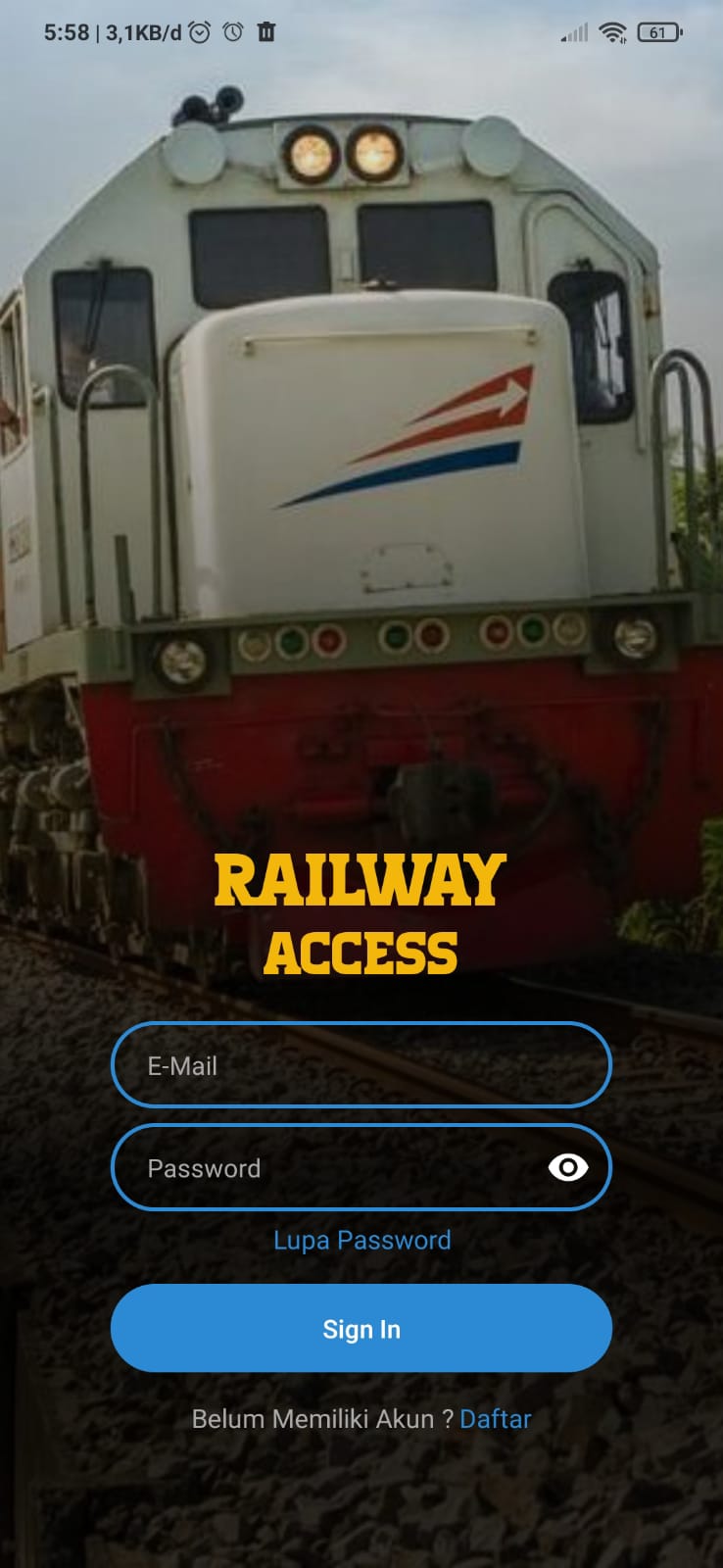
PEMBENTUKAN APLIKASI

4.1 Berikut tampilan antramuka yang dirancang untuk user melakukan login

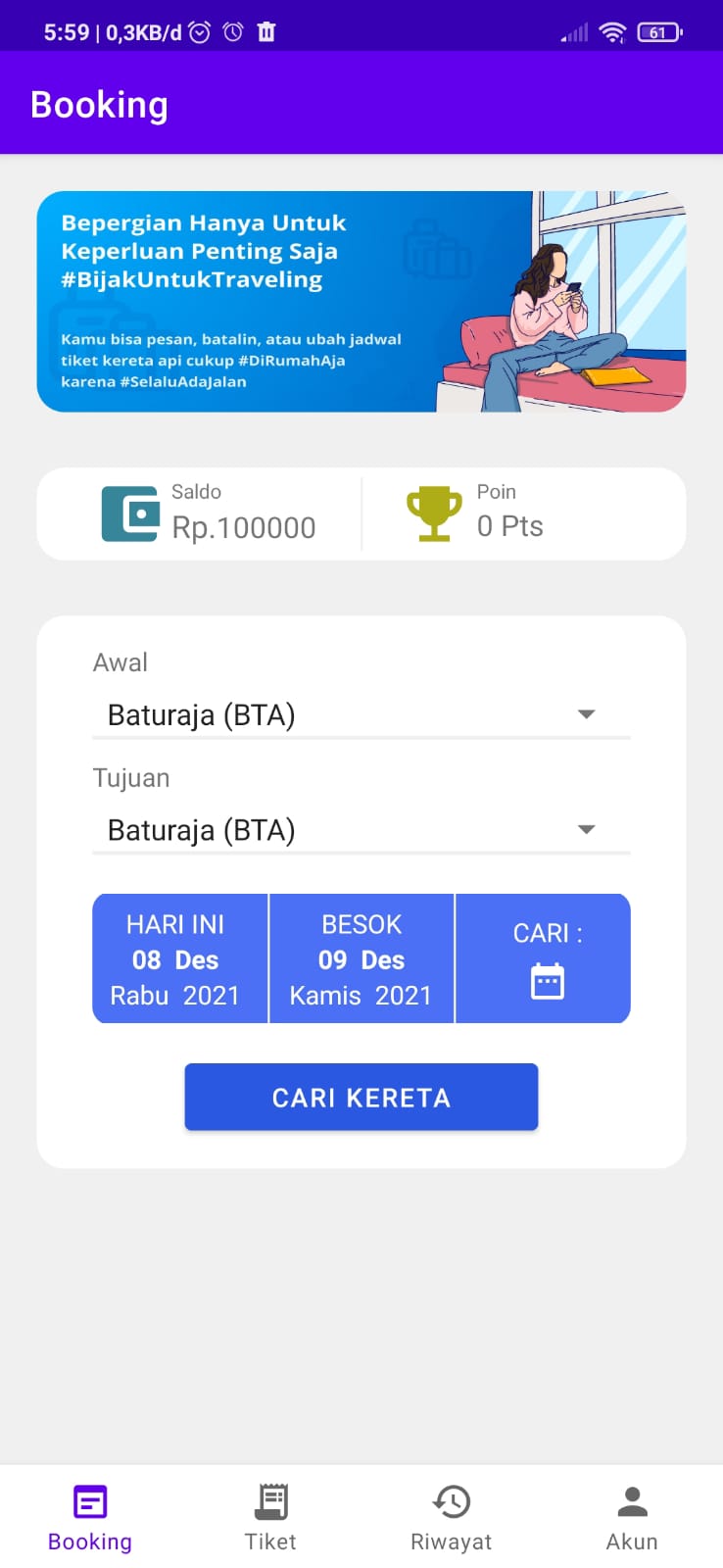


Kode Program :

|  |
| --- |
| 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



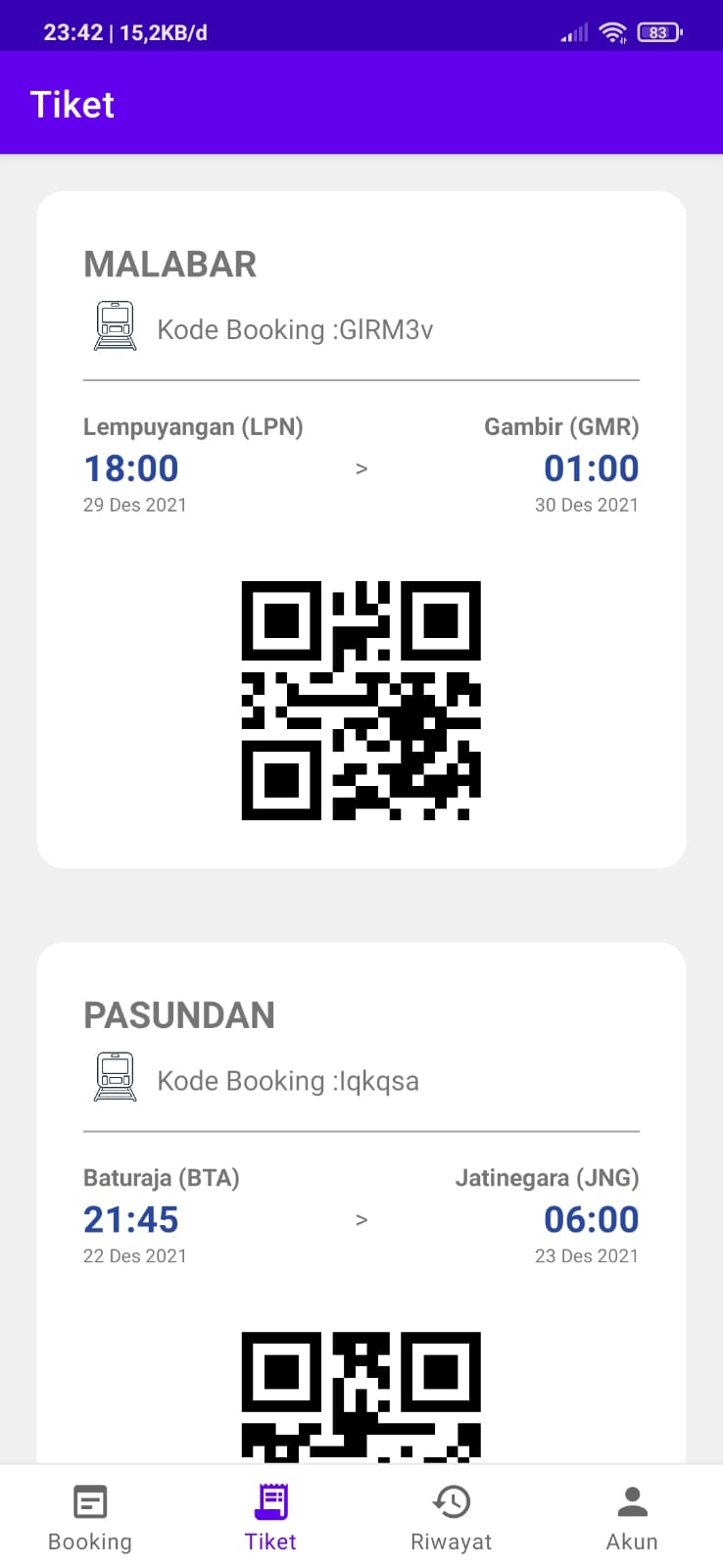
Kode program :

|  |
| --- |
| @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 tglSkrg = LocalDate.now()  val tglToday = DateTimeFormatter.ofPattern("dd MMM yyyy").format(tglSkrg)  tanggall = tglToday  isi += 1  }  besok?.onClick {  sekarang?.backgroundTintList = ColorStateList.valueOf(darker)  besok.backgroundTintList = ColorStateList.valueOf(lighter)  cari?.backgroundTintList = ColorStateList.valueOf(darker)  val tglSkrg = LocalDate.now().plusDays(1)  val tglToday = DateTimeFormatter.ofPattern("dd MMM yyyy").format(tglSkrg)  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 = tanggall  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 kebrangkatan maka tombil 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 = "qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM1234567890"  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