialogBuilder.setMessage("Apakah
anda    yakin    akan    keluar    dari
aplikasi ini ?")

.setCancelable(false)

.setPositiveButton("Ya",DialogIn
terface.OnClickListener { _, _ -
>
                finish()
            })

.setNegativeButton("keluar",Dial
ogInterface.OnClickListener     {
dialogInterface, i ->

dialogInterface.cancel()
            })

        val        alert        =
dialogBuilder.create()
        alert.setTitle("Keluar")
        alert.show()
    }


    /*private fun playAudio() {
       if (mediaPlayer == null){
           mediaPlayer          =
MediaPlayer.create(this,R.raw.mu
sic)

mediaPlayer!!.isLooping = true
           mediaPlayer!!.start()
       } else {
           mediaPlayer!!.start()
       }
    }*/
}

<?xml              version="1.0"
encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.an
droid.com/apk/res/android"

xmlns:tools="http://schemas.andr
oid.com/tools">

    <application
        android:name=".KuisApp"

android:allowBackup="true"

android:dataExtractionRules="@xm
l/data_extraction_rules"

android:fullBackupContent="@xml/
backup_rules"

android:icon="@mipmap/ic_launche
r"

android:label="@string/app_name"

android:roundIcon="@mipmap/ic_la
uncher_round"

android:supportsRtl="true"

android:theme="@style/Theme.Kuis
RambuLaluLintas"
        tools:targetApi="31">
        <activity

android:name=".ui.activities.Que
stionActivity"

android:exported="false">
            <meta-data

android:name="android.app.lib_na
me"
                 android:value=""
/>
        </activity>

android:name=".ui.activities.Mat
eriActivity"

android:theme="@style/AppTheme"

android:exported="false">
            <meta-data

android:name="android.app.lib_na
me"
                 android:value=""
/>
        </activity>
        <activity

android:name=".ui.activities.Lev
elActivity"

android:exported="false">
           <meta-data
```

```
android:name="android.app.lib_na
me"
                 android:value=""
/>
        </activity>
        <activity

android:name=".ui.activities.Get
NameActivity"

android:theme="@style/AppTheme"

android:exported="false">
            <meta-data

android:name="android.app.lib_na
me"
                 android:value=""
/>
        </activity>
        <activity

android:name=".ui.activities.Kui
sActivity"

android:exported="false">
            <meta-data

android:name="android.app.lib_na
me"
                 android:value=""
/>
        </activity>
        <activity

android:name=".ui.activities.Res
ultActivity"

android:theme="@style/AppTheme"

android:exported="false">
            <meta-data
```

```
android:name="android.app.lib_na
me"
                 android:value=""
/>
        </activity>
        <activity

android:name=".MainActivity"

android:theme="@style/AppTheme"

android:exported="true">
            <intent-filter>
                          <action
android:name="android.intent.act
ion.MAIN" />

                        <category
android:name="android.intent.cat
egory.LAUNCHER" />
            </intent-filter>

            <meta-data

android:name="android.app.lib_na
me"
                 android:value=""
/>
        </activity>

                         <service
android:name=".service.Backgroun
dSoundService"
            />
        <meta-data

android:name="preloaded_fonts"

android:resource="@array/preload
ed_fonts" />

    </application>

</manifest>
```