

**HALAMAN PERSETUJUAN**

**APLIKASI *EDUCATION GAME* DALAM PEMBELAJARAN  
INTERAKTIF PENGENALAN RAMBU LALU LINTAS PADA  
ANAK SD BERBASIS *MOBILE LEARNING***

**MUH. RIZAL SYAM  
NIM. 218280156**

Telah diperiksa dan disetujui untuk mengikuti Sidang Skripsi/ Tutup

Parepare, 25 Agustus 2024  
Komisi Pembimbing

**Pembimbing I**



**Ade Hastuty, S.T., S.Kom., M.T.  
NIP. 19720120 200901 2 002**

**Pembimbing II**



**Wahyuddin, S.Kom., M.Kom.  
NBM. 1208 048**

Mengetahui ;  
**Ketua Program Studi**



**Marlina S.Kom., M.Kom.  
NBM. 1162 680**

**HALAMAN PENGESAHAN**

**APLIKASI GAME EDUKASI INTERAKTIF  
PENGENALAN RAMBU LALU LINTAS UNTUK ANAK  
SD BERBASIS MOBILE LEARNING**

**MUH. RIZAL SYAM  
NIM. 218280156**

Telah dipertahankan di depan Komisi Penguji Ujian Skripsi pada tanggal  
30 Agustus 2024 dan dinyatakan telah memenuhi syarat

**Komisi Penguji**

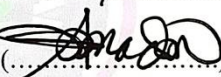
Ade Hastuty, S.T., S.Kom., M.T. (Ketua)

(.....)

Wahyuddin, S.Kom., M.Kom (Sekretaris)

(.....)

Ahmad Selao, S.T.P., M.Sc (Anggota)

(.....)

Marlina, S.Kom., M.Kom (Anggota)

(.....)

Mengetahui :

Ketua Program Studi  
Teknik Informatika

  
**Marlina, S.Kom., M. Kom.**  
NBM. 1162 680

Dekan  
Fakultas Teknik

  
**Muh Basri, S.T., M.T.**  
NBM. 959 773

## PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertandatangan di bawah ini :

Nama : **Muh. Rizal Syam**  
NIM : **218280156**  
Program Studi : Teknik Informatika  
Fakultas : Teknik Universitas Muhammadiyah Parepare  
Judul Skripsi : Aplikasi *Education Game* Dalam Pembelajaran Interaktif Pengenalan Rambu Lalu Lintas Pada Anak SD Berbasis *Mobile Learning*

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil dari karya saya sendiri, bukan merupakan pengambilalihan tulisan atau pemikiran orang lain. Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian dari keseluruhan skripsi ini hasil karya orang lain, saya bersedia menerima sanksi sesuai dengan aturan yang berlaku.

Parapare, 26 Agustus 2024

Yang menyatakan



**Muh. Rizal Syam**  
**NIM. 218280156**

## PRAKATA



*Assalamu'alaikum Warahmatullah Wabarakatu.*

Dalam skripsi ini penulis memohon maaf yang sebesar-besarnya apabila terdapat kekurangan dan kesalahan dalam penulisan dan artikulasi kata, kalimat maupun paragraf.

Tiada kata yang paling pantas untuk diucapkan selain rasa syukur atas nikmat dan kasih sayang Allah SWT. yang telah memberikan kesehatan dan kekuatan serta pertolongan-Nya untuk menyelesaikan skripsi ini. Tak lupa salam dan shalawat dihaturkan kepada Nabiullah Muhammad SAW beserta seluruh keluarga dan sahabatnya yang dengan setia berjuang menegakkan ajaran islam di muka bumi ini. Nabi utusan Allah SWT. yang mengajarkan wahyu Allah SWT. dan budi pekerti yang baik dan mulia bagi manusia.

Penulisan skripsi/ tugas akhir ini diajukan untuk memenuhi salah satu syarat untuk memperoleh gelar sarjana Strata Satu (S1) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Parepare (UM-Parepare). Adapun judul penelitian ini adalah **“Aplikasi *Education Game* Dalam Pembelajaran Interaktif Pengenalan Rambu Lalu Lintas Pada Anak SD Berbasis *Mobile Learning*”**.

Dalam pelaksanaan penelitian sampai pembuatan skripsi ini, penulis banyak sekali mengalami kesulitan dan hambatan. Tetapi berkat keteguhan dan kesabaran penulis akhirnya skripsi ini dapat diselesaikan juga. Terima kasih yang tak

terhingga pula kepada kedua orang tua penulis, ayahanda dan ibunda yang selalu memberikan doa, kasih sayang, dan dukungan baik moral maupun materil yang merupakan kekuatan besar bagi penulis untuk menyelesaikan skripsi ini. Bantuan dari berbagai pihak yang dengan senang hati meluangkan waktu, tenaga, pikiran, dan dukungan baik secara moril maupun materil yang tak henti-hentinya kepada penulis juga menjadi semangat positif untuk menyelesaikan skripsi ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang tiada terhingga kepada :

1. Allah SWT, Pencipta alam semesta beserta segala isinya dan pemilik segala ilmu. Terima kasih yaa Allah atas segala skenario kehidupan yang telah Engkau siapkan untuk hamba, hanya kepadaMu hamba menyembah, hanya kepadaMu hamba meminta pertolongan dan hanya kepadaMu hamba berserah diri, segala daya dan upaya serta kemampuan datang dariMu yaa Allah, Allahu Akbar.
2. Ayahanda Syamsuddin dan Ibunda Erni Sukardi jasa-jasanya, kesabaran, do'a yang tidak pernah lelah, mendidik dan memberi cinta yang tulus dan ikhlas kepada penulis semenjak kecil hingga sekarang beserta para kakak dan keluarga lainnya yang senantiasa memberikan dukungan.
3. Rektor Universitas Muhammadiyah Parepare, Prof. Dr. H. Jamaluddin Ahmad, S.Sos., M.Si.
4. Dekan Fakultas Teknik UM Parepare, Muh. Basri, S.T., M.T.
5. Ketua Program Studi Teknik Informatika Universitas Muhammadiyah Parepare, Marlina, S.Kom., M.Kom.

6. Sekretaris Program Studi Teknik Informatika Universitas Muhammadiyah Parepare, Wahyuddin, S.Kom., M.Kom.
7. Pembimbing I, Ade Hastuty, S.T., S.Kom., M.T. dan Pembimbing II, Wahyuddin, S.Kom., M.Kom. yang telah banyak meluangkan waktunya untuk membimbing penulis dalam menyelesaikan skripsi ini.
8. Penguji I, Marlina, S.Kom., M.Kom. dan Penguji II, Ahmad Selao, S.TP., M.Sc. yang telah banyak memberikan masukan dan ilmu yang sangat berguna bagi penulis.
9. Para Dosen dan Staf Fakultas Teknik Informatika Universitas Muhammadiyah Parepare yang telah banyak membantu dan juga atas partisipasinya selama penulis menempuh pendidikan di Universitas Muhammadiyah Parepare.
10. Saudara-saudara yang telah banyak memberikan dorongan, semangat, kasih sayang dan bantuan yang berarti baik secara moral maupun material demi lancarnya penyusunan tugas akhir ini.
11. Seluruh teman-teman Program Studi Teknik Informatika angkatan 2018 atas kebersamaan dan bantuan yang berarti bagi penulis.
12. Para senior dan adik-adik di Teknik Informatika Universitas Muhammadiyah Parepare yang telah membantu, memberi masukan, dan dorongan hingga selesainya skripsi ini.
13. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah dengan tulus ikhlas serta memberikan doa dan motivasi kepada penulis sehingga dapat terselesaikan skripsi ini.

Penulis sadar bahwa tentunya dalam penulisan skripsi ini masih banyak terdapat kekurangan untuk itu saran dan kritik dari pembaca yang sifatnya membangun sangat diharapkan demi pengembangan kemampuan penulis ke depan.

Akhirulkitab, hanya kepada Allah SWT. penulis memohon ridho dan magfirahNya, semoga keikhlasan dan bantuan yang telah diberikan kepada penulis dapat bernilai pahala disisi-Nya. Semoga karya ini dapat memberikan manfaat kepada mereka yang membutuhkan, semoga Allah SWT. melimpahkan rahmat-Nya kepada kita semua. Aamiin.

***Billahi Fii Sabilil Haq, Fastabiqul Khaerat***

***Wassalamu Alaikum Wr.Wb***

## ABSTRAK

**MUH. RIZAL SYAM.** Aplikasi *Education Game* Dalam Pembelajaran Interaktif Pengenalan Rambu Lalu Lintas Pada Anak SD Berbasis *Mobile Learning* (Dibimbing oleh Ade Hastuty dan Wahyuddin)

Game merupakan salah satu media audio visual yang dibuat untuk tujuan tertentu dan dirancang agar pengguna secara lebih aktif terlibat dengan sistem didalamnya. Perkembangan game saat ini sangat berkembang pesat sehingga menghasilkan berbagai macam aplikasi pembelajaran yang bisa dinikmati mulai dari anak-anak hingga dewasa, salah satu aplikasi pembelajaran yang cocok untuk saat ini adalah aplikasi *education game* dalam suatu pembelajaran edukasi. Rambu lalu lintas merupakan salah satu pengetahuan yang sangat penting bagi pengguna jalan dalam menertibkan lalu lintas, hal ini dikarenakan dapat mengurangi suatu permasalahan lalu lintas seperti macet, kecelakaan lalu lintas dan permasalahan lalu lintas lainnya yang dapat merugikan orang lain, meningkatkan pemahaman tentang rambu-rambu lalu lintas, bagi anak usia dini pemanfaatan suatu aplikasi. Metode penelitian yang digunakan mencakup pengujian *blackbox* dan *whitebox*. Pengujian *blackbox* berfokus pada evaluasi fungsionalitas aplikasi berdasarkan *input* dan *output* tanpa mempertimbangkan struktur internal aplikasi, sedangkan pengujian *whitebox* menilai logika dan alur kerja internal suatu aplikasi untuk memastikan tidak adanya kesalahan logika. Dihasilkan suatu Aplikasi *Education Game* Dalam Pembelajaran Interaktif Pengenalan Rambu Lalu Lintas Pada Anak SD Berbasis *Mobile Learning*. Dengan adanya suatu *education game* ini dapat menghemat dari segi biaya, waktu dan tenaga dalam suatu proses penyuluhan, sosialisasi dan kesadaran untuk belajar mengenai aturan berlalu lintas. Dengan cara ini, diharapkan aplikasi ini dapat meningkatkan motivasi belajar anak-anak SD. Faktor-faktor seperti elemen permainan, tantangan, dan *reward system* dalam aplikasi membuat pembelajaran menjadi lebih interaktif bagi anak-anak.

**Kata kunci :** Aplikasi, Game, Rambu Lalu Lintas, Mobile Learning



## **ABSTRACT**

**MUH. RIZAL SYAM.** *Educational Game Application in Interactive Learning on Introduction to Traffic Signs for Elementary School Children Based on Mobile Learning (Supervised by Ade Hastuty and Wahyuddin)*

*Games are an audio visual medium that is created for a specific purpose and is designed so that users are more actively involved with the system within it. The development of games is currently growing rapidly, resulting in various kinds of learning applications that can be enjoyed from children to adults. One of the learning applications that is suitable at this time is the educational game application in educational learning. Traffic signs are a very important piece of knowledge for road users in controlling traffic, this is because they can reduce traffic problems such as traffic jams, traffic accidents and other traffic problems that can harm other people, increase understanding of the signs. traffic, for young children the use of an application. The research methods used include black box and white box testing. Black box testing focuses on evaluating an application's functionality based on input and output without considering the internal structure of the application, while white box testing assesses the internal logic and workflow of an application to ensure that there are no logic errors. An educational game application was produced in interactive learning about recognizing traffic signs for elementary school children based on mobile learning. With this educational game, you can save money, time and energy in the process of counseling, socialization and awareness to learn about traffic rules. In this way, it is hoped that this application can increase elementary school children's learning motivation. Factors such as game elements, challenges, and reward systems in applications make learning more interactive for children.*

**Keywords :** *Applications, Games, Traffic Signs, Mobile Learning*

## DAFTAR ISI

SAMPUL	i
HALAMAN JUDUL	ii
LEMBAR PERSETUJUAN	iii
PERNYATAAN KEASLIAN SKRIPSI	iv
HALAMAN INSPIRASI	v
PRAKATA	vi
ABSTRAK	x
<i>ABSTRACT</i>	xi
DAFTAR ISI	xii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Rumusan Masalah	4
C. Batasan Masalah	4
D. Tujuan Penelitian	5
E. Manfaat Penelitian	5
F. Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA	
A. Penelitian Terdahulu	8
B. Tinjauan Teori	11

1. Aplikasi	11
2. Game Edukasi ( <i>Education Game</i> )	11
3. Rambu Lalu Lintas	14
4. <i>Mobile Learning</i>	16
C. Tinjauan Perangkat Lunak	19
D. Tahapan Perancangan Sistem	24
E. Metode Pengujian Perangkat Lunak	35
F. Kerangka Fikir	38
<b>BAB III METODOLOGI PENELITIAN</b>	
A. Tempat dan Waktu Penelitian	39
B. Metode Penelitian	39
C. Metode Pengumpulan Data	43
D. Sumber Data Penelitian	45
E. Alat dan Bahan	46
<b>BAB IV HASIL DAN PEMBAHASAN</b>	
A. Rancangan Sistem	47
B. Analisis Aliran Data UML	50
C. Detail Aplikasi	58
D. Rancangan <i>Output</i> Program	67
E. Implementasi	70
F. Metode Pengujian	72
G. Teknik Pengujian	72

## BAB V PENUTUP

A. Kesimpulan	83
B. Saran-saran	84

## DAFTAR PUSTAKA

## LAMPIRAN

## DAFTAR GAMBAR

Gambar 2.1 Contoh Rambu Lalu Lintas	17
Gambar 2.2 Skema M-Learning	19
Gambar 2.3 Logo Android	21
Gambar 2.4 Flutter	22
Gambar 2.5 Android Studio	23
Gambar 2.6 Java Development Kit (JDK)	24
Gambar 2.7 Software Development Kit (JDK)	24
Gambar 2.8 Visual Studio Code	25
Gambar 2.9 Pengujian White Box	38
Gambar 2.10 Pengujian Black Box	39
Gambar 2.11 Kerangka Fikir	40
Gambar 4.1 Use Case Diagram Sistem Diusulkan	49
Gambar 4.2 Flowchart Login	50
Gambar 4.3 Flowchart System	51
Gambar 4.4 Use Case Diagram Admin	52
Gambar 4.5 Activity Diagram Tambah Data Materi	53
Gambar 4.6 Activity Diagram Hapus Data Materi	54
Gambar 4.7 Activity Diagram Edit Data Materi	55
Gambar 4.8 Activity Diagram Lihat Materi	56
Gambar 4.9 Activity Diagram User	57
Gambar 4.10 Sequence Diagram Tambah Data Admin	58
Gambar 4.11 Sequence Diagram Edit Data Admin	58
Gambar 4.12 Sequence Diagram Hapus Data Admin	59
Gambar 4.13 Sequence Diagram Data Materi	59
Gambar 4.14 Halaman Utama Admin	60
Gambar 4.15 Halaman Menu Tambah Data Materi	61

Gambar 4.16 Halaman Menu Edit Data Materi	62
Gambar 4.17 Halaman Menu Hapus Data Materi	63
Gambar 4.18 Halaman Menu Home User	64
Gambar 4.19 Halaman Menu Pilih Level User	65
Gambar 4.20 Halaman Menu Masukkan User	66
Gambar 4.21 Halaman Menu Evaluasi (Quiz Interaktif)	67
Gambar 4.22 Halaman Menu Score	68
Gambar 4.23 Halaman Jawaban Benar	69
Gambar 4.24 Halaman Jawaban Salah	70
Gambar 4.25 Halaman Popup Score	71

## DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	8
Tabel 2.2 Simbol-Simbol Dalam Kamus Data	24
Tabel 2.3 Simbol <i>Flowchart</i>	26
Tabel 2.4 <i>Simbol Entity Relationship Diagram</i>	27
Tabel 2.5 <i>Table Use Case Diagram</i>	33
Tabel 2.6 <i>Simbol Activity Diagram</i>	34
Tabel 2.7 <i>Simbol Sequence Diagram</i>	34
Tabel 3.1 Spesifikasi Perangkat Lunak	46
Tabel 4.1 Spesifikasi <i>Hardware</i> Komputer	70
Tabel 4.2 Spesifikasi <i>Smartphone Admin</i>	70
Tabel 4.3 Spesifikasi <i>Smartphone User</i>	70
Tabel 4.4 Spesifikasi <i>Software</i> Komputer	71
Tabel 4.5 Spesifikasi <i>Software Admin</i>	71
Tabel 4.6 Spesifikasi <i>Software User</i>	71
Tabel 4.7 Hasil Pengujian <i>Input</i> Data Materi	74
Tabel 4.8 Hasil Pengujian Edit Data Materi	75
Tabel 4.9 Hasil Pengujian Hapus Data Materi	76
Tabel 4.10 Pengujian <i>Input</i> Data Materi	77
Tabel 4.11 <i>Grafik Matrix</i> pada <i>Flowgraph Input</i> Data Materi	78
Tabel 4.12 Pengujian <i>Edit</i> Data Materi	79
Tabel 4.13 <i>Grafik Matrix</i> pada <i>Flowgraph Edit</i> Data Materi	80
Tabel 4.14 Pengujian Hapus Data Materi	81
Tabel 4.15 <i>Grafik Matrix</i> pada <i>Flowgraph Hapus</i> Data Materi	82

# LAMPIRAN



➤ **Controller**

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

import android.content.Intent
import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.kuisrambulalulintas.R
import com.example.kuisrambulalulintas.databinding.ActivityGetNameBinding
import com.example.kuisrambulalulintas.utils.Constants

class GetNameActivity : AppCompatActivity() {

    private lateinit var binding: ActivityGetNameBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityGetNameBinding.inflate(layoutInflater)
        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.setBackgroundResource(R.drawable.bg_button)
    binding.btnStart.setBackgroundResource(R.drawable.bg_button)
} else if (idLevel == 2) {
    binding.clBackground.setBackgroundResource(R.drawable.bg_level2)
    binding.btnStart.setBackgroundResource(R.drawable.bg_level2)
} else if (idLevel == 3) {
    binding.clBackground.setBackgroundResource(R.drawable.bg_level3)
    binding.btnStart.setBackgroundResource(R.drawable.bg_level3)
} else if (idLevel == 4) {
    binding.clBackground.setBackgroundResource(R.drawable.bg_level4)
    binding.btnStart.setBackgroundResource(R.drawable.bg_level4)
} else {
    binding.clBackground.setBackgroundResource(R.drawable.bg_level5)
    binding.btnStart.setBackgroundResource(R.drawable.bg_level5)
}

binding.btnStart.setOnClickListener {
    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_NAME,
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.AppCompatActivity
```

```
import android.os.Bundle
```

```
import android.os.CountDownTimer
```

```
import android.view.View
```

```
import android.widget.*
```

```
import android.widget.SeekBar.OnSeekBarChangeListener
```

```
import androidx.core.content.ContextCompat
```

```
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.FirebaseFirestore
```

```
class KuisActivity : AppCompatActivity(), OnSeekBarChangeListener {
```

```

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(savedInstanceState)
    binding = ActivityKuisBinding.inflate(layoutInflater)
    setContentView(binding.root)

    binding.sbTime.max = 10

    binding.sbTime.progress = 10

    binding.sbTime.setOnSeekBarChangeListener(this)

```

```

timer = object : CountdownTimer((binding.sbTime.progress *
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_NAME, 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.setOnClickListener {
    if (!isAnswerChecked) {
        val anyAnswerIsChecked = selectedAlternativeIndex != -1
        if (!anyAnswerIsChecked) {
            Toast.makeText(this, "Please, pilih jawaban anda",
                Toast.LENGTH_SHORT).show()
        } else {
            val currentQuestion = questionsList[currentQuestionIndex]
            if (
                selectedAlternativeIndex == currentQuestion.jawabanBenarIndex
            ) {
                answerView(tvAlternatives!![selectedAlternativeIndex],
                    R.drawable.correct_option_border_bg
                )
                totalScore++
                timer.cancel()
            } else {
                kesalahan++
            }
        }
    }
}

```

```

timer.cancel()

answerView(tvAlternatives!![selectedAlternativeIndex],
    R.drawable.wrong_option_border_bg
)

answerView(tvAlternatives!![currentQuestion.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_NAME, 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_NAME, 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[currentQuestionIndex].teks_Soal
```

```
    // Render Question Image
```

```
binding.ivImage.setImageResource(questionsList[currentQuestionIndex].gambar)
```

```
    // progressBar
```

```

binding.progressBar.progress = currentQuestionIndex + 1

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

for (alternativeIndex in
questionsList[currentQuestionIndex].alternative.indices) {
    tvAlternatives!![alternativeIndex].text =
questionsList[currentQuestionIndex].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.typeface, Typeface.BOLD)
option.background = ContextCompat.getDrawable(
    this@KuisActivity,
    R.drawable.selected_option_border_bg
)
}

private fun answerView(view: TextView, drawableId: Int) {
    view.background = ContextCompat.getDrawable(
        this@KuisActivity,
        drawableId
    )
    tvAlternatives!![selectedAlternativeIndex].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.AppCompatActivity
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(savedInstanceState)
        binding = ActivityLevelBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }
}

```

```
binding.clLevel1.setOnClickListener {  
    val idLevel1 = 1  
    val question1 = 3  
    val tvLevel = "Level 1"  
    val tvDescLevel = "Newbie"  
    val intent = Intent(this,GetNameActivity::class.java)  
    intent.putExtra("idLevel",idLevel1)  
    intent.putExtra("question",question1)  
    intent.putExtra("level",tvLevel)  
    intent.putExtra("desc",tvDescLevel)  
    startActivity(intent)  
}
```

```
binding.clLevel2.setOnClickListener {  
    val idLevel2 = 2  
    val question2 = 5  
    val tvLevel = "Level 2"  
    val tvDescLevel = "Continuing"  
    val intent = Intent(this,GetNameActivity::class.java)  
    intent.putExtra("idLevel",idLevel2)  
    intent.putExtra("question",question2)  
    intent.putExtra("level",tvLevel)  
    intent.putExtra("desc",tvDescLevel)  
    startActivity(intent)  
}
```

```
binding.btnLevel3.setOnClickListener {  
    val idLevel3 = 3  
    val question3 = 7  
    val tvLevel = "Level 3"  
    val tvDescLevel = "Medium"  
    val intent = Intent(this,GetNameActivity::class.java)  
    intent.putExtra("idLevel",idLevel3)  
    intent.putExtra("question",question3)  
    intent.putExtra("level",tvLevel)  
    intent.putExtra("desc",tvDescLevel)  
    startActivity(intent)  
}
```

```
binding.btnLevel4.setOnClickListener {  
    val idLevel4 = 4  
    val question4 = 9  
    val tvLevel = "Level 4"  
    val tvDescLevel = "Pro"  
    val intent = Intent(this,GetNameActivity::class.java)  
    intent.putExtra("idLevel",idLevel4)  
    intent.putExtra("question",question4)  
    intent.putExtra("level",tvLevel)  
    intent.putExtra("desc",tvDescLevel)  
    startActivity(intent)  
}
```

```

binding.btnLevel5.setOnClickListener {
    val idLevel5 = 5
    val question5 = 11
    val tvLevel = "Level 5"
    val tvDescLevel = "Legend"
    val intent = Intent(this,GetNameActivity::class.java)
    intent.putExtra("idLevel",idLevel5)
    intent.putExtra("question",question5)
    intent.putExtra("level",tvLevel)
    intent.putExtra("desc",tvDescLevel)
    startActivity(intent)
}

}

}

```

```

package com.example.kuisrambulalulintas.ui.activities

```

```

import androidx.appcompat.app.AppCompatActivity

```

```

import android.os.Bundle

```

```

import com.example.kuisrambulalulintas.R

```

```

class MateriActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        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.AlertDialog
```

```
import androidx.appcompat.app.AppCompatActivity
```

```
import androidx.core.content.ContextCompat
```

```
import com.bumptech.glide.Glide
```

```
import com.example.kuisrambulalulintas.R
```

```
import com.example.kuisrambulalulintas.databinding.ActivityQuestionBinding
```

```
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.FirebaseFirestore
import dagger.hilt.android.AndroidEntryPoint

@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(savedInstanceState)
```

```
    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.setOnSeekBarChangeListener(this)
```

```
    timer = object : CountdownTimer((binding.sbTime.progress *  
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_NAME, 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.setOnClickListener {
    if (!isAnswerChecked) {
        val anyAnswerIsChecked = selectedAlternativeIndex != -1
        if (!anyAnswerIsChecked) {
            Toast.makeText(this, "Please, pilih jawaban anda",
                Toast.LENGTH_SHORT).show()
        } else {
            val currentQuestion = questionsList[currentQuestionIndex]
            if (
                selectedAlternativeIndex == currentQuestion.jawaban
            ) {
                answerView(tvAlternatives!![selectedAlternativeIndex],
                    R.drawable.correct_option_border_bg
                )
                totalScore++
                timer.cancel()
            } else {
                kesalahan++
                timer.cancel()
                answerView(tvAlternatives!![selectedAlternativeIndex],
                    R.drawable.wrong_option_border_bg
                )
                answerView(tvAlternatives!![currentQuestion.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_NAME, 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_NAME, userName)
        intent.putExtra(Constants.TOTAL_Soals, question!!)
        intent.putExtra(Constants.SCORE, totalScore)
        startActivity(intent)
        finish()
    }

    isChecked = false
}
}

tvAlternatives?.let {
    for (optionIndex in it.indices) {
        it[optionIndex].let {
            it.setOnClickListener {
                if (!isChecked) {
                    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", "$questionsList")
        }

        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[currentQuestionIndex].image)
    .into(binding.ivImage)

// progressBar
//binding.ivImage.setImage(questionsList[currentQuestionIndex].image)
binding.progressBar.max = question!!
binding.progressBar.progress = currentQuestionIndex + 1

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

        for (alternativeIndex in
questionsList[currentQuestionIndex].pilihan!!.indices) {
            tvAlternatives!![alternativeIndex].text =
questionsList[currentQuestionIndex].pilihan?.get(alternativeIndex)!!
        }

        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.typeface, Typeface.BOLD)
    option.background = ContextCompat.getDrawable(
        this@QuestionActivity,
        R.drawable.selected_option_border_bg
    )
}

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

```

```

    )
    tvAlternatives!![selectedAlternativeIndex].setTextColor(
        Color.parseColor("#FFFFFF")
    )
}

```

```
@SuppressWarnings("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.AppCompatActivity
import android.os.Bundle
import com.example.kuisrambulalulintas.MainActivity
import com.example.kuisrambulalulintas.databinding.ActivityResultBinding
import com.example.kuisrambulalulintas.utils.Constants

class ResultActivity : AppCompatActivity() {

    private lateinit var binding : ActivityResultBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityResultBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val userName = intent.getStringExtra(Constants.USER_NAME)
        val totalQuestions = intent.getIntExtra(Constants.TOTAL_SoalS, 0)
        val score = intent.getIntExtra(Constants.SCORE, 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.setOnClickListener{

```

```
        val intent = Intent(this, QuestionActivity::class.java)
        intent.putExtra("question",totalQuestions)
        intent.putExtra(Constants.USER_NAME,userName)
        startActivity(intent)
        finish()
    }
```

```
        binding.btnMenu.setOnClickListener{
            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.navArgs
import com.example.kuisrambulalulintas.R
```

```

import com.example.kuisrambulalulintas.databinding.FragmentMateriBinding
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(inflater, container, false)

        val currentGambar = rambuLaranganList[selectIndex]

```

```

if (args.rambu == 1){
    gambarList = rambuLaranganList
    binding.clBackground.setBackgroundResource(R.drawable.bg_larangan)
} else if (args.rambu == 2){
    gambarList = rambuPeringatanList

binding.clBackground.setBackgroundResource(R.drawable.bg_peringatan)
}
else if (args.rambu == 3){
    gambarList = rambuPerintahList
    binding.clBackground.setBackgroundResource(R.drawable.bg_perintah)
} else {
    gambarList = rambuPetunjukList
    binding.clBackground.setBackgroundResource(R.drawable.bg_petunjuk)
}

updateGambar(selectIndex)

binding.btnNext.setOnClickListener {
    if (selectIndex < gambarList!!.size - 1) {
        selectIndex++
        updateGambar(selectIndex)

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

```

```

    }

    binding.btnPrev.setOnClickListener {
        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].gambar)
    binding.tvKeterangan.text = gambarList!![selectIndex].keterangan
}
}

```



```

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.findNavController
import com.example.kuisrambulalulintas.R

import
com.example.kuisrambulalulintas.databinding.FragmentMenuMateriBinding

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.inflate(inflater, container,
false)

        binding.cvLarangan.setOnClickListener {
            val data =
MenuMateriFragmentDirections.actionMenuMateriFragmentToMateriFragment(1
)

```

```

        findNavController().navigate(data)
    }

    binding.cvPeringatan.setOnClickListener {
        val data =
            MenuMateriFragmentDirections.actionMenuMateriFragmentToMateriFragment(2
            )
        findNavController().navigate(data)
    }

    binding.cvPerintah.setOnClickListener {
        val data =
            MenuMateriFragmentDirections.actionMenuMateriFragmentToMateriFragment(3
            )
        findNavController().navigate(data)
    }

    binding.cvPetunjuk.setOnClickListener {
        val data =
            MenuMateriFragmentDirections.actionMenuMateriFragmentToMateriFragment(4
            )
        findNavController().navigate(data)
    }

    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,
            arrayOf("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.hatihatihati,
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,
    arrayOf("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,
    arrayOf("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,
    arrayOf("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,
    arrayOf("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,
```

```

        arrayOf("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,
        arrayOf("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_parkir)
    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.drawable.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_berjalan_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.larangan8_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_batasakhir)

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.larangan12_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.larangan14_belok_kanan)

gambarList.add(g14)

val g15 = Gambar(15,"Larangan kecepatan kendaraan lebih dari 40 km perjam
(Rambu Larangan).",R.drawable.larangan15_kecepatan40)

gambarList.add(g15)

val g16 = Gambar(16,"Larangan masuk bagi
becak.",R.drawable.larangan16_becak)

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.larangan18_delman)

gambarList.add(g18)

val g19 = Gambar(19,"Larangan masuk bagi gerobak dan
dokar.",R.drawable.larangan19_gerobak)

gambarList.add(g19)

val g20 = Gambar(20,"Larangan masuk bagi gerobak dorong dan
sejenisnya.",R.drawable.larangan20_gerobak_dorong)

gambarList.add(g20)

val g21 = Gambar(21,"Larangan masuk bagi gerobak pedati dan sejenisnya
(Rambu Larangan).",R.drawable.larangan21_batasakhir2)

gambarList.add(g21)

val g22 = Gambar(22,"Larangan masuk bagi kendaraan bermotor dan
mobil.",R.drawable.larangan22_mobil_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_kereta_tempel)

gambarList.add(g24)

val g25 = Gambar(25,"Larangan masuk bagi kendaraan bermotor roda empat
atau lebih.",R.drawable.larangan25_kendaraan_roda_empat)

gambarList.add(g25)

val g26 = Gambar(26,"Larangan masuk bagi kendaraan bermotor roda
tiga.",R.drawable.larangan26_rodatisiga)

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_mobil_barang)

gambarList.add(g28)

val g29 = Gambar(29,"Larangan masuk bagi pejalan
kaki.",R.drawable.larangan29_pejalan_kaki)

gambarList.add(g29)

val g30 = Gambar(30,"Batas kecepatan maksimum 40 km
jam.",R.drawable.larangan30_batas_kecepatan)

gambarList.add(g30)

val g31 = Gambar(31,"Larangan masuk bagi semua kendaraan bermotor
maupun tidak bermotor dari kedua
arah.",R.drawable.larangan31_semuakendaraan_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_sepeda)

gambarList.add(g33)

val g34 = Gambar(34,"Larangan masuk bagi kendaraan dengan lebar lebih dari
2,7 m.",R.drawable.larangan34_lebar27m)

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.larangan36_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.larangan37_3a)

gambarList.add(g37)

val g38 = Gambar(38,"Larangan masuk bagi kendaraan dengan tinggi lebih
dari 4,5 m.",R.drawable.larangan38_45meter)

gambarList.add(g38)

val g39 = Gambar(39,"Larangan masuk bagi kendaraan tidak bermotor dengan
panjang lebih dari ... m.",R.drawable.larangan39_panjang)

gambarList.add(g39)

val g40 = Gambar(40,"Larangan masuk bagi sepeda dan
becak.",R.drawable.larangan40_becak_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_suara)

    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.larangan45_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.peringatan2_banyak_pejalan)

    gambarList.add(g2)

    val g3 = Gambar(3,"banyak anak-
anak",R.drawable.peringatan3_banyak_anak)

    gambarList.add(g3)

```

```

    val g4 = Gambar(4,"banyak satwa jinak dan sering
menyebrang",R.drawable.peringatan4_satwa_jinak)

    gambarList.add(g4)

    val g5 = Gambar(5,"banyak satwa liar dan sering
menyebrang",R.drawable.peringatan5_satwa_liar)

    gambarList.add(g5)

    val g6 = Gambar(6,"banyak
tikungan.",R.drawable.peringatan6_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_hatihati)

    gambarList.add(g8)

    val g9 = Gambar(9,"jalan
bergelombang.",R.drawable.peringatan9_jalan_bergelombang)

    gambarList.add(g9)

    val g10 = Gambar(10,"jalan
cekung.",R.drawable.peringatan10_jalan_cekung)

    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_jalan_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_kerikil_lepas)

```

```

gambarList.add(g15)

val g16 = Gambar(16,"lalu lintas dua arah",R.drawable.peringatan15_lalulintas_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.peringatan18_ruang_bebas)

gambarList.add(g18)

val g19 = Gambar(19,"lintasan pesawat terbang.",R.drawable.peringatan19_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.drawable.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_penyebrangan_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.peringatan28_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.peringatan30_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_persimpangantiga_serong_kiri)

gambarList.add(g35)

val g36 = Gambar(36,"persimpangan tiga sisi
kiri.",R.drawable.peringatan36_persimpangantiga_kiri)

gambarList.add(g36)

```

```

    val g37 = Gambar(37,"persimpangan tiga type
T.",R.drawable.peringatan37_persimpangantiga_typed)
    gambarList.add(g37)
    val g38 = Gambar(38,"persimpangan tiga type
Y.",R.drawable.peringatan38_persimpangantiga_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_tikungan_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_tikungan_kekiri)
    gambarList.add(g42)
    val g43 = Gambar(43,"tikungan tajam ke
kanan",R.drawable.peringatan43_tikungan_tajam_kanan)
    gambarList.add(g43)
    val g44 = Gambar(44,"tinggi ruang
bebas",R.drawable.peringatan45_tinggi_ruang_bebas)
    gambarList.add(g44)
    val g45 = Gambar(45,"turunan .",R.drawable.peringatan45_turunan)
    gambarList.add(g45)
    val g46 = Gambar(46,"turunan terjal
.",R.drawable.peringatan46_turunan_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_jalan_satu_arah_kanan)  
    gambarList.add(g3)  
  
    val g4 = Gambar(4,"Lajur atau bagian jalan yang wajib  
    dilewati",R.drawable.perintah4_jalur_wajib_dilewati)  
    gambarList.add(g4)  
  
    val g5 = Gambar(5,"Perintah kecepatan minimum yang  
    diwajibkan",R.drawable.perintah5_kecepatan_minimun)  
    gambarList.add(g5)  
  
    val g6 = Gambar(6,"Wajib berjalan lurus  
    kedepan.",R.drawable.perintah6_wajib_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_wajib_jalur_kanan)
```

```

gambarList.add(g9)

val g10 = Gambar(10,"Wajib Mengikuti Arah Ke Kiri.",R.drawable.perintah10_wajib_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_wajib_mengikuti_salahsatu_jalur)

gambarList.add(g12)

val g13 = Gambar(13,"Wajib mengikuti salah satu arah yang ditunjuki.",R.drawable.perintah13_wajib_mengikuti_salahsatu_jalur)

gambarList.add(g13)

val g14 = Gambar(14,"Wajib untuk lalu lintas pedati.",R.drawable.perintah14_wajib_pedati)

gambarList.add(g14)

val g15 = Gambar(15,"Wajib untuk lalulintas becak.",R.drawable.perintah15_wajib_becak)

gambarList.add(g15)

val g16 = Gambar(16,"Wajib untuk lalulintas bersepeda",R.drawable.perintah16_wajib_sepeda)

gambarList.add(g16)

val g17 = Gambar(17,"Wajib untuk lalulintas dokar.",R.drawable.perintah17_wajib_dokar)

gambarList.add(g17)

val g18 = Gambar(18,"Wajib untuk lalulintas pedati, gerobak dorong dan dokar.",R.drawable.perintah18_wajib_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_wajib_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_bandara)
    gambarList.add(g2)
    val g3 = Gambar(3,"bengkel",R.drawable.petunjuk3_bengkel)
    gambarList.add(g3)
    val g4 = Gambar(4,"gelanggang olahraga",R.drawable.petunjuk4_olahraga)
    gambarList.add(g4)
    val g5 = Gambar(5,"gereja",R.drawable.petunjuk5_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_jalan_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_jalur_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_jembatan_timbang)

gambarList.add(g11)

val g12 = Gambar(12,"kamp
pengungsian",R.drawable.petunjuk12_kamp_pengungsian)

gambarList.add(g12)

val g13 = Gambar(13,"kantor pos",R.drawable.petunjuk13_kantor_pos)

gambarList.add(g13)

val g14 = Gambar(14,"kemah
karvan",R.drawable.petunjuk14_kemah_karvan)

gambarList.add(g14)

val g15 = Gambar(15,"kemah tenda",R.drawable.petunjuk15_kemah_tenda)

gambarList.add(g15)

val g16 = Gambar(16,"kendaraan umum selain bus dan
taksi",R.drawable.petunjuk16_kendaraan_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_kolam_renang)

gambarList.add(g18)

val g19 = Gambar(19,"lapangan olahraga
terbuka",R.drawable.petunjuk19_lapangan_olahraga)

gambarList.add(g19)

```

```

val g20 = Gambar(20,"lokasi berkumpul
darurat",R.drawable.petunjuk20_lokasi_berkumpul_darurat)
gambarList.add(g20)
val g21 = Gambar(21,"masjid",R.drawable.petunjuk21_masjid)
gambarList.add(g21)
val g22 = Gambar(22,"museum",R.drawable.petunjuk22_museum)
gambarList.add(g22)
val g23 = Gambar(23,"pantai",R.drawable.petunjuk23_pantai)
gambarList.add(g23)
val g24 = Gambar(24,"Pasar",R.drawable.petunjuk24_pasar)
gambarList.add(g24)
val g25 = Gambar(25,"pelabuhan",R.drawable.petunjuk25_pelabuhan)
gambarList.add(g25)
val g26 = Gambar(26,"penginapan",R.drawable.petunjuk26_penginapan)
gambarList.add(g26)
val g27 = Gambar(27,"perkemahan",R.drawable.petunjuk27_perkemahan)
gambarList.add(g27)
val g28 = Gambar(28,"perpustakaan",R.drawable.petunjuk28_perpustakaan)
gambarList.add(g28)
val g29 = Gambar(29,"pompa bahan bakar",R.drawable.petunjuk29_pompa_bahan_bakar)
gambarList.add(g29)
val g30 = Gambar(30,"Pura",R.drawable.petunjuk30_pura)
gambarList.add(g30)
val g31 = Gambar(31,"pusat pengungsian",R.drawable.petunjuk31_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_rumah_makan)
gambarList.add(g33)

val g34 = Gambar(34,"rumah sakit",R.drawable.petunjuk34_rumah_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_letusan_gunung)
gambarList.add(g36)

val g37 = Gambar(37,"rute evakuasi tsunami",R.drawable.petunjuk37_tsunami)
gambarList.add(g37)

val g38 = Gambar(38,"rute penjelajah alam",R.drawable.petunjuk38_penjelajah_alam)
gambarList.add(g38)

val g39 = Gambar(39,"sekolah",R.drawable.petunjuk39_sekolah)
gambarList.add(g39)

val g40 = Gambar(40,"stadion",R.drawable.petunjuk40_stadion)
gambarList.add(g40)

val g41 = Gambar(41,"stand taksi",R.drawable.petunjuk41_stand_taksi)
gambarList.add(g41)

val g42 = Gambar(42,"stasiun kereta api",R.drawable.petunjuk42_stasiun_keretaapi)
gambarList.add(g42)

val g43 = Gambar(43,"stasiun uji berkala",R.drawable.petunjuk43_stasiun_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.petunjuk45_taman)
gambarList.add(g45)

val g46 = Gambar(46,"telpon umum",R.drawable.petunjuk46_telepon_umum)
gambarList.add(g46)

val g47 = Gambar(47,"tempat pembuangan akhir",R.drawable.petunjuk47_penuangan_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_tempat_penyebrangan)
gambarList.add(g49)

val g50 = Gambar(50,"tempat putar balik arah",R.drawable.petunjuk50_putar_balik_arah)
gambarList.add(g50)

val g51 = Gambar(51,"terminal bus",R.drawable.petunjuk51_terminal_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_terowongan_bermula)
gambarList.add(g41)

val g55 = Gambar(55,"vila",R.drawable.petunjuk55_vila)

```

```

gambarList.add(g55)

val g56 = Gambar(56,"warung kopi",R.drawable.petunjuk56_warung_kopi)
gambarList.add(g56)

val g57 = Gambar(57,"Whara",R.drawable.petunjuk57_whara)
gambarList.add(g57)

val g58 = Gambar(58,"Zona Parkir",R.drawable.petunjuk58_zona_patkir)
gambarList.add(g58)

val g59 = Gambar(59,"zona parkir untuk penyadangan
cacat",R.drawable.petunjuk59_zona_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(savedInstanceState)
    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)

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

    //startService(Intent(this, BackgroundSoundService::class.java))

    binding.cvPlay.setOnClickListener {
        startActivity(Intent(this, LevelActivity::class.java))
    }

    binding.cvMusic.setOnClickListener {
        if (mPlay){
            mPlay = false
            status = 1

```

```

editor.apply {
    putInt("status", status!!)
}
val statusOFF = sharedPref.getInt("status",status!!)
status = statusOFF
binding.tvMusic.text = "MUSIC : OFF"
stopService(Intent(this, BackgroundSoundService::class.java))
} else {
    mPlay = true
    status = 0
    editor.apply {
        putInt("status", status!!)
    }
    val statusOFF = sharedPref.getInt("status",status!!)
    status = statusOFF
    binding.tvMusic.text = "MUSIC : ON"
    startService(Intent(this, BackgroundSoundService::class.java))
}
}

```

```

binding.cvMateri.setOnClickListener {
    startActivity(Intent(this,MateriActivity::class.java))
}

```

```

binding.cvExit.setOnClickListener {
    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",DialogInterface.OnClickListener { _, _ ->
        finish()
    })

    .setNegativeButton("keluar",DialogInterface.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.music)
        mediaPlayer!!.isLooping = true
        mediaPlayer!!.start()
    } else {
        mediaPlayer!!.start()
    }
}*/
}

```

```

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

    <application
        android:name=".KuisApp"
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.KuisRambuLaluLintas"
        tools:targetApi="31">

        <activity

```

```

        android:name=".ui.activities.QuestionActivity"
        android:exported="false">
        <meta-data
            android:name="android.app.lib_name"
            android:value="" />
    </activity>
    android:name=".ui.activities.MateriActivity"
    android:theme="@style/AppTheme"
    android:exported="false">
    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
    </activity>
    <activity
        android:name=".ui.activities.LevelActivity"
        android:exported="false">
        <meta-data
            android:name="android.app.lib_name"
            android:value="" />
    </activity>
    <activity
        android:name=".ui.activities.GetNameActivity"
        android:theme="@style/AppTheme"
        android:exported="false">
        <meta-data
            android:name="android.app.lib_name"
            android:value="" />

```

```
</activity>
```

```
<activity
```

```
    android:name=".ui.activities.KuisActivity"
```

```
    android:exported="false">
```

```
    <meta-data
```

```
        android:name="android.app.lib_name"
```

```
        android:value="" />
```

```
</activity>
```

```
<activity
```

```
    android:name=".ui.activities.ResultActivity"
```

```
    android:theme="@style/AppTheme"
```

```
    android:exported="false">
```

```
    <meta-data
```

```
        android:name="android.app.lib_name"
```

```
        android:value="" />
```

```
</activity>
```



```
<activity
    android:name=".MainActivity"
    android:theme="@style/AppTheme"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

    <meta-data
        android:name="android.app.lib_name"
        android:value="" />
</activity>

<service android:name=".service.BackgroundSoundService"
    />
<meta-data
    android:name="preloaded_fonts"
    android:resource="@array/preloaded_fonts" />

</application>
```

```
</manifest>
```



**KARTU MONITORING BIMBINGAN**  
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI  
HASIL

Mahasiswa : Muh. Rizal Syam	Pembimbing I : Ade Hastuty, S.T., S.Kom., M.T.
NIM : 218280156	Pembimbing II : Wahyuddin, S.Kom., M.Kom.
Judul Skripsi : APLIKASI EDUCATION GAME DALAM PEMBELAJARAN INTERAKTIF PENGENALAN RAMBU LALULINTAS PADA ANAK SD BERBASIS MOBILE LEARNING	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Abstrak berisi ringkasan bab 1 - bab 5		Konsultasi 1 - level 1 = 5 - level 2 = 10 -	
Konsultasi 2 Flowchart di revisi setelah star maka gunakan simbol deklarasi setelah itu simbol input dan seterusnya		Konsultasi 2 - History	
Konsultasi 3 kesimpulan berisikan jawaban dari rumusan masalah		Konsultasi 3 ACC seminar hasil	
Konsultasi 4		Konsultasi 4 ACC ujian tutup	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

**Perhatian :**

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

