

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3**



ANDROID NAVIGATION

Oleh:

Hari Octavian Delrossi

NIM. 2210817210033

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
JUNI 2024**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN MOBILE
MODUL 3

Laporan Praktikum Pemrograman Mobile Modul 2: Android Navigation ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Hari Octavian Delrossi
NIM : 2210817210033

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Asandy Putra
NIM. 2110817310002

Muti'a Maulida, S.Kom., M.T.I.
NIP. 198810272019032013

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
SOAL PRAKTIKUM.....	6
A. Source Code	7
B. Output Program.....	14
C. Pembahasan	16
Tautan Git	18

DAFTAR GAMBAR

Gambar 1. Screenshot Saat Menjawab Quiz	14
Gambar 2. Screenshot Saat Field Input Dibiarkan Kosong dan Disubmut	15
Gambar 3. Screenshot Menampilkan Hasil Score Akhir Setelah 10 Soal Terjawab.....	15

DAFTAR TABEL

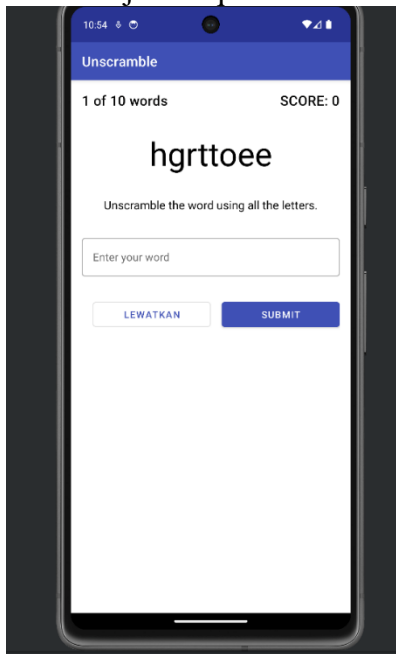
Tabel 1. Source Code Hasil Jawaban MainActivity.kt.....	7
Tabel 2. Source Code Hasil Jawaban GameFragment.kt	8
Tabel 3. Source Code Hasil Jawaban GameViewModel.kt.....	10
Tabel 4. Source Code Hasil Jawaban ListofWords.kt	13

SOAL PRAKTIKUM

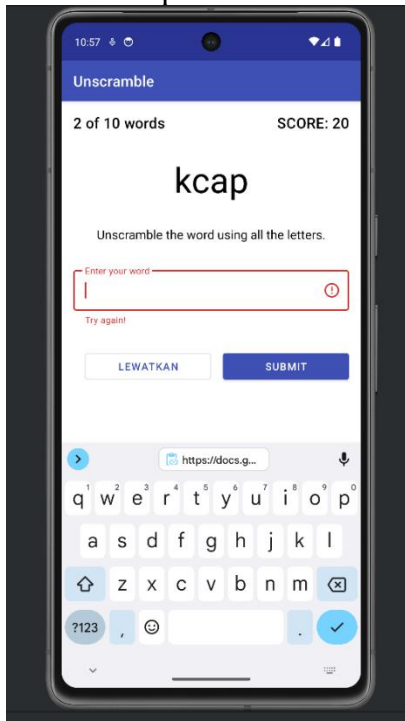
Buat sebuah aplikasi Android sederhana yang disebut "Unscramble" yang tampilannya menggunakan bahasa Indonesia. Aplikasi ini akan menampilkan kata-kata yang diacak sebanyak 10 soal, dan pengguna harus menebak kata yang benar. Soal yang tidak bisa dijawab bisa dilewati dan setiap satu soal yang benar bernilai 20.

Contoh Hasil Aplikasi :

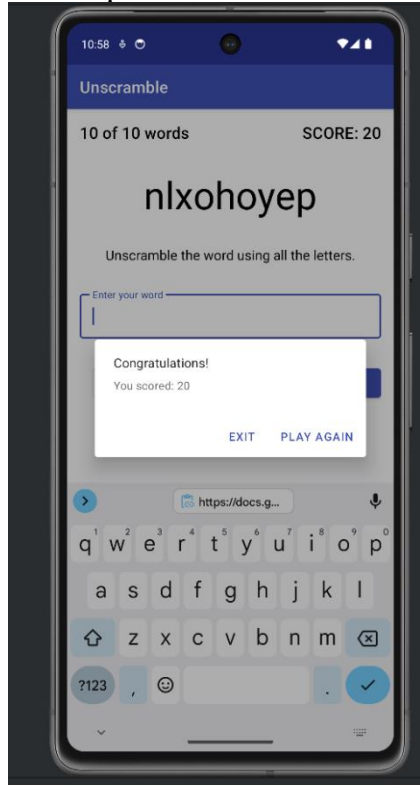
1. Saat menjawab quiz



2. Saat field input dibiarkan kosong dan disubmit



3. Menampilkan hasil score akhir setelah 10 soal terjawab



A. Source Code

Tabel 1. Source Code Hasil Jawaban MainActivity.kt

1	package com.example.android.unscramble
2	
3	import android.os.Bundle
4	import androidx.appcompat.app.AppCompatActivity
5	
6	class MainActivity : AppCompatActivity() {
7	
8	override fun onCreate(savedInstanceState: Bundle?)
9	{
10	super.onCreate(savedInstanceState)
11	setContentView(R.layout.main_activity)
12	}

Tabel 2. Source Code Hasil Jawaban GameFragment.kt

```
1 package com.example.android.unscramble.ui.game
2
3 import android.os.Bundle
4 import android.view.LayoutInflater
5 import android.view.View
6 import android.view.ViewGroup
7 import androidx.databinding.DataBindingUtil
8 import androidx.fragment.app.Fragment
9 import androidx.fragment.app.viewModels
10 import com.example.android.unscramble.R
11 import
12     com.example.android.unscramble.databinding.GameFragmentBinding
13     import
14     com.google.android.material.dialog.MaterialAlertDialogBuilder
15
16 class GameFragment : Fragment() {
17
18     private lateinit var binding: GameFragmentBinding
19
20     private val viewModel: GameViewModel by viewModels()
21
22     override fun onCreateView(
23         inflater: LayoutInflater, container: ViewGroup?,
24         savedInstanceState: Bundle?
25     ): View {
26
27         binding = DataBindingUtil.inflate(inflater,
28             R.layout.game_fragment, container, false)
29         return binding.root
30     }
31
32     override fun onViewCreated(view: View, savedInstanceState:
33         Bundle?) {
34         super.onViewCreated(view, savedInstanceState)
35
36         binding.gameViewModel = viewModel
37         binding.maxNoOfWords = MAX_NO_OF_WORDS
38
39         binding.lifecycleOwner = viewLifecycleOwner
```



```

37         binding.submit.setOnClickListener { onSubmitWord() }
38         binding.skip.setOnClickListener { onSkipWord() }
39     }
40
41     private fun onSubmitWord() {
42         val playerWord = binding.textInputEditText.text.toString()
43
44         if (viewModel.isUserWordCorrect(playerWord)) {
45             setErrorTextField(false)
46             if (!viewModel.nextWord()) {
47                 showFinalScoreDialog()
48             }
49         } else {
50             setErrorTextField(true)
51         }
52     }
53
54     private fun onSkipWord() {
55         if (viewModel.nextWord()) {
56             setErrorTextField(false)
57         } else {
58             showFinalScoreDialog()
59         }
60     }
61
62     private fun showFinalScoreDialog() {
63         MaterialAlertDialogBuilder(requireContext())
64             .setTitle(getString(R.string.congratulations))
65             .setMessage(getString(R.string.you_scored,
66 viewModel.score.value))
67             .setCancelable(false)
68             .setNegativeButton(getString(R.string.exit)) { _, _ ->
69                 exitGame()
70             }
71             .setPositiveButton(getString(R.string.play_again)) { _, _ ->
72                 restartGame()
73             }
74             .show()

```

74	}	
75		
76	private fun restartGame() {	
77	viewModel.reinitializeData()	
78	setErrorTextField(false)	
79	}	
80		
81	private fun exitGame() {	
82	activity?.finish()	
83	}	
84		
85	private fun setErrorTextField(error: Boolean) {	
86	if (error) {	
87	binding.textField.isErrorEnabled = true	
88	binding.textField.error	=
	getString(R.string.try_again)	
89	} else {	
90	binding.textField.isErrorEnabled = false	
91	binding.textInputEditText.text = null	
92	}	
93	}	
94	}	

Tabel 3. Source Code Hasil Jawaban GameViewModel.kt

1	package com.example.android.unscramble.ui.game
2	
3	import android.text.Spannable
4	import android.text.SpannableString
5	import android.text.style.TtsSpan
6	import android.util.Log
7	import androidx.lifecycle.LiveData
8	import androidx.lifecycle.MutableLiveData
9	import androidx.lifecycle.Transformations
10	import androidx.lifecycle.ViewModel
11	
12	class GameViewModel : ViewModel() {
13	private val _score = MutableLiveData(0)
14	val score: LiveData<Int>
15	get() = _score
16	
17	private val _currentWordCount = MutableLiveData(0)

```

18     val currentWordCount: LiveData<Int>
19         get() = _currentWordCount
20
21     private val _currentScrambledWord =
22     MutableLiveData<String>()
23     val currentScrambledWord: LiveData<Spannable> =
24     Transformations.map(_currentScrambledWord) {
25         if (it == null) {
26             SpannableString("")
27         } else {
28             val scrambledWord = it.toString()
29             val spannable: Spannable =
30             SpannableString(scrambledWord)
31             spannable.setSpan(
32             TtsSpan.VerbatimBuilder(scrambledWord).build(),
33             0,
34             scrambledWord.length,
35             Spannable.SPAN_INCLUSIVE_INCLUSIVE
36         )
37         spannable
38     }
39
40     private var wordsList: MutableList<String> =
41     mutableListOf()
42     private lateinit var currentWord: String
43
44     init {
45         getNextWord()
46     }
47
48     private fun getNextWord() {
49         currentWord = allWordsList.random()
50         val tempWord = currentWord.toCharArray()
51         tempWord.shuffle()
52
53         while (String(tempWord).equals(currentWord,
54         false)) {
55             tempWord.shuffle()
56         }

```

```

53         if (wordsList.contains(currentWord)) {
54             getNextWord()
55         } else {
56             Log.d("Unscramble",           "currentWord="
$currentWord")
57             _currentScrambledWord.value           =
String(tempWord)
58             _currentWordCount.value           =
_currentWordCount.value?.inc()
59             wordsList.add(currentWord)
60         }
61     }
62
63     fun reinitializeData() {
64         _score.value = 0
65         _currentWordCount.value = 0
66         wordsList.clear()
67         getNextWord()
68     }
69
70     private fun increaseScore() {
71         _score.value           =
_score.value?.plus(SCORE_INCREASE)
72     }
73
74     fun isUserWordCorrect(playerWord: String): Boolean {
75         if (playerWord.equals(currentWord, true)) {
76             increaseScore()
77             return true
78         }
79         return false
80     }
81
82     fun nextWord(): Boolean {
83         return if (_currentWordCount.value!! <
MAX_NO_OF_WORDS) {
84             getNextWord()
85             true
86         } else false
87     }
88 }

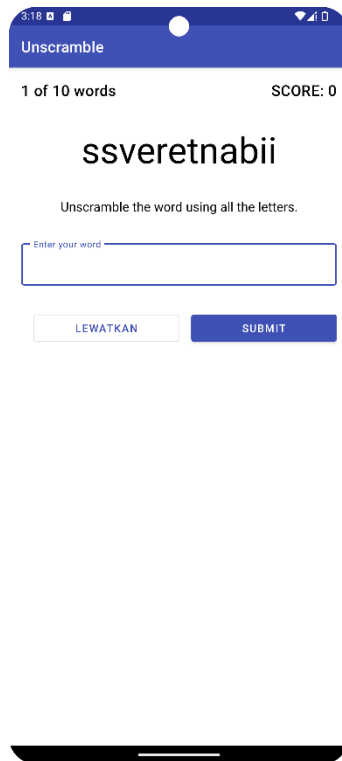
```

Tabel 4. Source Code Hasil Jawaban ListofWords.kt

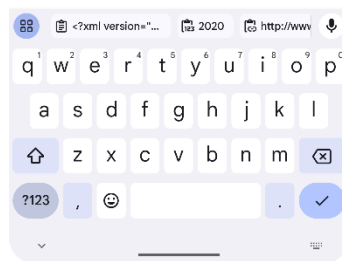
1	package com.example.android.unscramble.ui.game
2	
3	const val MAX_NO_OF_WORDS = 10
4	const val SCORE_INCREASE = 20
5	
6	val allWordsList: List<String> =
7	listOf("hewan", "mobil", "anekdot", "alfabet",
	"semua", "hebat", "bangkit", "balon", "keranjang",
	"bangku", "terbaik", "ulang tahun", "buku", "tas
	kerja", "kamera", "berkemah",
8	"lilin", "kucing", "kembang kol", "obrolan",
	"anak-anak", "kelas", "klasik", "ruang kelas", "kopi",
	"berwarna-warni", "kue", "kreatif", "pelayaran",
	"menari", "siang hari",
9	"dinosaur", "kenop pintu", "makan malam",
	"mimpi", "senja", "makan", "gajah", "zamrud",
	"mengerikan", "listrik", "selesai", "bunga",
	"mengikuti", "rubah", "bingkai",
10	"bebas", "sering", "corong", "hijau", "gitar",
	"belanja", "gelas", "hebat", "tertawa kecil", "potong
	rambut", "setengah", "buatan sendiri", "terjadi",
	"madu", "bergegas",
11	"seratus", "es", "iglo", "berinvestasi",
	"mengundang", "ikon", "memperkenalkan", "lelucon",
	"ceria", "jurnal", "melompat", "bergabung", "kanguru",
	"papan ketik", "dapur", "koala",
12	"baik hati", "kaleidoskop", "pemandangan",
	"terlambat", "tertawa", "belajar", "lemon", "surat",
	"lili", "majalah", "laut", "permen marshmallow",
	"labirin", "bermeditasi", "melodi",
13	"menit", "monumen", "bulan", "sepeda motor",
	"gunung", "musik", "utara", "hidung", "malam", "nama",
	"tidak pernah", "bernegosiasi", "nomor", "berlawanan",
	"gurita", "oak",
14	"pesanan", "buka", "kutub", "kemas",
	"lukisan", "orang", "piknik", "bantak", "pizza",
	"podcast", "presentasi", "anak anjing", "teka-teki",
	"resep", "melepaskan", "restoran",
15	"berputar", "mundur", "ruangan", "lari",
	"rahasia", "benih", "kapal", "kemeja", "seharusnya",

16	"kecil", "pesawat luar angkasa", "melihat bintang", "keterampilan", "jalan", "gaya", "matahari terbit", "taksi", "rapi", "pengatur waktu", "bersama", "gigi", "turis", "bepergian", "truk", "di bawah", "berguna", "unicorn", "unik", "meningkatkan", "seragam", "vas", "biola", "pengunjung", "visi",
17	"volume", "pemandangan", "walrus", "berkelana", "dunia", "musim dingin", "baik", "angin puyuh", "sinar-X", "xilofon", "yoga", "yogurt", "yoyo", "kamu", "tahun", "lezat", "zebra",
18	"zigzag", "zoologi", "zona", "semangat")

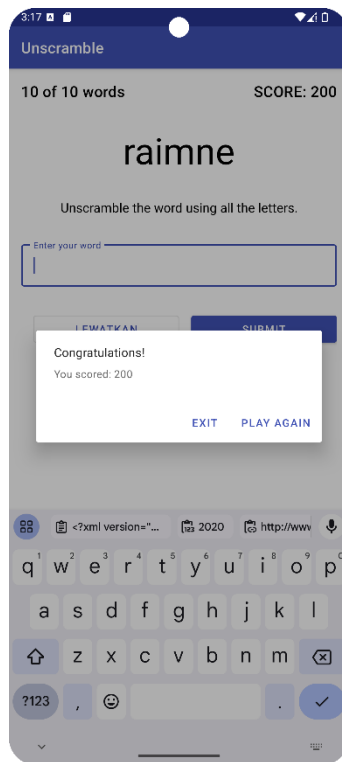
B. Output Program



Gambar 1. Screenshot Saat Menjawab Quiz



Gambar 2. Screenshot Saat Field Input Dibiarkan Kosong dan Disubmut



Gambar 3. Screenshot Menampilkan Hasil Score Akhir Setelah 10 Soal Terjawab

C. Pembahasan

- MainActivity.kt

MainActivity merupakan subclass dari AppCompatActivity, yang adalah kelas dasar untuk aktivitas dengan dukungan penuh terhadap fitur-fitur modern pada perangkat Android. Dalam metode onCreate, yang dipanggil saat aktivitas pertama kali dibuat, aktivitas ini menginisialisasi dirinya sendiri dan menetapkan tampilan konten menggunakan file layout main_activity dari resource R.layout. super.onCreate(savedInstanceState) memastikan bahwa aktivitas mengurus inisialisasi bawaan yang ditangani oleh AppCompatActivity.

- GameFragment.kt

Sebuah fragmen dalam aplikasi Android yang menampilkan antarmuka permainan. GameFragment mewarisi kelas Fragment dan menggunakan Data Binding untuk menghubungkan data dengan UI. Di dalam onCreateView, fragmen menginisialisasi binding menggunakan DataBindingUtil.inflate untuk mengembalikan root view yang ditetapkan dalam R.layout.game_fragment. Di dalam onViewCreated, view model (GameViewModel) dihubungkan dengan binding dan beberapa properti seperti maxNoOfWords diatur. Metode onSubmitWord memeriksa apakah kata yang dimasukkan pengguna benar, dan jika benar, lanjut ke kata berikutnya atau tampilkan dialog skor akhir jika permainan selesai. Metode onSkipWord memungkinkan pengguna melewati kata dan juga dapat memicu dialog skor akhir. showFinalScoreDialog menampilkan dialog akhir permainan dengan opsi untuk keluar atau bermain lagi. restartGame mengatur ulang data permainan dan menghapus pesan kesalahan, sementara exitGame mengakhiri aktivitas. setErrorTextField mengatur status pesan kesalahan pada input pengguna.

- GameViewModel.kt

GameViewModel adalah kelas yang bertindak sebagai ViewModel untuk permainan "Unscramble" dalam aplikasi Android. Kelas ini menggunakan LiveData untuk menyimpan dan memantau perubahan data permainan seperti skor, jumlah kata saat ini, dan kata yang diacak. MutableLiveData digunakan untuk memodifikasi nilai-nilai ini, sementara LiveData dipakai untuk memberikan akses baca saja dari data tersebut kepada UI. Transformations.map mengubah kata yang diacak menjadi Spannable, yang berguna untuk penanganan teks khusus seperti Text-to-Speech.

Dalam inisialisasinya, getNextWord dipanggil untuk mengambil kata acak dari allWordsList, mengacak hurufnya, dan memastikan kata acak tersebut tidak sama dengan kata aslinya. Kata yang baru kemudian disimpan dan daftar kata yang telah digunakan diperbarui.

Metode reinitializeData mengatur ulang skor, jumlah kata, dan daftar kata yang telah digunakan, kemudian memanggil getNextWord untuk memulai permainan baru. increaseScore meningkatkan skor ketika pengguna menebak kata dengan benar. isUserWordCorrect memeriksa apakah kata yang dimasukkan pengguna benar, dan jika benar, skor ditingkatkan. nextWord menentukan apakah permainan harus dilanjutkan dengan kata berikutnya atau tidak, berdasarkan batas jumlah kata yang telah ditetapkan.

- ListofWords.kt

Mendefinisikan dua konstanta dan sebuah daftar kata untuk permainan "Unscramble". Konstanta MAX_NO_OF_WORDS menetapkan jumlah maksimum kata yang akan digunakan dalam satu permainan sebagai 10, sedangkan SCORE_INCREASE menentukan peningkatan skor sebanyak 20 poin untuk setiap kata yang berhasil ditebak dengan benar. Daftar allWordsList berisi berbagai kata dalam bahasa Indonesia yang akan diacak dan digunakan dalam permainan, seperti "hewan", "mobil", "kamera", dan "musik". Konstanta dan daftar kata ini digunakan oleh GameViewModel untuk mengelola logika permainan, termasuk penilaian dan penyediaan kata-kata acak kepada pengguna.

Tautan Git

[Praktikum-Pemrograman-Mobile/Praktikum Modul 3 at main · harioct/Praktikum-Pemrograman-Mobile \(github.com\)](#)