BAB II
TINJAUAN PUSTAKA

2. 1 Profil Institusi

Profil Institusi adalah untuk mengetahui keadaan institusi diantaranya adalah sejarah berdirinya, struktur organisasi serta visi dan misi.

2.1.1 Sejarah JackTour

PT. JACK WISATA adalah sebuah perusahaan yang bergerak dibidang jasa pelayanan *tour* dan *travel*, dengan pengembangan misi dan berorientasi kepada kepentingan *customer*, juga merupakan suatu hal yang menjadi komitmen PT. JACK WISATA dalam menjalankan usahanya. Saat ini lebih dari 100 *partnership* yang bergabung dan turut serta dalam pelaksanaan *tour arrangement*, baik dari bidang perhotelan, tempat wisata, kuliner dan *factory outlet*. PT. JACK WISATA lebih dikenal dengan brand "jacktour" dikalangan pengguna media social, teknologi informasi dan *traveller*. Jumlah komunitas facebook jacktour saat ini memiliki anggota lebih dari 2100 anggota, 87% *traveller* berasal dari mancanegara.

2.1.2 Visi dan Misi

Visi dari Jacktour adalah menjadi perusahaan terdepan di Indonesia dalam bidang jasa pelayanan pariwisata dengan kualitas pelayanan terbaik yang dapat memberikan kontribusi nyata bagi masyarakat dan lingkungan sekitarnya.

Adapun misi dari Jacktour adalah sebagai berikut :

1. Menjadi perusahaan penyedia layanan wisata dengan kualitas layanan prima.
2. Menjadi perusahaan yang selalu *up to date* dalam memberikan kenyamanan dan keamanan *customer* yang melakukan wisata.
3. Menjadi perusahaan yang selalu *respect* terhadap karyawan, lingkungan dan masyarakat sekitarnya.
4. Kami berkomitmen untuk menjadi perusahaan yang bekerja keras dengan mengutamakan kepuasan *customer,* melayani secara *profesional* dan sepenuh hati,

kami terus berusaha untuk tetap memperbaiki dan mengembangkan pelayanan agar menjadi yang terbaik dari yang baik.

5. Menjadi perusahaan layanan jasa *one stop service for tour and travel* yang tangguh dan unggul dengan jaringan mitra kerja yang luas.

2.1.3 Logo JackTour

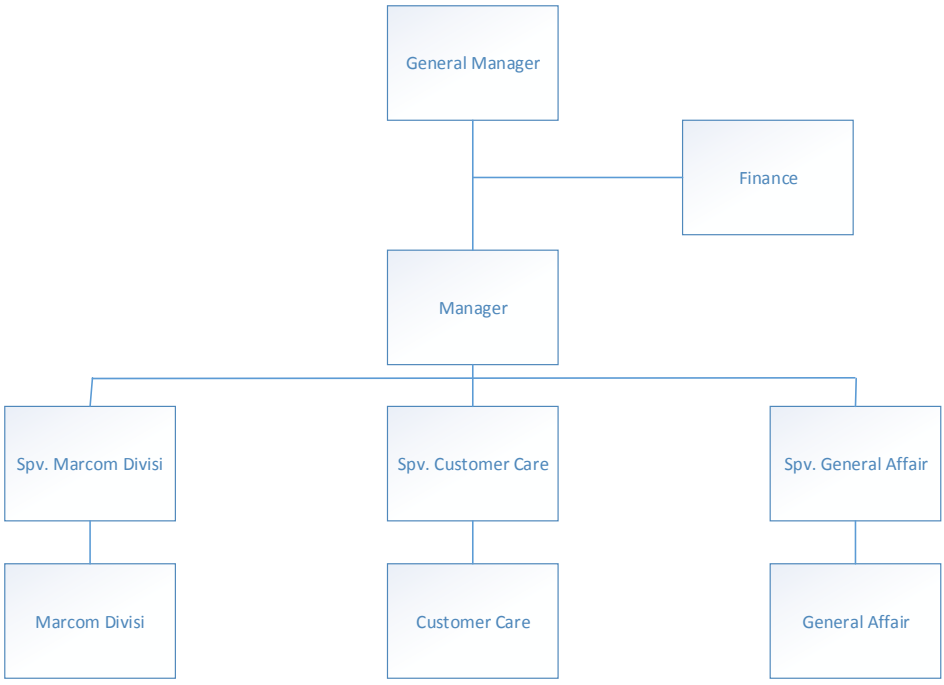
Berikut ini merupakan logo dari Jacktour seperti yang terdapat pada gambar 2.1



Gambar 2.1 Logo Jacktour

2.1.4 Struktur Organisasi

Berikut merupakan Struktur Organisasi dari Jacktour seperti yang terdapat pada gambar 2.2.



Gambar 2.2 Struktur Organisasi Jacktour

2.2 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan aplikasi mobile reservasi tour dan travel.

2.2.1 Pengertian Transportasi

Transportasi adalah pemindahan manusia atau barang dari satu tempat ke tempat lainnya dengan menggunakan sebuah wahana yang digerakkan oleh manusia atau mesin. Transportasi digunakan untuk memudahkan manusia dalam melakukan aktifitas sehari-hari. Transportasi dibagi menjadi 3 macam diantaranya

Transportasi Darat, Transportasi Laut, dan Transportasi Udara, namun transportasi yang paling banyak digunakan di Indonesia adalah transportasi darat dikarenakan biaya yang dikeluarkan lebih sedikit dibandingkan moda transportasi yang lain.

Transportasi darat adalah segala bentuk transportasi menggunakan jalan untuk mengangkut penumpang atau barang. Bentuk awal dari transportasi darat adalah menggunakan kuda, keledai atau bahkan manusia untuk membawa barang melewati jalan setapak. Seiring dengan berkembangkan perdagangan, jalan diratakan atau dilebarkan untuk mengakomodir aktivitas [2].

2.2.2 Travel (*point to point*)

Travel (*Point to Point*) adalah sebuah layanan angkutan penumpang dari Terminal / *Pool* Kota asal ke Terminal / *Pool* Kota tujuan sesuai trayek / jurusan yang dilayani Travel tersebut. Layanan Travel memiliki beberapa keuntungan bagi konsumen antara lain Lokasi Cabang/ *outlet* berada pada lokasi strategis, Cabang/*outlet* yang memiliki standar kenyamanan, keberangkatan terjadwal dan on time, seat penumpang yang terbatas, *driver* yang santun dengan melalui seleksi penerimaan yang ketat dan training yang berkelanjutan serta perlindungan asuransi kepada penumpang.

2.2.2.1 Pengertian Reservasi

Reservasi adalah perjanjian pemesanan, penyediaan atau pembukuan pada suatu tempat dan periode waktu tertentu bagi calon *customer* berikut dengan produk jasa tertentu. Pelayanan reservasi akan memberikan kemudahan dan kenyamanan kepada para *customer*.

Pelayanan reservasi akan memberikan jaminan kepastian mendapatkan tempat duduk yang merupakan tuntutan utama dari para calon penumpang, karena hal ini akan memberikan kepastian dan ketenangan bagi para calon penumpang untuk mencapai tujuan dalam menempuh perjalanan.

Disamping itu bila ada penumpang yang memerlukan pelayanan permintaan fasilitas-fasilitas tambahan (seperti kursi roda, menu makanan khusus, dan lainnya) akan dapat terlayani dengan baik. Pada umumnya kegiatan reservasi dapat dilaksanakan dengan dua metode, yaitu reservasi dengan sistem manual dan dengan komputer.

2.2.3 Kategori Transportasi

Transportasi sendiri dibagi 3 yaitu, transportasi udara, laut dan darat. Transportasi udara biasa kita kenal dengan pesawat, merupakan transportasi yang membutuhkan banyak uang untuk memakainya karena memiliki teknologi yang lebih canggih, serta merupakan alat transportasi tercepat dibandingkan dengan alat transportasi lainnya. Di laut biasa kita kenal dengan sebutan kapal, feri dan sampan. Sedangkan didarat kita kenal bermacam-macam kendaraan baik itu mobil, bus, ataupun kereta.

Dewasa ini, armada yang digunakan untuk transportasi umum didarat tidak hanya mobil, bus, ataupun kereta. Pekembangan zaman menuntut setiap 13 pengusaha dapat beradaptasi dengan perubahan dengan menciptakan strategi, inovasi, dan juga kreatifitas untuk menjaring penumpang.

Dibukanya akses tol mengakibatkan waktu tempuh suatu perjalan semakin cepat, hingga membuka kesempatan bagi pengusaha dalam sektor usaha jasa

transportasi. Persaingan yang semakin ketat membuat para pengusaha menciptakan kenyamanan bagi konsumen layaknya sedang dalam kendaraan pribadi.

Hingga mobil yang merupakan alat transpotasi keluarga telah beralih fungsi sebagai alat komersil, dijadikan sebagi ajang penghasil uang dengan menawarkan berbagai fasilitas mulai dari antar jemput, kemudahan pemesanan tiket, keamanan hingga kenyamanan didalam mobil, yang biasa disebut travel.

2.2.4 Kualitas Jasa

Pengertian kualitas jasa berpusat pada upaya pemenuhan kebutuhan dan keinginan *customer* serta ketetapan penyampaian untuk mengimbangi harapan pelangan.

Menurut Wykof "Kualitas jasa adalah tingkat keunggulan yang di harapkan dan pengendalian atas tinggkat keunggulan tersebut untuk memenuhi keinginan *customer*" [3].

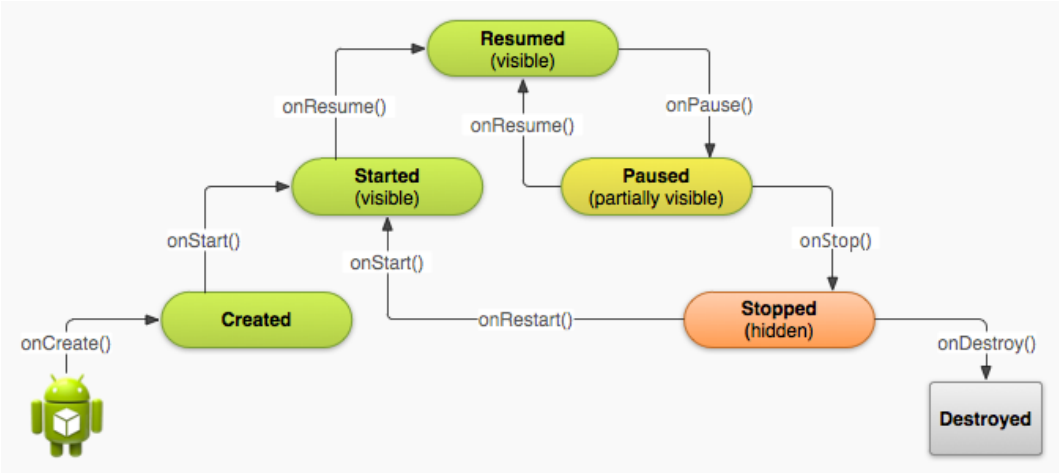
Ada dua factor utama yang mempengaruhi kualitas jasa, yaitu, *expected service* dan *perceived service* [4] . Apabila jasa yang diterima atau dirasakan sesuai dengan yang diharapkan, maka kualitas jasa dipersepsikan baik dan memuaskan. Jika jasa yang diterima melampaui harapan *customer*, maka kualitas jasa dipersepsikan kualitas yang ideal. Sebaliknya jika jasa yang diterima lebih rendah dari pada yang diharapkan,maka kualitas jasa dipersepsikan buruk.

2.2.5 *Android*

Android adalah sistem operasi *smartphone* layar sentuh sperti iOS iPhone dan OS Blackberry yang dalam pengembangannya dipimpin oleh google. Sistem Operasi ini bersifat *open source* dan dikembangkan berdasarkan kernel Linux yang disematkan pada gadget, baik itu *handphone* atau *tablet* [5]. Sedangkan android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi platform android menggunakan bahasa Java.

Di dalam *android* terdapat *activity* dimana komponen ini memberi interaksi antara user dan aplikasi yang dibangun melalui user *interface*. *Activity* ini memiliki

Siklus hidup yang dinamakan *Android Life Cycle*. Flowchart siklus hidup tersebut dapat dilihat pada Gambar 2.3.



Gambar 2.3 Android Life Cycle

Gambar 2.3 *Android Life Cycle* menjelaskan mengenai sebuah ilustrasi sederhana dari siklus hidup *activity* yang dinyatakan sebagai langkah piramida. Hal ini menunjukkan bagaimana untuk setiap *callback* yang digunakan untuk mengambil aktivitas langkah menuju *state* atas kemudian dilanjutkan ada metode *callback* yang mengambil langkah menurun. Kegiatan ini juga dapat kembali ke keadaan dari *statepause* dan *stop*. Siklus hidup *activity* ini menjadi sangat penting ketika kita hendak membuat aplikasi berbasis *android*. Gunanya adalah agar aplikasi yang dibangun berjalan dengan baik. Berikut penjelasan pada masing-masing *state*:

1. onCreate(), Method ini dipanggil ketika *activity* pertama kali dibuat.
2. onStart(),Method ini dipanggil ketika sebuah *activity* tampil ke pengguna.
3. onResume(), Method ini dipanggil ketika *activity* yang berjalan pada saat itu dihentikan sementara (*paused*) dan *activity* sebelumnya dijalankan kembali(*resumed*). (Hasil dari method OnRestart()) .
4. onPause(), Method ini di panggil ketika *activity* di hentikan sementara (*pause*) dan berikutnya ketika dijalankan kembali akan berada dalam posisi *resume* dan memanggil method OnResume() .
5. onStop(), Method ini dipanggil ketika *activity* tidak lagi tampak kepada pengguna .

6. onDestroy(), Method ini dipanggil sebelum *activity* dihancurkan (*destroy*) oleh sistem (baik secara manual maupun untuk kepentingan pelonggaran memori).

2.2.6 Java

Java adalah bahasa pemrograman serbaguna. Java dapat digunakan untuk membuat suatu program sebagaimana Anda membuatnya dengan bahasa seperti Pascal atau C++. Yang lebih menarik, Java juga mendukung sumber daya internet yang saat ini populer, yaitu *World Wide Web* atau yang sering disebut web. Java juga mendukung aplikasi klient/server, baik dalam jaringan local (LAN) maupun jaringan berskala luas (WAN) [5].

Java dikembangkan oleh Sun Microsystems pada Agustus 1991, dengan nama semula Oak. Konon Oak adalah pohon semacam Jati yang terlihat dari jendela tempat pembuatnya, James Gosling, bekerja. Ada yang mengatakan bahwa Oak adalah singkatan dari "Object Application Kernel", tetapi ada yang menyatakan hal itu muncul setelah nama Oak diberikan. Pada Januari 1995, karena nama Oak dianggap kurang komersial, maka diganti menjadi Java.

Dalam sejumlah literatur disebutkan bahwa Java merupakan hasil perpaduan sifat dari sejumlah bahasa pemrograman, yaitu C, C++, Object-C, SmallTalk, dan Common LISP. Selain itu Java juga dilengkapi dengan unsur keamanan. Yang tak kalah penting adalah bahwa Java menambahkan paradigma pemrograman yang sederhana. Jika telah mengenal C atau C++, yang mengandalkan pointer, Java justru meninggalkannya sehingga anda akan memperoleh kemudahan saat menggunakannya.

Program java bersifat tidak bergantung pada *platform*, artinya, Java dapat dijalankan pada sembarang komputer dan bahkan pada sembarang sistem operasi. Ketidakbergantungan pada platform sering dinyatakan dengan istilah portabilitas. Yang menarik, tingkat portabilitas java tidak hanya sebatas pada program sumber (*source code*), melainkan juga pada tingkat kode biner yang disebut *bytecode*. Dengan demikian bila telah mengkompilasi program java pada komputer berbasis

operasi Windows, dapat juga menjalankan hasil kompilasi pada Macintosh secara langsung, tanpa perlu mengkompilasi ulang.

2.2.7 *Android Studio*

Android Studio adalah sebuah *Integrated Development Environment* (IDE) untuk pengembangan aplikasi di platform *Android*. *Android Studio* pertama kali diperkenalkan pada tanggal 16 Mei 2013 oleh Google produk manager di Google I/O konference. *Android Studio* menyediakan alat pemrograman *Android* terintegrasi untuk pengembangan dan *debugging*.

Android Studio didasarkan pada kekuatan, *extensible IntelliJ IDEA Community Edition*, salah satu yang paling canggih dari Java IDE yang tersedia saat ini. Perangkat lunak ini telah ditambahkan dengan fitur yang dirancang dengan mengingat proses pengembangan aplikasi *Android*. Hal ini akan sangat menyederhanakan dan mengoptimalkan seluruh proses *software* desain dan pengembangan.

Android Studio mempunyai fitur Lint yang merupakan alat untuk mengatasi masalah kompabilitas versi dan bug potensial lainnya. Selain itu terdapat fitur ProGuard yang digunakan untuk menghambat *Reverse Engineering*. Perangkat lunak ini juga memiliki Gradle, yang digunakan sebagai alat otomatisasi dalam proses pengembangan perangkat lunak *mobile Android*.

Android Studio ini dipakai sebagai media untuk pembangunan aplikasi mobile reservasi paket tour ini.

2.2.8 Basis Data

Menurut Connoly dan Begg, basis data adalah sekumpulan koleksi data yang berhubungan secara logikal, dan sebuah deskripsi dari data tersebut, di desain untuk menemukan keperluan informasi pada sebuah perusahaan [7]. Menurut James A. O'Brien, basis data adalah kumpulan terintegrasi dari elemen data yang secara logika saling berhubungan [8]. Menurut Whitten, basis data adalah kumpulan *file* yang saling terkait [9].

Berdasarkan para ahli di atas, maka basis data adalah kumpulan data yang saling berhubungan secara logikal yang dapat digunakan untuk membantu dalam pengambilan keputusan pada sebuah organisasi atau perusahaan.

2.2.9 PHP

PHP adalah program aplikasi yang bersifat server side, artinya hanya dapat berjalan pada sisi server saja dan tidak dapat berfungsi tanpa adanya sebuah server di dalamnya. PHP juga bukan bahasa pemograman yang lengkap.

Maksudnya program ini tidak menyertakan compiler tersendiri yang membuat program hasilnya menjadi program .exe yang dapat di jalankan tersendiri. PHP (*Hypertext Preprocessor*) merupakan sebuah bahasa pemograman scripting berlisensi *Open Source*. *Script* ini dapat bercampur dengan *script Tag* HTML sehingga karena kemampuannya tersebut, ia disebut bahasa yang embeded pada *Tag* HTML.

Bahasa pemrograman PHP ini dipakai untuk pembuatan *web service* yang akan dipanggil oleh aplikasi *mobile*.

2.2.10 JSON PHP

JSON (*JavaScript Object Notation*) adalah sebuah format data-*interchange* yang ringan. Sangat mudah bagi manusia untuk membaca dan menuliskannya. Semudah mesin memparsing dan men-generate format datanya.

JSON merupakan bagian dari bahasa pemrograman JavaScript (*Standard ECMA-262 3rd Edition – December 1999*). JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C, termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan sebagainya. Kelebihan inilah yang membuat JSON menjadi sebuah bahasa data-*interchange* yang ideal.

JSON dibangun dalam dua struktur:

1. Beberapa pasangan dari nama/nilai. Dalam beberapa bahasa perograman biasa disebut dengan istilah *object, record, struct*, tabel *hash, key list* atau *associative array*.

2. Nilai-nilai yang terusun secara ordered list. Biasa disebut dengan *array, vector, list* atau daftar dalam bahasa pemrograman.

JSON adalah struktur data yang *universal*, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format.

Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON. Format data JSON mempunyai aturan sebagai berikut:

1. Object adalah satu set nama/nilai yang tidak terurut (*An object is an unordered set of name/value pairs*). Penulisan *object* dimulai dengan tanda { (*left brace*) dan diakhiri dengan tanda } (*right brace*). Setiap nama diikuti oleh tanda : (*colon*) dan pasangan nama/nilai dipisahkan dengan tanda , (*comma*).
2. *Array* adalah sekumpulan nilai yang teratur (*An array is an ordered collection of values*). Penulisan sebuah array dimulai dengan tanda [(*left bracket*) dan diakhiri dengan tanda] (*right bracket*). Nilai dipisahkan menggunakan tanda , (*comma*).
3. Nilai bisa berupa *string* dalam tanda kutip, atau *number* (angka), *TRUE* atau *FALSE* atau *NULL*, sebuah *object* atau sebuah array. Struktur ini dapat ditulis menggunakan metode bersarang.
4. *String* adalah rangkaian atau urutan karakter *unicode* yang berada dalam tanda kutip, bisa juga hanya berisi karakter kosong, menggunakan tanda \ (*backslash*) untuk *escape*. Karakter di representasikan sebagai *string* tunggal. Tipe data *string* pada JSON sangat mirip dengan definisi pada bahasa C atau Java.
5. *Number* sangat mirip dengan definisi pada bahasa C atau Java, hanya saja tipe bilangan oktal dan heksadesimal tidak digunakan.

Ruang kosong (*Whitespace*) dapat diselipkan diantara pasangan atau token apa saja. Kecuali ***detail encoding***, yang secara jelas akan berfungsi untuk menjelaskan bahasa yang digunakan.

JSON sangat berguna saat kita sering melakukan pertukaran data pada sebuah aplikasi. Contoh nyata adalah penggunaan tabulasi menggunakan JqGrid, yang

merupakan salah satu *plugin* jQuery untuk menampilkan data dalam bentuk spreadsheet. Plugin ini juga mendukung format JSON, selain format XML.

JSON disini dipakai sebagai pertukaran data dengan server, sehingga data api yang akan dipanggil oleh aplikasi *mobile* berbentuk JSON.

2.2.11 MySql

Menurut Abdul Kadir mendefinisikan bahwa MySQL merupakan *software* yang tergolong database *server* dan bersifat *Open Source*. *Open Source* menyatakan bahwa software ini di lengkapi oleh *source code* (kode yang di pakai untuk membuat *MySQL*), selain tentu saja bentuk *executable*-nya atau kode dapat di jalankan secara langsung di dalam sistem operasi, dan bias diperoleh dengan cara mengunduh di internet secara gratis. Hal lainnya adalah *MySQL* juga bersifat multiplatform. *MySQL* dapat dijalankan pada berbagai sistem operasi [10].

MySQL disini dipakai sebagai database dari aplikasi mobile reservasi paket tour yang dibangun.

2.2.11.1 DDL (*Data Definition Language*)

DDL merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur database yang digunakan untuk membuat mengubah dan menghapus struktur dan definisi metadata dari objek-objek database.

a. CREATE

Perintah ini digunakan untuk membuat, termasuk diantaranya membuat database baru, tabel baru, *view* baru, dan kolom. Contoh perintahnya :

Tabel 2.1 Contoh Perintah *Create*

```
create table mahasiswa (
nim  char(8) primary key,
nama_mahasiswa  varchar(20),
nilai integer(3),
alamat  varchar(25));
```

b. ALTER

Perintah ini digunakan untuk mengubah struktur tabel yang telah dibuat. Pekerjaannya mencakup mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom, maupun memberikan atribut pada kolom. Contoh perintahnya :

Tabel 2.2 Contoh Perintah *Alter*

```
alter table mahasiswa rename wisudawan;
```

c. RENAME

Perintah yang digunakan untuk merubah nama Objek

b. DROP

Perintah ini digunakan untuk menghapus database dan tabel. Contoh perintahnya :

Tabel 2.3 Contoh Perintah *Drop*

```
alter table ‘mahasiswa’ drop ‘alamat’;
```

2.2.11.2 DML (*Data Manipulation Language*)

DML merupakan perintah SQL yang digunakan untuk proses pengolahan isi data di dalam *table* seperti memasukkan, merubah dan menghapus isi data - dan tidak terkait dengan perubahan struktur dan definisi tipe data dari objek *database*. Contoh perintahnya :

a. UPDATE

Perintah ini digunakan untuk memperbaharui data lama menjadi data terkini. Jika anda memiliki data yang salah atau kurang *Up To Date* dengan kondisi sekarang, maka dapat diubah isi datanya dengan menggunakan perintah *UPDATE*. Contoh perintahnya :

Tabel 2.4 Contoh Perintah *Update*

```
Update mahasiswa set nama_mahasiswa = ‘Jourdan’ where nim = ‘08052927’;
```

b. INSERT

Perintah ini digunakan untuk menyisipkan atau memasukkan data baru ke dalam tabel. Penggunaannya setelah *database* dan tabel selesai dibuat. Contoh perintahnya :

Tabel 2.5 Contoh Perintah *Insert*

Insert into mahasiswa values (“08052926”, “Frenky”,”70”);

c. SELECT

Perintah ini digunakan untuk mengambil data atau menampilkan data dari satu tabel atau beberapa tabel dalam relasi. Data yang diambil dapat kita tampilkan dalam layar prompt MySQL secara langsung maupun ditampilkan pada tampilan aplikasi.

Tabel 2.6 Contoh Perintah *Select*

Select nama_mahasiswa from mahasiswa where nilai = 70;

d. DELETE

Perintah ini digunakan untuk menghapus data dari tabel. Biasanya data yang dihapus adalah data yang tidak diperlukan lagi. Pada saat menghapus data, perintah yang telah dijalankan tidak dapat digagalkan, sehingga data yang telah hilang tidak dapat dikembalikan lagi.

Tabel 2.7 Contoh Perintah *Delete*

delete form mahasiswa;

2.2.12 UML

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. *Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool

untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an.Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP.UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

UML ini dipakai untuk analisis dan perancangan sistem *mobile* reservasi paket *tour* yang dibangun.

2.2.12.1 Sejarah UML

Sampai era tahun 1990 puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan kelompok/perusahaan lain yang menggunakan metodologi yang berlainan. Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group* [10].

2.2.12.2 *Diagram UML*

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

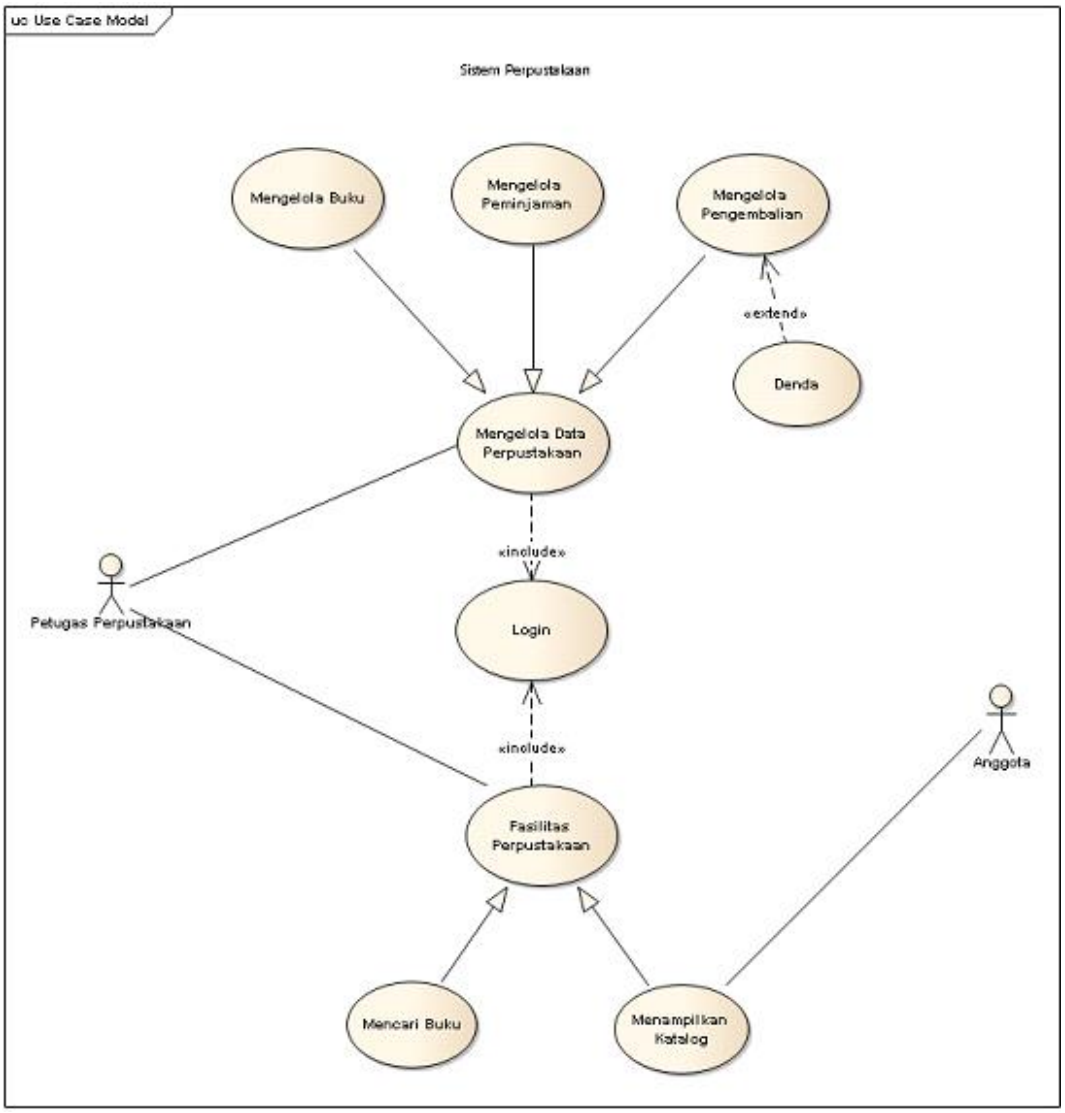
1. *Use Case* Diagram untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects*.
4. *Collaboration Diagram* untuk memodelkan interaksi antar *objects*.
5. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku *Use Case* dan *objects* di dalam system.
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur *object*.
9. *Component Diagram* untuk memodelkan komponen *object*.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

2.2.12.2.1 *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal. Sebuah *use case* dapat diinclude oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-

extend use case lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case*

yang satu merupakan spesialisasi dari yang lain. Penggambaran *Use case* diagram dapat di lihat dalam gambar 2.4 di bawah ini.

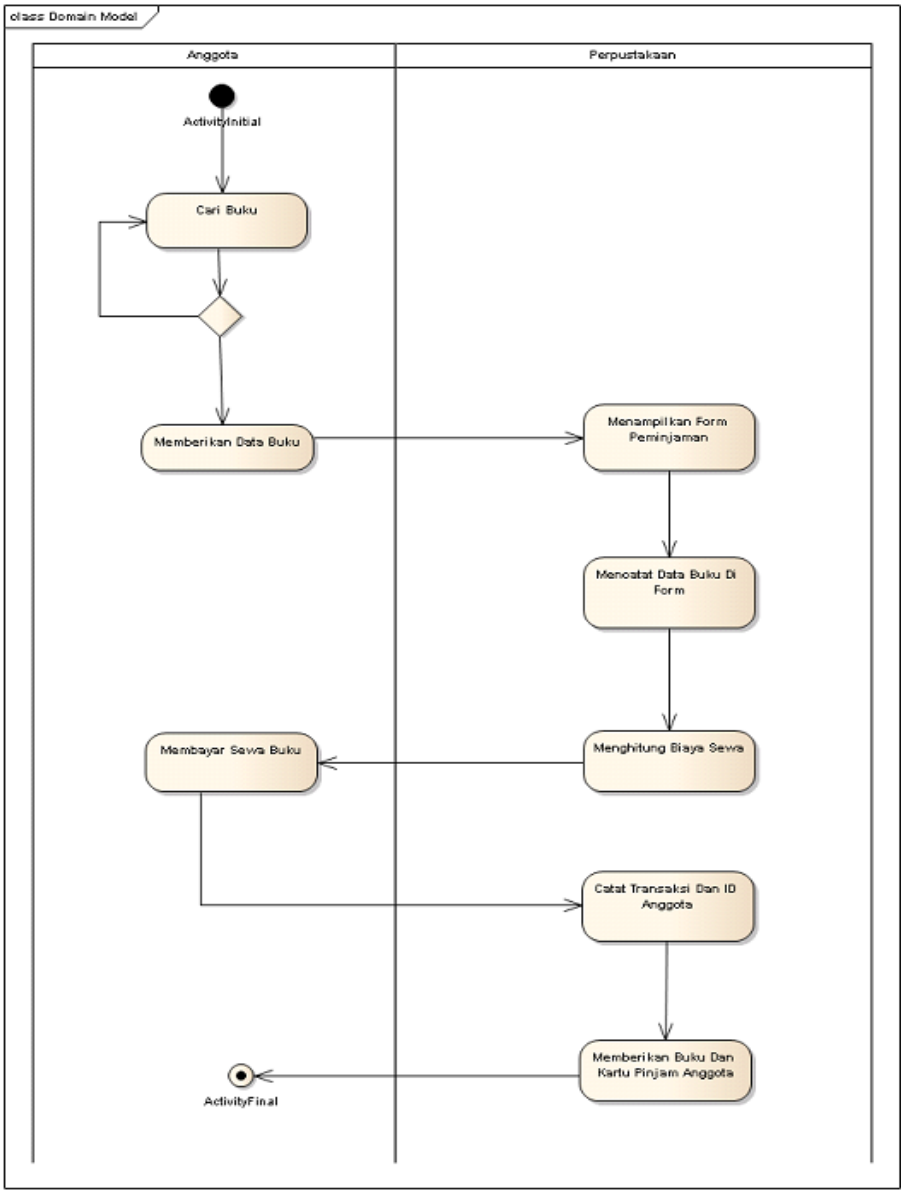


Gambar 2.4 ***Usecase Diagram***

2.2.12.2.2 *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih.

Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis *horizontal* atau *vertikal*. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Contoh *Activity diagram* dapat di lihat pada gambar 2.5 berikut ini.

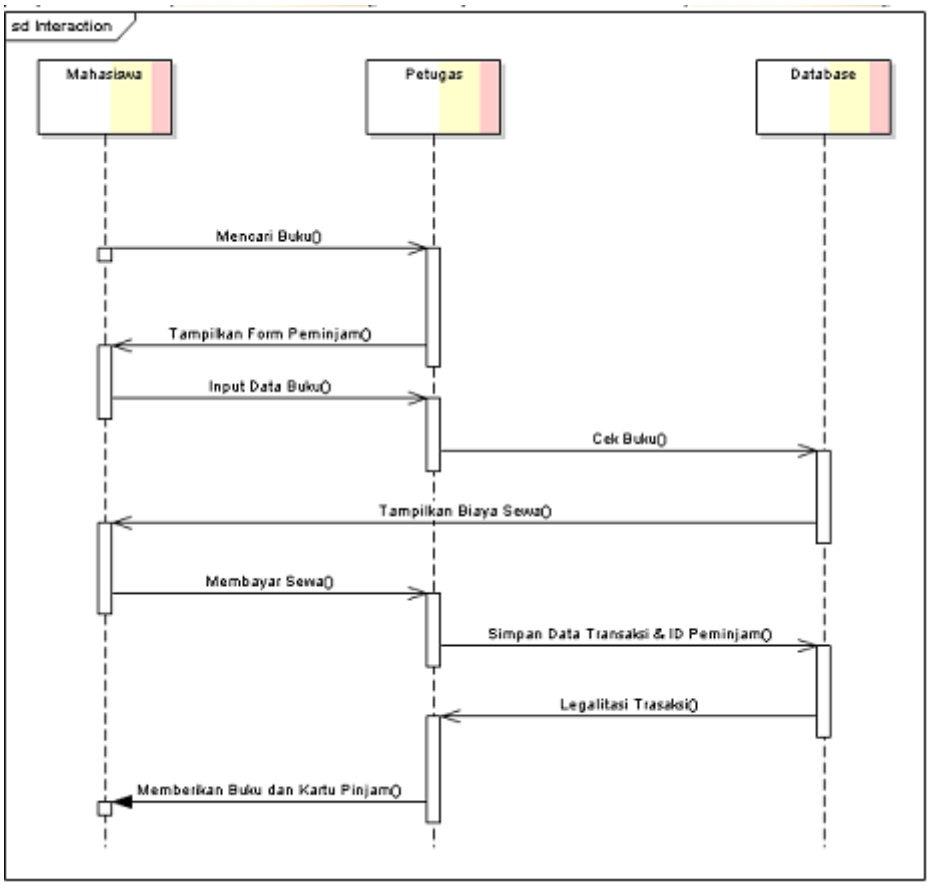


Gambar 2.5 ***Activity Diagram***

2.2.12.2.3 *Sequence Diagram*

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan

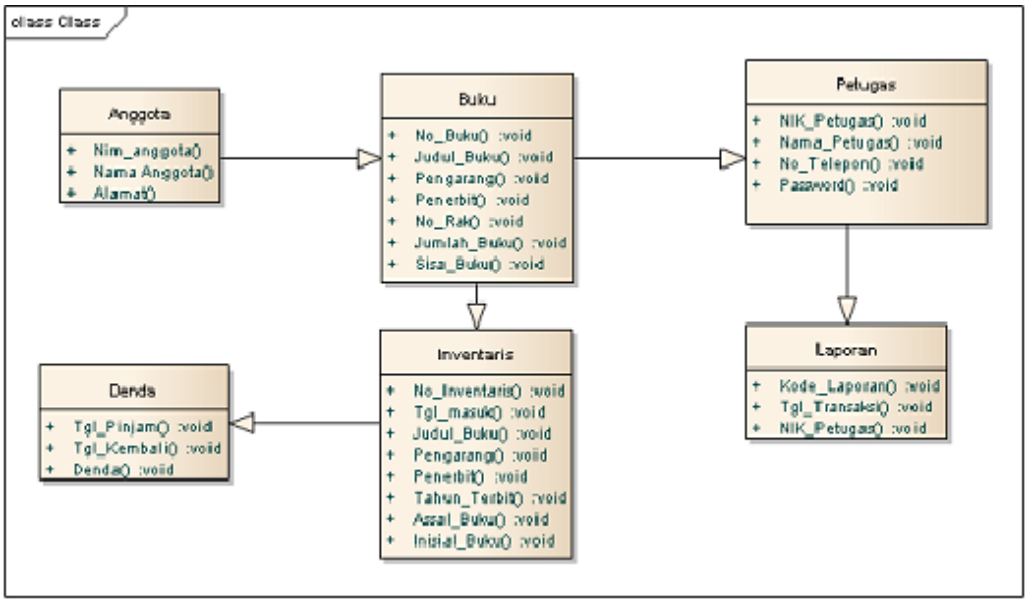
oleh masing-masing operasi. Contoh *sequence diagram* dapat di lihat pada gambar 2.6 berikut ini.



Gambar 2.6 *Sequence Diagram*

2.2.12.2.4 *Class Diagram*

Class diagram merupakan *diagram* yang selalu ada di permodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Penggambaran *class diagram* dapat di lihat dalam gambar 2.7 dibawah ini.



Gambar 2.7 *Class Diagram*

2.2.13 WEB

Layanan WWW (*World-Wide Web*) atau biasa juga disebut Web merupakan jenis layanan yang paling popular dikalangan pengguna internet. WWW tidak hanya berfungsi sebagai media untuk mencari informasi, tetapi web sudah banyak digunakan secara komersial oleh hampir semua perusahaanperusahaan diseluruh dunia untuk mengiklankan usaha mereka. Setiap dokumen yang ditulis menggunakan suatu format standar yang disebut HTML (*HypertText Markup Language*).

2.2.14 Dreamweaver

Dreamweaver merupakan *software* aplikasi yang digunakan sebagai HTML editor profesional untuk mendesain web secara *visual*. Aplikasi ini juga dikenal dengan istilah WYSIWYG (*What You See Is What You Get*) yaitu bahwa anda tidak harus berusaha dengan tag-tag HTML untuk membuat sebuah situs. Selain itu, Dreamweaver juga memberikan keleluasaan kepada pengguna untuk menggunakannya sebagai media penulisan bahasa pemrograman web. Dreamweaver memiliki kemampuan untuk dapat berinteraksi dengan beberapa bahasa pemrograman

seperti PHP, ASP, JavaScript, dan yang lainnya juga memberikan fasilitas maksimal kepada para desainer web yang menyertakan bahasa pemrograman web didalamnya.

Software aplikasi Dreamweaver dipakai untuk mendesain web yang dibangun dengan menggunakan bahasa pemrograman PHP.

2.2.15 PhpMyAdmin

PhpMyAdmin merupakan *tools open source* yang dibuat menggunakan program PHP untuk mengakses *database* MySQL *via* web. Dengan menggunakan program ini kita dapat mengelolah *database* MySQL dengan sangat mudah. Hampir semua operasi terhadap database dan perintah yang ada dapat dijalankan pada program tersebut, termasuk diantaranya menggunakan perintah *Database Definition Language* (DDL), *Database Manipulation Language* (DML) serta *Data Control Language* (DCL).