

**METODOLOGI DESAIN PERANGKAT LUNAK PRAKTIK**  
**RAPID APPLICATION DEVELOPMENT**  
**“SISTEM PEMESANAN TIKET KERETA”**



Disusun oleh :

1. 5200411156 Agus Ardiyanto
2. 5200411172 - isa mukti widodo
3. 5200411178 Muhamad Ali Nugroho Ramadhan

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI & ELEKTRO**  
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2021**

## PENDAHULUAN

### 1.1. Latar Belakang

Salah satu perkembangan yang paling pesat saat ini adalah perkembangan teknologi mobile atau telepon seluler. Telepon seluler sebagai salah satu maha karya anak manusia dalam bidang komunikasi, memang tidak dapat terlepas dari kehidupan masyarakat modern seperti sekarang. Dengan harga murah dan fungsinya yang sangat penting, maka tidak mengherankan telepon seluler menjadi primadona dalam masyarakat baik itu masyarakat kelas atas hingga bawah. Telepon seluler atau yang sering kita sebut dengan handphone mampu berkomunikasi dengan content yang tidak terbatas pada suara, melainkan gambar, film, atau sembarang data digital dapat dilewatkan melalui jalur komunikasi seluler. Lebih jauh teknologi seluler memungkinkan handphone untuk berkomunikasi dengan internet dalam memberikan informasi yang cepat dan efisien.

Penerapan Teknologi Informasi (TI) saat ini telah menyebar hampir di semua bidang tidak terkecuali di bidang transportasi darat khususnya kereta api. Transportasi darat dengan kereta api sebagai salah satu alat transportasi yang cepat dan murah, banyak menjadi pilihan utama masyarakat umum untuk aktifitas bepergian sehari-hari. Namun sebagai salah satu alat transportasi yang digemari masyarakat modern, sistem pemberitahuan jadwal serta pemesanan tiket kereta masih menggunakan metode lama yaitu data perjalanan yang dituliskan pada papan pengumuman, sehingga sistem yang berjalan kurang efisien. Meskipun saat ini telah ada sistem pemberitahuan melalui media cetak, namun di rasa masih memiliki kekurangan di beberapa hal. Sedangkan untuk pemesanan tiket sendiri, user memiliki beberapa alternatif yaitu melakukan pemesanan tiket pemesanan secara langsung, pemesanan melalui telepon atau pemesanan melalui agen-agen. Namun sistem tersebut juga kurang efisien karena topologi daerah Indonesia yang luas sedangkan letak stasiun yang jauh dari tempat tinggal membuat waktu yang dibutuhkan serta biaya yang 3 diperlukan tidak sedikit.

Sadar akan pentingnya sarana informasi perjalanan kereta yang mudah, murah dan cepat tanpa harus membuang-buang waktu yang ada. Membuat peneliti berupaya dalam menghasilkan sistem yang dapat menampilkan informasi perjalanan kereta di Daerah Operasi 6 ke layar handphone. Karena handphone merupakan teknologi penting dalam kehidupan sekarang, sehingga memungkinkan user selalu dapat online dimanapun dan kapanpun.

## 1.2. Tujuan Penelitian

Berdasarkan latar belakang masalah yang kami kemukakan di atas, timbul berbagai masalah antara lain :

1. Bagaimana mengimplementasikan suatu sistem informasi ke handphone Android sehingga mampu memberikan informasi perjalanan kereta yang lengkap dan detail.
2. Bagaimana membangun aplikasi yang mampu memberikan informasi melainkan juga mampu melakukan pemesanan ke server.
3. Bagaimana menggabungkan teknologi Android dengan bahasa pemrograman PHP dan database MySQL.
4. Bagaimana merancang suatu aplikasi sistem informasi yang sederhana, mudah digunakan, bermanfaat serta mudah digunakan oleh user.

## 1.3 Batasan Masalah

Dalam suatu penelitian perlu adanya suatu batasan agar penelitian tersebut dapat terarah dan memudahkan dalam pembahasan sehingga tujuan penelitian dapat tercapai. Beberapa batasan yang digunakan dalam penelitian adalah sebagai berikut :

1. Analisis, perancangan, serta implementasi yang di bahas dalam sistem mencakup informasi kereta, stasiun tujuan, jadwal keberangkatan, dan tanggal keberangkatan
2. Aplikasi ini memberikan informasi perjalanan kereta kelas Eksekutif dan Bisnis.
3. Proses pengambilan tiket pada sistem ini, calon penumpang dapat mencetaknya di stasiun asal keberangkatan kereta, sebelum hari keberangkatan atau pada saat akan melakukan keberangkatan dengan cara memasukkan kode booking atau scan qr pada printer stasiun.

## PEMODELAN BISNIS

2.1 Jenis sistem : Android

2.2 Bahasa pemrograman : Kotlin

2.3 Kebutuhan hardware :

a. Perangkat mobile Android

b. RAM 1 GB

c. koneksi ke internet

2.4 Kebutuhan software :

a. Sistem operasi Android

b. android > api 5.0

c. MySql

d. bahasa pemrograman kotlin

e. bahasa JSON

2.5 Kebutuhan input :

a. Data Penumpang

b. Data Kereta

c. Data Stasiun

d. Data Pemesanan

e. Data Pembayaran

2.6 Kebutuhan Informasi :

a. Data Tiket

b. Kode Booking

c. QR Code

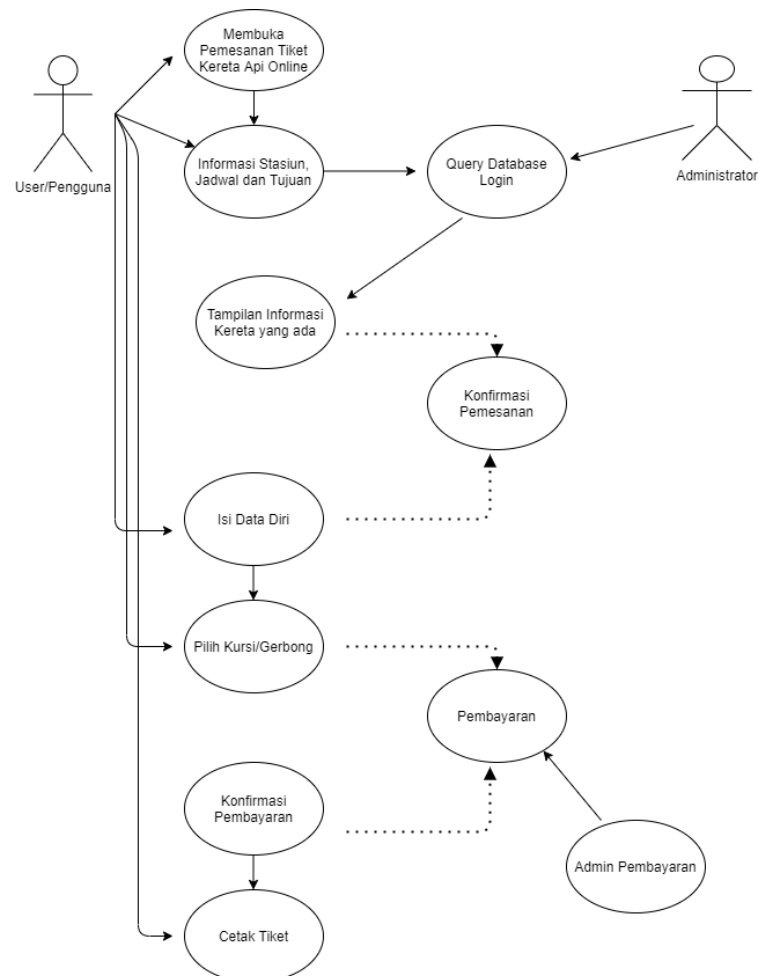
## PEMODELAN DATA

### 3.1 Arsitektur Sistem

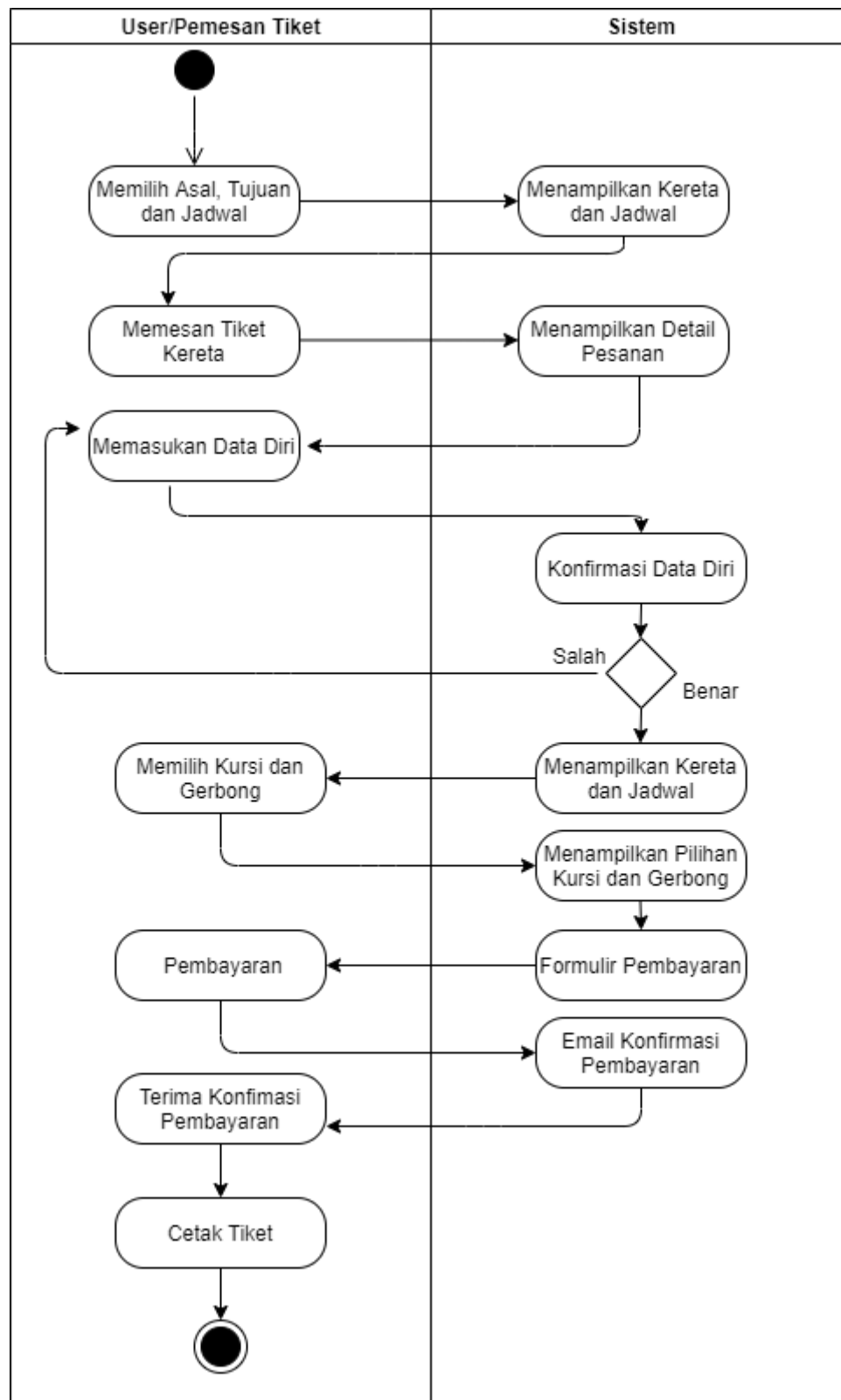
Database yang digunakan yaitu MySQL dan WebServer yang digunakan adalah Apache



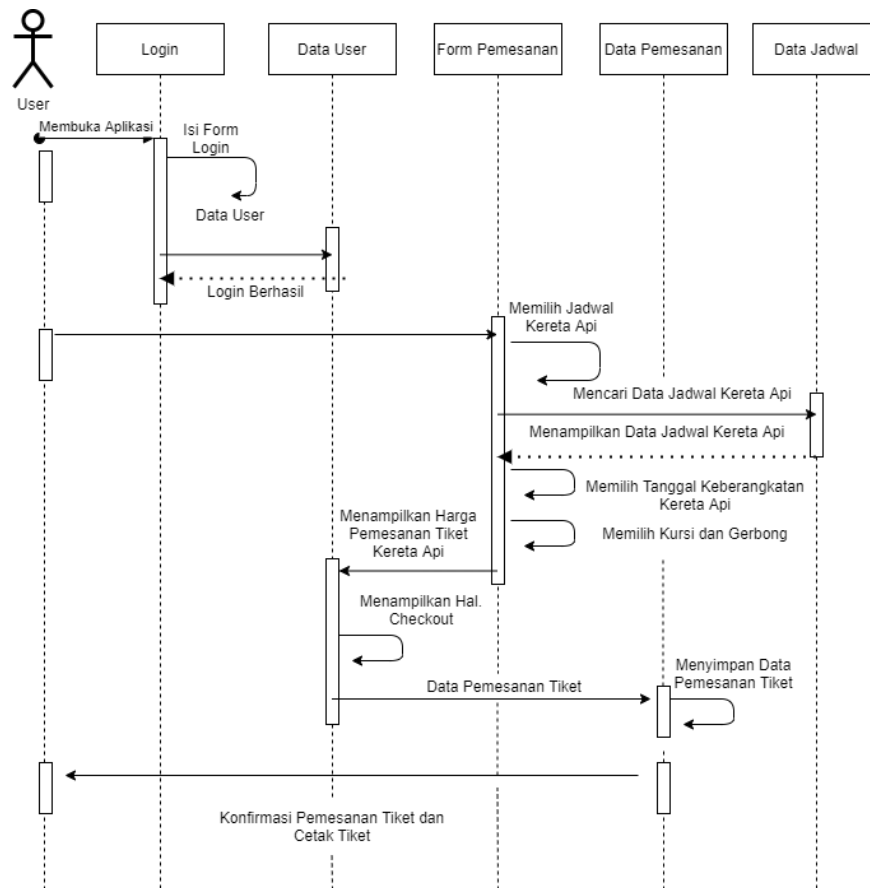
### 3.2 Use Case Diagram



### 3.3 Activity Diagram



### 3.4 Sequence Diagram

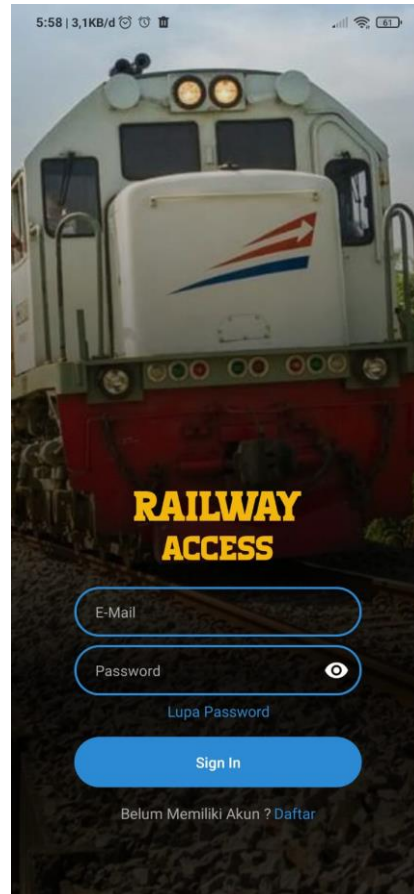


### 3.5 Class Diagram



## PEMBENTUKAN APLIKASI

4.1 Berikut tampilan antramuka yang dirancang untuk user melakukan login



Kode Program :

```
button.setOnClickListener {
    var valid = true
    if (txtEmail.text.toString()=="") {
        valid = false
        txtEmail.setError("Harap masukkan Email")
    }
    if (txtPass.text.toString()=="") {
        valid = false
        txtPass.setError("Harap masukkan password")
    }
    if (valid) {
        if (email == (txtEmail.text.toString()) &&
password == (txtPass.text.toString())) {
            val intent = Intent(this,
TiketActivity::class.java)

            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK or
Intent.FLAG_ACTIVITY_CLEAR_TASK)
```



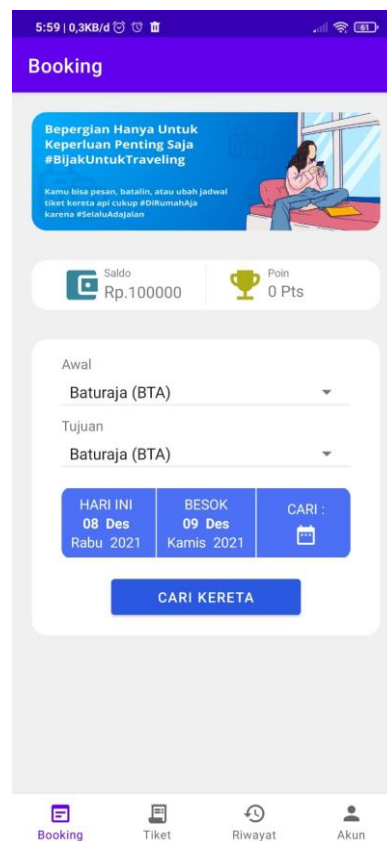
```

        startActivity(intent)
    } else {
        Toast.makeText(this, "username atau
password salah", Toast.LENGTH_SHORT).show()
    }
}
}

```

Apabila email dan password yang dimasukkan benar maka pengguna akan diarahkan pada TicketActivity

4.2 Pada menu ini pengguna akan memilih stasiun awal dan stasiun tujuan serta tanggal keberangkatannya



Kode program :

```

@RequiresApi(Build.VERSION_CODES.M)
override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    // Inflate the layout for this fragment
}

```

```

        val view = inflater.inflate(R.layout.fragment_booking,
        container, false)
        presenter = Presenter(this)
        presenter.getSaldo()
        (activity as AppCompatActivity).supportActionBar?.title
= "Booking"

        val spinner = view.findViewById(R.id.spinAwal) as
Spinner
        val adapter = ArrayAdapter.createFromResource(activity
as AppCompatActivity, R.array.stasiun,
android.R.layout.simple_spinner_item)

        val darker = (activity as
AppCompatActivity).getColor(R.color.darker)
        val lighter = (activity as
AppCompatActivity).getColor(R.color.lighter)
        val sekarang =
view?.findViewById<LinearLayout>(R.id.sekarang)
        val besok = view?.findViewById<LinearLayout>(R.id.besok)
        val cari = view?.findViewById<LinearLayout>(R.id.cari)

        var isi = 0

        sekarang?.onClick {
            sekarang.backgroundTintList =
ColorStateList.valueOf(lighter)
            besok?.backgroundTintList =
ColorStateList.valueOf(darker)
            cari?.backgroundTintList =
ColorStateList.valueOf(darker)
            val tglSkrng = LocalDate.now()
            val tglToday = DateTimeFormatter.ofPattern("dd MMM
yyyy").format(tglSkrng)
            tanggall = tglToday
            isi += 1
        }
        besok?.onClick {
            sekarang?.backgroundTintList =
ColorStateList.valueOf(darker)
            besok.backgroundTintList =
ColorStateList.valueOf(lighter)
            cari?.backgroundTintList =
ColorStateList.valueOf(darker)
            val tglSkrng = LocalDate.now().plusDays(1)
            val tglToday = DateTimeFormatter.ofPattern("dd MMM
yyyy").format(tglSkrng)
            tanggall = tglToday
            isi += 1
        }
        cari?.onClick {
            sekarang?.backgroundTintList =
ColorStateList.valueOf(darker)

```

```

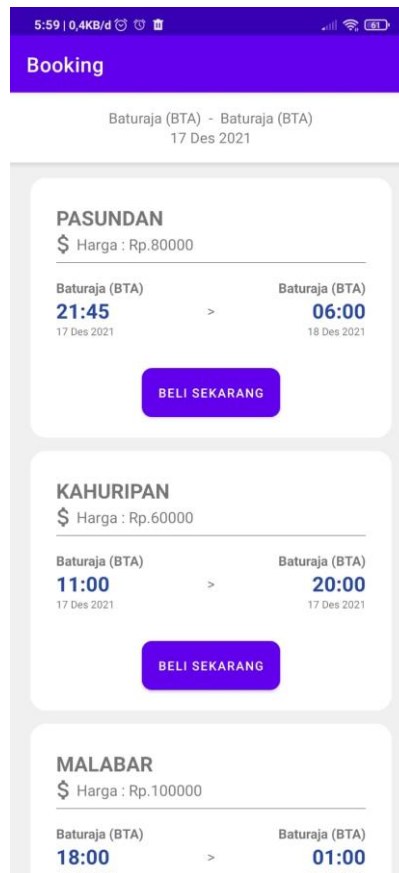
        besok?.backgroundTintList =
ColorStateList.valueOf(darker)
        cari.backgroundTintList =
ColorStateList.valueOf(lighter)
        tanggal.visibility = View.VISIBLE
        startTanggal(view)
        isi += 1
    }

    val btnCari =
view.findViewById<Button>(R.id.btnCari)
    btnCari.onClick {
        if (isi > 0) {
            activity?.let {
                val stasiunAwal =
spinner.selectedItem.toString()
                val stasiunTujuan =
spinnerr.selectedItem.toString()
                val berangkat = tanggal
                val saldo = txtSaldo.text.toString()
                val intent = Intent(activity,
BookActivity::class.java)
                intent.putExtra("stAwal", stasiunAwal)
                intent.putExtra("stTujuan", stasiunTujuan)
                intent.putExtra("tglBerangkat", berangkat)
                intent.putExtra("saldo", saldo)
                (activity as
AppCompatActivity).startActivity(intent)
            }
        }
    }
}

```

Pada kode diatas apabila user sudah memasukkan stasiun awal dan tujuan dan sudah memilih tanggal keberangkatan maka tombol cari akan di enable yang akan membawa user ke BookActivity.

#### 4.3 Tampilan user interface pemilihan tiket



Kode program :

```

beli1.setOnClickListener{

    val nama = findViewById<TextView>(R.id.txtKereta1)
    val price = findViewById<TextView>(R.id.txtHarga1)
    val jamBe = findViewById<TextView>(R.id.txtWaktuBerangkat1)
    val jamSa = findViewById<TextView>(R.id.txtWaktuSampai)
    val kereta = nama.text.toString()
    val harga = price.text.toString()
    val jamBerangkat = jamBe.text.toString()
    val jamSampai = jamSa.text.toString()

    val sb = StringBuilder (6)
    val alphabet =
"qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM12345
67890"
    val rand = Random()
    for (i in 0 until sb.capacity()) {
        val index = rand.nextInt(alphabet.length)
        sb.append((alphabet[index]))
    }
    val kodeBooking = sb.toString()

    val sisa = saldo.toInt() - harga.toInt()
    if (sisa >= 0) {

```

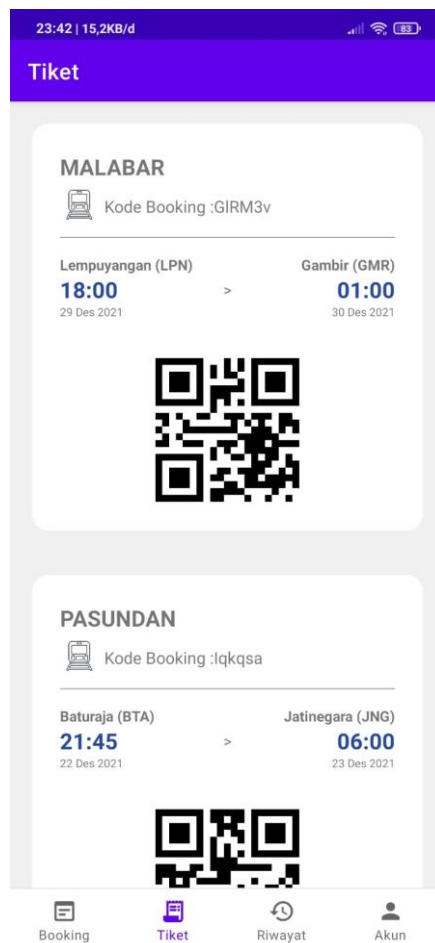
```

        presenter.updateData("1", sisa.toString())
        presenter.addData(kodeBooking, stAwal, stTujuan, tglBerangkat, dt,
        jamBerangkat, jamSampai, kereta)
    }
}

```

Apabila tombol beli ditekan maka data akan ditambahkan kedalam database dan saldo akan diupdate

4.5 Berikutnya yaitu user interface yang memuat tiket yang sudah dibeli



kode program :

```

class FragmentTiket : Fragment(), CrudView {

    private lateinit var presenter: Presenter
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
    }
}

```

```

        presenter = Presenter(this)
        presenter.getData()
        (activity as AppCompatActivity).supportActionBar?.title =
"Tiket"
        return inflater.inflate(R.layout.fragment_tiket,
container, false)
    }

    companion object {
        fun newInstance(): FragmentTiket{
            val fragment = FragmentTiket()
            val args = Bundle()
            fragment.arguments = args
            return fragment
        }
    }

    override fun onSuccessGet(data: List<DataItem>?) {
        rvCategory.adapter = DataAdapter(data)
    }

    override fun onFailedGet(msg: String) {
    }

    override fun onSaldoGet(data: List<DataSaldo>?) {
        TODO("Not yet implemented")
    }
}

```

Data JSON yang didapat dari database dipecah menjadi beberapa variabel yang kemudian ditampilkan dalam View

## PENGUJIAN & TURNOVER

1. Pengujian dilakukan dengan memesan sebuah tiket pada akun yang sudah memiliki saldo



2. Setelah tiket dipesan maka data pada tabel saldo akan berkurang dan data pada tabel tb\_data akan bertambah

```
MariaDB [db_tiket]> show tables;
+-----+
| Tables_in_db_tiket |
+-----+
| tb_data              |
| tb_saldo             |
+-----+
2 rows in set (0.000 sec)

MariaDB [db_tiket]> select * from tb_data;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | kode_booking | stasiun_awal | stasiun_tujuan | tgl_berangkat | tgl_sampai | jam_berangkat | jam_sampai | kereta |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 11 | GLRM3v       | Lempuyangan (LPN) | Gambir (GMR)   | 29 Des 2021   | 30 Des 2021 | 18:00         | 01:00      | MALABAR |
| 12 | Iqkqsa       | Baturaja (BTA)   | Jatinegara (JNG) | 22 Des 2021   | 23 Des 2021 | 21:45         | 06:00      | PASUNDAN |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [db_tiket]> select * from tb_saldo;
+----+-----+
| id | saldo |
+----+-----+
| 1  | 320000 |
+----+-----+
1 row in set (0.000 sec)
```

3. Data dari server harus dikirim dalam bentuk json

```
{
  pesan: "data ada",
  status: 200,
  data: [
    - {
      id: "11",
      kode_booking: "GIRM3v",
      stasiun_awal: "Lempuyangan (LPN)",
      stasiun_tujuan: "Gambir (GMR)",
      tgl_berangkat: "29 Des 2021",
      tgl_sampai: "30 Des 2021",
      jam_berangkat: "18:00",
      jam_sampai: "01:00",
      kereta: "MALABAR"
    },
    - {
      id: "12",
      kode_booking: "Iqkqsa",
      stasiun_awal: "Baturaja (BTA)",
      stasiun_tujuan: "Jatinegara (JNG)",
      tgl_berangkat: "22 Des 2021",
      tgl_sampai: "23 Des 2021",
      jam_berangkat: "21:45",
      jam_sampai: "06:00",
      kereta: "PASUNDAN"
    }
  ]
}
```

4. Data pada tabel tb\_data harus dapat ditampilkan pada FragmentTiket

