RANCANG BANGUN APLIKASI VERIFIKASI PEMESANAN TIKET DENGAN QR-CODE BERBASIS ANDROID MENGGUNAKAN ALGORITMA KRIPTOGRAFI ASIMETRIS RSA

SKRIPSI

Oleh

Febrianto Rama Anji

NIM 112410101047

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015

RANCANG BANGUN APLIKASI VERIFIKASI PEMESANAN TIKET DENGAN QR-CODE BERBASIS ANDROID MENGGUNAKAN ALGORITMA KRIPTOGRAFI ASIMETRIS RSA

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Sistem Informasi (S1)
dan mencapai gelar Sarjana Komputer

Oleh

Febrianto Rama Anji

NIM 112410101047

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Ibunda Sujinah dan Ayahanda Anton Kasiono yang tercinta;
2. Adekku tercinta Boma Indra Saputra;
3. Guru-guruku sejak sekolah dasar sampai dengan perguruan tinggi;
4. Teman-teman Nefotion;
5. Teman-teman terdekat Imaduddin, Sufianto, Jaya, Choiriawan, Widyastiti, Nur, Sri, Adindayu, Trisna, Nathanael, Mirza, Indra, Rawinka;
6. Teman-teman yang turut membantu dan memberi semangat Susanti, Kristinawati, Widya, Firdanasari
7. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

"Allah tidak akan menimpakan beban kepada hamba-Nya di luar kemampuannya".
(QS. Al - Baqarah: 286)

"Sesungguhnya sesudah kesulitan itu ada kemudahan".
(QS. Al - Insyirah: 6)

"Allah tidak hendak menyulitkan kamu, tetapi Dia hendak membersihkan kamu dan menyempurnakan nikmat-Nya bagimu, supaya kamu bersyukur".
(QS. Al Maa'idah: 6)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Febrianto Rama Anji

NIM : 112410101047

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Rancang Bangun Aplikasi Verifikasi Pemesanan Tiket dengan *QR-Code* Berbasis Android Menggunakan Algoritma Kriptografi Asimetris RSA", adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 24 Agustus 2015

Yang menyatakan,

Febrianto Rama Anji

NIM 112410101047

PENGESAHAN PEMBIMBING

Skripsi berjudul "Rancang Bangun Aplikasi Verifikasi Pemesanan Tiket dengan *QR-Code* Berbasis Android Menggunakan Algoritma Kriptografi Asimetris RSA", telah diuji dan disahkan pada:

Hari, tanggal : Senin, 24 Agustus 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui Oleh :

Pembimbing 1,	Pembimbing 2,
Dr. Saiful Bukhori, ST., M.Kom	Yanuar Nurdiansyah, ST., M.Cs
NIP 19681113 199412 1 001	NIP 19820101 201012 1 004

SKRIPSI

RANCANG BANGUN APLIKASI VERIFIKASI PEMESANAN TIKET DENGAN QR-CODE BERBASIS ANDROID MENGGUNAKAN ALGORITMA KRIPTOGRAFI ASIMETRIS RSA

Oleh
Febrianto Rama Anji
NIM 112410101047

Pembimbing:

Pembimbing Utama : Dr. Saiful Bukhori, ST., M.Kom
NIP 19681113 199412 1 001

Pembimbing Pendamping : Yanuar Nurdiansyah, ST., M.Cs
NIP 19820101 201012 1 004

PENGESAHAN

Skripsi berjudul "Rancang Bangun Aplikasi Verifikasi Pemesanan Tiket dengan *QR-Code* Berbasis Android Menggunakan Algoritma Kriptografi Asimetris RSA", telah diuji dan disahkan pada:

Hari, tanggal : Senin, 24 Agustus 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji

Penguji 1,	Penguji 2,
Anang Andrianto, ST., MT	M. Arief Hidayat, S.Kom., M.Kom
NIP 19690615 199702 1 002	NIP 19810123 201012 1 003

Mengesahkan

Ketua Program Studi,

Prof. Slamin, M.CompSc.,Ph.D

NIP 19670420 199201 1 001

RINGKASAN

Rancang Bangun Aplikasi Verifikasi Pemesanan Tiket dengan *QR-Code* Berbasis Android Menggunakan Algoritma Kriptografi Asimetris RSA; Febrianto Rama Anji, 112410101047; 2015: 140; Program Studi Sistem Informasi Universitas Jember.

PT. Kereta Api Indonesia (PT. KAI) merupakan instansi pemerintah yang bergerak dibidang jasa transportasi umum. Tiket kereta api awalnya hanya dapat dibeli dengan cara konvensional (datang ke stasiun) yang mengakibatkan calon penumpang mengalami kerugian dalam segi waktu dan biaya. Kerugian tersebut terjadi akibat calon penumpang biasanya menghabiskan waktu yang relatif lama untuk mengantri di loket. Kondisi yang demikian membuat PT. KAI melakukan inovasi dalam pelayanan bisnisnya, yaitu dengan menerapkan pemesanan tiket secara *online* yang diharapkan dapat mempermudah calon penumpang dalam melakukan kegiatan pemesanan tiket. Penggunaan kertas tiket khusus berhologram juga memerlukan biaya lebih dalam pembuatannya sedangkan tiket kereta hanya digunakan sekali saja. Tiket kereta api juga dilengkapi dengan *Quick Response Code* (*QR-Code*) yang diharapkan dapat mempermudah dalam melakukan proses verifikasi tiket. Pada penelitian ini dibangun Aplikasi Verifikasi Pemesanan Tiket untuk pengamanan data tiket agar mengurangi resiko penggandaan tiket. Metode yang digunakan adalah Algoritma Kriptografi Asimetris dimana algoritma kriptografi asimetris dapat digunakan pada proses penyisipan data pada *QR-Code* dalam pembuatan tiket yang tidak menggunakan hologram dan tidak menggunakan kertas khusus. Hal ini dikarenakan algoritma ini memiliki dua *key* berbeda dalam proses enkripsi dan dekripsinya sehingga memiliki keamanan lebih jika digunakan pada proses verifikasi. Sistem ini dibangun menggunakan bahasa pemrograman *Page Hypertext Pre-Processor* (PHP), framework *Code Ignither* (CI), dan java. Berdasarkan hasil pengujian terhadap sistem, Penggunaan algoritma dapat memberikan keamanan data tingkat tinggi khususnya pada proses verifikasi.

PRAKATA

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Karya Ilmiah Tertulis (Skripsi) berjudul "*Rancang Bangun Aplikasi Verifikasi Pemesanan Tiket dengan QR-Code Berbasis Android Menggunakan Algoritma Kriptografi Asimetris RSA*".

Pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Prof. Slamin, M.CompSc.,Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Dr. Saiful Bukhori, ST., M.Kom., selaku Dosen Pembimbing Utama, Yanuar Nurdiansyah, ST., M.Cs., selaku Dosen Pembimbing Pendamping, yang telah memberikan banyak arahan dan bimbingan dalam penulisan skripsi ini;
3. Anang Andrianto, ST., MT., selaku dosen penguji I, M. Arief Hidayat, S.Kom., M.Kom., selaku dosen penguji II, yang telah memberikan masukkan dalam penulisan skripsi ini;
4. Ibu Sujinah, Bapak Anton Kasiono, adikku Boma Indra Saputra yang telah memberikan dukungan dan doa yang tulus;
5. Teman-teman seperjuangan Program Studi Sistem Informasi angkatan 2011.
6. Kakak dan adek tingkat Program Studi Sistem Informasi angkatan 2010, 2012, 2013 dan 2014 yang telah membantu dan mendukung hingga selesainya naskah skripsi ini;
7. Semua pihak yang telah membantu baik tenaga maupun pikiran dalam pelaksanaan kegiatan penelitian dan penyusunan skripsi ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, Agustus 2015

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1. PENDAHULUAN

Bab pendahuluan akan membahas tentang latar belakang, perumusaan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.

1.1 Latar Belakang

Menurut data dari Badan Pusat Statistik (BPS), pertumbuhan penduduk di Indonesia terus meningkat dari tahun ke tahun. Peningkatan tersebut membuat kebutuhan penduduk terhadap jasa transportasi juga semakin meningkat. Salah satu perusahaan yang bergerak di bidang jasa transportasi, khususnya dalam bidang transportasi darat adalah PT. Kereta Api Indonesia (PT. KAI). Penggunaan jasa kereta api banyak menjadi pilihan masyarakat dikarenakan harga tiket yang terjangkau dan keamanan yang terjamin.

Tiket kereta api awalnya hanya dapat dibeli dengan cara konvensional (datang ke stasiun) yang mengakibatkan calon penumpang mengalami kerugian dalam segi waktu dan biaya. Kerugian tersebut terjadi akibat calon penumpang biasanya menghabiskan waktu yang relatif lama untuk mengantri di loket. Kondisi yang demikian membuat PT. KAI melakukan inovasi dalam pelayanan bisnisnya, yaitu dengan menerapkan pemesanan tiket secara *online* yang diharapkan dapat mempermudah calon penumpang dalam melakukan kegiatan pemesanan tiket (Damardono, 2012).

Berdasarkan hasil observasi yang dilakukan di Stasiun Jember pada April 2014, alur pembelian pembuatan tiket yang sedang diterapkan oleh PT. KAI masih memiliki beberapa kelemahan dalam alur pemesanan tiket dan teknologi pembuatan tiket. Hal ini dapat mengurangi efisiensi waktu calon penumpang dan menambah pengeluaran perusahaan dalam melakukan kegiatan bisnisnya karena proses pembuatan tiket kereta masih panjang. Penggunaan kertas tiket khusus berhologram juga memerlukan biaya lebih dalam pembuatannya sedangkan tiket kereta hanya digunakan sekali saja.

Tiket kereta api juga dilengkapi dengan *Quick Response Code* (*QR-Code*) yang diharapkan dapat mempermudah dalam melakukan proses verifikasi tiket. Namun setelah dilakukan proses pemindaian (*scan*) isi dari *QR-Code* tersebut bukan data calon penumpang yang terlihat tetapi hanya sebagian kecil dari tiket (seperti nomer tiket) saja yang terlihat. Pembuatan tiket juga mengharuskan calon penumpang menuju ke stasiun untuk mencetak tiket karena tiket yang digunakan harus menggunakan kertas khusus berhologram. Hal seperti ini membutuhkan waktu yang lebih lama dan memperpanjang alur pelayanan pada PT. KAI.

Penggunaan *QR-Code* pada tiket seharusnya dapat lebih ditingkatkan kegunaannya, seperti penyisipan kode unik atau data pemesan, nama kereta, nomor gerbong dan nomor kursi. Petugas dapat melakukan proses verifikasi tiket lebih cepat dan lebih mudah dengan adanya data di dalam *QR-Code*. Tiket kereta yang sekarang menggunakan hologram juga dapat diganti dengan tiket yang tidak menggunakan hologram dan tidak menggunakan kertas khusus. Hal ini dapat mengurangi biaya produksi pembuatan tiket yang nantinya dapat memberikan keuntungan lebih untuk PT. KAI. Penyisipan data pada *QR-Code* dan tidak digunakannya kertas khusus dalam pembuatan tiket memerlukan adanya keamanan dalam proses pembuatan dan verifikasi tiket. Hal tersebut diperlukan agar tidak terjadi penggandaan tiket atau pembuatan tiket palsu yang dapat menyebabkan kerugian pada PT. KAI. Algoritma kriptografi dapat digunakan dalam pengamanan data pada *QR*-Code.

Terdapat dua jenis algoritma kriptografi, yaitu algoritma kriptografi simetris dan algoritma kriptografi asimetris. Perkembangan kedua jenis kriptografi ini membutuhkan *cipher* dan kunci (*key*). Kristanto dan Munir dalam (Hartini, 2014) menyatakan bahwa algoritma kriptografi disebut sebagai *cipher*, yaitu aturan untuk melakukan enkripsi dan dekripsi. Pengertian *cipher* dapat juga dinyatakan sebagai fungsi matematika yang digunakan untuk melakukan enkripsi dan dekripsi. Konsep matematis dari algoritma kriptografi adalah hubungan antara dua buah himpunan yang terdiri dari elemen *plainteks* dan elemen *cipherteks*. Enkripsi dan dekripsi merupakan fungsi yang akan memetakan elemen dari kedua himpunan tersebut. Kriptografi modern dapat mengatasi keamanan algoritma

dengan menggunakan *key* yang tidak dirahasiakan tetapi kunci tersebut harus dijaga kerahasiaannya. Munir dalam (Hartini, 2014) menyatakan bahwa *key* adalah parameter yang digunakan untuk melakukan transformasi enkripsi dan dekripsi.

Perbedaan algoritma kriptografi simestris dan asimetris terletak pada penggunaan kunci yang berbeda untuk proses enkripsi dan dekripsi. Algoritma asimetris ini disebut juga algoritma kunci umum (*public key algorithm*) karena salah satu kunci dibuat umum (*public key*) atau dapat diketahui semua orang tetapi kunci lainnya hanya diketahui oleh orang yang berwenang mengetahui data yang disandikan atau sering disebut kunci pribadi (*private key*) (Hartini, 2014). Algoritma kriptografi asimetris dapat digunakan pada proses penyisipan data pada *QR-Code* dalam pembuatan tiket yang tidak menggunakan hologram dan tidak menggunakan kertas khusus. Hal ini dikarenakan algoritma ini memiliki dua *key* berbeda dalam proses enkripsi dan dekripsinya sehingga memiliki keamanan lebih jika digunakan pada proses verifikasi (Nivedita Bisht, 2015).

Berdasarkan uraian di atas maka perlu suatu inovasi pada proses pembuatan tiket dan verifikasi tiket. Inovasi yang bisa dilakukan yaitu dengan cara membuat tiket dengan tidak menggunakan kertas khusus dilengkapi *QR-Code* berisi data pemesan dan data yang berhubungan dengan informasi pemberangkatan pemesan. *QR-Code* ini pada proses pembuatannya dienkripsi oleh sistem pemesanan tiket *online*. *QR-Code* ini dapat secara mudah dipindai oleh *scanner* manapun akan tetapi pendekripsian kode hanya bisa dilakukan oleh petugas sebagai pemegang aplikasi verifikasi tiket.

Tiket yang didalamnya tertera *QR-Code* yang sudah dienkripsi oleh perusahaan dapat diunduh oleh pemesan secara *online*. Tiket tersebut hanya bisa diverifikasi oleh petugas menggunakan aplikasi khusus pada android miliknya. Penggunaan cara ini bertujuan agar tiket tidak mudah dipalsukan dan proses verifikasi tiket akan lebih efisien. Harga pembuatan tiket akan jauh lebih murah tanpa penggunaan hologram sehingga diharapkan keuntungan PT. KAI meningkat. Calon penumpang juga tidak perlu antri lama pada proses verifikasi tiket.

1.2 Rumusan Masalah

Berdasarkan uraian di atas maka dirumuskan permasalahan sebagai berikut:

1. Bagaimana cara mengenkripsi dan mendekripsi *QR-Code* menggunakan algoritma kriptografi asimetris?
2. Bagaimana cara merancang dan membangun aplikasi verifikasi pemesanan tiket dengan *QR-Code* berbasis android menggunakan algoritma kriptografi asimetris?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan.

1.3.1 Tujuan

Tujuan dari penelitian adalah:

1. Merancang cara mengenkripsi dan mendekripsi *QR-Code* menggunakan algoritma kriptografi asimetris.
2. Merancang dan membangun aplikasi verifikasi pemesanan tiket dengan *QR-Code* berbasis android menggunakan algoritma kriptografi asimetris.

1.3.2 Manfaat

Manfaat yang ingin didapatkan dari penelitian ini adalah :

1. Manfaat Akademis

Manfaat Akademis pada penelitian ini diharapkan dapat memberikan masukan dan kontribusi terkait informasi yang berhubungan dengan judul pada penelitian ini.

2. Manfaat bagi peneliti
    a. Mengetahui bagaimana proses penerapan algoritma Kriptografi Asimetris RSA pada proses verifikasi pemesanan tiket dengan menggunakan QR-Code.

b. Membantu instansi dalam pemesanan tiket untuk menyelesaikan permasalahan verifikasi pemesanan tiket dengn menggunakan QR-Code.

3. Manfaat bagi objek penelitian
   a. Memberikan inovasi baru kepada instansi tempat penelitian mengenai penggunaan algoritma Kriptografi Asimetris RSA untuk membangun aplikasi verifikasi pemesanan tiket dengan menggunakan QR-Code.
   b. Membantu instansi untuk melakukan pemesanan tiket dengan lebih optimal dan efektif.

1.4 Ruang Lingkup

Ruang lingkup adalah batasan untuk objek dan tema yang akan diteliti, yang diharapkan agar tidak terjadi penyimpangan ketika proses penulisan dan pembuatan sistem. Batasan masalah dapat dilihat seperti di bawah ini.

1. Aplikasi yang akan dibuat digunakan untuk mengenkripsi dan mendekripsi *QR-Code*.
2. Terdiri dari dua sistem, yaitu web untuk pemesanan tiket dan android untuk verifikasi tiket.
3. Sistem verifikasi tiket hanya digunakan pada peron di stasiun.
4. Enkripsi *QR-Code* menggunakan algoritma kriptografi asimetris.
5. Data yang dimasukkan sesuai dengan data yang diberikan oleh PT. KAI.

1.5 Sistematika Penulisan

Sistematika penulisan dan kerunutan skripsi ini disusun sebagai berikut:

1. Pendahuluan

   Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.

2. Tinjauan Pustaka

   Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.

3. Metodologi Penelitian

   Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.

4. Pengembangan Sistem

   Bab ini menjelaskan tentang pengembangan sistem yang dikembangkan.

5. Hasil dan Pembahasan

   Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.

6. Penutup

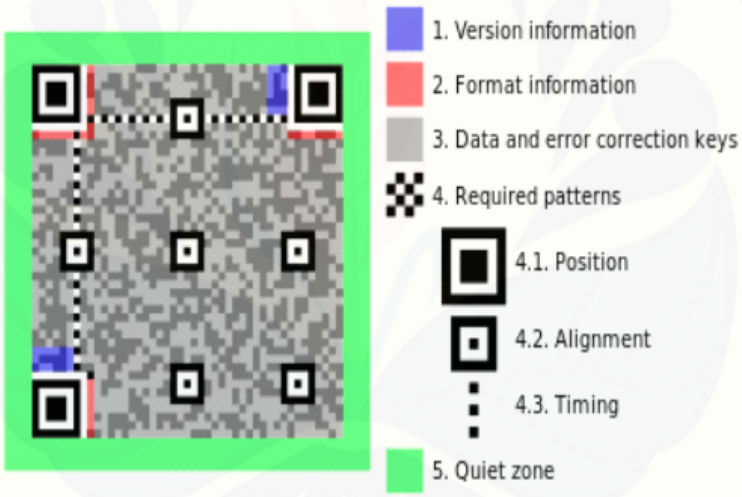
   Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Tinjauan pustaka adalah bagian yang akan menjelaskan teori-teori dan pustaka yang akan digunakan dalam penelitian, yakni tentang *QR-Code*, android, algoritma, kriptografi, enkripsi asimetris, dan enkripsi asimetris RSA.

2.1 QR-Code

QR-Code adalah *Barcode* dua dimensi yang diperkenalkan oleh perusahaan asal jepang bernama Denso-Wafe pada tahun 1994. Jenis *barcode* ini awalnya digunakan sebagai penanda inventaris pada industri *sparepart* kendaraan. *QR* yang berarti "*Quick Response*" karena sandinya dibaca dengan kecepatan tinggi. Struktur dari *QR-Code* terdiri dari 5 bagian yang dijelaskan pada Gambar 2.1.



Gambar 2.1 Struktur *QR-Code*

Sumber: (Abhishek Gandhi, 2014)

2.1.1 Teknologi *QR-Code*

Sebuah *QR-Code* adalah kode *matrix* yang dibuat dan diterbitkan ke dalam sebuah simbol yang mudah dibaca oleh peralatan pemindai (*scanner*). *QR-Code* mengandung informasi baik dari arah vertikal maupun horizontal, sedangkan pada *barcode* hanya memiliki satu arah data (biasanya hanya arah vertikal). *QR-Code*

juga dapat diisi dengan jumlah data yang lebih besar, yaitu 7089 karakter *numeric*, 4296 karakter *alphanumeric*, 2953 *byte* bilangan biner dan 1817 karakter kanji atau bahasa jepang.

2.1.2 Kegunaan *QR-Code*

Tanpa sebuah alat, tidak mungkin seseorang bisa membaca isi *QR-Code* secara manual tapi *QR-Code* akan lebih mudah dibaca dengan peralatan pemindai. Sekarang aplikasi pemindai *QR-Code* telah tersedia gratis di bebagai *app stores*. Pengguna dapat memindai *QR-Code* dan aplikasi terintegrasi dengan perangkat mereka untuk membaca isi *QR-Code* dan menampilkan informasi yang ada di perangkat mereka. Tergantung pada jenis data yang disembunyikan dalam *QR-Code* dan sifat aplikasi, tindakan alternatif dapat diambil pada tahap pembacaan sandi: nomor telpon dapat secara otomatis keluar, SMS dapat dikirim, halaman web berdasarkan *URL* dapat ditampilkan pada perangkat, atau dapat menjalankan aplikasi tertentu (Abhishek Gandhi, 2014).

2.2 Android

Android adalah sistem operasi berbasis linux yang dirancang khusus untuk perangkat layar sentuh seperti *smart phones* dan tablet. Android awalnya dibuat dan diluncurkan oleh Android Incorporation yang didukung oleh Google Financial dan kemudian dibeli pada 2005. Awalnya Android bekerja pada kernel Linux versi 2.6, dan dari Android versi 4.0 (*Ice Cream Sandwich*) dan seterusnya ia bekerja pada kernel versi 3.0 dengan *libraries* dan API. Android menggunakan *compiler* mesin virtual bernama Dalvik untuk menjalankan Dalvik *dex-code* (Dalvik *Executable*) yang biasanya diterjemahkan dari kode java.

Aplikasi peminda *QR-Code* ini dikembangkan menggunakan bahasa Java menggunakan *Android Software Development Kit* (SDK). Di dalam SDK terdapat alat pengembangan, termasuk *debugger*, *libraries*, aplikasi untuk emulator, dokumentasi, contoh kode, dan tutorial. IDE (*Integrated Development Environment*) yang didukung adalah Eclipse yang menggunakan *Plug in Android Development Tools* (Abhishek Gandhi, 2014).

2.3 Algoritma

Algoritma adalah sebuah himpunan terhingga dari instruksi yang mempunyai karakteristik berikut ini:

1. Presisi (*precision*), langkah-langkahnya dinyatakan dengan jelas.
2. Unik (*uniqueness*), hasil lanjutan dari setiap langkah dari pelaksanaan didefinisikan secara tunggal dan semata-mata bergantung pada masukan dan hasil dari langkah sebelumnya.
3. Terhingga (*finiteness*), yaitu algoritma berhenti setelah beberapa instruksi terhingga dilaksanakan.
4. Masukan *(input*), yaitu algoritma memerlukan masukan.
5. Keluaran (*output*), yaitu algoritma menghasilkan keluaran.
6. Umum (*generality*), algoritma berlaku pada himpunan masukan.

Algoritma juga diartikan sebagai metode langkah demi langkah dari pemecahan suatu masalah. Langkah-langkah dari suatu algoritma harus dinyatakan dengan jelas sehingga dapat ditulis dalam bahasa pemrograman dan dijalankan oleh komputer (Johnsonbaugh, 1998).

Kompleksitas dari suatu algoritma merupakan ukuran seberapa banyak komputasi yang dibutuhkan algoritma tersebut untuk menyelesaikan masalah. Secara informal, algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki kompleksitas yang rendah, sementara algoritma yang membutuhkan waktu lama untuk menyelesaikan suatu masalah membutuhkan kompleksitas yang tinggi.

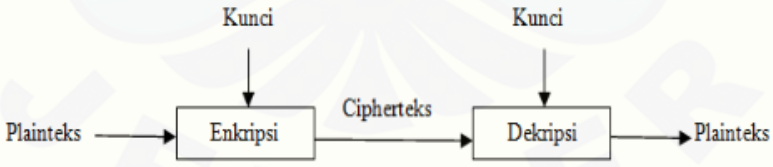
2.4 Kriptografi

Kata kriptografi (*cryptography*) berasal dari bahasa yunani yaitu dari kata *kryptos* yang artinya tersembunyi dan *graphein* yang artinya menulis. Kriptografi dapat diartikan tulisan yang dirahasiakan atau dapat diartikan juga sebagai suatu ilmu ataupun seni yang mempelajari bagaimana sebuah data, informasi dan dokumen dikonversi ke bentuk tertentu yang sulit dimengerti (Hartini, 2014). Kriptografi bertujuan untuk menjaga kerahasiaan data, informasi dan dokumen

supaya tidak dapat diketahui oleh pihak yang tidak berhak mengetahuinya (*unauthorized person*).

Herman dalam (Hartini, 2014) mengatakan terdapat bermacam sistem sandi yang tujuan penggunaan dan tingkat kerahasiaannya berbeda sesuai dengan permintaan *user*, tetapi dalam prakteknya user menginginkan kemudahan seperti: kerahasiaan data, kecepatan, ketepatan dan harga yang murah. Suatu data yang tidak disandikan disebut *plaintext* atau *cleartext* sedangkan data yang telah disandikan disebut *ciphertext*. Proses yang dilakukan untuk mengubah *plaintext* menjadi *chippertext* disebut enkripsi (*encryption*) atau *encipherment* sedangkan proses untuk merubah *ciphertext* kembali ke *plaintext* disebut dekripsi (*decryption*) atau *decipherment* (ISO 7498-2) yang digambarkan pada Gambar 2.2. Kriptografi memerlukan parameter untuk proses konversi yang dikendalikan oleh sebuah kunci (*key*) atau beberapa kunci.

Kriptografi kini telah menjadi salah satu syarat penting dalam keamanan teknologi informasi terutama dalam pengiriman pesan rahasia. Pengiriman pesan rahasia sangat rentan terhadap serangan yang dilakukan oleh pihak ketiga, seperti penyadapan, pemutusan komunikasi, pengubahan pesan yang dikirim, dan lain-lain. Kriptografi dapat meningkatkan keamanan dalam pengiriman pesan atau komunikasi data dengan cara menyandikan pesan tersebut berdasarkan algoritma dan kunci tertentu yang hanya diketahui oleh pihak-pihak yang berhak atas data, informasi, dan dokumen tersebut.



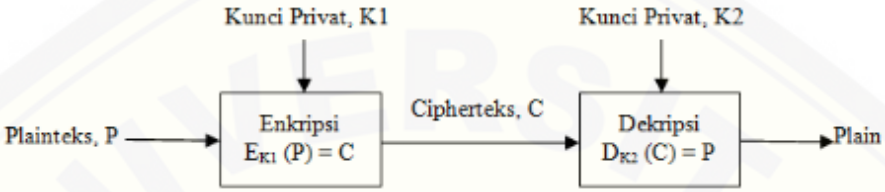
Gambar 2.2 Skema Enkripsi dan Deskripsi

Sumber: Munir dalam (Hartini, 2014)

2.5 Algoritma Kriptografi Asimetris

Algoritma kriptografi asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Algoritma ini disebut juga

algoritma kunci umum (*public key algorithm*) karena kunci untuk enkripsi dibuat umum (*public key*) atau dapat diketahui oleh setiap orang, tapi kunci untuk dekripsi hanya diketahui oleh orang yang berwenang mengetahui data yang disandikan atau sering disebut kunci pribadi (*private key*). Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA dan ECC. Skema dari algoritma kriptografi asimetris dijelaskan pada Gambar 2.3.



Gambar 2.3 Skema Algoritma Asimetris

Sumber: Munir dalam (Hartini, 2014)

2.6 Algoritma Kriptografi Asimetris RSA

Algoritma RSA adalah salah satu teknik kriptografi dimana kunci untuk melakukan enkripsi berbeda dengan kunci untuk melakukan dekripsi. Kunci untuk melakukan enkripsi disebut sebagai kunci publik, sedangkan kunci untuk melakukan dekripsi disebut sebagai kunci privat. Orang yang mempunyai kunci publik dapat melakukan enkripsi tetapi yang dalam melakukan dekripsi hanyalah orang yang memiliki kunci privat. Kunci publik dapat dimiliki oleh sembarang orang, tetapi kunci privat hanya dimiliki oleh orang tertentu saja (Rahajoeningroem & Aria, 2009).

Langkah pertama dalam algoritma RSA adalah membangkitkan pasangan kunci (kunci publik dan kunci privat). Berikut adalah langkah-langkah dalam membangkitkan pasangan kunci :

1. Memilih dua buah bilangan prima sembarang yang besar, p dan q. Nilai p dan q harus dirahasiakan.
2. Menghitung n = p x q. Besaran n tidak perlu dirahasiakan.
3. Menghitung m = (p – 1)(q – 1).

4. Memilih sebuah bilangan bulat sebagai kunci publik, disebut e, yang relatif prima terhadap m. e relatif prima terhadap m artinya faktor pembagi terbesar keduanya adalah 1, secara matematis disebut gcd (e,m) = 1.
5. Menghitung kunci privat, disebut d sedemikian agar (d x e) mod m =1.

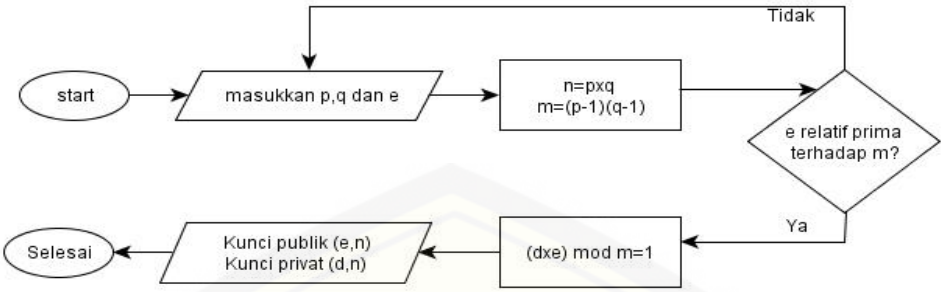
Hasil dari algoritma tersebut diperoleh :

1. Kunci publik adalah pasangan (e,n).
2. Kunci privat adalah pasangan (e,m) n tidak bersifat rahasia, namun ia diperlukan pada perhitungan enkripsi/dekripsi.

Proses setelah pembangkitan pasangan kunci adalah proses enkripsi. Langkah-langkah algoritma enkripsi menggunakan RSA adalah sebagai berikut:

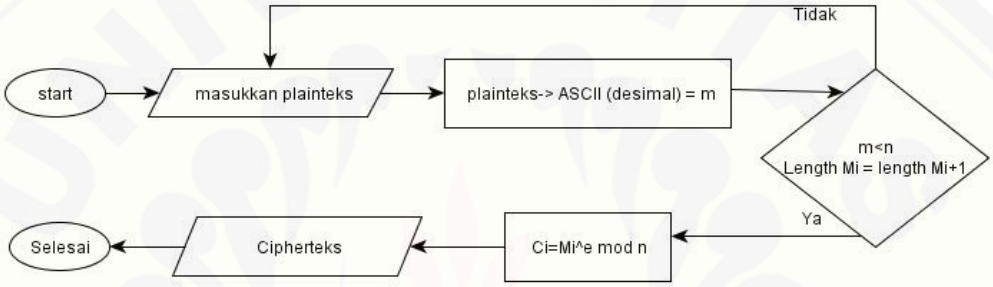
1. Langkah pertama mengambil nilai e dan n dari proses pembangkitan kunci,
2. Masukkan teks yang akan dienkripsi (*plainteks*),
3. Berkas yang akan dienkripsi diubah ke dalam bentuk desimal sesuai dengan tabel ASCII.
4. Membagi berkas tersebut menjadi beberapa blok (m_i), dengan syarat , $m_i < n$ dan *length*(m_i) = *length*(m_{i+1}) ,
5. Setelah itu setiap blok dari berkas tersebut dienkripsikan menggunakan pasangan kunci publik.

Setelah didapatkan *chipertext*, dengan begitu informasi tersebut tidak akan dapat dibaca lagi oleh orang tanpa melalui proses dekripsi. Proses dekripsi pada algoritma RSA memerlukan kunci privat. Kunci privat hanya diketahui oleh orang yang berhak atas informasi tersebut (Andri, 2010). Alur pembangkitan pasangan kunci RSA, enkripsi dan dekripsi RSA digambarkan pada Gambar 2.4, 2.5 dan 2.6.



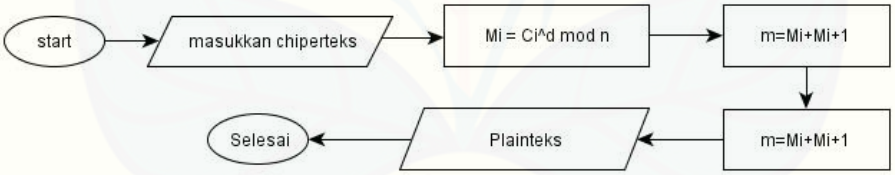
Gambar 2.4 Alur Pembangkitan Pasangan Kunci RSA

Sumber: Hasil analisis (Andri, 2010)



Gambar 2.5 Alur Enkripsi RSA

Sumber: Hasil analisis (Andri, 2010)



Gambar 2.6 Alur Deskripsi RSA

Sumber: Hasil analisis (Andri, 2010)

BAB 3. METODOLOGI PENELITIAN

Metodologi penelitian merupakan bagian yang ditujukan untuk mendapat gambaran tahapan yang sistematis, yang dilakukan untuk menganalisa data dan mengembangkan sistem pada penelitian ini.

3.1. Jenis Penelitian

Penelitian ini menggunakan dua jenis penelitian, yaitu penelitian kualitatif dan penelitian kuantitatif. Jenis penelitian kualitatif digunakan karena penelitian ini menganalisa studi kasus pada PT. KAI yang berada di Kabupaten Jember dan jenis penelitian kuantitatif digunakan karena dalam penelitian ini menerapkan serta mengkaji teori yang sudah ada sebelumnya.

3.2. Metode Penelitian

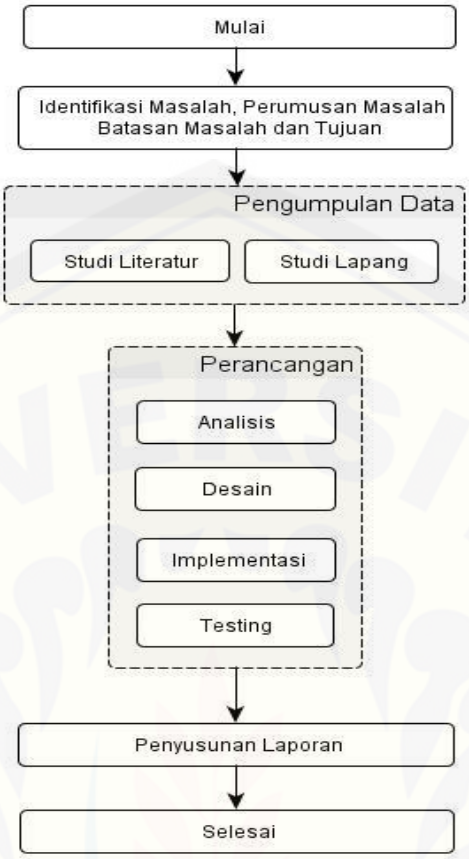
Metode penelitian yang digunakan pada penelitian ini adalah metode studi kasus. Metode studi kasus digunakan pada penelitian ini karena ditemukannya suatu proses kegiatan yang sangat sering dilakukan dan kasus tersebut akan diteliti untuk mendapatkan pemahaman yang lebih tentang kasus tersebut hingga dapat menyelesaikan permasalahan yang ada pada penelitian ini.

3.3. Tempat dan Waktu Penelitian

Penelitian dilakukan di PT. KAI yang berada di Kabupaten Jember. Waktu maksimal dilaksanakannya penelitian adalah selama 5 bulan yaitu pada Mei 2015 sampai September 2015.

3.4. Tahapan Penelitian

Tahapan yang digunakan dalam penelitian ini aadalah tahap indentifikasi masalah, tahap pengumpulan data, tahap perancangan yakni tahap analisis dan pengembangan sistem dan tahapan penyusunan laporan. Gambaran tahapan penelitian dapat dilihat pada diagram alir pada gambar 3.1.



Gambar 3.1 Diagram Alir Tahapan Penelitian

Sumber: Hasil analisis 2015

Tahapan pertama yang dilakukan dalam penelitian ini adalah tahapan identifikasi masalah. Masalah yang ada pada instansi ddidapatkan melalui pendekatan kualitatif dan kwantitaif dan didapatkan beberapa masalah, yaitu banyaknya user yang kesulitan karena mencetak tiket. Selain itu biaya produksi yang tinggi dapat dimimalisir.

Tahan kedua yaitu pengumpulan data yakni studi literatur dengan melakukan pencarian referensi dan studi lapang dengan melakukan wawancara pada pihak terkait.

Setelah pengumpulan data, maka dilakukan perancangan untuk membuat sistem, yakni analisis data untuk mendapatkan metode yang tepat, desain, implementasi dan testing untuk melakukan pengecekan terhadap sistem yang telah kita buat apakah telah sesuai atau tidak.

Langkah akhir adalah melakukan penyusunan laporan saat semua langkah telah sesuai dimana diawali dengan analisis data-data dan permasalahan secara keseluruhan untuk mendapatkan metode yang tepat agar dapat digunakan oleh sistem. Tahap penyusunan laporan dimana ketika semua tahapan sebelumnya telah berhasil di selesaikan. Tahapan-tahapan penelitian akan dijelaskan lebih rinci sebagai berikut.

3.5.1. Tahap Pengumpulan Data

Tahapan pengumpulan data adalah tahapan pencarian data-data dan informasi yang dibutuhkan untuk membangun sistem dengan beberapa teknik sebagai berikut.

Data yang dibutuhkan untuk menyelesaikan penelitian ini berdasarkan hasil dari teknik penelitian berupa data primer dan data sekunder:

1. Data Primer

   Data primer yang didapatkan dari observasi dan wawancara oleh narasumber PT. KAI adalah penjelasan tentang langkah pemesanan tiket yang ada dan dipergunakan saat ini.

2. Data Sekunder

   Data sekunder diperoleh dari studi literatur berupa buku, jurnal, referensi yang didapatkan dari internet, penelitian sebelumnya, dan data-data yang berhubungan dengan *QR-Code*, Android, Algoritma RSA, Kriptografi dan studi kasus penyelesaian untuk permasalahan enkripsi data.

Data hasil dari studi literatur dan studi lapang yang telah dikumpulkan adalah peta rute jalur kereta api dan jadwal pemberangkatan kereta api, yang akan dijelaskan di bawah ini.

3.5.1.1 Peta Rute Jalur Kereta Api

Peta rute jalur kereta api merupakan keterangan dari data kereta api yang dimiliki oleh PT. KAI, meliputi nama kereta api, stasiun pemberangkatan dan stasiun tujuan. Data ini merupakan salah satu inti dari sistem ini, yakni untuk pembuatan jadwal dan rute kereta.

3.5.1.2 Data Jadwal Pemberangkatan Kereta Api

Data jadwal pemberangkatan kereta api merupakan data yang didapatkan dari situs resmi PT. KAI. Data ini nantinya akan digunakan untuk pembuatan jadwal pemberangkatan kereta api sebagai acuan bagi pemesanan kereta. Data jadwal pemberangkatan kereta api dapat dilihat contohnya pada Gambar 3.2.

Argo Bromo Anggrek Pagi
(Surabaya Ps.Turi - Gambir PP.)

STASIUN	KA 1	
	DATANG	BERANGKAT
Surabaya Pasarturi	-	**08.00**
Semarang Tawang	11.23	11.30
Pekalongan	12.39	12.43
Cirebon	14.16	14.22
Jatinegara	16.47	16.49
Gambir	**17.00**	-

STASIUN	KA 2	
	DATANG	BERANGKAT
Gambir	-	**09.30**
Cirebon	12.06	12.12
Pekalongan	13.46	13.50
Semarang Tawang	14.59	15.06
Surabaya Pasarturi	**18.30**	-

Argo Bromo Anggrek Malam
(Surabaya Ps.Turi - Gambir PP.)

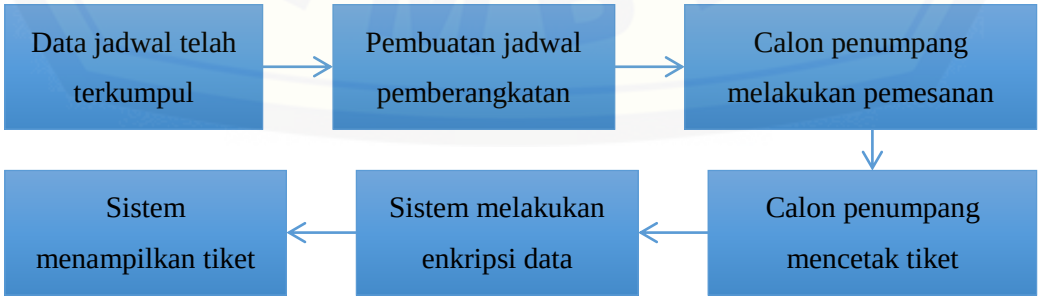
STASIUN	KA 3	
	DATANG	BERANGKAT
Surabaya Pasarturi	-	**20.00**
Semarang Tawang	23.24	23.31
Cirebon	02.12	02.19
Jatinegara	04.45	04.47
Gambir	**04.57**	-

STASIUN	KA 4	
	DATANG	BERANGKAT
Gambir	-	**21.30**
Cirebon	00.10	00.16
Semarang Tawang	02.58	03.04
Surabaya Pasarturi	**06.30**	-

Gambar 3.2 Jadwal Pemberangkatan Kereta

3.5.2. Tahap Analisis

Tahapan analisis merupakan tahapan dimana peneliti harus melakukan pemahaman data dengan baik sesuai dengan data yang telah diperoleh dari tahapan pengumpulan data. Tahapan selanjutnya adalah tahapan untuk menganalisa data yang telah didapatkan dengan Algoritma RSA. Tahapan proes analisis data dapat dilihat pada gambar 3.3.



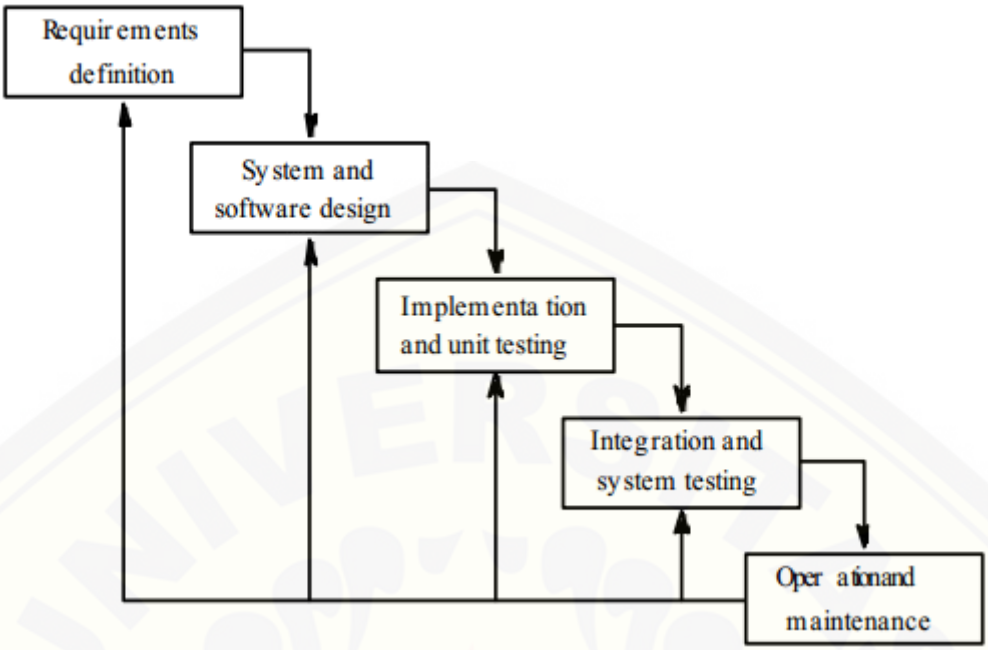
Gambar 3.3 Diagram Blok Tahapan Analisis

Pembuatan tiket memerlukan data-data diantaranya data lokasi, data jadwal, dan data calon penumpang yang digunkaan sebagai data input untuk melakukan tahapan pengenkripsian menggunakan Algoritma Kriptografi RSA. Enkripsi dilakukan saat user melakukan download tiket berdasarkan kode *booking* yang dimilikinya. Output yang didapatkan dari sistem ini adalah tiket yang berisi *QR-Code* yang telah dienkripsi.

3.5.3. Tahap Pengembangan Sistem

Tahap pengembangan sistem menggunakan model waterfall yang merupakan model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah *Linear Sequential Model*. Model ini sering disebut dengan *classic life cycle* atau model *waterfall*. Model ini melakukan pendekatan secara sistematis dan berurutan. Pemberian nama *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya secara berurutan.

Model *waterfall* adalah pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematik dan sekuensial yakni dimulai dari analisis, *design*, kode, pengujian hingga pemeliharaan. Urutan pengerjaan menggunakan model *waterfall* dapat dilihat pada Gambar 3.4 di bawah ini.



Gambar 3.4 *Waterfall* Model

3.5.3.1.Analisis Kebutuhan

Tahap analisis kebutuhan adalah peneliti setelah melakukan pencarian data, yakni dengan cara wawancara, melakukan studi literatur yang terkait dengan penelitian, melakukan studi dengan sistem yang telah ada menemukan permasalahan yang nantinya dianalisis kebutuhan yang diperlukan untuk mendapatkan solusi yang baik dalam penyelesaian masalah tersebut. Tahap ini juga dibutuhkan untuk mendapatkan data yang dibutuhkan dalam pembuatan sistem, seperti kebutuhan fungsional dan kebutuhan non fungsional dari sistem terkait yang akan dibangun.

Analisis kebutuhan yang dilakukan pada penelitian ini dengan melakukan wawancara pada PT. KAI yang berada di Kabupaten Jember, khususnya bagian petugas penjaga Peron, *Customer Service* (CS) dan staff managerial PT. KAI yang berada di Kabupaten Jember.

3.5.3.2.Desain Sistem

Pembuatan desain sistem pada aplikasi penelitian ini menggunakan *Unified Modeling Languange* (*UML*) yang dirancang dengan konsep *Object-Oriented Programming* (*OOP*). Pemodelan UML yang digunakan adalah sebagai berikut:

1. *Business Process*
2. *Use Case Diagram*
3. *Scenario* system.
4. *Activity Diagram.*
5. *Sequence Diagram*.
6. *Class Diagram*.

3.5.3.3.Implementasi

Langkah pada implementasi adalah pembuatan kode program yang dibuat berdasarkan desain yang telah dibuat sebelumnya. Sehingga pada tahap implementasi perlu dilakukan tahapan yakni penulisan kode program (*coding*) pada bagian pemesanan tiket menggunakan bahasa pemograman *Page Hyper Text Pre-Processor (PHP), Cascading Style Sheet (CSS),* dan *Hyper Text Markup Languange (HTML),* dan manajemen basis data menggunakan *DBMS MYSql.* Berbeda dengan bagian pemesanan tiket, penulisan kode program (*coding*) pada bagian verifikasi menggunakan bahasa pemrograman Java.

3.5.3.4.Pengujian

Pengujian sangat dibutuhkan untuk memastikan dan menjaga kualitas dari aplikasi yang kita buat, untuk melakukan evaluasi dari perbedaan antara fitur yang diinginkan pada tahap analisis dengan sistem yang telah terbentuk. Pengujian merupakan proses untuk mengidentifikasi adanya sesuatu yang tidak sesuai pada aplikasi yang diharapkan. Pada aplikasi ini dilakukan dua tahap pengujian, yakni *white box testing* dan *black box testing*.

1. *White box testing* adalah pengujian pada fungsi internal dari sudut pandang pengembang yang berfokus pada logika internal dan struktur kode. Teknik ini menguji setiap cabang dan keputusan dalam program untuk menemukan kesalahan ketik acak, *debugging* kode dan mengungkapkan pemrograman yang salah asumsi. Tahapan teknik pengujian *white box* meliputi *listing* program, grafik alir, kompleksitas siklomatik, jalur program independen dan pengujian basis *set*.

2. *Black box testing* adalah pengujian berdasarkan spesifikasi kebutuhan yang ada dan tidak memerlukan pemeriksaan kode yang ada. Pengujian ini dilakukan berdasarkan persyaratan-persyaratan yang diberikan, sehingga kesalahan-kesalahan yang ada dapat diidentifikasikan dan dapat segera ditangani dengan mudah.
   a) *Listing* Program
   b) Grafik Alir
   c) Kompleksitas Siklomatik
   d) Jalur Program Independen
   e) Pengujian Basis Set

3.5.3.5.Pemeliharaan

Pemeliharaan dilakukan ketika sistem mengalami perubahan yang disebabkan oleh kebutuhan pengguna yang dapat dikarenakan penyesuaian dengan kebutuhan yang sebenarnya.

3.5. Gambaran Umum Sistem

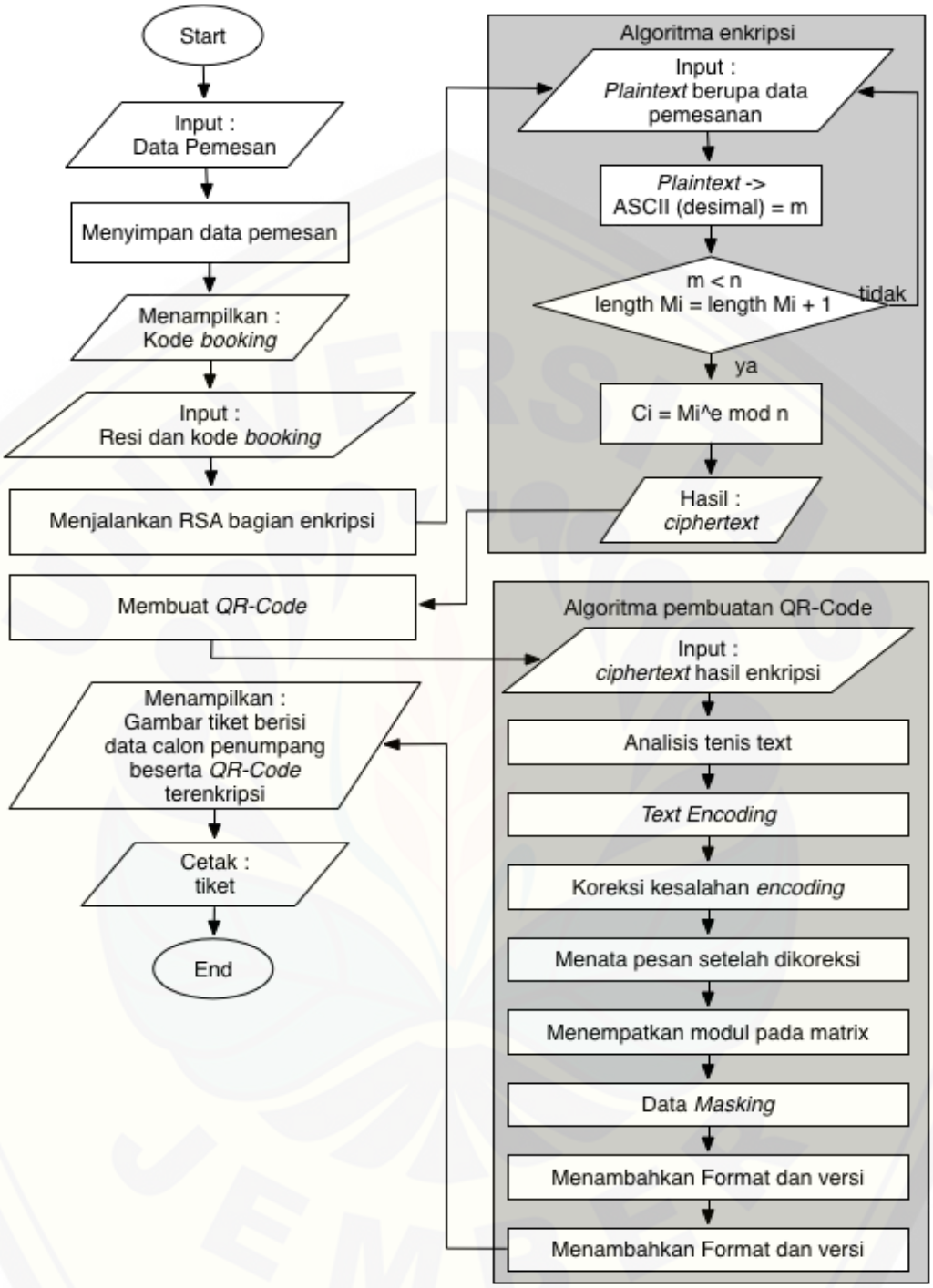
Aplikasi yang akan dibangun adalah aplikasi verifikasi pemesanan tiket dengan *QR-Code* berbasis android menggunakan algoritma kriptografi asimetris RSA. Aplikasi ini berguna untuk melakukan verifikasi terhadap tiket kereta api. Aplikasi ini dibuat untuk mempermudah pelanggan dalam hal pemesanan tiket dan pembuatan tiket serta memberikan keuntungan pada PT. KAI karena dapat mengurangi biaya operasional. Sistem ini didukung oleh sistem berbasis web yang digunakan pada proses pemesanan tiket. Saat dilakukan pemesanan tiket sistem berbasis web akan memberikan kode *booking* kepada calon penumpang yang berfungsi sebagai kode pembayaran. Setelah calon penumpang membayar dan melakukan konfirmasi pembayaran pada sistem, calon penumpang akan mendapat gambar tiket yang dapat langsung dicetak tanpa harus datang ke stasiun tertentu.

Aktor yang terlibat dalam penggunaan aplikasi ini adalah (1) *administrator* perusahaan yang bertugas untuk mengelola data pembayaran tiket penumpang, mengelola pemberian kode *booking* kepada calon penumpang dan melakukan

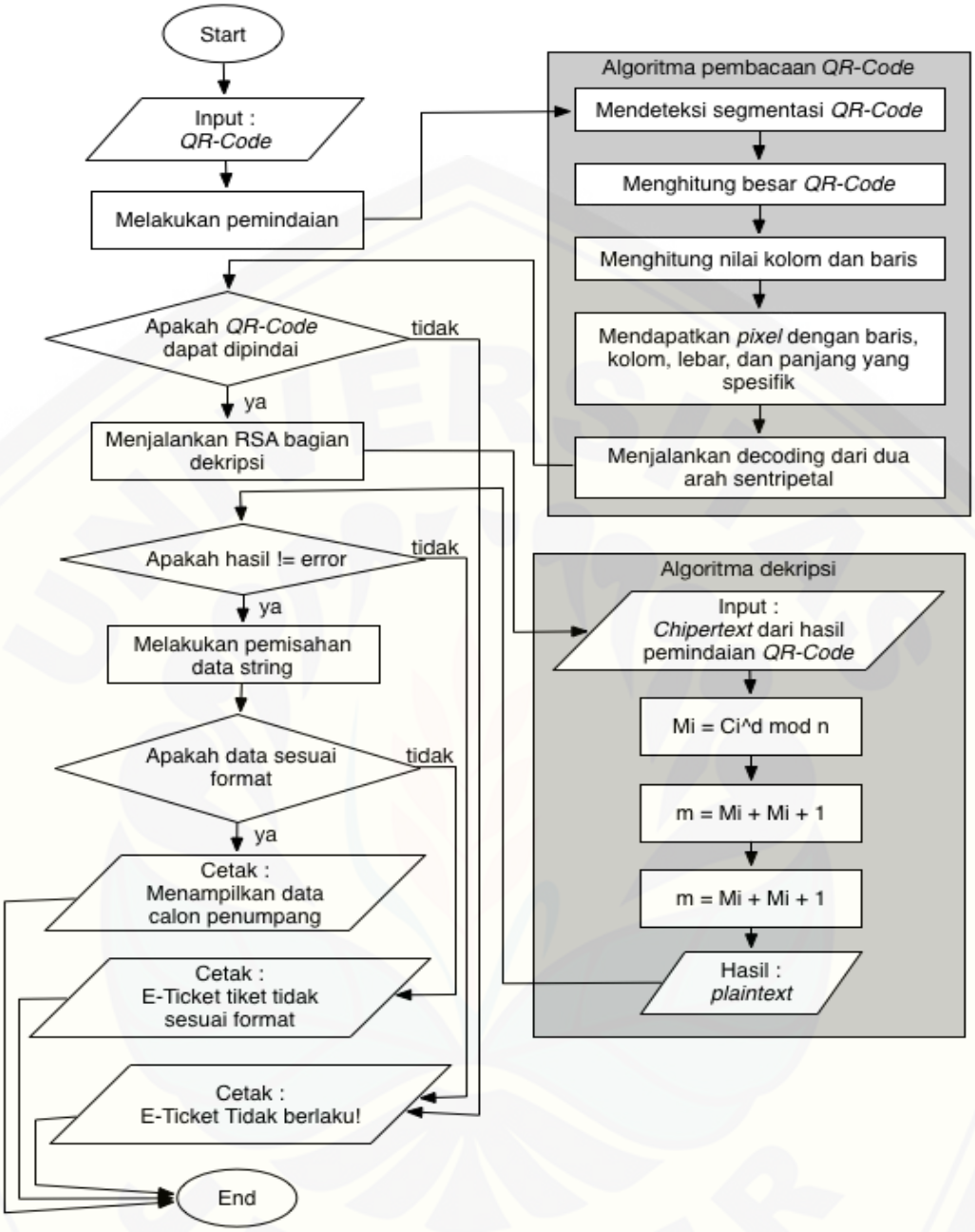
proses enkripsi *QR-Code*, (2) calon penumpang kereta api yang dapat melakukan pembayaran tiket dan mendapatkan *QR-Code* yang sudah terenkripsi dan (3) petugas PT. KAI yang betugas sebagai pemegang aplikasi verifikasi tiket.

Gambaran penggunaan aplikasi ini adalah calon penumpang kereta api dapat melakukan pemesanan tiket pada sistem pemesanan tiket *online*. Setelah itu calon penumpang dapat mengunduh tiket yang didalamnya tertera *QR-Code* yang sudah dienkripsi menggunakan algoritma kriptografi asimetris RSA oleh perusahaan. *QR-Code* berisi data pemesan dan data yang berhubungan dengan informasi pemberangkatan pemesan. *QR-Code* ini dapat secara mudah dipindai oleh *scanner* manapun akan tetapi pendekripsian kode hanya bisa dilakukan oleh petugas PT. KAI sebagai pemegang aplikasi verifikasi tiket. Algoritma kriptografi asimetris digunakan dalam aplikasi ini karena menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Kunci enkripsi hanya diketahui oleh orang yang berwenang sehingga *QR-Code* yang nantinya diisi dengan data pemesanan yang telah dienkripsi menggunakan algoritma kriptografi asimetris RSA tidak mudah untuk ditiru atau digandakan.

Berikut dua *flowchart* yang digunakan untuk menggambarkan alur sistem ini, yaitu *flowchart* pembuatan tiket pada Gambar 3.5 dan *flowchart* verifikasi tiket pada Gambar 3.6.



Gambar 3.5 *Flowchart* Pemesanan Tiket



Gambar 3.6 *Flowchart* Verifikasi Tiket

BAB 4. PENGEMBANGAN SISTEM

Bab ini merupakan bagian yang membahas tentang pengembangan sistem verifikasi pemesanan tiket secara keseluruhan dengan menggunakan algoritma Kriptografi RSA. Pengembangan sistem dilakukan dengan menggunakan model waterfall, dengan tahapan yakni analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program dan pengujian sistem.

4.1 Statement Of Purpose

Statement Of Purpose pada sistem ini digunakan untuk mengelola data-data yang diberikan oleh PT. KAI yakni data tiket, data penumpang dan data jadwa keberangkatan. Algoritma RSA dalam penggunaan sistem ini digunakan untuk melakukan verifikasi pemesanan pada tiket kereta api.

4.2 Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem merupakan tahapan yang sangat penting dalam pengembangan sebuah sistem informasi karena kebutuhan pengguna didefinisikan dan diformulasikan pada tahap ini.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional dari sistem verifikasi pemesanan tiket antara lain:

1. Sistem dapat menyimpan dan memanejemen (*create,update,delete*) seluruh data user.
2. Sistem dapat menyimpan dan memanejemen (*create,update,copy*) seluruh data jadwal keberangkatan.
3. Sistem dapat menyimpan dan memanejemen (*create,issued,print*) seluruh data transaksi pemesanan tiket.
4. Sistem dapat menyimpan dan memanejemen (*create,update*) seluruh data kota yang dilewati rute kereta api.
5. Sistem dapat menyimpan dan memanejemen (*create,update*) seluruh data member.

6. Sistem dapat menyimpan dan memanejemen (*search,print*) seluruh data laporan dari pemesanan tiket.
7. Sistem dapat menyimpan dan memanejemen (*search*) seluruh data saldo terbaru.
8. Sistem dapat menyimpan dan memanejemen (*search*) seluruh data laporan penjualan tiket.
9. Sistem dapat mendownload tiket yang telah dipesan menggunakan kode *booking* yang telah diterima.
10. Terdapat algoritma pengamanan data menggunakan kriptografi saat penyisipan data dalam proses pembuatan tiket.
11. Terdapat beberapa member utama sebagai aktor sistem, yaitu super admin, pengelola blog, dan member.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non fungsional dari sistem verifikasi pemesanan tiket antara lain:

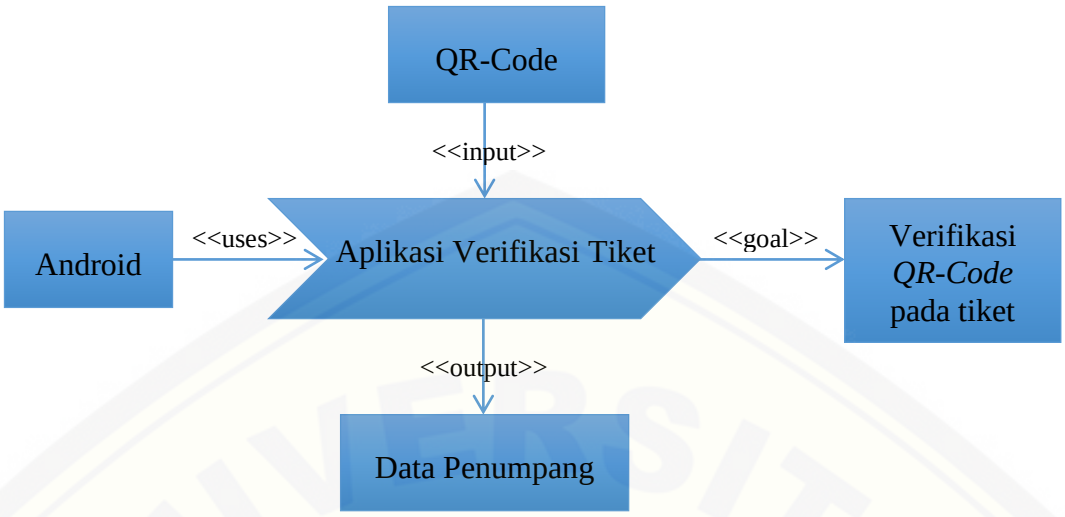
1. Tampilan sistem menarik dan mudah di pahami oleh seluruh user.
2. Sangat mudah untuk di operasikan.

4.3 Desain Sistem

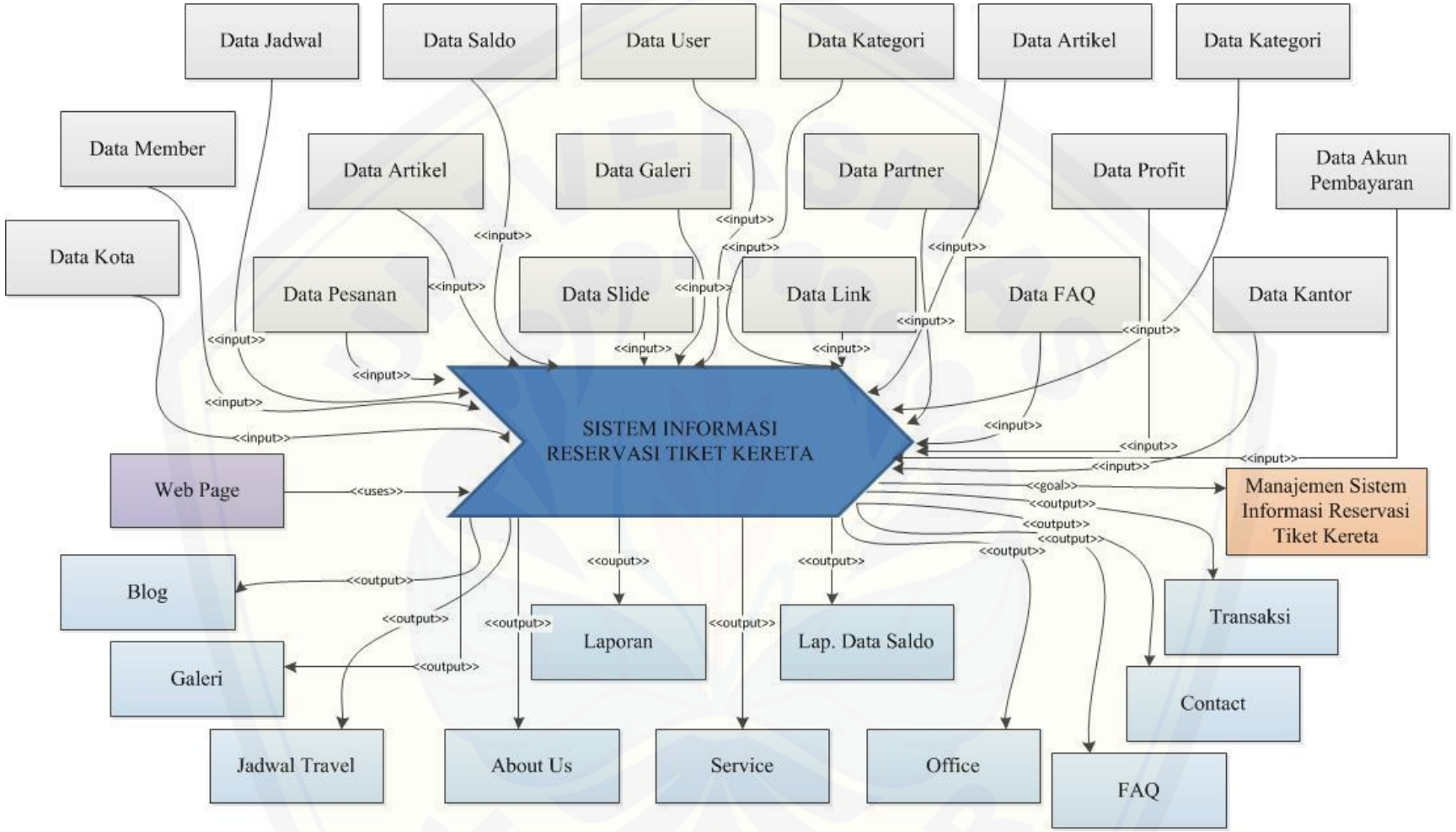
Tahapan ini menggambarkan aplikasi verifikasi pemesanan tiket dengan tahapan yakni bussines Process, Usecase Diagram, Usecase Skenario, Activity Diagram, Sequence Diagram, Class Diagram dan Entity Relationship Diagram (ERD).

4.3.1 Business Process

Business Process adalah sekumpulan proses yang dilakukan untuk mencapai hasil yang diinginkan dengan beberapa bagian yakni *input*, *output* dan *goal* yang ingin dicapai. *Business Process* aplikasi verifikasi tiket dapat dilihat pada gambar 4.1 dan *Business Process* sistem pemesanan tiket dapat dilihat pada gambar 4.2.



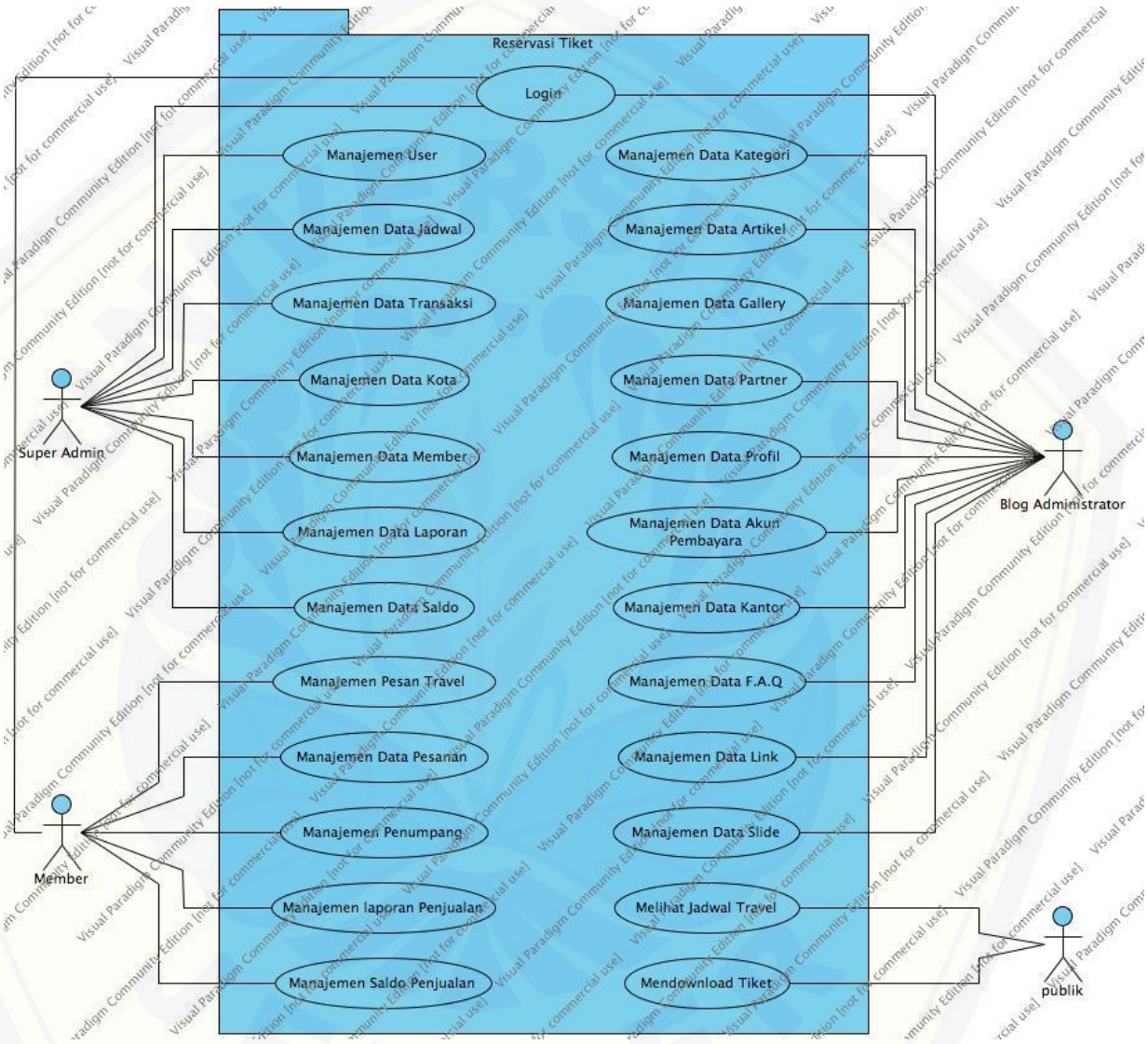
Gambar 4.1 *Bussiness Process* Aplikasi Verifikasi Tiket



Gambar 4.2 *Bussiness Process* Sistem Pemesanan Tiket

4.3.2 Usecase Diagram

Usecase Diagram adalah dokumentasi untuk menggambarkan fitur dan aktor yang terdapat pada sistem yang akan dibuat. *Usecase* diagram sistem pemesanan tiket dapat dilihat seperti yang dijelaskan pada Gambar 4.3.



Gambar 4.3 Usecase Diagram Sistem Pemesanan Tiket

Penjelasan tentang definisi aktor dan definisi *usecase* dalam *Usecase* diagram Sistem Pemesanan Tiket akan dijelaskan di bawah ini.

1. Definisi Aktor

Tahap ini menjelaskan tentang aktor yang terdapat pada sistem yang akan dibagun. Terdapat 4 (empat) aktor dari hasil analisis seperti yang dijelaskan pada table 4.1.

Tabel 4.1 Definisi Aktor

No	Aktor	Deskripsi
1.	*Super Admin*	Aktor Super Admin memiliki hak akses yang dapat melakukan proses login, manajemen user, manajemen data jadwal, manajemen data transaksi, manajemen data kota, manajemen data member, manajemen data laporan dan manajemen data saldo.
2.	*Member*	Aktor Member memiliki hak akses yang dapat melakukan proses login, manajemen pesan travel, manajemen data pesanan, manajemen penumpang, manajemen laporan penjualan, manajemen saldo penjualan.
3	*Blog Administrator*	Aktor Member memiliki hak akses yang dapat melakukan proses login, manajemen data kategori, manajemen data artikel, manajemen data galery, manajemen data partner, manajemen data profil, manajemen data akun pembayaran, manajemen data kantor, manajemen data F.A.Q, Manajemen data link, Manajemen data slide.
4	*Publik*	Aktor publik memiliki hak akses yang dapat melakukan proses melihat jadwal travel, mendownload tiket.

2. Definisi *Usecase*

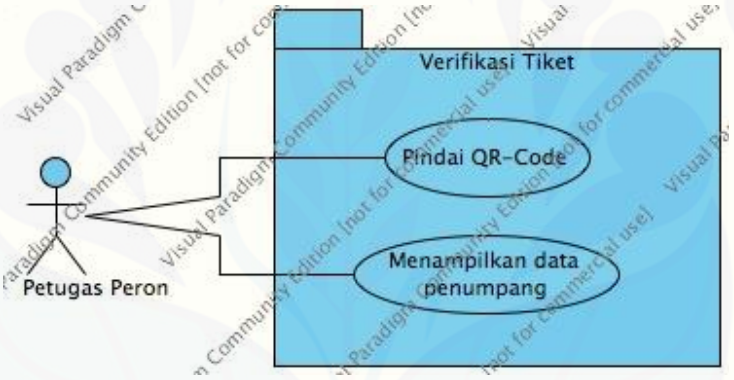
Definisi *Usecase* menjelaskan masing – masing *Usecase* atau fitur – fitur pada sistem pemesanan tiket. Penjelasan *usecase* dapat di lihat pada Tabel 4.2.

Tabel 4.2 Definisi Usecase

No	*Usecase*	Penjelasan
1	*Login*	*Usecase* untuk proses login atau autentifikasi untuk masuk sistem.
2	Manajemen User	*Usecase* untuk proses mengelola data user.
3	Manajemen Data Jadwal	*Usecase* untuk proses mengelola data jadwal keberangkatan.
4	Manajemen Data Transaksi	*Usecase* untuk proses mengelola data transaksi pemesanan tiket.
5	Manajemen Data Kota	*Usecase* untuk proses mengelola data kota.
6	Manajemen Data Member	*Usecase* untuk proses mengelola data member.
7	Manajemen Data Laporan	*Usecase* untuk proses mengelola data laporan.
8	Manajemen Data Saldo	*Usecase* untuk proses mengelola data saldo.
9	Manajemen Pesan Travel	*Usecase* untuk proses mengelola data pemesanan travel.
10	Manajemen Data Pesanan	*Usecase* untuk proses mengelola data pesanan tiket.
11	Manajemen Penumpang	*Usecase* untuk proses mengelola data penumpang.
12	Manajemen Laporan Penjualan	*Usecase* untuk proses mengelola data laporan penjualan tiket kereta api.
13	Manajemen Saldo Penjualan	*Usecase* untuk proses mengelola data saldo penjualan.
14	Manajemen Data Kategori	*Usecase* untuk proses mengelola data manajemen data kategori.
15	Manajemen Data Artikel	*Usecase* untuk proses mengelola data artikel.
16	Manajemen Data Galery	*Usecase* untuk proses mengelola data galeri.
17	Manajemen Data Partner	*Usecase* untuk proses mengelola data partner.
18	Manajemen Data Profil	*Usecase* untuk proses mengelola data profil PT.

		KAI.
19	Manajemen Data Akun Pembayaran	*Usecase* untuk proses mengelola data akun pembayaran.
20	Manajemen Data Kantor	*Usecase* untuk proses mengelola data kantor.
21	Manajemen Data F.A.Q	*Usecase* untuk proses mengelola data F.A.Q.
22	Manajemen Data Link	*Usecase* untuk proses mengelola data link.
23	Manajemen Data Slide	*Usecase* untuk proses mengelola data silde.
24	Melihat Jadwal Travel	*Usecase* untuk proses mengelola data jadwal travel.
25	Mendownload Tiket	*Usecase* untuk proses mendownload tiket.

Usecase diagram aplikasi verifikasi dapat dilihat seperti yang dijelaskan pada Gambar 4.4.



Gambar 4.4 Usecase Diagram Aplikasi Verifikasi Tiket

Penjelasan tentang definisi aktor dan definisi *usecase* dalam *Usecase* diagram Aplikasi Verifikasi Tiket akan dijelaskan di bawah ini.

1. Definisi Aktor

Tahap ini menjelaskan tentang aktor yang terdapat pada sistem yang akan dibagun. Terdapat 4 (empat) aktor dari hasil analisis seperti yang dijelaskan pada table 4.3.

Tabel 4.3 Definisi Aktor

No	Aktor	Deskripsi
1.	Petugas Peron	Aktor Petugas Peron memiliki hak akses yang dapat melakukan proses pemindaian *QR-Code* pada tiket dan melihat data hasil pindaian pada aplikasi.

2. Definisi *Usecase*

Definisi *Usecase* menjelaskan masing – masing *Usecase* atau fitur – fitur pada aplikasi verifikasi tiket. Penjelasan *usecase* dapat di lihat pada Tabel 4.4.

Tabel 4.4 Definisi Usecase

No	*Usecase*	Penjelasan
1	Memindai QR-Code	*Usecase* untuk melakukan pemindaian terhadap *QR-Code* yang ada pada tiket.
2	Menampilkan Data Penumpang	*Usecase* untuk proses mengelola data user.

4.3.3 Usecase Skenario

Usecase skenario adalah dokumentasi terhadap kebutuhan fungsional sistem. *Usecase* skenario sistem pemesanan tiket adalah sebagai berikut.

1. *Usecase* Skenario Manajemen Data Transaksi

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mendownload tiket dapat dilihat pada Tabel 4.5

Tabel 4.5 *Usecase* Skenario Mendownload Tiket

Nama Usecase	Mendownload tiket
Aktor	Member
Pre Kondisi	Member telah memilih jadwal travel
Post Kondisi	Member mendownload tiket
SKENARIO MENDOWNLOAD TIKET	

NORMAL SKENARIO DOWNLOAD TIKET	
1. Klik *icon* "cetak tiket"	
	2. Menampilkan halaman tiket pada *browser*.
3. Menyimpan file tiket	

2. *Usecase* Skenario *Login*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *login* dapat dilihat pada Lampiran A (*Usecase Scenario*).

3. *Usecase* Skenario Manajemen Data *User*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data *user* dapat dilihat pada Lampiran A (*Usecase Scenario*).

4. *Usecase* Skenario Manajemen Data Jadwal

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data jadwal dapat dilihat pada Lampiran A (*Usecase Scenario*).

5. *Usecase* Skenario Manajemen Data Transaksi

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data transaksi dapat dilihat pada Lampiran A (*Usecase Scenario*).

6. *Usecase* Skenario Manajemen Data Kota

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data kota dapat dilihat pada Lampiran A (*Usecase Scenario*).

7. *Usecase* Skenario Manajemen Data Member

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data member dapat dilihat pada Lampiran A (*Usecase Scenario*).

8. *Usecase* Skenario Manajemen Data Laporan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data laporan dapat dilihat pada Lampiran A (*Usecase Scenario*).

9. *Usecase* Skenario Manajemen Data Saldo

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data saldo dapat dilihat pada Lampiran A (*Usecase Scenario*).

10. *Usecase* Skenario Manajemen Pesan Travel

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen pesan travel dapat dilihat pada Lampiran A (*Usecase Scenario*).

11. *Usecase* Skenario Manajemen Data Pesanan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data pesanan dapat dilihat pada Lampiran A (*Usecase Scenario*).

12. *Usecase* Skenario Manajemen Penumpang

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen penumpang dapat dilihat pada Lampiran A (*Usecase Scenario*).

13. *Usecase* Skenario Manajemen Laporan Penjualan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen laporan penjualan dapat dilihat pada Lampiran A (*Usecase Scenario*).

14. *Usecase* Skenario Manajemen Saldo Penjualan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen saldo penjualan dapat dilihat pada Lampiran A (*Usecase Scenario*).

15. *Usecase* Skenario Manajemen Data Kategori

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data kategori dapat dilihat pada Lampiran A (*Usecase Scenario*).

16. *Usecase* Skenario Manajemen Data Artikel

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data artikel dapat dilihat pada Lampiran A (*Usecase Scenario*).

17. *Usecase* Skenario Manajemen Data Galery

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data galery dapat dilihat pada Lampiran A (*Usecase Scenario*).

18. *Usecase* Skenario Manajemen Data Partner

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data partner dapat dilihat pada Lampiran A (*Usecase Scenario*).

19. *Usecase* Skenario Manajemen Data Profil

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data profil dapat dilihat pada Lampiran A (*Usecase Scenario*).

20. *Usecase* Skenario Manajemen Data Akun Pembayaran

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data akun pembayaran dapat dilihat pada Lampiran A (*Usecase Scenario*).

21. *Usecase* Skenario Manajemen Data Kantor

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data kantor dapat dilihat pada Lampiran A (*Usecase Scenario*).

22. *Usecase* Skenario Manajemen Data *F.A.Q*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data *f.a.q* dapat dilihat pada Lampiran A (*Usecase Scenario*).

23. *Usecase* Skenario Manajemen Data *Link*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data link dapat dilihat pada Lampiran A (*Usecase Scenario*).

24. *Usecase* Skenario Manajemen Data *Slide*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data slide dapat dilihat pada Lampiran A (*Usecase Scenario*).

25. *Usecase* Skenario Melihat Jadwal Travel

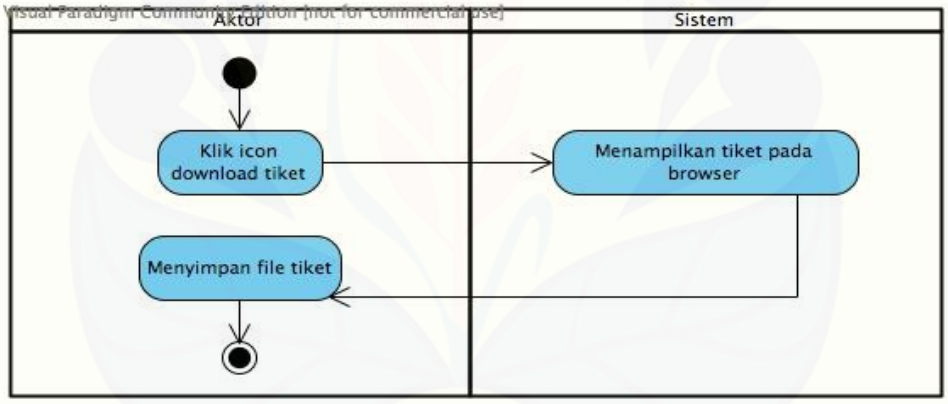
Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat jadwal travel dapat dilihat pada Lampiran A (*Usecase Scenario*).

4.3.4 Activity Diagram

Activity Diagram menggambarkan aliran aktivitas dalam system pemesanan tiket, yang dapat dilihat sebagai berikut:

1. *Activity* Diagram Mendownload Tiket

Activity diagram dalam mengunduh tiket dapat dilakukan oleh member. *Activity* diagram ini digunakan untuk melakukan melakukan unduh tiket yang akan dilakukan oleh member, yang dapat dilihat pada gambar 4.5.



Gambar 4.5 Activity Diagarm Mendownload Tiket

2. *Activity* Diagram *Login*

Activity diagram untuk *login* dapat digunakan oleh seluruh member. *Activity* diagram ini digunakan untuk melakukan login sebelum memasuki sistem dengan hak akses *super admin*, *blog administrator*, atau *member*, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

3. *Activity* Diagram Manajemen Data *User*

Activity diagram untuk manajemen data *user* dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap user dari sistem ini, seperti menambah, mengubah dan menghapus user, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

4. *Activity* Diagram Manajemen Data Jadwal

Activity diagram untuk manajemen data jadwal dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap jadwal pemberangkatan kereta, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

5. *Activity* Diagram Manajemen Data Transaksi

Activity diagram untuk manajemen data transaksi dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap transaksi pemesanan dan pembayaran kereta, seperti menambah data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

6. *Activity* Diagram Manajemen Data Kota

Activity diagram untuk manajemen data kota dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data kota yang dilalui oleh kereta, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

7. *Activity* Diagram Manajemen Data *Member*

Activity diagram untuk manajemen data *member* dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap member yang dapat memberli atau menjual tiket kereta, seperti menambah member, menambah saldo, mengubah, dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

8. *Activity* Diagram Manajemen Data Laporan

Activity diagram untuk manajemen data laporan dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data laporan transaksi tiket kereta, seperti melihat data transaksi, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

9. *Activity* Diagram Manajemen Data Saldo

Activity diagram untuk manajemen data saldo dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data saldo yang dimiliki oleh member, seperti menambah saldo, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

10. *Activity* Diagram Manajemen Pesan *Travel*

Activity diagram untuk manajemen pesan *travel* dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data pemesanan tiket kereta, seperti melakukan pemesanan tiket, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

11. *Activity* Diagram Manajemen Data Pesanan

Activity diagram untuk manajemen data pesanan dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data pesanan tiket kereta, seperti menyetujui dan mencetak data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

12. *Activity* Diagram Manajemen Penumpang

Activity diagram untuk manajemen data penumpang dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data penumpang pada kereta, seperti melihat data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

13. *Activity* Diagram Manajemen Laporan Penjualan

Activity diagram untuk manajemen laporan penjualan dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data transaksi penjualan tiket, seperti melihat data laporan, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

14. *Activity* Diagram Manajemen Saldo Penjualan

Activity diagram untuk manajemen saldo penjualan dapat digunakan oleh *super admin*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap saldo dari member, seperti menambah data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

15. *Activity* Diagram Manajemen Data Kategori

Activity diagram untuk manajemen data kategori dapat digunakan oleh *blog administrator*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data kategori berita pada *blog* PT. KAI, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

16. *Activity* Diagram Manajemen Data Artikel

Activity diagram untuk manajemen data artikel dapat digunakan oleh *blog administrator*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data artikel pada blog PT. KAI, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

17. *Activity* Diagram Manajemen Data *Galery*

Activity diagram untuk manajemen data *galery* dapat digunakan oleh *blog administrator*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap isi *galery* pada blog PT. KAI, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

18. *Activity* Diagram Manajemen Data Partner

Activity diagram untuk manajemen data partner dapat digunakan oleh *blog administrator*. *Activity* diagram ini digunakan untuk melakukan manajemen data partner yang ditampilkan pada blog PT. KAI, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

19. *Activity* Diagram Manajemen Data Profil

Activity diagram untuk manajemen data profil dapat digunakan oleh *blog administrator*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data artikel profil pada blog PT. KAI, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

20. *Activity* Diagram Manajemen Data Akun Pembayaran

Activity diagram untuk manajemen data akun pembayaran dapat digunakan oleh *blog administrator*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data akun pembayaran pemesanan tiket yang ditampilkan pada blog PT. KAI, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

21. *Activity* Diagram Manajemen Data Kantor

Activity diagram untuk manajemen data kantor dapat digunakan oleh *blog administrator*. *Activity* diagram ini digunakan untuk melakukan manajemen data kantor yang dimiliki PT. KAI, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

22. *Activity* Diagram Manajemen Data *F.A.Q*

Activity diagram untuk manajemen data *f.a.q* dapat digunakan oleh *blog administrator*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data pertanyaan dan jawaban yag ditampilkan pada blog PT. KAI, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

23. *Activity* Diagram Manajemen Data *Link*

Activity diagram untuk manajemen data profil dapat digunakan oleh *blog administrator*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data *link* pada blog PT. KAI, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

24. *Activity* Diagram Manajemen Data *Slide*

Activity diagram untuk manajemen data *slide* dapat digunakan oleh *blog administrator*. *Activity* diagram ini digunakan untuk melakukan manajemen terhadap data slide yang ditampilkan pada blog PT. KAI, seperti menambah, mengubah dan menghapus data, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

25. *Activity* Diagram Melihat Jadwal Travel

Activity diagram untuk melihat jadwal travel dapat digunakan oleh *member*. *Activity* diagram ini digunakan untuk melmelihat data jadwal pemberangkatan kereta, yang dapat dilihat pada Lampiran B (*Activity* Diagram).

4.3.5 Sequence Diagram

Sequence Diagram adalah tahapan dokumentasi suatu diagram terurut yang menampilkan interaksi - interaksi antar objek di dalam system, yang juga menggambarkan skenario dan memodelkan aliran logika dalam sistem dengan cara *visual*. *Sequence* diagram dari sistem pemesanan tiket berbasis web adalah sebagai berikut:

1. *Sequence* Diagram Manajemen Data Jadwal

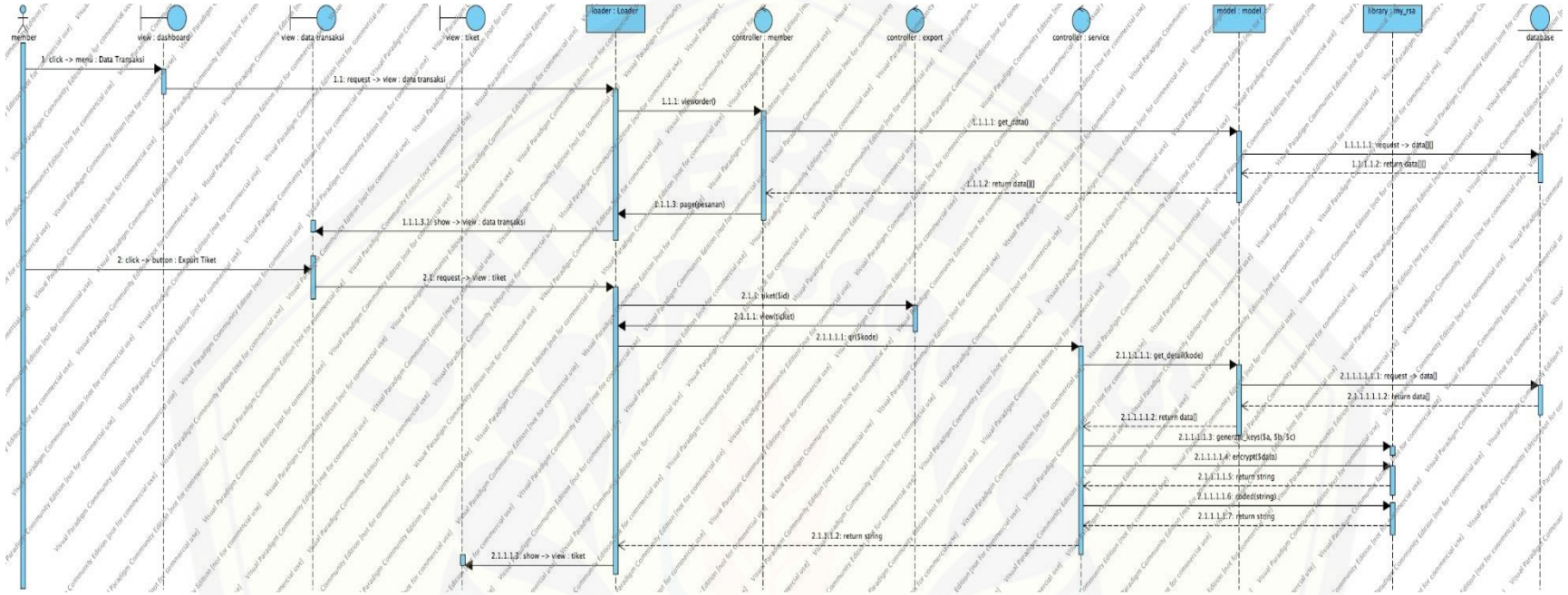
Sequence diagram manajemen data jadwal adalah tahapan yang digunakan untuk melakukan proses melihat jadwal pemberangkatan. *Class view* yang digunakan adalah jadwal yang merupakan tampilan dimana proses untuk menampilkan data jadwal pemberangkatan kereta. *Class controller* yang digunakan adalah *class* member yang merupakan *class* yang memiliki fungsi viewjadwal() untuk melakukan proses menampilkan halaman jadwal. *Class* model yang diguanakan adalah model yang digunakan untuk menampilkan jadwal. *Sequence* diagram manajemen data jadwal dapat dilihat pada Lampiran C (*Sequence* Diagram).

2. *Sequence* Diagram Manajemen Pesan Travel

Sequence diagram manajemen pesan travel adalah tahapan yang digunakan untuk melakukan proses memesan kereta. *Class view* yang digunakan adalah order yang merupakan tampilan dimana proses untuk menampilkan form pemesanan tiket kereta. *Class controller* yang digunakan adalah class member yang merupakan *class* yang memiliki fungsi order() untuk melakukan proses menampilkan form pemesanan tiket kereta. *Class* model yang diguanakan adalah model yang digunakan untuk menyimpan data pesanan tiket. *Sequence* diagram manajemen pesan travel dapat dilihat pada Lampiran C (*Sequence* Diagram).

3. *Sequence* Diagram Mendownload Tiket

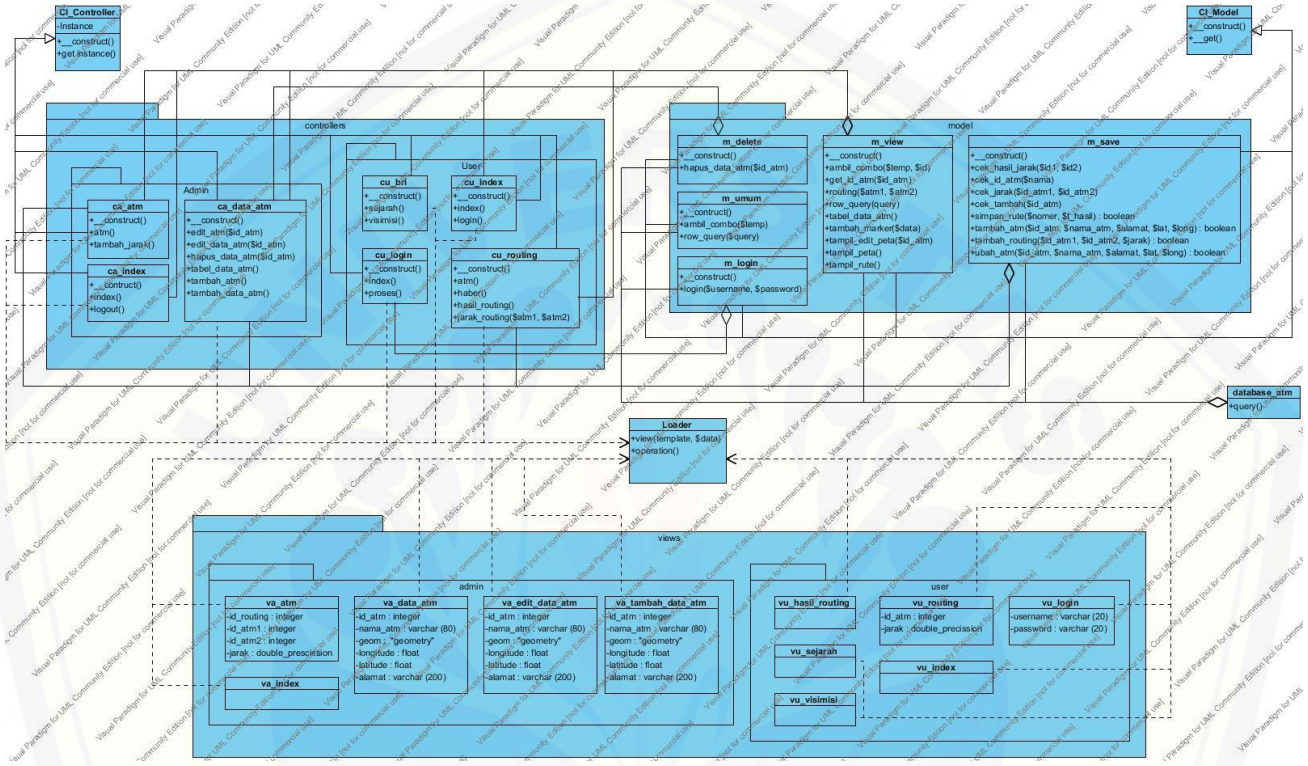
Sequence diagram mendownload tiket adalah tahapan yang digunakan untuk melakukan proses pengamanan data pada tiket menggunakan algoritma kriptografi RSA. Class view yang digunakan adalah tiket yang merupakan tampilan dimana proses pengamanan data dilakukan setelah menekan tombol print atau tombol *download* tiket. Class controller yang digunakan adalah class export yang merupakan class yang memiliki fungsi tiket() untuk melakukan proses pembuatan tiket dengan menggunakan metode dan juga fungsi export untuk menampilkan halaman tiket. Class model yang diguanakan adalah model yang digunakan untuk menampilkan tiket. Penggambaran sequence login dapat dilihat pada gambar 4.6.



Gambar 4.6 Sequence Diagram Download Tiket

4.3.6 Class Diagram

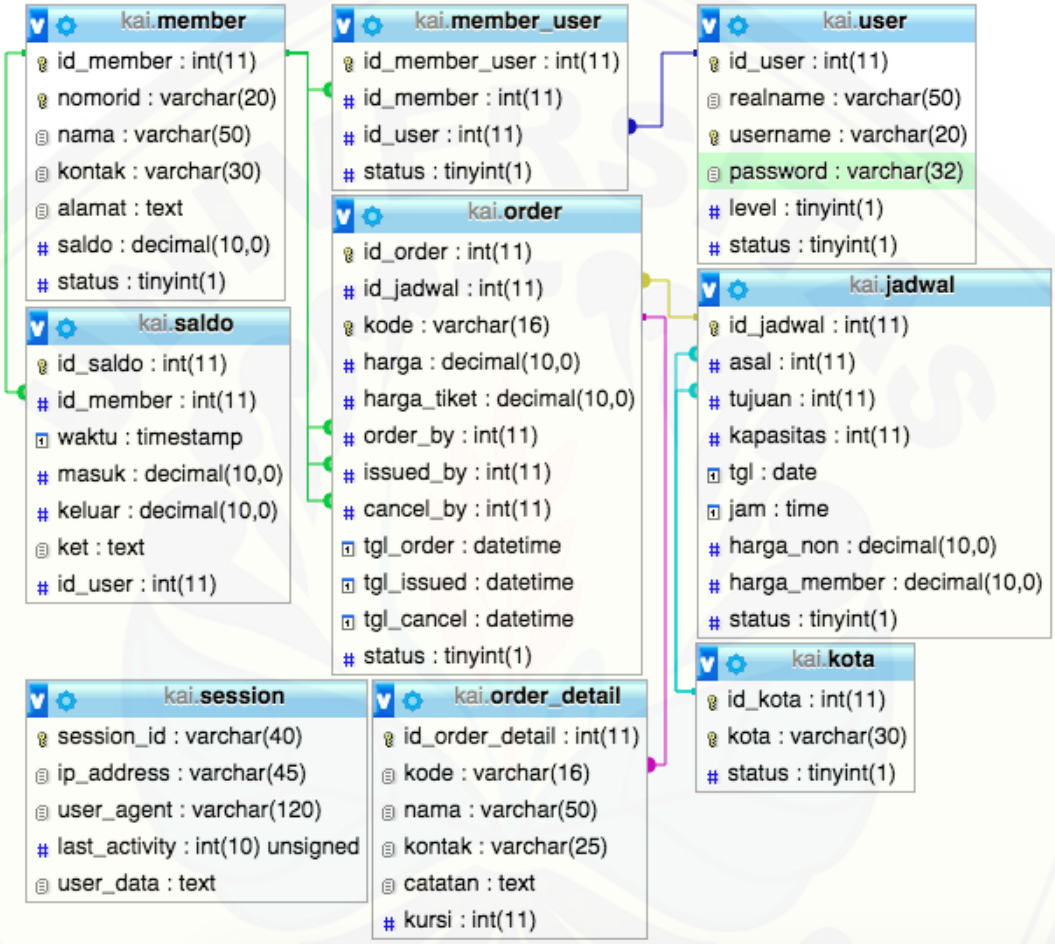
Class diagram menggambarkan hubungan antar kelas-kelas yang digunakan dalam membentuk sistem. MVC pada OOP (*Object Oriented Program)* adalah contoh-contoh kelas yang digunakan. Class Diagram yang dibangun berdasarkan *sequence* diagram pada sistem verifikasi, dapat dilihat pada gambar 4.7.



Gambar 4.7 Class Diagram Aplikasi Pemesanan Tiket

4.3.7 Entity Relationship Diagram

Entity Relationship Diagram menggambarkan komponen dan struktur dari database yang digunakan dalam membuat suatu sistem. ERD yang dihasilkan dari database sistem verifikasi pemesanan tiket, yang selanjutnya dapat dilihat pada gambar 4.8.



Gambar 4.8 Entity Relationship Diagram Aplikasi Pemesanan Tiket

4.4 Penulisan Kode Program

Penulisan kode program merupakan pengimplementasian yang dilakukan dari proses yang telah kita buat menjadi sebuah kode program.

1. Bahasa yang digunakan dalam penulisan kode program yakni bahasa pemrograman *Java*, *Page Hyper Text Pre-Processor* (PHP) dan juga menggunakan bantuan *framework Code Igniter* (CI).

2. Manajemen basis data menggunakan *DBMS MySQL*.

Kode program yang digunakan untuk proses perhitungan menggunakan metode Algoritma RSA terletak pada *library* my_rsa yang terdapat pada package user dalam package controller, yang dapat dilihat pada Tabel 4.6.

Tabel 4.6 Kode Program *library* my_rsa

```
class my_rsa {

    private static $keys;

    function __construct() {
    }

    public function generate_keys($p, $q, $show_debug = 0) {
        $n = bcmul($p, $q);

        $m = bcmul(bcsub($p, 1), bcsub($q, 1));
        if (strpos($m, ".") > -1) {
            $xxx = split(".", $m);
            $m = $xxx[0];
        }

        $e = $this->findE($m);
        $d = $this->extend($e, $m);
        $keys = array($n, $e, $d);

        if ($show_debug) {
            echo "P = $p<br>Q = $q<br><b>N = $n</b> - modulo<br>M = $m<br><b>E = $e</b> -
public key<br><b>D = $d</b> - private key<p>";
        }

        $this->keys = $keys;//return $keys;
    }

    private function extend($Ee, $Em) {
        $u1 = '1';
        $u2 = '0';
        $u3 = $Em;
        $v1 = '0';
        $v2 = '1';
        $v3 = $Ee;

        while (bccomp($v3, 0) != 0) {
            $qq = bcdiv($u3, $v3, 0);
            $t1 = bcsub($u1, bcmul($qq, $v1));
            $t2 = bcsub($u2, bcmul($qq, $v2));
            $t3 = bcsub($u3, bcmul($qq, $v3));
            $u1 = $v1;
            $u2 = $v2;
            $u3 = $v3;
```

```
            $v1 = $t1;
            $v2 = $t2;
            $v3 = $t3;
        }

        $vv = $u2;

        if (bccomp($vv, 0) == -1) {
            $inverse = bcadd($vv, $Em);
        } else {
            $inverse = $vv;
        }
        return $inverse;
    }

    private function GCD($e, $m) {
        $y = $e;
        $x = $m;

        while (bccomp($y, 0) != 0) {
            $w = bcsub($x, bcmul($y, bcdiv($x, $y, 0)));

            $x = $y;
            $y = $w;
        }
        return $x;
    }

    private function findE($m) {
        $e = '3';
        if (bccomp($this->GCD($e, $m), '1') != 0) {
            $e = '5';
            $step = '2';

            while (bccomp($this->GCD($e, $m), '1') != 0) {
                $e = bcadd($e, $step);

                if ($step == '2') {
                    $step = '4';
                } else {
                    $step = '2';
                }
            }
        }
        return $e;
    }

    public function encrypt($m, $s = 3) {
        $e = $this->keys[1];
        $n = $this->keys[0];

        $coded = '';
        $max = strlen($m);
        $packets = ceil($max / $s);
```

```
        for ($i = 0; $i < $packets; $i++) {
            $packet = substr($m, $i * $s, $s);
            $code = '0';

            for ($j = 0; $j < $s; $j++) {
                if (isset($packet[$j])) {
                    $code = bcadd($code, bcmul(ord($packet[$j]), bcpow('256', $j)));
                } else {
                    $code = bcadd($code, bcmul('0', bcpow('256', $j)));
                }
            }

            $code = bcpowmod($code, $e, $n);
            $coded .= $code . ' ';
        }
        return trim($coded);
    }

    public function decrypt($c) {
        $d = $this->keys[2];
        $n = $this->keys[0];

        $coded = split(' ', $c);
        $message = '';
        $max = count($coded);

        for ($i = 0; $i < $max; $i++) {
            $code = bcpowmod($coded[$i], $d, $n);

            while (bccomp($code, '0') != 0) {
                $ascii = bcmod($code, '256');
                $code = bcdiv($code, '256', 0);
                $message .= chr($ascii);
            }
        }
        return $message;
    }

    function coded($c) {
        $d = $this->keys[2];
        $n = $this->keys[0];
        $coded = split(' ', $c);

        $code = "";
        if (count($coded) > 0) {
            $code = bcpowmod($coded[0], $d, $n);
        }
        for ($i = 1, $max = count($coded); $i < $max; $i++) {
            $code .= " " . bcpowmod($coded[$i], $d, $n);
        }
        return $code;
    }
}
```

Penggambaran kode program di atas merupakan kode program *library* my_rsa yang digunakan untuk melakukan proses enkripsi menggunakan algoritma RSA. Tahapan pertama yakni melakukan *generate key* menggunakan *function* generate_keys() yang telah disediakan. Tahapan selanjutnya adalah melakukan enkripsi menggunakan *function* encrypt() yang telah disediakan di dalam *library*.

4.5 Pengujian Sistem

Pada penelitian ini peneliti menggunakan dua metode pengujian sistem yaitu *White Box Testing* dan *Black Box Testing* yang dapat dijelaskan di bawah ini:

4.5.1 *White Box Testing*

White Box Testing adalah pengujian yang dilakukan setelah sistem dibuat dengan melihat kode program, macam white box testing adalah listing program, grafik alir, kompleksitas siklomatis, pengujian jalur program dan test case. Pengujian yang akan dilakukan pada sistem pemesanan tiket adalah pengujian pada proses enkripsi, yang dijelaskan seperti berikut ini:

1. Pengujian *White Box* Sistem Pemesanan Tiket

Pengujian *white box* sistem pemesanan tiket adalah pengujian pada proses berjalannya algoritma kriptografi RSA, yang dapat dilihat di bawah ini:

(a) *Listing* Program *Library* my_rsa

1) *Listing* Program Fungsi Pembangkitan Kunci

```
public function generate_keys($p, $q, $show_debug = 0) {
    $n = bcmul($p, $q);

    //m (variable for calculate D and E)
    $m = bcmul(bcsub($p, 1), bcsub($q, 1));
    if (strpos($m, ".") > -1) {
        $xxx = split(".", $m);
        $m = $xxx[0];
    }

    // Public key E
    $e = $this->findE($m);
    // Private key D
    $d = $this->extend($e, $m);
    $keys = array($n, $e, $d);

    if ($show_debug) {
        echo "P = $p<br>Q = $q<br><b>N = $n</b> - modulo<br>M = $m<br><b>E = $e</b> - public key<br><b>D = $d</b> - p
    }

    $this->keys = $keys;//return $keys;
}
```

Gambar 4.9 Listing Program Fungsi Pembangkitan Kunci

2) Listing Program Fungsi Enkripsi

```
public function encrypt($m, $s = 3) {
    $e = $this->keys[1];
    $n = $this->keys[0];

    $coded = '';
    $max = strlen($m);
    $packets = ceil($max / $s);

    for ($i = 0; $i < $packets; $i++) {
        $packet = substr($m, $i * $s, $s);
        $code = '0';

        for ($j = 0; $j < $s; $j++) {
            if (isset($packet[$j])) {
                $code = bcadd($code, bcmul(ord($packet[$j]), bcpow('256', $j)));
            } else {
                $code = bcadd($code, bcmul('0', bcpow('256', $j)));
            }
        }

        $code = bcpowmod($code, $e, $n);
        $coded .= $code . ' ';
    }

    return trim($coded);
}
```

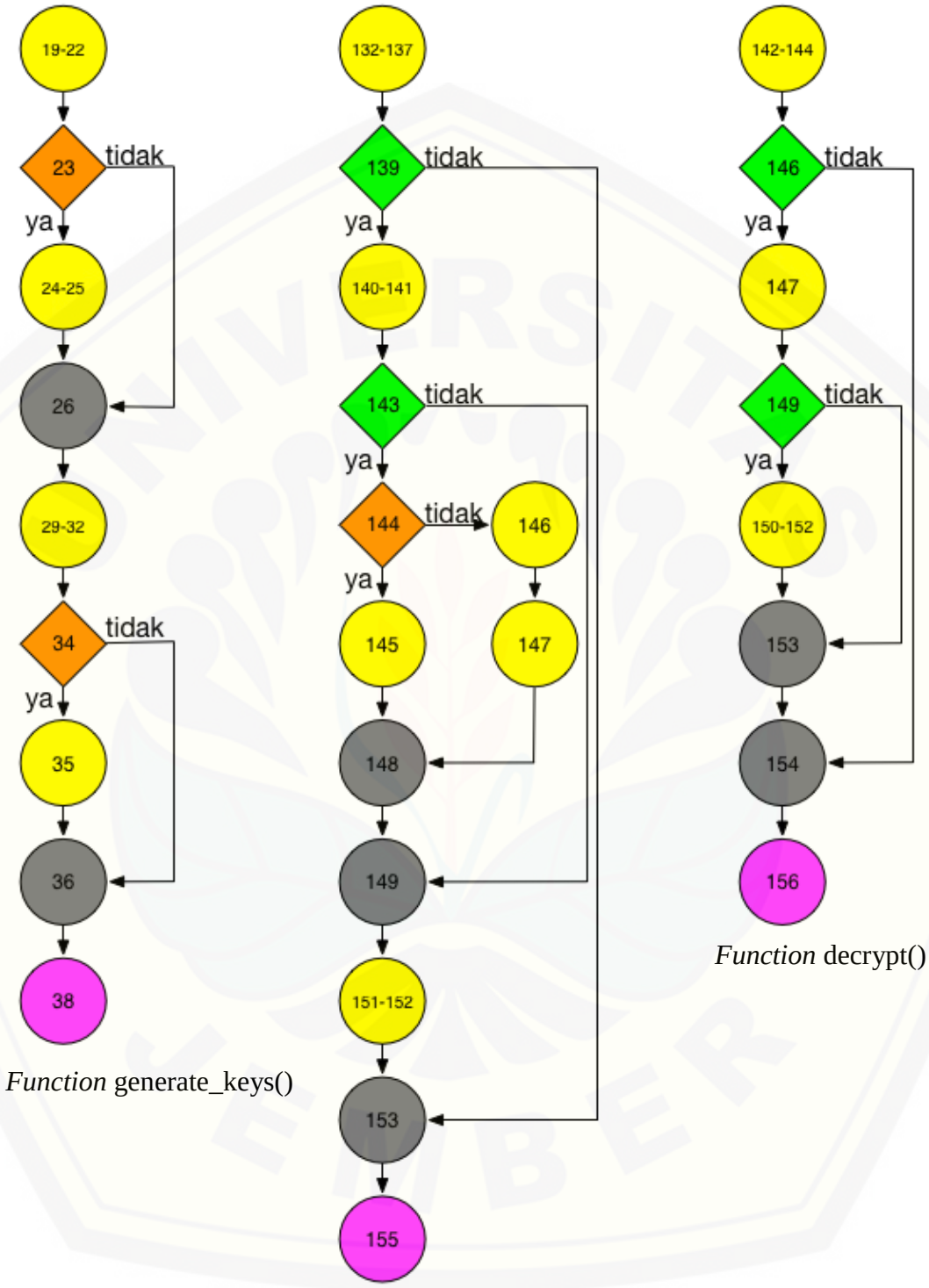
Gambar 4.10 Listing Program Fungsi Enkripsi

3) Listing Program Fungsi Dekripsi

```
public String decrypt(String c) {
    String cipher = new String(Base64.decode(c, 0));
    String[] coded = cipher.split(" ");
    String message = "";

    for(int i = 0, max = coded.length; i < max; i++) {
        String code = coded[i];

        while (bccomp(code, "0") != 0) {
            String ascii = bcmod(code, "256");
            code = bcdiv(code, "256", 0);
            message += (char) (Integer.parseInt(ascii));
        }
    }

    return message;
}
```

Gambar 4.11 Listing Program Fungsi Deskripsi

(b). Diagram Alir *Library* my_rsa



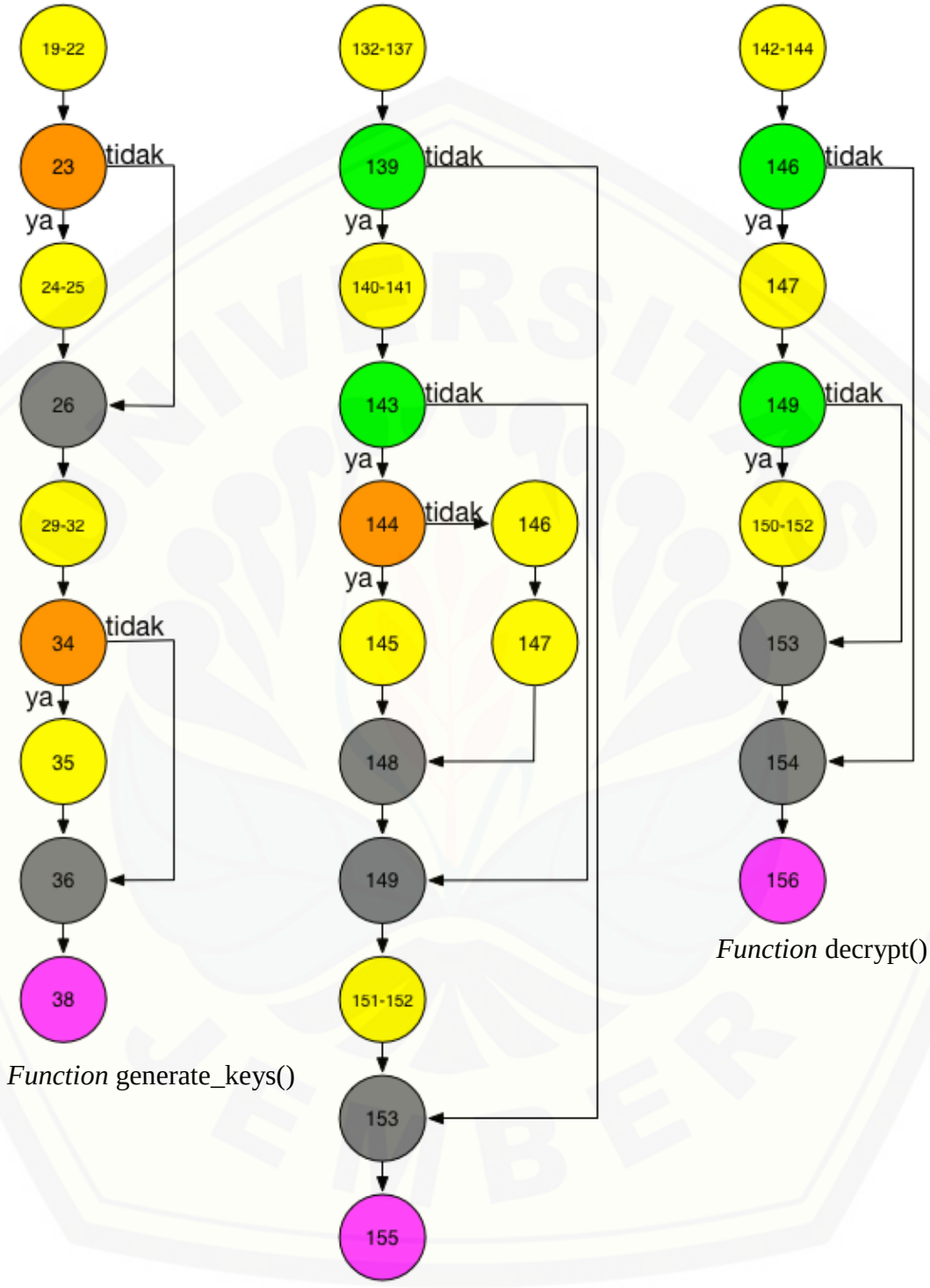
Function generate_keys()

Function encrypt()

Function decrypt()

Gambar 4.12 Diagram Alir *library* my_rsa

(c). Grafik Alir *Library* my_rsa



Function generate_keys()

Function encrypt()

Function decrypt()

Gambar 4.13 Grafik Alir *Library* my_rsa

(d). Komplekasitas Siklometik

Tahapan pengujian ini menggunkaan rumus V(G) = E – N + 2, dimana E adalah jumlah edge dan N adalah jumlah node grafik alir.

Function generate_keys() V(G) = E – N + 2 = 10 – 9 + 2 = 3

Function encrypt() V(G) = E – N + 2 = 15 – 13 + 2 = 4

Function decrypt() V(G) = E – N + 2 = 9 – 8 + 2 = 3

(e). Pengujian Jalur Program *Library* my_rsa

Pengujian jalur program *library* my_rsa berdasarkan gambar di atas adalah sebagai berikut:

1) *Function* generate_keys() :

Jalur 1: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

Jalur 2: 1 – 2 – 4 – 5 – 6 – 7 – 8 – 9

Jalur 3: 1 – 2 – 4 – 5 – 6 – 8 – 9

2) *Function* encrypt() :

Jalur 1: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11

Jalur 2: 1 – 2 – 3 – 4 – 5 – 12 – 13 – 7 – 8 – 9 – 10 – 11

Jalur 3: 1 – 2 – 3 – 4 – 8 – 9 – 10 – 11

Jalur 4: 1 – 2 – 10 – 11

3) *Function* decrypt() :

Jalur 1: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

Jalur 2: 1 – 2 – 3 – 4 – 6 – 7 – 8

Jalur 3: 1 – 2 – 7 – 8

(f). *Test Case Library* my_rsa

Tabel 4.7 *Test Case Library* my_rsa

Jalur 1	
Test Case	Jika nilai m merupakan bilangan desimal dan menampilkan hasil pembangkitan kunci
Target yang diharapkan	Nilai m dirubah menjadi bilangan bukan desimal dan

	menampilkan hasil pembangkitan kunci
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9
Jalur 2	
Test Case	Jika nilai m merupakan bukan desimal dan menampilkan hasil pembangkitan kunci
Target yang diharapkan	Menampilkan hasil pembangkitan kunci
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 4 – 5 – 6 – 7 – 8 – 9
Jalur 3	
Test Case	Jika nilai m merupakan bukan desimal dan tidak menampilkan hasil pembangkitan kunci
Target yang diharapkan	Melakukan pembangkitan kunci dan tidak menampilkan hasilnya
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 4 – 5 – 6 – 8 – 9
Jalur 1	
Test Case	Jika berhasil melakukan enkripsi
Target yang diharapkan	Memberi nilai balik berupa data yang telah dienkripsi
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11
Jalur 2	
Test Case	Jika ada paket yang tidak tersedia saat perulangan berlangsung
Target yang diharapkan	Memberi nilai “0” pada perhitungan lalu memberi nilai balik berupa data yang telah dienkripsi
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 12 – 13 – 7 – 8 – 9 – 10 – 11
Jalur 3	
Test Case	Jika nilai s adalah nol
Target yang diharapkan	Gagal melakukan enkripsi dan memberikan nilai balik berupa *string* kosong
Hasil Pengujian	Benar

Path/Jalur	1 – 2 – 3 – 4 – 8 – 9 – 10 – 11
Jalur 4	
Test Case	Jika tidak tidak ada data yang dienkripsi
Target yang diharapkan	Gagal melakukan enkripsi dan memberikan nilai balik berupa *string* kosong
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 10 – 11
Jalur 1	
Test Case	Jika berhasil melakukan dekripsi
Target yang diharapkan	Memberi nilai kembalian berupa string hasil dekripsi
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8
Jalur 2	
Test Case	Jika hasil perbandingan tidak sama dengan 0
Target yang diharapkan	Selesai melakukan perulangan kedua dan melakukan tahap selanjutnya
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 6 – 7 – 8
Jalur 3	
Test Case	Jika tidak ada data data yang akan didekripsi
Target yang diharapkan	Gagal melakukan dekripsi dan memberi nilai balik berupa *string* kosong
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 7 – 8

4.5.2 Black Box Testing

Tahapan pengujian black box dilakukan untuk menguji apakah kebutuhan yang dibutuhkan oleh user atau kebutuhan yang tertera pada kebutuhan fungsional sudah sesuai atau tidak, sehingga pengujian ini dilakukan pada setiap usecase yang berkaitan dengan pemesanan tiket dan proses unduh tiket yang dapat dilihat pada Tabel 4.8 di bawah ini:

Tabel 4.8 Hasil Pengujian *Black Box Download* Tiket

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Mendownload Tiket	Klik icon “cetak tiket”	Menampilkan halaman tiket pada *browser*	[√] Berhasil [ ] Gagal
		Menyimpan file tiket		[√] Berhasil [ ] Gagal

Tabel 4.8 adalah hasil pengujian yang dilakukan pada usecase download tiket. Hasil pengujian yang dilakukan oleh user membuktikan bahwa hasil program *download* tiket telah sesuai dengan kebutuhan yang diinginkan oleh user. Pengujian selanjutnya dapat dilihat pada Lampiran D.

BAB 5. HASIL DAN PEMBAHASAN

Hasil dan Pembahasan adalah tahapan dimana akan menggambarkan hasil dari pembangunan Aplikasi Verifikasi Tiket dengan menggunakan Algoritma Kriptografi RSA dan juga pembahasannya. Penjabaran hasil dan pembahasan ini nantinya akan bertujuan untuk menjelaskan bagaimana peneliti menyelesaikan perumusan masalah serta tujuan dan manfaat dari penelitian tentang pengamanan data menggunakan algoritma kriptografi.

5.1 Hasil Penerapan Algoritma Kriptografi RSA pada Proses Verivikasi Tiket

Hasil Penerapan Algoritma Kriptografi RSA pada proses verivikasi tiket bertujuan untuk memberikan kemudahan kepada PT. KAI, khususnya bagian pemeriksaan tiket sebagai user untuk melakukan verifikasi tiket kereta api.

Penerapan dengan menggunakan Algoritma Kriptografi RSA pada penelitian ini dilakukan untuk menganalisa dan menyelesaikan permasalahan tentang pengamanan data yang disisipkan pada *QR-Code* yang ada pada tiket. Adapun alur cara membuat *QR-Code* akan dijelaskan di bawah ini.

1. Analisis data

QR-Code mengkodekan teks dengan tipe *string*. *QR-Code* standard memiliki 4 mode teks: numerik, alfanumerik, alfabet, dan kanji. Setiap mode merupakan bagian dari bit string (0 dan 1), tetapi setiap mode memiliki cara yang berbeda dalam melakukan konfersi kedalam bit, dan setiap metode pengkodean dioptimalkan untuk menyandikan data dengan kemungkinan *string* yang terpendek. Oleh karena itu anda diharuskan melakukan analisis data untuk mengetaui jenis teks yang digunakan.

2. Pengkodean Data

Pengkodean data akan memberikan hasil berupa bit string yang dibagi menjadi kode kata yang masing-masing memliki panjang 8 bit.

3. Koreksi Kesalahan

Proses pembuatan *QR-Code* juga melakukan koreksi kesalahan yang berarti setelah membuat data bit string yang merepresentasikan teks, juga harus menggunakan bit string ini untuk membuat koreksi kesalahan dari codeword yang biasa disebut koreksi kesalahan "Reed Solomon". Alat pemindai *QR-Code* melakukan pembacaan terhadap bit string codeword dan koreksi kesalahan dari codeword. Pemindai selanjutnya melakukan perbandingan terhadap dua data tersebut sehingga pemindai dapat menyimpulkan apa sudah membaca data yang benar atau tidak, dan jika terjadi kesalahan pemindai dapat melakukan koreksi kesalahan sehingga dapat menemukan data string yang benar

4. Struktur Akhir Pesan

Data dan koreksi kesalahan dari kodeword yang sebelumnya telah dibuat, kini harus diurutkan dengan tepat. Bit string codeword dan koreksi kesalahan dari kodeword diletakkan dalam blok dan blok ini harus disisipkan sesuai dengan spesifikasi dari *QR-Code.*

5. Penempatan Modul Pada Matrix

Setelah mengatur blok pada urutan yang benar, data bit harus ditempatkan pada kode matrix. Codeword disusun pada matrix dengan cara tertentu. Pada tahapan ini juga akan dilakukan penempatan penanda persegi pada tiga sudut *QR-Code.*

6. Data Masking

Pola tertentu pada matrix *QR-Code* dapat mempersulit pemindai untuk membaca *QR-Code* dengan benar. Untuk mengatasi ini spesifikasi *QR-Code* membentuk delapan pola yang masing-masing merubah *QR-Code* menurut pola tertentu

7. Format dan Informasi Versi

Pada tahapan ini, dilakukan penambahan format serta informasi versi pada QR-Code

Peneliti juga melakukan percobaan dengan manual apakah perhitungan dengan algoritma ini layak atau tidak sebelum peneliti menggunakan dalam pembuatan sistem, yang perhitungannya dapat dilihat di bawah ini.

5.1.1 Membuat Dua Bilangan Prima Besar (p dan q)

Membuat contoh yang mudah, digunakan bilangan prima kecil, tetapi penggunaan bilangan ini tidak aman. Untuk mendapatkan bilangan prima acak, pertama dengan cara mengacak bilangan ganjil secara berurutan keatas sampai menemukan bilangan prima dengan nilai besar. Sebagai contoh p = 7 dan q = 19.

5.1.2 Menghitung nilai n

Nilai n didapat dari hasil perkalian dari kedua bilangan prima p dan q. Sehingga didapat nilai:

n = p * q
n = 7 * 19
n = 133

5.1.3 Menghitung nilai m

Nilai m didapat dari hasil perkalian (p-1) dan (q-1). Sehingga didapat nilai:
m = (p - 1) * (q - 1)
m = (7 – 1) * (19 – 1)
m = 6 * 18
m = 108

5.1.4 Menentukan nilai e yang relatif prima dengan m

Bilangan e relatif prima terhadap m berarti bilangan besar yang bisa dibagi dengan e maupun m (pembagi terbesar atau FPB) yang hasilnya sama dengan 1.

e = 2 => gcd(e, 108) = 2 (salah)

e = 3 => gcd(e, 108) = 3 (salah)

e = 4 => gcd(e, 108) = 4 (salah)

e = 5 => gcd(e, 108) = 1 (benar!)

5.1.5 Menentukan nilai d, sehingga (d * e) % m = 1

Hal ini berarti mencari nilai d sedemikian hingga (d * e) = 1 + (n * m) dimana n adalah sebarang bilangan bulat. Hal ini dapat ditulis dengan cara lain yakni d = (1 + (n * m)) / e.

d = 1 / 5 (salah)

n = 0

d = 109 / 5 (salah)

n = 1

d = 217 / 5 (salah)

n = 2

d = 325 / 5 = 65 (benar!)

n = 3

Hasil dari percobaan manual dengan menggunakan algoritma kriptografi RSA yang dapat dilihat pada pengerjaan sebelumnya menunjukan bahwa algoritma ini layak atau cocok digunakan dalam pembangunan aplikasi verifikasi tiket.

Kesimpulan yang dihasilkan dengan menggunakan algoritma kriptografi RSA ini pada proses pengamanan data pada *QR-Code* bahwa metode ini merupakan metode yang baik dan dapat melakukan pengamanan data dengan optimal dengan menggunakan waktu pengerjaan yang relatif cepat. Berdarkan kesimpulan tersebut maka, peneliti menyatakan bahwa algoritma ini layak untuk

menjadi algoritma dalam pembangunan aplikasi verifikasi tiket untuk pengamanan data pada QR-Code.

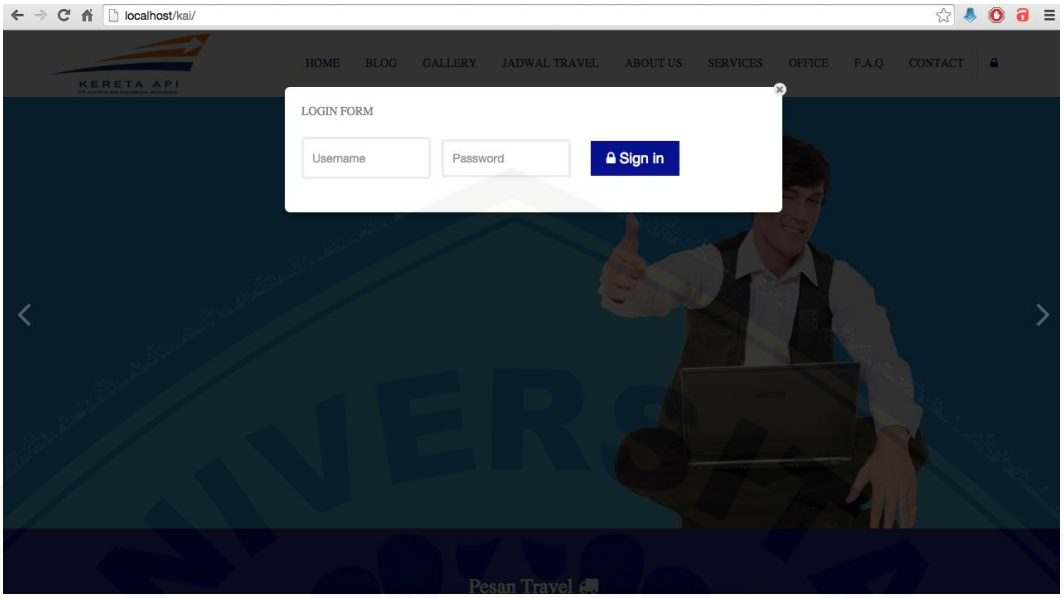
5.2 Hasil Pembuatan Aplikasi Verifikasi Tiket

Pembuatan aplikasi ini dibuat dengan menggunakan 2 basis sistem yang berbeda, yaitu berbasis web dan berbasis android. Sistem berbasis web mempunyai 4 jenis pengguna yang dapat mengakses fitur – fitur tertentu sesuai wewenang yang dimilikinya. Sistem berbasis android hanya memiliki 1 pengguna yang dapat mengakses fitur yang tersedia.

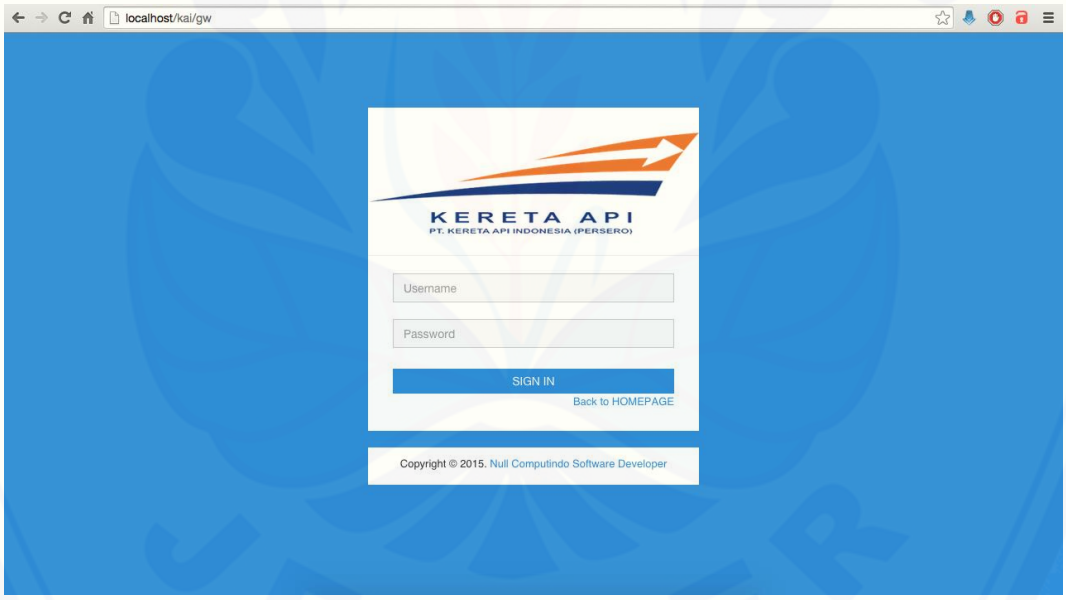
Pengguna pada sistem berbasis web yaitu sebagai *Super Admin*, *Blog Administrator*, *Member*, dan *Public*. *Super Admin* yakni bagian yang bertugas melakukan pembuatan jadwal serta pengisian saldo member. *Blog Administrator* yakni bagian yang bertugas mengelola blog sebagai halaman muka untuk pengunjung *website*. *Member* yakni yang pengguna yang dapat melakukan pembelian tiket, dalam hal ini bertindak seperti agen. *Puclic* yakni bagian pengunjung umum website yang juga dapat berarti calon penumpang yang ingin mendownload tiket yang dibeli. Pengguna pada sistem berbasis android adalah petugas pada peron yang ada di stasiun yang bertugas memeriksa tiket menggunakan aplikasi ini yang telah diinstall pada perangkat android.

5.2.1 Fitur *Login*

Sistem pemesanan tiket dibuat untuk pengguna agar dapat masuk menuju halaman pengguna sesuai hak akses dengan login terlebih dahulu. Fitur ini dibuat agar pengguna dapat mengelola data sesuai wewenangnya. Fitur keamanan dengan halaman login untuk masuk memiliki username dan password, yamg dapat dilihat pada Gambar 5.1 dan Gambar 5.2 di bawah ini.



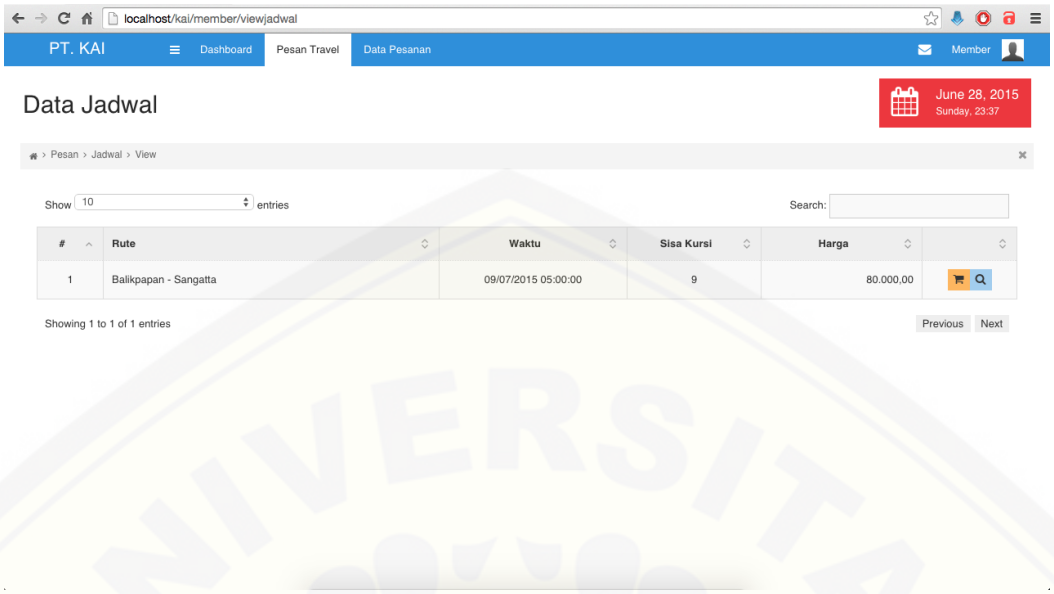
Gambar 5.1 Tampilan Halaman Login (menggunakan modal)



Gambar 5.2 Tampilan Halaman Login (mengakses url login)

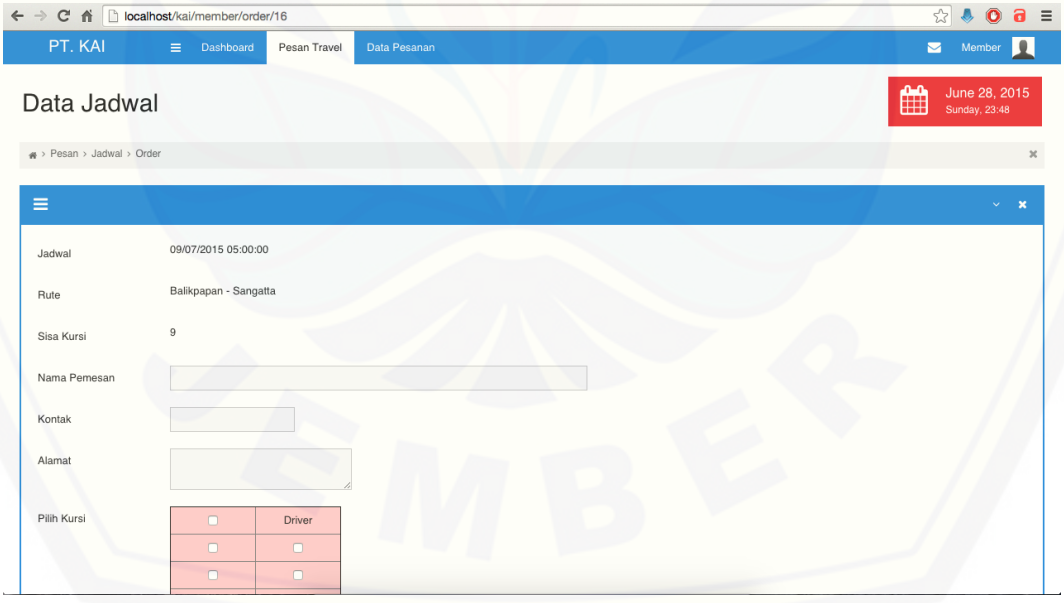
5.2.2 Halaman Memesan Tiket

Member yang masuk ke halaman pemesanan tiket, akan diberikan tampilan awal berupa jadwal pemberangkatan seperti pada Gambar 5.3.

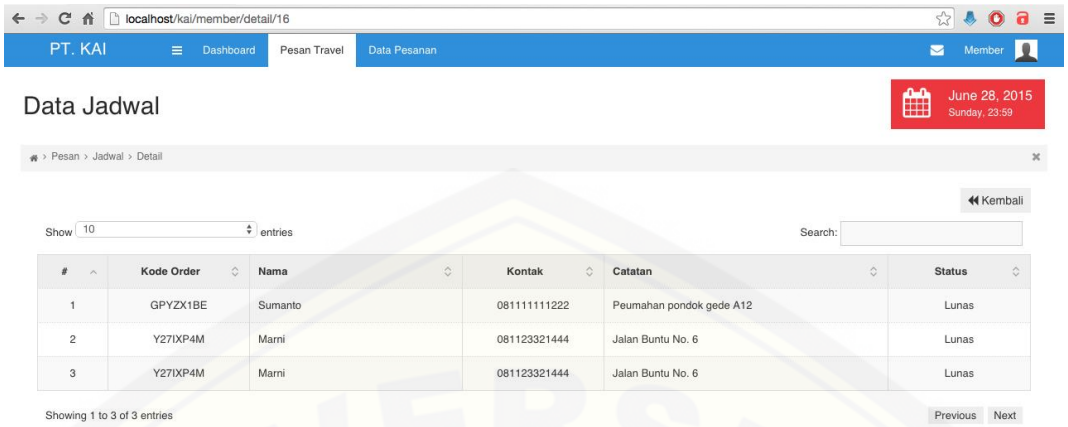


Gambar 5.3 Tampilan Halaman Jadwal Pemberangkatan

Setelah sistem menampilkan halaman jadwal, member dapat melakukan pemesanan dan melihat detail pesanan pada tabel jadwal. Form pemesanan dapat dilihat pada Gambar 5.4 dan detail pesanan dapat dilihat pada Gambar 5.5.



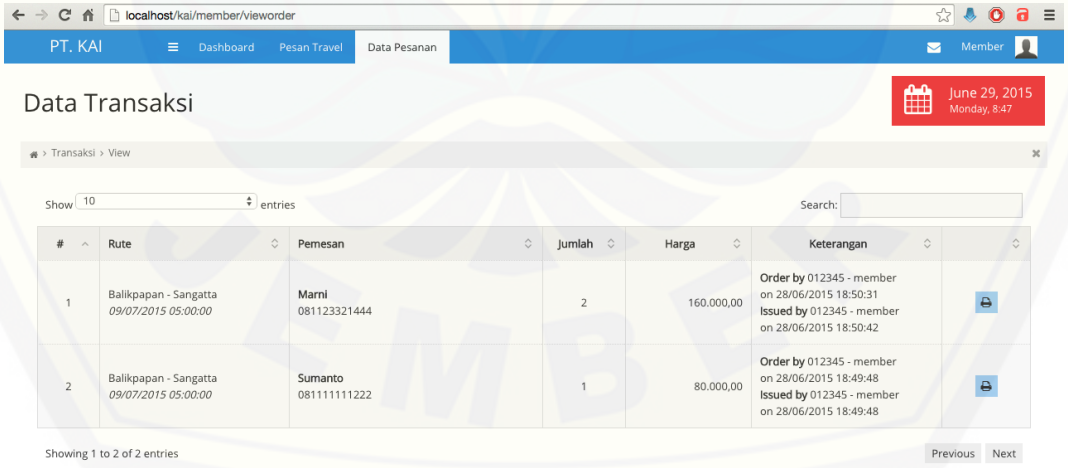
Gambar 5.4 Tampilan Halaman Form Pemesanan



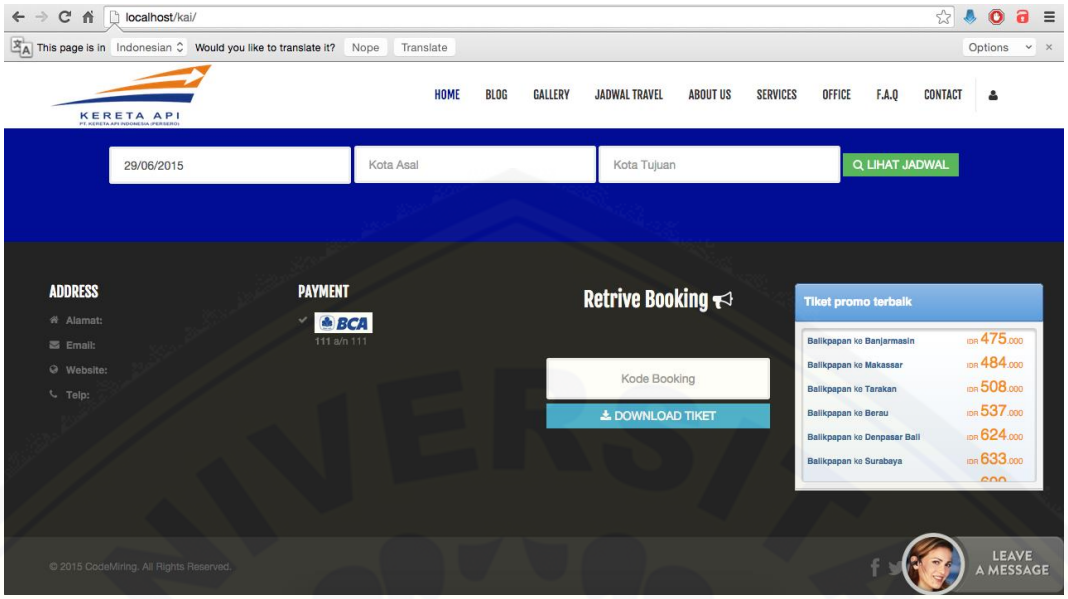
Gambar 5.5 Tampilan Halaman Detail Pemesanan

5.2.3 Fitur Mendownload Tiket

Member dapat mengakses Detail Pemesanan untuk melakukan pengunduhan tiket yang telah dibeli. Member atau pengguna umum juga dapat mendownload tiket dengan cara mengakss halaman home dengan memasukkan kode *booking*. Halaman untuk mengunduh tiket dapat dilihat pada Gambar 5.6 dan Gambar 5.7.

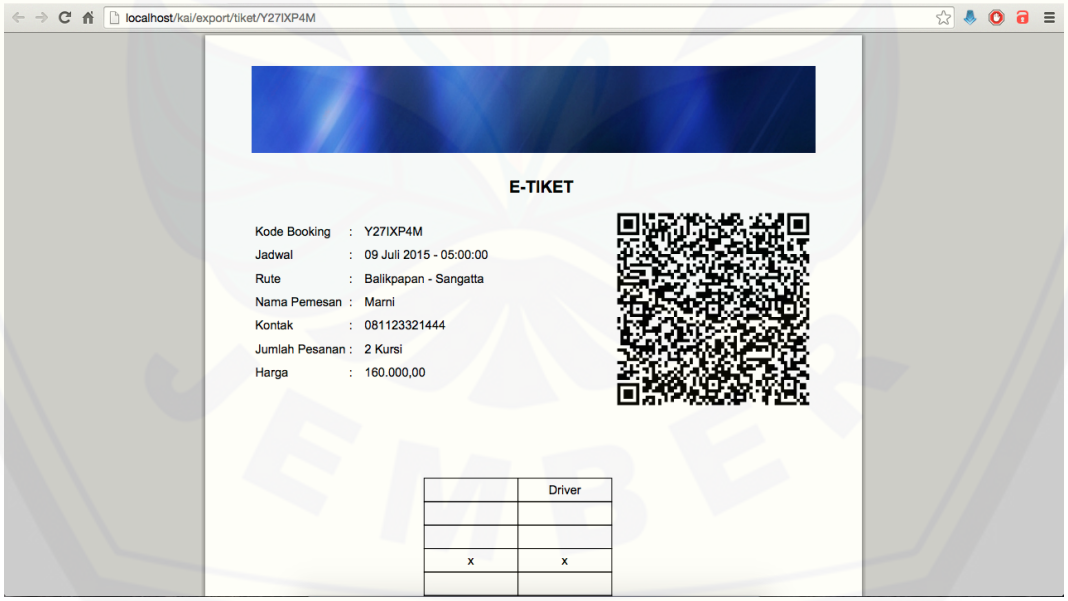


Gambar 5.6 Tampilan Halaman Unduh Tiket (login sebagai member)



Gambar 5.7 Tampilan Halaman Unduh Tiket (tanpa login ke dalam sistem)

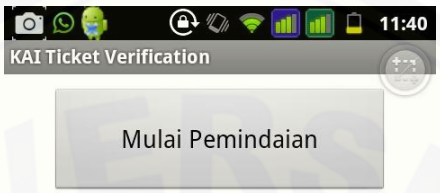
Setelah pengguna melakukan pengunduhan tiket sistem akan menampilkan halaman tiket berupa PDF yang dapat disimpan dengan cara dicetak atau dengan di foto. Gambar tampilan tiket dapat dilihat pada gambar 5.8.



Gambar 5.8 Tampilan Halaman Tiket

5.2.4 Fitur Verifikasi Tiket

Fitur Verifikasi Tiket terdapat pada sistem berbasis android. Pengguna yang dapat mengakses fitur ini hanya petugas kereta api pada peron. Tampilan Awal fitur ini dapat dilihat pada Gambar 5.9 dan tampilan pemindaian dapat dilihat pada Gambar 5.10.



Gambar 5.9 Tampilan Halaman Awal Fitur Verivikasi Tiket

Gambar 5.10 Tampilan Halaman Pemindaian Tiket

Saat petugas melakukan pemindaian sistem mendekripsi *QR-Code* lalu menampilkan hasil pada halaman awal. Gambar aplikasi saat menampilkan data *QR-Code* dapat dilihat pada Gambar 5.11.

Gambar 5.11 Tampilan Halaman Hasil Pemindaian Tiket

Fitur tambahan selain di atas dapat dilihat pada lampiran F.

5.3 Hasil Penerapan Algoritma Algoritma Kriptografi RSA Pada Aplikasi

Penerapan Algoritma kriptografi RSA pada aplikasi verifikasi tiket berbasis android terletak pada fitur download tiket pada sistem berbasis web dan verifikasi tiket pada sistem berbasis android. Tahap pertama penerapan metode ini adalah melakukan download tiket pada sistem berbasis web, selanjutnya adalah sistem menjalankan *controller* export yang berfungsi menampilkan tiket. Dalam proses penampilan tiket data penumpang dienkripsi menggunakan algoritma kriptografi ini. Langkah awal yang dilakukan saat proses enkripsi adalah pembangkitan kunci dari 2 bilangan prima bernilai besar. Selanjutnya data dienkripsi dengan kunci *public* lalu data hasil enkripsi diubah oleh library menjadi bentuk *QR-Code*

selanjutnya *QR-Code* yang telah dienkripsi diletakkan pada tiket yang tertera pada PDF yang ditampilkan oleh sistem berbasis web.

Pengguna yang dalam hal ini sebagai penumpang menyimpan tiket dan pergi ke stasiun tempat pemberangkatan lalu menunjukkan tiket yang disimpan kepada petugas peron. Selanjutnya proses verifikasi dimulai dengan melakukan pemindaian *QR-Code* pada tiket penumpang, lalu sistem melakukan dekripsi berdasarkan data yang ada pada *QR-Code*, selanjutnya sistem menampilkan hasil dekripsi berupa data penumpang seperti yang telah di jelaskan pada subbab sebelumnya.

Penjelasan kode program untuk melakukan pembangkitan kunci, enkripsi, dan dekripsi akan dijelaskan lebih mendetail di bawah ini.

5.3.1 Pembangkitan Kunci

Tahap ini melakukan pembangkitan kunci menggunakan dua bilangan prima bernilai besar. Sistem melakukan pembangkitan kunci dengan dua bilangan prima, yaitu 9990454949 dan 9990450271. Langkah pertama pada tahapan ini adalah mencari nilai m, lalu mencari nilai kunci *public*, dan yang terakhir adalah mencari kunci *private*. Kode program untuk melakukan pembangkitan kunci dapat dilihat pada Gambar 5.12.

```
public function generate_keys($p, $q, $show_debug = 0) {
    $n = bcmul($p, $q);

    //m (variable for calculate D and E)
    $m = bcmul(bcsub($p, 1), bcsub($q, 1));
    if (strpos($m, ".") > -1) {
        $xxx = split(".", $m);
        $m = $xxx[0];
    }

    // Public key E
    $e = $this->findE($m);
    // Private key D
    $d = $this->extend($e, $m);
    $keys = array($n, $e, $d);

    if ($show_debug) {
        echo "P = $p<br>Q = $q<br><b>N = $n</b> - modulo<br>M = $m<br><b>E = $e</b> - public key<br><b>D = $d</b> - priva
    }

    $this->keys = $keys;//return $keys;
}
```

Gambar 5.12 Kode program pembangkitan kunci

5.3.2 Proses Enkripsi

Tahap ini adalah tahap dimana melakukan proses enkripsi setelah melakukan pembangkitan kunci. Enkripsi dilakukan menggunakan kunci *public*. Kode program untuk melakukan enkripsi dapat dilihat pada Gambar 5.13.

```php
public function encrypt($m, $s = 3) {
    $e = $this->keys[1];
    $n = $this->keys[0];

    $coded = '';
    $max = strlen($m);
    $packets = ceil($max / $s);

    for ($i = 0; $i < $packets; $i++) {
        $packet = substr($m, $i * $s, $s);
        $code = '0';

        for ($j = 0; $j < $s; $j++) {
            if (isset($packet[$j])) {
                $code = bcadd($code, bcmul(ord($packet[$j]), bcpow('256', $j)));
            } else {
                $code = bcadd($code, bcmul('0', bcpow('256', $j)));
            }
        }

        $code = bcpowmod($code, $e, $n);
        $coded .= $code . ' ';
    }

    return trim($coded);
}
```

Gambar 5.13 Kode program enkripsi

5.3.3 Proses Dekripsi

Tahap ini adalah tahap dimana melakukan proses dekripsi, tahapan ini berada pada sistem berbasis android yang digunakan oleh petugas. Dekripsi dilakukan menggunakan kunci *private*. Kode program untuk melakukan dekripsi dapat dilihat pada Gambar 5.14.

```java
public String decrypt(String c) {
    String cipher = new String(Base64.decode(c, 0));
    String[] coded = cipher.split(" ");
    String message = "";

    for(int i = 0, max = coded.length; i < max; i++) {
        String code = coded[i];

        while (bccomp(code, "0") != 0) {
            String ascii = bcmod(code, "256");
            code = bcdiv(code, "256", 0);
            message += (char) (Integer.parseInt(ascii));
        }
    }

    return message;
}
```

Gambar 5.14 Kode program dekripsi

5.4 Pembahasan Sistem

Rancang bangun aplikasi verifikasi pemesanan tiket dengan QR-Code berbasis Android menggunakan algoritma kriptografi asimetris (RSA) memiliki beberapa manfaat. Beberapa manfaat dari sistem ini adalah:

1. Sistem dapat menyimpan dan memanejemen (*create,update,delete*) seluruh data user.
2. Sistem dapat menyimpan dan memanejemen (*create,update,copy*) seluruh data jadwal keberangkatan.
3. Sistem dapat menyimpan dan memanejemen (*create,issued,print*) seluruh data transaksi pemesanan tiket.
4. Sistem dapat menyimpan dan memanejemen (*create,update*) seluruh data kota yang dilewati rute kereta api.
5. Sistem dapat menyimpan dan memanejemen (*create,update*) seluruh data member.
6. Sistem dapat menyimpan dan memanejemen (*search,print*) seluruh data laporan dari pemesanan tiket.
7. Sistem dapat menyimpan dan memanejemen (*search*) seluruh data saldo terbaru.
8. Sistem dapat menyimpan dan memanejemen (s*earch*) seluruh data laporan penjualan tiket.
9. Sistem dapat mendownload tiket yang telah dipesan menggunakan kode *booking* yang telah diterima.
10. Terdapat algoritma pengamanan data menggunakan kriptografi saat penyisipan data dalam proses pembuatan tiket.
11. Terdapat beberapa member utama sebagai aktor sistem, yaitu super admin, pengelola blog, dan member.

Sistem dibuat sesuai kebutuhan instansi dan belum ada kekurangan dari sistem ini.

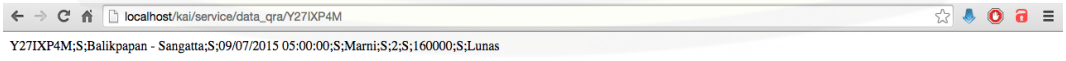
5.4.1 Fitur pada sistem

Fitur-fitur yang terdapat di dalam rancang bangun aplikasi verifikasi pemesanan tiket dengan QR-Code berbasis Android menggunakan algoritma kriptografi asimetris (RSA) sesuai dengan kebutuhan user yang berupa data user, data jadwal, managemen transaksi, data kota, data member, data saldo, pemesanan travel, managemen data pemesanan, managemen data penumpang, laporan penjualan, laporan saldo penumpang, data kategori, data artikel, data galery, data partner, data profil, data akun pembayaran, data kantor, data F.A.Q, data pembayaran, data link, data slide, data jadwal travel, dan data tiket. Admin dapat memanajemen fitur-fitur tersebut namun user hanya dapat melihat detail pemesanan. Selain itu user juga dapat mencetak data penumpang dan data jadwal seta laporan bulanan.

Dari hasil implementasi sistem didapatkan bahwa output dari rancang bangun aplikasi verifikasi pemesanan tiket dengan QR-Code berbasis Android menggunakan algoritma kriptografi asimetris (RSA) beserta dengan fitur-fiturnya dapat berjalan dengan baik dan *user friendly*.

5.5 Hasil Perbandingan Isi Data Tanpa Menggunakan Sistem dan Menggunakan Sistem

Pembuatan sistem ini bertujuan untuk mengamankan data penumpang dan menghindari dari kasus pembajakan atau penggandaan tiket secara ilegal. Dengan penggunaan data ini diharap data penumpang menjadi lebih aman dan pihak yang tidak bertanggung jawab tidak dapat melakukan penggandaan tiket sehingga jelas terdapat perbedaan isi data antara sistem yang tidak menggunakan algoritma kriptografi dan sistem yang menggunakan kriptografi RSA ini. Isi data *QR-Code* tanpa menggunaan algoritma akan ditunjukkan pada Gambar 5.15 dan isi data *QR-Code* dengan menggunakan algoritma akan ditunjukkan pada gambar 5.16.



Gambar 5.15 Isi Data Tanpa Menggunakan Algoritma

Gambar 5.16 Isi Data Dengan Menggunakan Algoritma

BAB 6. PENUTUP

Penutup menggambarkan tentang kesimpulan dari seluruh sistem yang telah dibangun oleh peneliti, dan diharapkan nantinya dari kesimpulan dan saran yang diberikan akan digunakan sebagai acuan untuk digunakan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah:

1. Aplikasi sistem verifikasi pemesanan tiket pada PT. KAI telah dirancang dan dibangun dengan mempunyai 4 hak akses dan berbagai fitur. Hak akses pertama adalah hak akses untuk super admin yang dapat mengakses login, manajemen user, manajemen data jadwal, manajemen data transaksi, manajemen data kota, manajemen data member, manajemen data laporan, manajemen data saldo. Hak akses kedua adalah Member yang dapat mengakses login, manajemen pesan travel, manajemen data pesananan, manajemen penumpang, manajemen laporan penjualan, manajemen saldo penjualan. Hak akses ketiga adalah blog Administrator yang dapat mengakses login, manajemen data kategori, manajemen data artikel, manajemen data galery, manajemen data partner, manajemen data profil, manajemen data akun pembayaran, manajemen data kantor, manajemen data F.A.Q, manajemen data link, manajemen data slide. Hak akses yang terakhir adalah publik yan dapat mengakses melihat jadwal travel dan mendownload tiket.
2. Pengamanan data untuk pemesanan tiket dengan menggunakan algoritma kriptografi RSA adalah dengan cara enkripsi dan dekripsi. Langkah dalam proses enkripsi data yakni melakukan pembangkitan kunci menggunakan dua bilangan prima bernilai besar lalu menjalankan proses enkripsi data. Langkah dalam proses dekripsi data yakni melakukan pembangkitan kunci

menggunakan dua bilangan prima bernilai besar yang sama dengan pada saat proses enkripsi lalu menjalankan proses dekripsi data.

3. Aplikasi verifikasi pemesanan tiket berbasis android ini dirancang dan dibangun dengan melalui beberapa tahapan yakni menganalisa data jadwal, data calon penumpang dan data pemberangkatan yang nantinya akan dibutuhkan untuk melakukan sebuah proses pengamanan data menggunakan algoritma kriptografi RSA.

6.2 Saran

Saran dan masukan berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu :

1. Diperlukan adanya perbandingan sistem yang dibangun dengan metode lain.
2. Membuat sistem pemesanan tiket juga dapat dilakukan melalui android.

DAFTAR PUSTAKA

Abhishek Gandhi, B. S. (2014). Advanced Online Banking Authentication System Using One Time Passwords Embedded in Q-R Code. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 1327-1329.

Andri, M. Y. (2010, September 6). Implementasi Algoritma Kriptografi DES, RSA dan Algoritma Kompresi LZW pada Berkas Digital. Retrieved from http://repository.usu.ac.id/bitstream/123456789/7843/1/10E00140.pdf

Damardono, H. (2012, August 14). *Tiket Kereta Api Kini Bisa Dipesan via Internet*. Retrieved April 23, 2015, from kompas.com: http://tekno.kompas.com/read/2012/08/14/12180837/tiket.kereta.api.kini.bisa.dipesan.via.internet

Hartini, S. P. (2014, March). Kriptografi Password menggunakan Modifikasi Metode Affine Ciphers. *JURNAL SIGMATA | LPPM AMIK SIGMA*, 40-50.

Johnsonbaugh, R. (1998). *Matematika Diskrit Edisi 4 Jilid 1*. Jakarta: PT Prenhalliondo.

Nidhra, S., & Dondeti, J. (2012). Black Box And White Box Testing Techniques –A Literature Review. *International Journal of Embedded Systems and Applications (IJESA)*, 29-50.

Nivedita Bisht, S. S. (2015, March). A Comparative Study of Some Symmetric and Asymmetric Key Cryptography Algorithms. *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, 1028-1031.

Pressman, R. S. (1998). *Software Engineering : A Practioner's Approach.* McGraw Hill.

Rahajoeningroem, T., & Aria, M. (2009). Studi dan Implementasi Algoritma RSA untuk Pengamanan dan Transkrip Akademik Mahasiswa. *Majalah Ilmiah UNIKOM, 8, No.1*. Retrieved from http://jurnal.unikom.ac.id/_s/data/jurnal/v08-n01/volume-81-artikel-9.pdf/pdf/volume-81-artikel-9.pdf

Sommerville, I. (2001). *Software Engineering.* Addison Wesley.

LAMPIRAN

LAMPIRAN A (*Usecase* Skenario)

A.1 Usecase Skenario memanajemen data user

Nama Usecase	Memanajemen data user
Aktor	Super Admin
Pre Kondisi	Super Admin telah melakukan login
Post Kondisi	Super Admin memanajemen data user

SKENARIO MANAJEMEN DATA USER	
NORMAL SKENARIO MENAMBAH DATA USER	
1. Klik menu "Data User"	
	2. Menampilkan tabel data user disertai tombol "Tambah Data", "*edit*" serta "hapus".
3. Klik tombol "Tambah Data".	
	4. Menampilkan form data user.
5. Mengisi form.	
6. Klik tombol "Submit"	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman "Data User."
ALTERNATIF SKENARIO KLIK TOMBOL SUBMIT KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	8.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong atau belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	8.b. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENG*EDIT* DATA USER	
1. Klik menu "Data User"	
	2. Menampilkan tabel data user disertai tombol "Tambah Data", "*edit*" serta "hapus".
3. Klik tombol "pensil" yang menandakan "*Edit*"	
4. Mengubah data pada kolom yang dituju.	

5. Klik tombol “Submit”	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman “Data User”.
ALTERNATIF SKENARIO KLIK TOMBOL SUBMIT KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENGHAPUS DATA USER	
1. Klik menu “Data User”	
2. Klik tombol “*trash*” yang menandakan “*Delete*”	
	3. Menampilkan pesan “apakah anda akan menghapus data?”
4. Klik tombol “ya”.	
	5. Menghapus data.
	6. Menampilkan halaman data user.
ALTERNATIF SKENARIO SAAT TIDAK JADI MENGHAPUS DATA	
4. Klik tombol “tidak”.	
	5. Menampilkan halaman data user.

A.2 Usecase Skenario memanajemen data jadwal

Nama Usecase	Memanajemen data jadwal
Aktor	Super Admin
Pre Kondisi	Super Admin telah melakukan login
Post Kondisi	Super Admin memanajemen data jadwal

SKENARIO MANAJEMEN DATA JADWAL	
NORMAL SKENARIO MENAMBAH DATA JADWAL	
1. Klik menu “Data Jadwal”	
	2. Menampilkan tabel data jadwal disertai tombol “Tambah”, dan “copy, *edit*, lihat denah”.
3. Klik tombol “Tambah Data”.	
	4. Menampilkan form data jadwal.
5. Mengisi form.	
6. Klik tombol “Submit”	

	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data jadwal.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	8.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	8.b. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MELIHAT DENAH TEMPAT DUDUK	
1. Klik menu “manajemen data jadwal”	
	2. Menampilkan tabel data jadwal disertai tombol “Tambah”, dan “copy, *edit*, lihat denah”.
3. Klik tombol “lihat denah”.	
	4. Menampilkan denah tempat duduk.
	5. Menampilkan halaman data jadwal.
NORMAL SKENARIO MENG*EDIT* DATA JADWAL	
1. Klik menu “manajemen data jadwal”	
	2. Menampilkan tabel data jadwal disertai tombol “Tambah”, dan “copy, *edit*, lihat denah”.
3. Klik tombol “pensil” yang menandakan “*Edit*”	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol “Submit”	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data jadwal.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	

	9. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format
NORMAL SKENARIO COPY JADWAL	
1. Klik menu "manajemen data jadwal"	
	2. Menampilkan tabel data jadwal disertai tombol "Tambah", dan "copy, *edit*, lihat denah".
3. Klik tombol "copy"	
	4. Menampilkan form jadwal
5. Klik tombol "submit".	
	6. Mengcopy data jadwal.
	7. Menampilkan halaman data jadwal.
ALTERNATIF SKENARIO SAAT TIDAK JADI MENGHAPUS DATA	
4. Klik tombol "back".	
	5. Menampilkan halaman data jadwal.

A.3 Usecase Skenario memanajemen data kota

Nama Usecase	Memanajemen data kota
Aktor	Super Admin
Pre Kondisi	Super Admin telah melakukan login
Post Kondisi	Super Admin memanajemen data kota

SKENARIO MANAJEMEN DATA KOTA	
NORMAL SKENARIO MENAMBAH DATA KOTA	
1. Klik menu "manajemen data kota"	
	2. Menampilkan tabel data kota disertai tombol "Tambah" dan "*edit*".
3. Klik tombol "tambah".	
	4. Menampilkan form data kota.
5. Mengisi form.	
6. Klik tombol "Submit"	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data kota.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	8. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	

	8. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENG*EDIT* DATA KOTA	
1. Klik menu “manajemen data kota”	
	2. Menampilkan tabel data kota disertai tombol “Tambah” dan “*edit*”.
3. Klik tombol “pensil” yang menandakan “*Edit*”	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol “Submit”	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data kota.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	8. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format

A.4 Usecase Skenario memanajemen data member

Nama Usecase	Memanajemen data member
Aktor	Super Admin
Pre Kondisi	Super Admin telah melakukan login
Post Kondisi	Super Admin memanajemen data member

SKENARIO MANAJEMEN DATA MEMBER	
NORMAL SKENARIO MENAMBAH DATA MEMBER	
1. Klik menu “manajemen data member”	
	2. Menampilkan tabel data member disertai tombol “Tambah”, dan “tambah jaringan, *edit*, penambahan saldo”.
3. Klik tombol “tambah”.	
	4. Menampilkan form data member.
5. Mengisi form.	

6. Klik tombol "Submit"	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data member.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENG*EDIT* DATA MEMBER	
1. Klik menu "manajemen data member"	
	2. Menampilkan tabel data member disertai tombol "Tambah", dan "tambah jaringan, *edit*, penambahan saldo".
3. Klik tombol "pensil" yang menandakan "*Edit*"	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol "Submit"	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data member.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format
NORMAL SKENARIO TAMBAH JARINGAN	
1. Klik menu "manajemen data member"	
	2. Menampilkan tabel data member disertai tombol "Tambah", dan "tambah jaringan, edit,

	penambahan saldo".
3. Klik tombol "tambah jaringan".	
	4. Menampilkan form data member.
5. Klik tombol "Submit"	
	6. Memeriksa data
	7. Menyimpan data
	8. Menampilkan halaman data member.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO TAMBAH SALDO	
1. Klik menu "manajemen data member"	
	2. Menampilkan tabel data member disertai tombol "Tambah", dan "tambah jaringan, edit, penambahan saldo".
3. Klik tombol "tambah saldo".	
	4. Menampilkan form data member.
5. Klik tombol "Submit"	
	6. Memeriksa data
	7. Menyimpan data
	8. Menampilkan halaman data member.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format

A.5 Usecase Skenario memanajemen data laporan

Nama Usecase	Memanajemen data laporan
Aktor	Super Admin

Pre Kondisi	Super Admin telah melakukan login
Post Kondisi	Super Admin memanajemen data laporan

SKENARIO MANAJEMEN DATA LAPORAN	
NORMAL SKENARIO MELIHAT DETAIL DATA LAPORAN	
1. Klik menu “manajemen data laporan”	
	2. Menampilkan tabel data laporan disertai tombol “Filter”.
3. Klik tombol “filter”.	
	4. Menampilkan halaman data laporan.

A.6 Usecase Skenario memanajemen data saldo

Nama Usecase	Memanajemen data saldo
Aktor	Super Admin
Pre Kondisi	Super Admin telah melakukan login
Post Kondisi	Super Admin memanajemen data saldo

SKENARIO MANAJEMEN DATA SALDO	
NORMAL SKENARIO MELIHAT DETAIL DATA SALDO	
1. Klik menu “manajemen data saldo”	
	2. Menampilkan tabel data saldo disertai tombol “detail”.
3. Klik tombol “detail”.	
	4. Menampilkan halaman data saldo.

A.7 Usecase Skenario memanajemen data kategori

Nama Usecase	Memanajemen data kategori
Aktor	Blog Administrator
Pre Kondisi	Blog Administrator telah melakukan login
Post Kondisi	Blog Administrator memanajemen data kategori

SKENARIO MANAJEMEN DATA KATEGORI	
NORMAL SKENARIO MENAMBAH DATA KATEGORI	
1. Klik menu “manajemen data kategori”	
	2. Menampilkan tabel data kategori disertai tombol “Tambah”, dan

	"*edit*".
3. Klik tombol "tambah".	
	4. Menampilkan form data kategori.
5. Mengisi form.	
6. Klik tombol "Submit"	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data kategori.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENG*EDIT* DATA KATEGORI	
1. Klik menu "manajemen data kategori"	
	2. Menampilkan tabel data kategori disertai tombol "Tambah", dan "*edit*".
3. Klik tombol "pensil" yang menandakan "*Edit*"	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol "Submit"	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data kategori.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format

A.8 Usecase Skenario memanajemen data artikel

Nama Usecase	Memanajemen data artikel
Aktor	Blog Administrator
Pre Kondisi	Blog Administrator telah melakukan login
Post Kondisi	Blog Administrator memanajemen data artikel

SKENARIO MANAJEMEN DATA ARTIKEL	
NORMAL SKENARIO MENAMBAH DATA ARTIKEL	
1. Klik menu "manajemen data artikel"	
	2. Menampilkan tabel data artikel disertai tombol "Tambah", dan "*edit*".
3. Klik tombol "tambah".	
	4. Menampilkan form data artikel.
5. Mengisi form.	
6. Klik tombol "Submit"	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data artikel.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENG*EDIT* DATA ARTIKEL	
1. Klik menu "manajemen data artikel"	
	2. Menampilkan tabel data artikel disertai tombol "Tambah", dan "*edit*".
3. Klik tombol "pensil" yang menandakan "*Edit*"	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol "Submit"	
	6. Memeriksa data.

	7. Menyimpan data.
	8. Menampilkan halaman data artikel.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format

A.9 Usecase Skenario memanajemen data gallery

Nama Usecase	Memanajemen data galery
Aktor	Blog Administrator
Pre Kondisi	Blog Administrator telah melakukan login
Post Kondisi	Blog Administrator memanajemen data galery

SKENARIO MANAJEMEN DATA GALERY	
NORMAL SKENARIO MENAMBAH DATA GALERY	
1. Klik menu “manajemen data galery”	
	2. Menampilkan tabel data galery disertai tombol “Tambah”, dan “*edit*”.
3. Klik tombol “tambah”.	
	4. Menampilkan form data galery.
5. Mengisi form.	
6. Klik tombol “Submit”	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data galery.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	8. Menampilkan pesan bahwa data yang tekah diinputkan tidak

	sesuai format
NORMAL SKENARIO MENG*EDIT* DATA GALERY	
1. Klik menu “manajemen data galery”	
	2. Menampilkan tabel data galery disertai tombol “Tambah”, dan “*edit*”.
3. Klik tombol “*Edit*”	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol “Submit”	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data galery.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format

A.10 Usecase Skenario memanajemen data partner

Nama Usecase	Memanajemen data partner
Aktor	Blog Administrator
Pre Kondisi	Blog Administrator telah melakukan login
Post Kondisi	Blog Administrator memanajemen data galery

SKENARIO MANAJEMEN DATA PARTNER	
NORMAL SKENARIO MENAMBAH DATA PARTNER	
1. Klik menu “manajemen data partner”	
	2. Menampilkan tabel data partner disertai tombol “Tambah”, dan “*edit*”.
3. Klik tombol “tambah”.	
	4. Menampilkan form data partner.
5. Mengisi form.	
6. Klik tombol “Submit”	

	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data partner.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	1. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	2. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENG*EDIT* DATA PARTNER	
1. Klik menu "manajemen data partner"	
	2. Menampilkan tabel data partner disertai tombol "Tambah", dan "*edit*".
3. Klik tombol "pensil" yang menandakan "*Edit*"	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol "Submit"	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data partner.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format

A.11 Usecase Skenario memanajemen data profil

Nama Usecase	Memanajemen data profil
Aktor	Blog Administrator
Pre Kondisi	Blog Administrator telah melakukan login
Post Kondisi	Blog Administrator memanajemen

	data profil

SKENARIO MANAJEMEN DATA PROFIL	
NORMAL SKENARIO MENAMBAH DATA PROFIL	
1. Klik menu “manajemen data profil”	
	2. Menampilkan tabel data profil disertai tombol “Tambah”, dan “*edit*”.
3. Klik tombol “tambah”.	
	4. Menampilkan form data profil partner.
5. Mengisi form.	
6. Klik tombol “Submit”	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data profil.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENG*EDIT* DATA PROFIL	
1. Klik menu “manajemen data profil”	
	2. Menampilkan tabel data profil disertai tombol “Tambah”, dan “*edit*”.
3. Klik tombol “pensil” yang menandakan “*Edit*”	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol “Submit”	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data profil.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih

	kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format

A.12 Usecase Skenario memanajemen data akun pembayaran

Nama Usecase	Memanajemen data akun pembayaran
Aktor	Blog Administrator
Pre Kondisi	Blog Administrator telah melakukan login
Post Kondisi	Blog Administrator memanajemen data akun pembayaran

SKENARIO MANAJEMEN DATA AKUN PEMBAYARAN	
NORMAL SKENARIO MENAMBAH DATA AKUN PEMBAYARAN	
1. Klik menu “manajemen data akun pembayaran”	
	2. Menampilkan tabel data akun pembayaran disertai tombol “Tambah”, dan “*edit*, hapus”.
3. Klik tombol “tambah”.	
	4. Menampilkan form data akun pembayaran.
5. Mengisi form.	
6. Klik tombol “Submit”	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data akun pembayaran.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENG*EDIT* DATA AKUN PEMBAYARAN	
1. Klik menu “manajemen data akun pembayaran”	
	2. Menampilkan tabel data akun pembayaran disertai tombol

	"Tambah", dan "*edit*, hapus".
3. Klik tombol "pensil" yang menandakan "*Edit*"	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol "Submit"	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data akun pembayaran.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format
NORMAL SKENARIO MENGHAPUS DATA AKUN PEMBAYARAN	
1. Klik menu "manajemen data akun pembayaran"	
2. Klik tombol "*trash*" yang menandakan "*Delete*"	
	3. Menampilkan pesan "apakah anda akan menghapus data?"
4. Klik tombol "ya".	
	5. Menghapus data.
	6. Menampilkan halaman data akun pembayaran.
ALTERNATIF SKENARIO SAAT TIDAK JADI MENGHAPUS DATA	
4. Klik tombol "tidak".	
	2. Menampilkan halaman data akun pembayaran.

A.13 Skenario memanajemen data kantor

Nama Usecase	Memanajemen data kantor
Aktor	Blog Administrator
Pre Kondisi	Blog Administrator telah melakukan login
Post Kondisi	Blog Administrator memanajemen data kantor

SKENARIO MANAJEMEN DATA KANTOR	
NORMAL SKENARIO MENAMBAH DATA KANTOR	
1. Klik menu "manajemen data kantor"	
	2. Menampilkan tabel data kantor disertai tombol "Tambah", dan "*edit*, hapus".
3. Klik tombol "tambah".	
	4. Menampilkan form data kantor.
5. Mengisi form.	
6. Klik tombol "Submit"	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data kantor.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	8. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENG*EDIT* DATA KANTOR	
1. Klik menu "manajemen data kantor"	
	2. Menampilkan tabel data kantor disertai tombol "Tambah", dan "*edit*, hapus".
3. Klik tombol "pensil" yang menandakan "*Edit*"	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol "Submit"	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data kantor.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	

	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format
NORMAL SKENARIO MENGHAPUS DATA KANTOR	
1. Klik menu "manajemen data kantor"	
2. Klik tombol "*trash*" yang menandakan "*Delete*"	
	3. Menampilkan pesan "apakah anda akan menghapus data?"
4. Klik tombol "ya".	
	5. Menghapus data.
	6. Menampilkan halaman data kantor.
ALTERNATIF SKENARIO SAAT TIDAK JADI MENGHAPUS DATA	
4. Klik tombol "tidak".	
	5. Menampilkan halaman data kantor.

A.15 Usecase Skenario memanajemen data F.A.Q

Nama Usecase	Memanajemen data F.A.Q
Aktor	Blog Administrator
Pre Kondisi	Blog Administrator telah melakukan login
Post Kondisi	Blog Administrator memanajemen data F.A.Q

SKENARIO MANAJEMEN DATA F.A.Q	
NORMAL SKENARIO MENAMBAH DATA F.A.Q	
1. Klik menu "manajemen data F.A.Q"	
	2. Menampilkan tabel data F.A.Q disertai tombol "Tambah", dan "*edit*, hapus".
3. Klik tombol "tambah".	
	4. Menampilkan form data F.A.Q.
5. Mengisi form.	
6. Klik tombol "Submit"	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data F.A.Q.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	

	9. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	10. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENGEDIT DATA F.A.Q	
1. Klik menu "manajemen data F.A.Q"	
	2. Menampilkan tabel data F.A.Q disertai tombol "Tambah", dan "edit, hapus".
3. Klik tombol "pensil" yang menandakan "*Edit*"	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol "Submit"	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data F.A.Q.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	8. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format
NORMAL SKENARIO MENGHAPUS DATA F.A.Q	
1. Klik menu "manajemen data F.A.Q"	
2. Klik tombol "*trash*" yang menandakan "*Delete*"	
	3. Menampilkan pesan "apakah anda akan menghapus data?"
4. Klik tombol "ya".	
	5. Menghapus data.
	6. Menampilkan halaman data F.A.Q.
ALTERNATIF SKENARIO SAAT TIDAK JADI MENGHAPUS DATA	
4. Klik tombol "tidak".	
	5. Menampilkan halaman data

	F.A.Q.

A.16 Usecase Skenario memanajemen data link

Nama Usecase	Memanajemen data link
Aktor	Blog Administrator
Pre Kondisi	Blog Administrator telah melakukan login
Post Kondisi	Blog Administrator memanajemen data link

SKENARIO MANAJEMEN DATA LINK	
NORMAL SKENARIO MENG*EDIT* DATA LINK	
1. Klik menu "manajemen data link"	
	2. Menampilkan tabel data link disertai tombol "*edit*".
3. Klik tombol "pensil" yang menandakan "*Edit*"	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol "Submit"	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data link.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format
	2. Menampilkan halaman data link.

A.17 Usecase Skenario memanajemen data slide

Nama Usecase	Memanajemen data slide
Aktor	Blog Administrator
Pre Kondisi	Blog Administrator telah melakukan login
Post Kondisi	Blog Administrator memanajemen data slide

SKENARIO MANAJEMEN DATA SLIDE	
NORMAL SKENARIO MENG*EDIT* DATA SLIDE	
1. Klik menu "manajemen data slide"	
	2. Menampilkan tabel data slide disertai tombol "*edit*".
3. Klik tombol "pensil" yang menandakan "*Edit*"	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol "Submit"	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data slide.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format

A.18 Usecase Skenario memanajemen pesan travel

Nama Usecase	Memanajemen data pesan travel
Aktor	Member
Pre Kondisi	Member telah melakukan login
Post Kondisi	Member memanajemen data pesan travel

SKENARIO MANAJEMEN DATA PESAN TRAVEL	
NORMAL SKENARIO MENAMBAH DATA PESAN TRAVEL	
1. Klik menu "manajemen data pesan travel"	
	2. Menampilkan tabel data pesan travel disertai tombol "Tambah" dan "detail".
3. Klik tombol "tambah".	
	4. Menampilkan form data pesan travel.
5. Mengisi form.	
6. Klik tombol "Submit"	
	7. Memeriksa data
	8. Menyimpan data

	9. Menampilkan halaman data pesan travel.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	1. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	2. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MELIHAT DETAIL DATA PESAN TRAVEL	
1. Klik menu "manajemen data pesan travel"	
	2. Menampilkan tabel data pesan travel disertai tombol "Tambah" dan "detail".
3. Klik tombol "detail".	
	4. Menampilkan halaman data pesan travel.

A.19 Usecase Skenario memanajemen data pesanan

Nama Usecase	Memanajemen data pesanan
Aktor	Member
Pre Kondisi	Member telah melakukan login
Post Kondisi	Member memanajemen data pesanan

SKENARIO MANAJEMEN DATA PESANAN	
NORMAL SKENARIO MELIHAT TIKET YANG DIPESANAN	
1. Klik menu "manajemen data pesanan"	
	2. Menampilkan tabel pesanan disertai tombol "Print".
3. Klik tombol "print".	
	4. Menampilkan tiket dalam bentuk PDF.
NORMAL SKENARIO MENGHAPUS DATA PESANAN	
1. Klik menu "manajemen data pesanan"	
2. Klik tombol "*trash*" yang menandakan "*Delete*"	
	3. Menampilkan pesan "apakah anda akan menghapus data?"
4. Klik tombol "ya".	

	5. Menghapus data.
	6. Menampilkan halaman pesanan.
ALTERNATIF SKENARIO SAAT TIDAK JADI MENGHAPUS DATA	
4. Klik tombol “tidak”.	
	2. Menampilkan halaman data pesanan.

A.20 Usecase Skenario memanajemen data penumpang

Nama Usecase	Memanajemen data penumpang
Aktor	Member
Pre Kondisi	Member telah melakukan login
Post Kondisi	Member memanajemen data penumpang

SKENARIO MANAJEMEN DATA PENUMPANG	
NORMAL SKENARIO MENAMBAH DATA PENUMPANG	
1. Klik menu “manajemen data penumpang”	
	2. Menampilkan tabel penumpang disertai tombol “Tambah”, dan “hapus”.
3. Klik tombol “tambah”.	
	4. Menampilkan form data penumpang.
5. Mengisi form.	
6. Klik tombol “Submit”	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data penumpang.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	8. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MENGHAPUS DATA PENUMPANG	
1. Klik menu “manajemen data penumpang”	
2. Klik tombol “hapus”	
	3. Menampilkan pesan “apakah

	anda akan menghapus data?"
4. Klik tombol "ya".	
	5. Menghapus data.
	6. Menampilkan halaman penumpang.
ALTERNATIF SKENARIO SAAT TIDAK JADI MENGHAPUS DATA	
4. Klik tombol "tidak".	
	5. Menampilkan halaman data penumpang.

A.21 Usecase Skenario memanajemen data laporan penjualan

Nama Usecase	Memanajemen data laporan penjualan
Aktor	Member
Pre Kondisi	Member telah melakukan login
Post Kondisi	Member memanajemen data laporan penjualan

SKENARIO MANAJEMEN DATA LAPORAN PENJUALAN	
NORMAL SKENARIO MELIHAT DETAIL DATA LAPORAN PENJUALAN	
1. Klik menu "manajemen data laporan penjualan"	
	2. Menampilkan tabel data laporan penjualan disertai tombol "Filter".
3. Masukkan data yang akan difilter	
4. Klik tombol "Filter".	
	5. Menampilkan halaman data laporan penjualan.

A.22 Usecase Skenario memanajemen data saldo penjualan

Nama Usecase	Memanajemen data saldo penjualan
Aktor	Member
Pre Kondisi	Member telah melakukan login
Post Kondisi	Member memanajemen data saldo penjualan

SKENARIO MANAJEMEN DATA SALDO PENJUALAN	
NORMAL SKENARIO MELIHAT DETAIL DATA SALDO PENJUALAN	
1. Klik menu "manajemen data saldo penjualan"	
	2. Menampilkan tabel data saldo penjualan disertai tombol "Tambah", dan "detail, *edit*,

	hapus".
3. Masukkan data yang akan difilter	
4. Klik tombol "detail".	
	5. Menampilkan halaman data saldo penjualan.

A.23 Usecase Skenario memanajemen pesan travel

Nama Usecase	Memanajemen data pesan travel
Aktor	Super Member
Pre Kondisi	Super Member telah melakukan login
Post Kondisi	Super Member memanajemen data pesan travel

SKENARIO MANAJEMEN DATA PESAN TRAVEL	
NORMAL SKENARIO MENAMBAH DATA PESAN TRAVEL	
1. Klik menu "manajemen data pesan travel"	
	2. Menampilkan tabel data pesan travel disertai tombol "Tambah" dan "detail".
3. Klik tombol "tambah".	
	4. Menampilkan form data pesan travel.
5. Mengisi form.	
6. Klik tombol "Submit"	
	7. Memeriksa data
	8. Menyimpan data
	9. Menampilkan halaman data pesan travel.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	3. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	10. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MELIHAT DETAIL DATA PESAN TRAVEL	
1. Klik menu "manajemen data pesan travel"	
	3. Menampilkan tabel data pesan travel disertai tombol "Tambah" dan "detail".

4. Klik tombol “detail”.	
	5. Menampilkan halaman data pesan travel.

A.24 Usecase Skenario memanajemen data pesanan

Nama Usecase	Memanajemen data pesanan
Aktor	Super Member
Pre Kondisi	Super Member telah melakukan login
Post Kondisi	Super Member memanajemen data pesanan

SKENARIO MANAJEMEN DATA PESANAN	
NORMAL SKENARIO MENAMBAH DATA PESANAN	
1. Klik menu “manajemen data pesanan”	
	3. Menampilkan tabel pesanan disertai tombol “Tambah” dan “detail”.
4. Klik tombol “tambah”.	
	5. Menampilkan form data pesanan.
6. Mengisi form.	
7. Klik tombol “Submit”	
	8. Memeriksa data
	9. Menyimpan data
	10. Menampilkan halaman data pesanan.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	9. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	4. Menampilkan pesan bahwa data yang tekah diinputkan tidak sesuai format
NORMAL SKENARIO MELIHAT DETAIL DATA PESANAN	
1. Klik menu “manajemen data pesanan”	
	2. Menampilkan tabel data pesanan disertai tombol “Tambah” dan “detail”.
3. Klik tombol “detail”.	
	4. Menampilkan halaman data pesanan.

NORMAL SKENARIO MENG*EDIT* DATA PESANAN	
1. Klik menu "manajemen data pesanan"	
	2. Menampilkan tabel data pesanan disertai tombol "Tambah" dan "detail".
3. Klik tombol "*Edit*"	
4. Mengubah data pada kolom yang dituju.	
5. Klik tombol "Submit"	
	6. Memeriksa data.
	7. Menyimpan data.
	8. Menampilkan halaman data pesanan.
ALTERNATIF SKENARIO KLIK TOMBOL SIMPAN KETIKA TERDAPAT FIELD/KOLOM YANG MASIH KOSONG	
	7.a. Menampilkan pesan bahwa terdapat kolom yang masih kosong/belum diisi
ALTERNATIF SKENARIO DATA TIDAK SESUAI FORMAT	
	7.b. Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format

A.26 Melihat jadwal travel

Nama Usecase	Melihat jadwal travel
Aktor	Publik
Pre Kondisi	Publik telah membuka web PT KAI
Post Kondisi	Publik melihat jadwal travel

SKENARIO MELIHAT JADWAL TRAVEL	
NORMAL SKENARIO MELIHAT DETAIL JADWAL TRAVEL	
1. Klik menu "jadwal travel"	
	2. Menampilkan jadwal travel dan button"cari travel".
3. Masukkan data yang akan dicari	
4. Klik tombol "cari travel".	
	5. Menampilkan halaman jadwal travel.

A.27 Mendownload tiket

Nama Usecase	Mendownload tiket

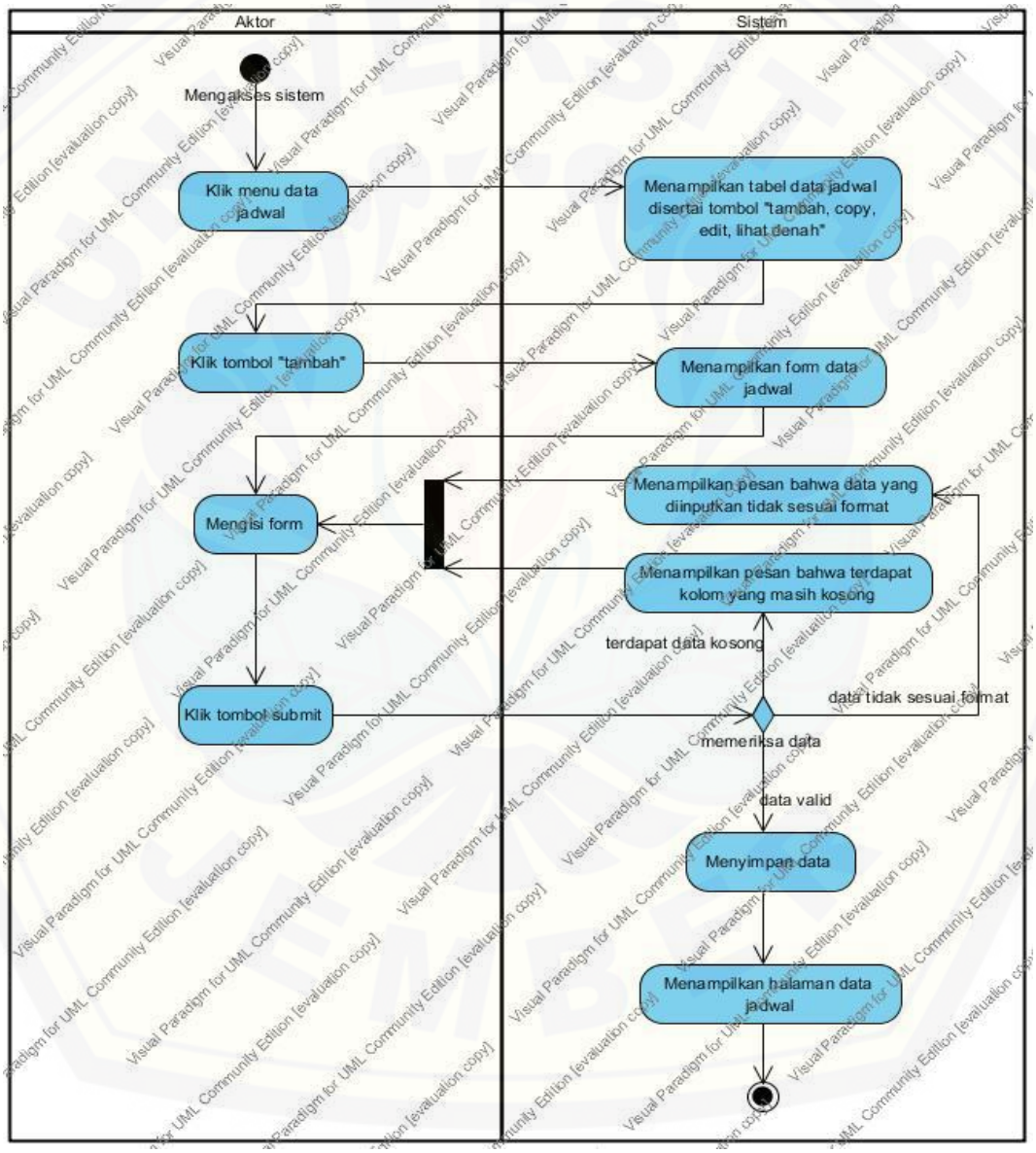
Aktor	Publik
Pre Kondisi	Publik telah membuka halaman home
Post Kondisi	Publik mendownload tiket

SKENARIO MENDOWNLOAD TIKET	
NORMAL SKENARIO DOWNLOAD TIKET	
1. Memasukkan kode *booking* pada form download.	
1. Klik tombol “download”.	
	2. Menampilkan tiket pada browser.
3. Menyimpan file tiket.	

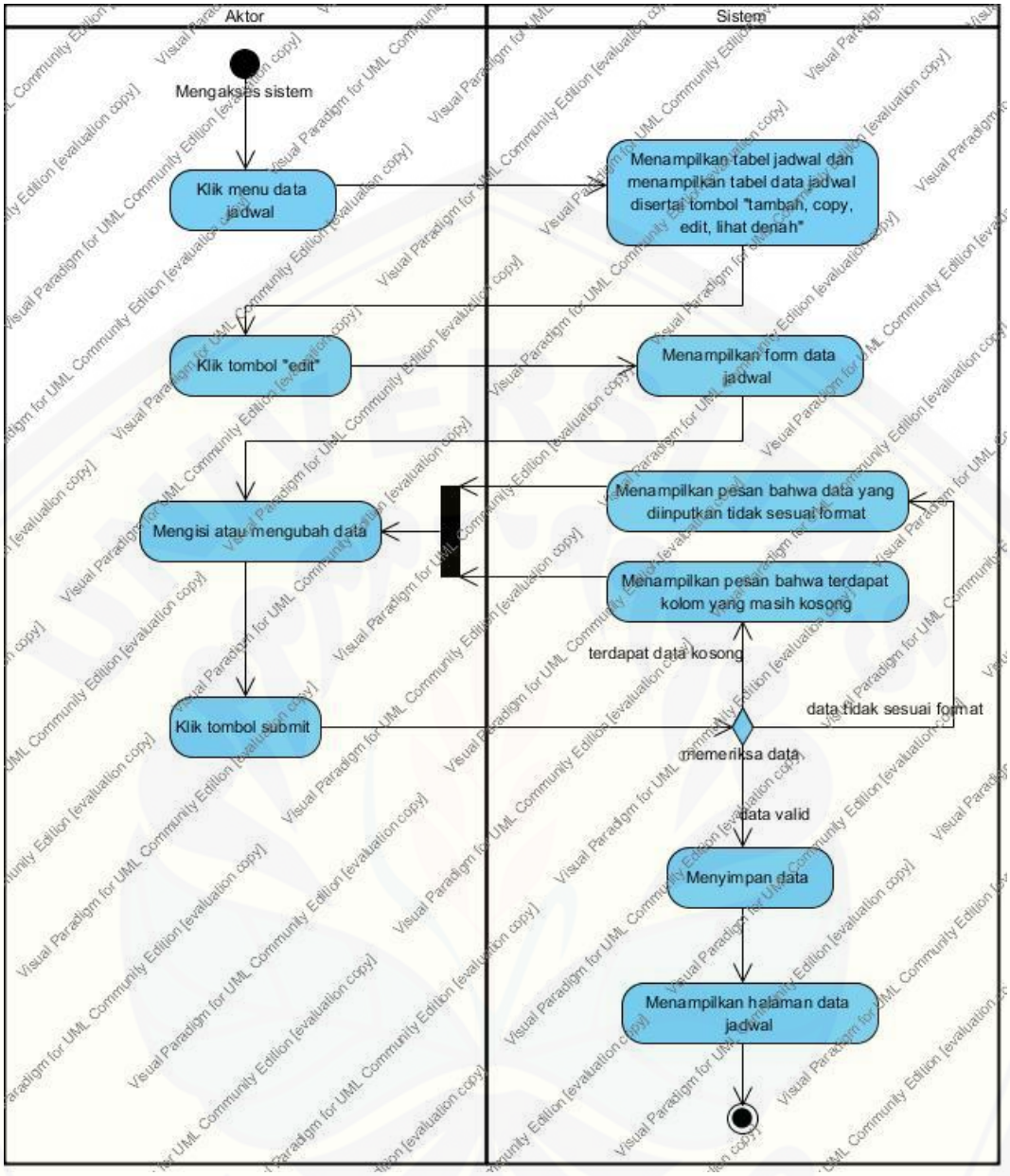
LAMPIRAN B (*Activity Diagram*)

B.1 Activity Diagram Manajemen Data Jadwal

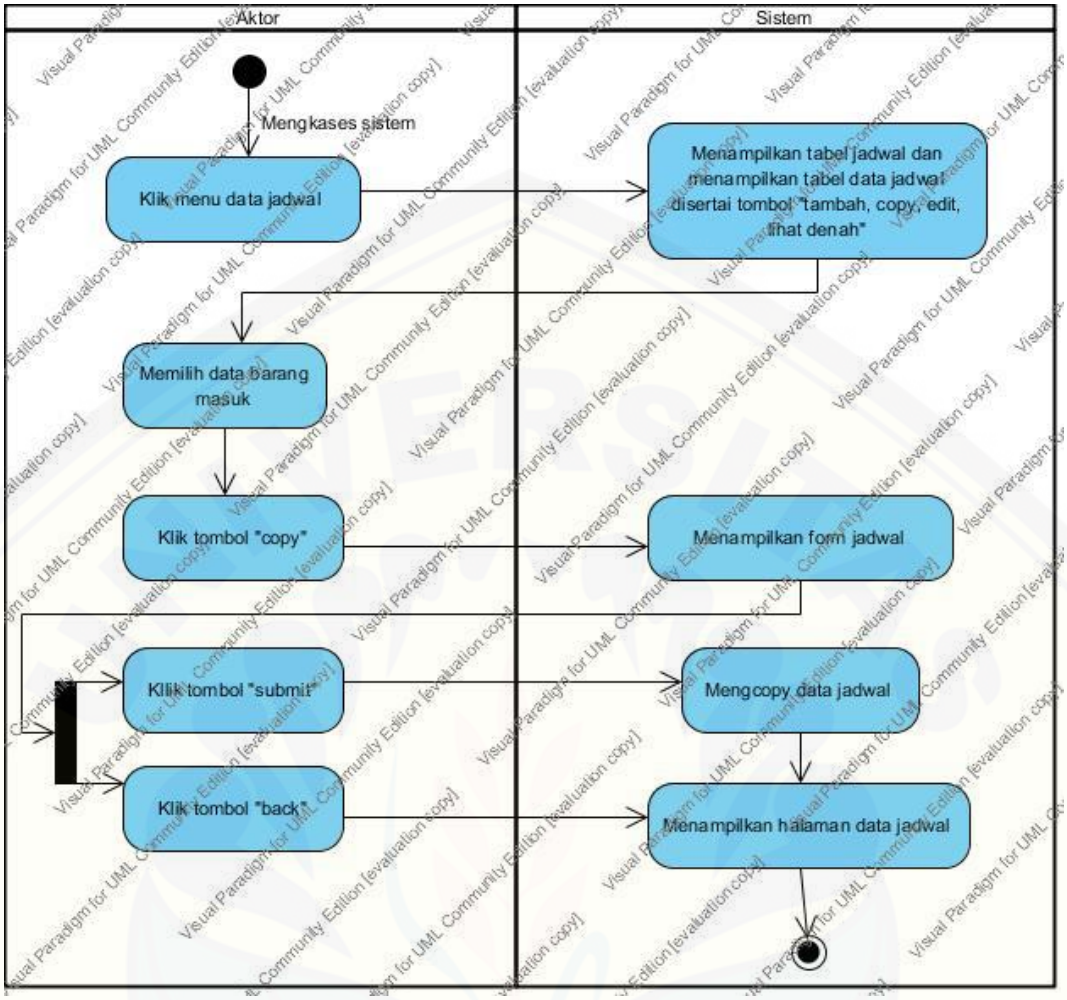
Activity diagram Manajemen Data Jadwal dapat dilakukan oleh super admin. *Activity* diagram ini digunakan pada saat proses mengelola jadwal pemberangkatan oleh super admin yang terdiri dari tambah data jadwal, ubah data jadwal, dan *copy* data jadwal, yang dapat dilihat pada gambar di bawah.



Gambar B.1.1 *Activity* Diagram Tambah Jadwal



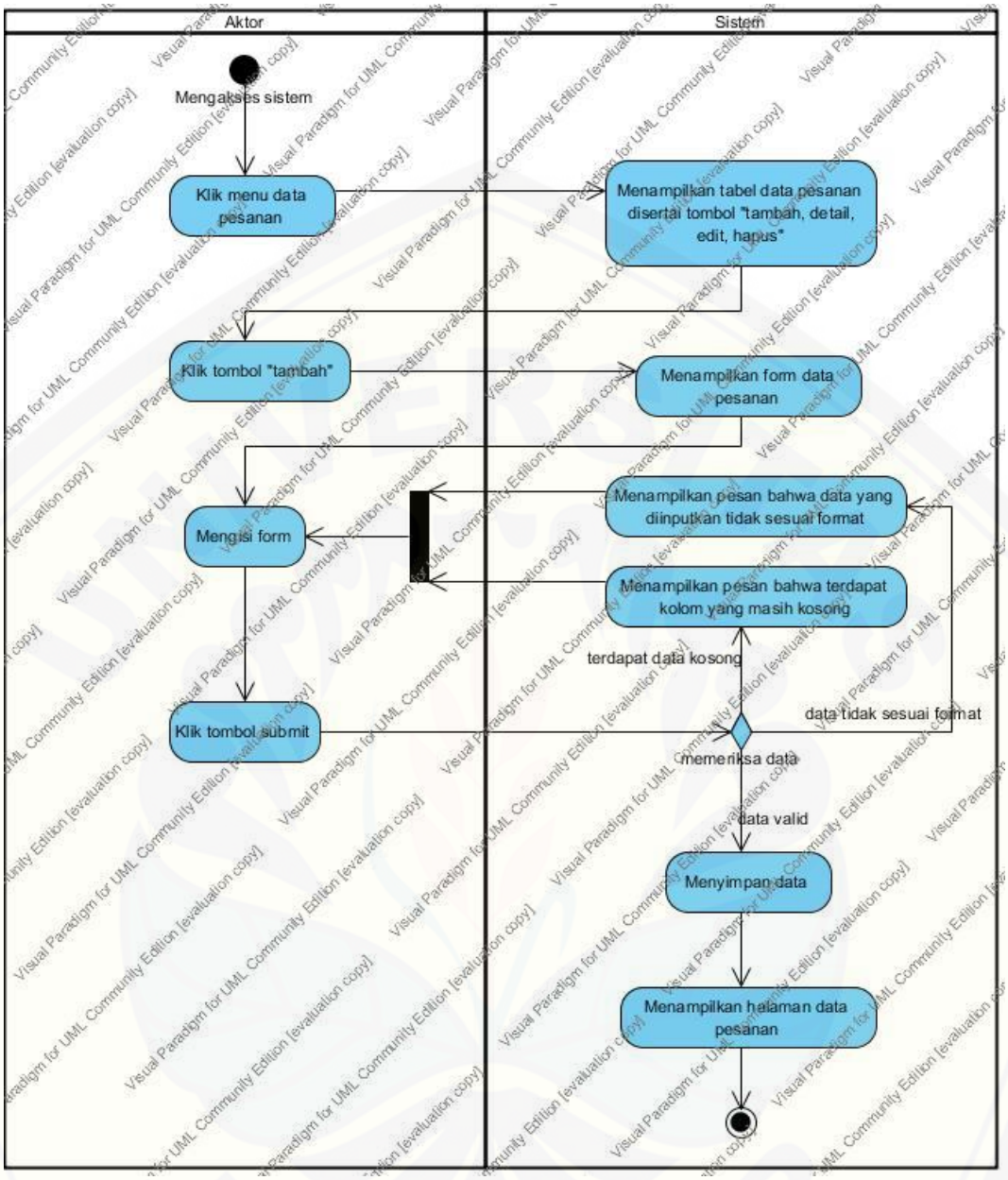
Gambar B.1.2 *Activity* Diagram Ubah Jadwal



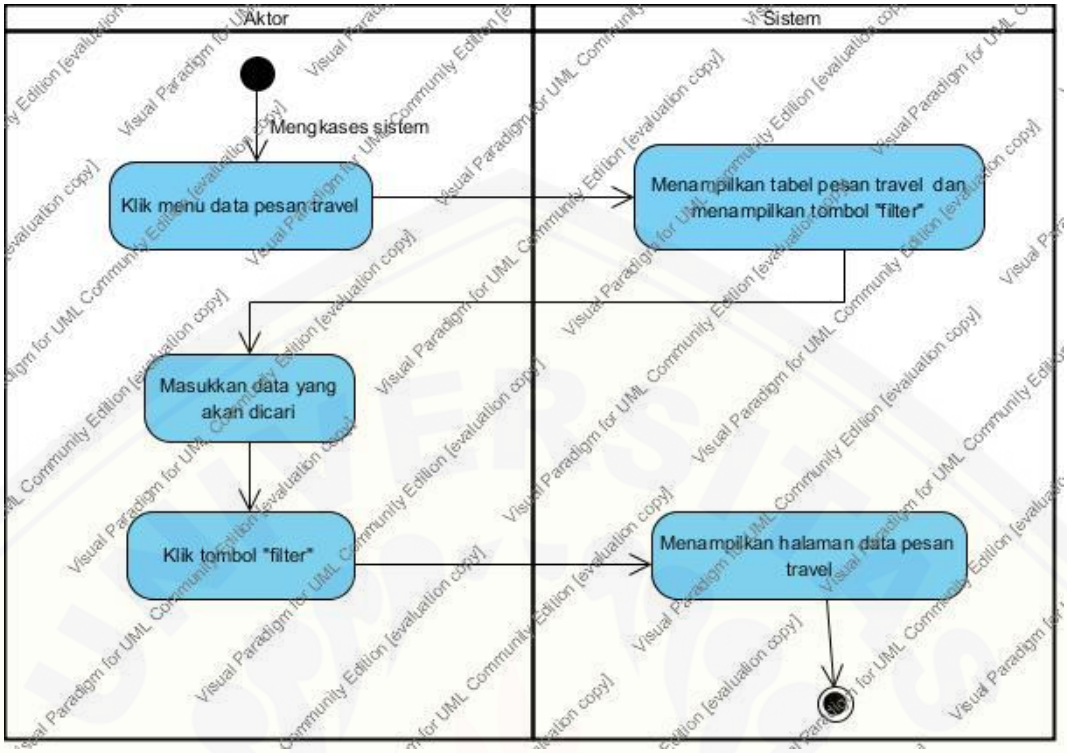
Gambar B.1.3 *Activity* Diagram *Copy* Jadwal

B.2 Activity Diagram Manajemen Pesan Tiket

Activity diagram Manajemen Pesan Travel dapat dilakukan oleh member. *Activity* diagram ini digunakan pada saat proses melakukan pemesanan tiket oleh member yang terdiri dari tambah pesanan dan daftar pemesan, yang dapat dilihat pada gambar di bawah.



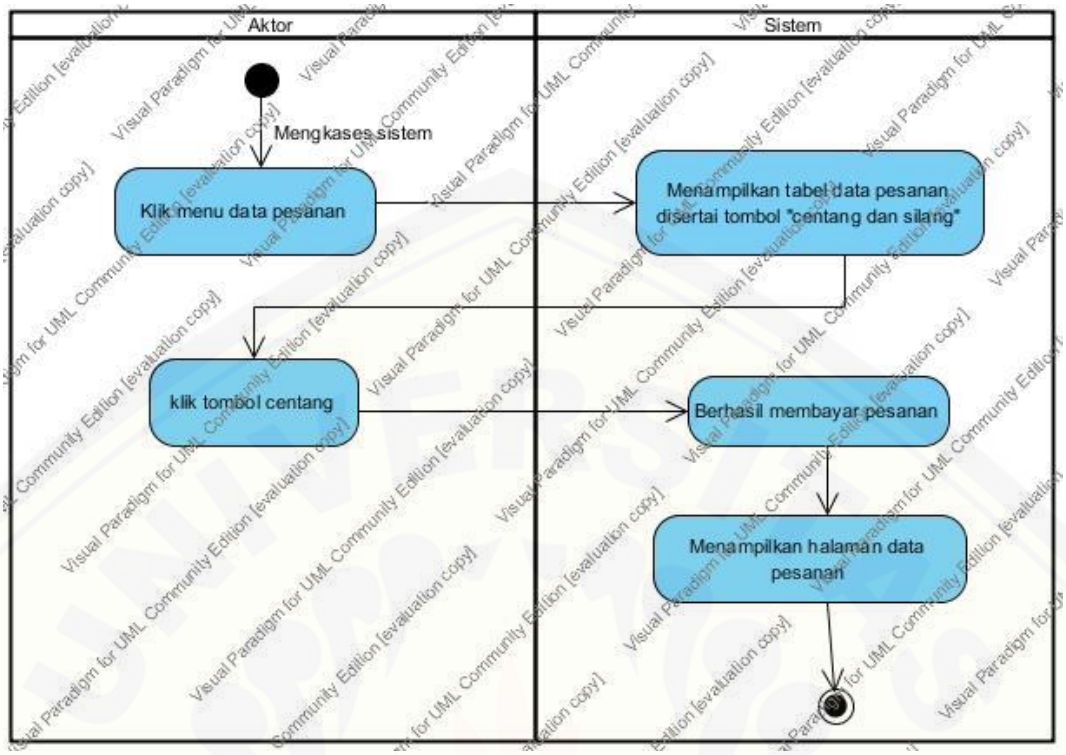
Gambar B.2.1 *Activity* Diagram Tambah Pesanan



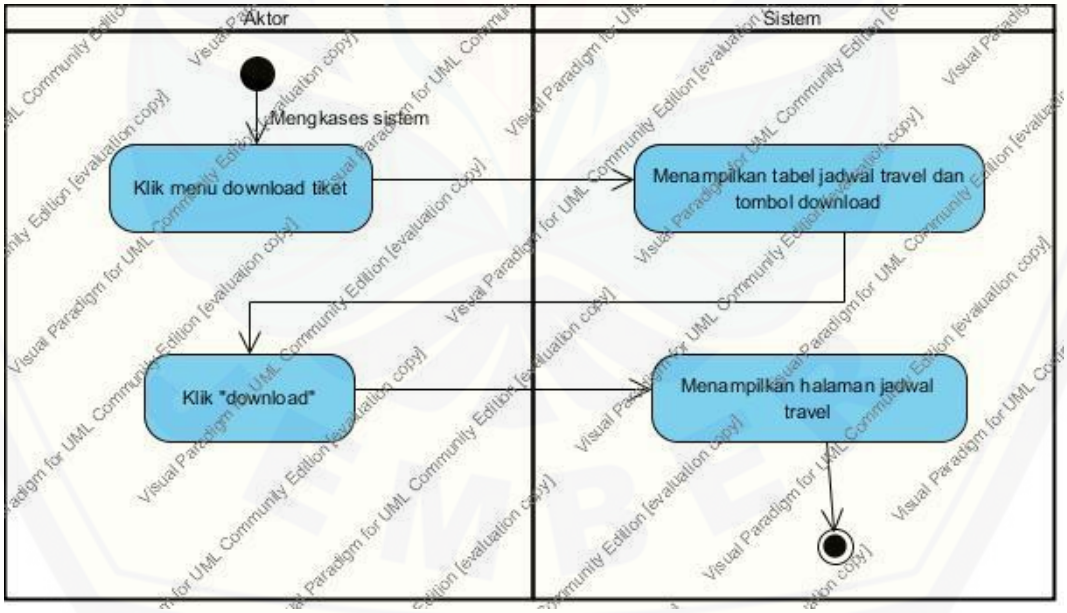
Gambar B.2.2 *Activity* Diagram Daftar Pemesan

B.3 Activity Diagram Mendownload Tiket

Activity diagram Mendownload Tiket dapat dilakukan oleh member. *Activity* diagram ini digunakan pada saat proses melakukan *download* tiket oleh member yang terdiri dari data pesanan dan *download* tiket, yang dapat dilihat pada gambar di bawah.



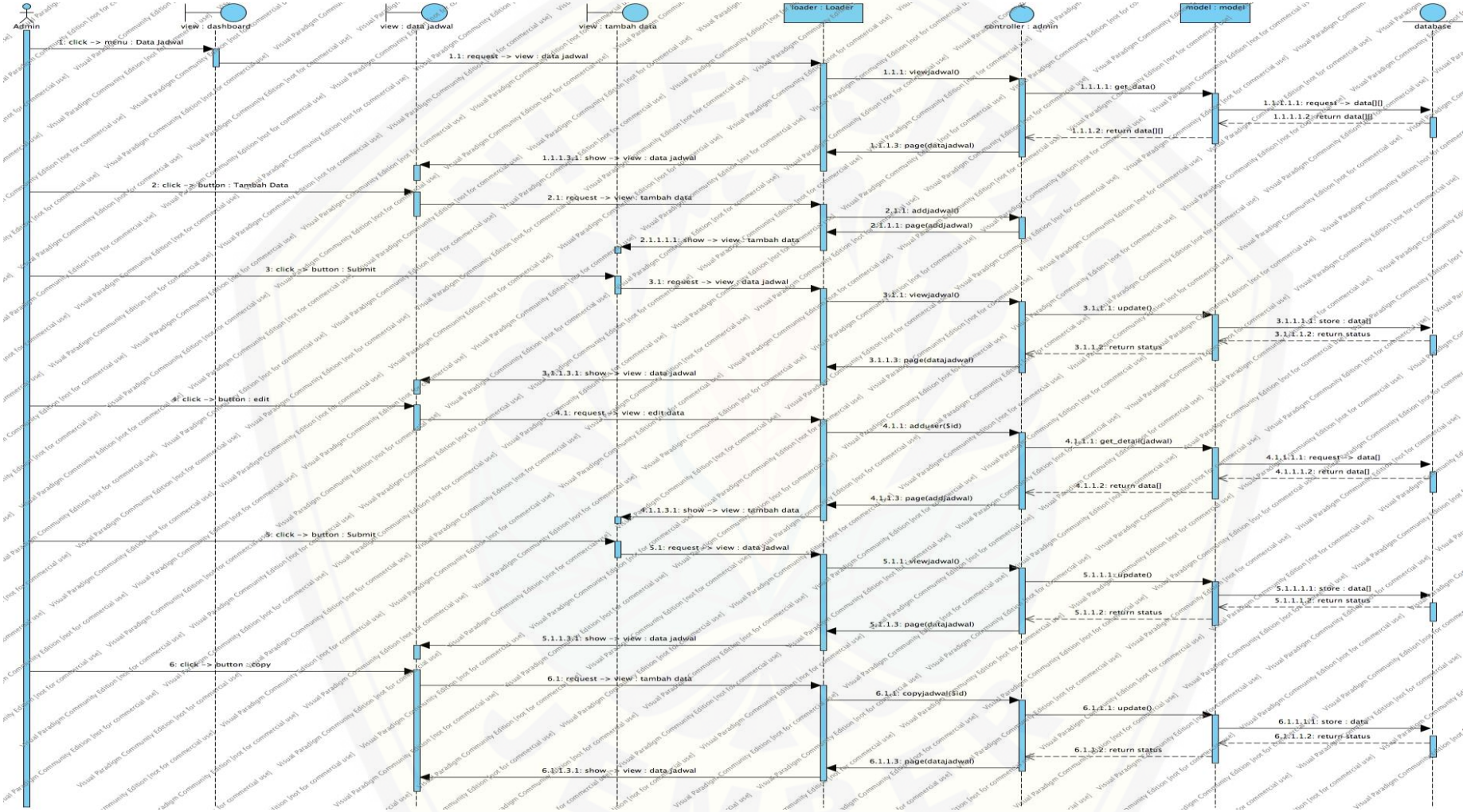
Gambar B.3.1 *Activity* Diagram Data Pesanan



Gambar B.3.2 *Activity* Diagram *Download* Tiket

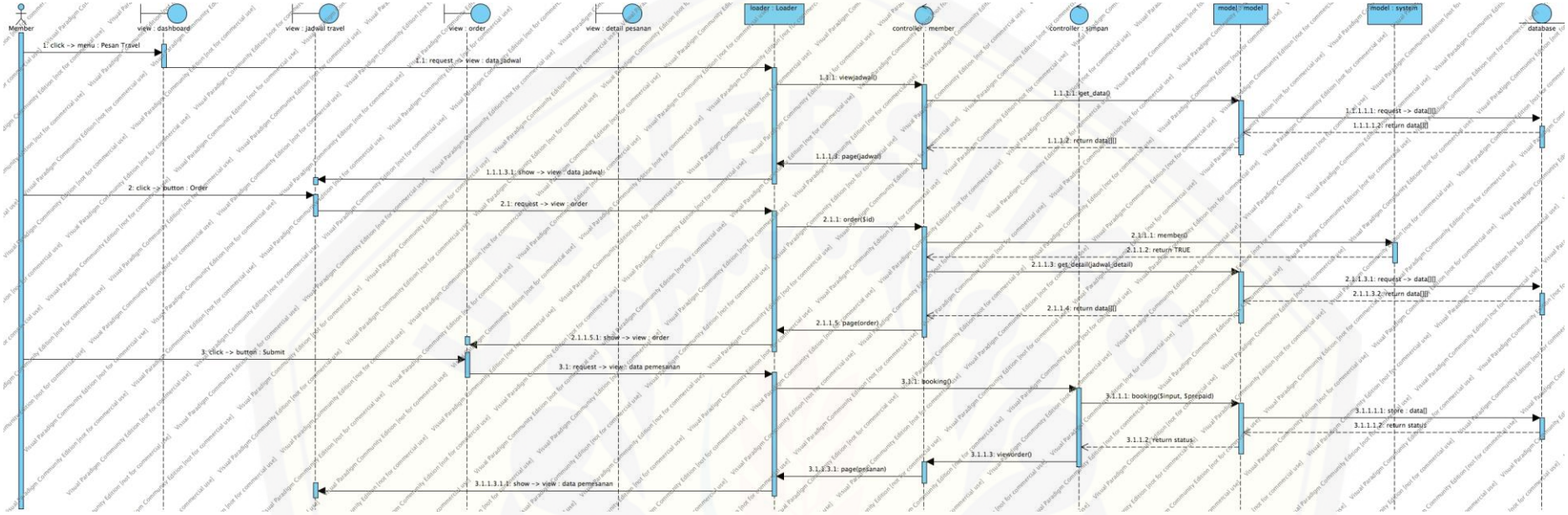
LAMPIRAN C (*Sequence* Diagram)

C.1 Sequence Diagram Manajemen Data Jadwal



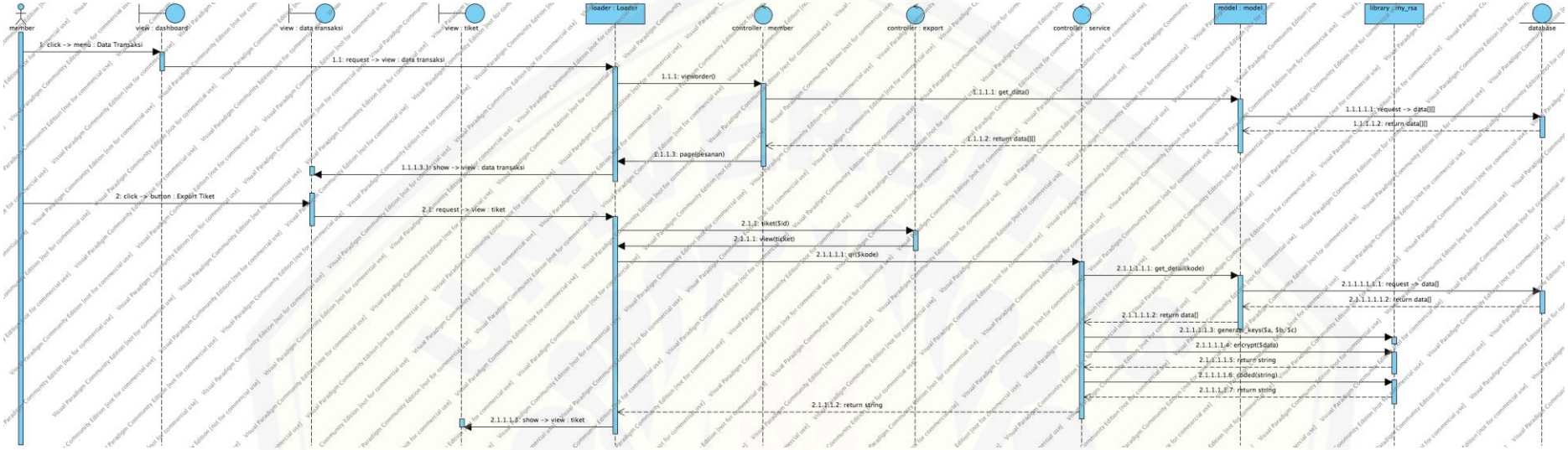
Gambar C.1.1 *Sequence* Diagram Manajemen Data Jadwal

C.2 Sequence Diagram Manajemen Pesan Tiket



Gambar C.2.1 *Sequence* Diagram Manajemen Pesan Tiket

C.3 Sequence Diagram Mendownload Tiket



Gambar C.3.1 *Sequence* Diagram Mendownload Tiket

LAMPIRAN D (Kode Program)

D.1 Kode Program Pada *library* my_rsa

1. *Class my_rsa*

```
class my_rsa {

   private static $keys;

   function __construct() {
   }

   public function generate_keys($p, $q, $show_debug = 0) {
      $n = bcmul($p, $q);

      $m = bcmul(bcsub($p, 1), bcsub($q, 1));
      if (strpos($m, ".") > -1) {
         $xxx = split(".", $m);
         $m = $xxx[0];
      }

      $e = $this->findE($m);
      $d = $this->extend($e, $m);
      $keys = array($n, $e, $d);

      if ($show_debug) {
         echo "P = $p<br>Q = $q<br><b>N = $n</b> - modulo<br>M = $m<br><b>E = $e</b> - public key<br><b>D = $d</b> - private key<p>";
      }

      $this->keys = $keys;//return $keys;
   }

   private function extend($Ee, $Em) {
      $u1 = '1';
      $u2 = '0';
      $u3 = $Em;
      $v1 = '0';
      $v2 = '1';
      $v3 = $Ee;

      while (bccomp($v3, 0) != 0) {
         $qq = bcdiv($u3, $v3, 0);
         $t1 = bcsub($u1, bcmul($qq, $v1));
         $t2 = bcsub($u2, bcmul($qq, $v2));
         $t3 = bcsub($u3, bcmul($qq, $v3));
         $u1 = $v1;
         $u2 = $v2;
         $u3 = $v3;
         $v1 = $t1;
         $v2 = $t2;
         $v3 = $t3;
      }

      $vv = $u2;



      if (bccomp($vv, 0) == -1) {
```

```
            $inverse = bcadd($vv, $Em);
        } else {
            $inverse = $vv;
        }

        return $inverse;
    }

    private function GCD($e, $m) {
        $y = $e;
        $x = $m;

        while (bccomp($y, 0) != 0) {
            $w = bcsub($x, bcmul($y, bcdiv($x, $y, 0)));

            $x = $y;
            $y = $w;
        }

        return $x;
    }

    private function findE($m) {
        $e = '3';
        if (bccomp($this->GCD($e, $m), '1') != 0) {
            $e = '5';
            $step = '2';

            while (bccomp($this->GCD($e, $m), '1') != 0) {
                $e = bcadd($e, $step);

                if ($step == '2') {
                    $step = '4';
                } else {
                    $step = '2';
                }
            }
        }

        return $e;
    }

    public function encrypt($m, $s = 3) {
        $e = $this->keys[1];
        $n = $this->keys[0];

        $coded = '';
        $max = strlen($m);
        $packets = ceil($max / $s);

        for ($i = 0; $i < $packets; $i++) {
            $packet = substr($m, $i * $s, $s);
            $code = '0';

            for ($j = 0; $j < $s; $j++) {
                if (isset($packet[$j])) {
                    $code = bcadd($code, bcmul(ord($packet[$j]), bcpow('256', $j)));
                } else {
                    $code = bcadd($code, bcmul('0', bcpow('256', $j)));
                }
```

```
            }

            $code = bcpowmod($code, $e, $n);
            $coded .= $code . ' ';
        }

        return trim($coded);
    }

    public function decrypt($c) {
        $d = $this->keys[2];
        $n = $this->keys[0];

        $coded = split(' ', $c);
        $message = '';
        $max = count($coded);

        for ($i = 0; $i < $max; $i++) {
            $code = bcpowmod($coded[$i], $d, $n);

            while (bccomp($code, '0') != 0) {
                $ascii = bcmod($code, '256');
                $code = bcdiv($code, '256', 0);
                $message .= chr($ascii);
            }
        }

        return $message;
    }

    function coded($c) {
        $d = $this->keys[2];
        $n = $this->keys[0];
        $coded = split(' ', $c);

        $code = "";
        if (count($coded) > 0) {
            $code = bcpowmod($coded[0], $d, $n);
        }
        for ($i = 1, $max = count($coded); $i < $max; $i++) {
            $code .= " " . bcpowmod($coded[$i], $d, $n);
        }

        return $code;
    }
}
```

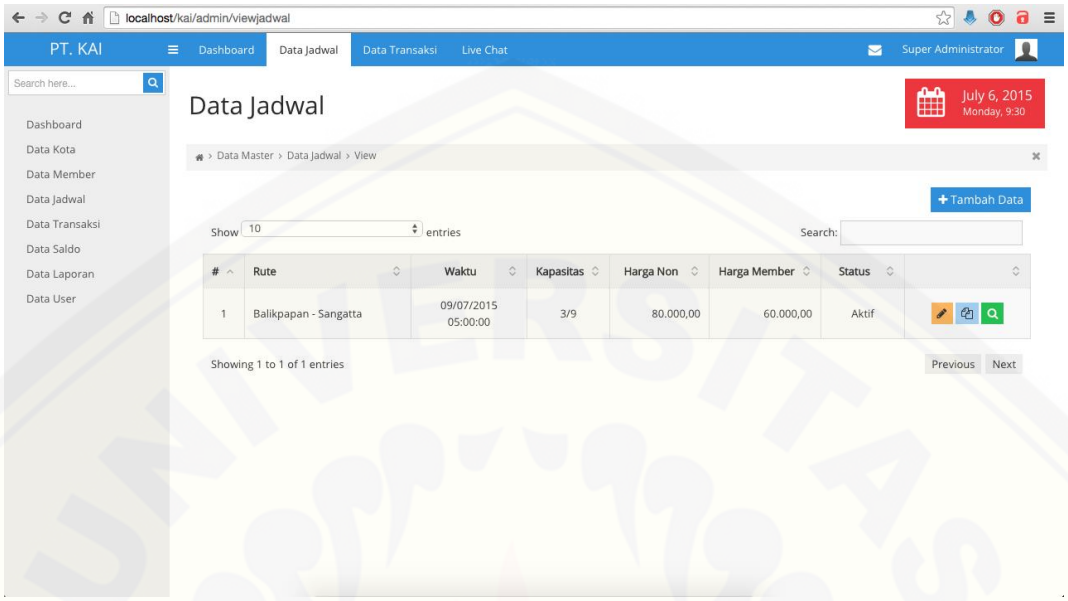
LAMPIRAN E (Pengujian *Black Box*)

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Pesan Travel	Klik menu "manajemen data pesan travel"	a) Menampilkan tabel data pesan travel disertai tombol "Tambah" dan "detail".	[√] Berhasil [ ] Gagal
		Klik tombol "tambah".	b) Menampilkan form data pesan travel.	[√] Berhasil [ ] Gagal
	Tambah Pesanan	Mengisi form lalu klik tombol "Submit"	a) Memeriksa data b) Menyimpan data c) Menampilkan halaman data pesan travel.	[√] Berhasil [ ] Gagal
	Detail Pesanan	Klik menu "manajemen data pesan travel"	a) Menampilkan tabel data pesan travel disertai tombol "Tambah" dan "detail".	[√] Berhasil [ ] Gagal
		Klik tombol "detail".	a) Menampilkan halaman data pesan travel.	[√] Berhasil [ ] Gagal
2.	Melihat Jadwal	Klik menu "Data Jadwal"	a) Menampilkan tabel data jadwal disertai tombol "Tambah", dan "copy, *edit*, lihat denah".	[√] Berhasil [ ] Gagal
	Tambah Jadwal	Klik tombol "Tambah Data".	a) Menampilkan form data jadwal.	[√] Berhasil [ ] Gagal
		Mengisi form lalu klik tombol "Submit"	a) Menampilkan pesan bahwa data yang telah diinputkan tidak sesuai format	[√] Berhasil [ ] Gagal

	Copy Jadwal	Klik menu “manajemen data jadwal”.	a) Menampilkan tabel data jadwal disertai tombol “Tambah”, dan “copy, *edit*, lihat denah”	[√] Berhasil [ ] Gagal
		Klik tombol “copy”.	a) Menampilkan form jadwal	[√] Berhasil [ ] Gagal
		Klik tombol “submit”.	a) Mengcopy data jadwal. b) Menampilkan halaman data jadwal.	[√] Berhasil [ ] Gagal

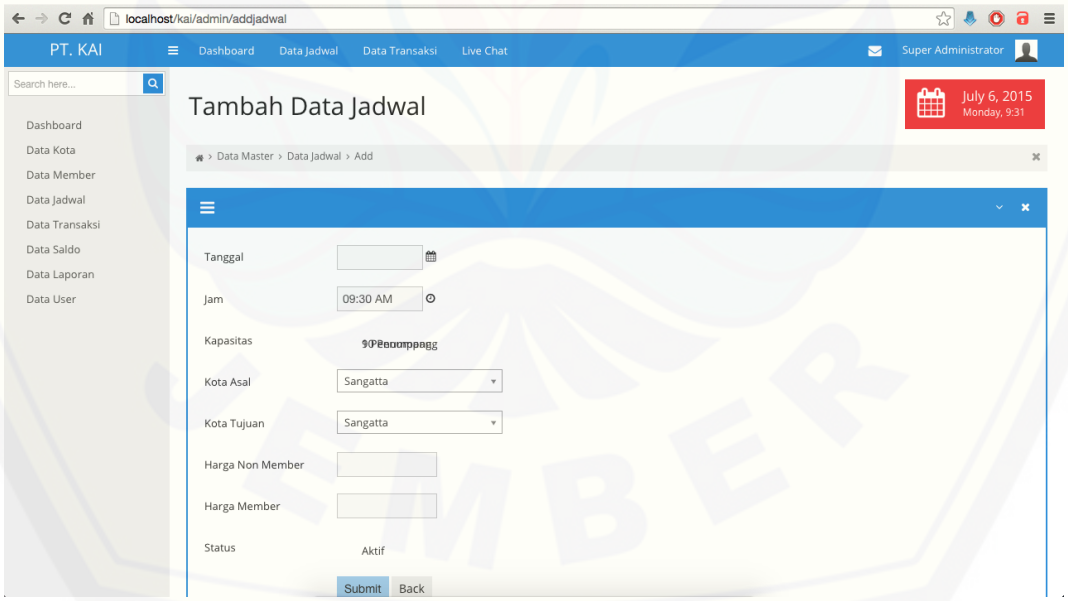
LAMPIRAN F (Hasil Pembuatan Sistem)

F.1 Tampilan Fitur Melihat Data Jadwal



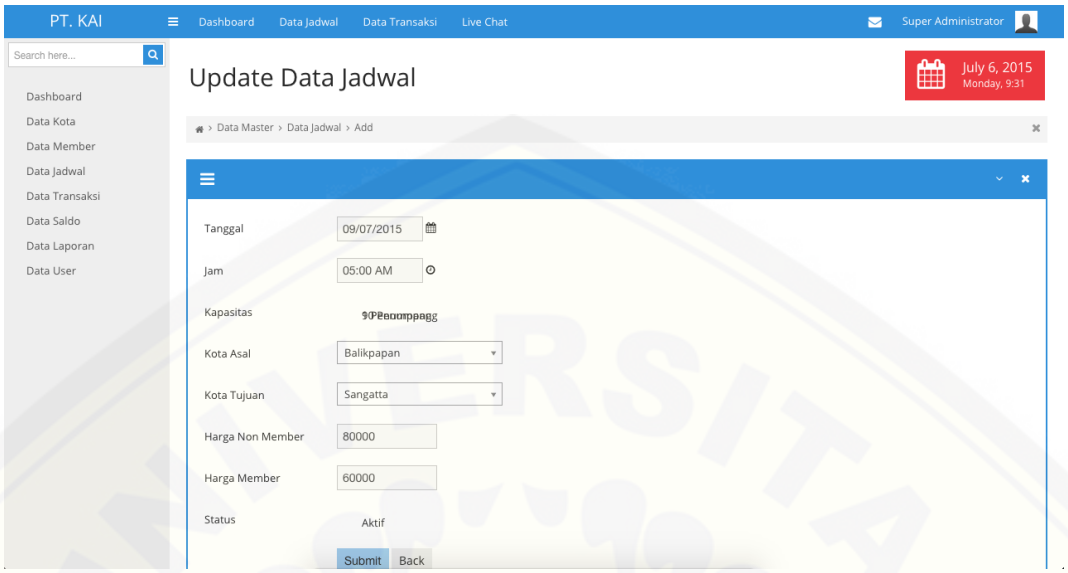
Gambar F.1 Tampilan Fitur Melihat Data Jadwal

F.2 Tampilan Fitur Tambah Data Jadwal



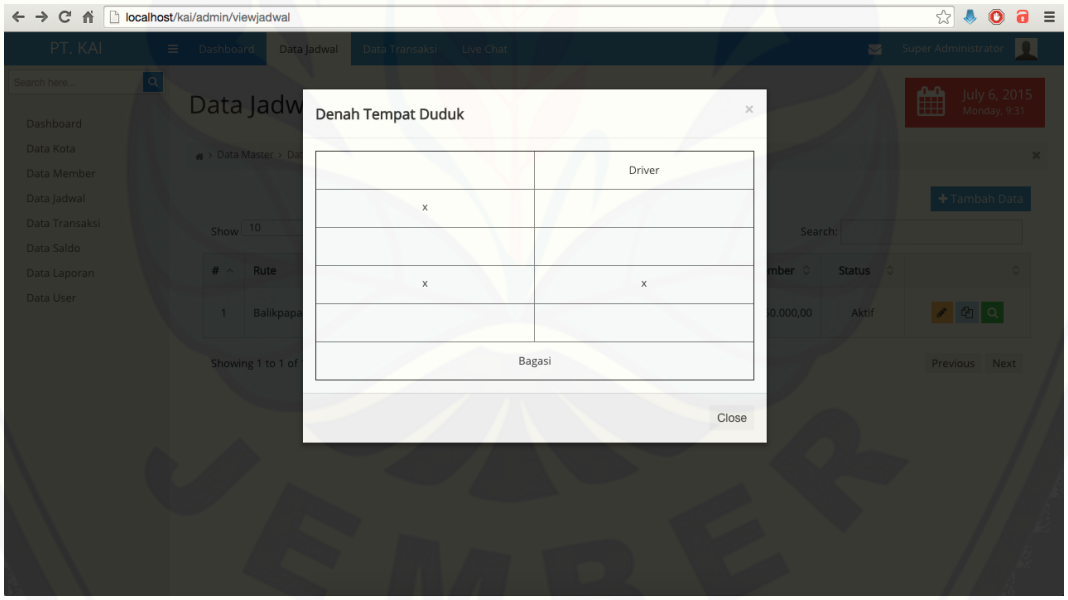
Gambar F.2 Tampilan Fitur Tambah Data Jadwal

F.3 Tampilan Fitur Update Data Jadwal



Gambar F.3 Tampilan Fitur Update Data Jadwal

F.4 Tampilan Fitur Detail Kursi



Gambar F.4 Tampilan Fitur Detail Kursi