BAB II
TINJAUAN PUSTAKA

2.1 Profil Perusahaan

Profil perusahaan tempat penelitian tugas akhir ini akan menjelaskan mengenai sejarah perusahaan, struktur organisasi perusahaan serta visi dan misi di PT.Tunas Jaya Persada.

2.1.1 Sejarah Singkat Perusahaan

PT. Tunas Jaya Persada terletak di jalan Juanda 477 Dago Bandung 40135 . lokasinya sangat strategis terletak di jalan utama Dago di pusat kota. Pada tanggal 03 Desember 2011. Perkembangan PT. Tunas Jaya Persada yang begitu pesat. PT. Tunas Jaya Persada dibuka dengn alasan beberapa hal antara lain semakin meningkatnya jumlah wisatawan yang datang ke Bandung, dengan alasan itulah maka dibuka PT. Tunas Jaya Persada, dengan begitu kota bandung menjadi ramai dan persaingan antar travel-travel lain begitu pesat. PT. Tunas Jaya Persada memilih tempat di jalan Juanda 477 Dago Bandung 40135 dengan pertimbangan sudah cukup strategis karena arus lalu lintas kendaraan cukup padat sehingga mudah terlihat oleh para konsumen yang membutuhkan pelayanan jasa tour dan travel.

PT.Tunas Jaya Persada adalah sebuah perusahaan jasa transportasi di Bandung yg menyediakan unit kendaraan berbagai jenis dan berbagai kapasitas tempat duduk yg bisa anda pilih sesuai kebutuhan anda terutama kendaraan bus bisata bi Bandung. Kantor pemasaran berlokasi di Bandung dan memiliki wilayah pemasaran seluruh Jawa Barat, DKI Jakarta Jogjakarta dan Bali

Didukung dengan legalitas yang memadai kami sangat yakin dan percaya diri bisa memberikan layanan maksimal untuk anda jadikan referensi dan mitra kerja anda dalam urusan transportasi terutama untuk keperluan wisata, bisnis, lembaga pendidikan dan umum. Dengan tekad kuat untuk menjadi salah satu perusahaan jasa transportasi terdepan di Indonesia maka kami senantias

melakukan evaluasi terhadap seluruh aktivitas kami dari semua lini dan komponen bisnis terutama pengembangan sistim pemasaran, pelayanan konsumen serta pertumbuhan usaha. PT. Tunas Jaya Persada sebagai perusahaan jasa penyedia armada transportasi umum terbaik di Bandung, terdaftar, periszinan lengkap, aman & nyaman siap melayani anda dengan kemampuan terbaik kami. Berikut ini adalah logo perusahaan yang diberi nama seratour.



Gambar 2.1 Logo PT.Tunas Jaya Persada

Adapun arti logo dari Sera Tour itu senditi sebagai berikut:

Warna Orange dan Matahari Menunjukkan kehangatan, menerangi, antusiasme, persahabatan, pencapaian bisnis, karier, kesuksesan, kesehatan pikiran, keadilan, daya tahan, kegembiraan, gerak cepat, sesuatu yang tumbuh, ketertarikan, disamping itu warna Orange memberi kesan yang kuat pada elemen yang dianggap penting.

Warna Biru dan Gunung Mencerminkan kekokohan keamanan, kepandaian, tenang dan serius, kesederhanaan, kedewasaaan, praktis, profesional, kualitas.

Kapal Pesiar, merupakan sebagai alat transportasi yang biasa dipakai oleh wisatawan yang ingin berpergian hal ini juga bermakna sebagai perusahaan tour premium. Sudah menjadi aware masyarakat kalo kapal simbolisasi perjalan yang premium dan ekslusif. Dan ini merupakan maksud dari sebagai visulisasi Sera Tour sebagai travel agen yang premium dalam pelayanan kepada pelanggan.

2.1.2 Visi dan Misi Perusahaan

Berikut ini adalah visi dan misi dari PT.Tunas Jaya Persada:

a. Visi

PT. Tunas Jaya Persada menjadi salah satu perusahaan jasa transportasi terdepan di Indonesia yang mampu menciptakan kontribusi positif bagi pertumbuhan ekonomi dari sektor pariwisata dan UKM serta mampu menciptakan lapangan pekerjaan bagi banyak orang sehingga memberi manfaat bagi banyak orang.

b. Misi

Menyediakan jasa transportasi dengan layanan terbaik namun dengan harga murah.

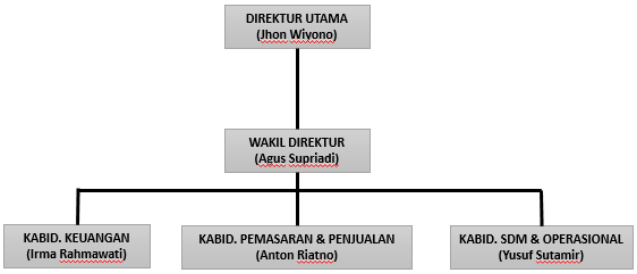
2.1.3 Stuktur Organisasi Perusahaan

Struktur adalah cara sesuatu disusun atau dibangun. Sedangkan organisasi sendiri adalah suatu wadah berkumpulnya minimal dua orang untuk mencapai sebuah tujuan. Maka struktur organisasi adalah suatu susunan dan hubungan antara tiap bagian baik secara posisi maupun tugas yang ada pada perusahaan dalam menjalin kegiatan operasional untuk mencapai tujuan. Struktur organisasi merupakan pembagian, pengelompokan dan pengkoordinasian pekerjaan secara formal. Dalam struktur organisasi terdapat beberapa elemen yang perlu diperhatikan dalam mendesain struktur organisasi yaitu : The Strategic Apex (Top Manager/Manajer Tingkat Puncak),The Midle Line (Midle Manager/Manajer Bagian Penghubung Top Manager dengan Employee), The Operating Core (Karyawan), The Technostructure (BagianAnalisis), The Support Staff (Staff Pendukung). Dan berikut beberapa desain organisasi :

1. Struktur Sederhana: berlaku bagi organisasi kecil.
2. Birokrasi Mesin: untuk organisasi dengan rutinitas berulang dan diatur oleh peraturan yang berlaku.
3. Birokrasi Profesional : untuk organisasi besar, dengan kegiatan rutin dan terdiri dari pada spesialis.

4. Struktur Divisional: untuk organisasi yang membutuhkan banyak pembagian divisi.
5. Adhocracy: untuk organisasi dengan tenaga profesional yang terkumpul pada suatu kegiatan tertentu.

Berikut gambaran struktur organisasi pada PT. Tunas Jaya Persada.



Gambar 2.2 Struktur Organisasi PT.Tunas Jaya Persada

Berikut ini adalah deskripsi jabatan di PT.Tunas Jaya Persada:

1. Direktur Utama

Direktur utama pada perusahaan ini merupakan posisi jabatan tertinggi yang memiliki tugas sebagai berikut :

a. Memimpin Perusahaan, Staff, Karyawan
b. Mengambil Keputusan (terutama dalam situasi darurat) jika cara demokrasi tidak memungkinkan lagi.
c. Mengontrol Kinerja Bawahan.
d. Mengevaluasi Perkembangan Perusahaan.
e. Pembuat Kebijakan

2. Wakil Direktur

Tugas-tugas wakil direktur pada perusahaan ini adalah :

a. Membantu Direktur Utama.
b. Menggantikan posisi Direktur Utama sebagai Pimpinan tertinggi jika beliau sibuk dengan aktifitas lainnya.

c. Meninjau perkembangan kinerja Kepala Bidang dan
d. Memberikan laporan kepada Direktur utama

3. Kepala Bidang Keuangan

Tugas-tugas kepala bidang keuangan :

a. Melaporkan & mempertanggungjawabkan data-data keuangan selama periode yang ditentukan (harian, mingguan,bulanan) kepada pimpinan.
b. Mengkordinasikan bawahan.
c. Memberikan usulan mengenai investasi keuangan.
d. Mengurus masalah penggajian Karyawan.

4. Kepala Bidang Pemasaran & Penjualan

Tugas-tugas kepala bidang pemasaran :

a. Mencatat Data-data transaksi
b. Memberikan Laporan penjualan / pemasaran yang telah dicatat
c. Mempromosikan Perusahaan ke pasar yang lebih luas
d. Menjaga kestabilan pelanggan
e. Mengelola anggaran pemasaran perusahaan

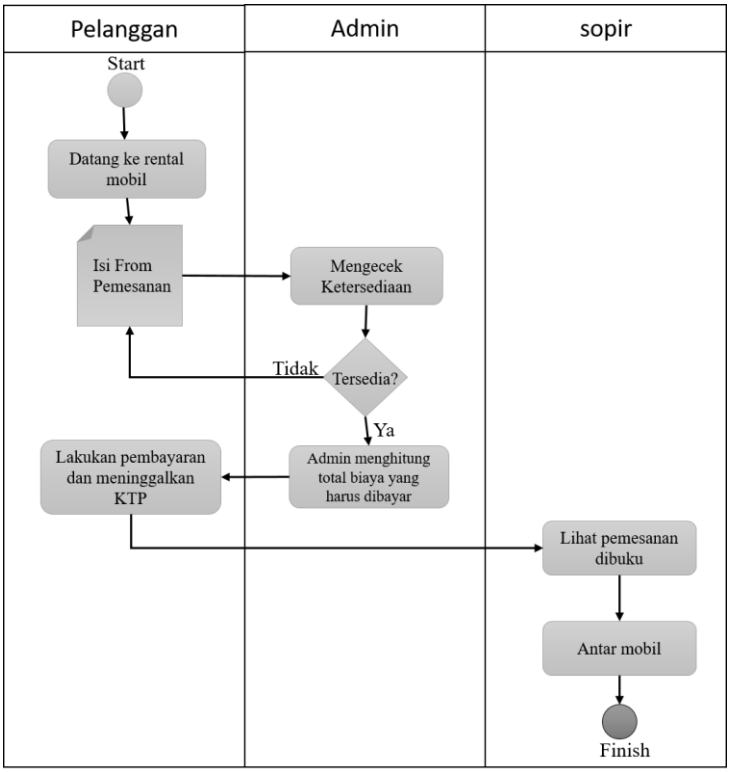
5. Kepala Bidang SDM & Operasional

Tugas-tugas kepala bidang SDM (Sumber Daya Manusia) :

a. Memberikan pelatihan kepada tenaga kerja
b. Merekrut tenaga kerja baru
c. Mempertanggungjawabkan kelancaran Operasional

2.1.4 Proses bisnis penyewaan

Berikut alur Proses bisnis penyewaan



Keterangan gambar:

1. Pelanggan datang ke rental mobil dan mengisi form penyewaan seperti identitas, tanggal penyewaan, lama waktu pemsanan, alamat pengataran, apakah pakai supir atau tidak.
2. Admin mengecek data dan ketersediaan mobil dibuku.
3. Jika tersedia admin melakukan perhitungan total biaya peminjaman mobil .
4. Pelanggan melakukan pembayaran dan meninggalkan KTP.
5. Admin memberi data penyewaan mobil berdasarkan form penyewaan pelanggan kepada supir.
6. Supir mengantar mobil ke pelanggan .
7. Proses selesai.

2.2 Driver

Driver adalah pengemudi profesional yang dibayar oleh majikan untuk mengemudi kendaraan bermotor. Driver dibagi menjadi dua bagian yaitu driver pribadi yang menggunakan kendaraan pribadi dan yang kedua adalah driver perusahaan yang bekerja untuk perusahaan angkutan umum seperti taksi, bus, ataupun angkutan barang [3].

2.3 Aplikasi Mobile

Aplikasi *Mobile* adalah sebuah aplikasi yang memungkinkan Anda melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau *Handphone*. Dengan menggunakan aplikasi *mobile*, Anda dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing* dan lain sebagainya[4]. Contohnya mobile programing untuk ponsel:

1. Javafx mobile, digunakan di smartphone Android.
2. J2ME (Java 2 Mobile Edition) yang paling umum dan paling banyak yang support, blackberry pun masih menggunakan J2ME dengan tambahan API. J2ME adalah satu set spesifikasi dan teknologi yang fokus kepada perangkat konsumen. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterei, layar yang kecil dan bandwith jaringan yang rendah.
3. C++ dalam symbian framework, ini khusus untuk handphone dengan OS symbian.
4. C# .NET, untuk handphone dengan OS Windows mobile
5. Flash Lite, game flash di mobile.

Situs web (bahasa Inggris: *website*) adalah suatu halaman web yang saling berhubungan yang umumnya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi. Sebuah situs web biasanya ditempatkan setidaknya pada sebuah server web yang dapat diakses melalui jaringan seperti Internet, ataupun jaringan wilayah lokal (LAN) melalui alamat Internet yang dikenali sebagai URL. Gabungan atas semua situs

yang dapat diakses publik di Internet disebut pula sebagai World Wide Web atau lebih dikenal dengan singkatan WWW.

Meskipun setidaknya halaman beranda situs Internet umumnya dapat diakses publik secara bebas, pada prakteknya tidak semua situs memberikan kebebasan bagi publik untuk mengaksesnya, beberapa situs web mewajibkan pengunjung untuk melakukan pendaftaran sebagai anggota, atau bahkan meminta pembayaran untuk dapat menjadi aggota untuk dapat mengakses isi yang terdapat dalam situs web tersebut, misalnya situs-situs yang menampilkan pornografi, situs-situs berita, layanan surel (*e-mail*), dan lain-lain. Pembatasan-pembatasan ini umumnya dilakukan karena alasan keamanan, menghormati privasi, atau karena tujuan komersil tertentu.

Sebuah halaman web merupakan berkas yang ditulis sebagai berkas teks biasa (plain text) yang diatur dan dikombinasikan sedemikian rupa dengan instruksi-instruksi berbasisHTML, atau XHTML, kadang-kadang pula disisipi dengan sekelumit bahasa skrip. Berkas tersebut kemudian diterjemahkan oleh peramban web dan ditampilkan seperti layaknya sebuah halaman pada monitor komputer.

Halaman-halaman web tersebut diakses oleh pengguna melalui protokol komunikasi jaringan yang disebut sebagai HTTP, sebagai tambahan untuk meningkatkan aspek keamanan dan aspek privasi yang lebih baik, situs web dapat pula mengimplementasikan mekanisme pengaksesan melalui protokol HTTPS.

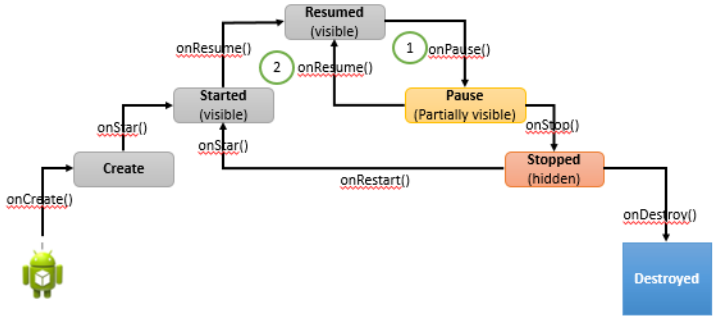
2.4 Android

Android adalah sistem operasi bersifat *Open Source* berbasis Linux dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance*, *konsorsium* dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk

memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008 [5]. Android memiliki OS yang sangat baik, cepat dan kuat serta memiliki antarmuka pengguna intuitif yang dikemas dengan pilihan dan fleksibilitas.

1. Android SDK (*Software Development Kit*) menyediakan *tools* dan API yang diperlukan untuk mengembangkan aplikasi pada platform android dengan menggunakan bahasa pemrograman Java.

2. Android memiliki paradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi main(), sistem android menjalankan kode dalam *method Activity* dengan menerapkan metode *callback* tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi android memiliki siklus hidup yang berbeda dengan aplikasi desktop atau *web*, Hal ini dikarenakan aplikasi *mobile* memiliki tingkat interupsi proses yang lumayan tinggi seperti ketika *handling* panggilan masuk aplikasi diharuskan menghentikan proses sementara, Penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai pengguna.
3. 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.



Gambar 2.3 Android Life Cycle

4. Terdapat beberapa *state* dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gambar 2.3 *Android Life Cycle*, akan tetapi hanya beberapa dari state tersebut yang menjadi statis diantaranya:
    1. Resumed

       Resumed terjadi ketika aplikasi berjalan setelah state paused. State ini akan menjalankan perintah program yang ditulis pada method onResume().
    2. Paused

       Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karena terdapat proses yang memiliki prioritas lebih tinggi seperti panggilan telepon. Aplikasi tidak dapat menjalankan perintah apapun ataupun menampilkan apapun dalam state ini.
    3. Stopped

       Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan service dibackground.
5. State lain seperti Created dan Started bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode *life cycle callback* berikutnya. Artinya, setelah sistem OnCreate() dipanggil, dengan cepat sistem akan memanggil method OnStart(), krmudian diikuti oleh onResume().
6. Android memiliki beberapa fitur utama yang sering digunakan dalam proses pembangunan aplikasi diantaranya adalah:
    1. Multi-proses dan *App Widgets*

       Sistem operasi android tidak melarang prosesor menjalankan lebih dari satu aplikasi dalam satu waktu. Sistem operasi android dapat mengatur aplikasi dan thread yang berjalan secara *multitasking.* Keuntungan yang didapat adalah ketika aplikasi berjalan dan berinteraksi dengan pengguna di layer depan sistem operasi, proses dari aplikasi lain dapat berjalan untuk melakukan pembaruan informasi. Sebagai contoh misalnya ketika pengguna memainkan game, proses lain dapat berjalan di

belakang aplikasi seperti memeriksa harga saham dan memunculkan peringatan.

App Widgets adalah mini aplikasi yang dapat *embedded* dalam aplikasi seperti home screen. App widgets dapat menjalankan proses *request* seperti musik streaming atau mendeteksi suhu ruangan secara *background*. *Multi-proses* dapat memberikan manfaat berupa *user experience* yang lebih banyak, namun penggunaan fitur tersebut dapat menghabiskan banyak energi baterai jika penggunaan tidak benar.

2. *Touch Gestures* dan *Multi-touch*

   Touchscreen adalah *user interface* intuitif yang digunakan banyak *smartphone* di dunia. Dengan fitur ini interaksi dapat dibuat lebih mudah karena cukup dengan menggunakan jari tangan. *Multi-touch* adalah kemampuan yang dapat melakukan tracking lebih dari satu tangan dalam satu waktu, Fitur ini sering digunakan untuk interaksi memperbesar atau memutar objek. Selain itu pengembang dapat membuat interaksi baru dengan memanfaatkan fitur tersebut.

3. *Hard* dan *Soft Keyboard*

   Salah satu fitur pada perangkat smartphone adalah tombol fisik dan non fisik, tombol fisik digunakan untuk navigasi pendukung dalam pengoperasian android. Pengembang aplikasi tidak perlu secara manual untuk mengintegrasikan tombol tersebut dalam aplikasi. Tombol non fisik adalah tombol yang dibuat oleh sistem operasi seperti *virtual keyboard*, dan tombol navigasi aplikasi.

2.4.1 Android SDK

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google[6]. Saat ini disediakan Android SDK (*Software*

Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. sebagai platform aplikasi netral, Android memberi kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone/Smartphone*. Beberapa fitur - fitur Android yang paling penting adalah:

a. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*
b. Mesin *virtual Dalvik* dioptimalkan untuk perangkat *mobile*
c. *Integrated browser* berdasarkan *engine open source WebKit*
d. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasrkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware)
e. SQLite untuk penyimpanan data (*database*)
f. Media Support yang mendukung audio, video, dan gambar.
g. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
h. Kamera, GPS, kompas, dan *accelerometer* (tergantung hardware)
i. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

2.5 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)[7].

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.

2.6 ECLIPSE JUNO

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent)[8]. Berikut ini adalah sifat dari Eclipse:

1. *Multi-platform*: Target sistem operasi *Eclipse* adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX* dan *Mac OS X.*
2. *Mulit-language*: *Eclipse* dikembangkan dengan bahasa pemrograman Java, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++, Cobol, Python, Perl, PHP*, dan lain sebagainya.

3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari *Eclipse* yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in.*

2.7 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.[9].

2.8 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[9].

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.

2.9 HTML

HTML adalah, (*HyperText Markup Language*) sebuah bahasa standar yang di gunakan oleh browser internet untuk membuat halaman dan dekumen pada sebuah web yang kemudian dapat diakses dan dibaca layaknya sebuah artikel. HTML juga dapat digunakan sebagai *link-link* antara data-data dalam situs atau dalam komputer dengan menggunakan *localhost,*atau *link* yang menghubungkan antara situs dalam dunia internet [10].

2.10 CSS

CSS(*Cascading Style Sheet*) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft word* yang dapat mengatur beberapa *style,* misalnya *heading,* subbab, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa data. Pada umumnya CSS dipakai untuk memfotmat tampilan web yang dibuat dengan bahasa HTML dan XHTML[10].

2.11 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generated*) oleh komputer[11]. JSON terbuat dari dua struktur yaitu:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik *(array*), vektor (*vector*), daftar (*list*), atau urutan (*Sequence*).

JSON menggunakan bentuk sebagai berikut:

1. Objek

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

2. Larik

   Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).

3. Nilai

   Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau true atau false atau null, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.

4. String

   String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. *String* sangat mirip dengan *string* C atau Java.Angka, *Format oktal* dan heksadesimal tidak digunakan.

2.12 Structure Analysis dan Desagn

Struktur analisis dan desain adalah sebuah metodologi yang di gunakan pada rekayasa perangkat lunak untuk mendeskripsikan sistem kearah fungsional. Pendekatan ini memecahkan masalah-masalah dalam aktifitas bisnis menjadi bagian-bagian kecil yang dapat disatukan kembali menjadi satu kesatuan yang utuh untuk memecahkan masalah

Tools yang dapat digunakan pada pendekatan analisis pengembangan sistem secara terstruktur adalah[12]:

1. **ERD (*Entity Relationship Diagram*)**

Merupakan satu model jaringan yang menggunakan susunan data yang disimpan dalam sistem. ERD merupakan model jaringan data yang menekankan pada struktur dan hubungan antar data Elemen-elemen ERD

a. Entitas

Entity (entitas) adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak di mana data tersimpan atau di mana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam 4 (empat) kelas, yaitu : *role* (peran), *events* (kejadian), *locations* (lokasi), *tangible things* / *concepts* (sesuatu yang tidak nyata / konsep). Contoh: pegawai, pembelian, kampus, buku, pembayaran, mahasiswa dan dosen.

b. Relationship

Relationship adalah hubungan alamiah yang terjadi antara entitas. Contohnya: Entitas Mahasiswa dengan NIM = "14534" dan NamaMhs = "Dodo" yang mempunyai relasi dengan Entitas Kuliah dengan KodeKul = "SI-140" dan NamaMK = "Basis Data", sehingga struktur data dari Relasi ini bahwa mahasiswa tersebut mengambil mata kuliah pada suatu perguruan tinggi.

c. Atribut

Deskripsi kelompok data yang mempunyai karakteristik yang sama (data yang mendeksripsikan entity dan relationship), merupakan field yang akan disimpan.

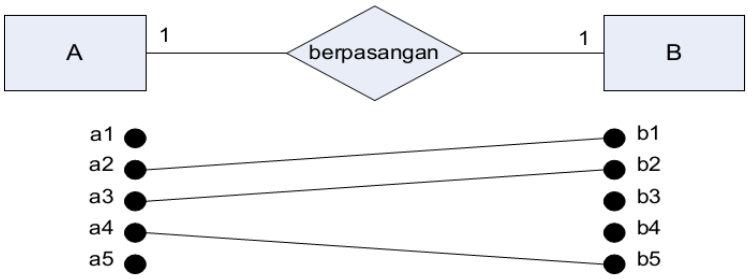
Hubungan antara entitas akan menyangkut dua komponen yang menyatakan jalinan ikatan yang terjadi, yaitu[10]:

a. Derajat (kardinalitas).

Kardinalitas menyatakan jumlah anggota entitas yang terlibat di dalam relasi yang terjadi. Dalam hal ini relasi yang terjadi akan membentuk relasi hubungan (relationship instances). Derajat kardinalitas dibagi menjadi 3, yaitu:

a) Derajat kardinalitas 1 : 1

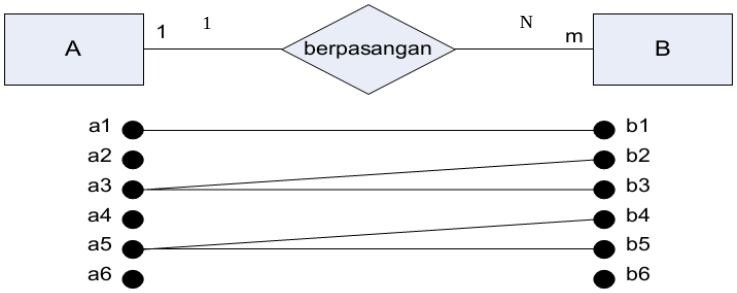
Derajat hubungan 1:1 terjadi bila setiap anggota entitas A hanya boleh berpasangan dengan satu anggota dari entitas B, dan sebaliknya tiap anggota entitas B hanya boleh berpasangan dengan satu anggota dari entitas A. Contoh dari kardinalitas 1:1 dapat dilihat pada Gambar 2.4 Derajat kardinalitas 1:1.



Gambar 2.4 Derajat kardinalitas 1:1

b) Derajat kardinalitas 1:N

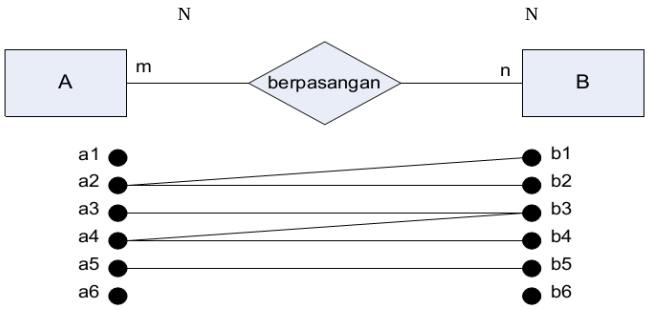
Derajat hubungan ini terjadi bila tiap anggota entitas A boleh berpasangan dengan lebih dari satu anggota entitas B. Sebaliknya tiap anggota entitas B hanya boleh berpasangan dengan satu anggota entitas A. Contoh dari kardinalitas 1:N dapat dilihat pada Gambar 2.5 Derajat kardinalitas 1:N.



Gambar 2.5 Derajat kardinalitas 1:N

c) Derajat kardinalitas N:N

Derajat hubungan antar entitas N:N terjadi bila tiap anggota entitas A dapat berpasangan dengan lebih dari satu anggota entitas B. Sebaliknya setiap anggota entitas B juga dapat berpasangan dengan lebih dari satu anggota entitas A. Contoh dari kardinalitas N:N dapat dilihat pada Gambar 2.6 Derajat kardinalitas N:N.



Gambar 2.6 Derajat kardinalitas N:N

2. Partisipasi hubungan.

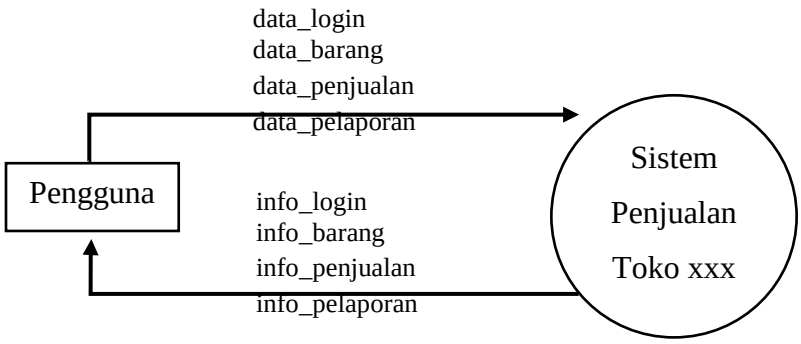
Partisipasi hubungan (*participation*) menyatakan sifat keterlibatan tiap anggota entitas dalam relasi terjadinya hubungan. Partisipasi atau keterlibatan tiap anggota entitas dalam membentuk instance hubungan dapat bersifat wajib (mandatory) dan tidak wajib (non mandatory). Dalam pemodelan data, interpretasi jenis partisipasi hubungan dituliskan dalam aturan data. Contoh misal hubungan antara entitas dosen dan mata kuliah. Berikut keterhubungan kardinalitas dan partisipasi pada tabel 2.1

Tabel 2.1 Kardinalitas dan partisipasi

Kardinalitas	Partisipasi
1-1 (*one to one*)	0..1 *(optional one*)
1-N (*one to many*)	0..N atau 1..N (*optinal many)*
N-1 (*many to one*)	1 (*mandatory one*)
N-N (*many to many*)	N (*mandatory many*)

3. Diagram Konteks

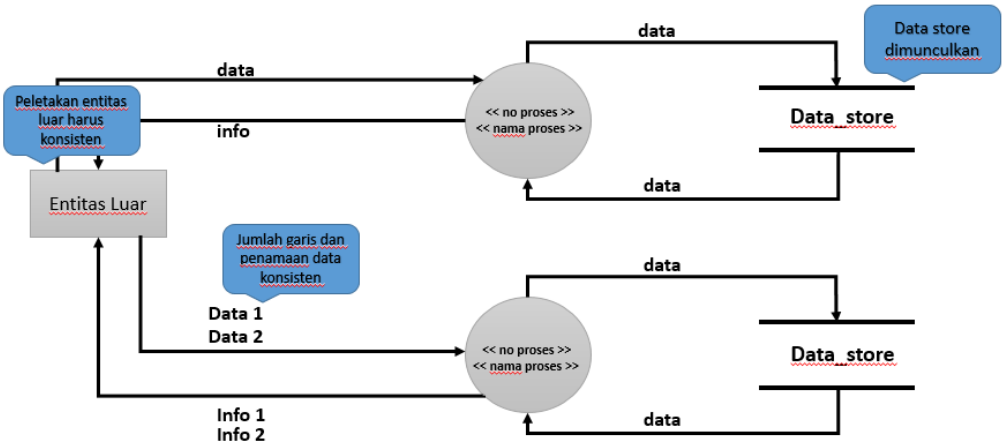
Diagram konteks adalah diagram yang menggambarkan sebuah sistem secara menyeluruh yang selanjutnya akan lebih didetailkan oleh DFD. Konteks diagram juga biasa disebut sebagai DFD level . Contoh dari diagram konteks dapat dilihat pada gambar 2.7 Contoh diagram komteks.



Gambar 2.7 Contoh diagram komteks

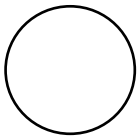
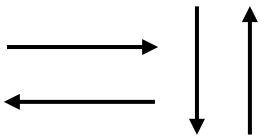
4. Data Flow Diagram (DFD)

Data Flow Diagram merupakan serangkaian diagram yang menggambarkan kegiatan-kegiatan yang ada dalam satu sistem. Teknik pembangunan DFD dimulai dengan menggambarkan sistem secara global dan dilanjutkan dengan analisis masing-masing bagian. Pada awalnya digambarkan konteks diagram yang menggambarkan sebuah sistem secara menyeluruh yang akan diinvestigasi. Konteks diagram tersebut dapat dikatakan sebagai DFD level 0. Analisis sistem yang lebih detail selanjutnya dapat dilakukan dengan menggambarkan DFD level 1,2 dan seterusnya. Contoh penggambaran pembuatan DFD dapat dilihat pada gambar 2.8 Contoh DFD. Simbol-simbol yang digunakan di diagram kc
DFD dapat dilihat pada tabel 2.2 Simbol diagram konteks dan DFD.



Gambar 2.8 Contoh DFD

Tabel 2.2 Simbol diagram konteks dan DFD

Simbol	Nama Simbol	Fungsi
	Entitas Luar	Menggambarkan entitas eksternal yang berhubungan dengan sistem
	Sistem (konteks)/ Proses (DFD)	Menggambarkan proses yang ada pada suatu sistem
	Aliran data/informasi	Menggambarkan aliran data antar proses, data store, dan entitas luar
	Data Store	Menggambarkan kumpulan dokumen-dokumen sejenis yang disimpan

5. Spesifikasi Proses

Spesifikasi Proses merupakan tabel yang berisi keterangan deskripsi dari semua proses yang terdapat di DFD. Logika proses harus dituliskan secara jelas baik menggunakan bahasa deskriptif atau *pseudo code* (tidak boleh campuran). Contoh dari spesifikasi proses dapat dilihat pada Tabel 2.3 Contoh Spesifikasi Proses.

Tabel 2.3 Contoh Spesifikasi Proses

No	Proses	Keterangan
1.	No Proses	Menyatakn no proses
	Nama Proses	Menyatakan nama proses
	Source (Sumber)	(Menyatakan sumber data input menuju proses)
	Input	(menyatakan isi data yang masuk ke proses)
	Output	(menyatakan informasi yang keluar dari proses)
	Destination	(menatakan tujuan informasi output dari proses)

	Logika Proses	(menyatakan algoritma dari proses)

6. Kamus Data (*Data Dictionary*)

Kamus Data merupakan fakta tentang data dan kebutuhan-kebutuhan informasi dari sistem informasi. Dengan menggunakan *data dictionary*, analis sistem dapat mendefinisikan data yang mengalir dalam sistem dengan lengkap. Contoh dari kamus data dapat dilihat pada Tabel 2.4 Contoh Kamus Data.

Tabel 2.4 Contoh Kamus Data

Nama aliran data	Data Nama dari data yang digunakan
Where used/How used	Daftar pada poin proses apa data digunakan
Keterangan	Uraian singkat dari data yang digunakan
Struktur Data	Daftar komponen data yang ada pada data yang digunakan
Deskripsi	Jenis data dalam representasi komputer untuk masing-masing data.

2.13 Object Oriented Analysis and Design

Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan satu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Tujuan dari analisis berorientasi objek adalah untuk mengembangkan model yang menggambarkan perangkat lunak komputer karena bekerja untuk memenuhi seperangkat persyaratan yang ditentukan user. *Tools* yang dapat digunakan pada pendekatan analisis pengembangan sistem secara objek dapat menggunakan UML.

2.14 Unified Modeling Language (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[13]. 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.

2.14.1 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[13].

2.14.2 Diagram UML

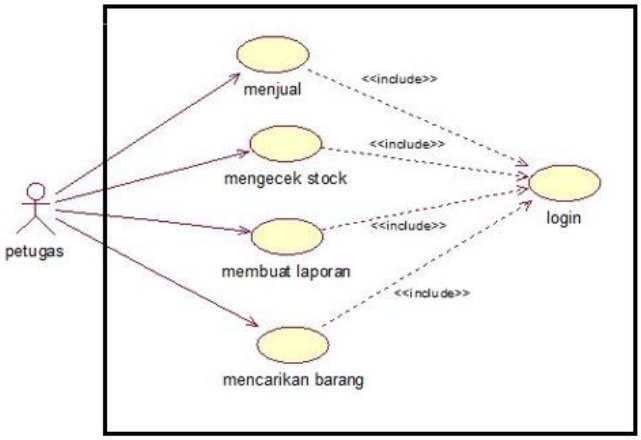
UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek[13], yaitu:

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 Cases 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.

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 2.9 di bawah ini.

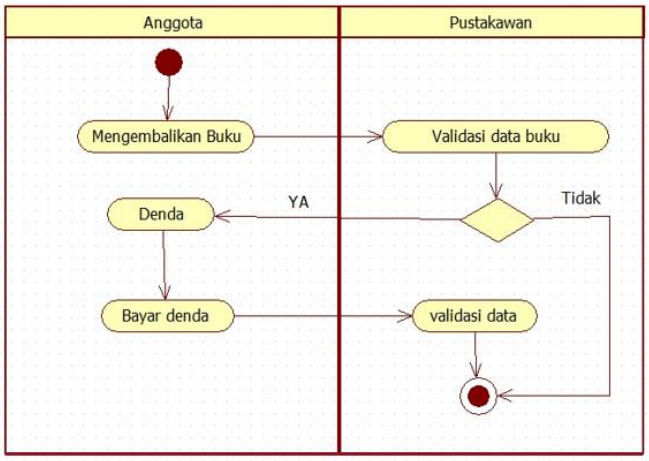


Gambar 2.9 Usecase Diagram

2. Activity Diagram

Activity diagrams 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

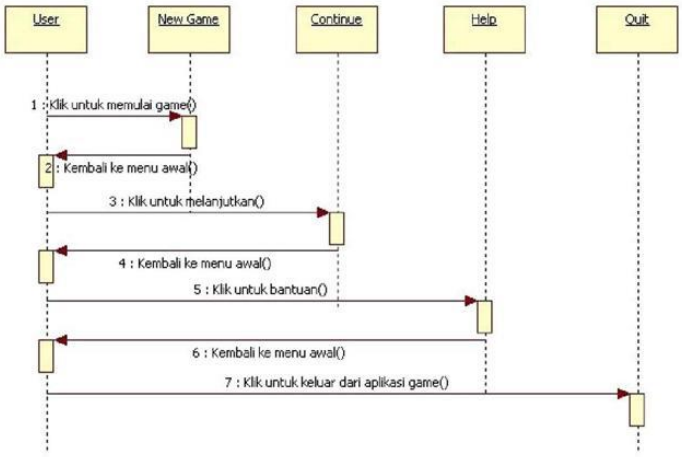
titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Contoh Activity diagram dapat di lihat pada gambar 2.10 berikut ini.



Gambar 2.10 Activity Diagram

3. Sequence Diagram

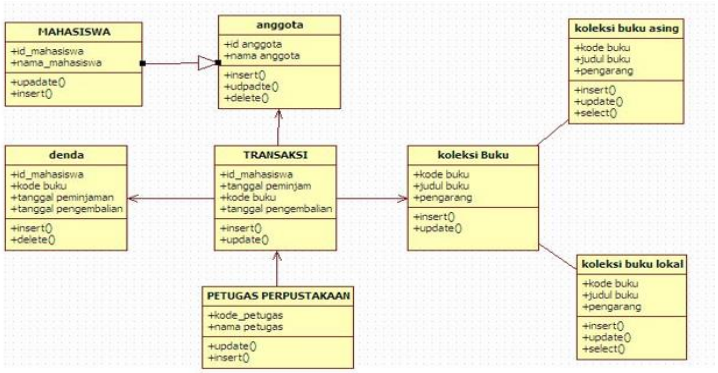
Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case: interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Contoh sequence diagram dapat di lihat pada gambar 2.11 berikut ini.



Gambar 2.11 Sequence

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 2.12 dibawah ini.



Gambar 2.12 Class Diagram

2.15 Pengujian Kuesioner

Kuesioner adalah sebuah daftar pernyataan yang harus diisi oleh orang yang akan dievaluasi (responden). Metode yang digunakan dalam kuesioner pada penelitian ini adalah skala Likert. Dalam skala likert, responden diminta untuk membaca dengan seksama setiap pernyataan yang disajikan, kemudian ia diminta untuk menilai pernyataan-pernyataan tersebut.

Derajat penilaian responden terhadap suatu pernyataan terbagi dalam 5 kategori yang tersusun secara bertingkat, mulai dari Sangat Tidak Setuju (ST), Tidak Setuju (T), Ragu-ragu (N), Setuju (S), dan Sangat Setuju (SS). Atau dapat pula sebaliknya. Pernyataan tiap kuesioner dibuat berdasarkan aspek-aspek yang diteliti. Bobot pemberian skor yang digunakan dapat dilihat pada Tabel 2.5.

Tabel 2.5 Panduan pemberian skor[14]

Jenis Pernyataan	Bobot pendapat				
	SS	S	R	TS	STS

Positif	5	4	3	2	1
Negatif	1	2	3	4	5

Skor yang telah dihitung pada setiap pernyataan kemudian dikalikan dengan masing-masing bobot tersebut sesuai dengan skenario kuesioner yang telah dibuat. Setelah itu total kan seluruh bobot jawaban tersebut kemudian bagi dengan total responden yang nantinya menjadi nilai rata-rata. Nilai rata-rata inilah yang diambil sebagai acuan sikap dimana jika nilai rata-rata kurang dari 3, maka dapat diartikan responden bersikap negatif dan jika nilai rata-rata lebih dari sama dengan 3, maka dapat diartikan responden bersikap positif terhadap tujuan yang ingin peneliti capai. Untuk lebih jelasnya dapat dilihat pada rumus dibawah ini.

$$\bar{X} = \frac{\sum Total}{\mathbf{n}} \text{ .....I}$$

Dimana,

$\bar{X} \geq 3$ *bersikap positif*

$\bar{X} < 3$ *bersikap negatif*

Keterangan :

$\bar{X}$ = rata-rata

$\sum Total$ = *jumlah seluruh nilai setelah dikalikan dengan bobot*

n = total responden