**DPPL**-**SISKA**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SISTEM INFORMASI KERETA API

untuk:

PT. CAHKERETA

Dipersiapkan oleh:

Halim Wajdi (1301164207)

Indera Ihsan (1301160243)

Panji Bagaskara (1301160098)

Program Studi S1 Teknik Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

|  |  |  |  |
| --- | --- | --- | --- |
| Program Studi S1 Teknik Informatika  Fakultas Informatika | Nomor Dokumen | | Halaman |
| *DPPL-SISKA* | | 1 |
| Revisi |  |  |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | - | A | B | C | D | E | F | G |
| TGL |  |  |  |  |  |  |  |  |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

Daftar Halaman Perubahan

|  |  |  |  |
| --- | --- | --- | --- |
| Halaman | Revisi | Halaman | Revisi |
|  |  |  |  |

# 

# Daftar Isi

[Daftar Isi 5](#_Toc473623343)

[Daftar Gambar 6](#_Toc473623344)

[Daftar Tabel 6](#_Toc473623345)

[Daftar Lampiran 6](#_Toc473623346)

[1 Pendahuluan 7](#_Toc473623347)

[1.1 Tujuan Penulisan Dokumen 7](#_Toc473623348)

[1.2 Lingkup Masalah 7](#_Toc473623349)

[1.3 Definisi dan Istilah 7](#_Toc473623350)

[1.4 Referensi 10](#_Toc473623351)

[1.5 Deskripsi Umum Dokumen 10](#_Toc473623352)

[2 Deskripsi Perancangan 11](#_Toc473623353)

[2.1 Rancangan Lingkungan Implementasi 11](#_Toc473623354)

[2.2 Dekomposisi Fungsional Modul 12](#_Toc473623355)

[2.3 Perancangan Data 28](#_Toc473623356)

[2.3.1 Deskripsi Data 28](#_Toc473623357)

[2.3.2 Kamus Data 28](#_Toc473623358)

[2.4 Dekomposisi Fisik Modul 30](#_Toc473623359)

[2.5 Deskripsi Rinci Modul 31](#_Toc473623360)

[2.5.1 Deskripsi Layar 32](#_Toc473623361)

[2.5.2 Deskripsi Proses 45](#_Toc473623362)

[2.5.3 Deskripsi Laporan 45](#_Toc473623363)

[3 Perancangan Antaramuka Manusia 47](#_Toc473623364)

[3.1 Gambaran Umum Antarmuka Pengguna 47](#_Toc473623365)

[3.2 Tampilan layar 47](#_Toc473623366)

# Daftar Gambar

Hanya dicantumkan dan diisi jika ada gambar pada badan dokumen

# Daftar Tabel

Hanya dicantumkan dan diisi jika ada tabel pada badan dokumen

# Daftar Lampiran

Hanya dicantumkan dan diisi jika ada lampiran setelah badan dokumen

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) merupakan dokumen deskripsi dari perancangan perangkat lunak yang akan dikembangkan dan bertujuan untuk memberikan landasan yang diperlukan dalam proses pengkodean aplikasi SISKA. Dokumen ini digunakan oleh pengembang sebagai acuan teknis pengembagan perangkat lunak.

## Lingkup Masalah

SISKA (Sistem Informasi Kereta Api) adalah perangkat lunak untuk memudahkan pemesanan tiket kereta api, melihat jadwal keberangkatan kereta api serta memudahkan admin atau petugas dari perusahaan untuk membuat laporan harian, mengelola jadwal keberangkatan serta melakukan boarding pass bagi penumpang yang akan berangkat. Fungsional yang disediakan yaitu:

1. Untuk user customer, melakukan booking tiket.
2. Untuk user customer, melihat jadwal keberangkatan.
3. Untuk user customer, mengelola riwayat pemesanan.
4. Untuk user admin, mengelola jadwal keberangkatan.
5. Untuk user admin, melakukan boarding pass terhadap penumpang.
6. Untuk user admin, membuat laporan harian.

## Definisi dan Istilah

Definisi dan istilah dapat dilihat pada tabel 1.3.1

Table 1.3.1 Tabel Definisi dan Istilah

|  |  |  |
| --- | --- | --- |
| Istilah | Kepanjangan | Deskripsi Singkat |
| DPPL | Deskripsi Perancangan Perangkat Lunak | Acuan pengguna bagi user dan acuan pengembangan bagi pengembang |
| SISKA | Sistem Informasi Kereta Api | Sistem informasi yang sedang dikembangkan oleh pengembang |
| DPPL-xxxx | - | Merupakan kode yang merepresentasikan nama dari DPPL yang sedang dikembangkan |
| DFD | Data Flow Diagram | DFD adalah data flow program, diagram dan notasi yang digunakan untuk menunjukan aliran data pada perangkat lunak |
| ERD | Entity Relationship Diagram | ERD adalah entity relationship diagram, diagram dan notasi yang digunakan untuk mempresentasikan struktur data statis pada perangkat lunak. |
| UML | Unified Modelling Language | UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. |
| HTML | Hyper Text Markup Language | HTML adalah Hyper Text Markup Language, sintaks bahas yang digunakan dalam world wide web. |
| JavaScript | - | JavaScript adalah bahasa skrip yang populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Internet |
| IE | Internet Explorer | Explorer(IE), Mozilla Firefox, Netscape dan Opera. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT. |
| CSS | Cascading Style Sheet | CSS(Cascading Style Sheet) Merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. |
| PHP | PHP Hypertext Preprocessor | PHP (Hypertext Preprocessor) Adalah bahasa skrip yang dapat ditanamkan atau disisipkan kedalam HTML. PHP banyak dipakai untuk memprogram situs web dinamis. |
| MySQL | - | MySQL adalah perangkat lunak yang memiliki fasilitas sebagai basis data dari sebuah program. MySQL merupakan DBMS yang cocok untuk program-program yang tidak memiliki lingkup yang luas. |

## Referensi

* Kelompok 5. *Spesifikasi Kebutuhan Perangkat Lunak.* Jurusan Teknik Informatika Telkom University.
* Gumelar, Ashary. *dkk.GL02 Dokumen Perancangan Perangkat Lunak.* Jurusan Program Keahlian Teknik Komputer IPB.

## Deskripsi Umum Dokumen

Dokumen DPPL menjelaskan dan memberi informasi tentang software atau aplikasi akademik yang berfungsi untuk mempermudah pengguna kereta api memesan tiket serta mempermudah admin atau petugas dari perusahaan untuk mengelola jadwal keberangkatan, membuat laporan serta melakukan boarding pass bagi penumpang yang akan naik kereta api. Pada dokumen ini terdapat 3 bagian utama yaitu pertama berisi penjelasan tentang dokumen DPPL yang mencakup tujuan penulisan dokumen, lingkup masalah, definisi dan istilah, referensi dan deskripsi umum dokumen. Kedua berisi penjelasan tentang perangkat lunak yang akan dikembangkan yang meliputi rancangan lingkungan implementasi, dekomposisi fungsional modul, perancangan data, dekomposisi fisik modul, dan deskripsi rinci modul. Ketiga berisi uraian kebutuhan perangkat lunak secara lebih rinci yang meliputi gambaran umum antarmuka pengguna dan tampilan layar.

# Deskripsi Perancangan

## Rancangan Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk pengembangan Sistem Informasi Kereta Api adalah sebagai berikut:

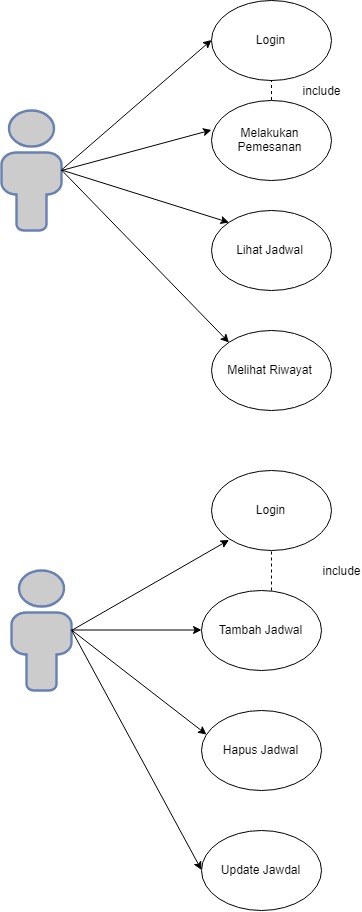
Table 2.1.1 Rancangan Lingkungan Implementasi

|  |  |
| --- | --- |
| **Sistem Peminjaman** | **Spesifikasi** |
| Sistem Operasi | Windows 10 |
| DBMS | MySql |
| Development Tools | Xampp, Sublime Text 3, Google Chrome |
| Bahasa Pemrograman | HTML, PHP, CSS, Javascript |

## Dekomposisi Fungsional Modul

### Use Case Diagram

Use Case Diagrambisa dilihat pada gambar 2.2.1



Gambar 2.2.1 Use Case Diagram

Diagram diatas merupakan use case diagram pada system informasi cahkereta. Use case diagram digunakan oleh dua orang user yang dapat di segmentasi menjadi dua bagian, yaitu user customer dan user admin. Pada use case diagram ini, Customer memiliki 4 buah fungsionalitas sedangkan admin memiliki 3 buah fungsionalitas. Fungsionalitas Customer antara lain : Login, Melakukan Pemesanan, Melihat Jadwal, Mengelola Riwayat Pemesanan. Customer wajib melakukan proses login untuk dapat menjalani mekanisme fungsionalitas.

Sedangkan admin memiliki fungsionalitas antara lain : Membuat laporan, mengelola jadwal, mengelola boarding pass. Sama halmya dengan customer, admin berhak melakukan login untuk dapat menjalani mekanisme fungsionalitas yang ada. Secara garis besar, use case diagram yang digambarkan memaparkan proses transaksi antara customer dan admin

### Definisi Aktor

Definisi actor bisa dilihat pada tabel 2.2.1

Table 2.2.1 Tabel Definisi Aktor

|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder** | **Tanggung Jawab** | **Kemampuan yang dimiliki** | **Pengalaman menggunakan komputer** |
| **Admin** | Memperbaharui data penjadwalan kereta, Menambah Jadwal, Menghapus Jadwal | Dapat menggunakan web browser | Mengenal operasi dasar pada computer dan dapat megaplikasikanya sesuai keadaan |
| **Customer** | Pemesanan tiket, melihat jadwal keberangkatan dan riwayat | Dapat menggunakan web browser | Mengenal operasi dasar pada computer dan web browser |

### Definisi Use Case

Definisi use case dapat dilihat pada tabel 2.2.2

Table 2.2.2 Tabel Definisi Use Case

|  |  |
| --- | --- |
| ID | Use Case |
| UC-001 | Melakukan Pemesanan |
| UC-002 | Melihat Jadwal |
| UC-003 | Melihat Riwayat |
| UC-004 | Mengupdate Jadwal |
| UC-005 | Tambah Jadwal |

### Definisi Use Case Realization

Definisi Use Case Realization dapat dilihat pada tabel 2.2.3

Table 2.2.3 Definisi Use Case Realization

|  |  |  |
| --- | --- | --- |
| ID | Use Case Realization | Deskripsi |
| UCR-001-01 | Melakukan Pemesanan | Use Case Realization ini diakses oleh user (customer) untuk pemesanan tiket kereta api. |
| UCR-002-01 | Melihat Jadwal | Use Case Realization ini diakses oleh user (customer) untuk melihat jadwal keberangkatan kereta api. |
| UCR-003-01 | Melihat Riwayat | Use Case Realization ini diakses oleh user (customer) untuk melihat riwayat pemesanan |
| UCR-005-01 | Menambahkan Jadwal | Use Case Realization ini diakses oleh user (admin) untuk menambahkan jadwal keberangkatan kereta. |
| UCR-005-03 | Mengupdate Jadwal | Use Case Realization ini diakses oleh user (admin) untuk mengubah jadwal keberangkatan kereta. |
| UCR-006-01 | Menambah Jadwal | Use Case Realization ini diakses oleh user (admin) untuk melakukan penambahan jadwal oleh admin |

### Spesifikasi Use Case

Berikut penjelasan spesifikasi dari setiap use case yang ada. Spesifikasi tersebut akan meliputi :

* Use Case Realization ID : merupakan id use case realization
* Use Case Realization Name : merupakan nama dari use case realization.
* Deskripsi Singkat : menjelaskan fungsi dan esensi dari use case.
* Aktor : pemeran yang terlibat pada use case
* Pre Condition : merupakan kondisi awal yang harus terpenuhi sebelum use case berjalan
* Post Condition : merupakan kondisi akhir yang terjadi setelah use case berjalan.
* Normal Flow : merupakan langkah-langkah yang secara normal dijalankan dalam use case
* Alternate Flow : merupakan langkah-langkah alternative dari normal flow

Berikut scenario semua use case spesifikasi.

#### Use Case Melakukan Pemesanan (UC-001)

Use case scenario melakukan pemesanan bisa dilihat pada tabel 2.2.4

Table 2.2.4 Use Case Melakukan Pemesanan

|  |  |
| --- | --- |
| Use Case Realization ID | UCR-001-01 |
| Use Case Realization Name | Melakukan Pemesanan |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (customer) untuk pemesanan tiket kereta api. |
| Aktor | Customer |
| Pre Condition | Customer ingin memesan tiket |
| Post Condition | Customer sudah mendapatkan tiket sesuai jadwal yang dipesan |
| Normal Flow | |
| Aksi Aktor | Reaksi Sistem |
| 1. Customer membuka laman pemesanan |  |
|  | 2. Sistem menampilkan form pemesanan |
| 3. Customer mengisi form pemesanan |  |
| 4. Customer menekan tombol next |  |
|  | 5. Sistem menampilkan jadwal yang tersedia |
| 6. Customer memilih jadwal |  |
| 7. Customer menekan tombol submit |  |
|  | 8. Data yang diinputkan customer direkam di dalam database dan memproduksi kode pembayaran |
| 9. Customer membayar melalui metode pembayaran yang tertera |  |
|  | 10. Sistem menunggu status pembayaran dari unpaid menjadi paid |
|  | 11. Sistem memproduksi kode booking agar customer dapat melakukan boarding pass |
|  | 12. Tiket yang sudah dipesan akan masuk kedalam riwayat transaksi. |
| 13. Customer bisa melihat tiketnya dihalaman riwayat transaksi. |  |
| Alternate Flow | |
| 3.1 Customer melakukan input form yang salah atau masih ada field yang kosong |  |
|  | 3.1.1 Sistem akan menampilkan notifikasi untuk mengisi setiap field |
| 9.1 Customer tidak melakukan pembayaran dalam jangka waktu 30 menit |  |
|  | 9.1.1 Sistem akan menghapus record yang telah direkam dengan status unpaid dan id customer tertentu. |

#### Use Case Melihat Jadwal (UC-002)

Use case melihat jadwal dapat dilihat pada tabel 2.2.5

Table 2.2.5 Use Case Melihat Jadwal

|  |  |
| --- | --- |
| Use Case Realization ID | UCR-002-01 |
| Use Case Realization Name | Melihat Jadwal |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (customer) untuk melihat jadwal kereta api. |
| Aktor | Customer |
| Pre Condition | Customer ingin mengetahui jadwal kereta pada hari yang ditentukan |
| Post Condition | Customer mengetahui jadwal keberangkatan kereta api. |
| Normal Flow | |
| Aksi Aktor | Reaksi Sistem |
| 1. Customer membuka laman pemesanan |  |
|  | 2. Sistem menampilkan form pemesanan |
| 3. Customer mengisi tanggal beserta asal dan tujuan kereta yang diinginkan |  |
| 4. Customer menekan tombol next |  |
|  | 5. Sistem menampilkan jadwal yang tersedia |
| Alternate Flow | |
| 3.1 Customer melakukan input form yang salah atau masih ada field yang kosong |  |
|  | 3.1.1 Sistem akan menampilkan notifikasi untuk mengisi setiap field |
| 3.2 Customer mengisi input dengan benar |  |
|  | 3.2.1 Sistem tidak menemukan jadwal yang di inputkan |
|  | 3.2.2 Sistem memberikan notifikasi |
| 3.3 Customer menekan tombol back |  |
|  | 3.3.1 Sistem akan membuka halaman awal |

#### Use Case Mengelola Jadwal (UC-005)

Use case mengelola jadwal dapat dilihat pada tabel 2.2.8

Table 2.2.8 Use Case Mengelola Jadwal

|  |  |
| --- | --- |
| Use Case Realization ID | UCR-005-01 |
| Use Case Realization Name | Menambahkan Jadwal |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (admin) untuk menambahkan jadwal keberangkatan kereta api. |
| Aktor | Admin |
| Pre Condition | Admin ingin menambahkan jadwal keberangkatan kereta api. |
| Post Condition | Admin telah menambahkan jadwal keberangkatan kereta api. |
| Normal Flow | |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin menyiapkan data yang akan ditambahkan kedalam database dan menekan tombol insert jadwal |  |
|  | 2. Sistem membuka laman insert jadwal |
| 3. Admin mengisi form untuk mengisi jadwal |  |
| 4. Admin menekan tombol submit |  |
|  | 5. Sistem menyimpan jadwal yang baru ke dalam database jadwal |
|  | 6. Sistem mengeluarkan notifikasi sukses |
| Alternate Flow | |
| 3.1 Admin belum mengisi setiap field yang ada di form ataupun salah input |  |
|  | 3.1.1 Sistem mengirimkan notifikasi ataupun alert |
| 3.2 Admin mengisi setiap field dan tidak salah menginputkan field |  |
|  | 3.2.1 Sistem cek apakah akan terjadi duplikasi atau tidak |
|  | 3.2.2 Sistem akan mengeluarkan notifikasi jika akan terjadi notifikasi |
| Use Case Realization ID | UCR-005-02 |
| Use Case Realization Name | Menghapus Jadwal |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (admin) untuk menghapus jadwal keberangkatan kereta api. |
| Aktor | Admin |
| Pre Condition | Terdapat data yang akan dihapus pada daftar jadwal |
| Post Condition | Terhapus nya data yang dipilih pada daftar jadwal. |
| Normal Flow | |
| 1. Admin menekan tombol hapus jadwal |  |
|  | 2. Sistem membuka laman untuk menghapus jadwal |
| 3. Admin mengisi form untuk menghapus data jadwal |  |
|  | 4. Sistem akan melakukan cek pada table jadwal. |
|  | 5. Sistem akan menghapus jadwal sesuai field yang diinputkan oleh Admin |
|  | 6. Sistem akan mengeluarkan notifikasi sukses |
| Use Case Realization ID | UCR-005-03 |
| Use Case Realization Name | Menyunting Jadwal |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (admin) untuk menyunting jadwal keberangkatan kereta api. |
| Aktor | Admin |
| Pre Condition | Terdapat data yang akan disunting pada daftar jadwal |
| Post Condition | Tersunting nya data yang dipilih pada daftar jadwal. |
| Normal Flow |  |
| 1. Admin menekan tombol update jadwal |  |
|  | 2. Sistem membuka laman untuk menyunting jadwal |
| 3. Admin mengisi field sesuai dengan jadwal yang ingin dicari |  |
| 4. Admin menekan tombol cari |  |
|  | 5. Sistem akan menampilkan informasi tentang jadwal yang dicari |
| 6. Admin mengisi field untuk menyunting jadwal |  |
| 7. Admin menekan tombol update |  |
|  | 8. Sistem merubah data yang ada di database |
| Alternate Flow | |
| 4.1 Admin menekan tombol cari |  |
|  | 4.1.1 Sistem tidak menemukan data |
|  | 4.1.2 Sistem mengeluarkan notifikasi bahwa data tidak ditemukan |

#### Use Case Melihat Riwayat (UC-003)

Use case melihat jadwal dapat dilihat pada tabel 2.2.5

Table 2.2.5 Use Case Melihat Jadwal

|  |  |
| --- | --- |
| Use Case Realization ID | UCR-002-01 |
| Use Case Realization Name | Melihat Jadwal |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (customer) untuk melihat riwayat pemesanan |
| Aktor | Customer |
| Pre Condition | Customer ingin mengetahui riwayat pemesanan kereta kereta pada hari yang ditentukan |
| Post Condition | Customer mengetahui riwayat pemesanan tiket kereta api. |
| Normal Flow | |
| Aksi Aktor | Reaksi Sistem |
| 1. Customer membuka laman riwayat |  |
|  | 2. Sistem menampilkan form riwayat |
| Alternate Flow | |
| 3.1 Customer masuk ke laman riwayat |  |
|  | 3.1.1 Sistem akan menampilkan notifikasi bahwa belum ada riwayat yang tertera |

#### Use Case Menambah Jadwal (UC-005)

Use case menambah jadwal dapat dilihat pada tabel 2.2.5

Table 2.2.5 Use Case Melihat Jadwal

|  |  |
| --- | --- |
| Use Case Realization ID | UCR-004-01 |
| Use Case Realization Name | Menambah Jadwal |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (Admin) untuk menambah jadwal |
| Aktor | Admin |
| Pre Condition | Customer ingin menambah jadwal kereta |
| Post Condition | Terdapat data kereta terbaru |
| Normal Flow | |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin membuka laman tambah jadwal kereta |  |
|  | 2. Sistem menampilkan form input jadwal |
| 1. Admin melakukan penginputan jadwal |  |
|  | 1. Record tersimpan dan ditampung oleh database |
| Alternate Flow | |
| 3.1 Admin masuk ke laman tambah jadwal |  |
|  | 3.2. Sistem menampilkan form input jadwal |
| 3.3 Admin melakukan penginputan jadwal dengan salah |  |
|  | 3.4 Melakukan pengiriman noifikasi bahwa terdapat kesalahan penginputan data |

## Perancangan Data

### Deskripsi Data

Deskripsi data dapat dilihat pada tabel 2.3.1

Table 2.3.1 Deskripsi Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama Tabel | Jenis | Volume | Laju | Primary Key | Contstraint Integrity |
| Customer | Master | - | - | username |  |
| Stasiun | Master | - | - | idstasiun |  |
| Kereta | Master | - | - | idkereta |  |
| Jadwal | Relasi | - | - | idjadwal | Stasiun(idstasiun), Kereta(idkereta) |
| Keberangkatan | Relasi | - | - | idboarding | Pemesanan(noktp) |
| Gerbong | Master | - | - | idgerbong |  |
| Riwayat | Relasi | - | - | idriwayat | Pemesanan(noktp) |

### Kamus Data

Kamus data bisa dilihat pada tabel 2.3.2 – tabel 2.3.9

Table 2.3.1 User

|  |  |  |
| --- | --- | --- |
| Atribut | Tipe Data | PK/FK |
| username | VARCHAR(10) | PK |
| Password | VARCHAR(20) |  |
| Email | VARCHAR(50) |  |

Table 2.3.2 Gerbong

|  |  |  |
| --- | --- | --- |
| Atribut | Tipe Data | PK/FK |
| Idgerbong | VARCHAR(10) | PK |
| Idkereta | VARCHAR(10) | FK |
| Kelas | VARCHAR(15) |  |
| jumkursi | Int(11) |  |

Table 2.3.3 Stasiun

|  |  |  |
| --- | --- | --- |
| Atribut | Tipe Data | PK/FK |
| Idstasiun | VARCHAR(10) | PK |
| sta\_awal | VARCHAR(20) |  |
| sta\_akhir | VARCHAR(20) |  |

Table 2.3.4 Kereta

|  |  |  |
| --- | --- | --- |
| Atribut | Tipe Data | PK/FK |
| nama\_kereta | VARCHAR(20) |  |
| Idkereta | VARCHAR(10) | PK |

Table 2.3.5 Jadwal

|  |  |  |
| --- | --- | --- |
| Atribut | Tipe Data | PK/FK |
| jam\_berangkat | Date |  |
| Jamtiba | Date |  |
| tanggalberangkat | Date |  |
| Idkereta | VARCHAR(10) | FK |
| Idstasiun | VARCHAR(10) | FK |
| Idjadwal | Int(10) | PK |

Table 2.3.6 tiket

|  |  |  |
| --- | --- | --- |
| Atribut | Tipe Data | PK/FK |
| Idtiket | VARCHAR(10) | PK |
| Username | VARCHAR(10) | FK |
| Idjadwal | Int(11) | FK |
| Idgerbong | VARCHAR(10) | FK |
| Noktp | VARCHAR(20) |  |
| Kursi | VARCHAR(3) |  |
| statusbayar | Int(11) |  |
| statuscheckin | Int(11) |  |

Table 2.3.7 Riwayat

|  |  |  |
| --- | --- | --- |
| Atribut | Tipe Data | PK/FK |
| Idtiket | VARCHAR(10) | FK |
| Idriwayat | VARCHAR(10) | PK |

Table 2.3.8 Admin

|  |  |  |
| --- | --- | --- |
| Atribut | Tipe Data | PK/FK |
| Username | VARCHAR(10) | FK |
| Password | VARCHAR(20) |  |
| Divisi | VARCHAR(20) |  |

## Dekomposisi Fisik Modul

Dekomposisi fisik modul bisa dilihat pada tabel 2.4.1

Table 2.4.1 Dekomposisi Fisik Modul

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sub Aplikasi | Modul | Nama File | Input | Output |
| Pemesanan | Lihat Jadwal | penjadwalan.php | Field tanggal dan table jadwal | Tabel Jadwal |
| Booking tiket | bookingtiket.php | Input form dan table jadwal | Tabel Pemesanan |
|
| Kelola Jadwal | Tambah Jadwal | tambahjadwal.php | Input form | Tabel jadwal |
| Update Jadwal | updatejadwal.php | Field pencarian | Tabel jadwal |
|
| Riwayat Pemesanan | Lihat Riwayat | riwayat.php | Tabel Keberangkatan | Tabel Keberangkatan |
| Hapus Riwayat | riwayat.php | Field pencarian | Tabel Keberangkatan |
| Laporan | Laporan Harian | report.php | Tabel keberangkatan | Tabel keberangkatan |

## Deskripsi Rinci Modul

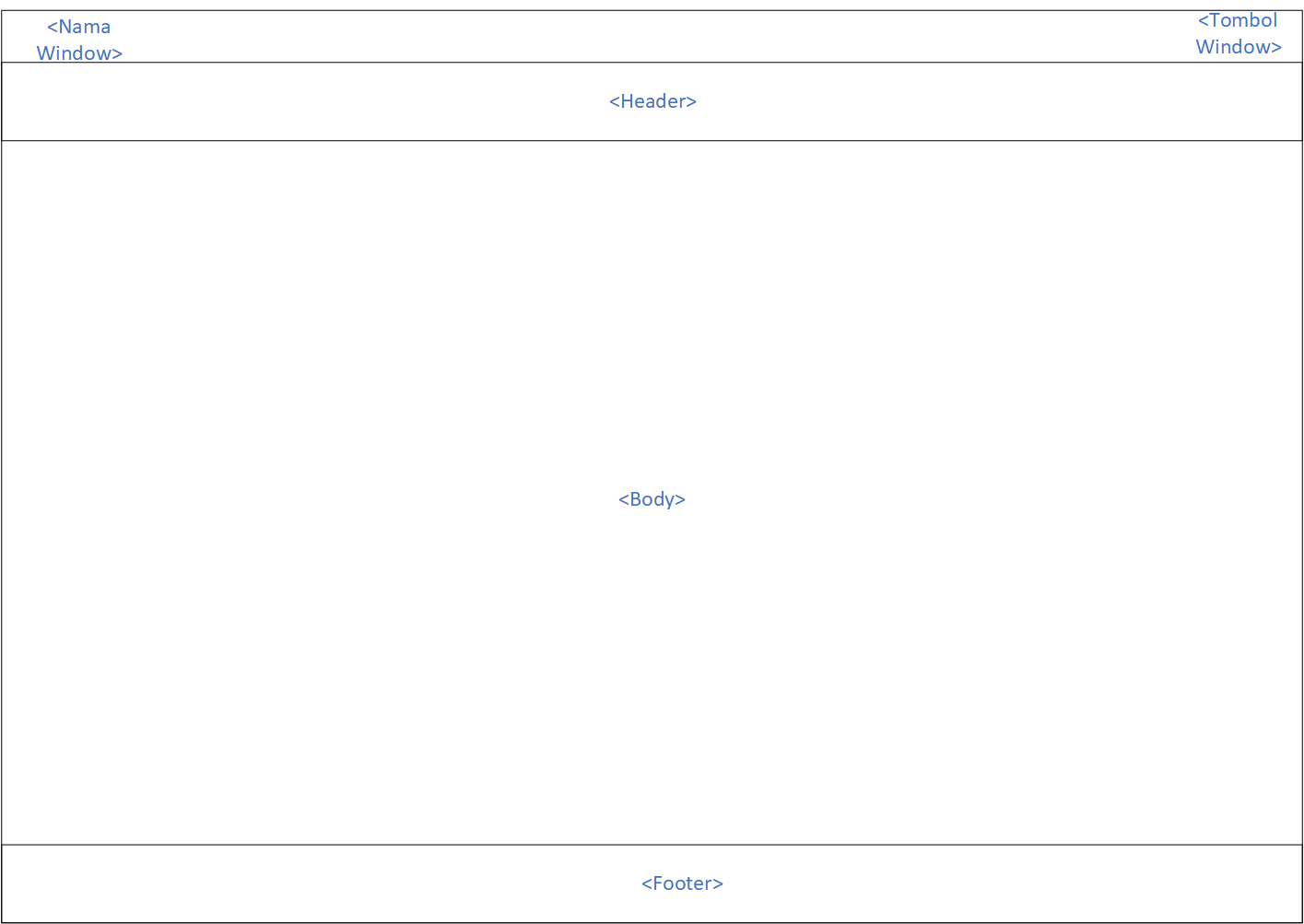
Deskripsi rinci modul bisa dilihat pada tabel 2.5.1

Table 2.5.1 Deskripsi Rinci Modul

|  |  |
| --- | --- |
| Nama Layar | Keterangan |
| Tampil\_list\_jadwal | Tampilan jadwal keberangkatan kereta berdasarkan tanggal |
| Booking\_tiket | Menginputkan data keberangkatan, memilih jadwal serta memasukkan data diri sesuai dengan kartu identitas yang dimiliki |
| Tambah\_jadwal | Menginputkan data jadwal keberangkatan |
| Cari\_jadwal | Mencari jadwal sesuai dengan inputan yang dimasukkan |
| Update\_jadwal | Mengupdate jadwal keberangkatan yang diinginkan |
| Lihat\_riwayat | Tampilan riwayat keberangkatan yang dilakukan oleh penumpang |

### Deskripsi Layar

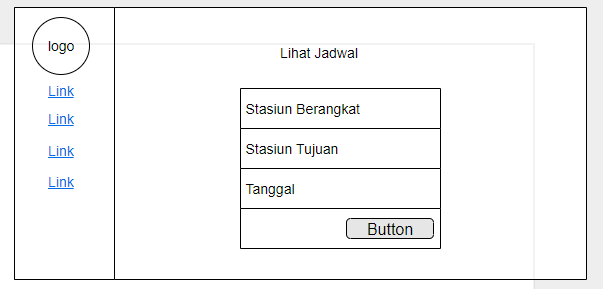
Layout dari setiap halaman bisa dilihat pada gambar 2.5.1



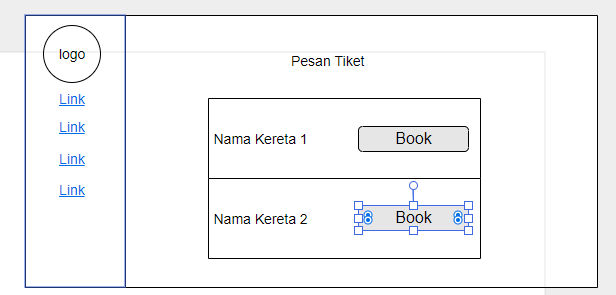
Gambar 2.5.1 Layout Layar

#### Lihat List Jadwal

Lihat list jadwal bisa dilihat pada gambar 2.5.2 dan 2.5.3



Gambar 2.5.2 Form Lihat Jadwal



Gambar 2.5.3 Tampilan setelah Form

##### Deskripsi Objek

Deskripsi Objek bisa dilihat pada tabel 2.5.2 dan 2.5.3

Table 2.5.2 Deskripsi objek gambar 2.5.2

|  |  |  |
| --- | --- | --- |
| Nama Objek | Jenis Objek | Keterangan |
| inputstawal | dropdown | Untuk menampilkan jadwal dengan stasiun awal sesuai input |
| inputstakhir | dropdown | Untuk menampilkan jadwal dengan stasiun akhir sesuai input |
| inputtanggal | Date | Untuk menampilkan jadwal dengan tanggal sesuai input |
| inputkelas | dropdown | Untuk menampilkan jadwal dengan kelas sesuai dengan input |
| Btnsubmit | button | Untuk mengarahkan kehalaman selanjutnya |

Table 2.5.3 Deskripsi Objek gambar 2.5.3

|  |  |  |
| --- | --- | --- |
| Nama Objek | Jenis Objek | Keterangan |
| tablejadwal | table | Untuk menampilkan jadwal sesuai dengan input yang dimasukkan dihalaman sebelumnya dengan format table namakereta, tanggal serta jam keberangkatan, stasiun awal dan tujuan, kelas, serta harga. |
| btnbook | Button | Untuk melakukan booking pada button jadwal yang klik. |

##### Algoritma

Secara singkat, algoritma yang diperlukan untuk mencari sebuah jadwal kereta berdasarkan input yang dilakukan oleh user dalam bahasa pemrograman sql adalah :

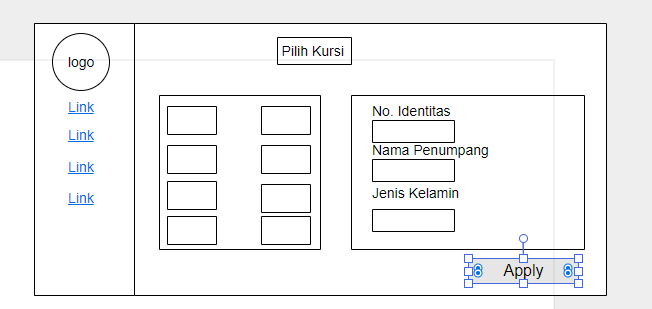
|  |
| --- |
| SELECT idjadwal, namakereta, awal, akhir, waktu\_berangkat, waktu\_sampai, kelas, harga FROM jadwal JOIN stasiun USING (idstasiun) JOIN kereta USING (idkereta) JOIN gerbong USING (idgerbong) WHERE (awal=’$awal’ and akhir=’$akhir’ and waktu\_berangkat=’$date’ and kelas=’$kelas’); |

Perlu diketahui bahwa algoritma hanya akan dieksekusi atau berjalan ketika user menekan button submit, input yang dilakukan oleh user akan disimpan dalam sebuah variabel string yang kemudian akan dicocokan pada record yang sebelumnya telah tertera di database. Ketika terdapat sebuah record yang memiliki valu atau nilai yang identik dengan hasil input dari user, maka record tersebut akan ditampilkan pada laman website.

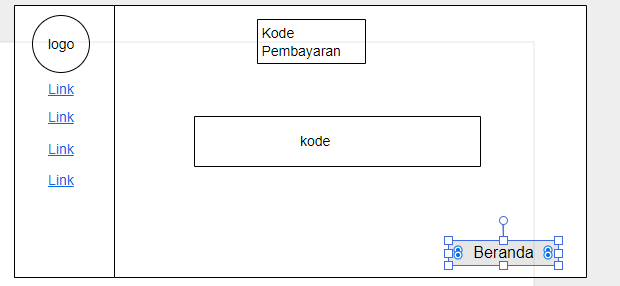
Dan dengan begitu user mendapatkan mekanisme fungsionalitas melihat jadwal yang tertera di use case diagram.

#### Booking Tiket

Berikut layar yang digunakan untuk booking tiket ,pilih kursi dan input data diri penumpang



Gambar 2.5.4 Form input Data Diri



Gambar 2.5.5 Launch Kode Pembayaran setelah Memasukkan Data Diri

##### Deskripsi Objek

Berikut merupakan deskripsi objek dari layar 2.5.4 dan 2.5.5

Table 2.5.4 Deskripsi Objek gambar 2.5.4

|  |  |  |
| --- | --- | --- |
| Nama Objek | Jenis Objek | Keterangan |
| inputnoidentitas | text | Untuk memasukkan no identitas penumpang ke database |
| inputnama | text | Untuk memasukkan nama penumpang ke database |
| inputtempatlahir | text | Untuk memasukkan tempat lahir ke database |
| inputtanggallahir | date | Untuk memasukkan tanggal lahir ke database |
| Btnnext | button | Untuk mengarahkan kehalaman selanjutnya untuk diproses permintaan bookingnya. |

Table 2.5.5 Deskripsi Objek untuk gambar2.5.5

|  |  |  |
| --- | --- | --- |
| Nama Objek | Jenis Objek | Keterangan |
| btnok | button | Untuk mengarahkan ke home awal setelah login |

##### Algoritma

Secara singkat, algoritma yang diperlukan untuk membuat sebuah rekaman pemesanan berdasarkan input yang dilakukan oleh user dalam bahasa pemrograman sql adalah :

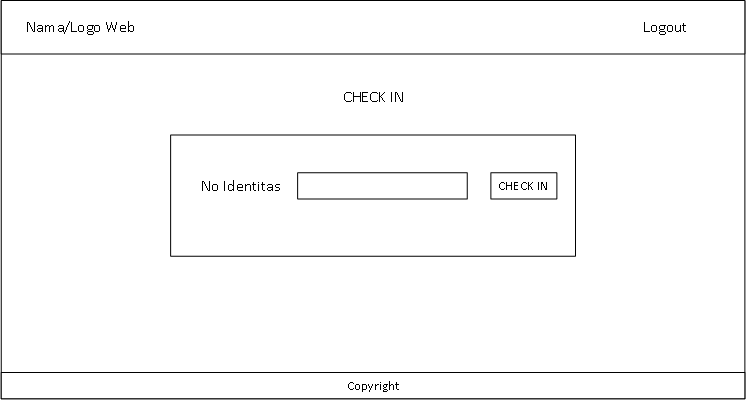
|  |
| --- |
| INSERT INTO pemesanan (noktp,namapemesan, tempatlahirpesan, tanggallahirpesan,idjadwal,username,status,tanggal\_hariini) VALUES (’$noktp’,’$nama’,’$tl’,’$tgllhr’,’$idjadwal’,’$uname’,’unpaid’,’$tglhrini’);  INSERT INTO riwayat (noktp, idriwayat) VALUES (’$noktp’,’$idriwayat’); |

Perlu diketahui bahwa algoritma hanya akan dieksekusi atau berjalan ketika user menekan button submit. input yang dilakukan oleh user akan disimpan dalam sebuah variabel string yang kemudian akan dijadikan sebagai record baru pada tabel pemesanan. Hasil Input user berupa nomor ktp, nama, tempat lahir, tanggal lahir, id jadwal, username, status pembayaran, dan tanggal pemesanan yang kemudian akan menambahkan record baru pada tabel pemesanan. Data yang tersimpan pada database merupakan sebuah informasi mengenai tiket yang dimiliki oleh user yang melakukan pemesanan.

Dengan berjalanya algoritma tersebut, maka user dapat menjalani mekanisme fungsionalitas Booking ticket yang tertera pada use case diagram.

#### Boarding Pass

Berikut merupakan layout layout untuk kegiatan boarding pass



Gambar 2.5.6 Form Check In

##### Deskripsi Objek

Deskripsi objek dari gambar 2.5.6 dapat dilihat pada tabel berikut

Table 2.5.6 Deskripsi Objek untuk gambar 2.5.6

|  |  |  |
| --- | --- | --- |
| Nama Objek | Jenis Objek | Keterangan |
| inputnoidentitas | text | Sebagai parameter pencarian di dalam database |
| btncheckin | button | Untuk memulai pencarian dimana noidentitas akan check in |

##### Algoritma

Secara singkat, algoritma yang diperlukan untuk memeriksa record pada database berdasarkan input yang dilakukan oleh user dalam bahasa pemrograman sql adalah :

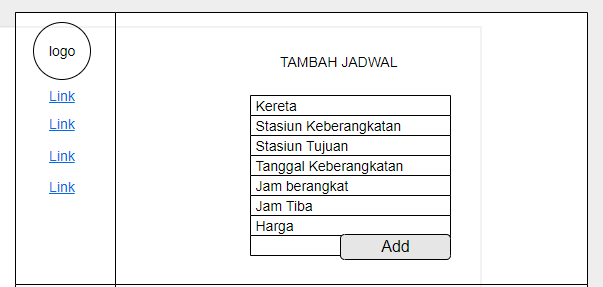
|  |
| --- |
| SELECT noktp, status FROM jadwal WHERE (noktp=’$noktp’ and status=’paid’);  INSERT INTO keberangkatan (idboarding, noktp) VALUES (’$idboard’,’$noktp’); |

Perlu diketahui bahwa algoritma hanya akan dieksekusi atau berjalan ketika user menekan button submit. input yang dilakukan oleh user akan disimpan dalam sebuah variabel string yang kemudian akan digunakan sebagai kata kunci pencarian pada tabel jadwal. Algoritma melakukan proses pencocokan no. Ktp user yang diinputkan oleh admin pada tabel jadwal. Ketika algoritma mendapatkan data yang dicari maka mekanisme selanjutnya adalah melakukan pembuatan track record baru pada tabel keberangkatan dengan nilai idboarding yang baru dan no ktp yang merupakan hasil input dari user pada form yang tertera pada laman website.

Dengan berhasilnya mekanisme tersebut, maka user admin berhasil menjalani mekanisme fungsionalitas Mengelola Boarding pass yang tertera pada use case diagram

#### Tambah Jadwal

Berikut merupakan layout untuk tambah jadwal



Gambar 2.5.7 Form Tambah Jadwal

##### Deskripsi Objek

Deskripsi Objek dapat dilihat pada tabel dibawah ini

Table 2.5.7 Deskripsi Objek untuk gambar 2.5.7

|  |  |  |
| --- | --- | --- |
| Nama Objek | Jenis Objek | Keterangan |
| inputberangkat | Date | Input jadwal berangkat untuk dimasukkan kedalam database jadwal |
| Inputsampai | Date | Input jadwal sampai untuk dimasukkan kedalam database |
| inputidkereta | dropdown | Input idkereta untuk dimasukkan kedalam database jadwal |
| inpudidgerbong | dropdown | Input idkereta untuk dimasukkan kedalam database jadwal |
| Inputstasiunawal | Text | Input stasiunawal untuk dimasukkan ke dalam database jadwal |
| inputstasiunakhir | Text | Input stasiun akhir untuk dimasukkan ke dalam database jadwal |
| btnadd | Button | Memulai proses penambahan jadwal kedalam database jadwal |

##### Algoritma

|  |
| --- |
| INSERT INTO stasiun (idstasiun,awal,akhir) VALUES (’$idstasiun’,’$awal’,’$akhir’);  INSERT INTO jadwal (waktu\_berangkat,waktu\_sampai,idkereta,idgerbong,idstasiun,idjadwal) VALUES (’$go’,’$arrive’,’$idkereta’,’$idgerbong’,’idstasiun’,’$idjadwal’); |

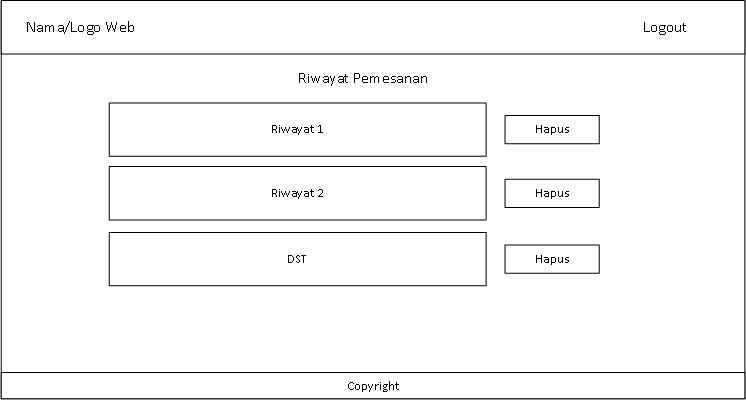
Perlu diketahui bahwa algoritma hanya akan dieksekusi atau berjalan ketika user menekan button submit. input yang dilakukan oleh user akan disimpan dalam sebuah variabel string yang kemudian akan digunakan sebagai record baru yang kemudian akan dijadikan sebagai nilai atau value yang akan ditambahkan. Algoritma melakukan penambahan record pada tabel stasiun dengan nilai hasil dari input id stasiun, stasiun awal, stasiun akhir.

Algoritma juga melakukan penambahan record pada tabel jadwal dengan nilai hasil dari input waktu berangkat, waktu sampai, id kereta, id stasiun dan id jadwal.

Dengan berhasilnya mekanisme tersebut, maka user admin berhasil menjalani mekanisme fungsionalitas Menambah jadwal yang tertera pada use case diagram

#### Lihat Riwayat

Berikut merupakan layout lihat riwayat.



Gambar 2.5.11 Tampilan lihat riwayat

##### Deskripsi Objek

Deskripsi objek gambar 2.5.11 dapat dilihat pada tabel 2.5.11

Table 2.5.11 Deskripsi Objek Lihat Riwayat

|  |  |  |
| --- | --- | --- |
| Nama Objek | Jenis Objek | Keterangan |
| tableriwayat | table | Tampilan list riwayat pemesanan yang telah dilakukan oleh penumpang |
| btndelete | button | Memulai menghapus riwayat pemesanan yang dipilih |

##### Algoritma

|  |
| --- |
| SELECT \* FROM riwayat; |

# Perancangan Antaramuka Manusia

## Gambaran Umum Antarmuka Pengguna

Gambarkan fungsionalitas sistem dari sudut pandang pengguna. Jelaskan bagaimana pengguna akan menggunakan sistem guna menyelesaikan semua fitur sesuai dengan yang diharapkan dan juga informasi umpan balik (notifikasi, pesan error, dll.) yang akan ditampilkan kepada pengguna.

## Tampilan layar

Tampilkan screenshot, mockup atau wireframe yang menggambarkan antarmuka sistem dari sudut pandang pengguna. Gambar yang dimaksud dapat berupa gambar yang digambar tangan (hand-drawn) atau hasil dari alat bantu menggambar terotomasi (wireframe, dsj.).