

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

| | | | | |
|---|---|--------------------------|--|----------------|
|  UNIVERSITAS Telkom | Program Studi S1 Teknik Informatika Fakultas Informatika | Nomor Dokumen | | Halaman |
| | | <i>DPPL-SISKA</i> | | 89 |
| | | Revisi | | |

DAFTAR PERUBAHAN

| Revisi | Deskripsi |
|----------|-----------------------------------|
| A | Lingkup Masalah BAB 2 BAB 3 |
| B | |
| C | |
| D | |
| E | |
| F | |
| G | |

| INDEX | - | A | B | C | D | E | F | G |
|----------------|---|------------|---|---|---|---|---|---|
| TGL | | | | | | | | |
| Ditulis oleh | | Kelompok 3 | | | | | | |
| Diperiksa oleh | | | | | | | | |
| Disetujui oleh | | | | | | | | |

Daftar Halaman Perubahan

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

Daftar Isi

| | |
|---|-------------------------------------|
| Daftar Isi | 5 |
| Daftar Gambar | 7 |
| Daftar Tabel | 8 |
| Daftar Lampiran | Error! Bookmark not defined. |
| 1 Pendahuluan..... | 10 |
| 1.1 Tujuan Penulisan Dokumen | 10 |
| 1.2 Lingkup Masalah | 10 |
| 1.3 Definisi dan Istilah | 11 |
| 1.4 Referensi..... | 14 |
| 1.5 Deskripsi Umum Dokumen | 14 |
| 2 Deskripsi Perancangan | 15 |
| 2.1 Rancangan Lingkungan Implementasi | 15 |
| 2.2 Dekomposisi Fungsional Modul | 16 |
| 2.2.1 Use Case Diagram..... | 16 |
| 2.2.2 Definisi Aktor | 18 |
| 2.2.3 Definisi Use Case..... | 19 |
| 2.2.4 Definisi Use Case Realization | 19 |
| 2.2.5 Spesifikasi Use Case | 21 |
| 2.3 Perancangan Data | 35 |
| 2.3.1 Deskripsi Data..... | 35 |
| 2.3.2 Kamus Data | 36 |
| 2.4 Dekomposisi Fisik Modul | 39 |
| 2.5 Deskripsi Rinci Modul..... | 40 |
| 2.5.1 Deskripsi Layar | 41 |
| 2.5.2 Deskripsi Proses | 67 |
| 2.5.3 Deskripsi Laporan | 68 |
| 3 Perancangan Antarmuka Manusia..... | 70 |

| | | |
|-------|--|----|
| 3.1 | Gambaran Umum Antarmuka Pengguna | 70 |
| 3.1.1 | Gagal Update | 70 |
| 3.1.2 | Berhasil Login | 70 |
| 3.1.3 | Notifikasi Penambahan Jadwal | 71 |
| 3.1.4 | Gagal Menambahkan..... | 71 |
| 3.1.5 | Update Berhasil | 72 |
| 3.1.6 | Berhasil Menghapus | 72 |
| 3.1.7 | Gagal Login..... | 73 |
| 3.1.8 | Gagal Menghapus..... | 73 |
| 3.2 | Tampilan layar | 74 |
| 3.2.1 | Home | 74 |
| 3.2.2 | Login | 76 |
| 3.2.3 | Home Admin | 78 |
| 3.2.4 | Home Customer | 80 |
| 3.2.5 | Tambah Jadwal..... | 82 |
| 3.2.6 | Update Jadwal | 84 |
| 3.2.7 | Riwayat | 86 |
| 3.2.8 | Booking Tiket | 88 |

Daftar Gambar

| | |
|---|----|
| Gambar 2.2.1 Use Case Diagram..... | 16 |
| Gambar 2.5.1 Layout Layar | 41 |
| Gambar 2.5.2 Form Lihat Jadwal..... | 42 |
| Gambar 2.5.3 Tampilan setelah Form | 43 |
| Gambar 2.5.4 Form input Data Diri | 46 |
| Gambar 2.5.5 Launch Kode Pembayaran setelah Memasukkan Data Diri | 47 |
| Gambar 2.5.6 Form Check In..... | 50 |
| Gambar 2.5.7 Form Tambah Jadwal | 52 |
| Gambar 2.5.8 Form untuk Jadwal..... | 55 |
| Gambar 2.5.9 Form Update Jadwal | 57 |
| Gambar 2.5.10 Form Hapus jadwal | 60 |
| Gambar 2.5.11 Tampilan lihat riwayat | 63 |
| Gambar 2.5.12 form Hapus riwayat | 65 |
| Gambar 2.5.13 Tampilan Laporan | 69 |

Daftar Tabel

| | |
|--|----|
| Table 1.3.1 Tabel Definisi dan Istilah | 11 |
| Table 2.1.1 Rancangan Lingkungan Implementasi | 15 |
| Table 2.2.1 Tabel Definisi Aktor | 18 |
| Table 2.2.2 Tabel Definisi Use Case..... | 19 |
| Table 2.2.3 Definisi Use Case Realization | 19 |
| Table 2.2.4 Use Case Melakukan Pemesanan | 22 |
| Table 2.2.5 Use Case Melihat Jadwal..... | 24 |
| Table 2.2.6 Use Case Mengelola Riwayat Pesanan | 26 |
| Table 2.2.7 Use Case Membuat Laporan | 28 |
| Table 2.2.8 Use Case Mengelola Jadwal..... | 30 |
| Table 2.2.9 Use case Mengelola Boarding Pass | 34 |
| Table 2.3.1 Deskripsi Data..... | 35 |
| Table 2.3.2 User | 36 |
| Table 2.3.3 Gerbong | 36 |
| Table 2.3.4 Stasiun | 36 |
| Table 2.3.5 Kereta | 36 |
| Table 2.3.6 Jadwal..... | 37 |
| Table 2.3.7 Pemesanan..... | 37 |
| Table 2.3.8 Keberangkatan..... | 38 |
| Table 2.3.9 Riwayat | 38 |
| Table 2.4.1 Dekomposisi Fisik Modul | 39 |
| Table 2.5.1 Deskripsi Rinci Modul..... | 40 |
| Table 2.5.2 Deskripsi objek gambar 2.5.2..... | 43 |
| Table 2.5.3 Deskripsi Objek gambar 2.5.3..... | 44 |
| Table 2.5.4 Deskripsi Objek gambar 2.5.4..... | 47 |
| Table 2.5.5 Deskripsi Objek untuk gambar2.5.5 | 48 |
| Table 2.5.6 Deskripsi Objek untuk gambar 2.5.6 | 50 |
| Table 2.5.7 Deskripsi Objek untuk gambar 2.5.7 | 53 |
| Table 2.5.8 Deskripsi Objek Cari Jadwal | 55 |

| | |
|--|----|
| Table 2.5.9 Deskripsi Objek gambar 2.5.9 | 58 |
| Table 2.5.10 Deskripsi Objek gambar 2.5.10 | 61 |
| Table 2.5.11 Deskripsi Objek Lihat Riwayat | 63 |
| Table 2.5.12 Deskripsi Objek Hapus riwayat | 65 |
| Table 2.5.13 Deskripsi Laporan | 68 |
| Table 2.5.14 Deskripsi Masukkan | 69 |

Pendahuluan

1.1 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 pengembangan perangkat lunak.

1.2 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.

1.3 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. |
|--|--|-----------------------------|

1.4 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.
- Kelompok 3. *Spesifikasi Kebutuhan Perangkat Lunak*. Jurusan Teknik Informatika Telkom University

1.5 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.

2 Deskripsi Perancangan

2.1 Rancangan Lingkungan Implementasi

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

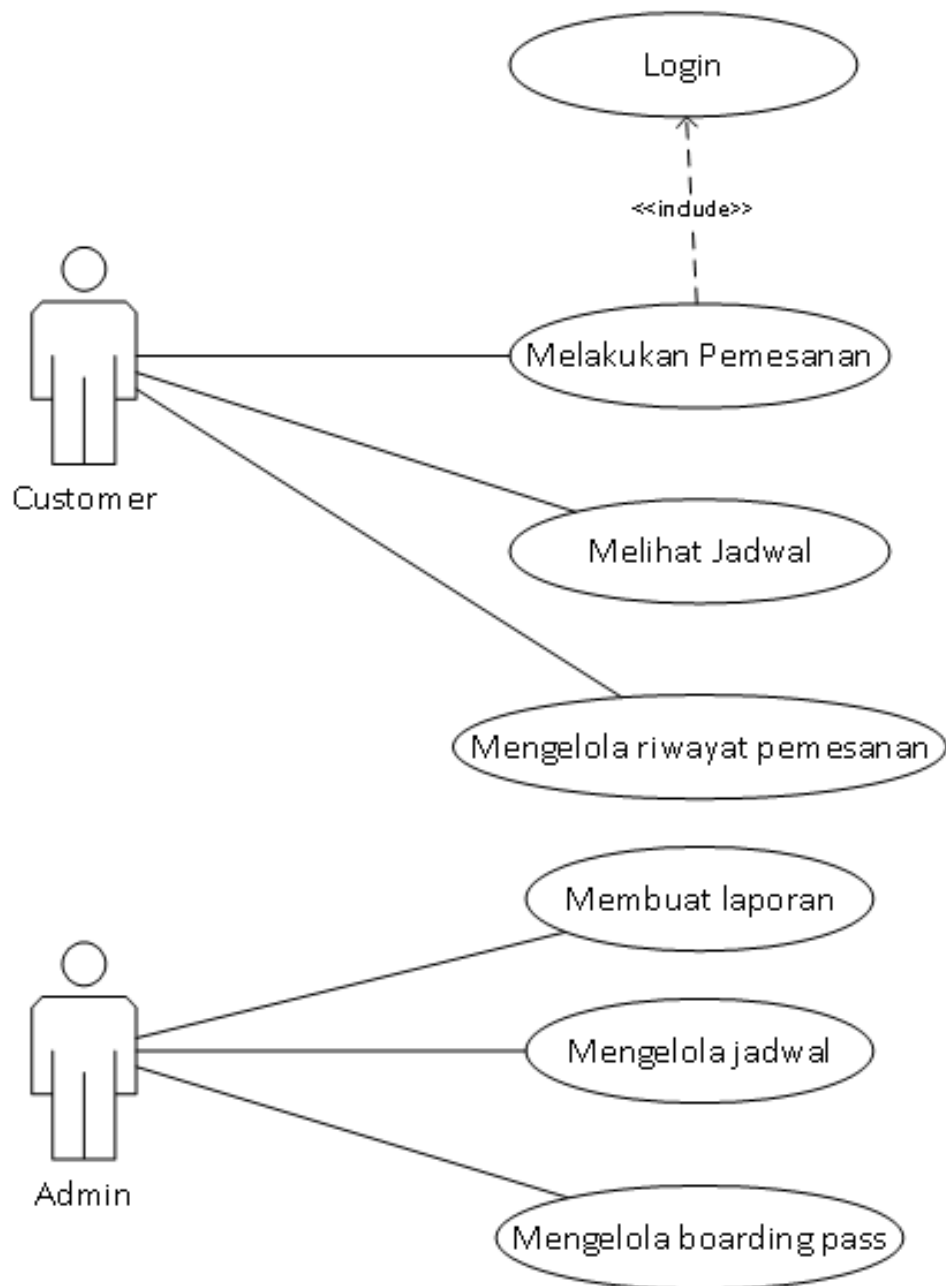
Table 2.1.1 Rancangan Lingkungan Implementasi

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

2.2 Dekomposisi Fungsional Modul

2.2.1 Use Case Diagram

Use Case Diagram bisa 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 halnya 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

2.2.2 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 | Mengelola data penjadwalan kereta, melakukan validasi terhadap customer, Membuat laporan harian | 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 |

2.2.3 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 | Mengelola Riwayat Pemesanan |
| UC-004 | Membuat Laporan |
| UC-005 | Mengelola Jadwal |
| UC-006 | Melakukan Check-In (Boarding Pass) |

2.2.4 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-003-02 | Menghapus Riwayat | Use Case Realization ini diakses oleh user (customer) untuk menghapus riwayat pemesanan |
| UCR-004-01 | Membuat Laporan | Use Case Realization ini diakses oleh user (admin) untuk membuat laporan harian yang berisi data keberangkatan, data penumpang dan data pemesanan. |
| UCR-005-01 | Menambahkan Jadwal | Use Case Realization ini diakses oleh user (admin) untuk menambahkan jadwal keberangkatan kereta. |
| UCR-005-02 | Menghapus Jadwal | Use Case Realization ini diakses oleh user (admin) untuk menghapus jadwal keberangkatan kereta. |
| UCR-005-03 | Menyunting Jadwal | Use Case Realization ini diakses oleh user (admin) untuk mengubah jadwal keberangkatan kereta. |
| UCR-006-01 | Melakukan Check-In | Use Case Realization ini diakses oleh user (admin) untuk melakukan boarding pass pengguna kereta api yang telah memiliki tiket. |

2.2.5 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.

2.2.5.1 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. |

2.2.5.2 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 |

2.2.5.3 Use Case Mengelola Riwayat Pesanan (UC-003)

Use case mengelola riwayat dapat dilihat pada tabel 2.2.6

Table 2.2.6 Use Case Mengelola Riwayat Pesanan

| | |
|---|--|
| Use Case Realization ID | UCR-003-01 |
| Use Case Realization Name | Melihat Riwayat |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (customer) untuk melihat riwayat pemesanan tiket kereta api yang telah dilakukan. |
| Aktor | Customer |
| Pre Condition | Customer ingin melihat riwayat pemesanan tiket |
| Post Condition | Customer mengetahui riwayat pemesanan tiket yang telah dilakukan |
| Normal Flow | |
| Aksi Aktor | Reaksi Sistem |
| 1. Customer membuka laman home | |
| | 2. Sistem menampilkan laman home |
| 3. Customer menekan tombol riwayat pemesanan | |
| | 4. Sistem mengambil data dari database pemesanan berdasarkan id customer |
| | 5. Sistem menampilkan riwayat pemesanan dari customer tersebut |
| Alternate Flow | |
| 3.1 Customer menekan tombol back atau kembali | |

| | |
|--|--|
| | 3.1.1 Sistem akan menampilkan laman home |
| Use Case Realization ID | UCR-003-02 |
| Use Case Realization Name | Menghapus Riwayat |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (customer) untuk menghapus riwayat pemesanan tiket kereta api yang telah dilakukan. |
| Aktor | Customer |
| Pre Condition | Terdapat data yang akan dihapus pada daftar riwayat pemesanan |
| Post Condition | Terhapus nya data yang dipilih pada daftar riwayat pemesanan. |
| Normal Flow | |
| 1. Customer menekan tombol laman riwayat pemesanan | |
| | 2. Sistem membuka laman riwayat pemesanan |
| | 3. Sistem mengambil data dari database pemesanan berdasarkan id customer dan menampilkannya dilaman riwayat pemesanan |
| 4. Customer memilih riwayat yang ingin dihapus | |
| 5. Customer menekan tombol hapus | |
| | 6. Sistem melakukan penghapusan pada database |
| | 7. Sistem mengeluarkan notifikasi sukses |

2.2.5.4 Use Case Membuat Laporan (UC-004)

Use case membuat laporan dapat dilihat pada tabel 2.2.7

Table 2.2.7 Use Case Membuat Laporan

| | |
|---|--|
| Use Case Realization ID | UCR-004-01 |
| Use Case Realization Name | Membuat Laporan |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (admin) untuk membuat laporan harian yang berisi data keberangkatan, data penumpang dan data pemesanan. |
| Aktor | Admin |
| Pre Condition | Admin ingin membuat laporan harian untuk diberikan pada Manager |
| Post Condition | Laporan harian terbentuk dari data pemesanan dan data keberangkatan |
| Normal Flow | |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin menekan tombol membuat laporan | |
| | 2. Sistem membuka laman laporan |
| 3. Admin memilih antara membuat laporan data pemesanan atau data keberangkatan. | |
| | 4. Sistem akan mengambil data dari database sesuai pilihan yang dimasukkan oleh Admin |
| | 5. Sistem menjadikan table ke dokumen pdf supaya bisa didownload oleh Admin |

| | |
|----------------------------------|--|
| 6. Admin menekan tombol download | |
|----------------------------------|--|

2.2.5.5 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 |

2.2.5.6 Use Case Mengelola Boarding Pass (UC-006)

Use case mengelola boarding pass dapat dilihat pada tabel 2.2.9

Table 2.2.9 Use case Mengelola Boarding Pass

| | |
|---|--|
| Use Case Realization ID | UCR-006-01 |
| Use Case Realization Name | Melakukan Check In |
| Deskripsi Singkat | Use Case Realization ini diakses oleh user (admin) untuk melakukan check in terhadap customer atau penumpang kereta api. |
| Aktor | Admin |
| Pre Condition | Penumpang kereta tidak diperkenankan untuk naik kereta sebelum melakukan check in |
| Post Condition | Penumpang diperkenankan untuk naik kereta api |
| Normal Flow | |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin memindai tiket atau memasukkan kode booking yang dimiliki oleh penumpang | |
| | 2. Sistem mencari di database pemesanan, apakah tiket yang dimiliki penumpang valid |
| | 3. Jika ya, maka data yang valid tersebut masuk ke dalam database keberangkatan |
| | 4. Sistem mengeluarkan notifikasi sukses |

2.3 Perancangan Data

2.3.1 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 |
|---------------|--------|--------|------|-------------|--|
| User | Master | - | - | username | |
| Stasiun | Master | - | - | idstasiun | |
| Kereta | Master | - | - | idkereta | |
| Jadwal | Relasi | - | - | idjadwal | Stasiun(idstasiun), Kereta(idkereta),ger bong(idgerbong) |
| Pemesanan | Relasi | - | - | noktp | User(username), Jadwal(idjadwal) |
| Keberangkatan | Relasi | - | - | idboarding | Pemesanan(noktp) |
| Gerbong | Master | - | - | idgerbong | |
| Riwayat | Relasi | - | - | idriwayat | Pemesanan(noktp) |

2.3.2 Kamus Data

Kamus data bisa dilihat pada tabel 2.3.2 – tabel 2.3.9

Table 2.3.2 User

| Atribut | Tipe Data | PK/FK |
|---------------|-------------|-------|
| username | VARCHAR(20) | PK |
| password | VARCHAR(20) | |
| tempat_lahir | VARCHAR(30) | |
| tanggal_lahir | DATE | |
| gender | VARCHAR(10) | |
| nama | VARCHAR(50) | |
| email | VARCHAR(30) | |
| jenis_login | VARCHAR(10) | |
| nohp | VARCHAR(15) | |

*Table
2.3.3
Gerbong*

| Atribut | Tipe Data | PK/FK |
|-----------|-------------|-------|
| kelas | VARCHAR(10) | |
| kapasitas | INT | |
| idgerbong | VARCHAR(10) | PK |
| harga | NUMBER | |

Table 2.3.4 Stasiun

| Atribut | Tipe Data | PK/FK |
|---------------|-------------|-------|
| idstasiun | VARCHAR(10) | PK |
| stasiun_awal | VARCHAR(20) | |
| stasiun_akhir | VARCHAR(20) | |

Table 2.3.5 Kereta

| Atribut | Tipe Data | PK/FK |
|-------------|-------------|-------|
| nama_kereta | VARCHAR(20) | |
| idkereta | VARCHAR(10) | PK |

Table 2.3.6 Jadwal

| Atribut | Tipe Data | PK/FK |
|-----------------|-------------|-------|
| waktu_berangkat | Date | |
| waktu_sampai | Date | |
| idkereta | VARCHAR(10) | FK |
| igerbong | VARCHAT(10) | FK |
| idstasiun | VARCHAR(10) | FK |
| idjadwal | VARCHAR(10) | PK |

Table 2.3.7 Pemesanan

| Atribut | Tipe Data | PK/FK |
|-------------------|-------------|-------|
| noktp | VARCHAR(16) | PK |
| namapemesan | VARCHAR(30) | |
| tempatlahirpesan | VARCHAR(20) | |
| tanggallahirpesan | DATE | |
| idjadwal | VARCHAR(10) | FK |
| username | VARCHAR(20) | FK |
| status | VARCHAR(10) | |
| tanggal_hariini | DATE | |

Table 2.3.8 Keberangkatan

| Atribut | Tipe Data | PK/FK |
|------------|-------------|-------|
| idboarding | VARCHAR(10) | PK |
| noktp | VARCHAR(16) | FK |

Table 2.3.9 Riwayat

| Atribut | Tipe Data | PK/FK |
|-----------|-------------|-------|
| noktp | VARCHAR(16) | FK |
| idriwayat | VARCHAR(10) | PK |

2.4 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 |
| | Boarding Pass | checkin.php | Tabel Pemesanan | Tabel Keberangkatan |
| Kelola Jadwal | Tambah Jadwal | tambahjadwal.php | Input form | Tabel jadwal |
| | Update Jadwal | updatejadwal.php | Field pencarian | Tabel jadwal |
| | Hapus Jadwal | hapusjadwal.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 |

2.5 Deskripsi Rinci Modul

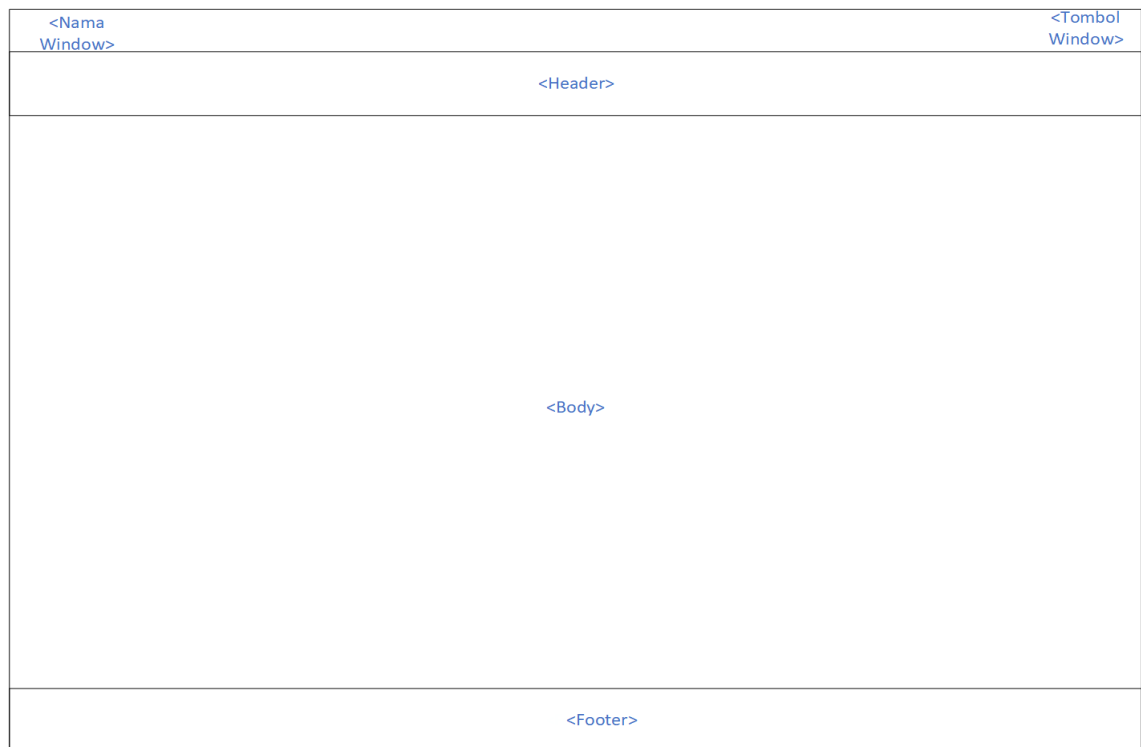
Deskripsi rinci modul bisa dilihat pada tabel 2.5.1

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 |
| Boarding _pass | Menginputkan nomor identitas yang dimiliki oleh penumpang |
| Tambah_jadwal | Menginputkan data jadwal keberangkatan |
| Cari_jadwal | Mencari jadwal sesuai dengan inputan yang dimasukkan |
| Update_jadwal | Mengupdate jadwal keberangkatan yang diinginkan |
| Hapus_jadwal | Menghapus jadwal keberangkatan yang diinginkan |
| Lihat_riwayat | Tampilan riwayat keberangkatan yang dilakukan oleh penumpang |
| Hapus riwayat | Menghapus riwayat keberangkatan yang dilakukan oleh penumpang |
| Laporan_harian | Membuat laporan dari semua data keberangkatan. |

2.5.1 Deskripsi Layar

Layout dari setiap halaman bisa dilihat pada gambar 2.5.1



Gambar 2.5.1 Layout Layar

2.5.1.1 Lihat List Jadwal

Lihat list jadwal bisa dilihat pada gambar 2.5.2 dan 2.5.3

| | |
|---|--------|
| Nama/Logo Web | Logout |
| <div><div>St. Awal <input type="text"/></div><div>St. Akhir <input type="text"/></div><div>Tanggal <input type="text"/></div><div>Kelas <input type="text"/></div><div>Submit</div></div> | |
| Copyright | |

Gambar 2.5.2 Form Lihat Jadwal

| | |
|---------------|-------------------------------------|
| Nama/Logo Web | Logout |
| List Jadwal | |
| Jadwal 1 | <input type="button" value="Book"/> |
| Jadwal 2 | <input type="button" value="Book"/> |
| Jadwal 3 | <input type="button" value="Book"/> |
| DST | <input type="button" value="DST"/> |
| Copyright | |

Gambar 2.5.3 Tampilan setelah Form

2.5.1.1.1 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. |

2.5.1.1.2 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 dicocokkan 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.

2.5.1.2 Booking Tiket

Berikut layar yang digunakan untuk booking tiket

| | |
|---|--------|
| Nama/Logo Web | Logout |
| <p>Data Diri</p> <div><p>No. Identitas <input type="text"/></p><p>Nama <input type="text"/></p><p>Tempat Lahir <input type="text"/></p><p>Tanggal Lahir <input type="text"/></p><p>Next</p></div> | |
| Copyright | |

Gambar 2.5.4 Form input Data Diri

| | |
|---|--------|
| Nama/Logo Web | Logout |
| <div style="border: 1px solid black; padding: 10px; margin: 20px auto; width: 60%;"> <p>Kode Pembayaran</p> <p>Kode : 998237462374</p> <p>Silahkan lakukan pembayaran dengan maksimal waktu 1x24 jam</p> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="OK"/> </div> </div> | |
| Copyright | |

Gambar 2.5.5 Launch Kode Pembayaran setelah Memasukkan Data Diri

2.5.1.2.1 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 |

| | | |
|------------------|--------|---|
| inputtanggalahir | 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 |

2.5.1.2.2 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,
tanggalahirpesan,idjadwal,username,status,tanggal_hariini) VALUES
('$noktp','$nama','$tl','$tglhr','$idjadwal','$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.

2.5.1.3 Boarding Pass

Berikut merupakan layout layout untuk kegiatan boarding pass

The image shows a web application layout for a boarding pass check-in. At the top, there is a header bar with 'Nama/Logo Web' on the left and 'Logout' on the right. Below the header, the main content area is titled 'CHECK IN'. In the center of this area is a rectangular box containing the text 'No Identitas', a text input field, and a button labeled 'CHECK IN'. At the bottom of the page, there is a footer bar with the word 'Copyright'.

Gambar 2.5.6 Form Check In

2.5.1.3.1 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 |

2.5.1.3.2 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

2.5.1.4 Tambah Jadwal

Berikut merupakan layout untuk tambah jadwal

| | |
|---|--------|
| Nama/Logo Web | Logout |
| <div>Tambah Jadwal</div> <div><div>Jadwal Berangkat</div><div>Jadwal Sampai</div><div>ID Gerbong</div><div>ID Kereta</div><div>Stasiun Awal</div><div>Stasiun Akhir</div><div>ADD</div></div> | |
| Copyright | |

Gambar 2.5.7 Form Tambah Jadwal

2.5.1.4.1 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 |

2.5.1.4.2 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

2.5.1.5 Cari Jadwal

Berikut layout untuk mencari jadwal

| | | | | | | | | | | | | | |
|--|----------------------|------------------|----------------------|---------------|----------------------|-----------|----------------------|--------------|----------------------|---------------|----------------------|-------------------------------------|--|
| Nama/Logo Web | Logout | | | | | | | | | | | | |
| Update Jadwal | | | | | | | | | | | | | |
| <table border="1"><tr><td>Jadwal Berangkat</td><td><input type="text"/></td></tr><tr><td>Jadwal Sampai</td><td><input type="text"/></td></tr><tr><td>ID Kereta</td><td><input type="text"/></td></tr><tr><td>Stasiun Awal</td><td><input type="text"/></td></tr><tr><td>Stasiun Akhir</td><td><input type="text"/></td></tr><tr><td colspan="2" style="text-align: right;"><input type="button" value="Cari"/></td></tr></table> | | Jadwal Berangkat | <input type="text"/> | Jadwal Sampai | <input type="text"/> | ID Kereta | <input type="text"/> | Stasiun Awal | <input type="text"/> | Stasiun Akhir | <input type="text"/> | <input type="button" value="Cari"/> | |
| Jadwal Berangkat | <input type="text"/> | | | | | | | | | | | | |
| Jadwal Sampai | <input type="text"/> | | | | | | | | | | | | |
| ID Kereta | <input type="text"/> | | | | | | | | | | | | |
| Stasiun Awal | <input type="text"/> | | | | | | | | | | | | |
| Stasiun Akhir | <input type="text"/> | | | | | | | | | | | | |
| <input type="button" value="Cari"/> | | | | | | | | | | | | | |
| Copyright | | | | | | | | | | | | | |

Gambar 2.5.8 Form untuk Jadwal

2.5.1.5.1 Deskripsi Objek

Berikut merupakan deskripsi objek bisa dilihat pada tabel 2.5.8

Table 2.5.8 Deskripsi Objek Cari Jadwal

| Nama Objek | Jenis Objek | Keterangan |
|----------------|-------------|--|
| inputberangkat | Date | Input jadwal berangkat untuk parameter pencarian |
| inputsampai | Date | Input jadwal sampai untuk parameter pencarian |
| inputidkereta | dropdown | Input idkereta untuk parameter pencarian |

| | | |
|-------------------|--------|---|
| Inputstasiunawal | Text | Input stasiunawal untuk parameter pencarian |
| inputstasiunakhir | Text | Input stasiun akhir untuk parameter pencarian |
| btncari | Button | Memulai proses pencarian jadwal |

2.5.1.5.2 Algoritma

```
SELECT idjadwal, waktu_berangkat, waktu_sampai, idkereta, awal, akhir
FROM jadwal;
```

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 dicari pada tabel jadwal di database. Algoritma melakukan pencarian record pada tabel jadwal dengan nilai cari dari input User.

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

2.5.1.6 Update Jadwal

Berikut merupakan layout update jadwal

| | | | | | | | | | | | | | | | |
|--|----------------------|-----------------|----------------------|--------------|----------------------|-----------|----------------------|--------------|----------------------|---------------|----------------------|------------|----------------------|---------------------------------------|--|
| Nama/Logo Web | Logout | | | | | | | | | | | | | | |
| <div>Update Jadwal</div> <div>Id Jadwal :</div> <div><table><tr><td>Waktu Berangkat</td><td><input type="text"/></td></tr><tr><td>Waktu Sampai</td><td><input type="text"/></td></tr><tr><td>ID Kereta</td><td><input type="text"/></td></tr><tr><td>Stasiun Awal</td><td><input type="text"/></td></tr><tr><td>Stasiun Akhir</td><td><input type="text"/></td></tr><tr><td>ID Gerbong</td><td><input type="text"/></td></tr><tr><td colspan="2"><input type="button" value="Update"/></td></tr></table></div> | | Waktu Berangkat | <input type="text"/> | Waktu Sampai | <input type="text"/> | ID Kereta | <input type="text"/> | Stasiun Awal | <input type="text"/> | Stasiun Akhir | <input type="text"/> | ID Gerbong | <input type="text"/> | <input type="button" value="Update"/> | |
| Waktu Berangkat | <input type="text"/> | | | | | | | | | | | | | | |
| Waktu Sampai | <input type="text"/> | | | | | | | | | | | | | | |
| ID Kereta | <input type="text"/> | | | | | | | | | | | | | | |
| Stasiun Awal | <input type="text"/> | | | | | | | | | | | | | | |
| Stasiun Akhir | <input type="text"/> | | | | | | | | | | | | | | |
| ID Gerbong | <input type="text"/> | | | | | | | | | | | | | | |
| <input type="button" value="Update"/> | | | | | | | | | | | | | | | |
| Copyright | | | | | | | | | | | | | | | |

Gambar 2.5.9 Form Update Jadwal

2.5.1.6.1 Deskripsi Objek

Deskripsi Objek bisa dilihat pada tabel 2.5.9

Table 2.5.9 Deskripsi Objek gambar 2.5.9

| 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 |
| Inputstasiunawal | Text | Input stasiunawal untuk dimasukkan ke dalam database jadwal |
| inputstasiunakhir | Text | Input stasiun akhir untuk dimasukkan ke dalam database jadwal |
| inputidgerbong | text | Input id gerbong untuk dimasukkan ke dalam database jadwal |
| btnupdate | Button | Memulai proses update jadwal kedalam database jadwal |

2.5.1.6.2 Algoritma

```
UPDATE jadwal SET waktu_berangkat='$time', waktu_sampai='$time2',  
idkereta='$idkereta', idgerbong='$idgerbong', awal='$awal', akhir='$akhir'  
WHERE idjadwal='$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 keyword yang kemudian akan dijadikan sebagai nilai atau value yang akan dicari pada tabel jadwal di database. Algoritma melakukan pencarian record pada tabel jadwal dengan nilai cari dari input User.

Setelah berhasil mendapatkan record yang didapatkan, isi dari record yang didapatkan akan diganti berdasarkan hasil dari input user. Maka dengan berjalanya algoritma ini, terdapat pergantian data pada tabel record

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

2.5.1.7 Hapus Jadwal

Berikut layout untuk hapus jadwal

| | | | |
|--|--------|---|--------|
| Nama/Logo Web | Logout | | |
| <p>Hapus Jadwal</p> <table border="1"><tr><td>Jawal yang ingin dihapus dan telah dicari</td><td>Delete</td></tr></table> | | Jawal yang ingin dihapus dan telah dicari | Delete |
| Jawal yang ingin dihapus dan telah dicari | Delete | | |
| Copyright | | | |

Gambar 2.5.10 Form Hapus jadwal

2.5.1.7.1 Deskripsi Objek

Berikut deskripsi objek gambar 2.5.10

Table 2.5.10 Deskripsi Objek gambar 2.5.10

| Nama Objek | Jenis Objek | Keterangan |
|------------|-------------|--|
| btndelete | button | Memulai proses penghapusan pada jadwal yang telah dicari |

2.5.1.7.2 Algoritma

```
DELETE FROM jadwal WHERE idjadwal='$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 keyword yang kemudian akan dijadikan sebagai nilai atau value yang akan dicari pada tabel jadwal di database. Algoritma melakukan pencarian record pada tabel jadwal dengan nilai cari dari input User.

Setelah berhasil mendapatkan record yang didapatkan, isi dari record yang didapatkan akan dihapus. Maka dengan berjalanya algoritma ini, terdapat penghapusan data pada tabel record

Dengan berhasilnya mekanisme tersebut, maka user customer berhasil menjalani mekanisme fungsionalitas delete jadwal yang tertera pada use case diagram .

2.5.1.8 Lihat Riwayat

Berikut merupakan layout lihat riwayat.

| | |
|-------------------|--------|
| Nama/Logo Web | Logout |
| Riwayat Pemesanan | |
| Riwayat 1 | Hapus |
| Riwayat 2 | Hapus |
| DST | Hapus |
| Copyright | |

Gambar 2.5.11 Tampilan lihat riwayat

2.5.1.8.1 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 |

2.5.1.8.2 Algoritma

```
SELECT * FROM riwayat;
```

Perlu diketahui bahwa algoritma hanya akan dieksekusi atau berjalan ketika user menekan sebuah button. input yang dilakukan oleh user akan membuat laman website melakukan pengalihan ke laman viewhistory.php yang berfungsi untuk menampilkan riwayat transaksi yang telah dilakukan oleh user.

Dengan menjalankan algoritma ini, user berhasil menjalankan mekanisme fungsionalitas mengolah riwayat pada use case diagram

2.5.1.9 Hapus Riwayat

Berikut merupakan layout hapus riwayat

| | |
|-------------------|--------|
| Nama/Logo Web | Logout |
| Riwayat Pemesanan | |
| Riwayat 1 | Hapus |
| Riwayat 2 | Hapus |
| DST | Hapus |
| Copyright | |

Gambar 2.5.12 form Hapus riwayat

2.5.1.9.1 Deskripsi Objek

Berikut merupakan deskripsi objek pada tabel

Table 2.5.12 Deskripsi Objek Hapus riwayat

| Nama Objek | Jenis Objek | Keterangan |
|--------------|-------------|--|
| tableriwayat | table | Tampilan untuk melihat riwayat pemesanan |
| btndelete | button | Memulai menghapus riwayat yang dipilih |

2.5.1.9.2 Algoritma

```
DELETE FROM riwayat WHERE noktp='$noktp';
```

Perlu diketahui bahwa algoritma hanya akan dieksekusi atau berjalan ketika user menekan sebuah button. input yang dilakukan oleh user akan membuat laman website melakukan pengalihan ke laman viewhistory.php yang berfungsi untuk menampilkan riwayat transaksi yang telah dilakukan oleh user.

Terdapat Button hapus yang akan menjadi event trigger untuk menjalankan algoritma tersebut, dengan menekan button tersebut maka algoritma delete akan mencari sebuah record dengan nilai \$nomorktp pada tabel riwayat

Setelah menemukan record tersebut, maka record akan dihapus.

Dengan berhasilnya mekanisme tersebut, User berhasil menjalankan proses Hapus Jadwal yang tertera pada use case diagram

Dengan menjalankan algoritma ini, user berhasil menjalankan mekanisme fungsionalitas mengolah riwayat pada use case diagram

2.5.2 Deskripsi Proses

Proses ini merupakan proses yang tidak ada campur tangan dari actor atau bahasa lainnya otomatis.

2.5.2.1 Laporan Pemasukkan

Laporan pemasukkan ini digunakan untuk mengkaji hasil evaluasi hasil pemasukkan perusahaan. Semua perusahaan butuh laporan pemasukkan untuk merencanakan apa yang akan dilakukan oleh perusahaan tersebut untuk kedepannya

2.5.2.1.1 Deskripsi Masukan

Masukkan atau input yang diperlukan dari pembuatan laporan ini adalah harga di setiap pemesanan pada tabel gerbong.

2.5.2.1.2 Deskripsi Keluaran

Keluaran yang akan dimunculkan adalah berupa total dari pemasukkan atau transaksi yang telah dilakukan

2.5.2.1.3 Algoritma

```
SELECT sum(harga) as Total FROM jadwal JOIN gerbong USING (idgerbong)
HAVING status='paid';
```

2.5.3 Deskripsi Laporan

Deskripsi Laporan dapat dilihat pada tabel 2.5.13

Table 2.5.13 Deskripsi Laporan

| Nama Layar | Keterangan |
|--------------------|---|
| Laporan Harian | Menampilkan laporan harian berdasarkan parameter tanggal dihari yang ingin dilihat |
| Laporan Pemasukkan | Merupakan proses tanpa mengandalkan interaksi actor, system langsung mengambil data dari database dan menghitung total pemasukkan |

2.5.3.1 Laporan Harian

Menampilkan laporan harian berdasarkan parameter tanggal dihari yang ingin dilihat. Berikut penjelasan tentang layout, masukan dan algoritma.

2.5.3.1.1 Lay out

Berikut layout pembuatan laporan

| | |
|---|--------|
| Nama/Logo Web | Logout |
| <p>Laporan</p> <div>List 1</div> <div>List 2</div> <div>DST</div> | |
| Copyright | |

Gambar 2.5.13 Tampilan Laporan

2.5.3.1.2 Deskripsi Masukan

Table 2.5.14 Deskripsi Masukan

| Nama Parameter | Keterangan |
|----------------|---|
| tanggal | Untuk melakukan query dan menampilkan transaksi pada tanggal yang diinputkan. |

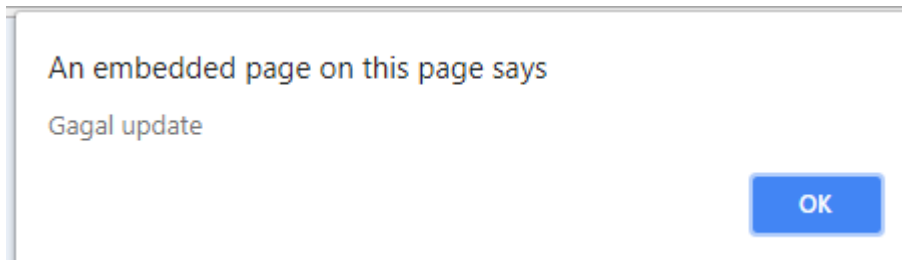
2.5.3.1.3 Algoritma

```
SELECT * FROM pemesanan WHERE (tanggal_hariini='$tanggal' and status='paid');
```

3 Perancangan Antarmuka Manusia

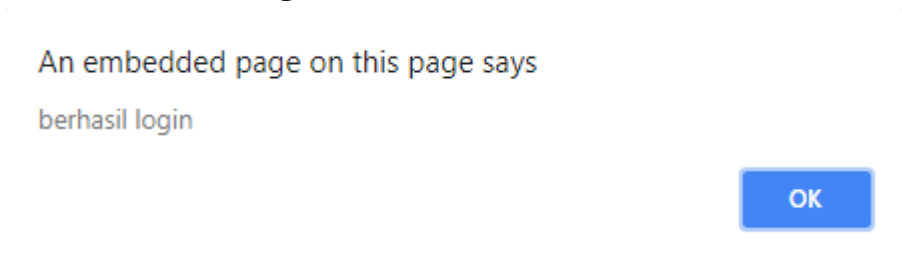
3.1 Gambaran Umum Antarmuka Pengguna

3.1.1 Gagal Update



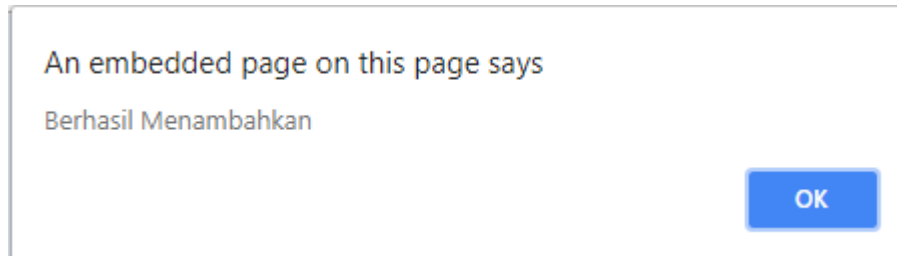
Komunikasi antar user dan system merupakan hal yang penting. Notifikasi berikut ditulis dalam bahasa javascript dan dipergunakan sebagai sebuah notifikasi berupa alert yang berfungsi untuk menginformasikan user bahwa terdapat kesalahan dalam melakukan perubahan jadwal. Dengan adanya notifikasi berikut maka terjadi komunikasi yang baik antar system dan user, guna menjalankan sebuah kesatuan system yang lebih baik.

3.1.2 Berhasil Login



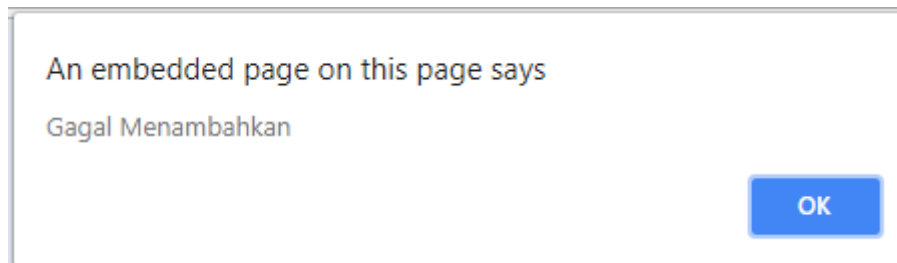
Komunikasi antar user dan system merupakan hal yang penting. Notifikasi berikut ditulis dalam bahasa javascript dan dipergunakan sebagai sebuah notifikasi berupa alert yang berfungsi untuk menginformasikan user bahwa terdapat kesuksesan dalam melakukan perubahan proses login. Dengan adanya notifikasi berikut maka terjadi komunikasi yang baik antar system dan user, guna menjalankan sebuah kesatuan system yang lebih baik.

3.1.3 Notifikasi Penambahan Jadwal



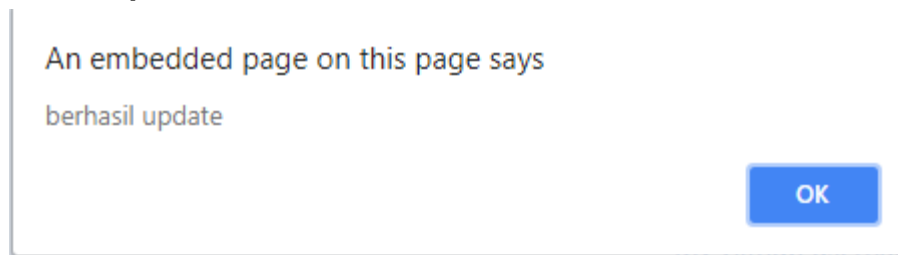
Komunikasi antar user dan system merupakan hal yang penting. Notifikasi berikut ditulis dalam bahasa javascript dan dipergunakan sebagai sebuah notifikasi berupa alert yang berfungsi untuk menginformasikan user bahwa terdapat kesuksesan dalam melakukan proses penambahan pada sistem. Dengan adanya notifikasi berikut maka terjadi komunikasi yang baik antar system dan user. guna menjalankan sebuah kesatuan system yang lebih baik.

3.1.4 Gagal Menambahkan



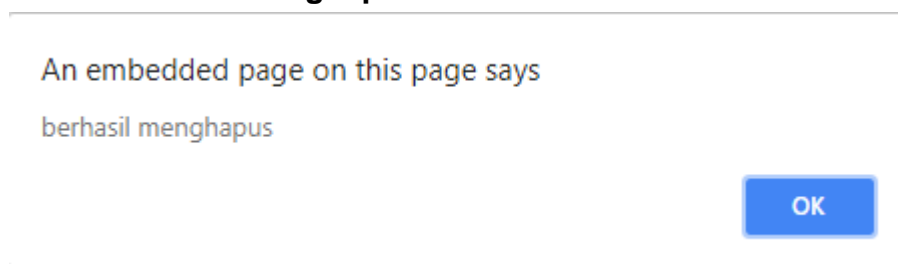
Komunikasi antar user dan system merupakan hal yang penting. Notifikasi berikut ditulis dalam bahasa javascript dan dipergunakan sebagai sebuah notifikasi berupa alert yang berfungsi untuk menginformasikan user bahwa terdapat kegagalan dalam melakukan perubahan proses pada sistem. Dengan adanya notifikasi berikut maka terjadi komunikasi yang baik antar system dan user. guna menjalankan sebuah kesatuan system yang lebih baik.

3.1.5 Update Berhasil



Komunikasi antar user dan system merupakan hal yang penting Notifikasi berikut ditulis dalam bahasa javascript dan dipergunakan sebagai sebuah notifikasi berupa alert yang berfungsi untuk menginformasikan user bahwa terdapat kesuksesan dalam melakukan mekanisme sebuah fungsionalitas. Dengan adanya notifikasi berikut maka terjadi komunikasi yang baik antar system dan user. guna menjalankan sebuah kesatuan system yang lebih baik.

3.1.6 Berhasil Menghapus



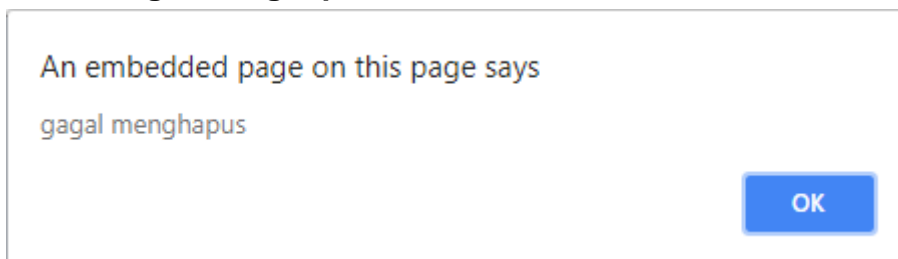
Komunikasi antar user dan system merupakan hal yang penting Notifikasi berikut ditulis dalam bahasa javascript dan dipergunakan sebagai sebuah notifikasi berupa alert yang berfungsi untuk menginformasikan user bahwa terdapat kesuksesan dalam melakukan mekanisme penghapusan record pada database. Dengan adanya notifikasi berikut maka terjadi komunikasi yang baik antar system dan user. guna menjalankan sebuah kesatuan system yang lebih baik.

3.1.7 Gagal Login



Komunikasi antar user dan system merupakan hal yang penting Notifikasi berikut ditulis dalam bahasa javascript dan dipergunakan sebagai sebuah notifikasi berupa alert yang berfungsi untuk menginformasikan user bahwa terdapat kegagalan dalam melakukan proses login. Dengan adanya notifikasi berikut maka terjadi komunikasi yang baik antar system dan user. guna menjalankan sebuah kesatuan system yang lebih baik.

3.1.8 Gagal Menghapus



Komunikasi antar user dan system merupakan hal yang penting Notifikasi berikut ditulis dalam bahasa javascript dan dipergunakan sebagai sebuah notifikasi berupa alert yang berfungsi untuk menginformasikan user bahwa terdapat kegagalan dalam melakukan proses penghapusan. Dengan adanya notifikasi berikut maka terjadi komunikasi yang baik antar system dan user. guna menjalankan sebuah kesatuan system yang lebih baik.

3.2 Tampilan layar

3.2.1 Home



Pada tampilan home, user diwajibkan untuk melakukan proses login dengan menekan tombol button. Hal ini diperlukan untuk men-segmentasi user yang pada fungsionalitasnya dibagi menjadi dua bagian. Yaitu admin dan user, admin dan user memiliki masing masing fungsionalitas yang berbeda. Dengan men-segmentasi user maka system akan dapat mendeteksi sebagai siapa user tersebut melakukan proses login, Dengan begitu maka system akan dapat melakukan navigasi kemana yang berbeda berdasarkan status login sang user. Pada website pengguna akan terdapat button yang menavigasi user berdasarkan status login mereka.

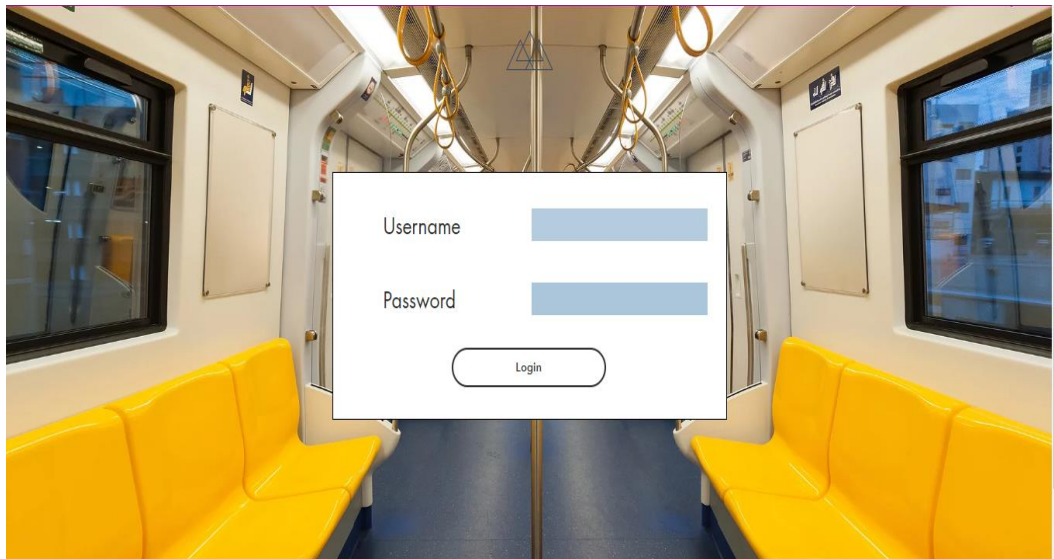
Namun, dengan adanya segmentasi pada status login user tidak berarti bahwa tidak terjadi transaksi antara user dan admin. Karena pada dasarnya keseluruhan kesatuan merupakan bentuk umum dari transaksi antar customer dan admin. Dapat disimpulkan bahwa segmentasi user hanyalah pembeda status user untuk login sebagai admin atau user.

Penjelasan dari segmentasi berikut digambarkan dalam deskripsi table dimana terdapat status login pada user. Setelah user melakukan proses login, system akan melakukan pemeriksaan dengan sintaks if yang

akan mendeteksi jenis dari status login tersebut. Setelah mendapatkan status user kemudian akan membedakan jenis login dan menavigasi user berdasarkan status login yang dimiliki.. apabila login tersebut dideteksi dengan status admin, maka user akan dialihkan ke laman homepage admin, dan begitu pula apabila statusnya sebagai user maka navigasi akan beralih dari laman homepage ke laman homepage customer yang masing masing memiliki fungsionalitas berbeda didalamnya.

User tidak akan bisa mendapatkan akses ke halaman admin, namun admin dapat melakukan login sebagai user dengan membuat akun sebagai user.

3.2.2 Login



Pada tampilan home, user diwajibkan untuk melakukan proses login dengan menekan tombol button. Hal ini diperlukan untuk men-segmentasi user yang pada fungsionalitasnya dibagi menjadi dua bagian. Yaitu admin dan user, admin dan user memiliki masing masing fungsionalitas yang berbeda. Dengan men-segmentasi user maka system akan dapat mendeteksi sebagai siapa user tersebut melakukan proses login, Dengan begitu maka system akan dapat melakukan navigasi kemana yang berbeda berdasarkan status login sang user. Pada website pengguna akan terdapat button yang menavigasi user berdasarkan status login mereka.

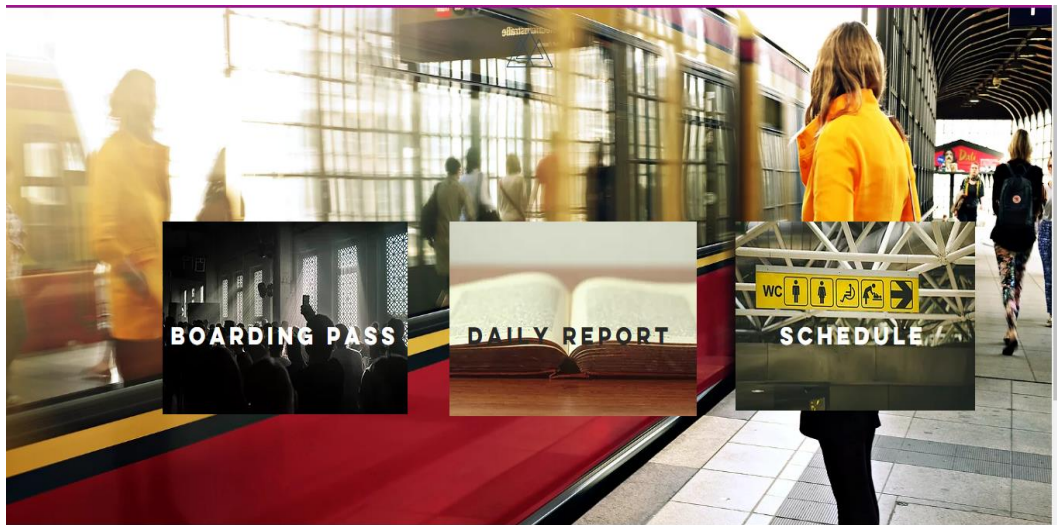
Namun, dengan adanya segmentasi pada status login user tidak berarti bahwa tidak terjadi transaksi antara user dan admin. Karena pada dasarnya keseluruhan kesatuan merupakan bentuk umum dari transaksi antar customer dan admin. Dapat disimpulkan bahwa segmentasi user hanyalah pembeda status user untuk login sebagai admin atau user.

Penjelasan dari segmentasi berikut digambarkan dalam deskripsi table dimana terdapat status login pada user. Setelah user melakukan proses login, system akan melakukan pemeriksaan dengan sintaks if yang akan mendeteksi jenis dari status login tersebut. Setelah mendapatkan status user kemudian akan membedakan jenis login dan menavigasi user

berdasarkan status login yang dimiliki.. apabila login tersebut dideteksi dengan status admin, maka user akan dialihkan ke laman homepage admin, dan begitu pula apabila statusnya sebagai user maka navigasi akan beralih dari laman homepage ke laman homepage customer yang masing masing memiliki fungsionalitas berbeda didalamnya.

User tidak akan bisa mendapatkan akses ke halaman admin, namun admin dapat melakukan login sebagai user dengan membuat akun sebagai user.

3.2.3 Home Admin



Pada tampilan home, user diwajibkan untuk melakukan proses login dengan menekan tombol button. Hal ini diperlukan untuk men-segmentasi user yang pada fungsionalitasnya dibagi menjadi dua bagian. Yaitu admin dan user, admin dan user memiliki masing masing fungsionalitas yang berbeda. Dengan men-segmentasi user maka system akan dapat mendeteksi sebagai siapa user tersebut melakukan proses login, Dengan begitu maka system akan dapat melakukan navigasi kemana yang berbeda berdasarkan status login sang user. Pada website pengguna akan terdapat button yang menavigasi user berdasarkan status login mereka.

Namun, dengan adanya segmentasi pada status login user tidak berarti bahwa tidak terjadi transaksi antara user dan admin. Karena pada dasarnya keseluruhan kesatuan merupakan bentuk umum dari transaksi antar customer dan admin. Dapat disimpulkan bahwa segmentasi user hanyalah pembeda status user untuk login sebagai admin atau user.

Penjelasan dari segmentasi berikut digambarkan dalam deskripsi table dimana terdapat status login pada user. Seteah user melakukan proses login, system akan melakukan pemeriksaan dengan sintaks if yang akan mendeteksi jenis dari status login tersebut. Setelah mendapatkan status user kemudian akan membedakan jenis login dan menavigasi user berdasarkan status login yang dimiliki.. apabila login tersebut dideteksi

dengan status admin, maka user akan dialihkan ke laman homepage admin, dan begitu pula apabila statusnya sebagai user maka navigasi akan beralih dari laman homepage ke laman homepage customer yang masing masing memiliki fungsionalitas berbeda didalamnya.

User tidak akan bisa mendapatkan akses ke halaman admin, namun admin dapat melakukan login sebagai user dengan membuat akun sebagai user.

Laman diatas merupakan homepage dari admin yang berisi 3 buah button. Yaitu Boarding pass, pembuatan jadwal, dan juga pembuatan laporan. Boarding pass merupakan button yang akan melakukan navigasi ke halaman boarding pass yang digunakan untuk melakukan pemeriksaan user yang telah melakukan pembayaran. Pembuatan jadwal adalah button yang akan menavigasi admin untuk melakukan pengolahan jadwal yang berisi update, tambah dan hapus jadwal.

3.2.4 Home Customer



Pada tampilan home, user diwajibkan untuk melakukan proses login dengan menekan tombol button. Hal ini diperlukan untuk men-segmentasi user yang pada fungsionalitasnya dibagi menjadi dua bagian. Yaitu admin dan user, admin dan user memiliki masing masing fungsionalitas yang berbeda. Dengan men-segmentasi user maka system akan dapat mendeteksi sebagai siapa user tersebut melakukan proses login, Dengan begitu maka system akan dapat melakukan navigasi kemana yang berbeda berdasarkan status login sang user. Pada website pengguna akan terdapat button yang menavigasi user berdasarkan status login mereka.

Namun, dengan adanya segmentasi pada status login user tidak berarti bahwa tidak terjadi transaksi antara user dan admin. Karena pada dasarnya keseluruhan kesatuan merupakan bentuk umum dari transaksi antar customer dan admin. Dapat disimpulkan bahwa segmentasi user hanyalah pembeda status user untuk login sebagai admin atau user.

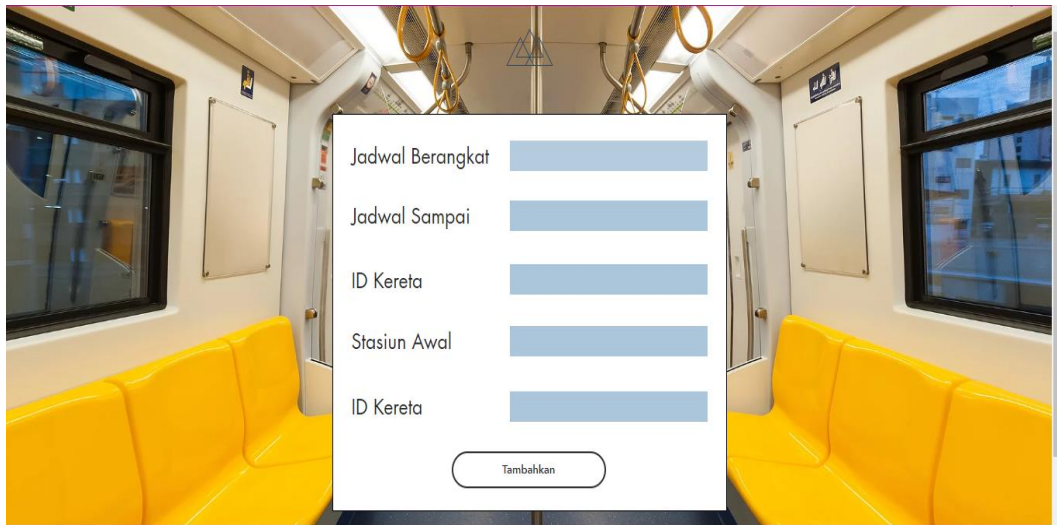
Penjelasan dari segmentasi berikut digambarkan dalam deskripsi table dimana terdapat status login pada user. Setelah user melakukan proses login, system akan melakukan pemeriksaan dengan sintaks if yang akan mendeteksi jenis dari status login tersebut. Setelah mendapatkan

status user kemudian akan membedakan jenis login dan menavigasi user berdasarkan status login yang dimiliki.. apabila login tersebut dideteksi dengan status admin, maka user akan dialihkan ke laman homepage admin, dan begitu pula apabila statusnya sebagai user maka navigasi akan beralih dari laman homepage ke laman homepage customer yang masing masing memiliki fungsionalitas berbeda didalamnya.

User tidak akan bisa mendapatkan akses ke halaman admin, namun admin dapat melakukan login sebagai user dengan membuat akun sebagai user.

Laman diatas merupakan homepage dari user yang berisi 3 buah button. Yaitu Boarding pass, lihat jadwal, dan juga pengolahan riwayat pemesanan . Boarding pass merupakan button yang akan melakukan navigasi ke halaman boarding pass yang digunakan untuk melakukan pemesanan tiket berdasarkan jadwal yang diinginkan. Pembuatan jadwal adalah button yang akan menavigasi admin untuk melakukan pengolahan jadwal yang berisi update, tambah dan hapus jadwal.

3.2.5 Tambah Jadwal



Pada tampilan home, user diwajibkan untuk melakukan proses login dengan menekan tombol button. Hal ini diperlukan untuk men-segmentasi user yang pada fungsionalitasnya dibagi menjadi dua bagian. Yaitu admin dan user, admin dan user memiliki masing masing fungsionalitas yang berbeda. Dengan men-segmentasi user maka system akan dapat mendeteksi sebagai siapa user tersebut melakukan proses login, Dengan begitu maka system akan dapat melakukan navigasi kemana yang berbeda berdasarkan status login sang user. Pada website pengguna akan terdapat button yang menavigasi user berdasarkan status login mereka.

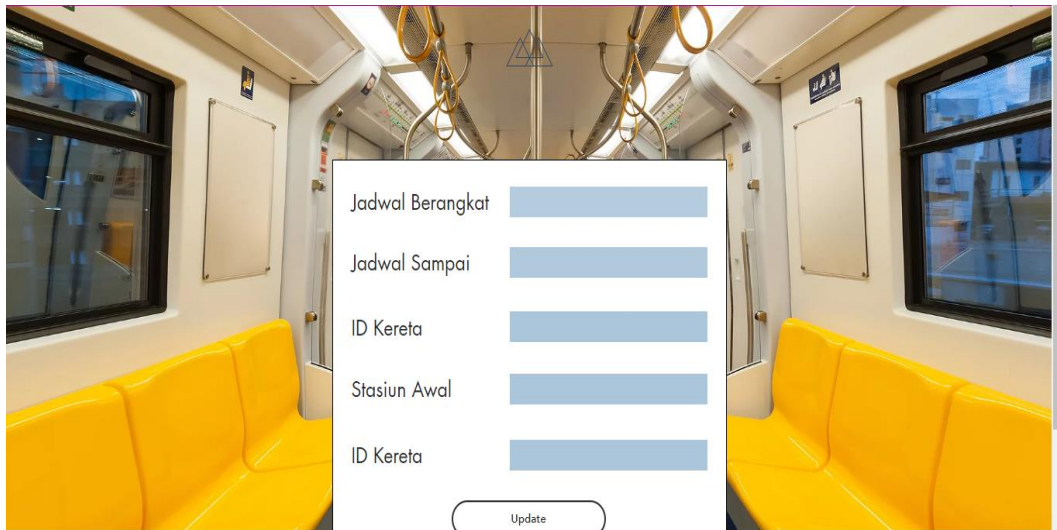
Namun, dengan adanya segmentasi pada status login user tidak berarti bahwa tidak terjadi transaksi antara user dan admin. Karena pada dasarnya keseluruhan kesatuan merupakan bentuk umum dari transaksi antar customer dan admin. Dapat disimpulkan bahwa segmentasi user hanyalah pembeda status user untuk login sebagai admin atau user.

Penjelasan dari segmentasi berikut digambarkan dalam deskripsi table dimana terdapat status login pada user. Seteah user melakukan proses login, system akan melakukan pemeriksaan dengan sintaks if yang akan mendeteksi jenis dari status login tersebut. Setelah mendapatkan status user kemudian akan membedakan jenis login dan menavigasi user berdasarkan status login yang dimiliki.. apabila login tersebut dideteksi

dengan status admin, maka user akan dialihkan ke laman homepage admin, dan begitu pula apabila statusnya sebagai user maka navigasi akan beralih dari laman homepage ke laman homepage customer yang masing masing memiliki fungsionalitas berbeda didalamnya.

Laman berikut merupakan laman penambahan jadwal yang merupakan fungsionalitas dari admin untuk menambahkan jadwal pada table jadwal di database system informasi. Dengan menambahkan jadwal maka secara otomatis tiket yang dapat dibeli oleh user menjadi lebih bervariasi karena tiket mereferensikan jadwal sebagai sebuah record pada table tersebut. Laman tambah jadwal memuat form action dengan method post dimana terdapat 6 buah input (termasuk input bertipe submit). Record input tersebut merupakan sebuah string yang didalamnya akan ditambahkan menjadi sebuah baris baru pada database jadwal. Laman berikut hanya dapat diakses oleh admin, karena user tidak memiliki fungsionalitas untuk mengubah jadwal.

3.2.6 Update Jadwal



Pada tampilan home, user diwajibkan untuk melakukan proses login dengan menekan tombol button. Hal ini diperlukan untuk men-segmentasi user yang pada fungsionalitasnya dibagi menjadi dua bagian. Yaitu admin dan user, admin dan user memiliki masing masing fungsionalitas yang berbeda. Dengan men-segmentasi user maka system akan dapat mendeteksi sebagai siapa user tersebut melakukan proses login, Dengan begitu maka system akan dapat melakukan navigasi kemana yang berbeda berdasarkan status login sang user. Pada website pengguna akan terdapat button yang menavigasi user berdasarkan status login mereka.

Namun, dengan adanya segmentasi pada status login user tidak berarti bahwa tidak terjadi transaksi antara user dan admin. Karena pada dasarnya keseluruhan kesatuan merupakan bentuk umum dari transaksi antar customer dan admin. Dapat disimpulkan bahwa segmentasi user hanyalah pembeda status user untuk login sebagai admin atau user.

Penjelasan dari segmentasi berikut digambarkan dalam deskripsi table dimana terdapat status login pada user. Seteah user melakukan proses login, system akan melakukan pemeriksaan dengan sintaks if yang akan mendeteksi jenis dari status login tersebut. Setelah mendapatkan status user kemudian akan membedakan jenis login dan menavigasi user

berdasarkan status login yang dimiliki.. apabila login tersebut dideteksi dengan status admin, maka user akan dialihkan ke laman homepage admin, dan begitu pula apabila statusnya sebagai user maka navigasi akan beralih dari laman homepage ke laman homepage customer yang masing masing memiliki fungsionalitas berbeda didalamnya.

Laman berikut merupakan laman penambahan jadwal yang merupakan fungsionalitas dari admin untuk menambahkan jadwal pada table jadwal di database system infomasi. Dengan menambahkan jadwal maka secara otomatis tiket yang dapat dibeli oleh user menjadi lebih bervariasi karena tiket mereferensikan jadwal sebagai sebuah record pada table tersebut. Laman tambah jadwal memuat form action dengan method post dimana terdapat 6 buah input (termasuk input bertipe submit). Record input tersebut merupakan sebuah string yang didalamnya akan ditambahkan menjadi sebuah record baru yang merupakan hasil pembaruan dari record yang sebelumnya telah ada. Laman berikut hanya dapat diakses oleh admin, karena user tidak memiliki fungsionalitas untuk mengubah jadwal.

3.2.7 Riwayat



Pada tampilan home, user diwajibkan untuk melakukan proses login dengan menekan tombol button. Hal ini diperlukan untuk men-segmentasi user yang pada fungsionalitasnya dibagi menjadi dua bagian. Yaitu admin dan user, admin dan user memiliki masing masing fungsionalitas yang berbeda. Dengan men-segmentasi user maka system akan dapat mendeteksi sebagai siapa user tersebut melakukan proses login, Dengan begitu maka system akan dapat melakukan navigasi kemana yang berbeda berdasarkan status login sang user. Pada website pengguna akan terdapat button yang menavigasi user berdasarkan status login mereka.

Namun, dengan adanya segmentasi pada status login user tidak berarti bahwa tidak terjadi transaksi antara user dan admin. Karena pada dasarnya keseluruhan kesatuan merupakan bentuk umum dari transaksi antar customer dan admin. Dapat disimpulkan bahwa segmentasi user hanyalah pembeda status user untuk login sebagai admin atau user.

Penjelasan dari segmentasi berikut digambarkan dalam deskripsi table dimana terdapat status login pada user. Seteah user melakukan proses login, system akan melakukan pemeriksaan dengan sintaks if yang akan mendeteksi jenis dari status login tersebut. Setelah mendapatkan status user kemudian akan membedakan jenis login dan menavigasi user

berdasarkan status login yang dimiliki.. apabila login tersebut dideteksi dengan status admin, maka user akan dialihkan ke laman homepage admin, dan begitu pula apabila statusnya sebagai user maka navigasi akan beralih dari laman homepage ke laman homepage customer yang masing masing memiliki fungsionalitas berbeda didalamnya.

Laman ini merupakan laman yang merepresentasikan fungsionalitas dari user untuk melihat riwayat pemesanan dan penghapusan riwayat pemesanan tersebut. Tampilan riwayat tersebut diambil dari table riwayat. pengambilan data record mengacu pada id user yang login. Laman login berisi button yang akan menjalankan algoritma untuk melakukan penghapusan record yang dipilih.

3.2.8 Booking Tiket



| | | | | | | |
|----------------------------------|-----------------------|-------------------------------|-----------------|-------------------------------|-----------|--------|
| Serayu Economy (IC) | 00:32 Kiaracondong | → | 03:33 Bekasi | 3h 1m | Rp 67.000 | Choose |
| | | Train Details | | Price Details | | |
| Serayu Economy (IC) | 00:51 Cimahi | → | 03:33 Bekasi | 2h 42m | Rp 67.000 | Choose |
| | | Train Details | | Price Details | | |
| Argo Parahyangan Economy (IC) | 05:00 Bandung | → | 07:41 Bekasi | 2h 41m Direct | Rp 90.000 | Choose |
| | | Train Details | | Price Details | | |
| Argo Parahyangan Economy (IC) | 05:12 Cimahi | → | 07:41 Bekasi | 2h 29m Direct | Rp 90.000 | Choose |

Pada tampilan home, user diwajibkan untuk melakukan proses login dengan menekan tombol button. Hal ini diperlukan untuk men-segmentasi user yang pada fungsionalitasnya dibagi menjadi dua bagian. Yaitu admin dan user, admin dan user memiliki masing masing fungsionalitas yang berbeda. Dengan men-segmentasi user maka system akan dapat mendeteksi sebagai siapa user tersebut melakukan proses login, Dengan begitu maka system akan dapat melakukan navigasi kemana yang berbeda

berdasarkan status login sang user. Pada website pengguna akan terdapat button yang menavigasi user berdasarkan status login mereka.

Namun, dengan adanya segmentasi pada status login user tidak berarti bahwa tidak terjadi transaksi antara user dan admin. Karena pada dasarnya keseluruhan kesatuan merupakan bentuk umum dari transaksi antar customer dan admin. Dapat disimpulkan bahwa segmentasi user hanyalah pembeda status user untuk login sebagai admin atau user.

Penjelasan dari segmentasi berikut digambarkan dalam deskripsi table dimana terdapat status login pada user. Setelah user melakukan proses login, system akan melakukan pemeriksaan dengan sintaks if yang akan mendeteksi jenis dari status login tersebut. Setelah mendapatkan status user kemudian akan membedakan jenis login dan menavigasi user berdasarkan status login yang dimiliki.. apabila login tersebut dideteksi dengan status admin, maka user akan dialihkan ke laman homepage admin, dan begitu pula apabila statusnya sebagai user maka navigasi akan beralih dari laman homepage ke laman homepage customer yang masing masing memiliki fungsionalitas berbeda didalamnya.

Laman berikut merupakan laman booking ticket atau merupakan laman yang menampilkan tampilan fisik dari ticket yang telah berhasil dipesan. Ticket tersebut berisi data diri berupa nama dari pemesan, id ticket, dan notifikasi untuk melakukan pembayaran dalam jangka waktu 24 jam.