**ANALISIS PEMESANAN TIKET KERETA API *LOCAL* MELALUI APLIKASI KAI *ACCESS***

**LAPORAN PROYEK I**

Diajukan untuk memenuhi kelulusan mata kuliah Proyek I  
pada Program Studi DIV Teknik Informatika

Oleh :

**Aditya Rahman   
1.18.4.021**



**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA**

**POLITEKNIK POS INDONESIA**

**BANDUNG**

**2019**

**SURAT PERNYATAAN**

**TIDAK MELAKUKAN PLAGIARISME**

Yang bertanda tangan di bawah ini:

Nama : Aditya Rahman

NPM : 1.18.4.021

Program Studi : DIV Teknik Informatika

Judul : Analisis Pemesanan Tiket Kereta Api *Local* Melalui Aplikasi   
 KAI *ACCESS*

Menyatakan bahwa:

1. Proyek Pemrograman Aplikasi (PROYEK I) saya ini adalah asli dan belum pernah diajuakan untuk memenuhi kelulusan matakuliah Proyek I pada Program Studi DIV Teknik Informatika baik di Politeknik Pos Indonesia maupun di Perguruan Tinggi lainnya.
2. Proyek Pemrograman Aplikasi (PROYEK I) ini adalah murni gagasan, rumusan, dan penelitian saya sendiri tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam Proyek Pemrograman Aplikasi (PROYEK I) ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan-penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku diperguruan tinggi lain.

Bandung, 14 Agustus 2019

Yang Membuat Pernyataan,

**ADITYA RAHMAN**

**NPM. 1.18.4.021**

# ABSTRAK

Moda transportasi masal yang masih ada hingga saat ini adalah kereta api, khususnya dipulau jawa masih diminati oleh masyarakat sebagai angkutan penumpang dan barang. Maka dari itu PT. KAI (persero) selaku penyelenggara angkutan jasa kereta api harus memiliki sistem pelayanan terbaik, dari segi pemesanan dan pembelian tiket kereta api, yang relatif efektif dan efisien waktu.

Tujuan analisis ini untuk mempelajari system aplikasi pemesanan dan pembelian tiket kereta api *local*, pemesanan dan pembelian tiket kereta api tidak lagi harus mengantri di loket stasiun, tetapi pemesanan bisa dilakukan melalui aplikasi mobile phone yang bernama KAI *ACCESS*, aplikasi KAI *ACCESS* mengeluarkan fitur baru yaitu *LOCAL TRAIN*.

*LOCAL TRAIN* dapat digunakan oleh calon penumpang yang hendak menaikin kereta api jarak pedek, untuk menikmati perjalanan yang menghubungkan antar kota dengan kota lain dengan jarak tempuh waktu yang lebih efektif, serta dapat langsung mengatur jadwal dari H-7 sampai 2 jam sebelum keberangkatan, oleh karena itu PT. Kereta Api Indonesia memberikan kemudahkan calon penumpang untuk mendapatkan informasi dan melakukan pemesanan tiket kereta api secara online dimanapun dan kapanpun tanpa harus datang ke loket stasiun.

Kata Kunci : Sistem, Informasi, Aplikasi, Pemesanan.

# *ABSTRACT*

*Mass transportation modes that still exist today are trains, especially on the island of Java that are still in demand by the community as railroad and goods transportation. Therefore PT. KAI (Persero) as the train operator must have the best service system, in terms of ordering and purchasing train tickets, which is relatively effective and efficient.*

*The purpose of this analysis is for the application system for ordering and purchasing local train ticket, booking and purchasing train tickets no longer have to queue at the counter station, but reservations can be made through a mobile application called KAI ACCESS, KAI ACCESS application, new features namely LOCAL TRAIN.*

*LOCAL TRAIN can be used by prospective passengers who take the train, to enjoy a trip that connects between cities and other cities with a more effective travel time, and can be directly accessed by flight schedule H-7 to 2 hours before being invited, therefore PT. Kereta Api Indonesia provides prospective passengers to get information and make train ticket reservations online wherever and whenever without having to come to the ticket window.*

*Keywords: System, Information, Application, Ordering.*

# KATA PENGANTAR

Puji Syukur atas kehadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya sehingga pada akhirnya laporan analisis aplikasi yang berjudul “Analisis Pemesanan Tiket Kereta Api *Local* Melalui Aplikasi KAI *ACCESS*” ini dapat selesai pada waktunya sebagai syarat kelulusan matakuliah proyek 1.

Dalam proses penulisan laporan ini banyak kendala yang dihadapi mulai dari pencarian data dan informasi penunjang laporan ini merupakan salah satu kendala yang penulis hadapi. Akan tetapi berkat dukungan dan semangat yang diberikan oleh rekan-rakan terdekat serta tekad yang kuat akhirnya penulis dapat menyelesaikan laporan ini.

Kiranya tidak ada ucapan yang lebih indah kecuali terimakasih yang sebesar-besarnya kepada:

1. DR. Ir. Agus Purnomo selaku direktur Politeknik Pos Indonesia.

2. [M. Yusril Helmi Setyawan, S.Kom., M.Kom](https://forlap.ristekdikti.go.id/dosen/detail/MkY5NjkwRUEtQzY1Ri00NDBGLUFGNjAtNjE5N0RFQjgwNUI2). Selaku ketua program studi D-IV Teknik Informatika.

3. Rd. Nuraini Siti Fatonah, S.S., M.Hum. Selaku koordinator proyek 1 yang telah memberikan arahan dan bimbingan kepada penulis.

4. Roni Habibi, S.Kom., M.T. Selaku pembimbing proyek 1 yang telah memberikan arahan dan bimbingan kepada penulis .

5. Semua pihak yang tidak bisa penulis sebutkan satu per satu.

Akhir kata penulis berharap semoga laporan ini bisa bermanfaat bagi para pembaca khususnya penulis sendiri karena penulis sadar bahwasannya laporan ini masih banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu penulis sangat mengharapkan kritik dan saran yang membangun agar bisa memotivasi penulis supaya lebih baik kedepannya.

|  |
| --- |
| Bandung, 14 Agustus 2019 |
| Penyusun |

**DAFTAR ISI**

ABSTRAK i

*ABSTRACT* . ii

KATA PENGANTAR iii

DAFTAR ISI iv

DAFTAR GAMBAR vii

DAFTAR TABEL viii

DAFTAR SIMBOL ix

BAB I PENDAHULUAN I-1

1.1 Latar Belakang I-1

1.2 Identifikasi Masalah I-1

1.3 Tujuan I-1

1.4 Ruang Lingkup I-2

1.5 Sistematika Penulisan I-2

BAB II LANDASAN TEORI II-3

2.1 Kereta Api II-3

2.2 Sistem Informasi II-3

2.3 *Database* II-3

2.4 *Flowmap* II-4

2.5 *Internet* II-4

2.6 My SQL *(My structured query language)* II-4

2.7 Tiket II-4

2.8 KAI *Access* II-4

2.9 *E-Wallet* II-4

2.10 Analisis sistem II-5

2.11 XAMPP II-5

2.10 UML *(Unified Modeling Language)* II-5

BAB III ANALISIS III-9

3.1 Analisis Sistem III-9

3.1.1 Analisis sistem yang sedang berjalan III-9

3.1.1.1 Analisis Prosedur III-9

3.1.1.2 Analisis dokumen yang digunakan III-11

3.1.2 Analisis Sistem yang akan dibangun III-11

3.1.2.1 Kebutuhan Fungsional III-14

3.1.2.2 Kebutuhan Non Fungsional III-15

3.2 Perancangan UML III-17

3.2.1 *Usecase* Diagram III-17

3.2.2 *Class* Diagram III-18

3.2.3 *Sequence* Diagram III-19

3.2.4 *Activity* Diagram III-20

3.2.5 *Component* Diagram III-21

3.2.6 *Develpoment* Diagram III-22

3.2.7 Perancangan Antarmuka III-23

3.2.7.1 Tampilan *login* III-23

3.2.7.2 *Flowchart* algoritma *login* III-24

3.2.7.3 Tampilan *form* pemesanan III-25

3.2.7.4 *Flowchart* algoritma *form* pemesanan III-26

3.2.7.5 Tampilan keberangkatan III-27

3.2.7.6 *Flowchart* algoritma keberangkatan III-27

3.2.7.7 Tampilan data penumpang III-28

3.2.7.8 *Flowchart* algoritma data penumpang III-28

3.2.7.9 Tampilan pembayaran III-30

3.2.7.10 *Flowchart* algoritma pembayaran III-31

3.2.7.11 Tampilan tiket III-32

3.2.7.12 *Flowchar*t algoritma tiket III-32

BAB IV KESIMPULAN DAN SARAN IV-34

4.1 Kesimpulan IV-34

4.2 Saran IV-34

Daftar pustaka 35

**DAFTAR GAMBAR**

Gambar 3.1 *Flowmap* analisis *system* yang sedang berjalan pada pemesanan tiket kereta api III-10

Gambar3.2 *Flowmap* pemesanan dan pembelian tiket kereta api *local* III-13

Gambar 3.3 *Usecase* Diagram III-17

Gambar 3.4 *Class* Diagram III-18

Gambar 3.5 *Sequence* Diagram III-19

Gambar 3.6 *Acivity* Diagram III-20

Gambar 3.7 *Component* diagram III-21

Gambar 3.8 *Deployment* Diagram III-22

Gambar 3.9 Tampilan *login* III-23

Gambar 3.10 Flowchart algoritma *login* III-24

Gambar 3.11 Tampilan form pemesanan III-25

Gambar 3.12 *Flowchart* algoritma *form* pemesanan III-26

Gambar 3.13 Tampilan keberangkatan III-27

Gambar 3.14 *Flowchart* algoritma keberangkatan III-27

Gambar 3.15 Tampilan data penumpang III-28

Gambar 3.16 *Flowchart* algoritma data penumpang III-28

Gambar 3.17 Tampilan pembayaran III-30

Gambar 3.18 *Flowchart* algoritma pembayaran III-31

Gambar 3.19 Tampilan *E*-*ticket* III-32

Gambar 3.20 *Flowchart* algoritma *E-ticket* III-32

DAFTAR TABEL

[Tabel 3. 1 Dokumen yang digunakan III-11](#_Toc17243400)

[Tabel 3. 2 Deskripsi perangkat keras server III-15](#_Toc17243401)

[Tabel 3. 3 Deskripsi perangkat keras client III-15](#_Toc17243402)

[Tabel 3. 4 Deskripsi perangkat lunak server III-16](#_Toc17243403)

[Tabel 3. 5 Deskripsi Perangkat Lunak Client III-16](#_Toc17243404)

**DAFTAR SIMBOL**

Simbol Flowmap

*Tabel 1: Daftar Simbol Flowmap*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1 |  | Terminal / *Terminator* | Menunjukkan awal atau akhir dari aliran proses. |
| 2 |  | *Arrows* | Menunjukan arus data antar simbol / proses. |
| 3 |  | Data | Menunjukkan data yang menjadi input / output proses. |
| 4 |  | *Process* | Menunjukan kegiata proses dari operasi program komputer. |
| 5 |  | *Decision* | Menunjukan pilihan yang akan dikerjakan atau keputusan yang harus dibuat dalam proses pengolahan data. |
| 6 |  | *Preparation* | Pemberian nilai awal suatu *variable*. |
| 7 |  | *Connector* (*On-page connector*) | Digunakan untuk penghubung dalam satu halaman. |
| 8 |  | *Connector* (*Off-page connector*) | Digunakan untuk penghubung berbeda halaman. |
| 9 |  | *Document* | Menunjukan dokumen sebagai yang digunakan untuk merekam data terjadinya suatu transaksi. |
| 10 |  | *Database* | Menyimpan data berbasis *database.* |
| 11 |  | Manual *Operation* | Menunjukan proses yang dikerjakan secara manual. |

Simbol *Usecase*

*Tabel 2: Daftar Simbol Usecase*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

Simbol *Class* Diagram

*Tabel 3: Daftar Simbol class diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Generalization* | Hubungan dimana objek anak *(descendent)*  berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung  padanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

Simbol *Sequence* Diagram

*Tabel 3: Daftar Simbol Sequence* Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

Simbol *Activity* Diagram

*Tabel 3: Daftar Simbol Activity* Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| **1** |  | *Actifity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| **2** |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| **3** |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| **4** |  | *Actifity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| **5** |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

Simbol *Component*Diagram

*Tabel 3: Daftar Simbol Component*Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 | [http://4.bp.blogspot.com/-sc_hs41py7c/VR5gbgH2yHI/AAAAAAAAAXU/xAgOjf7Nc-g/s1600/Snap%2B2015-04-03%2Bat%2B16.41.02.png](http://4.bp.blogspot.com/-sc_hs41py7c/VR5gbgH2yHI/AAAAAAAAAXU/xAgOjf7Nc-g/s1600/Snap+2015-04-03+at+16.41.02.png) | *Package* | Package merupakan sebuah bungkusan dari satu atau lebih komponen. |
| 2 | [http://2.bp.blogspot.com/-jw5E7Ur1GXo/VR5gbhk-eTI/AAAAAAAAAW0/dW5iIAUFVvY/s1600/Snap%2B2015-04-03%2Bat%2B16.40.48.png](http://2.bp.blogspot.com/-jw5E7Ur1GXo/VR5gbhk-eTI/AAAAAAAAAW0/dW5iIAUFVvY/s1600/Snap+2015-04-03+at+16.40.48.png) | Komponen | Komponen system. |
| 3 | [http://4.bp.blogspot.com/-jIjEiK0Ysw0/VR5gc0GJ37I/AAAAAAAAAXI/Kfw5Iz4HjZ0/s1600/Snap%2B2015-04-03%2Bat%2B16.41.40.png](http://4.bp.blogspot.com/-jIjEiK0Ysw0/VR5gc0GJ37I/AAAAAAAAAXI/Kfw5Iz4HjZ0/s1600/Snap+2015-04-03+at+16.41.40.png) | *Ketergantungan (Dependency)* | Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai. |
| 4 | [http://1.bp.blogspot.com/-mXyAjAh_BVc/VR5gbpHDlXI/AAAAAAAAAW4/OxGU0-56nEc/s1600/Snap%2B2015-04-03%2Bat%2B16.41.14.png](http://1.bp.blogspot.com/-mXyAjAh_BVc/VR5gbpHDlXI/AAAAAAAAAW4/OxGU0-56nEc/s1600/Snap+2015-04-03+at+16.41.14.png) | *Antarmuka / interface* | Sama dengan interface pada pemrograman berbasis objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen. |
| 5 | [http://1.bp.blogspot.com/-Dv5DVFKzu40/VR5gcaBNhKI/AAAAAAAAAXA/58V0dnrmLpo/s1600/Snap%2B2015-04-03%2Bat%2B16.41.23.png](http://1.bp.blogspot.com/-Dv5DVFKzu40/VR5gcaBNhKI/AAAAAAAAAXA/58V0dnrmLpo/s1600/Snap+2015-04-03+at+16.41.23.png) | *Link* | Relasi antar komponen. |

# BAB I

# PENDAHULUAN

**1.1** **Latar Belakang**

Saat ini perkembangan teknologi semakin cepat semua aktivitas yang dilakukan oleh sebuah bidang usaha dan jasa semakin tidak lepas dari pengaruh teknologi dan informasi, pesatnya perkembangan ini menuntut suatu sistem menjadi lebih baik. Karena itu sangat penting dipergunakan untuk memberikan pelayanan terbaik misalnya dalam pembelian dan pemesanan tiket kereta api jarak pendek.

Dahulu sistem kinerja pelayanan yang masih manual menjadi lebih kurang baik. seringkali pada saat masyarakat membeli tiket ke stasiun terdapat masalah yang sering dihadapi yaitu terjadi adanya antrian panjang yang menyebabkan ketidak efisien waktu serta penumpukan orang yang rela menunggu lama didepan loket stasiun untuk mendapatkan tiket kereta api jarak pendek.

Seiring dengan sistem layanan sekarang, sudah ada aplikasi yang dikeluarkan oleh PT.KAI (persero) yang bernama KAI *Access*, dalam aplikasi ini terdapat fitur *local train*, *local train* adalah fitur tambahan untuk menerapkan sistem pemesanan dan pembelian tiket kereta api jarak pendek dimanapun dengan *connection* *internet* yang memadai.

**1.2 Identifikasi Masalah**

Berdasarkan latar belakang diatas, maka dapat identifikasi masalah sebagai berikut :

1. Akses pemesanan tiket yang masih manual.
2. Proses pembelian melalui loket distasiun yang tidak efektif.

**1.3 Tujuan**

Adapun tujuannya sebagai berikut :

1. KAI *Access* membantu pemesanan dan pembelian tiket kereta api.

**1.4 Ruang Lingkup**

Ruang ligkup pembahasan kami adalah menganalisis bagaimana cara kerja suatu fitur dalam aplikasi yang bernama KAI *Access* fitur tersebut adalah fitur *local train*. *Local train* itu sendiri adalah pemesanan tiket kereta api jarak pendek.

**1.5 Sistematika Penulisan**

Berdasarkan latar belakang dan perumusan masalah diatas, maka penyusunan laporan ini dibuat dalam suatu sistematika yang terdiri dalam empat bab, yaitu.:

**BAB I PEDAHULUAN**

Berisi tentang keadaan atau kondisi yang melatar belakangi kasus yang akan dianalisis. Pada bagian pendahuluan terdapat beberapa topik yang isinya latar belakang, identifikasi masalah, tujuan, ruang lingkup dan sistematika penulisan.

**BAB II LANDASAN TEORI**

Bab ini berisikan teori-teori yang menunjang terbentuk suatu laporan yang berhubungan dengan apa yang sedang dianalisis.

**BAB III ANALISIS**

Bab ini membahas tentang penganalisisan sistem yang terdapat dalam sebuah aplikasi dengan menggambarkan proses bisnis pembentukan suatu aplikasi dengan menggunakan *flowmap*. Pada bab ini juga merupakan inti dari laporan yang dibuat.

**BAB IV KESIMPULAN DAN SARAN**

Pada bab ini berisi kesimpulan dan saran yang berkaitan dengan analisa sistem yang telah diuraikan pada bab-bab sebelumnya.

# BAB II

## LANDASAN TEORI

1. **Kereta Api**

Kereta api merupakan alat transportasi darat bertenaga uap ataupun listrik yang terdiri dari rangkaian gerbong yang ditarik oleh lokomotif dan media yang digunakan untuk berjalan adalah menggunakan rel[1]. kereta api pada saat ini merupakan transportasi yang banyak dipilih oleh penumpang karena selain harganya yang terjangkau dan juga pelayanannya juga cukup memuaskan serta pihak kereta api juga sangat mementingkan kenyamanan dari penumpang.

1. **Sistem Informasi**

Menurut O’Brien & Marakas (dikutip dalam Hartanto, 2018), menyatakan sistem informasi dapat didefinisikan sebagai suatu kombinasi yang terorganisir dari orang, perangkat keras, perangkat lunak, jaringan komunikasi, sumber data serta kebijakan dan prosedur yang menyimpan, memperoleh kembali, mengubah dan menyebarkan informasi[2].

1. ***Database***

*Database* adalah sekumpulan informasi yang tersimpan di komputer secara sistematis, sehingga mudah untuk diakses oleh sebuah program komputer untuk memperoleh informasi data. Perangkat lunak akan memanggil *query* basis data (DBMS) *database management system*. Penggunaan kata *database* kini tidak hanya di dunia komputer, telah meluas di luar bidang komputer. Secara umum konsep dasar data adalah sekumpulan catatan, potongan-potongan pengetahuan yang tersusun secara sistematis dan teratur yang memiliki penjelasan, atau disebut skema. Skema menggambarkan obyek yang diwakili suatu *database* dan ada hubungan antar obyek tersebut[5].

1. ***Flowmap***

*Flowmap* atau bagan alir adalah bagan yang menunjukkan aliran di dalam program atau prosedur sistem secara logika. *Flowmap* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu[3]. Pembuatan *flowmap* harus dapat memudahkan pemakai dalam memahami alur dari sistem atau transaksi.

1. ***Internet***

*Internet* adalah sistem global jaringan komputer yang saling berhubungan yang menggunakan standar *transmission control protocol/internet protocol (TCP/IP)* untuk melayani miliaran pengguna di seluruh dunia. *Internet* membawa berbagai macam sumber informasi dan layanan, seperti antar-*linked hypertext* dokumen dari *World Wide Web (WWW)* dan infrastruktur untuk mendukung surat elektronik.[6]

1. **My SQL *(My Structured Query Language)***

Menurut Astria (2016:15), “*MySQL* adalah sebuah basis data yang mengandung satu atau jumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel”. Menurut Wahana Komputer (dikutip dalam Astria, 2016), “*MySQL* adalah *database* server *open source* yang cukup popular keberadaannya, dengan berbagai keunggulan yang dimiliki, membuat *software* *database* ini banyak digunakan oleh praktisi untuk membangun suatu project. Adanya fasilitas (API) *Application Programming Interface* yang dimiliki oleh *MySQL*, memungkinkan bermacam – macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data *MySQL*. [7]

1. **Tiket**

Tiket adalah suatu dokumen perjalanan yang dikeluarkan oleh suatu perusahaan yang berisi rute, tanggal, harga, data penumpang yang digunakan untuk melakukan suatu perjalanan[11].

1. **KAI Access**

Kereta Api Indonesia (KAI) *Access* adalah *Official Mobile Application* dari PT. Kereta Api Indonesia untuk memudahkan calon penumpang mendapatkan informasi dan melakukan pemesanan tiket kereta api secara online dimanapun dan kapanpun, dengan Aplikasi ini, pemesanan tiket menjadi mudah karena bisa dilakukan di rumah sambil menonton TV bersama keluarga untuk kereta yang sudah terjadwalkan sebelumnya. Jadwal yang dikeluarkan *online real time* dari server kereta api indonesia sendiri[12].

**2.9 *E-Wallet***

*E-Wallet* atau dompet elektronik adalah alat pembayaran digital yang menggunakan media elektronik berupa server based. Pada umumnya *e-wallet*

berupa aplikasi yang berbasis di server dan dalam proses pemakaiannya memerlukan sebuah koneksi terlebih dulu dengan penerbitnya[8]. Terdapat beberapa *e-wallet* yang saat ini ada di pasar yaitu LinkAja Telkomsel, XL Tunai, Rekening Ponsel CIMB Niaga, BBM *Money* Permata Bank, DOKU, dan lain sebagainya[8].

**2.10 Analisis Sistem**

Menurut Jogiyanto (dikutip dalam Hanik, 2014) Analisis dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya[9].

**2.11 XAMPP**

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri *(localhost)*, yang terdiri atas program *Apache* HTTP Server, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache, MySQL*, PHP dan Perl. Program ini tersedia dalam (GPL) *General Public License* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis[10].

**2.12 UML(*Unified Modeling Language)***

**A. Pengertian UML**

UML adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem[4].

**B. Komponen-komponen UML**

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu *Grady Booch,* OOD (*Object-Oriented Design*), *Jim Rumbaugh*, OMT (*Object Modelling Technique*), dan *Ivar Jacobson* OOSE (*Object-Oriented Software Engineering*)[13].

Pada UML terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu :

**1. Struktur Diagram**

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Struktur diagram dalam UML terdiri atas :

A. *Class* diagram

*Class* diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class* diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai[14]. Selama tahap desain, *class* diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

B. *Object* diagram

*Object* diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak[14].

C. *Component* diagram

*Component* diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan[14].

D. *Deployment* diagram

*Deployment* diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment* diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan *system*[14].

E. *Composite* diagram

Sebuah diagram struktur *composite* mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi[14].

F. *Package* diagram

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software,* atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam pohon model[14].

**2. *Behavior* Diagram**

Menggambarkan ciri-ciri perilaku / interaksi / metode / fungsi dari sebuah sistem atau *business process*. *Behavior* diagram dalam UML terdiri atas :

A. *Usecase* diagram

Diagram yang menggambarkan aktor, *usecase* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk actor[15]. Sebuah *usecase* digambarkan sebagai elips horizontal dalam suatu diagram UML use case.

B. *Activity* diagram

Menggambarkan aktifitas - aktifitas, objek, *state*, transisi *state* dan *event*[15]. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

**3. *Interaction* Diagram**

Bagian dari *behavior* diagram yang menggambarkan interaksi objek. *Interaction* diagram dalam UML terdiri atas :

A. *Sequence* diagram

*Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *usecase* diagram[16].

B. *Communication* Diagram

Serupa dengan *sequence* diagram, tetapi diagram komunikasi juga digunakan untuk memodelkan perilaku dinamis dari *usecase*. Bila dibandingkan dengan *sequence* diagram, diagram komunikasi lebih terfokus pada menampilkan kolaborasi benda daripada urutan waktu[16].

C. *Interaction Overview* Diagram

Interaksi *overview* diagram berfokus pada gambaran aliran kendali interaksi dimana node adalah interaksi atau kejadian interaksi[16].

D. *Timing* diagram

*Timing* diagram di UML didasarkan pada diagram waktu *hardware* awalnya dikembangkan oleh para insinyur listrik[16].

# BAB III

## ANALISIS DAN PERANCANGAN

## Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Pada bagian ini, akan dibahas mengenai analisis prosedur dan aliran dokumen yang sedang berjalan yang digambarkan dalam bentuk *flowmap*, pengkodean dan analisis sistem non fungsional yang meliputi perangkat keras dan perangkat lunak yang digunakan.

### Analisis Sistem Yang Sedang Berjalan (*Current System*)

Analisis sistem yang berjalan saat ini terdiri yaitu prosedur pembelian tiket kereta api jarak pendek (*local*)*.*

### 3.1.1.1. Analisis Prosedur (*Flowmap*)

Pada analisis prosedur ini, pemesanan atau pembeliannya dilakukan via loket stasiun, adapun *flowmap* prosedur pemesanan atau pembelian tiket kereta api jarak pendek *(local)*, adalah sebagai berikut :



*Gambar 3.1 Flowmap analisi sistem yang sedang berjalan pada pemesanan tiket kereta api (Sumber KAI.co.id)*

Pada prosuder ini, penumpang yang hendak melakukan pembelian dan pemesanan tiket diharuskan mengambil *form* yang nantinya di isi nama, no identitas, serta data perjalanan untuk memastikan data tersebut valid, selanjutnya petugas loket menginputkan data yang sudah ditulis di*form* lalu petugas loket mengelola pendistribusian tiket. Hal selanjutnya penumpang mempersiapkan uang, untuk

melakukan pembayaran. Tiket didapat setelah penumpang telah membayaran ke petugas loket.

**3.1.1.2 Analisis Dokumen yang Digunakan**

Dari hasil analisis yang dilakukan, dokumen yang digunakan diantaranya adalah dokumen data diri. Untuk mempermudah pendokumentasian maka dibutuhkan suatu sistem yang dapat mempermudah dalam proses pencarian datanya. Adapun dokumen yang dimaksud adalah sebagai berikut :

*Tabel 3.1 Dakumen Yang Digunakan*

|  |  |
| --- | --- |
| **Dibuat oleh** | Petugas loket |
| **Dibuat untuk** | Calon penumpang (*costumer*) |
| **Isi** | Tiket perjalanan kereta api |
| **Frekuensi** | Dibuat sesuai jadwal yang tersedia |
| **Tujuan** | Mempermudah pembelian tiket |

**3.1.2 Analisis Sistem yang akan Dibangun**

Analisis kebutuhan yang dimaksud berupa *flowmap* mengenai sistem yang akan dibangun meliputi prosedur pemesanan atau pembelian tiket kereta api. Adapun *Flowmap* yang akan dibangun adalah sebagai berikut :



*Gambar 3. 2 Flowmap pemesanan dan pembelian tiket kereta api local*

Pada prosuder ini penumpang yang telah memiliki aplikasi bisa langsung *login* dengan mengisikan no hp dan *password* yang telah didaftarkan, tetapi jika belum terdaftar maka lakukan lah *sign up* dengan mengisi biodata diri.

Setelah login berhasil penumpang bisa langsung memilih menu perjalanan, pada analisis ini, maka menu perjalanan yang dipilih adalah *local train* (kereta api jarak pendek), selanjutnya isi jadwal perjalanan, dengan pilih stasiun asal, stasiun tujuan, tanggal keberangkatan, dan jumlah penumpang yang akan berangkat, selanjutnya aplikasi akan mencari ketersediaan kereta untuk memastikan dan memilih kereta api, setelah memilih kereta langkah selanjutnya memastikan order confirmasi bahwa data perjalanan, dan nama kereta telah sesuai, maka masukan data penumpang yang akan berangkat dengan mengisi nama, no Id.

Selanjutnya proses pembayaran akan dilakukan, dengan menvalidasi kode pembayaran yang tertera dalam aplikasi tersebut. Setelah pembayaran sukses maka kode *booking* akan muncul dan tiket telah dimiliki.

**3.1.2.1 Kebutuhan Fungsional (*Fuctioanal Requirements*)**

Analisis kebutuhan fungsional merupakan suatu kebutuhan yang berhubungan dengan kebutuhan sistem yang akan dibuat. Dimana menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sistem, adapun kebutuhan fungsional yang akan di buat yaitu pengelolaan data diri terdiri dari 5 (lima) proses sesuai dengan urutan sebagai berikut :

1. Kelola pemesananan

2. Kelola menu perjalanan

3. Kelola report

4. Kelola penjualan

Setiap proses memiliki repesentasi masing-masing pada sebuah tabel atau data yang terdapat pada *database* yang telah dirancang sebelumnya. Setiap proses berhubungan langsung dengan entitas atau user.

**3.1.2.2 Kebutuhan Non-Fungsional (*Non- Functional Requirement*)**

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras/*hardware*, analisis perangkat lunak/*software*, analisis pengguna/*User*. Adapun kebutuhan fungsional yang akan dibuat adalah sebagai berikut :

1. Kebutuhan perangkat keras

*Tabel 3.2 Deskripsi Perangkat keras Server*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Perangkat | Spesifikasi | Keterangan |
| 1 | Hardisk | 500 GB | Media untuk data aplikasi yang dibuat |
| 2 | Memory | 2 GB | Memori sistem yang digunakan |
| 3 | Processor | Intel (R) Celeron(R) CPU N3060 @ 1.60Hz 1.60GHz | Untuk kecepatan tranfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer |
| 4 | Infrastruktur Jaringan | *Internet* | Bisa dianalogikan sebagai alur proses dari titik awal proses sampai pada akhir proses |

*Tabel 3.3 Perangkat Keras client*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Perangkat | Spesifikasi | Keterangan |
| 1 | *Mobile Phone* | *Android* Kitkat dan Selebihnya | Sebagai media untuk menjalankan aplikasi |

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | Memory | 2 GB | Kecepatan client dalam mengakses sistem ini |
| 3 | InfrastrukturJaringan | *Internet* | Server dan host |

1. Kebutuhan Perangkat Lunak

*Tabel 3.4 Deskripsi Perangkat Lunak Server*

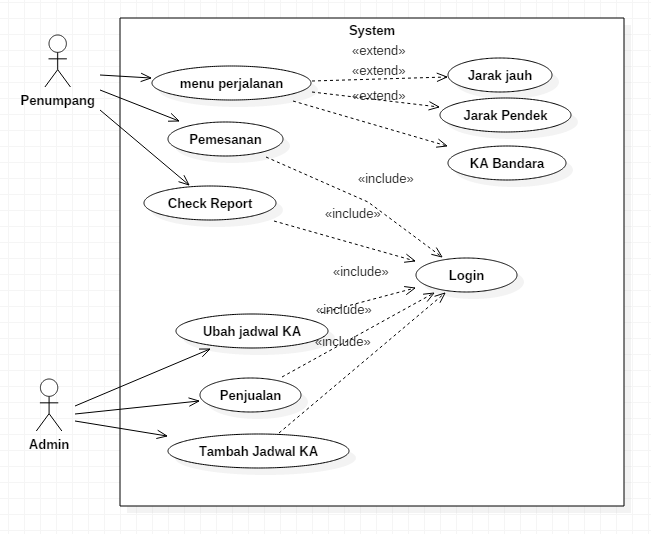
|  |  |  |  |
| --- | --- | --- | --- |
| No | Tools/Software | Fungsi | Keterangan |
| 1 | Windows10 | Sistem Operasi | - |
| 2 | PHP versi 7.3.8 | Bahasa pemograman yang digunakan | - |
| 3 | MySql 8.0.17 | Penyimpanan data di *database* | - |
| 4 | *Apache* 2.4.39 | Web server yang digunakan | - |

*Tabel 3.5 Deskripsi Perangkat Lunak client*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Tools/*Software* | Fungsi | Keterangan |
| 1 | Android | Sistem Operasi | - |

* 1. **Perancangan UML**
     1. ***Usecase* Diagram**

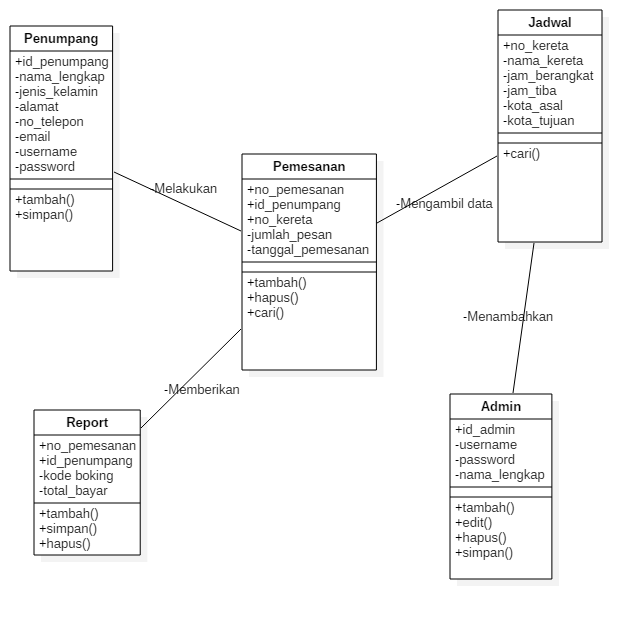
Model *usecase* diagram ini merupakan gambaran aktivitas yang berjalan dilihat dari kebutuhan. Pada tahapan ini penumpang selaku aktor bisa melakukan tiga langkah di sistem ini, yaitu memilih menu perjalanan, yang diextendkan menjadi tiga kategori yaitu perjalanan jarak jauh, jarak pendek, dan kereta khusus bandara, selain itu juga penumpang dapat melakukan pemesanan, dan *check report* melalui sitem ini. Selanjutnya admin dapat mengubah jadwal KA, melakukan penjualan, dan menambahkan jadwal KA, yang di *include* kan ke dalam *usecase login.*



*Gambar 3.3 Usecase Diagram*

**3.2.2 *Class* Diagram**

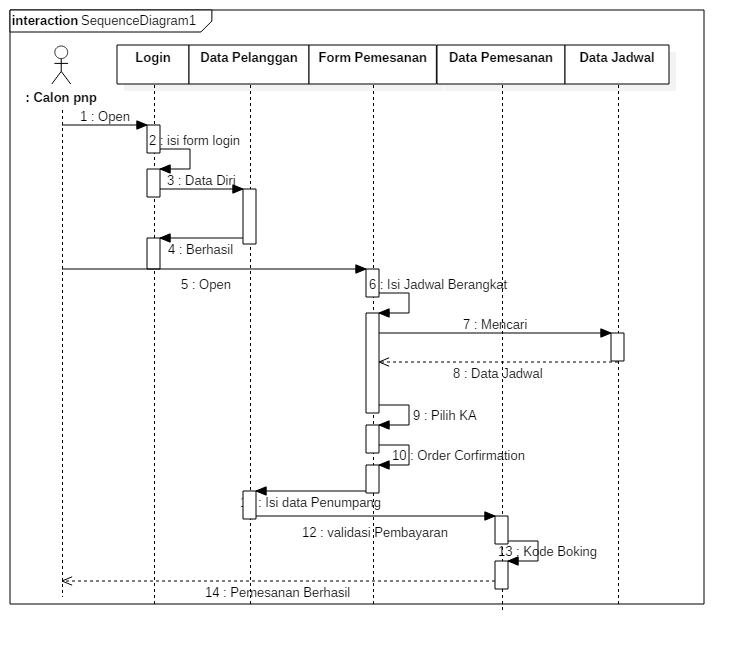
*Class* Diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Pada tahap ini ada 5 object yang digambarkan, disetiap object mempunyai atribut dan *method* yang digunakan. Contohnya dalam gambar dibawah ini.



*Gambar 3.4 Class Diagram*

**3.2.3 *Sequence* Diagram**

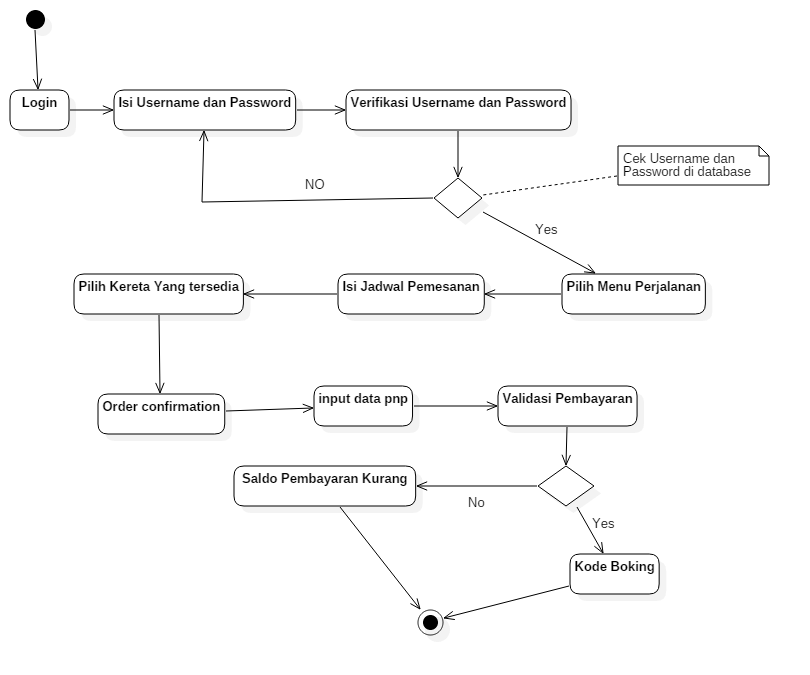
Pada *Sequance* diagram melakukan suatu interaksi objek yang disusun dalam suatu urutan waktu di sekitar sistem. *Sequence* diagram terdiri dari dimensi waktu dan dimensi dari objek-objek yang terkait didalam sistem ini memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam use case.

****

*Gambar 3.5 Sequence Diagram*

**3.2.4 *Activity* Diagram**

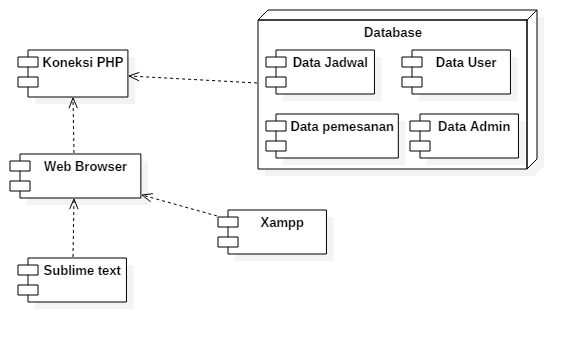
*Activity* Diagram atau Diagram aktivitas yaitu penggambaran rangkaian aliran aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

****

*Gambar 3.6 Acivity Diagram*

**3.2.5 *Component* Diagram**

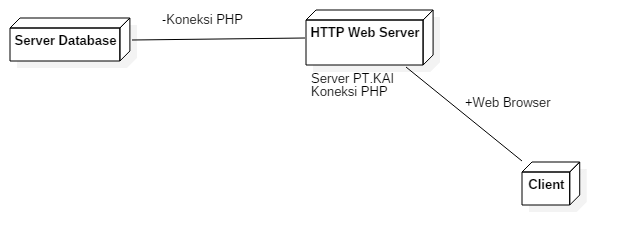
*Component* diagram terdiri dari satu atau beberapa diagram komponen dan menggambarkan pengorganisasian dan keterkaitan dari semuanya ke dalam komponen desain fisik.



*Gambar 3.7 Component diagram*

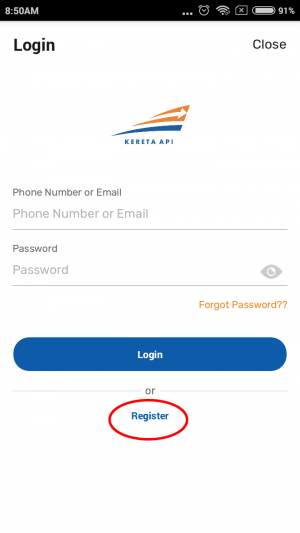
**3.2.6 *Deployment* Diagram**

*Deployment/physical* diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal.

****

*Gambar 3.8 Deployment Diagram*

**3.2.7 Perancangan Antarmuka**

**3.2.7.1 Tampilan Login**



*Gambar 3.9 Tampilan login*

Pada tampilan ini terdapat button login yang digunakan untuk mengakses aplikasi, digunakannya ketika telah mengisi *phone number* atau *e-mail* dan *password* yang sudah terdaftar, jika penumpang yang ingin melakukan pemesanan tetapi belum mempunyai akun pada aplikasi ini, penumpang dapat melakukan *register* dengan mengisi data diri yang tertera pada aplikasi ini.

**3.2.7.2 *Flowchart* Algoritma Login**

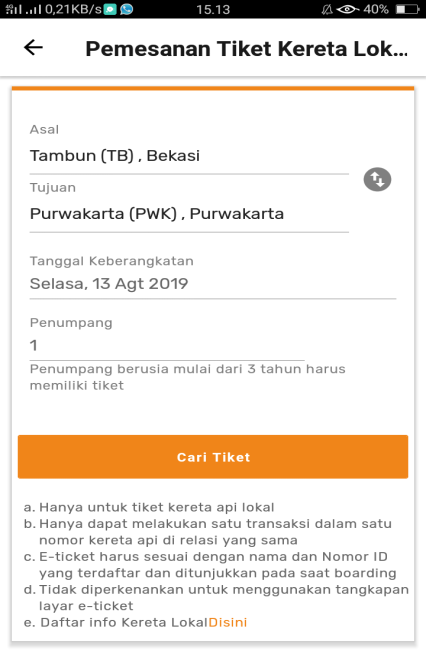
Adapun *flowchart* algoritma yang diterapkan pada prosedur login yaitu :



*Gambar 3.10 Flowchart Algoritma Login*

*Buka* aplikasi KAI *access* selanjutnya akan melakukan tahapan login dengan memasukan atau mengisi no hp atau *email* yang sudah *teregister* pada aplikasi ini, jika belum maka lakukanlah *registrasi* dengan mengisi data diri, setelah *teregistrasi* maka akan diminta untuk memasukan no hp atau *email* dan *password*, setelah itu proses *login* berhasil akan tampil menu aplikasi pada perangkat *client* tersebut.

**3.2.7.3 Tampilan *Form* Pemesanan**

****

*Gambar 3.11 Tampilan Form Pemesanan*

Tampilan *form* pemesanan ini merupakan bagian dari tahap pertama melakukan pemesanan, pemesanan dilakukan dengan mengisi nama asal stasiun keberangkatan dan nama stasiun tujuan, dengan menggunakan *list* daftar stasiun yang tertera pada aplikasi tersebut, setelah itu masukan tanggal keberangkatan dan jumlah penumpang yang akan berangkat, selanjutkan *link button* akan mencari ketersediaan kereta berdasarkan pada stasiun asal keberangkatan.

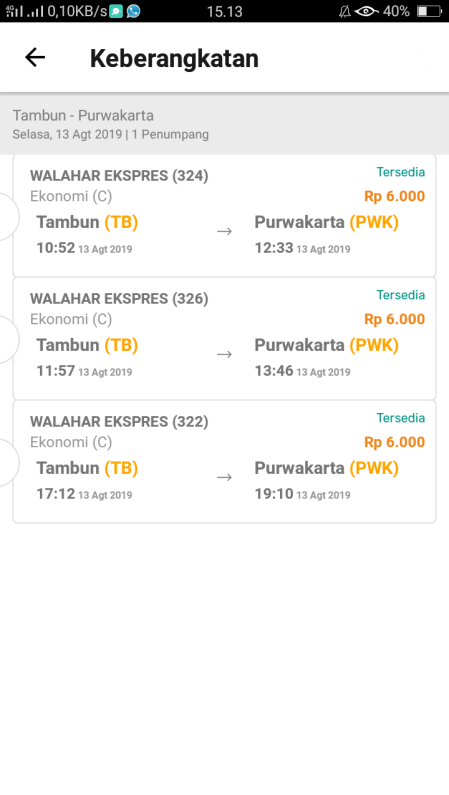
**3.2.7.4 *Flowchart* Algoritma *Form* Pemesanan**



*Gambar 3.12 Flowchart Algoritma Form Pemesanan*

Tahap selanjutnya adalah memilih menu perjalanan pada aplikasi tersebut, untuk tahapan ini pilihlah *local train* sebagai tahapan *flowchart* algoritma *form* pemesanan, pilih stasiun asal, stasiun tujuan, serta tanggal keberangkatan, dan jumlah penumpang yang akan melakukan perjalanan, maka *button* cari pada aplikasi akan memproses, ke tahap selanjutnya.

**3.2.7.5 Tampilan Keberangkatan**

****

*Gambar 3.13 Tampilan keberangkatan*

Pada tampilan ini menampilkan data-data kereta apasaja yang tesedia distasiun asal keberangkatan hingga tiba distasiun tujuan atau stasiun yang disinggahin oleh kereta tersebut.

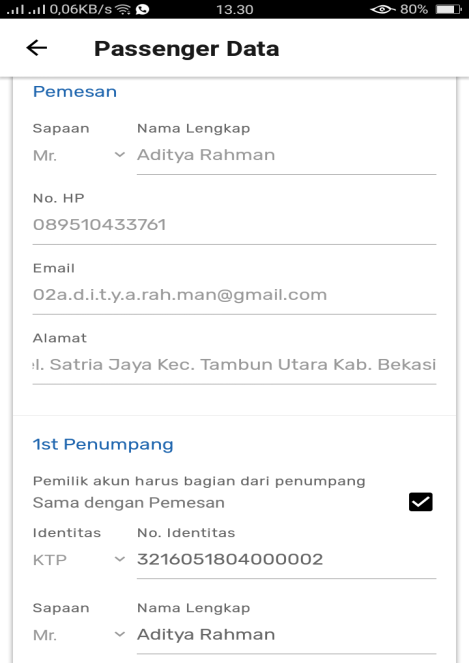
**3.2.7.6 *Flowchart* Algoritma Keberangkatan**



*Gambar 3.14 Flowchart algoritma keberamgkatan*

Melanjutkan dari tahap pemesanan, lalu memilih ketersediaan kereta yang ditampilkan pada tampilan keberangkatan, jika kereta tersedia maka pilihlah 1 jadwal keberangkatan kereta yang nantinya akan di proses pada tahap selanjutnya, jika kereta tidak tesedia maka kembali pada proses *form* pemesanan.

**3.2.7.7 Tampilan Data Penumpang**

****

*Gambar 3.15 Tampilan Data Penumpang*

Tampilan ini merupakan tahapan penting dalam melakukan pemesanan, data pemesan merupakan penanggung jawab untuk pemasanan tiket tersebut, di isi sesaui data *register* pada tahap *login,* serta menginputkan data penumpang sesuai dengan nama, dan no id yang melakukan perjalanan.

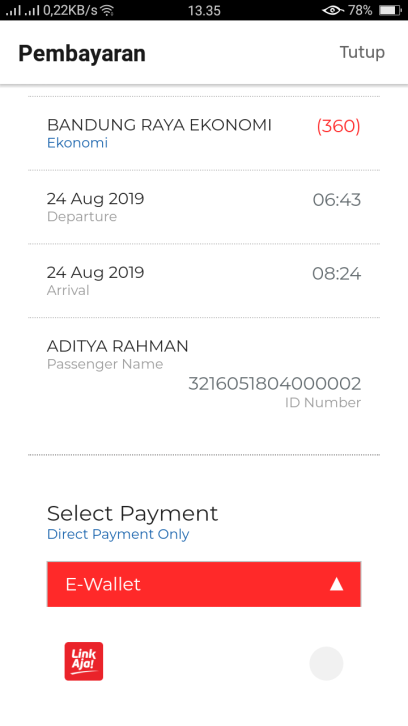
**3.2.7.8 *Flowchart* Algoritma Data Penumpang**



*Gambar 3.16 Flowchart Algoritma Data Penumpang*

Setelah mengisi tahap keberangkatan maka, selajutnya isi nama, no hp, email, dan alamat sebagai kontak penanggung jawab pemesan, diisi berdasarkan data yang teregister pada tahap login tadi, lalu isi nama dan no id penumpang yang akan melakukan perjalanan, sebagai contoh dalam gambar 3.16 tersebut.

**3.2.7.9 Tampilan Pembayaran**

****

*Gambar 3.17 Tampilan Pembayaran*

Gambar diatas merupakan tahapan dalam melakukan pembayaran, pada aplikasi ini khususnya dalam perjalanan kereta api jarak pendek hanya bisa dilakukan dengan metode pembayaran *via e-wallet*, serta dalam aplikasi ini juga hanya tersedia *e-wallet* dengan nama LinkAja.

**3.2.7.10 *Flowchart* Algoritma Pembayaran**



*Gambar 3.18 Flowchart Algoritma Pembayaran*

Prosesnya yaitu, setelah menginputkan data pemesan dan data penumpang dengan benar, maka pembayaran dilakukan dengan metode *e-wallet*, pastikan data-data telah sesuai.

**3.2.7.11 Tampilan E-Ticket**

****

*Gambar 3.19 Tampilan E-Ticket*

Setelah pembayaran dilakukan, maka dalam kurun waktu 60 menit tiket sudah dimiliki, dan data yang tertera benar atau valid untuk melakukan perjalanan.

**3.2.7.12 *Flowchart* Algoritma *E-Ticket***



*Gambar 3.20 Flowchart Algoritma E-Ticket*

Tahap ini merupakan langkah terakhir dalam proses pemesanan atau pembelian tiket kereta api jarak pendek, setelah tahap pembayaran dilakukan dalam kurun waktu 60 menit, maka tiket akan terbit, beserta kode *booking*, data penumpang, dan data kereta api keberangkatan, jika dalam waktu 60 menit tiket tidak terbit, maka saldo pada e-wallet kurang dari jumlah pembayaran yang ada.

# BAB IV

# KESIMPULAN DAN SARAN

**4.1 Kesimpulan**

Berdasarkan analisis dan perancanngan sistem informasi yang dibuat untuk pengelolaan pemesanan tiket kereta api local melalui aplikasi kai *access* maka dapat ditarik kesimpulan sebagai berikut:

Dengan adanya KAI *Access* sistem pembelian tidak lagi dilakukan secara manual dan harus mengantri lama di loket, untuk mendapatkannya calon penumpang dapat memesan tiket tersebut melalui aplikasi dimanapun dan kapanpun sesuai keinginan, PT. KAI juga dapat mengurangi kelemahan tersebut dengan menerapkan sistem informasi pemesanan tiket secara online yang sudah dapat dipesan 7 hari sebelum keberangkatan.

**4.2 Saran**

Setelah melakukan analisis pada proyek kami yang berjudul “Analisis Pemesanan Tiket Kereta Api *Local* melalui aplikasi KAI *ACCESS*” maka saran yang diberikan dalam pengembangan aplikasi ini agar dapat lebih baik lagi kedepannya, yaitu:

1. Pada pemesanan tiket kereta api *local* disini data pemesan akan. langsung secara otomatis dibuat oleh sistem sehingga pemesan tidak perlu memasukan data diri lagi, tetapi ini hanya berlaku bagi pengguna yang telah melakukan login terlebih dahulu pada aplikasi kai *access* dan itu hanya berlaku untuk satu orang saja. Lain halnya lagi jika melakukan pemesanan yang lebih dari satu orang maka calon penumpang yang lain kecuali yang telah melakukan login maka harus melakukukan pengisian data lagi.

2. Pada aplikasi ini seharusnya pada saat pertama kali pengguna membuka aplikasi KAI *ACCESS* maka tampilan pertama yang akan mucul adalah pengguna disuruh untuk melakukan login atau sigin terlebih dahulu.

# DAFTAR PUSTAKA

[1] AM, Wardhani (2016). Fungsional Perkeretaapian. Ambarawa: Gava media.

[2] Hartanto,dkk. (2018). PERANCANGAN APLIKASI E-COMMERCE. Jurnal Ilmu Komputer Dan Sistem Informasi, 6(2).

[3] Lisnawanty. (2014). Perancangan Sistem Informasi Kearsipan Surat Masuk dan Surat Keluar Berbasis Multiuser. Jurnal Khatulistiwa Informatika. 2(2), 162-163.

[4] Dharwiyanti, Sri. (2003). Pengantar Unified Modeling Language (UML). Jurnal Ilmu Komputer.

[5] Charzon. (2018). Pengertian dan fungsi database untuk program DELPHI. Teknologi informasi dan bisnis pengabdian masyarakat Darmajaya, 1(1), 77-80

[6] Harihanto,dkk. (2013). Study Tentang Pengguna Internet Oleh Pelajar. eJournal Sosiatri-Sosiologi.1(4)

[7] Firman, Astria, dkk. (2016). Sistem Informasi Perpustakaan Online Berbasis Web. E-journal Teknik Elektro dan Komputer. 5(2).

[8] A.Mulyana. (2018). Perancangan E-Payment System pada E-Wallet Menggunakan Kode QR Berbasis Android. Jurnal Sistem Komputer.7(2).

[9] Mujiati,Hanik. (2014). Analisis Dan Perancangan Sistem Informasi Stok Obat Pada Apotek Arjawinangun. Speed Journal – Sentra Penelitian Engineering dan Edukasi.11(2).

[10] Palit,V.Randi. (2015). Rancangan Sistem Informasi Keuangan Berbasis Web Di CV Ratmat Sentosa Prupuk. E-Journal Sistem informasi Komputer. 4(7).

[11] Any. (2013). Sistem pemesanan dan pembelian tiket di <http://any.web.id/sistem-pemesanan-dan-pembelian-tiket>-1099 (di akses 23 agustus 2019).

[12] <http://jurnalapps.co.id/Kai-access-2474> (di akses 23 agustus 2019).

[13] Pandu, Mulya. (2013). Tinjauan pustaka tentang unified modeling language. [Internet]. Tersedia di: <http://e-jurnal.unikom.ac.id/39273/4/UML088.pdf>.

[14] Septiani, Hanin. (2016). Tinjauan pustaka tentang struktur diagram pada UML. [Internet]. Tersedia di: <http://e-jurnal.unikom.ac.id/23542/2/35454.pdf>.

[15] Kurniawan, Gilang. (2012). Tinjauan pustaka tentang kinerja usecase. [Internet]. Tersedia di: <http://e-jurnal.unpas.ac.id/01829/12/kinerjausecase.pdf>.

[16] Bagus, dkk. (2012). Pengantar UML. Yogyakarta. Galang Media Utama.