

# BAB 4

## DESKRIPSI PERANCANGAN PERANGKAT LUNAK

### Aplikasi Pemesanan Tiket Kereta


Dipersiapkan oleh:

1301180529 – Dananjaya Cahyo Kusumo

Program Studi Informatika

Fakultas Informatika

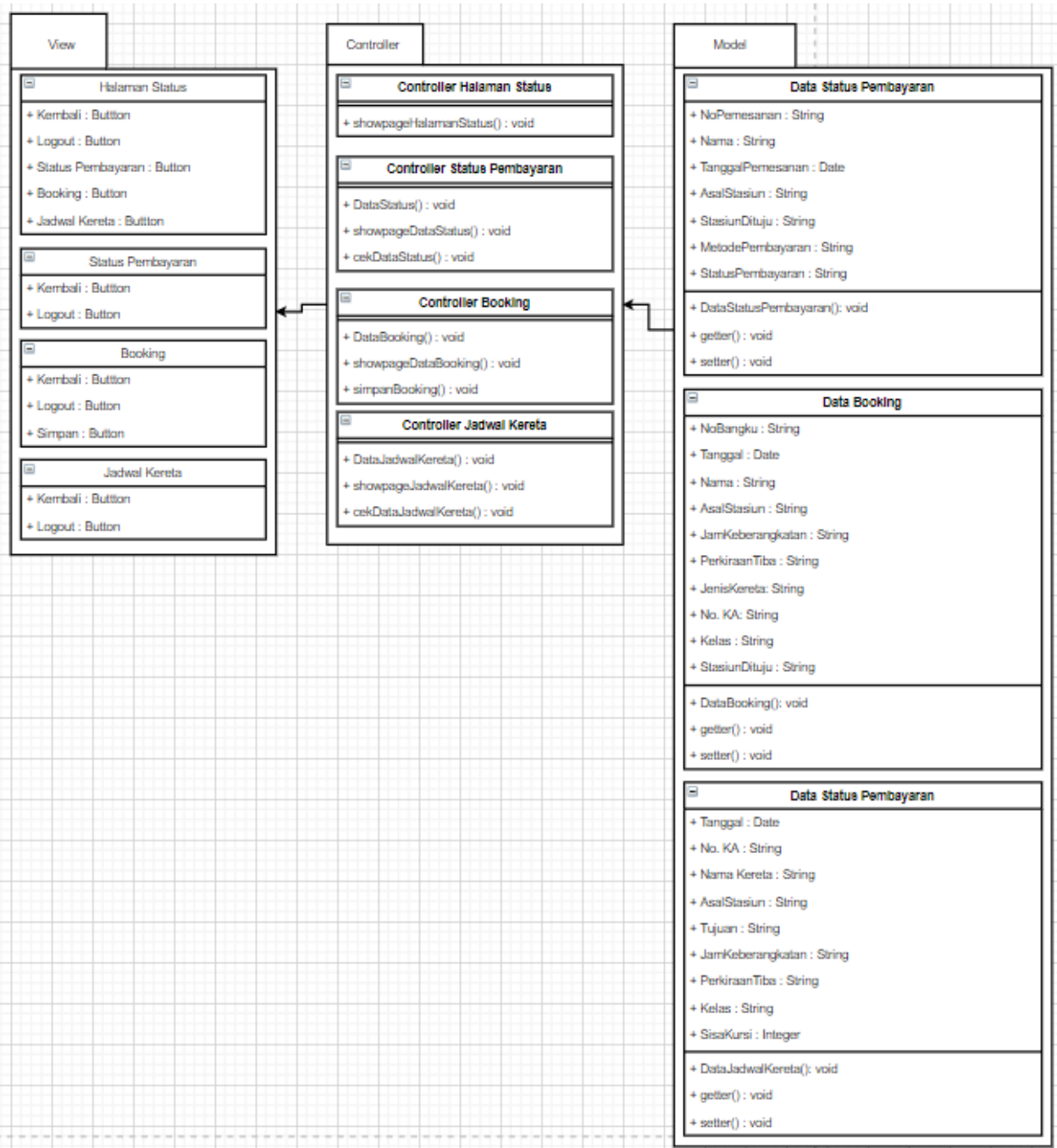
Jl. Telekomunikasi 1, Dayeuhkolot Bandung

	Prodi S1- Informatika Universitas Telkom	Nomor Dokumen		Halaman
		DPPL BAB 4		
		Revisi	0	

## 4 Perancangan Detil

### 4.1 Perancangan Detil Kelas

Bagian ini diisi dengan daftar keseluruhan kelas yang akan digunakan dalam PL menggunakan model MVC



**TABEL KELAS :**

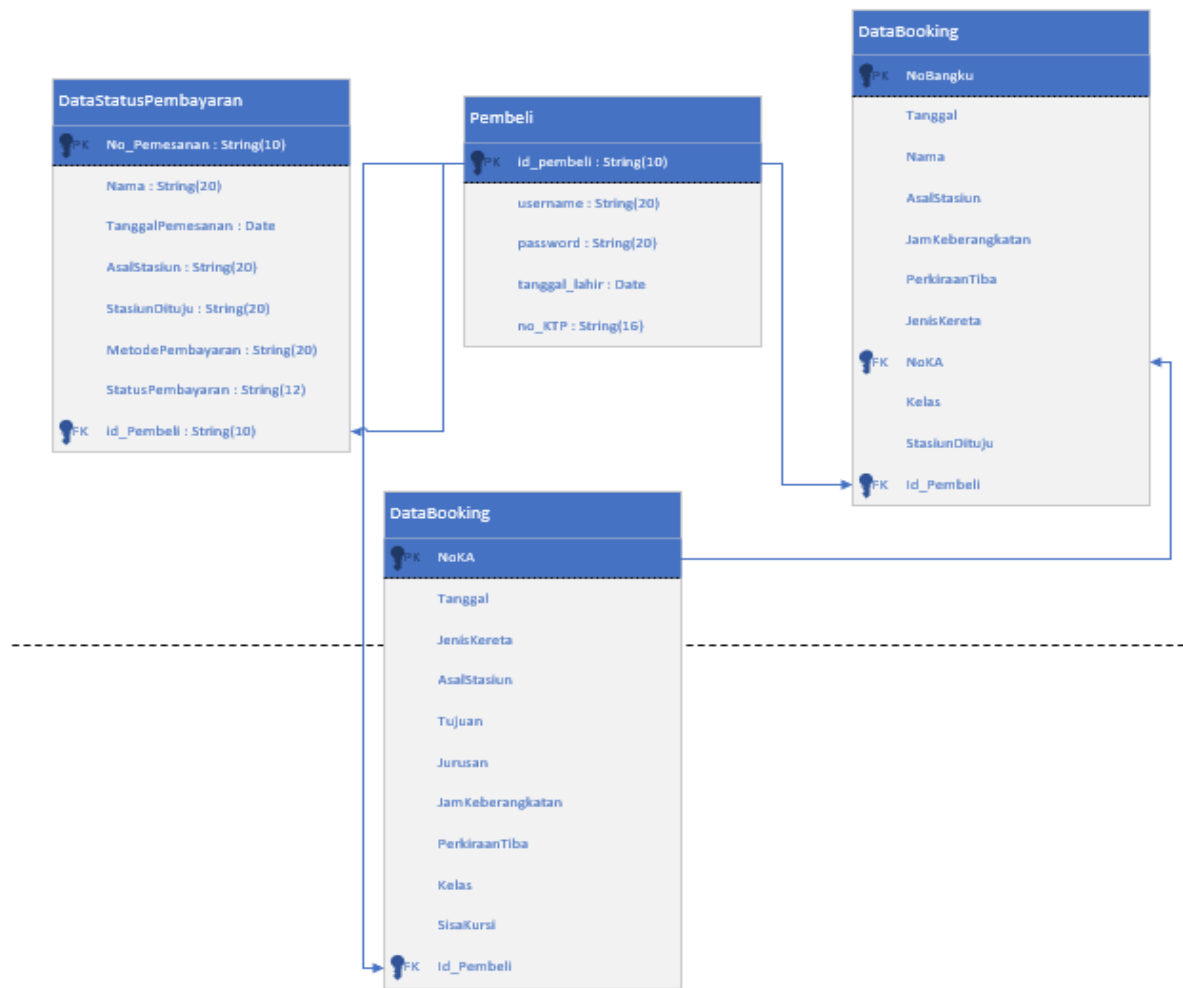
ID Kelas	Nama Kelas Perancangan	Atribute (visibility)	Method / Operation
1.	Data Status Pembayaran	NoPemesanan (private)	DataStatusPembayaran()
		Nama (private)	Getter()
		TanggalPemesanan (private)	Setter()
		AsalStasiun (private)	
		StasiunDituju (private)	
		MetodePembayaran (private)	
		StatusPembayaran	
2.	Data Booking	NoBangku (private)	DataBooking()
		Tanggal (private)	Getter()
		Nama (private)	Setter()
		AsalStasiun (private)	
		JamKeberangkatan (private)	
		PerkiraanTiba (private)	
		JenisKereta (private)	
		Kelas (private)	
		StasiunDItuju (private)	
3.	Data Jadwal Kereta	Tanggal	DataJadwalKereta()
		No. KA	Getter()
		Nama Kereta	Setter()
		Asal Stasiun	
		Tujuan	
		Jurusan	
		Jam Keberangkatan	
		Perkiraan Tiba	
		Kelas	
		Sisa Kursi	

Untuk setiap kelas:

- *identifikasi operasi (mengacu pada tanggung-jawab kelas),*
- *identifikasi atribut, termasuk visibility-nya*

## 4.2 Perancangan Kelas Persistensi (\*\*Basis Data Skema Tabel)

Skema Relasi database yang digunakan



## 4.3 Perancangan Algoritma

Bagian ini hanya diisi untuk kerangka algoritma untuk proses-proses yang dianggap cukup penting. Implementasi skeleton code juga sudah dapat dilakukan untuk kelas-kelas yang terdefinisi pada bahasa pemrograman tertentu. Boleh dibuat subbab per kelas.

### 4.3.1 Algoritma #1

Nama Kelas: Status Pembayaran

Nama Operasi: Melihat data Status Pembayaran

Algoritma: JK - AL001

```
id_Pembeli = string
Sql, sql2 : string
Sql ← "SELECT * FROM DataStatusPembayaran"
Sql2 ← "SELECT * FROM Pembeli where id_pembeli = + 'id_pembeli' "

Scan(id_pembeli)
If (id_pembeli <> NULL) then
    If (sql <> NULL) then
        Output (sql)
    Else
        Output ("Maaf, anda tidak mempunyai Status Pembayaran")
```

### 4.3.2 Algoritma #2

Nama Kelas: Booking

Nama Operasi: Melihat data Booking

Algoritma: JK - AL002

```
Id_Pembeli = string
Sql, sql2 : string
Sql ← "SELECT * FROM DataStatusPembayaran"
Sql2 ← "SELECT * FROM DataBooking"
Sql3 ← "SELECT * FROM Pembeli where id_pembeli = + 'id_pembeli' AND
DataStatusPembayaran.StatusPembayaran = 'Sukses'"

Scan(id_pembeli)
If (id_pembeli <> NULL) then
    If (sql2.DataStatusPembayaran == "Sukses")
        Output (sql2)
    Else
        Output ("Maaf, anda tidak mempunyai booking")
```

### 4.3.3 Algoritma #3

Nama Kelas: Jadwal Kereta

Nama Operasi: Melihat data Jadwal Kereta dari KAI

Algoritma: JK - AL003

```
Id_Pembeli = string
Sql, sql2 : string
Sql ← "SELECT * FROM DataJadwalKereta"

Scan(id_pembeli)
If (id_pembeli <> NULL) then
    Output(Sql)
Else
    Output("Belum ada Jadwal Kereta")
```

## 4.4 Perancangan Query

*S{Jika mengacu query tertentu, lengkapi tabel query di bawah} Query :*

No Query	Query	Keterangan
JK - Q001	SELECT * FROM DataStatusPembayaran	Untuk menampilkan semua data Status Pembayaran yang telah dipesan
JK - Q002	SELECT * FROM DataBooking	Untuk menampilkan semua data Booking yang telah dianggap sukses di Status Pembayaran
JK - Q003	SELECT * FROM Pembeli where id_pembeli = + 'id_pembeli' AND DataStatusPembayaran.StatusPembayaran = 'Sukses'	Untuk memastikan id_Pembeli terdapat didalam database dan sesuai dengan apa yang dipesan dan dibooking
JK - Q004	SELECT * FROM DataJadwalKereta	Untuk menampilkan semua data jadwal kereta dari KAI