

ABSTRAK

Mobidu merupakan salah satu perusahaan dibidang IT yang dibentuk pada tahun 2013. produk dan jasa yang biasa ditawarkan yaitu: *software*, *website*, layanan hosting domain dan desain grafis. Berdasarkan surat balasan dari Mobidu dengan nomor 122-MS/SBPKL/Surat Balasan/VIII/2019, adapun tugas yang diberikan dan dilaksanakan yaitu membuat aplikasi pemesanan tiket umrah untuk Mobidu. Metodologi yang digunakan adalah *Rational Unified Process*, sebagai salah satu tahap pengembangan aplikasi dengan empat tahapan yaitu *inception*, *elaboration*, *contruction*, serta *transition* dengan menggunakan pemodelan *Unified Modelling Language*, serta menggunakan metode pengujian *black-box*. Hasil yang didapat yaitu berhasil dibuatnya aplikasi pemesanan tiket umrah berbasis *web*. Didalam kesempatan kerja praktik ini, terdapat pengetahuan dan keterampilan baru yang diperoleh dilapangan, dalam hal ini adalah penggunaan *laravel*. Dengan menggunakan *framework laravel*, bisa mempercepat proses editing serta membuat kerangka kerja program agar lebih mudah dipakai secara bersama atau tim. Lingkungan Mobidu bisa dibilang sangat ramah dan para pegawai adalah para alumni dan mahasiswa Sekolah Tinggi Teknologi Garut, jadi tidak sulit untuk beradaptasi. Aturan tidak tertulis yang berlaku di Mobidu adalah bekerja santai akan tetapi pekerjaan harus tetap terselesaikan dengan semestinya. Mobidu juga memberikan keleluasaan untuk beribadah, kemudian jam istirahat dan jam kerja seperti tempat kerja pada umumnya.

Kata Kunci : Tiket, *Laravel*, *Framework*

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatu.

Puji dan sayukur penulis panjatkan ke adirat Allah SWT yang telah memberikan berkat, rahmat, dan karunianya. Shalawat serta salam mudah-mudahan tercurah limpahkan kepada junjungan Nabi kita semua Nabi Muhammad ASW, berkat ridho serta hidayahnya penulis dapat menyelesaikan laporan kerja praktik ini tepat pada waktunya.

Laporan ini dibuat dengan tujuan supaya memenuhi syarat kuliah kerja praktek pada program studi Strata Satu di Studi Informatika Sekolah Tinggi Teknologi Garut (STTG), judul yang penulis ambil dalam pembuatan laporan ini adalah **“RANCANG BANGUN APLIKASI PEMESANAN TIKET UMRAH BERBASIS WEB DI MOBIDU SINERGI”**.

Dalam penulisan laporan kerja praktek ini penulis mendapat bantuan dan dukungan dari beberapa pihak, oleh karena itu penulis mengucapkan rasa terimakasih kepada yang terhormat:

1. Bapak Dr. H. Hilmi Aulawi, ST., MT., selaku ketua Sekolah Tinggi Teknologi Garut (STTG).
2. Bapak Dede Kurniadi, S.Kom., M.Kom selaku Ketua Jurusan Program Studi Informatika.
3. Ibu Leni Fitriani, M.Kom., selaku dosen pembimbing dalam dukungan bimbingan penulisan Laporan Kerja Praktek ini, untuk itu penulis ucapkan terimakasih yang sebesar-besarnya.
4. Bapak dan ibu tercinta, kakak dan adik, saudara-saudara dan seluruh keluarga serta orang-orang terdekat yang sudah memberikan bantuan, dorongan dan motivasi bagi penulis selama melakukan penelitian ini.
5. Sahabat dan rekan-rekan Program Studi Informatika angkatan 2016 yang telah membantu dan memberikan dukungan.
6. Seluruh Staff Dosen dan aktivitas akademik Sekolah Tinggi Teknologi Garut.
7. Dan pihak lain ang tidak bisa disebutkan satu persatu yang telah memberikan doa dalam penulisan laporan kerja praktik ini.

Penulis menyadari dalam laporan Kerja Praktek ini masih jauh dari sempurna. Karenannya, penulis minta maaf apabila banyak kesalahan dalam penulisa maupun kata-kata yang tidak dimasukan ke dalam laporan kerja praktek ini dan penulis mengharapkan kritik dan saran membangun untuk kesempurnaan laporan Kerja Praktek ini.

Wasalammu'alaikum Warahmatullahi Wabarakatu.

Garut, 12 Desember 2019

Penulis

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
DAFTAR LAMPIRAN	viii
1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan Pekerjaan	1
1.3. Ruang Lingkup Pekerjaan	2
1.4. Tempat dan Waktu Kerja	2
1.5. Sistematika Penulisan	2
2. LANDASAN TEORI	4
2.1 Rancang Bangun	4
2.2. Aplikasi <i>Web</i> Pemesanan Tiket Umrah	4
2.3. <i>Rational Unified Process</i>	5
2.3.1. <i>Unified Modeling Language</i>	7
2.4. Perangkat Pendukung Pengembangan	8
2.4.1. Pemodelan	8
2.4.2. Implementasi	8
3. METODOLOGI PEKERJAAN	11
3.1. <i>Work Breakdown Structur (WBS)</i>	11
3.2. <i>Gantt Chart</i>	13
3.3. Sumber Daya Penelitian	13
4. HASIL DAN PEMBAHASAN	16
4.1. Hasil Pekerjaan	16
4.2. Pengetahuan dan Keterampilan Empiris	23
4.3. Etika Kerja	26
5. KESIMPULAN DAN SARAN	27
5.1. Kesimpulan	27

5.2. Saran.....	27
DAFTAR PUSTAKA	28

DAFTAR GAMBAR

Gambar 2. 1 Tahapan RUP (<i>Rational Unified Process</i>)	5
Gambar 3. 1 <i>Work Breakdown Structure</i>	11
Gambar 3. 2 Diagram Alur Aktivitas	13
Gambar 4. 1 Proses Bisnis Rancang Bangun Pemesanan Tiket di Jalankan	16
Gambar 4. 2 <i>Use Case Diagram</i> Aplikasi pemesanan tiket umrah	18
Gambar 4. 3 Tampilan <i>Admin</i> Aplikasi Pemesanan Tiket Umrah	21
Gambar 4. 4 Tampilan <i>client</i> Aplikasi Pemesanan Tiket Umrah	22
Gambar 4. 5 <i>Install Laravel</i>	24
Gambar 4. 6 Memilih <i>Directory</i>	24
Gambar 4. 7 Membuat <i>Project Laravel</i>	25
Gambar 4. 8 Masuk Ke <i>Directory Project</i>	25
Gambar 4. 9 Menjalankan <i>Project</i>	25
Gambar 4. 10 Tampilan <i>Project Awal Laravel</i>	25

DAFTAR TABEL

Tabel 3. 1 Sumber Daya Penelitian.....	13
--	----

DAFTAR LAMPIRAN

LAMPIRAN A : LANDASAN PEKERJAAN	28
LAMPIRAN B : PRESENSI KERJAAN.....	30
LAMPIRAN C : HASIL PEKERJAAN.....	32

1. PENDAHULUAN

1.1. Latar Belakang

Mobidu Sinergi adalah sebuah perusahaan yang bergerak di bidang teknologi informasi dan multimedia kelas dunia yang berdiri pada tahun 2017 beralamatkan di Jl. Rancabango *Estate* Ruko A2 Tarogong Kaler Jawa Barat 44151. Mobidu berasal dari kata *mobile education* sedangkan nama sinergi memiliki makna saling melengkapi perbedaan untuk mencapai hasil lebih besar dalam bidang teknologi informasi dan umum. Inovator Mobidu Sinergi yaitu Yusep Maulana ingin mendirikan perusahaan penyedia produk dan jasa pembuat dan pengembang teknologi informasi dan multimedia yang memberikan solusi pengembangan perangkat lunak berbasis web, desktop dan android. Mobidu membawahi unit kerja *Programmer* dan Sistem analis, *Programmer* yaitu menerjemahkan dari spesifikasi desain sistem ke kode komputer (Rainer & Cegielski, 2011), sedangkan Sistem Analis yaitu kegiatan untuk melihat sistem yang sudah berjalan, melihat mana yang bagus dan tidak bagus dan, kemudian mendokumentasikan kebutuhan yang akan dipenuhi dalam sistem yang baru (Rosa & Salahudin, 2018).

Saat ini masyarakat sering menggunakan jasa penyedia online untuk mendukung aktivitasnya salah satu contohnya adalah pemesanan untuk tiket kereta, hotel, dan tiket lainnya. Mobidu Sinergi memanfaatkan peluang tersebut untuk menciptakan aplikasi pemesanan tiket jasa umrah berbasis web. Maka dari itu penulis akan membuat **“Rancang Bangun Pemesanan Tiket Umrah Berbasis Web di Mobidu Sinergi”** yang di harapkan dapat memudahkan pemesanan tiket umrah berbasis web tersebut.

1.2. Tujuan Pekerjaan

Adapun tujuan dalam Kerja Praktik ini adalah membuat aplikasi pemesanan tiket umrah berbasis web di Mobidu Sinergi dengan fitur transaksi pemesanan tiket, pembayaran tiket, pengelolaan tiket dan fitur registrasi untuk pembeli yang ingin bergabung. Aplikasi pemesanan tiket umrah ini juga memiliki fitur *multiuser* yang membedakan fitur antara *admin* dengan *user*.

1.3. Ruang Lingkup Pekerjaan

Ruang lingkup pada pekerjaan ini yaitu:

1. Bahasa pemrograman yang digunakan adalah PHP dengan menggunakan *framework* Laravel;
2. Pelaksanaan kerja praktik ini beranggotakan 2 orang, Rifan Alamsyah berperan sebagai *Programmer* dan Sinta Nurfatonah berperan sebagai Analis sistem;
3. Adapun spesifikasi perangkat yang digunakan, diantaranya.
 - a. Perangkat yang dipakai oleh *Programmer* menggunakan *Windows 10 Pro* 64-bit, *processor* INTEL core i3-3217U, 6 GB RAM; dan
 - b. Perangkat yang dipakai oleh Sistem Analis, menggunakan *Windows 10 Pro* 64-bit, *processor* INTEL core i3-5005U, 2 GB RAM.

1.4. Tempat dan Waktu Kerja

Tempat penelitian yaitu di Mobidu Sinergi yang beralamat di Jl. Rancabango, Ruko A2 Kec Tarogong Kaler, Rancabango *Estate*, Jawa Barat 44151. Sedangkan waktu penelitian dilaksanakan pada tanggal 12 Agustus 2019 sampai 12 September 2019 dari hari senin sampai dengan hari jum'at, dengan jadwal masuk pukul 08:30 – 17:00.

1.5. Sistematika Penulisan

Laporan kerja praktik ini disusun dengan sistematika sebagai berikut :

1. **PENDAHULUAN**, bab ini berisi latar belakang, tujuan pekerjaan, ruang lingkup pekerjaan, tempat dan waktu kerja, serta sistematika penulisan;
2. **LANDASAN TEORI**, bab ini berisi kutipan konsep, teori dan metode terkait tujuan pekerjaan yang bersumber dari sejumlah pustaka;
3. **METODOLOGI PEKERJAAN**, berisi penjelasan tentang tahapan kerja berikut aktivitas dan teknik yang digunakan, serta pelaksana pekerjaan, waktu, tempat, dan sumber daya yang digunakan, dengan memperhatikan landasan teori;
4. **HASIL DAN PEMBAHASAN**, berisi penjelasan tentang hasil pekerjaan, pengetahuan dan keterampilan empiris yang diperoleh, serta bagaimana

kearifan lokal diperhatikan di dalam pekerjaan dengan merujuk kepada pustaka; dan

5. **KESIMPULAN DAN SARAN**, berisi ringkasan penelitian berdasarkan hasil dan pembahasan, serta rekomendasi pekerjaan terkait kelemahan kemampuan mahasiswa atau hambatan di tempat kerja.

2. LANDASAN TEORI

2.1. Rancang Bangun

Rancang adalah proses menganalisa kebutuhan dan mendeskripsikan dengan detail komponen-komponen yang akan di implementasi, sedangkan bangun yaitu menciptakan sistem atau memperbaiki sistem yang sudah ada. (Pressman, 2012). Tujuan akhir dari kegiatan rancang bangun adalah menjawab sebuah permasalahan dengan memberikan solusi yang sesuai dengan kebutuhan yang dapat memudahkan seseorang membuat sistem.

2.2. Aplikasi Web Pemesanan Tiket Umrah

Aplikasi yaitu merupakan salah satu faktor kebutuhan di zaman sekarang untuk mempermudah suatu pekerjaan. Perangkat lunak aplikasi adalah program-program mandiri yang menjawab kebutuhan bisnis yang terperinci. Aplikasi-aplikasi dalam bidang ini melakukan pemrosesan data bisnis atau data teknis yang mendukung berjalannya operasi-operasi bisnis atau pengaturan / pengambilan keputusan teknis. Selain ada aplikasi pemrosesan data konvensional, perangkat lunak aplikasi juga digunakan untuk mengendalikan fungsi-fungsi bisnis secara *actual* (Pressman, 2012).

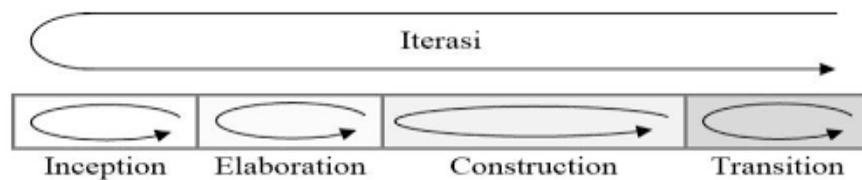
Web atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman (Hidayat, 2010).

Tiket merupakan suatu alat/media yang digunakan oleh perusahaan tertentu sebagai pengganti uang langsung. Tiket biasanya berupa kertas yang didalamnya terdapat item-item tertentu yang menunjukkan suatu nilai. Menurut Kamus Besar Bahasa Indonesia yang dimaksud dengan tiket adalah sesuatu yang dianggap sebagai alat pembayaran yang digunakan oleh suatu alat transportasi yang ada atau suatu wahana, sedangkan dalam proses bisnis aplikasi pemesanan tiket umrah, tiket umrah adalah suatu bukti transaksi dari pembelian kursi kuota umrah dan hak berangkat umrah beserta pesawat yang digunakan untuk berangkat.

Umrah yakni kegiatan menziarahi ka'bah dan berthawaf disekelilingnya, kemudian bersa'i antara shafa dan marwa, serta mencukur rambut (tahallul) tanpa wukuf diarafah (Al-Habsi, 1999).

2.3. Rational Unified Process

Rational Unified Process (RUP) adalah pendekatan pengembangan perangkat lunak yang dilakukan secara berulang-ulang, fokus pada arsitektur, lebih diarahkan berdasarkan penggunaan kasus. RUP juga merupakan sebuah produk proses perangkat lunak yang dikembangkan oleh *Rational Software* yang diakuisisi oleh IBM pada bulan Februari 2003. RUP memiliki empat buah tahapan yang dapat dilakukan secara iterative.



Gambar 2. 1 Tahapan RUP (*Rational Unified Process*)

(Sukamto & Shalahuddin 2018)

Tahap pertama adalah *Inception* (permulaan) lebih pada pemodelan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Berikut adalah tahapan yang dibutuhkan pada tahap ini:

1. Memahami ruang lingkup dari proyek (termasuk pada biaya, waktu, kebutuhan, resiko dan lain sebagainya)
2. Membangun kasus bisnis yang dibutuhkan.

Ditahap ini memetakan kebutuhan sistem, dengan menggunakan *Work Breakdown Structure* (WBS).

Tahap kedua adalah *Elaboration* (perluasan/perencanaan), tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*). Pada tahap ini menggunakan pemodelan dan UML.

Tahap ketiga adalah *Construction* (konstruksi), tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Pada tahap ini, mengimplementasikan kode program dengan XAMPP yang didukung dengan perangkat lunak lainnya. Tahapan pengujian juga termasuk dalam tahapan ketiga ini yang berguna mengecek sistem apakah sudah memenuhi harapan *user*. Tahapan uji coba dilakukan untuk menguji sistem dengan analisis atau tahapan yang ada pada metodologi RUP.

Tahap keempat adalah *Transition* (transisi), tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengunggahan aplikasi agar dapat digunakan secara luas. Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap. Keempat fase pada RUP dijalankan secara berurutan dan iteratif dimana sistem iterasi dapat digunakan untuk memperbaiki iterasi berikutnya.

Kelebihan metodologi RUP (*Rational Unified Process*) adalah (Ependi, Kunang , & Nofika S, 2013) :

1. Menyediakan akses yang mudah terhadap pengetahuan dasar bagi anggota tim.
2. Menyediakan petunjuk bagaimana menggunakan UML secara efektif.
3. Mendukung proses pengulangan dalam pengembangan *software*.
4. Memungkinkan adanya penambahan-penambahan pada proses.
5. Memungkinkan untuk secara sistematis mengontrol perubahan-perubahan yang terjadi pada *software* selama proses pengembangannya.
6. Memungkinkan untuk menjalankan *test*.

2.3.1. *Unified Modeling Language*

UML adalah salah satu standar bahasa yang banyak digunakan didunia industry untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Sukamto & Shalahuddin, 2018). UML meliputi diantaranya.

1. *Use Case* atau diagram *use case* merupakan pemodelan untuk melakukan kelakuan(*behavior*) sistem informasi yang akan dibuat. *Use case* Mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Sukamto & Shalahuddin, 2018);
2. Diagram aktivitas atau *activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Sukamto & Shalahuddin, 2018);
3. *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Sukamto & Shalahuddin, 2018); dan
4. *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas (Sukamto & Shalahuddin, 2018).

2.4. Perangkat Pendukung Pengembangan

Adapun beberapa alat bantu dalam pengimplementasian aplikasi pengembangan aplikasi pemesanan tiket umrah ini akan menggunakan bahasa pemrograman PHP dengan *Framework Laravel* dan penyimpanan data menggunakan *database* MySQL, serta untuk melakukan koneksi *database* menggunakan XAMPP.

2.4.1. Pemodelan

Microsoft visio adalah sebuah program aplikasi komputer yang dirilis oleh *Microsoft Corporation*. Aplikasi ini menggunakan grafik vektor untuk membuat diagram (Ratna, 2014).

2.4.2. Implementasi

Didalam implementasian pengembangan aplikasi menggunakan bahasa pemrograman dan kode editor. Untuk mengimplementasikannya maka dibutuhkan:

1. *Sublime Text*, merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau meng-edit suatu aplikasi. *Sublime Text* mempunyai fitur *plugin* tambahan yang memudahkan *programmer*. Selain itu, *Sublime Text* juga memiliki desain yang simpel dan keren menjadikan *Sublime Text* terkesan elegan untuk sebuah *syntax editor* (Putratama, 2016);
2. Bahasa pemrograman, yang digunakan adalah sebagai berikut.
 - a. HTML CSS (*Cascading Style Sheet*), adalah bahasa yang dapat digunakan untuk mendefinisikan bagaimana suatu bahasa *markup* ditampilkan pada suatu media dimana bahasa *markup* ini salah satunya adalah HTML (Rerung, 2018) ;
 - b. PHP, merupakan singkatan dari *Hypertext Preprocessor*. PHP merupakan bahasa berbentuk skrip yang ditempatkan dalam server. Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser* (Kadir, 2002) ; dan
 - c. Javascript, adalah bahasa *scripting* yang populer di *internet* dan dapat bekerja di sebagian besar *browser* populer seperti *Internet Explorer* (IE), *Mozilla FireFox*, *Netscape* dan *Opera*. Kode *Javascript* dapat disisipkan dalam halaman *web* menggunakan *SCRIPT* (Sunyoto, 2017).

3. *Framework*, adalah sebuah kerangka kerja. *Framework* juga dapat diartikan sebagai kumpulan *script* (terutama *class* dan *function*) yang dapat membantu *developer / programmer* dalam menangani berbagai masalah-masalah dalam pemrograman seperti koneksi ke *database*, pemanggilan variabel, *file,dll* sehingga *developer* lebih fokus dan lebih cepat membangun aplikasi (Warsito, Yusup, & Yulianto, 2014). Pada pengembangan aplikasi pemesanan tiket umrah, menggunakan *framework Laravel* dan *Bootstrap*.
 - a. *Laravel*, adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT, dengan kode sumber yang sudah disediakan oleh Github, sama halnya seperti *framework-framework* yang lain, *Laravel* dibangun dengan konsep MVC (*Model-View-Controller*), kemudian *Laravel* dilengkapi juga *command line tool* yang bernama *Artisan* yang bisa digunakan untuk *packaging bundle* dan instalasi *bundle* melalui *command prompt* (Aminudin, 2015) ; dan
 - b. *Bootstrap*, adalah sebuah *open source toolkit* untuk pengembangan menggunakan HTML,CSS dan JS. Dengan cepat membuat prototipe dari ide kamu atau membuat seluruh aplikasi kamu dengan *Ssas variables* dan *mixins*, *grid system* yang responsif, komponen bawaan yang luas, dan *plugins* yang kuat yang dibuat dengan jQuery (Bootstrap, 2018);
4. *Command Prompt*, untuk memakai fungsi dari *framework Laravel*, maka dibutuhkan *command prompt* (CMD) untuk penggunaannya. Pengertian *Command Prompt* atau yang sering kita sebut dengan istilah CMD adalah suatu perintah DOS berbasis teks pada sistem operasi Windows (mastekno, 2017) ;
 - a. XAMPP, adalah sebuah paket kumpulan software yang terdiri dari apache, mysql, phpmyadmin, php, perl, *freetype2*, dll. Xampp berfungsi untuk memudahkan instalasi lingkungan php, dimana biasanya lingkungan pengembangan *web* memerlukan php, apache, mysql, dan phpmyadmin serta *software* yang terkait dengan pengemabngan web (Hanif & Fitriani, 2016).
 - b. Apache, adalah *server web* yang terkenal *secure* atau tingkat keamanannya terpercaya dibandingkan dengan *web server* IIS. Tetapi walaupun

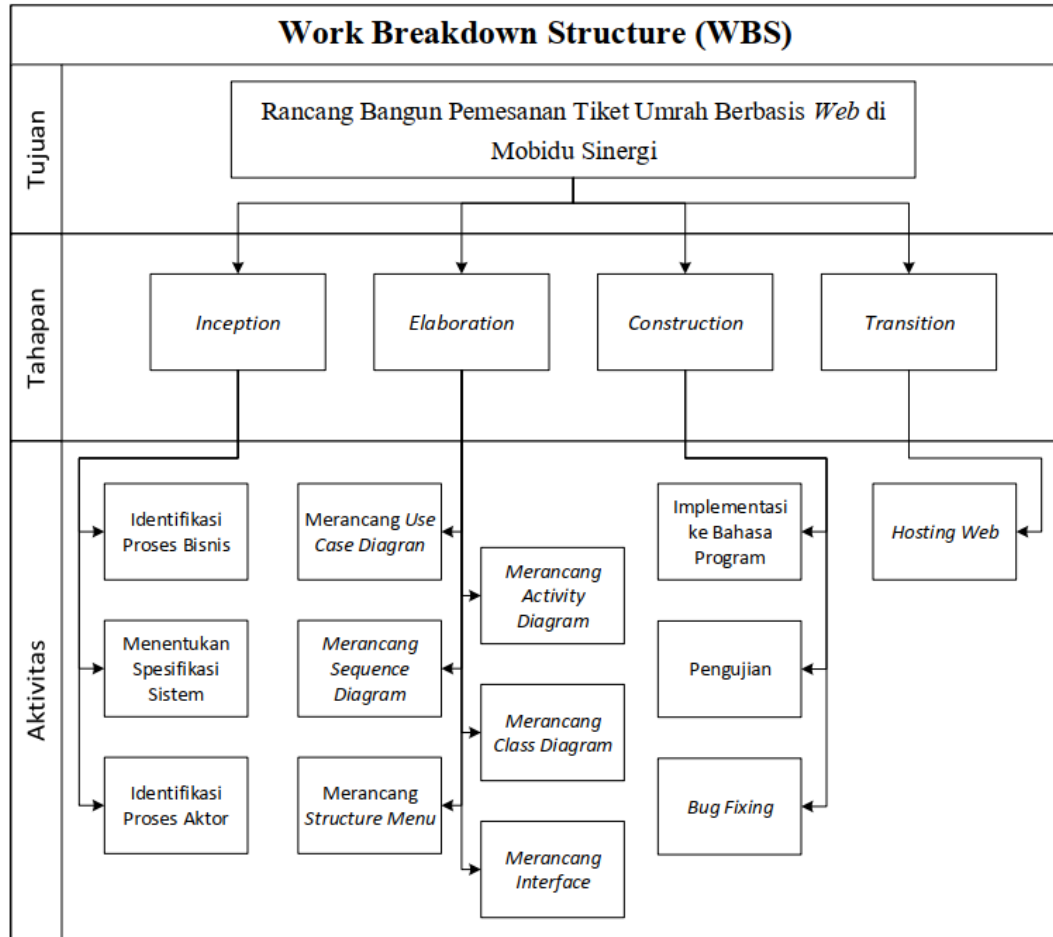
demikian kita bisa masuk ke direktori *web* yang menggunakan *server* Apache dengan bantuan *Google* (Juju & Studio, 2008).

- c. MySQL, berfungsi untuk mengolah *database* menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung/*support* dengan *database* MySQL (Anhar, 2010); dan
 - d. PHPMysqladmin, adalah suatu program *open source* berbasis *web*, dimana Program ini berguna untuk mengakses *database* MySQL. Dengan adanya Program ini akan mempermudah dan mempersingkat kerja kita, dengan beberapa kelebihan, antara lain pengguna awam tidak harus mengenal *syntax* SQL dalam pembuatan *database* dan tabel (Nugroho B. , 2005). Untuk menampilkan hasil dari editing program maka dibutuhkan GoogleChrome.
- 5. *Google Chrome*, adalah salah satu produk unggulan Google yang disebar secara canggih. Memiliki desain *interface* namun dapat mengakses *website* dengan lebih cepat, aman, dan praktis (Enterprise, 2009); dan
 - 6. *Composer*, adalah alat manajemen *dependency* pada PHP. *Composer* memungkinkan untuk membuat *library* dan menginstal atau mengubah secara otomatis tanpa anda harus menginstal *manual* pada *project* yang dibuat (IDCloudHost, 2016) .

3. METODOLOGI PEKERJAAN

3.1. *Work Breakdown Structur (WBS)*

Tahapan kerja meliputi aktivitas yang disajikan dalam skema WBS (*Work Breakdown Structure*).



Gambar 3. 1 *Work Breakdown Structure*

Kerja praktik ini dilaksanakan dengan mengikuti *Work Breakdown Structure* tersebut telah menghasilkan:

1. *Inception*

Pada tahap *inception* dimulai dengan melakukan identifikasi proses bisnis dimana pada aktivitas ini bertujuan untuk mengetahui identifikasi masalah aplikasi yang akan dibangun, kemudian dilanjutkan dengan aktivitas menentukan spesifikasi sistem yang bertujuan untuk menentukan spesifikasi sistem yang akan digunakan untuk membangun aplikasi dan yang terakhir adalah identifikasi aktor

untuk mengetahui siapa saja yang terlibat dalam proses bisnis serta siapa saja yang menggunakan aplikasi.

2. *Elaboration*

Pada tahap *elaboration* menggunakan permodelan UML (*Unified Modelling Language*) yang terdiri dari *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*. Pada tahap ini juga dilakukan perancangan merancang struktur menu untuk memperlihatkan sebuah gambaran umum aplikasi yang akan di rancang, lalu proses berikutnya adalah merancang interface dimana pada aktivitas ini menghasilkan tampilan *mock up* dari aplikasi.

3. *Construction*

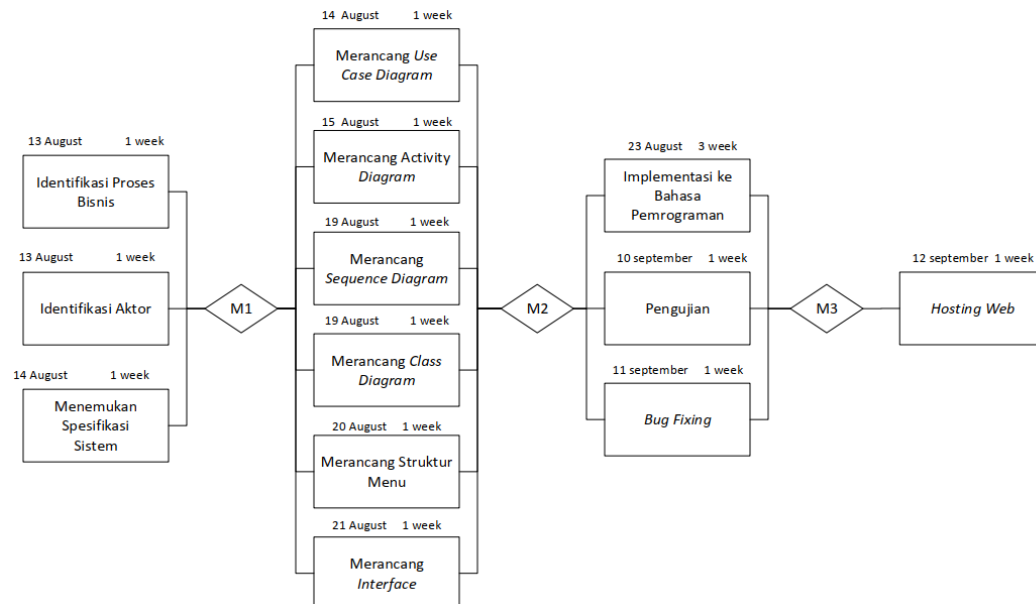
Pada tahapan *construction* implementasi pada bahasa program yang berfungsi untuk membuat tampilan aplikasi yang telah dibuat berfungsi sesuai dengan pemodelan UML (*Unified Modelling Language*) yang telah dibuat. Tahapan selanjutnya dari tahapan *contruction* yaitu pengujian aplikasi dimana aplikasi diuji untuk mencari kesalahan/*bug*, langkah selanjutnya adalah *bug fixing* yaitu berfungsi untuk memperbaiki masalah yang terjadi pada aplikasi selama proses pembuatan aplikasi ataupun sesudah aplikasi di buat

4. *Transition*

Pada tahapan *transition* dilakukan tahapan terakhir yaitu *hosting web* yang berguna untuk membuat aplikasi dapat diakses secara *global*.

3.2. Gantt Chart

Berdasarkan *Work Breakdown Structure* yang sudah digambarkan maka aktivitas penelitian akan di sajikan dalam diagram alur aktivitas dari Gambar 3.2 ;



Gambar 3. 2 Diagram Alur Aktivitas

Keterangan:

◇ M1 = Selesai Tahap Analisis

◇ M2 = Selesai Tahap Perancangan

◇ M3 = Selesai Tahap Implementasi

3.3. Sumber Daya Penelitian

Tabel 3. 1 Sumber Daya Penelitian

No	Aktivitas	Manusia	Perangkat	Tanggal
1.	Pengumpulan kebutuhan dan proses bisnis untuk aplikasi yang akan dibuat	Rifan Alamsyah, Sinta Nurfatonah & Yusef Maulana	Laptop, Seperangkat alat tulis	13 August 2019
2.	Identifikasi aktor	Sinta Nurfatonah &	Laptop, Microsoft Visio	13 August 2019

No	Aktivitas	Manusia	Perangkat	Tanggal
		Yusef Maulana		
3	Menemukan spesifikasi system	Sinta Nurfatonah & Yusef Maulana	Laptop, Microsoft Visio	14 August 2019
4	Merancang <i>usecase</i> diagram	Rifan Alamsyah, Sinta Nurfatonah & Leni Fitriani	Laptop, Microsoft Visio	14 August 2019
5	Merancang <i>activity diagram</i>	Rifan Alamsyah, Sinta Nurfatonah, & Leni Fitriani	Laptop, Microsoft Visio	15 August 2019
6	Merancang <i>Sequence diagram</i>	Rifan Alamsyah, Sinta Nurfatonah & Leni Fitriani	Laptop, Microsoft Visio	19 August 2019
7	Merancang <i>class diagram</i>	Rifan Alamsyah, Sinta Nurfatonah & Leni Fitriani	Laptop, Microsoft Visio	19 August 2019
8	Perancangan struktur menu dan <i>database</i>	Rifan Alamsyah & Sinta Nurfatonah	Laptop, Microsoft Visio, Sublime Text, XAMPP, GoogleChorme	20 August 2019
9.	Desain <i>interface</i> aplikasi	Rifan Alamsyah &	Laptop, Microsoft Visio,	21 August 2019

No	Aktivitas	Manusia	Perangkat	Tanggal
		Sinta Nurfatonah	BOOTSTRAP, Sublime Text, XAMPP, GoogleChorme	
10.	Implementasi desain ke bahasa pemrograman	Rifan Alamsyah & Halim Hasanudin	Laptop, BOOTSTRAP, Sublime Text, XAMPP, CMD, GoogleChorme	23 August 2019
11.	Ujicoba aplikasi	Sinta Nurfatonah, Muhammad Fahru R, & Ratu Delima	Laptop, XAMPP, GoogleChorme	10 September 2019
12.	Penanganan terjadinya error saat pembuatan aplikasi	Rifan Alamsyah & Halim Hasanudin	Laptop, Sublime Text, XAMPP, CMD, GoogleChorme	11 September 2019
13.	<i>Hosting web</i>	Rifan Alamsyah, Sinta Nurfatonah, & Halim Hasanudin	Laptop, GoogleChorme	12 September 2019

Bila dapat selesai dalam rentan waktu jam 08:30 – 17:00, pelaksanaan pembuatan aplikasi di lakukan di Mobidu. Bila diluar jam itu, maka pelaksanaan pembuatan aplikasi dilakukan dirumah.

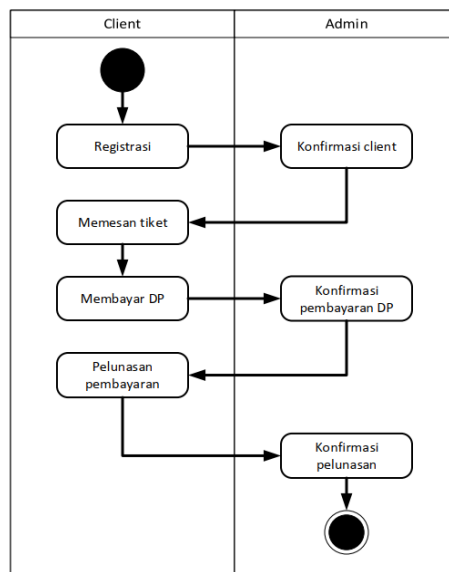
4. HASIL DAN PEMBAHASAN

4.1. Hasil Pekerjaan

Pada bagian ini, menjelaskan hasil pekerjaan yang telah di buat sebagai berikut.

1. **Inception**, pada tahap pertama ini dilakukan identifikasi proses bisnis, menentukan spesifikasi sistem dan mengidentifikasi aktor.

- a. Mengidentifikasi proses bisnis yang menghasilkan pembuatan fitur pemesanan tiket umrah yang meliputi registrasi yang dilakukan oleh *client* untuk akses terhadap aplikasi dan harus dikonfirmasi oleh *admin*, selanjutnya *client* bisa memesan serta memulai transaksi dengan membayar DP tiket, setelah *client* membayar DP *admin* dapat memeriksa serta mengkonfirmasi pembayaran dan langkah terakhir adalah pelunasan transaksi oleh *client* serta konfirmasi pembayaran ulang oleh *admin*. Penggambaran dapat dilihat pada gambar 4.1 berikut;



Gambar 4. 1 Proses Bisnis Rancang Bangun Pemesanan Tiket di Jalankan

Adapun deskripsi dari proses bisnis rancang bangun pemesanan tiket yang akan dijalankan dilampirkan pada bagian lampiran c.

- b. Menentukan spesifikasi sistem, untuk menentukan spesifikasi sistem diperlukan rincian persyaratan untuk sistem yang akan dibuat.

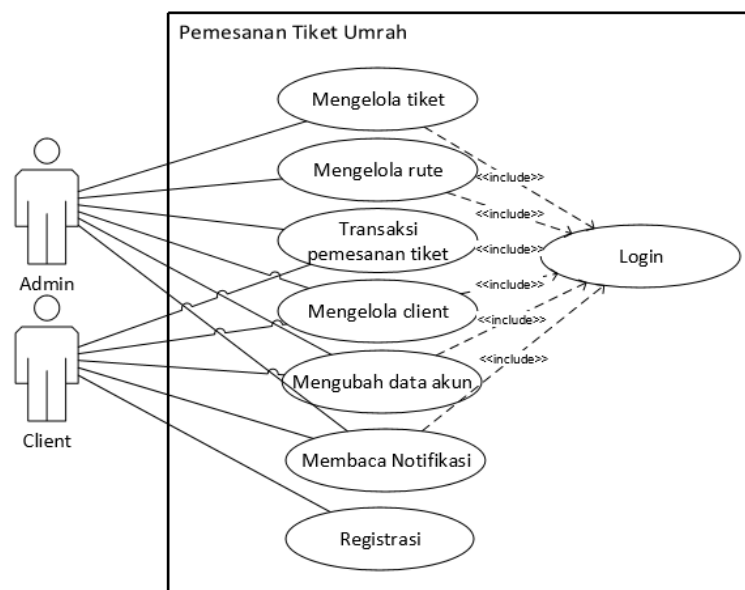
- 1) Rincian persyaratan, meliputi persyaratan tampilan, persyaratan sistem dan persyaratan pengembangan.
 - a) Persyaratan tampilan, aplikasi ini harus memiliki tampilan awal yang mengandung bagian untuk *user* yaitu tampilan login dan juga register yang ditujukan untuk *admin* dan *client*. Kemudian pada *admin* terdapat menu *Admin* yang didalamnya dapat melihat *Dashboard*, Mengelola Rute, Mengelola Tiket, Mengelola Transaksi, Mengelola *Client*, Melihat Notifikasi, dan Mengubah Informasi Akun. Kemudian *Client* terdapat menu *Client* yang didalamnya terdapat Melihat *Dashboard*, Membayar Transaksi, Melihat Notifikasi, dan Mengubah Informasi Akun;
 - b) Persyaratan sistem, untuk memenuhi permintaan dari pengguna, di spesifikasi sebagai berikut:
 - 1) Spesifikasi fungsional sistem
 - a. sistem menyediakan fitur Pemesanan Tiket, untuk disajikan secara *online*.
 - b. Terdapat hak akses untuk *user* yang ditujukan untuk *admin* dan *client* yang mempunyai hak akses berbeda.
 - 2) Spesifikasi non fungsional sistem
 - a. Kebutuhan perangkat lunak, dibutuhkan perangkat lunak untuk pembuatan aplikasi yaitu bahasa pemrograman PHP dengan *framework laravel*.
 - b. Kebutuhan perangkat keras, dibutuhkan perangkat komputer yang digunakan untuk pembuatan aplikasi ini serta pengembangan program (pengembang) dan orang yang menjadi pengguna aplikasi (*user*).
 - c) Persyaratan pengembangan
 - 1) Menggunakan *usecase diagram* sebagai gambaran interaksi antara pengguna dengan sistem;
 - 2) Perancangan *database* diimplementasikan menggunakan XAMMP; dan

3) dalam pengimplementasian bahasa program menggunakan *framework laravel*.

c. Mengidentifikasi Aktor, menghasilkan beberapa aktor yang teridentifikasi yaitu *user* yang terdiri dari *admin* dan *client*. *Admin* yang berfungsi sebagai pengelola aplikasi dan *client* berfungsi sebagai pengguna aplikasi.

2. **Elaboration**, pada tahap ini dirancang arsitektur sistem, sebagaimana terlampir pada lampiran c dengan rincian sebagai berikut:

a. *Use case diagram* berikut adalah *use case diagram* untuk aplikasi pemesanan tiket umrah berbasis *web*:



Gambar 4. 2 Use Case Diagram Aplikasi pemesanan tiket umrah berbasis Web

Adapun deskripsi dari proses *usecase* yang di bangun sebagai berikut :

(1) *use case login* menggambarkan proses hak akses *admin* dan *client*; (2) *use case* mengelola tiket menggambarkan proses pengelolaan tiket oleh *admin*; (3) *use case* mengelola rute menggambarkan proses pengelolaan rute yang dilakukan di *admin*; (4) *use case* transaksi pemesanan tiket menggambarkan proses transaksi pemesanan tiket yang dikelola oleh *admin* dan proses pembayaran tiket oleh *client*; (5) *use case* mengelola *client* menggambarkan proses pengeloaan *client* yang terdaftar oleh *admin* dan *client*; (6) *use case* mengubah data akun menggambarkan proses pengelolaan akun yang dilakukan *client* dan *admin*; (7) *use case* membaca

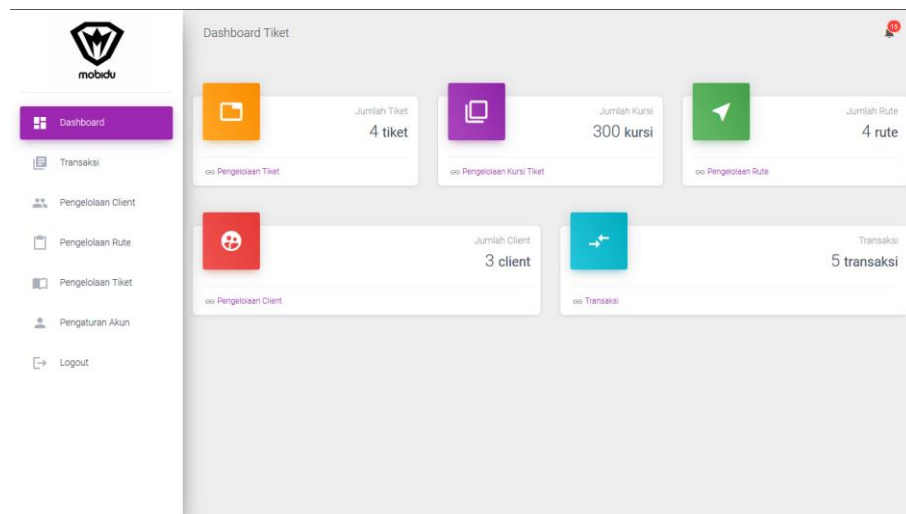
notifikasi menggambarkan proses melihat status notifikasi yang dilakukan *client* dan *admin*; (8) *use case* registrasi menggambarkan proses pendaftaran yang dilakukan oleh *client*;

- b. *Activity diagram*, sebagaimana yang terlampir pada lampiran c yang menggambarkan aktivitas yang bisa dilakukan oleh *admin* dan *user*. *Activity diagram* berdasarkan skenario *use case* yang menggambarkan alur kerja dari *admin* maupun *user* didalam sebuah sistem yang meliputi : (1) *login* yaitu aktivitas awal untuk masuk ke dalam program yang dilakukan *admin* dan *client*; (2) Registrasi yaitu aktivitas pendafrtan untuk *user*; (3) Pengelolaan *client* yaitu aktivitas pengeloaan *client* yang dilakukan oleh *admin*; (4) Pengelolaan tiket yaitu aktivitas pengeloaan tiket yang dilakukan oleh *admin*; (5) Pengelolaan rute yaitu aktivitas pengeloaan rute tiket yang dilakukan oleh *admin*; (6) transaksi pemesanan yaitu aktivitas pemesanan tiket yang dilakukan oleh *admin* dan *client*; (7) Merubah data akun yaitu aktivitas perubahan data yang dilakukan oleh *admin* dan *client*; (8) Membaca notifikasi yaitu aktivitas membaca notifikasi sebagai pemberitahuan yang dilakukan oleh *admin* dan *client*;
- c. *Sequence diagram*, sebagaimana yang terlampir pada lampiran c yang meliputi aktivitas: (1) *login* yang terdiri dari objek *admin*, *main*, *antarmuka*, *validasi* dan *database user*; (2) Mengelola tiket yang terdiri dari objek *admin*, *main*, *antarmuka*, *ticketcontroller* dan *databaseticket*; (3) Mengelola rute yang terdiri dari objek *admin*, *main*, *antarmuka*, *rutecontroller* dan *databaserute*; (4) Transaksi pemesanan tiket yang terdiri dari objek *client*, *admin*, *main*, *antarmuka*, *dashboardcontroller*, *databasetransaction* dan *databasenotification*; (5) Mengelola *client* yang terdiri dari objek *admin*, *main*, *antarmuka*, *clientcontroller* dan *databaseuser*; (6) Mengubah data akun yang terdiri dari objek *admin/client*, *main*, *antarmuka*, *clientcontroller*, dan *databaseuser*; (7) Membaca notifikasi yang terdiri dari objek *admin/client*, *main*, *antarmuka*, *notificationcontroller*, dan *databasenotification*; (8) Registrasi yang terdiri dari objek *client*, *main*, *antarmuka*, *validasi*, *RegisterController*, *database user*, dan *database notification*;

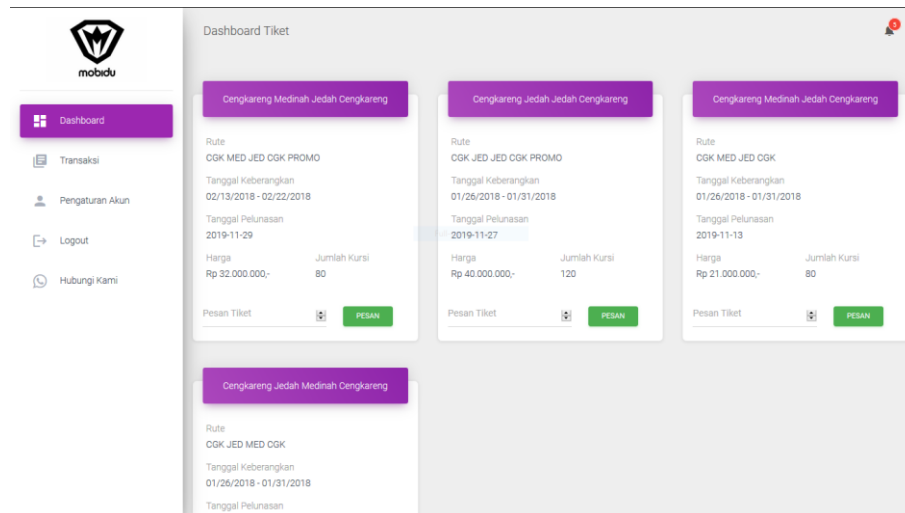
- d. *Class diagram*, yang menggambarkan hubungan antara *class user*, *class notification*, *class rute*, *class ticket* dan *class transaction* dengan koneksi basis data;
 - e. Merancang struktur menu, yang menggambarkan rincian sistem berupa struktur menu. sebagaimana yang terlampir pada lampiran c yang meliputi: tampilan *admin* terdiri (1) menu *login* sebagai hak akses dari *admin*; (2) menu kelola *client*; (3) menu kelola rute; (4) menu kelola tiket; dan (5) menu kelola transaksi. Sedangkan untuk tampilan *client* terdiri dari (1) *register* untuk pendaftaran *client*; (2) menu *login* sebagai hak akses dari *client*; (3) menu pemesanan tiket untuk memulai transaksi pemesanan tiket; dan (4) menu pembayaran tiket untuk membayar transaksi pemesanan tiket; dan
 - f. Merancang tampilan *interface*, sebagaimana yang terlampir pada lampiran c yang menggambarkan tampilan sementara: tampilan *interface admin* yang terdiri dari (1) *dashboard admin*; (2) halaman transaksi; (3) halaman pengelolaan *client*; (4) halaman pengelolaan rute; (5) halaman pengelolaan tiket; (6) halaman pengaturan akun; dan (7) halaman notifikasi. Sedangkan tampilan *interface* untuk *client* terdiri dari (1) *dashboard client*; (2) halaman transaksi; (3) halaman pengaturan akun; dan (4) halaman notifikasi;
3. **Construction**, pada tahapan ketiga ini mengaplikasikan hasil proses analisis ke bahasa program serta pengujian demi menjaga kualitas program. Adapun deskripsi dari proses bisnis rancang bangun pemesanan tiket yang akan dijalankan terlampir pada lampiran c.
- a. Implementasi ke bahasa program menghasilkan kode dari pemesanan tiket umrah berbasis *web* yang dibangun dengan menggunakan bahasa pemrograman PHP dengan bantuan *framework laravel*;
 - b. Pengujian sistem, kesimpulan dari aktivitas ini adalah sistem berhasil melewati skenario uji 100%, yang artinya fitur-fitur penting dalam kebutuhan aplikasi dapat berjalan. Yang diuji adalah bagian-bagian kelas.

- c. *Bug fixing*, merupakan proses pengkodean ulang dari web yang dibuat apabila terdapat kesalahan atau kegagalan, baik dari logika program ataupun kesalahan penulisan kode.
- 4. **Transition**, pada tahapan keempat ini, menghasilkan *web* yang siap diupload secara global.
 - a. *Hosting web*, kesimpulan dari aktivitas ini adalah proses mengunggah *web* yang sudah berfungsi dengan baik secara global sehingga dapat diakses oleh semua orang.

Adapun tujuan pembangunan aplikasi pemesanan tiket umrah berbasis web ini menggunakan bahasa pemrograman PHP dengan *framework laravel* sudah berhasil dicapai dengan bukti sebagaimana tampak pada gambar pada gambar 4.3. Kemudian untuk tampilan *client* pada gambar 4.4. Pengkodean bahasa pemrograman PHP menggunakan *framework laravel* tampak pada gambar 4.5 dan 4.6. Pengembangan aplikasi dibuat oleh 2 orang dengan peranan sebagai programmer dan sistem analis.



Gambar 4. 3 Tampilan Admin Aplikasi Pemesanan Tiket Umrah



Gambar 4. 4 Tampilan *client* Aplikasi Pemesanan Tiket Umrah

Dalam *laravel* terdapat fungsi `@section()` dan `@yield()`. Seperti namanya `@section()`, mendefinisikan bagian konten, sedangkan `@yield()` digunakan untuk menampilkan konten dari bagian tertentu (Otwell, 2019). Sebagai contoh dalam *index* aplikasi terdapat fungsi `@yield()` yang diberi istilah '*content*', selanjutnya dalam bagian tiket terdapat fungsi `@section()` yang di beri istilah sesuai dengan istilah *content* dalam fungsi `@yield()`, artinya bagian *client* bisa tampil di dalam *index* aplikasi sebagaimana tampak pada kode dibawah ini.

Rangkaian kode untuk modul *client* sebagai berikut.

```
@extends('layout/main')
@section('title','Pengelolaan Client')
@section('container')
<div class="card">
    <div class="card-header card-header-primary">
        <h4 class="card-title">Pendaftaran Client</h4>
    </div>
    <div class="card-body table-responsive">
        <table class="table table-hover data" id="client1_table">
            <thead class="text-primary">
                <th>#</th>
                <th>NIK</th>
                <th>Nama</th>
                <th>Email</th>
                <th>Perusahaan</th>
                <th>Aksi</th>
            </thead>
            <tbody>
            </tbody>
        </table>
    </div>
</div>
<div class="card">
    <div class="card-header card-header-success">
        <h4 class="card-title">Client Terdaftar</h4>
    </div>
```

```

<div class="card-body table-responsive">
    <table class="table table-hover" id="client2_table">
        <thead class="text-primary">
            <th>#</th>
            <th>NIK</th>
            <th>Nama</th>
            <th>Email</th>
            <th>Perusahaan</th>
            <th>Aksi</th>
        </thead>
        <tbody>
        </tbody>
    </table>
</div>
</div>
@include('client/accClient')
@include('client/detail')
@include('client/edit')
@endsection

```

Rangkaian kode untuk modul *index* aplikasi sebagai berikut.

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSRF Token -->
    <meta name="csrf-token" content="{{ csrf_token() }}">
</head>
<body>

    <main class="py-4">
        @yield('content')
    </main>
</div>
</body>
</html>

```

4.2. Pengetahuan dan Keterampilan Empiris

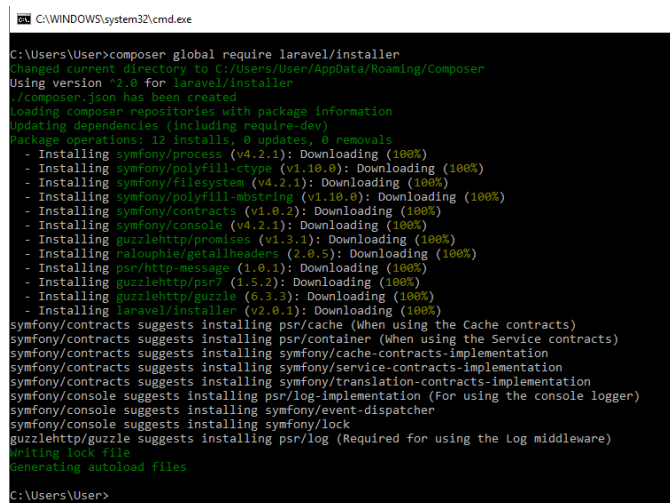
Pengembangan *website* menjadi tugas yang diberikan oleh Mobidu, berbekal matakuliah basis data, sistem basis data pada semester 4 dan 5, sebagai pengetahuan untuk pembuatan *database* dan *website*. Kemudian pembekalan matakuliah analisis dan desain dari kampus, sebagai analisis kebutuhan yang diinginkan oleh klien untuk menyelesaikan pengembangan *website*. Jika *website* ini selesai dalam waktu 1 bulan sejak hari pertama masuk kerja, maka pihak Mobidu memberikan kesempatan untuk mempublikasikannya ke internet.

Adanya hambatan dari perusahaan ketika pengerjaan *website*, yaitu disaat perusahaan sedang menangani pembuatan *website* / aplikasi yang cukup banyak, dimana pihak mobidu sendiri tidak bisa mengontrol dan memberi arahan secara

penuh, sehingga proses pengerjaan terkadang tidak sesuai dengan jadwal yang telah ditetapkan.

Dalam kerja praktik ini diperoleh pengetahuan dan keterampilan empiris seputar penggunaan *laravel* yang belum diajarkan di Sekolah Tinggi Teknologi Garut, serta menjadi salah satu faktor tidak selesainya pekerjaan sesuai target. Berikut ini cara *install laravel*.

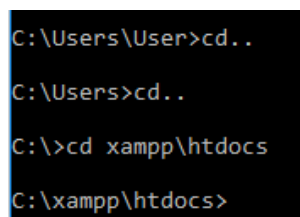
1. Pastikan *composer* sudah di *install*;
2. Buka CMD dan ketikkan perintah seperti pada gambar lalu ketik *enter*, terlihat pada gambar 4.5;



```
CA\WINDOWS\system32\cmd.exe
C:\Users\User>composer global require laravel/installer
Changed current directory to C:\Users\User\AppData\Roaming\Composer
Using version ^2.0 for laravel/installer
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 12 installs, 0 updates, 0 removals
- Installing symfony/process (v4.2.1): Downloading (100%)
- Installing symfony/polyfill-ctype (v1.10.0): Downloading (100%)
- Installing symfony/filesystem (v4.2.1): Downloading (100%)
- Installing symfony/polyfill-mbstring (v1.10.0): Downloading (100%)
- Installing symfony/contracts (v1.0.2): Downloading (100%)
- Installing symfony/console (v4.2.1): Downloading (100%)
- Installing guzzlehttp/promises (v1.3.1): Downloading (100%)
- Installing ralouphie/getallheaders (2.0.5): Downloading (100%)
- Installing psr/http-message (1.0.1): Downloading (100%)
- Installing guzzlehttp/psr7 (1.5.2): Downloading (100%)
- Installing guzzlehttp/guzzle (6.3.3): Downloading (100%)
- Installing laravel/installer (v2.0.1): Downloading (100%)
symfony/contracts suggests installing psr/cache (When using the Cache contracts)
symfony/contracts suggests installing psr/container (When using the Service contracts)
symfony/contracts suggests installing symfony/cache-contracts-implementation
symfony/contracts suggests installing symfony/service-contracts-implementation
symfony/contracts suggests installing symfony/translation-contracts-implementation
symfony/console suggests installing psr/log-implementation (For using the console logger)
symfony/console suggests installing symfony/event-dispatcher
symfony/console suggests installing symfony/lock
guzzlehttp/guzzle suggests installing psr/log (Required for using the Log middleware)
Writing lock file
Generating autoload files
C:\Users\User>
```

Gambar 4. 5 Install Laravel

3. Setelah proses instalasi selesai, kemudian masuk ke *directory* yang ingin di simpan *project laravel* nya, terlihat pada gambar 4.6;



```
C:\Users\User>cd ..
C:\Users>cd ..
C:\>cd xampp\htdocs
C:\xampp\htdocs>
```

Gambar 4. 6 Memilih Directory

4. Lalu ketikkan perintah *laravel new project* pada *directory* yang diinginkan sehingga keluar tampilan seperti pada gambar 4.7;


```
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: fideloper/proxy
Discovered Package: laravel/nexmo-notification-channel
Discovered Package: laravel/slack-notification-channel
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
Application ready! Build something amazing.
```

Gambar 4. 7 Membuat *Project Laravel*

5. Setelah proses instalasi selesai, kemudian masuk ke *directory project* yang sudah di buat seperti pada gambar 4.8;

```
C:\xampp\htdocs>cd project
```

Gambar 4. 8 Masuk Ke *Directory Project*

6. Setelah masuk ke *directory project* ketikkan perintah untuk menjalankan *project laravel* yang sudah di buat seperti pada gambar 4.9; dan

```
C:\xampp\htdocs\project>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

Gambar 4. 9 Menjalankan *Project*

7. Setelah itu buka *web browser chrome* dan masukan *link*, untuk melihat hasil yang sudah di buat seperti pada gambar 4.10;



Gambar 4. 10 Tampilan *Project Awal Laravel*

Dengan menggunakan *framework laravel*, bisa mempercepat proses editing serta membuat kerangka kerja program agar lebih mudah dipakai secara bersama atau tim serta dokumentasi yang lengkap dari *laravel*. Kerja praktik di Mobidu menambah pengalaman kerja yang berharga serta penambah wawasan

yang lebih luas lagi, mengenal bahasa pemrograman lain, serta waktu *deadline* pekerjaan, menjadi gambaran tersendiri bagi dunia sebenarnya setelah lulus dari kampus.

4.3. Etika Kerja

Mobidu merupakan salah satu unit perusahaan yang didirikan dan dikelola oleh beberapa alumni mahasiswa Sekolah Tinggi Teknologi Garut yang memberikan keleluasaan untuk beribadah salah satunya ketika adzan berkumandang langsung melaksanakan shalat berjamaah di masjid terdekat, kemudian jam istirahat yang dilangsungkan ketika sesudah melaksanakan shalat dan jam kerja seperti tempat kerja pada umumnya. Etika dalam waktu kerja, di Mobidu berlangsung selama 7 jam setiap harinya dan ketentuan jam kerja ini telah diatur yaitu 7 jam kerja dalam 1 hari atau 35 jam kerja dari hari senin sampai dengan hari jum'at;

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil dari rancang bangun aplikasi pemesanan tiket umrah berbasis *web* yang telah dilakukan, maka berikut kesimpulan yang dapat diambil :

1. Penelitian ini menghasilkan sebuah aplikasi pemesanan tiket berbasis web yang di peruntukkan untuk perusahaan mobidu sinergi.
2. Pemesanan tiket tidak perlu lagi dilakukan secara manual tetapi cukup menggunakan aplikasi mulai dari pendaftaran *client* sampai pembayaran.
3. Memberikan kemudahan dalam membantu pekerjaan *admin* dalam mengelola transaksi serta pencatatan *client* dan memudahkan *client* dalam pembayaran tiket umrah.

5.2. Saran

Berdasarkan kesimpulan yang telah dibuat, adapun saran yaitu meningkatkan kemudahan dan pelayanan aplikasi pemesanan tiket umrah berbasis *web* di mobidu bisa dilakukan pengembangan fitur seperti cetak laporan untuk perekapan transaksi pembayaran dan notifikasi email untuk pengingat tenggat waktu pembayaran.

Adapun hambatan pengetahuan dari bahasa pemrograman maupun metode analisis pembuatan perangkat lunak yang digunakan di Mobidu Sinergi yang menggunakan bahasa pemrograman PHP dengan *framework laravel*. Diharapkan bagi Sekolah Tinggi Teknologi Garut untuk lebih memperbarui pengetahuan dari pemrograman atau memberikan pelatihan *framework* seputar pemrograman, sebagai pembekalan bagi para mahasiswa pada saat kerja praktik atau ketika sesudah lulus dari kampus.

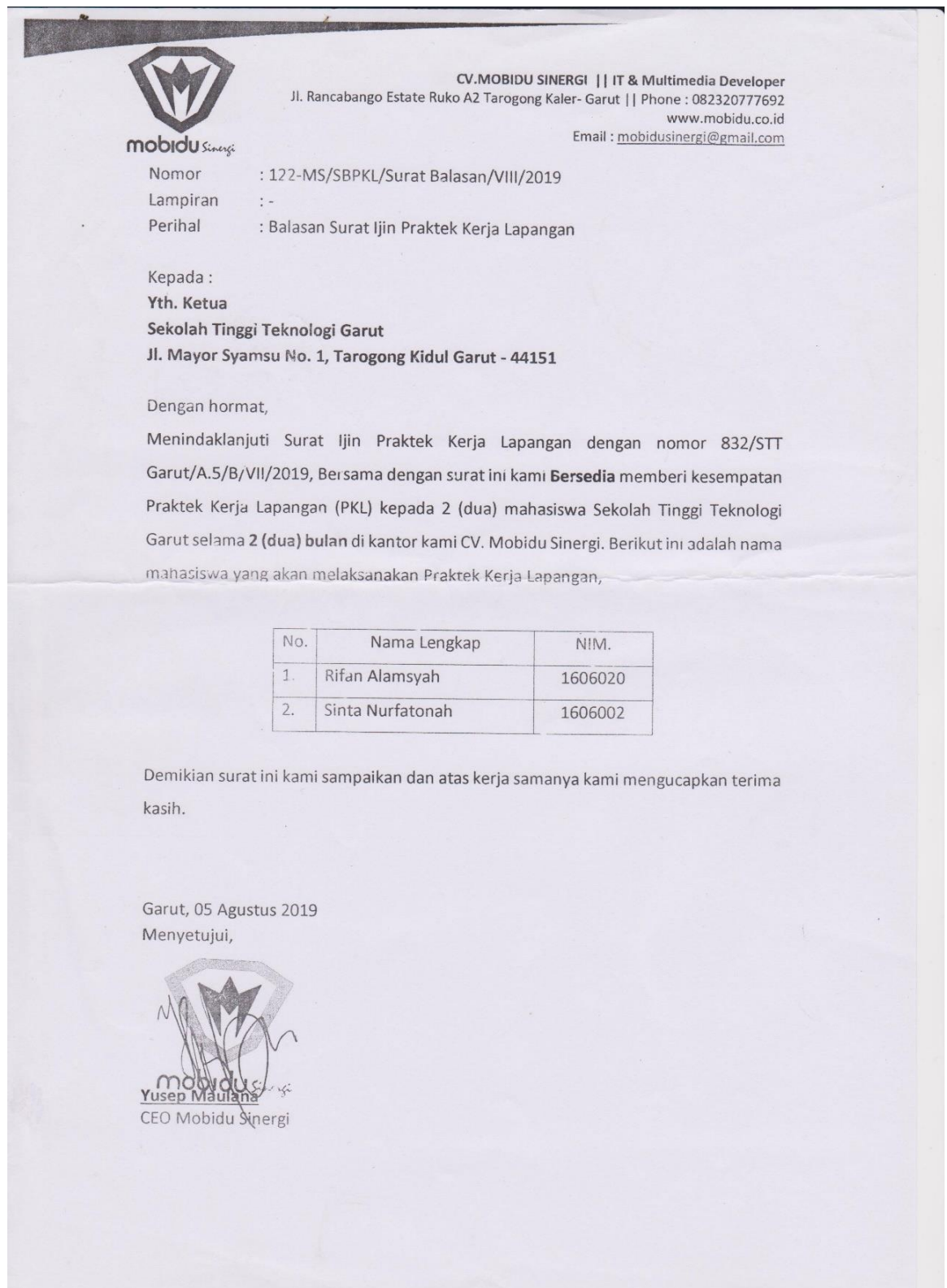
DAFTAR PUSTAKA

- Al-Habsi, M. (1999). *Fiqh Praktis*. Bandung: Mizan.
- Aminudin. (2015). *Cara Efektif Belajar Framework Laravel* (L. Hakim, Ed. Yogyakarta: Lokomedia.
- Anhar. (2010). *Panduan menguasai PHP & MySQL Secara Otodidak*. Jakarta: Media kita.
- Enterprise, J. (2016). *Pemrograman Bootstrap Untuk Pemula*. Jakarta: PT Elex Media Komputindo.
- Ependi U., Kunang Y, & Nofika S,. (2013). Implementasi metode Rational Unified Process pada Mobil Digital Library. *Jurnal ilmiah MATRIK* , 33-34.
- Hanif, I. M., & Fitriani, L. (2016). Perancangan Aplikasi Perangkat Lunak Sistem Informasi Rekam Medis Berbasis *Client-Server* Studi Kasus Klinik Cipanas. *Journal Algoritma*, <https://drive.google.com/file/d/0B0EdWn-d3T9xZDRoUUdDZTVuOEE/view>.
- Hidayat, R. (2010). *Cara Praktis Membangun Website Gratis*. Jakarta: PT Elex Media Komputindo.
- IDCloudHost. (05, Juni 2016). Retrieved from idcloudhost: from <https://idcloudhost.com/pengertian-dan-manfaat-composer-bagi-developer/>
- Juju, D. &. (2008). *Teknik Rahasia Keyword Google Pemula*. Jakarta: PT Elex Media Komputindo.
- Kadir, A. (2002). *Dasar Pemrograman WEB Dinamis Menggunakan PHP (2 ed.)*. Yogyakarta: Andi.
- Mastekno. (2017, Desember 7). Retrieved from Matekno: <https://www.mastekno.com/id/pengertian-fungsi-command-prompt-cmd-lengkap/>
- Nugroho, B. (2005). *Database Relasional dengan MySQL*. Yogyakarta: Andi.
- Pressman, R. S. (2012). *Rekayasa Perangkat Lunak*. Yogyakarta: ANDI.

- Putratama, S. V. (2016). *Pemrograman Web dengan Menggunakan PHP dan Framework Codeigniter*. Yogyakarta: deepublish.
- Rainer, K., , & Cegielski. (2011). *Introduction to Information System* . United State: United State of America.
- Rerung, R. R. (2018). *Pemrograman Web Dasar*. Yogyakarta: CV BUDI UTAMA.
- Rosa A.S, & Salahudin . (2018). *Rekayasa Perangkat Lunak*. Bandung: Informatika Bandung.
- Sukamto, R. A., , & Shalahuddin, M. (2018). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Sunyoto, A. (2017). *AJAX Membangun Web dengan Teknologi Asynchrone JavaScript dan XML*. Yogyakarta: ANDI.
- Warsito, A. B, Yusup, M., & Yulianto. (2014). Kajian Yii Framework Dalam Pengembangan. *JOURNAL CCIT*, 439.

LAMPIRAN A : LANDASAN PEKERJAAN

1. Surat Balasan




Lampiran Surat Balasan

2. Formulir Intansi

**PROGRAM KERJA PRAKTEK INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI GARUT**


FORMULIR INSTANSI

1	Nama Instansi	CV. Mobidu Sinergi	
2	Alamat Instansi	Jl. Rantabango Estate Ruko A2	
	Kecamatan	Tarongong Kaler	
	Kabupaten	Garut	
	Provinsi	Jawa Barat	
3	Jenis Instansi *	<input checked="" type="checkbox"/>	Perusahaan / Swasta
		<input type="checkbox"/>	Pemerintahan
		<input type="checkbox"/>	Perguruan Tinggi
		<input type="checkbox"/>	Komunitas / Yayasan
4	Nama Unit		
5	Nama Kepala Unit	Yusep Maulana	
6	Nama P/L **	Ratu Delims	
7	No. Telp P/L **	081220055522	
8	Jenis Pekerjaan *	<input type="checkbox"/>	Arsitek Data (Perancangan Basis Data)
		<input type="checkbox"/>	Administrator Basis Data (Pengelolaan Sistem Basis Data)
		<input type="checkbox"/>	Pemrogram Basis Data (Penerapan Rancangan Basis Data)
		<input checked="" type="checkbox"/>	Software Engineer (Pemodelan Analisis dan Perancangan)
		<input type="checkbox"/>	Pemrogram Objek (Penerapan Rancangan Perangkat Lunak)
		<input type="checkbox"/>	Analisis Program (Dokumentasi dan Pengujian Program)
		<input type="checkbox"/>	Perancang Jaringan (Pengembangan Jaringan)
9	Rentang Waktu *	<input type="checkbox"/> 1 (satu) bulan	<input checked="" type="checkbox"/> 2 (dua) bulan
10	Jumlah hari kerja	5	Hari per minggu
11	Jumlah jam kerja	7	Jam per hari
Ditandatangani di			
Catatan : *) Beri tanda cek **) P/L : Pembimbing Lapangan ***) Pejabat instansi yang berwenang disertai cap instansi		Mengetahui ***  Yusep Maulana	

Lampiran Formulir Intansi

LAMPIRAN B : PRESENSI KERJAAN

1. Kartu Kerja Praktek



SEKOLAH TINGGI TEKNOLOGI GARUT
PROGRAM STUDI INFORMATIKA
<http://informatika.sttgarut.ac.id>

BERITA ACARA BIMBINGAN KERJA PRAKTIK

A DATA MAHASISWA			
1	Nama Mahasiswa	Rifan Alamsyah	
2	Nomor Induk	1606020	
3	Kontrak Kerja Praktik	[✓] Ganjil [] Genap, tahun akademik 2019/2020	
4	Tujuan Pekerjaan		
B DATA PEMBIMBING			
1	Pembimbing Lapangan	Ratu Delima, S.T.	
	Jabatan	Administrasi	
	Instansi / Unit Kerja	CV. Mobidu Sinergi / Bidang IT	
2	Pembimbing Akademik	Leni Fitriani, M.kom	
	Nomor Induk	0429058704	
	Jabatan		
C DATA BIMBINGAN INDUSTRIAL / MINGGU			
NO	TANGGAL / JAM	KEGIATAN BIMBINGAN	PARAF
1	15-Agustus-2019 / 16:06 wib	Presentasi normalisasi + Tampilan Aplikasi Awal	[Signature]
2	23-Agustus-2019 / 16:16 wib	Pembuatan ERD, use case, & Flowchart + Tampilan dan database Aplikasi	[Signature]
3	30-Agustus-2019 / 16:25 wib	Revisi ERD, use case & Flowchart + Tampilan aplikasi persediaan client, rute tiket	[Signature]
4	06-September-2019 / 16:09 wib	Revisi tampilan di tambah notif dan Revisi ERD, use case & flowchart.	[Signature]
5	11-September-2019 / 13:00 wib.	Aplikasi beres + Hosting	[Signature]
6			✓
7			
8			

Berita Acara Bimbingan Kerja Praktik | 1 / 2

Lampiran Kartu Bimbingan Lapangan Rifan Alamsyah



BERITA ACARA BIMBINGAN KERJA PRAKTIK

A DATA MAHASISWA			
1	Nama Mahasiswa	Sinta Nurfatona	
2	Nomor Induk	1606002	
3	Kontrak Kerja Praktik	[X] Ganjil [] Genap, tahun akademik 2019/2020	
4	Tujuan Pekerjaan		
B DATA PEMBIMBING			
1	Pembimbing Lapangan	Ratu Delima, S.T.	
	Jabatan	Administrasi	
	Instansi / Unit Kerja	c.v. Mobidu sinergi / Bidang IT	
2	Pembimbing Akademik	Leri Fitriani, M.kom	
	Nomor Induk	042 905 8704	
	Jabatan		
C DATA BIMBINGAN INDUSTRIAL / MINGGU			
NO	TANGGAL / JAM	KEGIATAN BIMBINGAN	PARAF
1	15-Agustus-2019/ 16:06 wib	Presentasi normalisasi + Tampilan Aplikasi awal	[Signature]
2	23-Agustus-2019/ 16:16 wib	Pembuatan ERD, UML, use case, & flowchart + Tampilan dan database Aplikasi	[Signature]
3	30-Agustus-2019/ 16:25 wib	Revisi ERD, use case & flowchart + Tampilan aplikasi pengelolaan klien, rute & tiket	[Signature]
4	06-September-2019/ 16:07 wib	Revisi tampilan ditambah notifikasi dan Revisi ERD, use case & flowchart.	[Signature]
5	11-September-2019/ 13.00 wib	Aplikasi beres + Hosting.	[Signature]
6			
7			
8			

Lampiran Kartu Bimbingan Lapangan Sinta Nurfatona

LAMPIRAN C : HASIL PEKERJAAN

1. Identifikasi Proses Bisnis

No	Aktivitas	Deskripsi
1.	Registrasi <i>client</i>	<i>Client</i> melakukan registrasi agar mendapatkan izin untuk mengakses aplikasi serta melakukan transaksi.
2.	Konfirmasi <i>client</i>	<i>Client</i> yang sudah mendaftar dapat di kelola oleh <i>admin</i> apakah dapat diterima atau ditolak.
3.	Transaksi pemesanan tiket	<i>Client</i> dapat memesan tiket yang disediakan oleh <i>admin</i> .
4.	Membayar DP	<i>Client</i> harus membayar DP dari tiket yang telah dipesan untuk melanjutkan aplikasi.
5.	Konfirmasi pembayaran DP	<i>Admin</i> memeriksa bukti pembayaran DP yang diserahkan <i>client</i> dan mengubah status pembayaran.
6.	Pelunasan pembayaran	<i>Client</i> harus membayar pelunasan dari tiket dalam tenggat waktu yang telah ditentukan.
7.	Konfirmasi pelunasan	<i>Admin</i> memeriksa bukti pelunasan yang diserahkan <i>client</i> dan mengubah status pembayaran.

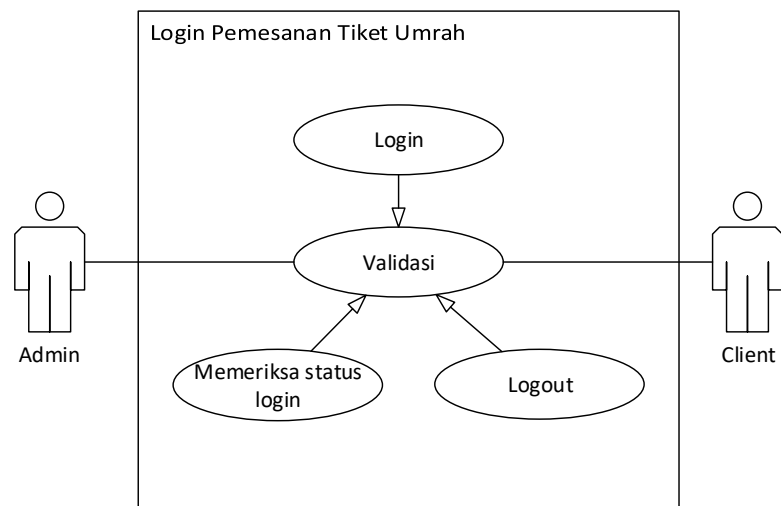
a. Aktivitas Aktor

Aktor	Aktivitas
<i>Admin</i>	<ul style="list-style-type: none">- Membuka web.- Login aplikasi.- Mengelola <i>client</i>,- Mengelola rute yang tersedia.- Mengelola tiket yang dapat dipesan.- Mengelola transaksi pemesanan tiket.

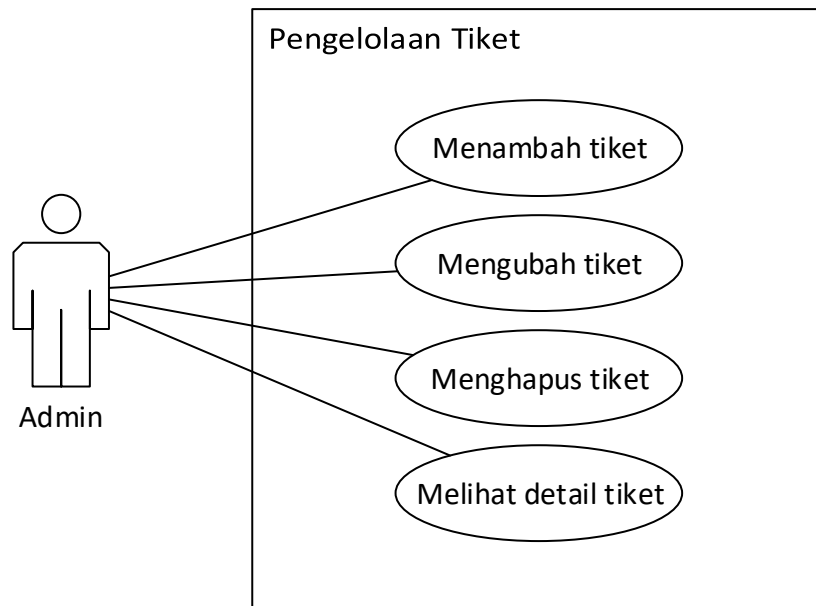
Aktor	Aktivitas
	<ul style="list-style-type: none"> - Melihat notifikasi. - Mengatur data akun pribadi.
<i>Client</i>	<ul style="list-style-type: none"> - Membuka web. - Login aplikasi. - Melakukan registrasi untuk menjadi <i>client</i>. - Melihat daftar tiket yang tersedia. - Melakukan transaksi pemesanan tiket. - Melihat notifikasi. - Pengaturan akun pribadi.

2. Rancangan Use Case Diagram

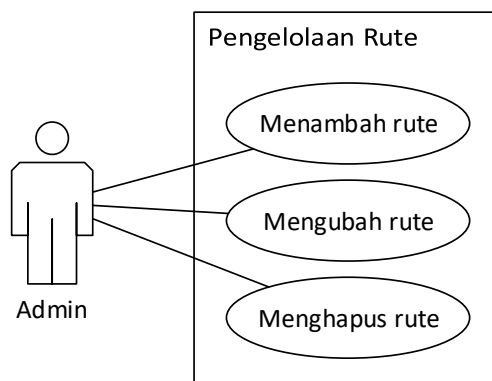
a. Use case login



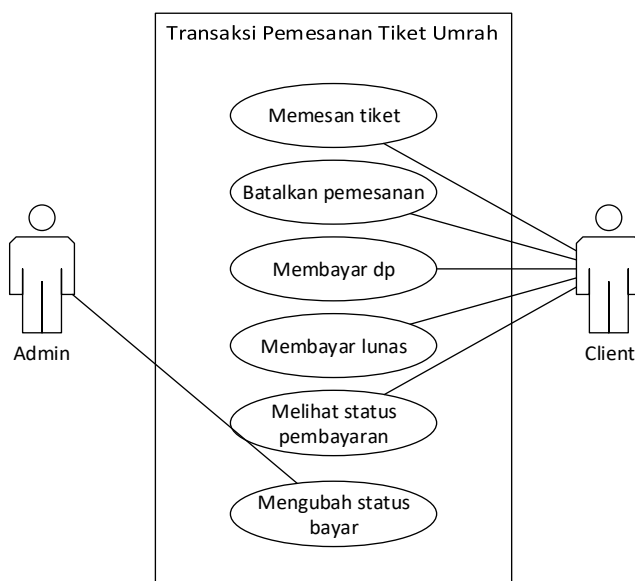
b. *Use case mengelola tiket*



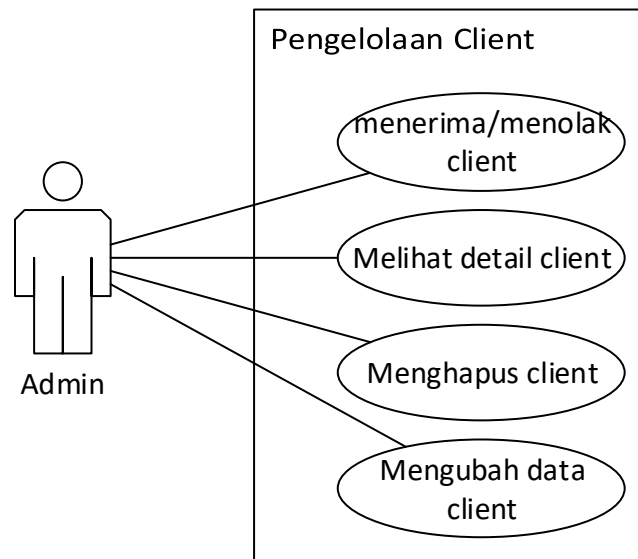
c. *Use case mengelola rute*



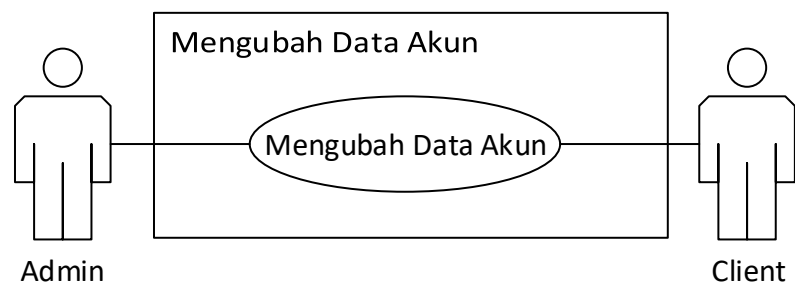
d. *Use case mengelola transaksi*



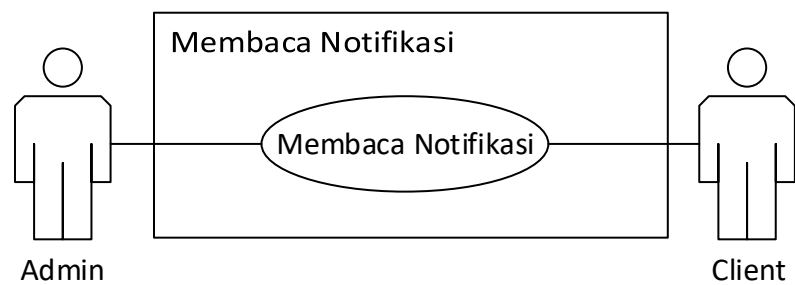
e. *Use case mengelola client*



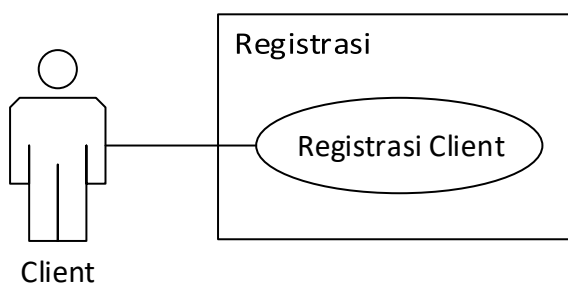
f. *Use case mengubah data akun*



g. *Use case membaca notifikasi*



h. *Use case registrasi client*



3. Skenario *Use Case*

a. Skenario *use case login*

Identifikasi	
Nama	Login
Tujuan	Agar dapat mengelola aplikasi dengan <i>role admin</i> .
Aktor	<i>Admin/Client</i>
Skenario	
Aksi Aktor	Reaksi Sistem
1. Memasukkan <i>username</i> dan <i>password admin</i> tidak valid	2. Memeriksa <i>valid</i> tidaknya data masukan
	3. Menampilkan pesan <i>login</i> tidak <i>valid</i>
4. Memasukkan <i>username</i> dan <i>password admin</i> yang <i>valid</i>	5. Memeriksa <i>valid</i> tidaknya data masukan
	6. Masuk ke aplikasi dengan <i>role admin</i> .
7. Memasukkan <i>username</i> dan <i>password client</i> tidak valid	8. Memeriksa <i>valid</i> tidaknya data masukan
	9. Menampilkan pesan <i>login</i> tidak <i>valid</i>
10. Memasukkan <i>username</i> dan <i>password client</i> valid	11. Memeriksa <i>valid</i> tidaknya data masukan
	12. Masuk ke aplikasi dengan <i>role</i>

Aksi Aktor	Reaksi Sistem
	<i>client.</i>
13. <i>Admin/Client</i> memilih menu <i>logout</i>	
	14. Melakukan <i>logout</i>
	15. Menampilkan ke halaman login

b. Skenario use case mengelola tiket

Identifikasi	
Nama	Mengelola tiket
Tujuan	Mengelola tiket
Aktor	<i>Admin</i>
Skenario	

Aksi Aktor	Reaksi Sistem
	1. Memeriksa status <i>login</i>
2. Memilih menu pengelolaan tiket	
	3. Menampilkan halaman pengelolaan tiket
4. Klik tombol tambah tiket	
	5. Menampilkan modal form tambah tiket
6. Mengisi data kolom pada form tambah tiket	
	7. Memeriksa <i>valid</i> tidaknya data masukan
	8. Menyimpan data yang telah diubah ke database
	9. Menampilkan pesan berhasil disimpan
10. Memilih tiket yang ingin di edit	

Aksi Aktor	Reaksi Sistem
11. Klik tombol edit tiket pada data yang ingin di edit	
	12. Menampilkan modal form edit tiket dengan data sesuai id tiket yang dipilih
13. Mengubah data kolom pada form edit tiket	
	14. Memeriksa <i>valid</i> tidaknya data masukan
	15. Menyimpan data yang telah diubah ke database
	16. Menampilkan pesan berhasil diubah
17. Memilih tiket yang akan dihapus	
18. Mengklik tombol hapus data	
	19. Menampilkan konfirmasi penghapusan
20. <i>Admin</i> mengkonfirmasi	
	21. Menghapus data tiket dari database
	22. Menampilkan pesan berhasil dihapus
23. Memilih tiket yang akan dilihat detail	
24. Klik tombol detail	
	25. Menampilkan modal dan data detail <i>client</i>

c. Skenario *use case* mengelola rute

Identifikasi	
Nama	Mengelola rute
Tujuan	Mengelola rute yang tersedia
Aktor	<i>Admin</i>
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Memeriksa status <i>login</i>
2. Memilih menu pengelolaan rute	
	3. Menampilkan halaman pengelolaan rute
4. Klik tombol tambah rute	
	5. Menampilkan modal form tambah rute
6. Mengisi data kolom pada form tambah rute	
	7. Memeriksa <i>valid</i> tidaknya data masukan
	8. Menyimpan data yang telah diubah ke database
	9. Menampilkan pesan berhasil disimpan
10. Memilih rute yang ingin di edit	
11. Klik tombol edit rute pada data yang ingin di edit	
	12. Menampilkan modal form edit rute dengan data sesuai id rute yang dipilih
13. Mengubah data kolom pada form edit rute	

Aksi Aktor	Reaksi Sistem
	14. Memeriksa <i>valid</i> tidaknya data masukan
	15. Menyimpan data yang telah diubah ke database
	16. Menampilkan pesan berhasil diubah
17. Memilih rute yang akan dihapus	
18. Mengklik tombol hapus data	
	19. Menampilkan konfirmasi penghapusan
20. <i>Admin</i> mengkonfirmasi	
	21. Menghapus data rute dari database
	22. Menampilkan pesan berhasil dihapus

d. Skenario *use case* mengelola transaksi

Identifikasi	
Nama	Mengelola transaksi
Tujuan	Mengelola transaksi pemesanan tiket
Aktor	<i>Admin/Client</i>
Skenario	

Aksi Aktor	Reaksi Sistem
	1. Memeriksa status <i>login</i>
2. Memilih menu dashboard	
	3. Menampilkan halaman dashboard
4. <i>Client</i> memilih tiket yang akan dibeli	
5. Memasukan jumlah tiket yang dipesan dan klik tombol pesan tiket	

Aksi Aktor	Reaksi Sistem
	6. Memeriksa <i>valid</i> tidaknya data masukan
	7. Menyimpan data pemesanan ke <i>database transaction</i> dan <i>notification</i>
	8. Menampilkan halaman transaksi
	9. Menampilkan pesan pemesanan berhasil dilakukan
10. Memilih menu transaksi	
	11. Menampilkan halaman transaksi
12. <i>Client</i> memilih tiket yang akan dibayarkan DP	
13. Memasukan bukti bayar dan nominal yang dibayarkan serta klik tombol bayar	
	14. Memeriksa <i>valid</i> tidaknya data masukan
	15. Menyimpan data pembayaran DP ke <i>database transaction</i> dan <i>notification</i>
	16. Menampilkan halaman transaksi
	17. Menampilkan pesan pembayaran berhasil dilakukan dan menunggu konfirmasi <i>admin</i>
18. <i>Client</i> memilih tiket yang akan dibayar lunas	
19. Memasukan bukti bayar dan nominal yang dibayarkan serta klik tombol bayar	
	20. Memeriksa <i>valid</i> tidaknya data masukan

Aksi Aktor	Reaksi Sistem
	21. Menyimpan data pembayaran lunas ke <i>database transaction</i> dan <i>notification</i>
	22. Menampilkan halaman transaksi
	23. Menampilkan pesan pembayaran berhasil dilakukan dan menunggu konfirmasi <i>admin</i>
24. <i>Client</i> memilih tiket yang akan dibatalkan	
	25. Menampilkan konfirmasi pembatalan tiket
26. Konfirmasi <i>client</i>	
	27. Menghapus data tiket dari database
	28. Menampilkan pesan berhasil dihapus
29. Memilih transaksi yang akan dilihat detail pembayarannya	
30. Klik tombol detail	
	31. Menampilkan data bukti bayar
32. <i>Admin</i> memilih tiket yang akan diubah statusnya	
33. Klik tombol ubah status	
	34. Menampilkan data bukti bayar
35. Memilih status pembayaran	
36. Klik tombol ubah	
	37. Menyimpan data status pembayaran ke <i>database transaction</i> dan <i>notification</i>
	38. Menampilkan halaman transaksi
	39. Menampilkan pesan status

Aksi Aktor	Reaksi Sistem
	pembayaran telah diubah

e. Skenario *use case* mengelola *client*

Identifikasi	
Nama	Mengelola <i>client</i>
Tujuan	Mengelola <i>client</i> yang dapat mengakses aplikasi
Aktor	<i>Admin</i>
Skenario	

Aksi Aktor	Reaksi Sistem
	1. Memeriksa status <i>login</i>
2. Memilih menu pengelolaan <i>client</i>	
	3. Menampilkan halaman pengelolaan <i>client</i>
4. Mengklik tombol untuk terima	
	5. Mengubah status <i>client</i> di <i>database</i>
	6. Menampilkan pesan <i>client</i> diterima
7. memilih data <i>client</i> yang akan di ubah	
8. Mengklik tombol edit <i>client</i>	
	9. Menampilkan formulir edit <i>client</i>
	10. Menampilkan data sesuai id
11. Memasukkan data <i>client</i> yang ada sesuai kolom	
	12. Memeriksa <i>valid</i> tidaknya data masukan
	13. Menyimpan data yang telah

Aksi Aktor	Reaksi Sistem
	diubah ke database
	14. Menampilkan pesan berhasil disimpan
15. Memilih <i>client</i> yang akan dihapus	
16. Mengklik tombol hapus data	
	17. Menampilkan konfirmasi penghapusan
18. <i>Admin</i> mengkonfirmasi	
	19. Menghapus data <i>client</i> dari database
	20. Menampilkan pesan berhasil dihapus
21. Memilih <i>client</i> yang akan dilihat detail	
22. Klik tombol detail	
	23. Menampilkan data detail <i>client</i>

f. Skenario *use case* mengubah data akun

Identifikasi	
Nama	Mengubah data akun
Tujuan	Mengubah data akun
Aktor	<i>Admin/Client</i>
Skenario	
Aksi Aktor	Reaksi Sistem
1. Memilih menu pengaturan akun	
	2. Menampilkan halaman pengaturan akun dan form pengaturan akun

Aksi Aktor	Reaksi Sistem
3. Mengubah data aku yang diinginkan	
4. Klik tombol perbarui status	
	5. Memeriksa <i>valid</i> tidaknya data masukan
	6. Menyimpan data yang telah diubah ke database
	7. Menampilkan pesan akun telah diperbaharui

g. Skenario *use case* membaca notifikasi

Identifikasi	
Nama	Melihat detail notifikasi
Tujuan	Melihat detail notifikasi
Aktor	<i>Admin/Client</i>
Skenario	

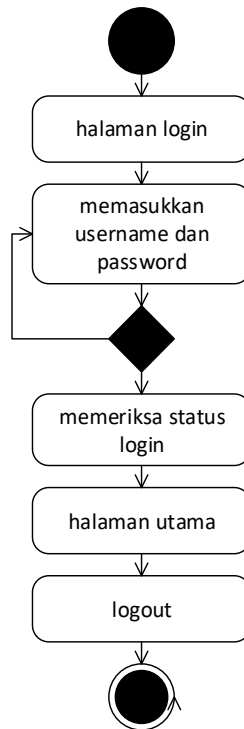
Aksi Aktor	Reaksi Sistem
	1. Memeriksa status <i>login</i>
2. Memilih ikon notifikasi dan memilih tampilan semua notifikasi	
	3. Menampilkan menu notifikasi dan data semua notifikasi
4. Klik tombol baca semua	
	5. Mengubah status dibaca di <i>database notification</i>
	6. Menampilkan halaman notifikasi

h. Skenario use case registrasi

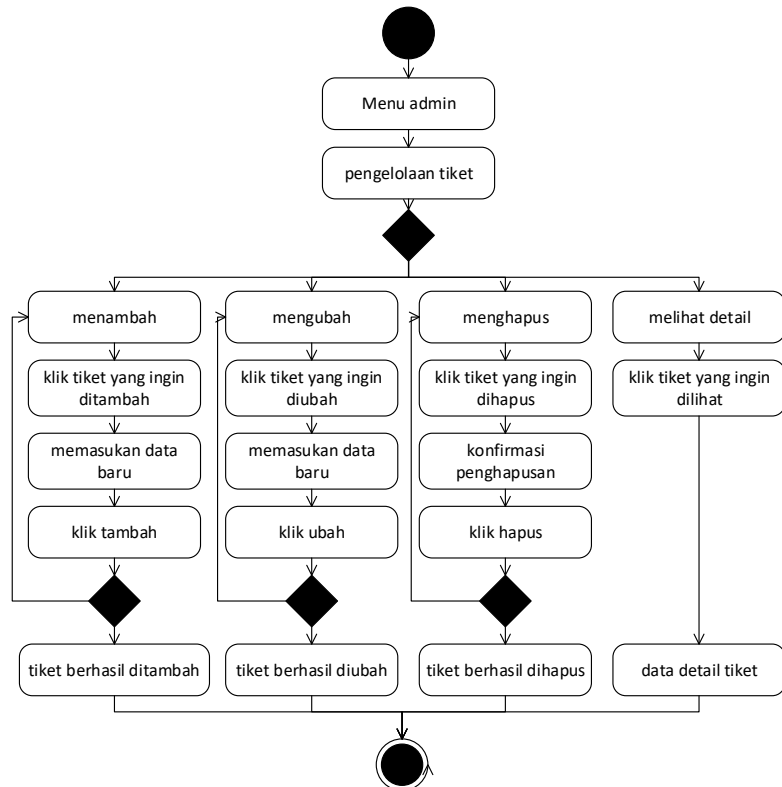
Identifikasi	
Nama	Registrasi <i>Client</i>
Tujuan	Registrasi untuk mendaftar menjadi <i>client</i> untuk dapat mengakses aplikasi
Aktor	<i>Client</i>
Skenario	
Aksi Aktor	Reaksi Sistem
1. Memilih menu registrasi	
	2. Menampilkan menu dan <i>form</i> registrasi
3. Memasukan data sesuai kolom dalam <i>form</i> dan klik submit	
	4. Memeriksa <i>valid</i> tidaknya data masukan
	5. Memasukan data kedalam <i>database users</i>
	6. Memasukan database ke <i>database notifications</i>
	7. Masuk ke halaman dashboard sementara <i>client</i>

4. Rancangan Activity Diagram

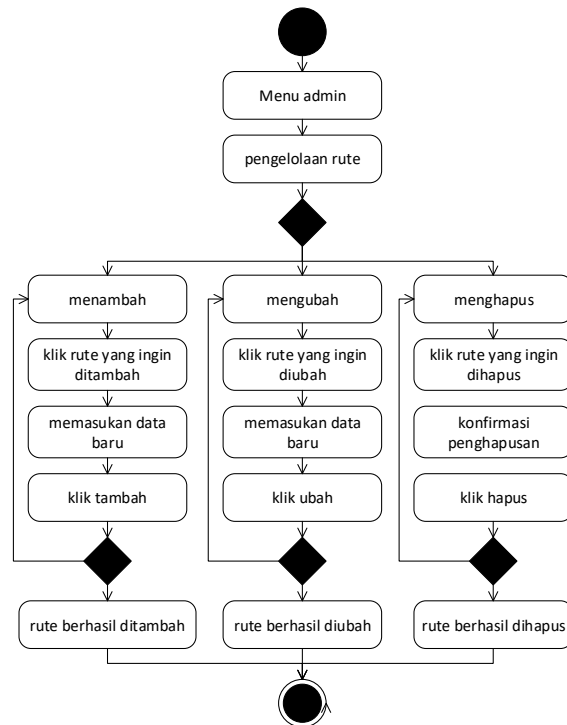
a. Rancangan activity diagram login



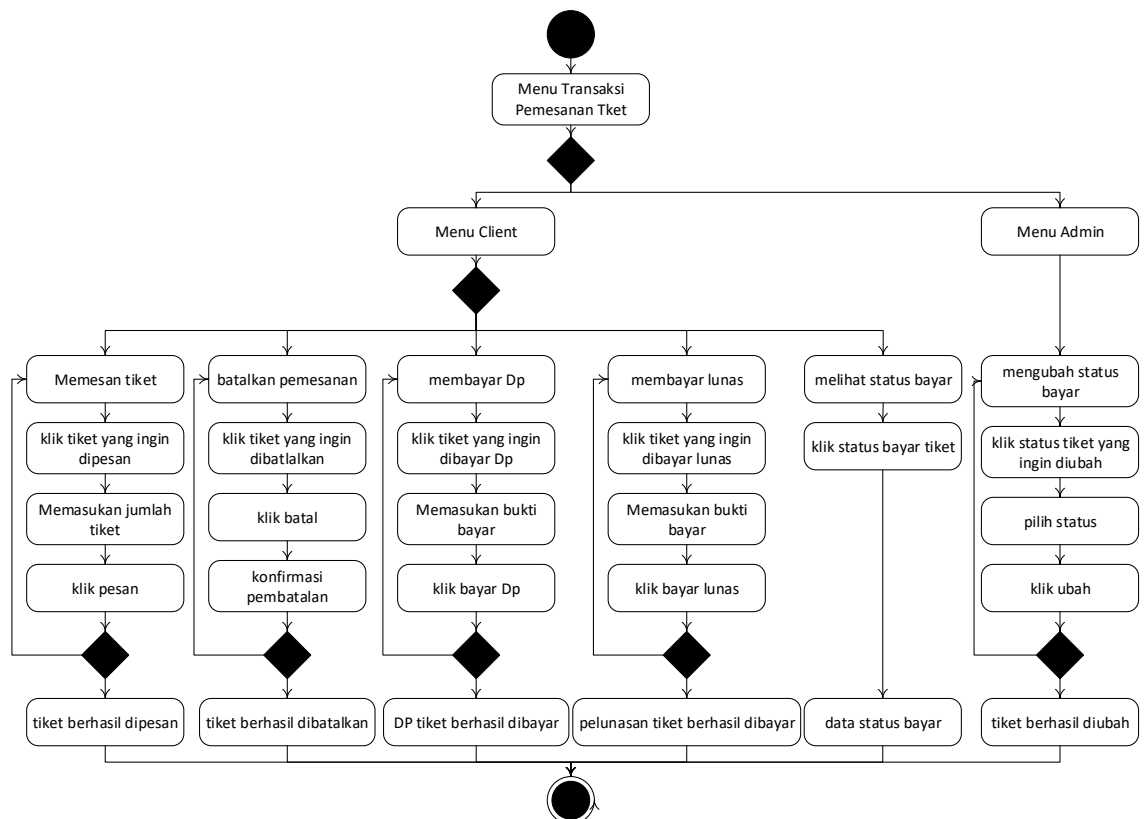
b. Rancangan activity diagram mengelola tiket



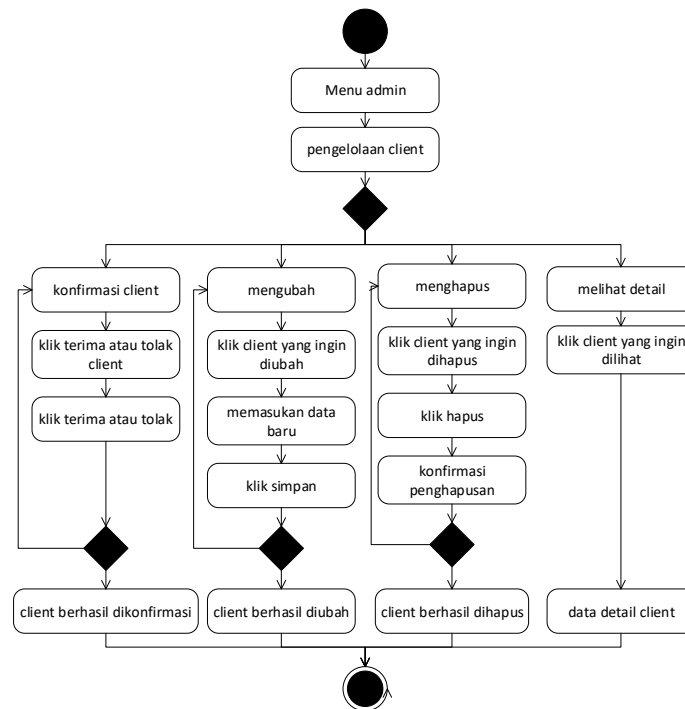
c. Rancangan *activity diagram* mengelola rute



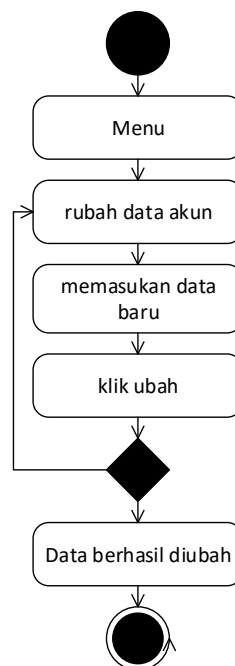
d. Rancangan *activity diagram* mengelola transaksi



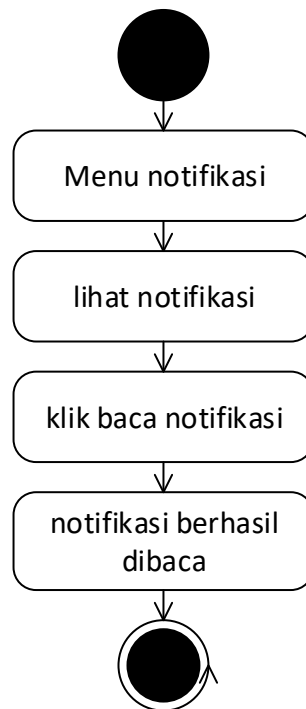
e. Rancangan *activity diagram* mengelola *client*



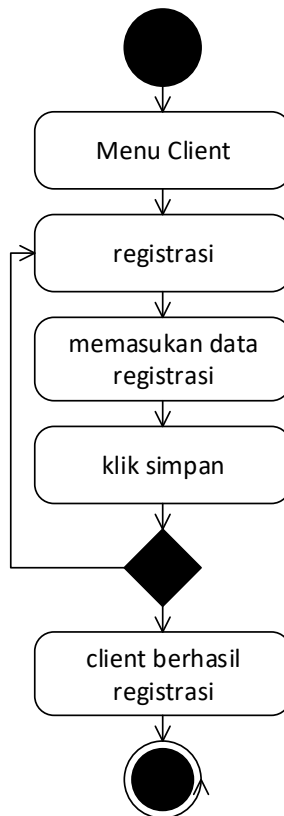
f. Rancangan *activity diagram* mengubah data akun



g. Rancangan *activity diagram* membaca notifikasi

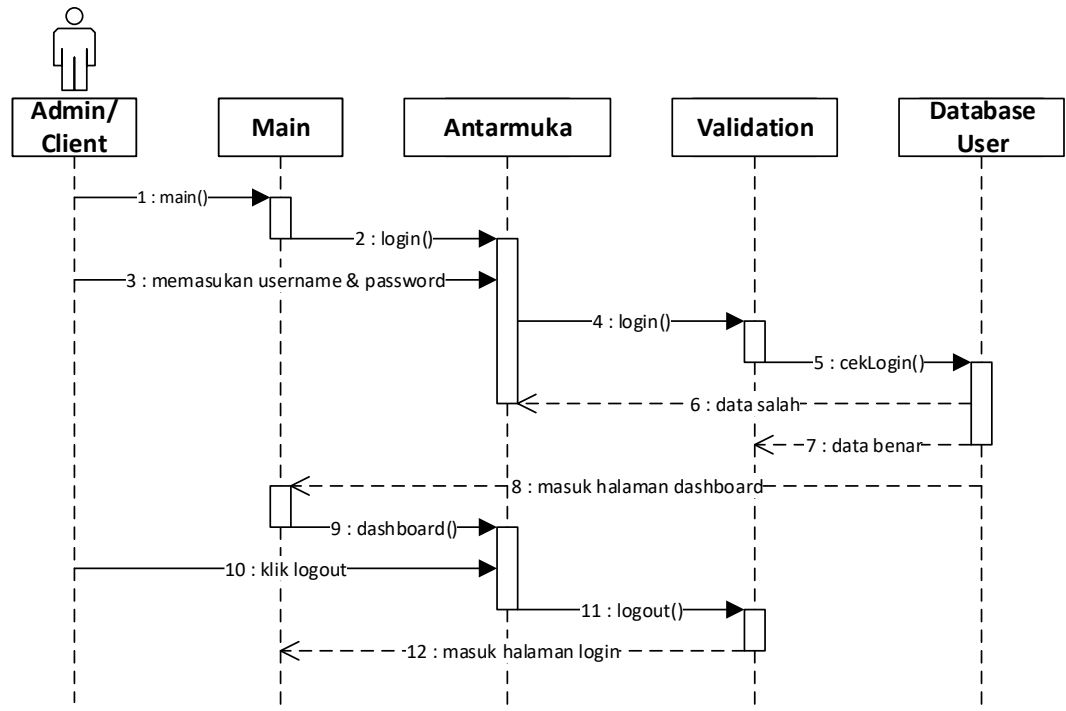


h. Rancangan *activity diagram* registrasi *client*

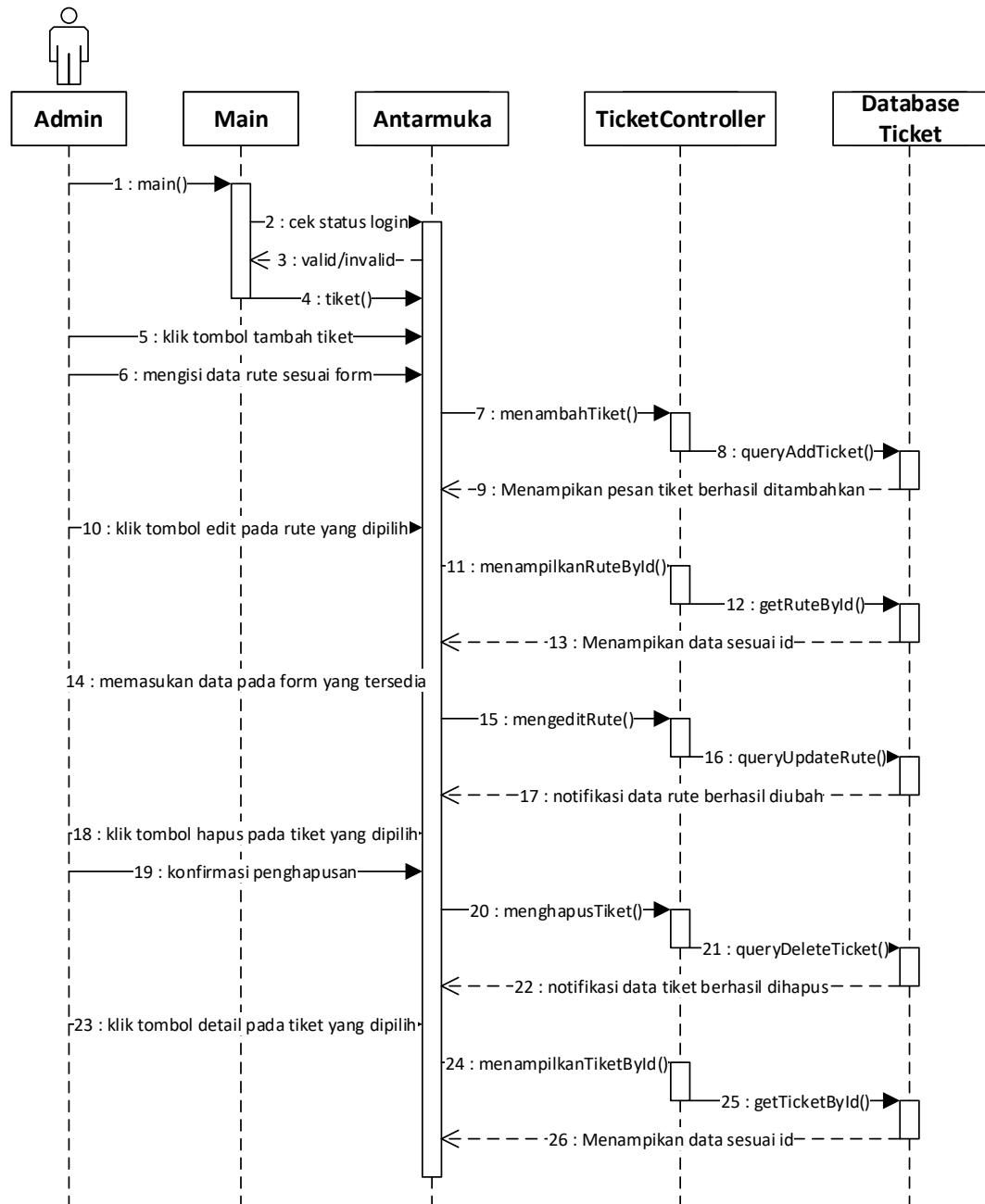


5. Rancangan *Sequence Diagram*

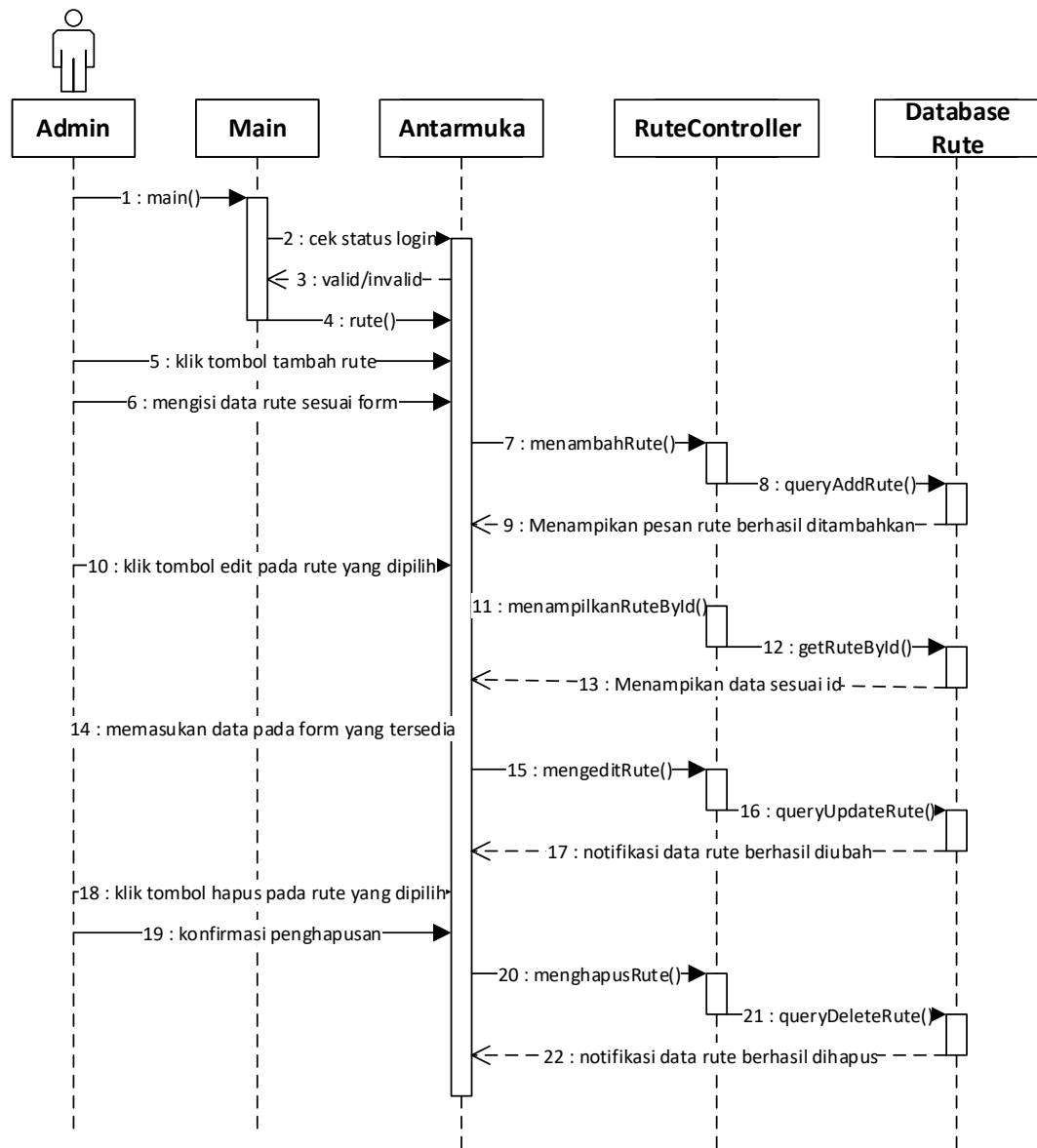
a. Skenario *sequence diagram login*



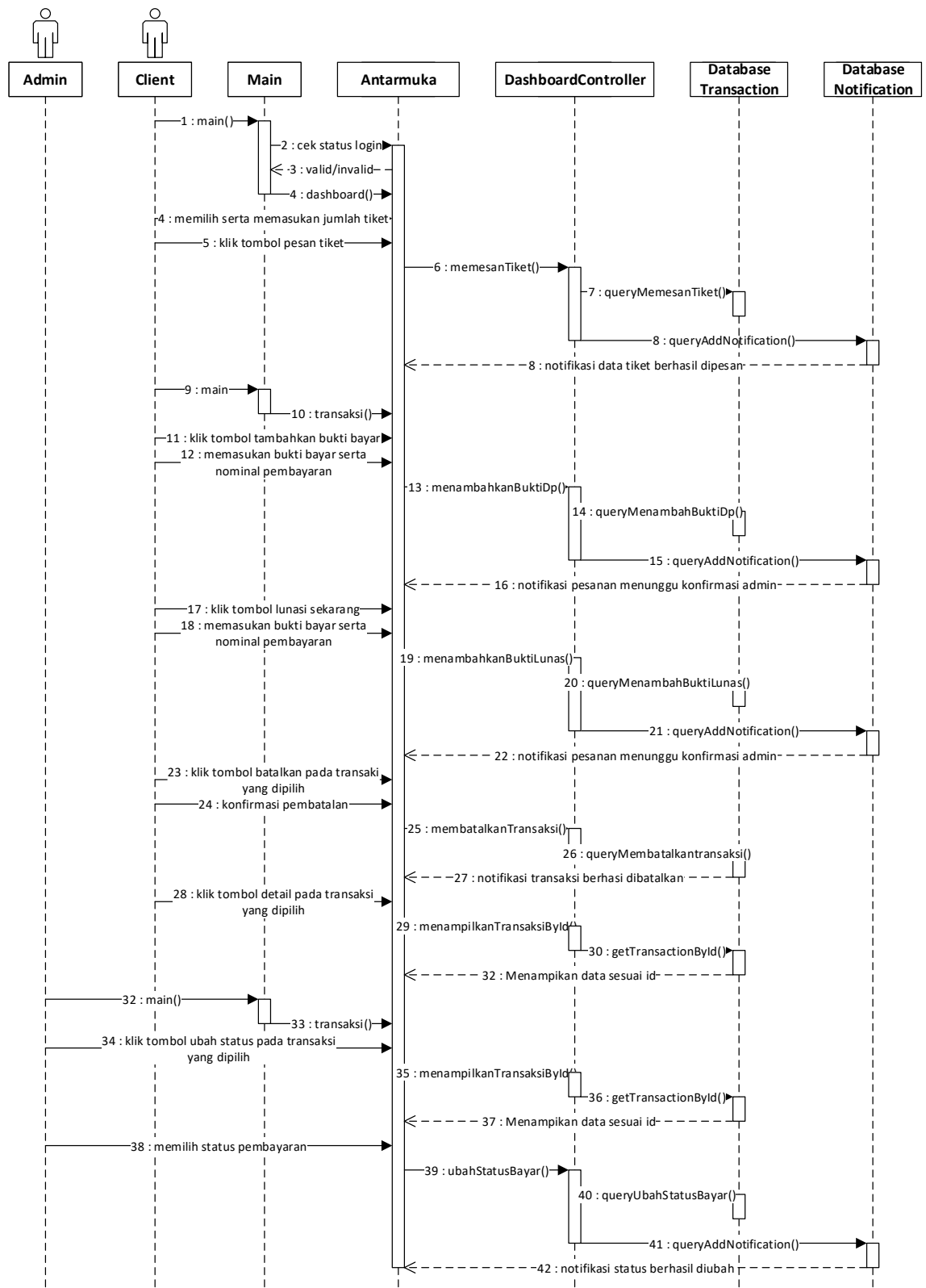
b. Skenario *sequence diagram* mengelola tiket



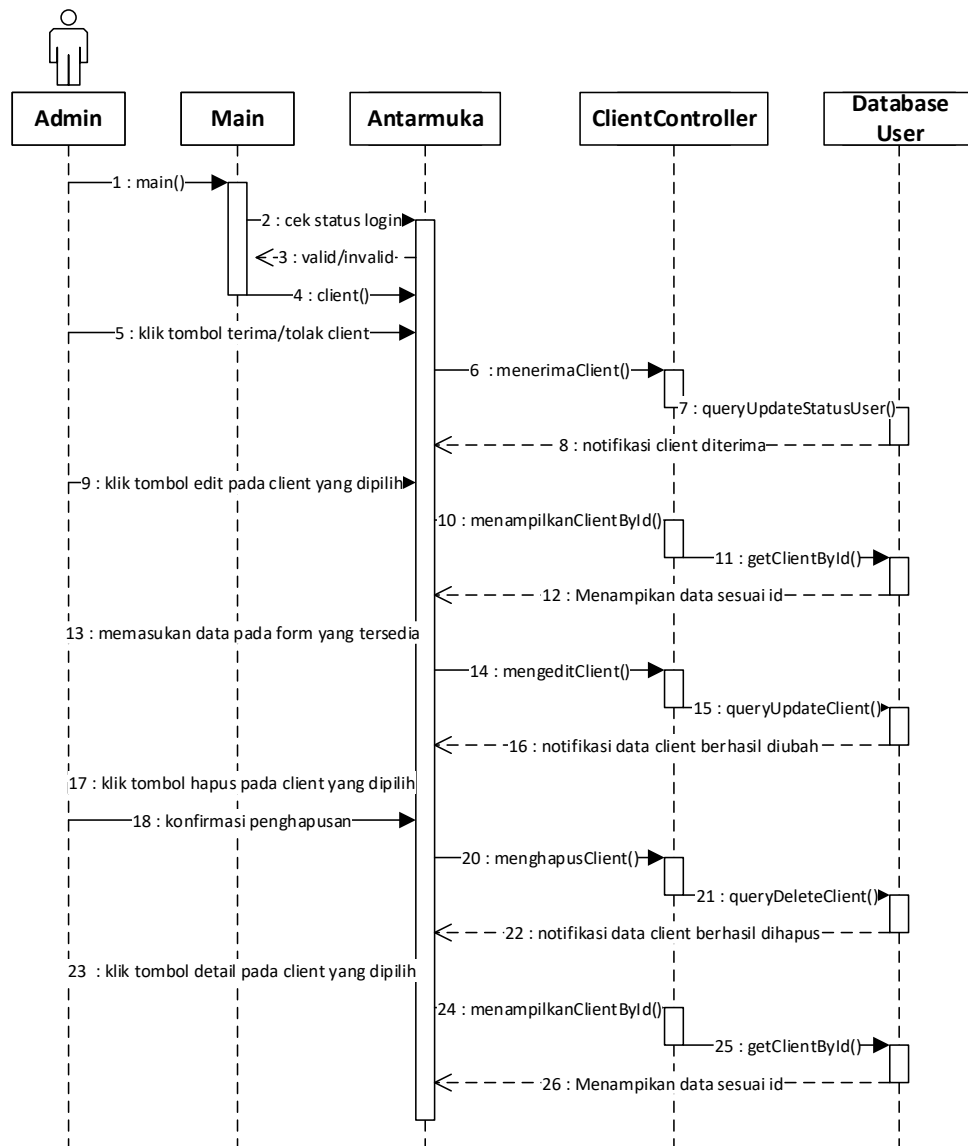
c. Skenario *sequence diagram* mengelola rute



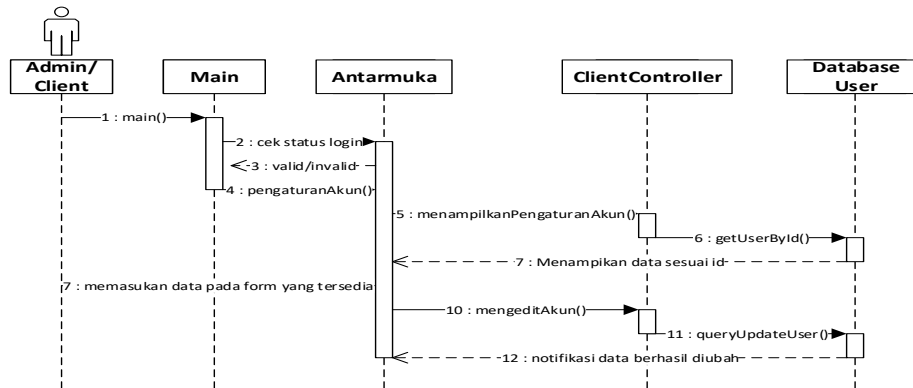
d. Skenario *sequence diagram* mengelola transaksi



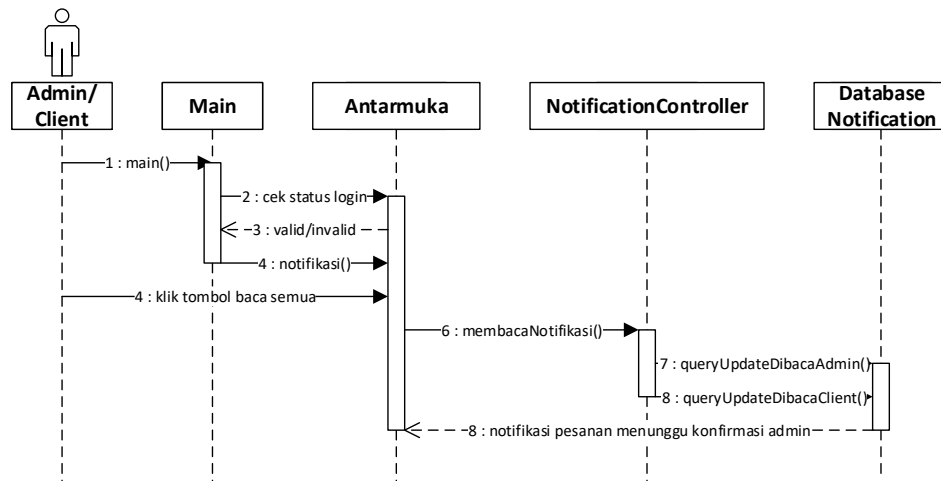
e. Skenario *sequence diagram* mengelola *client*



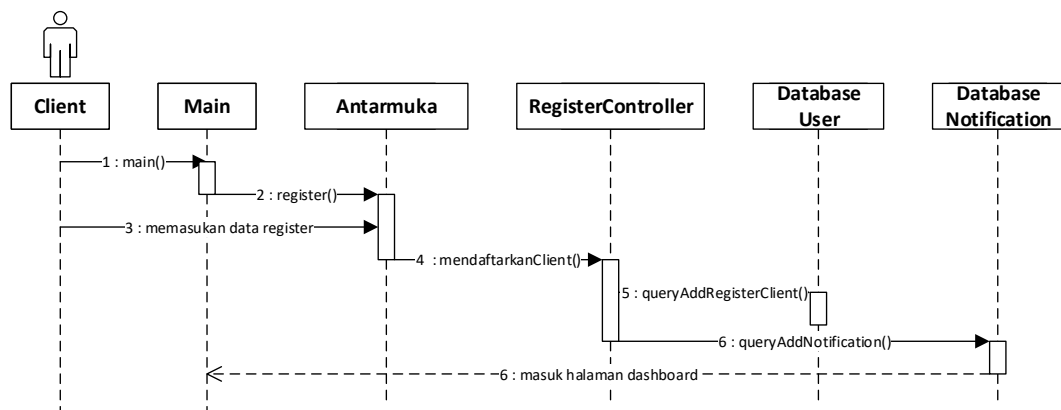
f. Skenario *sequence diagram* mengubah data akun



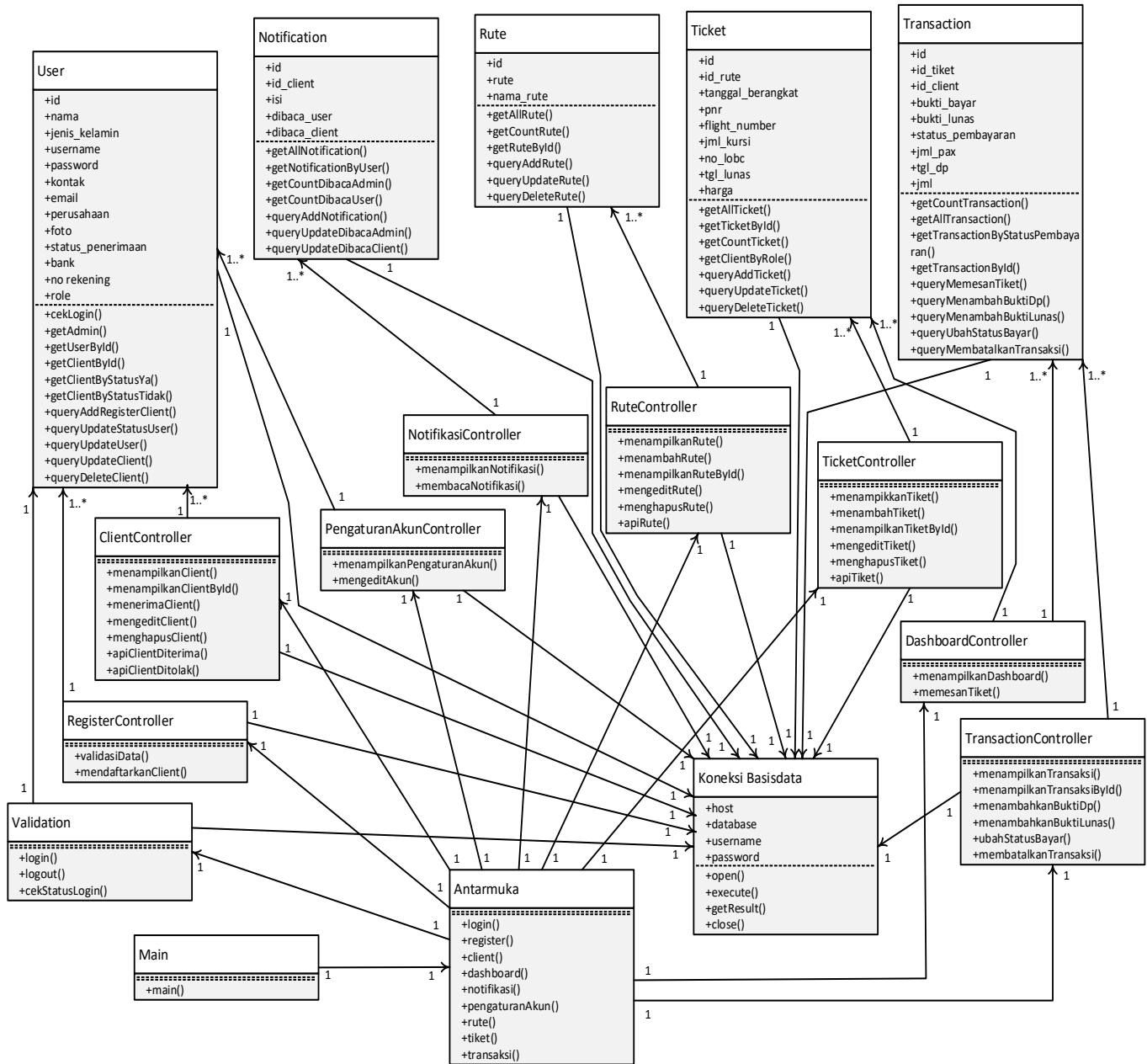
g. Skenario *sequence diagram* membaca notifikasi



h. Skenario *sequence diagram* registrasi client

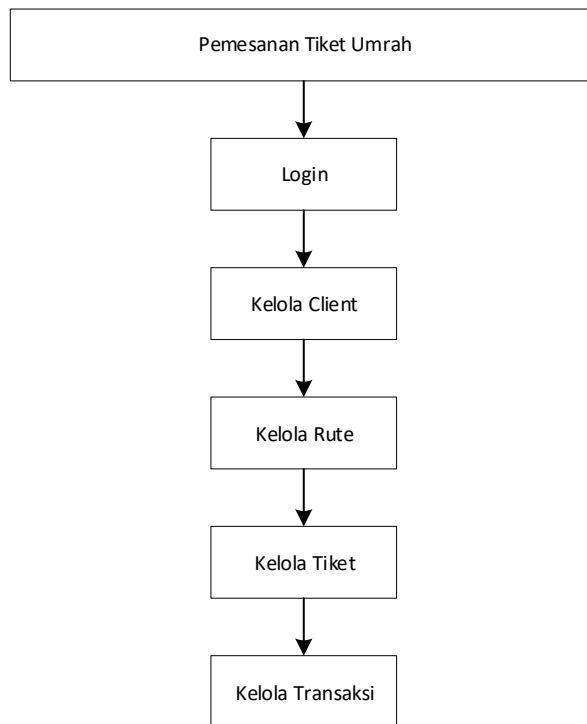


6. Rancangan Class Diagram

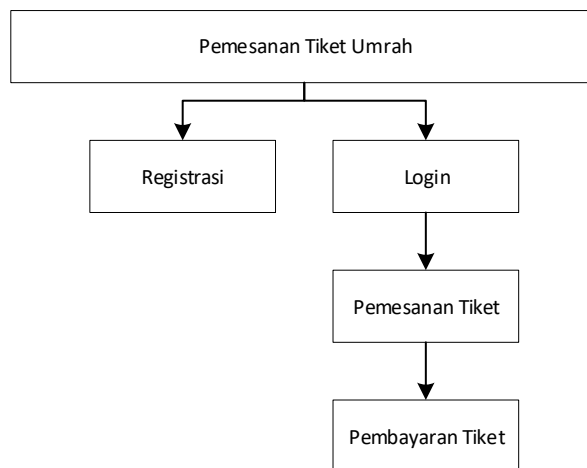


7. Rancangan Struktur Menu

a. Admin

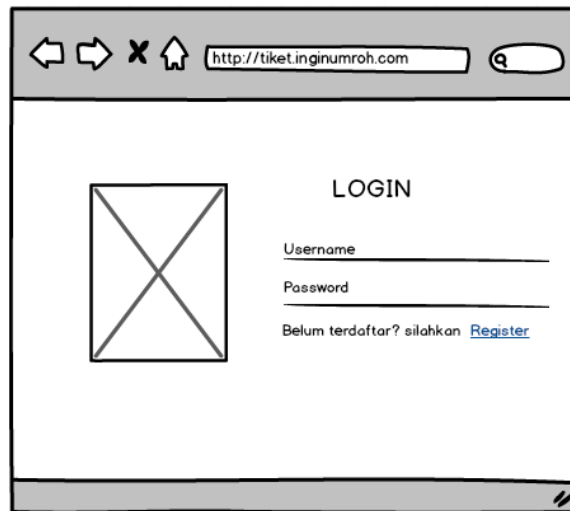


b. Client



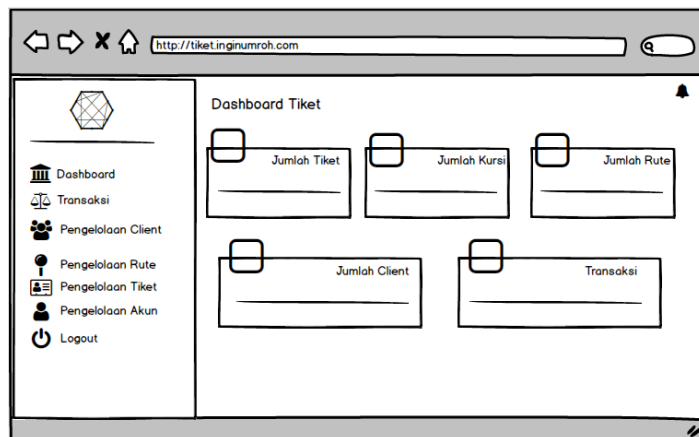
8. Hasil Perancangan

a. Login



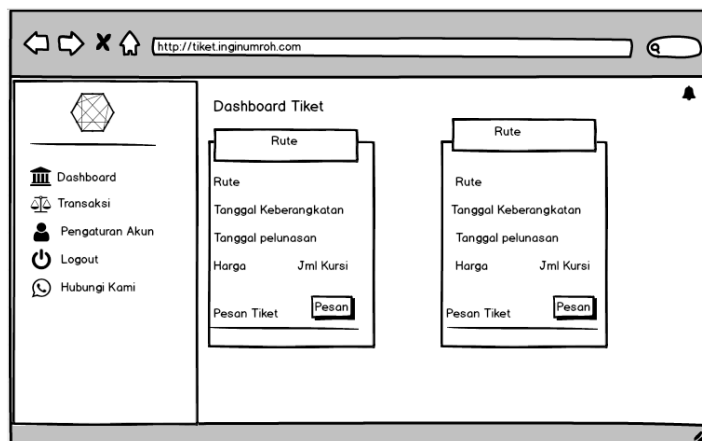
A browser window mockup showing a login page. The address bar contains 'http://tiket.inginumroh.com'. The page has a large square placeholder with an 'X' on the left. To the right, the title 'LOGIN' is centered. Below it are two input fields labeled 'Username' and 'Password'. At the bottom, there is a link that says 'Belum terdaftar? silahkan [Register](#)'.

b. Tampilan Admin



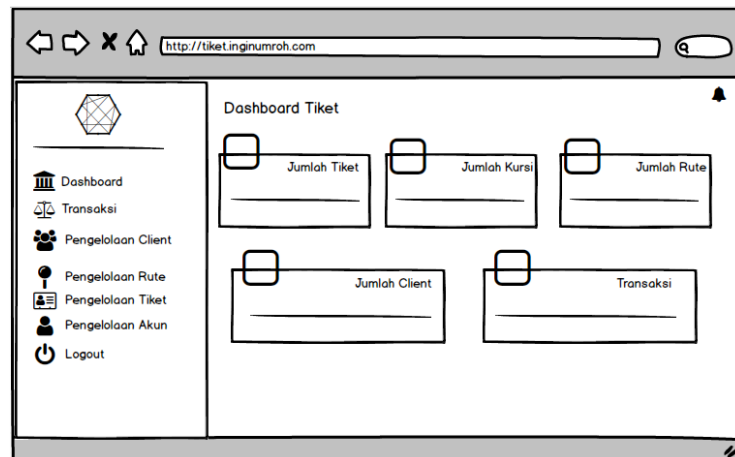
A browser window mockup showing an admin dashboard. The address bar contains 'http://tiket.inginumroh.com'. On the left is a sidebar menu with icons and labels: Dashboard, Transaksi, Pengelolaan Client, Pengelolaan Rute, Pengelolaan Tiket, Pengelolaan Akun, and Logout. The main area is titled 'Dashboard Tiket' and contains five summary cards: 'Jumlah Tiket', 'Jumlah Kursi', 'Jumlah Rute', 'Jumlah Client', and 'Transaksi'.

c. Tampilan Client

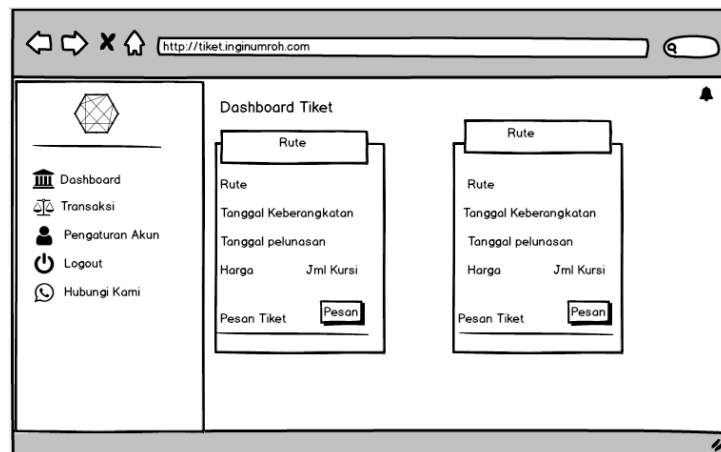


A browser window mockup showing a client dashboard. The address bar contains 'http://tiket.inginumroh.com'. On the left is a sidebar menu with icons and labels: Dashboard, Transaksi, Pengaturan Akun, Logout, and Hubungi Kami. The main area is titled 'Dashboard Tiket' and contains two identical booking forms. Each form has a 'Rute' header, followed by fields for 'Rute', 'Tanggal Keberangkatan', 'Tanggal pelunasan', 'Harga', and 'Jml Kursi'. At the bottom of each form is a 'Pesan Tiket' label and a 'Pesan' button.

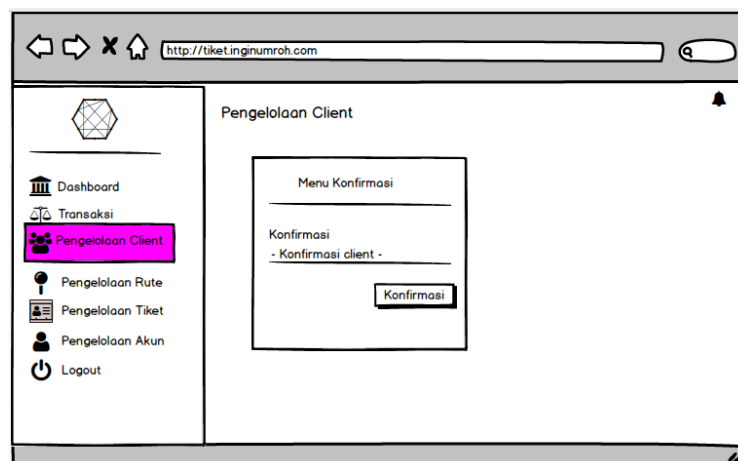
d. Dashboard *Admin*



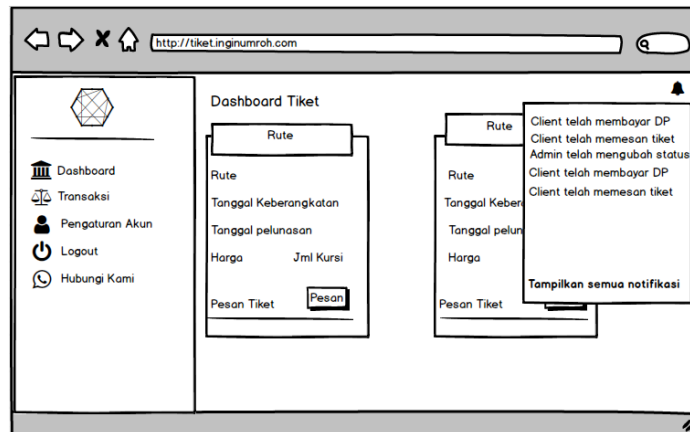
e. Dashboard *Client*



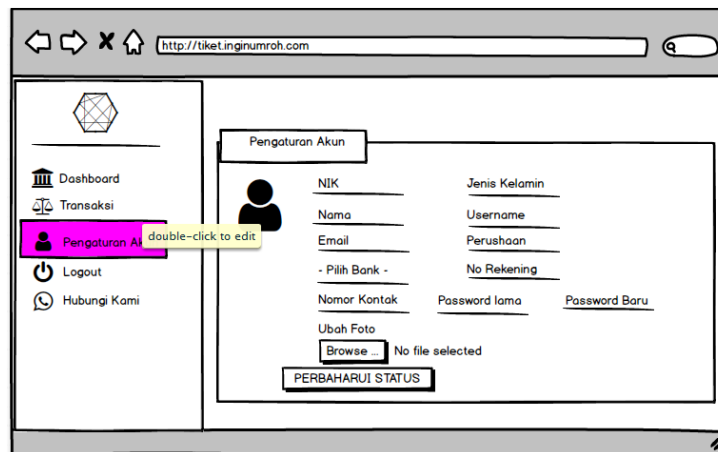
f. Konfirmasi *Client*



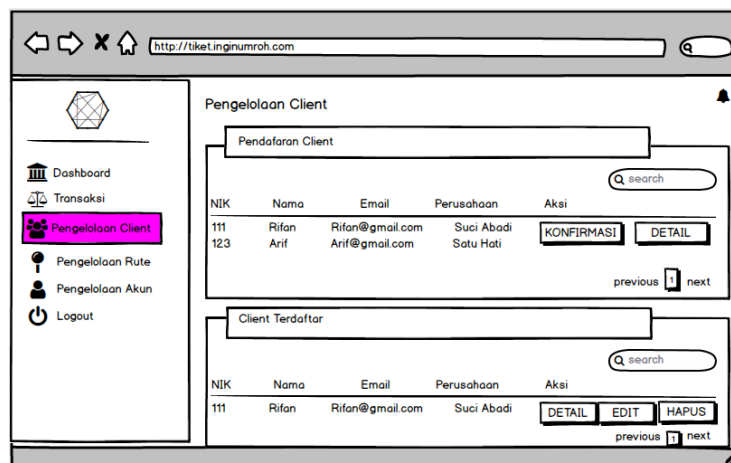
g. Notifikasi



h. Pengelolaan Akun



i. Pengelolaan Client



j. Pengelolaan Route

The screenshot shows a web browser at <http://tiket.inginumroh.com>. The left sidebar contains a menu with the following items: Dashboard, Transaksi, Pengelolaan Client, **Pengelolaan Route** (highlighted), Pengelolaan Tiket, Pengelolaan Akun, and Logout. The main content area is titled 'Pengelolaan Route' and features a 'TAMBAH ROUTE' button. Below this is a search bar labeled 'Rute Tersedia' with a 'search' button. A table displays the following data:

Rute	Nama rute	Aksi
jkt-jdh	Jakarta -Jeddah	EDIT HAPUS
cgk-med-jkt	cengkareng-medinah-jakarta	

At the bottom right of the table, there are 'previous' and 'next' pagination controls.

k. Pengelolaan Tiket

The screenshot shows a web browser at <http://tiket.inginumroh.com>. The left sidebar is identical to the previous screenshot, with 'Pengelolaan Tiket' highlighted. The main content area is titled 'Pengelolaan Tiket' and features a 'TAMBAH TIKET' button. Below this is a search bar labeled 'Tiket tersedia' with a 'search' button. A table displays the following data:

No Rute	Jumlah Kursi	Tgl Berangkat	Harga	Tgl Lunas	Aksi
111	20	25 april 2019	Rp...	Rp...	DETAIL EDIT HAPUS
143	30	27 mei 2019	Rp...	Rp...	

At the bottom right of the table, there are 'previous' and 'next' pagination controls.

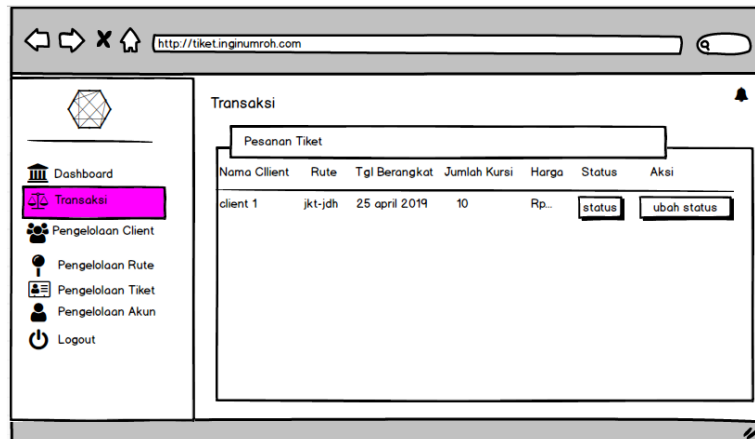
l. Register

The screenshot shows a web browser at <http://tiket.inginumroh.com>. The main content area is titled 'Register' and contains a registration form with the following fields:

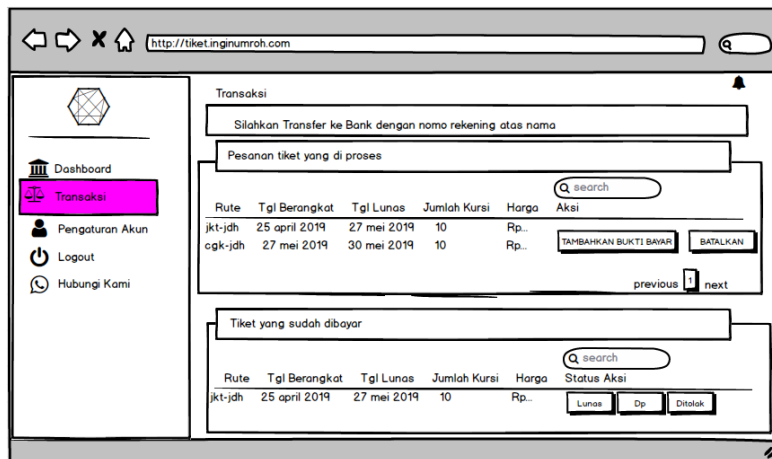
- NIK
- Nama
- Jenis Kelamin (dropdown menu)
- Username
- Email
- Perusahaan
- No Kontak
- Password
- Konfirmasi Password
- Foto (with a 'browse' button and 'No file selected' text)

A 'Submit' button is located at the bottom right of the form. Below the form, there is a link: 'Sudah punya akun? silahkan [Login](#)'.

m. Transaksi *Admin*

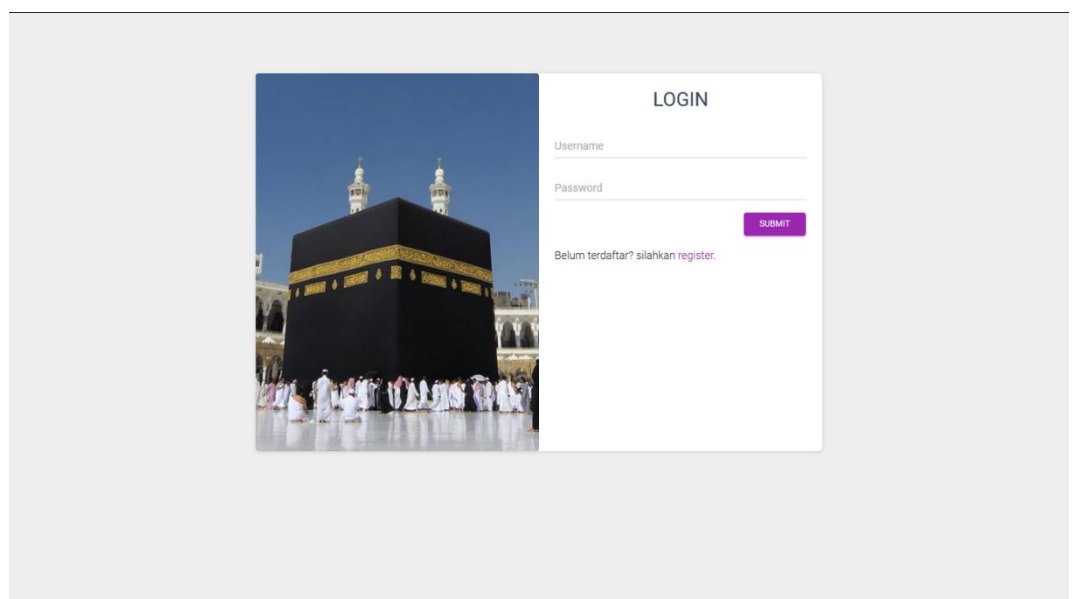


n. Transaksi *Client*



8. Rancangan Aplikasi

a. Login



b. Register

Register

No file selected.

Sudah punya akun? silahkan login.

c. Dashboard Admin

- Dashboard
- Transaksi
- Pengelolaan Client
- Pengelolaan Rute
- Pengelolaan Tiket
- Pengaturan Akun
- Logout

Dashboard Tiket

Jumlah Tiket

4 tiket

< Pengelolaan Tiket

Jumlah Kursi

300 kursi

< Pengelolaan Kursi Tiket

Jumlah Rute

4 rute

< Pengelolaan Rute

Jumlah Client

3 client

< Pengelolaan Client

Transaksi

5 transaksi

< Transaksi

d. Menu Transaksi Admin

- Dashboard
- Transaksi
- Pengelolaan Client
- Pengelolaan Rute
- Pengelolaan Tiket
- Pengaturan Akun
- Logout

Transaksi

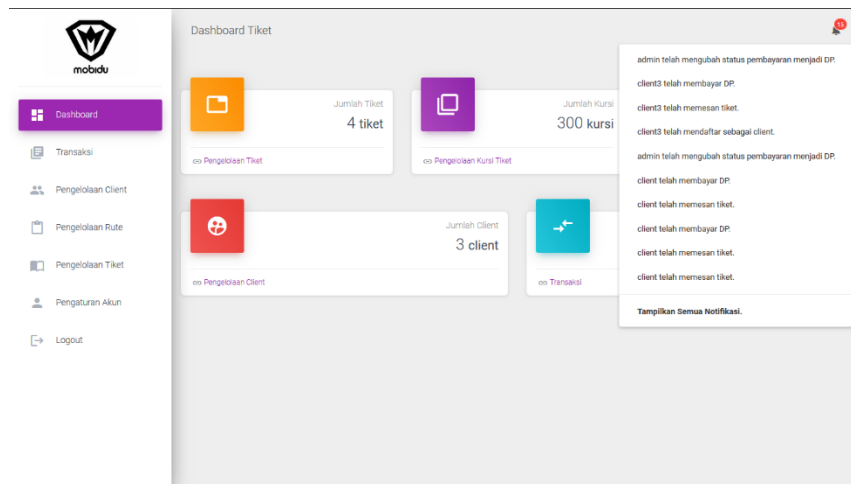
Pesanan Tiket

Show: 10 entries
Search:

#	Nama Client	Rute	Tgl Berangkat	Jumlah Kursi	Harga	Status	Aksi
1	client3	CGK JED CGK PROMO	01/26/2018 - 01/31/2018	20	Rp 2.000.000,-	Open	UBAH STATUS
2	client3	CGK JED MED CGK	01/26/2018 - 01/31/2018	10	Rp 100.200.000,-	Open	UBAH STATUS
3	client	CGK MED JED CGK	01/26/2018 - 01/31/2018	80	Rp 10.000.000,-	Open	UBAH STATUS
4	client	CGK JED JED CGK PROMO	01/26/2018 - 01/31/2018	23	Rp 0,-	Open	UBAH STATUS
5	client	CGK MED JED CGK PROMO	02/13/2018 - 02/23/2018	20	Rp 2.000.000,-	Open	UBAH STATUS

Showing 1 to 5 of 5 entries
Previous 1 Next

e. Notifikasi Admin

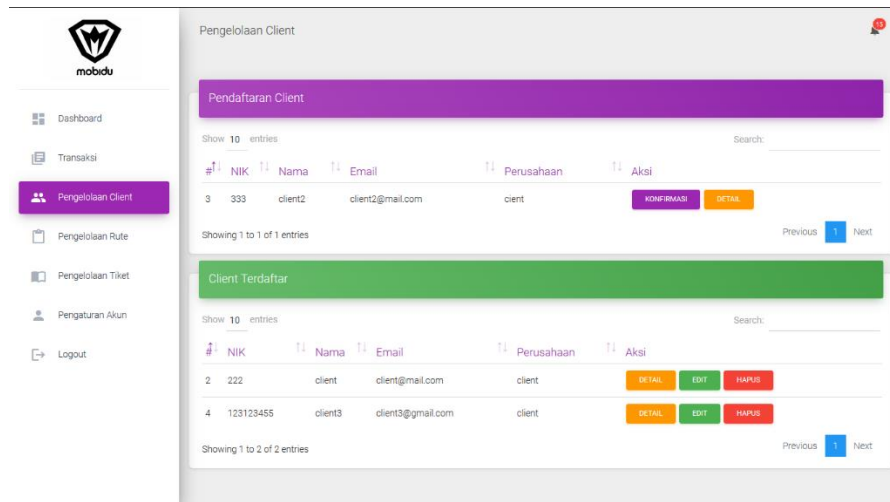


The screenshot shows the 'Dashboard Tiket' interface. On the right, a notification box displays a list of recent actions:

- admin telah mengubah status pembayaran menjadi DP.
- client3 telah membayar DP.
- client3 telah memesan tiket.
- client3 telah mendaftar sebagai client.
- admin telah mengubah status pembayaran menjadi DP.
- client telah membayar DP.
- client telah memesan tiket.
- client telah membayar DP.
- client telah memesan tiket.
- client telah memesan tiket.

At the bottom of the notification box, there is a button labeled 'Tampilkan Semua Notifikasi'.

f. Pengelolaan Client (Admin)



The screenshot shows the 'Pengelolaan Client' page. It contains two tables:

Pendaftaran Client

#	NIK	Nama	Email	Perusahaan	Aksi
3	333	client2	client2@gmail.com	client	KONFIRMASI DETAIL

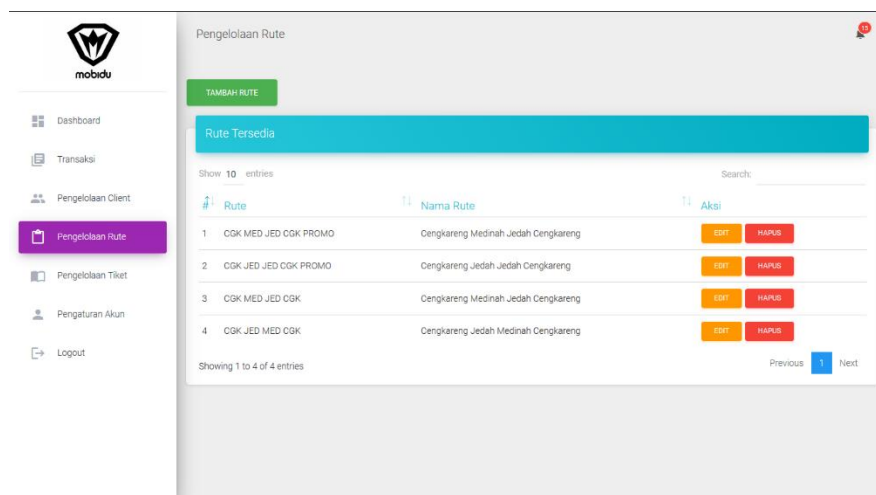
Showing 1 to 1 of 1 entries

Client Tendaftar

#	NIK	Nama	Email	Perusahaan	Aksi
2	222	client	client@mail.com	client	DETAIL EDIT HAPUS
4	123123455	client3	client3@gmail.com	client	DETAIL EDIT HAPUS

Showing 1 to 2 of 2 entries

g. Pengelolaan Rute (Admin)



The screenshot shows the 'Pengelolaan Rute' page. It includes a 'TAMBAH RUTE' button and a table of available routes:

Rute Tersedia

#	Rute	Nama Rute	Aksi
1	CGK MED JED CGK PROMO	Cengkareng Medinah Jedah Cengkareng	EDIT HAPUS
2	CGK JED JED CGK PROMO	Cengkareng Jedah Jedah Cengkareng	EDIT HAPUS
3	CGK MED JED CGK	Cengkareng Medinah Jedah Cengkareng	EDIT HAPUS
4	CGK JED MED CGK	Cengkareng Jedah Medinah Cengkareng	EDIT HAPUS

Showing 1 to 4 of 4 entries

h. Pengelolaan Tiket (Admin)

Pengelolaan Tiket

TAMBAH TIKET

Tiket Tersedia

Show 10 entries

No	Route	Jumlah Kursi	Tgl Berangkat	Harga	Tgl Lunas	Aksi
1	CGK MED JED CGK PROMO	80	02/13/2018 - 02/22/2018	32000000	2019-11-29	DETAIL EDIT HAPUS
2	CGK JED JED CGK PROMO	120	01/26/2018 - 01/31/2018	40000000	2019-11-27	DETAIL EDIT HAPUS
3	CGK MED JED CGK	80	01/26/2018 - 01/31/2018	21000000	2019-11-13	DETAIL EDIT HAPUS
4	CGK JED MED CGK	20	01/26/2018 - 01/31/2018	43000000	2019-11-11	DETAIL EDIT HAPUS

Showing 1 to 4 of 4 entries

Previous 1 Next

i. Ubah status Bayar (Admin)

Ubah Status Pembayaran

Bukti DP

Ubah Status

ditolak

UBAH

j. Dashboard Client

Dashboard Tiket

Cengkareng Medinah Jeddah Cengkareng

Route: CGK MED JED CGK PROMO

Tanggal Keberangkatan: 02/13/2018 - 02/22/2018

Tanggal Pelunasan: 2019-11-29

Harga: Rp 32.000.000,-

Jumlah Kursi: 80

Pesan Tiket: [PESAN](#)

Cengkareng Jeddah Jeddah Cengkareng

Route: CGK JED JED CGK PROMO

Tanggal Keberangkatan: 01/26/2018 - 01/31/2018

Tanggal Pelunasan: 2019-11-27

Harga: Rp 40.000.000,-

Jumlah Kursi: 120

Pesan Tiket: [PESAN](#)

Cengkareng Medinah Jeddah Cengkareng

Route: CGK MED JED CGK

Tanggal Keberangkatan: 01/26/2018 - 01/31/2018

Tanggal Pelunasan: 2019-11-13

Harga: Rp 21.000.000,-

Jumlah Kursi: 80

Pesan Tiket: [PESAN](#)

Cengkareng Jeddah Medinah Cengkareng

Route: CGK JED MED CGK

Tanggal Keberangkatan: 01/26/2018 - 01/31/2018

Tanggal Pelunasan: 2019-11-11

Pesan Tiket: [PESAN](#)

k. Pengaturan Akun (Client)

l. Transaksi (Client)

#	Rute	Tgl Berangkat	Tgl Lunas	Jumlah Kursi	Harga	Aksi
1	CGK JED JED CGK PROMO	01/26/2018 - 01/31/2018	2019-11-27	23	Rp 40.000.000,-	TAMBAH BUKTI BAYAR BATALAKAN

#	Rute	Tgl Berangkat	Tgl Lunas	Jumlah Kursi	Harga	Status Aksi
1	Cengkareng Medinah Jeddah Cengkareng	01/26/2018 - 01/31/2018	2019-11-13	80	Rp 10.000.000,-	LUNAS TRANSKASI LUNAS BUKTI DP

m. Bukti Transaksi (Client)

10. Pengujian kelas

Aktifitas	Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
Login	Melakukan login	Memasukan username & password valid	login berhasil	sesuai
	Melakukan logout	Klik tombol logout	logout	sesuai
Mengelola Tiket	Menambah tiket	Menambah data	Pengisian form berfungsi dengan baik	sesuai
	Mengubah tiket	Memilih data yang akan dirubah	Data akan berubah sesuai inputan	sesuai
	Menghapus tiket	Memilih data yang akan dihapus	Data berhasil dihapuskan	sesuai
	Melihat detail surat	Melihat detail yang diinginkan	Data berhasil dilihat	sesuai
Mengelola rute	Menambah rute	Menambah data	Pengisian form berfungsi dengan baik	sesuai
	Mengubah rute	Memilih data yang akan dirubah	Data akan berubah sesuai inputan	sesuai
	Menghapus rute	Memilih data yang akan dihapus	Data berhasil dihapuskan	sesuai
Mengelola Transaksi	Memesan Tiket	Memesan tiket yang disediakan	Tiket dipesan dengan jumlah yang dicantumkan	sesuai
	Membatalkan pesanan	Memilih tiket yang akan dibatalkan	Tiket akan berhasil dibatalkan	sesuai
	Membayar DP	Memilih tiket yang akan dibayar DP	Pembayaran DP berhasil dilakukan	sesuai
	Membayar	Memilih tiket	Pembayaran Lunas	sesuai

Aktifitas	Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
	Lunas	yang akan dibayar Lunas	berhasil dilakukan	
	Melihat status bayar	Memilih transaksi yang diinginkan	Data berhasil dilihat	sesuai
	Mengubah status bayar	Memilih status pembayaran	Pembayaran berhasil diperbaharui	sesuai
Mengelola <i>Client</i>	Konfirmasi <i>client</i>	Mengkonfirmasi <i>client</i>	<i>Client</i> terkonfirmasi	sesuai
	Menambah <i>client</i>	Menambah data	Pengisian form berfungsi dengan baik	sesuai
	Mengubah <i>client</i>	Memilih data yang akan dirubah	Data akan berubah sesuai inputan	Sesuai
	Menghapus <i>client</i>	Memilih data yang akan dihapus	Data berhasil dihapuskan	sesuai
	Melihat detail <i>client</i>	Melihat detail yang diinginkan	Data berhasil dilihat	sesuai
Mengubah Data Akun	Mengubah data akun	Memasukan data yang akan diubah	Data berhasil diubah	sesuai
Membaca notifikasi	Membaca notifikasi	Membaca notifikasi	Notifikasi berhasil dibaca	sesuai
Registrasi	registrasi	Memasukan data registrasi <i>client</i>	<i>Client</i> berhasil registrasi	sesuai