BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Tahapan tinjauan perusahaan ini merupakan peninjauan terhadap tempat penelitian studi kasus yang dilakukan di PT. Dealpro Indonesia. Adapun tinjauan perusahaan yang dimaksud meliputi profil perusahaan, visi dan misi perusahaan, serta struktur organisasinya.

2.1.1 Profil Perusahaan

PT. Dealpro Indonesia merupakan salah satu perusahaan yang berfokus pada bidang usaha *event, design*, dan *print* yang sudah beroperasi dari 30 Januari 2015 yang diprakarsai oleh Yosep Sobandi sebagai pemilik. Perusahaan ini termasuk perusahaan dengan fasilitas modern sehingga dapat berkompetitif dalam bidang *event, design, print* dan hingga kini perusahaan ini telah memiliki 2 cabang di kota Bandung yang pertama, *head office* yang bertempat di Jalan Ciumbuleuit No. 180 dan yang kedua *branch office* yang bertempat di jalan Tubagus Ismail No. 40.

PT. Dealpro Indonesia dengan legal perusahaannya telah sah menjadi perusahaan berbadan hukum, perusahaan ini memiliki kelengkapan legalitas seperti Tanda Daftar Usaha Pariwisata (TDUP), Surat Izin Usaha Perdagangan (SIUP), Nomor Tanda Daftar Perusahaan (TDP) dan juga legalitas lainnya. Atas legalitas tersebut PT. Dealpro Indonesia dipercaya untuk memegang *event* dari perusahaan-perusahaan besar di Indonesia, seperti Honda, Summarecon Bandung, Agung Podomoro Land, Trans Studio Bandung, Cihampelas Walk, Citilink dan perusahaan besar lainnya.

2.1.2 Visi dan Misi Perusahaan

Perusahaan PT. Dealpro Indonesia mempunyai visi jangka panjang serta misi utama yang kuat dalam menentukan arah gerak dan peta jalannya ke depan, visi dan misi tersebut yaitu:

Visi :

Menjadi perusahaan *event*, *design*, *print* dan produksi terdepan dan terkemuka di Indonesia melalui pelayanan yang kreatif.

Misi :

Mengoptimalkan pelayanan dibidang *event*, *design. print*, dan produksi dengan menerapkan strategi promosi berbeda dari pesaing lainnya demi mencapai target dan menambah konsumen, serta menjaga hubungan baik antar konsumen.

2.1.3 Logo Perusahaan

Logo dari perusahaan PT. Dealpro Indonesia adalah sebagai berikut :

Gambar 2.1 Logo PT. Dealpro Indonesia

Makna dari bentuk logo :

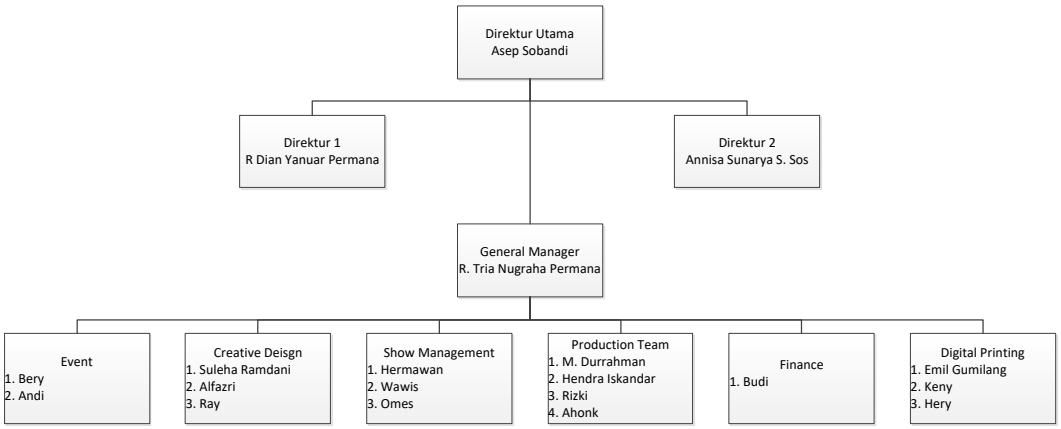
Logo dari PT. Dealpro Indonesia mempresentasikan singkatan D dan P yang diambil dari Dealpro itu sendiri.

Makna dari warna logo :

1. Warna merah pada logo melambangkan keberanian.
2. Warna abu pada logo melambangkan suatu hal yang kuat dan praktikal.
3. Warna putih melambangkan kebebasan dalam kreativitas mengelola perusahaan.
4. Warna hitam pada logo melambangkan keagungan dan kemakmuran serta percaya diri yang kuat.

2.1.4 Struktur Organisasi

Struktur organisasi yang ada di PT. Dealpro Indonesia dapat digambarkan sebagai berikut :



Gambar 2.2 Struktur Organisasi PT. Dealpro Indonesia

2.2 Kajian Teori

Sebelum pembangunan aplikasi tentu diperlukan pengetahuan-pengetahuan yang terkait dengan penelitian yang diambil, diantaranya yaitu:

1. Publikasi
2. *Event*
3. Android
4. *Global Positioning System* (GPS)
5. *Location Based Services* (LBS)
6. *Application Programming Interface* (API)
7. *Google Maps*
8. *Mapbox*
9. *Geocoding*
10. *Payment Gateway*
11. *Mobile Ticketing*
12. *QR Code*
13. PHP
14. *CodeIgniter*
15. *Unified Modeling Language* (UML)

2.2.1 Publikasi

Proses publikasi adalah proses pelaksanaan dari kegiatan penyebaran informasi. Publikasi memiliki arti pengumuman, penyiaran, atau penerbitan yang dilakukan dengan tujuan mengumumkan, menyiarkan, dan menerbitkan informasi tentang produk, lembaga/organisasi, aktivitas dan sebagainya kepada publik, hal ini sesuai yang dijabarkan oleh Ruslan [11], ia mengatakan bahwa publikasi merupakan salah satu relasi komponen yang cukup berperan penting untuk menunjang keberhasilan dalam promosi suatu kegiatan.

Sebagaimana publikasi berperan sebagai sistem, tahapan dari publikasi dimulai dari pengolahan informasi hingga pada *output* dimana *output* itu bisa sampai kepada publik. Publikasi merupakan suatu kegiatan yang krusial dan harus bisa dipertanggungjawabkan, sebab proses publikasi ini berhubungan dengan informasi yang akan diketahui oleh publik, hal ini juga mengharuskan informasi yang diolah dan disampaikan harus benar-benar berupa fakta.

Proses publikasi juga dapat memberikan dampak positif pada perusahaan atau instansi yang terkait, semakin bertanggungjawabnya pembuat informasi maka semakin besar juga kepercayaan yang akan didapat dari masyarakat. Sehingga perusahaan itu layak untuk disebut perusahaan *good well*. Publikasi dapat dilakukan dengan berbagai cara mulai dari media cetak seperti koran, majalah, *flyer*, spanduk, dan baligho, media elektronik seperti televisi dan radio, hingga media *online* seperti *website*, portal berita, dan media sosial.

2.2.2 *Event*

Menurut John E. Kennedy [12] *event* dapat diartikan sebagai pameran, pertunjukan, atau festival dengan syarat adanya penyelenggara, peserta, dan pengunjung. *Event* juga dapat diartikan sebagai kegiatan yang dilakukan oleh sekelompok orang atau organisasi yang tujuannya adalah untuk mengumpulkan orang-orang ke suatu tempat agar mereka dapat memperoleh informasi yang ingin disampaikan oleh penyelenggara. Setiap *event* selalu mempunyai tujuan utama untuk apa diselenggarakan, salah satu tujuan utamanya adalah target sasarannya atau target pengunjung yang diharapkan hadir dalam *event* yang diadakan.

Tujuan diadakannya *event* adalah sebagai berikut :

a. Merubah pemikiran khalayak banyak.
b. Memperkenalkan sebuah produk untuk merubah gaya hidup seseorang.
c. Menjangkau masyarakat yang lebih luas.
d. Meningkatkan *awareness* mengenai suatu *issue*.

Dalam menyelenggarakan sebuah *event* pasti tidak akan lepas dari sebuah perencanaan, karena tanpa sebuah perencanaan maka *event* tersebut akan menjadi berantakan. Adapun tahapan perencanaan sebuah *event* adalah sebagai berikut :

1. Konsep *event*, dalam proses mengkonsep sebuah *event* yang harus ditentukan adalah tujuan dari *event* tersebut. *Event* pada pelaksanaannya harus memberikan kesan dan pengalaman kepada mereka yang terlibat.
2. Target *event*, setelah selesai memantapkan konsep maka tahapan selanjutnya adalah menentukan target *event* mulai dari target pengunjung hingga target sponsor.
3. Perencanaan *event*, dalam hal ini yang harus ditentukan adalah strategi untuk mencapai target dan tujuan *event*, menentukan tempat *event*, menentukan kebutuhan *event*, membuat *timeline* kegiatan, hingga perencanaan biaya.
4. Produksi, yaitu proses menyiapkan segala alat dan bahan yang dibutuhkan untuk berlangsungnya *event*, sehingga alat dan bahan tersebut tersedia dan siap instalasi sebelum *event* dilaksanakan.
5. Promosi, pada hakikatnya kegiatan promosi bertujuan untuk menaikkan *awereness* dan harapannya dapat membuat sebuah keputusan (datang, membeli tiket, dan menjadi berlangganan) promosi dapat dilakukan dengan media cetak, elektronik, dan internet.
6. Persiapan dan instalasi tempat, merupakan proses menginstal semua kebutuhan *event* mulai dari dekorasi, *gimmick, stage, stand*.
7. Registrasi, proses dimana pendaftaran pengunjung/peserta untuk mengikuti sebuah *event*.
8. *Ticketing*, merupakan salah satu tanda bukti masuk untuk mengikuti sebuah *event*.

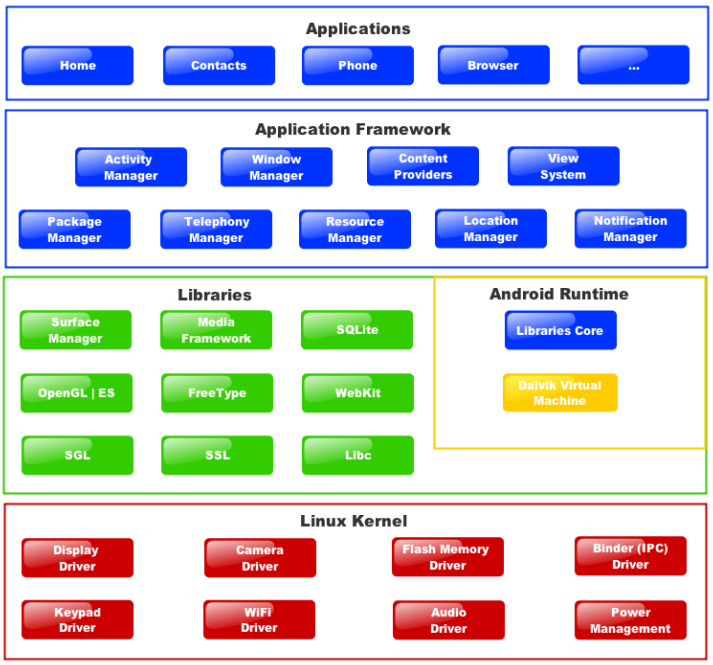
2.2.3 Android

Android merupakan sebuah sistem operasi berbasis *linux* yang dirancang untuk perangkat telepon seluler layar sentuh seperti *smartphone* dan tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri [4]. 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.

Sistem Android menggunakan *database* untuk menyimpan informasi penting yang diperlukan agar tetap tersimpan meskipun *device* dimatikan. Sistem Android menggunakan SQLite untuk melakukan penyimpanan pada *database*, dimana SQLite tersebut merupakan suatu *open source database* yang banyak digunakan di berbagai perangkat keras berukuran kecil.

2.2.3.1 Arsitektur Android

Android dibangun dengan berbagai macam arsitektur sebagai komponen yang terdapat pada perangkat tersebut adapun arsitektur dari android tersebut dapat ditunjukan seperti Gambar 2.3.



Sumber gambar : httpwww.eazytutz.comandroidandroid-architecture

Gambar 2.3 Arsitektur Android

Berikut ini adalah penjelasan dari arsitekur android :

1. *Application* ini adalah *layer* dimana kita terhubung dengan aplikasi saja, dimana biasanya kita mengunduh aplikasi kemudian kita melakukan instalasi dan menjalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien, *email*, program SS, kalender, peta, *browser*, dan kontak. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *Java*.
2. *Appllication frameworks*, pengembang memiliki akses penuh ke APIs *frameworks* yang sama digunakan oleh aplikasi inti sehingga dapat memberikan kesempatan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *services backgrounds*, mengatur alarm, dan menambahkan suatu *notifications*, dan lain sebagainya.
3. *Libraries*, merupakan *layer* dimana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas kernal, layer ini meliputi berbagai C/C++ inti seperti Libc dan SSL.
4. Android *Run Time*, *layer* yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi *Linux*. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi *Core Libraries* dan *Dalvik Virtual Machine*.
5. *Linux Kernel*, merupakan *layer* dimana inti *operating system* dari android itu berada. Berisi *file-file system* yang mengatur *system processing*, *memory*, *resource*, *drives*, dan sistem-sistem operasi android. *Kernel* juga bertindak sebagai lapisan abstraksi antara *hardware* dan *software stack.*

2.2.3.2 Komponen Aplikasi Android

Aplikasi android ditulis dalam bahasa pemrograman *Java*. Kode *java* di-*compile* bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, prosesnya di-*package* oleh *tools* yang dinamakan "*apt tools*" kedalam paket android sehingga menghasilkan file dengan ektensi .apk (dot apk). *File* apk ini

yang disebut dengan aplikasi, dan kemudian dapat di instal di perangkat android. Komponen aplikasi android diantaranya:

1. ***Activities***

Suatu *activity* menyediakan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android mungkin hanya terdapat satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi itu sendiri, satu *activity* biasanya dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan oleh pengguna. Untuk pindah dari satu *activity* ke *activity* lainnya, dapat dilakukan dengan *even*, misalnya klik tombol, memilih pilihan atau menggunakan *trigger* tertentu.

2. ***Service***

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*, sebagai contoh dalam menjalankan musik, *service* mungkin menjalankan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Misalnya media *player* sedang memutar lagu dari *list* yang ada, aplikasi ini memiliki dua atau lebih *activity* yang memungkinkan pengguna untuk memilih lagu, atau menulis pesan/*chat* dengan media *player* tetap berjalan. Untuk menjaga musik tetap dijalankan, *activity player* dapat menjalankan *service*. *Service* dijalankan pada *thread* utama dari proses aplikasi.

3. ***Broadcast Reciever***

Broadcast reciever berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Sebagai contoh *broadcast* seperti notifikasi zona waktu berubah, baterai lemah, gambar telah selesai diambil oleh kamera, atau pengubahan referensi bahasa yang digunakan.

Broadcast receiver tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang diterima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau getaran perangkat.

4. *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file system* seperti *database* SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika menggunakan aplikasi yang membutuhkan peta (*map*), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka inilah fungsi *Content Provider* untuk menyediakan data yang diperlukan oleh aplikasi.

2.2.4 *Android Development Tools*

Dalam proses merancang dan membangun sebuah aplikasi berbasis android tentu dibutuhkan beberapa *tools* pendukung, diantaranya :

2.2.4.1 *Android Software Development Kit* (SDK)

Android SDK adalah *tools Application Programming Interface* (API) yang dibutuhkan untuk mengembangkan aplikasi android dengan menggunakan bahasa pemrograman *java*. Android SDK menyediakan *library* API dan *developer tools* yang digunakan untuk *build, test*, dan *debug* sebuah aplikasi android.

2.2.4.2 *Android Development Tools* (ADT)

Android Development Tools (ADT) adalah *plugin* untuk *eclipse* yang menyediakan seperangkat alat yang terintegrasi dengan *eclipse* IDE (*Integrated Development Environment*). ADT memberikan akses ke banyak fitur sehingga mudah untuk membangun aplikasi android. Berikut ini adalah penjelasan fitur antara *eclipse* dan IDE.

2.2.4.3 *Eclipse Integrated Development Environment* (IDE)

Eclipse merupakan program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. *Eclipse* tersedia secara bebas untuk merancang dan mengembangkan aplikasi android. *Eclipse* merupakan IDE terpopuler dikalangan *developer* android, karena *eclipse* memiliki android

plugin lengkap yang tersedia untuk mengembangkan aplikasi android, selain itu *eclips*e juga mendapat dukungan langsung dari *Google* untuk menjadi IDE pengembangan android, membuat *project* android dimana *source software* langsung dari situs resminya *Google*. Sampai saat ini *eclipse* memiliki 4 versi *package*, yaitu: *Indigo Package, Helios Package, Galileo Package, Ganymade Package*, dan *Europa Package*.

2.2.5 *Global Positioning System* (GPS)

Global Positioning System (GPS), merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunaanya dimana dia berada (secara global) di permukaan bumi yang berbasiskan satelit. Data dikirim dari satelit berupa sinyal rasio dengan data digital.

Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang bernama GPS *reciever* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi yang diubah menjadi titik yang dikenal dengan *Way-point* nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi yang tertera pada layar peta elektronik.

Satelit GPS berkekuatan energi sinar matahari, mempunyai baterai cadangan untuk menjaga agar tetap berjalan pada saat gernaha matahari atau pada saat tidak ada energi matahari. Satelit-satelit GPS selalu berada pada orbit yang tepat untuk menjaga akurasi data yang dikirim ke GPS *reciever*, sehingga harus selalu dipelihara agar posisinya tepat.

2.2.6 *Location Based Services* (LBS)

Menurut Qusay H. Mahmoud (J2ME and Location-Based Services, 2004), *Location Based Service* (LBS) adalah sebuah layanan yang digunakan untuk mengetahui posisi dari pengguna, kemudian menggunakan informasi tersebut untuk menyediakan jasa dan aplikasi yang personal [13].

LBS memberikan kemungkinan komunikasi dan interaksi dua arah. Oleh karena itu pengguna memberi tahu penyedia layanan untuk mendapatkan informasi yang dia butuhkan, dengan referensi posisi pengguna tersebut. Layanan

berbasis lokasi dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu : *Geographic Information System*, *Internet Service*, dan *Mobile Device*.

2.2.6.1 Unsur-Unsur *Location Based Services* (LBS)

Adapaun unsur-unsur utama LBS adalah sebagai berikut :

a. *Location Manager* (API *Maps*): Menyediakan *tools/source* untuk LBS, *Aplication Programming Interface* (API) *maps* menyediakan fasilitas untuk menampilkan, memanipulasi peta beserta *feature* lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya. Paket ini berada pada com.google.android.maps.
b. *Location Providers* (API *location*): Menyediakan teknologi pencarian lokasi yang digunakan oleh *device*/perangkat. API *location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. API *location* berada pada paket android yaitu *android.location*. Dengan *location manager*, kita dapat menentukan lokasi kita saat ini dan rute menuju tempat tertentu.

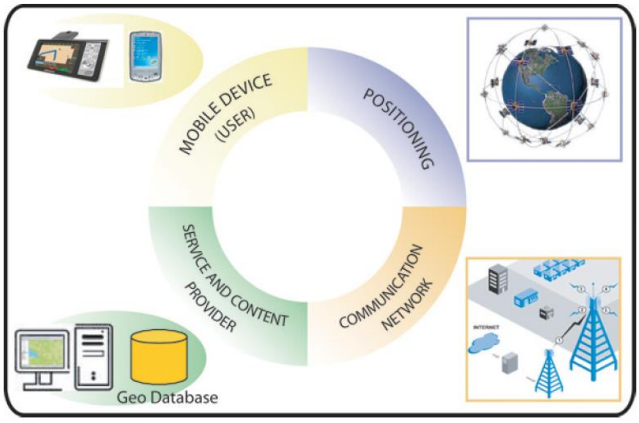
2.2.6.2 Komponen-Komponen *Location Based Services* (LBS)

Penggunaan *Location Based Services* (LBS) juga memerlukan beberapa komponen yang terbagi menjadi 5 elemen, diantaranya [14]:

1. *Mobile Device* yaitu sebuah alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Perangkat yang memungkinkan yaitu perangkat yang memiliki fasilitas navigasi PDA, *mobile phone*, laptop, dan lainnya.
2. *Communication Network* adalah jaringan selular yang mengirimkan data pengguna dan permintaan layanan.
3. *Positioning Component*, biasanya posisi pengguna harus ditentukan untuk pengolahan layanan. Posisi pengguna dapat diperoleh menggunakan jaringan komunikasi atau dengan menggunakan *Global Positioning System* (GPS).

4. *Services and Application Provider*, adalah penyedia layanan pengguna selular yang bertanggung jawab untuk memproses layanan.
5. *Data and Content Provider*, yaitu penyedia layanan informasi data yang dapat diminta oleh pengguna.

Kelima komponen tersebut dapat ditunjukan pada gambar berikut :



Sumber gambar : Jurnal Teknologi dan Sistem Komputer [14]

Gambar 2.4 Komponen LBS

2.2.6.3 Kategori *Location Based Services* (LBS)

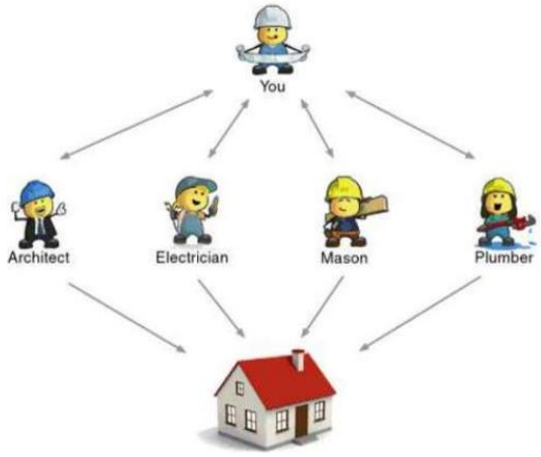
Secara garis besar jenis *Location Based Services* dapat dibagi menjadi dua bagian yaitu:

1. *Pull Services:* Layanan diberikan berdasarkan permintaan dari pelanggan akan kebutuhan suatu informasi.
2. *Push Service*: Layanan ini diberikan langsung oleh *service provider* tanpa menunggu permintaan dari pelanggan, tentu saja informasi yang diberikan tetap berkaitan dengan kebutuhan pelanggan.

2.2.8 *Application Programming Interface* (API)

Application Programming Interface (API) merupakan *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana agar suatu *software* dapat berinteraksi dengan *software* lain. Penjelasan ini dapat dicontohkan dengan analogi apabila akan dibangun suatu rumah. Dengan menyewa kontraktor yang dapat menangani bagian yang

berbeda, pemilik rumah dapat memberikan tugas yang perlu dilakukan oleh kontraktor tanpa harus mengetahui bagaimana cara kontraktor menyelesaikan pekerjaan tersebut. Dari analogi tersebut, rumah merupakan *software* yang akan dibuat, dan kontraktor merupakan API yang mengerjakan bagian tertentu dari *software* tersebut tanpa harus diketahui bagaimana prosedur dalam melakukan pekerjaan tersebut [15].



Sumber gambar : Jurnal Teknologi Terpadu [15]
Gambar 2.5 Analogi API

Dalam API terdapat fungsi-fungsi atau perintah-perintah untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh *programmer*. Keuntungan menggunakan API diantaranya:

1. Probabilitas, API dapat digunakan untuk bahasa pemrograman ataupun untuk sistem operasi mana saja asalkan paket-paket API sudah terpasang.
2. Lebih mudah dimengerti, API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call*. Hal ini sangat penting dalam hal *editing* dan pengembangan.
3. Mudah dikembangkan, dengan adanya API memudahkan *programmer* untuk mengembangkan suatu sistem.

2.2.9 *Google Maps*

Seperti yang tercatat oleh Svennerberg (*Beginning Google Maps API 3*), *Google Maps* adalah yang paling populer di internet. Pencatatan yang dilakukan pada bulan Mei tahun 2010 ini menyatakan bahwa 43% *mashup* (aplikasi dan situs web yang menggabungkan dua atau lebih sumber data) menggunakan *Google Maps*. Beberapa tujuan dari penggunaan *Google Maps* adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya. Hampir semua hal yang berhubungan dengan peta dapat memanfaatkan *Google Maps* [16].

Dalam pembangunan aplikasi dengan *Google Maps API* dapat menggunakan urutan sebagai berikut:

1. Memasukkan *Maps API JavaScrip*t ke dalam HTML kita.
2. Membuat *element div* dengan nama *map_canvas* untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan properti-properti pada peta.
4. Menuliskan fungsi *JavaScript* untuk membuat objek peta.
5. Meng-inisiasi peta dalam *tag body* HTML dengan *event onload*.

Sumber gambar : https://www.google.com/maps
Gambar 2.6 *Google Maps*

2.2.10 *Mapbox*

Mapbox adalah sebuah *platform* pemetaan *open source* yang bekerja dan merilis sebagai kode sebanyak mungkin. Sebagian besar data *Mapbox* menggunakan bantuan serta berinvestasi pada berbagai macam sumber data misalnya *OpenStreetMap*, USGS, *Landsat*, dan *OpenAddresses* [17]. *Mapbox*

mendukung berbagai macam aplikasi yang akan digunakan oleh penggunanya, baik *mobile* maupun *online*. Produk yang tersedia di *Mapbox* terdiri dari peta, satelit, server atlas, *geocoding*.

Mapbox mendukung beberapa aplikasi pengembang, diantarannya *Javascript*, iOS, Android dan API. *Mapbox* telah digunakan untuk aplikasi *Foursquare*, *Pinterest* dan *Evernote* yang memudahkan penggunanya untuk menandai lokasi mereka kapanpun dan dimanapun. Pengguna *Mapbox* yang akan mendaftar disediakan berbagai pilihan akses data dengan berbagai pilihan biaya yang tentu saja mempengaruhi keberagaman fasilitas yang didapat oleh pengguna.

Sumber gambar: https://blog.mapbox.com
Gambar 2.7 *Mapbox*

2.2.11 *Geocoding*

Geocoding adalah salah satu teknologi yang merupakan proses untuk mencari koordiat geografis (*latitude* dan *longitude* atau garis lintang dan bujur) dari alamat tertentu (nama jalan, nama kota, kode pos, nama negara) kebalikan dari *geocoding* adalah *reverse geocoding*, yaitu mencari data geografis berdasarkan koordinat geografis. Jika ada suatu koordinat yang dimasukan lalu akan muncul data geografis yang ada pada koordinat itu seperti nama jalan, kode pos, negara, kota, provinsi.

Geocoding diartikan sebagai proses penyimpanan identifikasi lokasi menjadi bagian dari *record*. Jika data telah di *geocoding* berarti pengidentifikasian telah ditambahkan pada *record* ke lokasi pada peta. *Geocoding* sebenarnya adalah proses penggabungan dua tabel dimana kedua tabel tersebut dapat dilakukan penggabungan karena memiliki *field* yang sama.

2.2.12 *Payment Gateway*

Menurut Gulati & Srivastava [18] *payment gateway* merupakan komponen infrasturktur penting untuk memastikan transaksi berlangsung tanpa hambatan dan terlindungi total melalui jaringan internet. *Payment gateway* adalah sebuah akses poin ke dalam jaringan perbankan nasional. Semua transaksi secara *online* harus melalui *payment gateway* untuk diproses. Secara teorinya *payment gateway* yaitu sebuah layanan yang bertindak sebagai jembatan antara *merchant* dan institusi keuangan yang melakukan proses transaksi.

Dengan tersedianya layanan tersebut maka *merchant* dapat menyediakan *online payment* pada *website online*-nya dengan cara menghubungkan *website* mereka pada *payment gateway service* menggunakan *service* dari *Application Program Interface* (API). Keunggulan dengan adanya *payment gateway service* diantaranya:

1. Kenyamanan transaksi selama 24x7x365
2. Penggunaan *credit/debit* secara langsung
3. Memproses transaksi secara cepat dan *real-time*
4. Memungkinkan untuk melakukan berbagai jenis pembayaran
5. Informasi data transaksi diproses secara aman
6. Fleksibel dan memungkinkan untuk memproses *report* dan *history* transaksi secara langsung
7. Pembayaran dengan menggunakan *multi-currency*
8. Mempermudah *merchant* untuk mengatasi masalah pembayaran
9. Menggunakan *certifiying authority* dengan *secure server*
10. Filterasi awal sebelum mengirim informasi pembayaran kepada pihak bank (mempermudah pihak bank)
11. Lengkap dan memiliki kontrol administrasi yang sederhana

2.2.13 *Payment Gateway* Midtrans

Untuk pembayaran tiket *event* pada aplikasi akan menggunakan *payment gateway* Midtrans. Midtrans merupakan salah satu *online payment gateway* di

Indonesia yang diluncurkan pada tahun 2012 bertujuan untuk memfasilitasi bisnis *online* di Indonesia dengan pembayaran yang terpercaya mudah dan aman.

Midtrans memberikan biaya yang murah per transaksi kartu kredit, yaitu *charge* sebesar 4% + Rp.2.200 kepada *merchant* yang sudah termasuk biaya pajak. Sebagai *payment gateway*, Midtrans menyediakan infrastruktur pembayaran yang aman, terpercaya dan bebas penipuan, serta sudah bekerja sama dengan lembaga finansial, bank ternama, dan alat pembayaran di Indonesia.

Sumber gambar: https://midtrans.com
Gambar 2.8 Midtrans

2.2.14 *Mobile Ticketing*

Perkembangan teknologi informasi memberikan banyak terobosan-terobosan baru bagi dunia bisnis. Gambaran yang nyata adalah terobosan baru dalam pelayanan pemesanan tiket yang semula konvensional kini dapat dilakukan dengan *mobile ticketing*.

Mobile ticketing adalah proses dimana konsumen dapat melakukan pemesanan, melakukan pembayaran, memperoleh dan melakukan validasi tiket dari manapun dan kapanpun menggunakan perangkat *mobile*. *Mobile ticketing* merupakan salah satu bentuk layanan *mobile content* dalam dunia bisnis yang merupakan layanan untuk pemesanan tiket secara *online* menggunakan piranti *mobile*. Aplikasi *mobile ticketing* berguna untuk mempermudah pemesanan tiket serta mempercepat pembayaran tiket.

Pada saat sekarang ini mulai digunakannya aplikasi *content* pada dunia bisnis, yang menuntut akan penggunaan teknologi penyedia layanan *content* yang aman dan *reliable*. Teknologi yang aman akan menciptakan peluang-peluang baru dalam dunia bisnis. Dengan *mobile ticketing* ini dapat melakukan pembayaran secara *online*, kemudian sebagai bukti telah melakukan pembayaran akan mendapatkan kode tiket secara *online*. Hal ini sangat membantu pengguna dalam

melakukan pemesanan tiket dan melakukan pembayaran karena layanan ini sangat praktis dan dapat menghemat waktu.

2.2.14.1 Manfaat *Mobile Ticketing*

Mobile ticketing dapat menjadi kemudahan bagi orang-orang yang menginginkan tiket untuk mengikuti sebuah acara atau mendapatkan pelayanan dan jasa tanpa harus beranjak dari rumah atau kantor untuk mendatangi tempat penjualan tiket. Dengan *mobile ticketing*, berbagai kemudahan didapatkan bagi konsumen, dan mendatangkan keuntungan bagi produsen. *Mobile ticketing* mempunyai beberapa kelebihan, diantaranya :

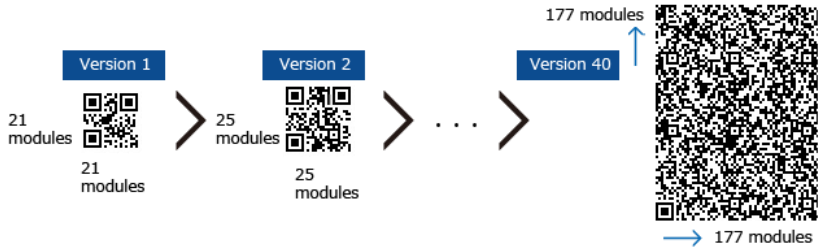
1. Mengurangi biaya yang berkaitan dengan percetakan dan surat tiket.
2. Mengurangi tenaga kerja yang terkait dengan pencetakan dan surat tiket.
3. Keamanan terjamin, karena *QR code* validasi dan menghilangkan kemungkinan tiket palsu atau duplikat.
4. Memberikan informasi tambahan yang perlu diketahui bagi pelanggan.

2.2.15 *Quick Response* (QR) *Code*

QR code merupakan teknik yang mengubah data tertulis menjadi kode-kode 2-dimensi yang tercetak kedalam suatu media yang ringkas. QR adalah singkatan dari *Quick Response* karena ditujukan untuk diterjemahkan isinya dengan cepat. *QR code* merupakan pengembangan dari barcode satu dimensi, *QR code* salah satu tipe dari *barcode* yang dapat dibaca menggunakan kamera *handphone. QR code* mampu menyimpan semua jenis data, seperti data angka/numerik, *alphanumeric,* biner, kanji/kana. Selain itu *QR code* memiliki tampilan yang lebih kecil dari pada *barcode*. Hal ini dikarenakan *QR code* mampu menampung data secara horizontal dan vertikal, jadi secara otomatis ukuran dari tampilannya, gambar *QR code* bisa hanya sepersepuluh dari ukuran sebuah barcode [19].

Versi simbol *QR code* berkisar dari versi 1 ke versi 40. Setiap versi memiliki konfigurasi modul yang berbeda atau jumlah modul (Modul ini mengacu pada titik-titik hitam dan putih yang membentuk *QR code*). Konfigurasi modul

mengacu pada jumlah modul yang terkandung dalam simbol, dimulai dengan versi 1 (21 x 21 modul) sampai ke versi 40 (177 x 177 modul). Setiap nomor versi lebih tinggi terdiri dari 4 modul tambahan per samping. Setiap versi simbol *QR code* memiliki kapasitas data yang sesuai dengan jumlah data, jenis karakter dan tingkat kesalahan koreksi. Untuk itu pemeriksaan data dengan kapasitas maksimum ditentukan pada setiap versinya. Untuk versi dan kapasitas data maksimum, maka jumlah data dan modul akan meningkat sehingga simbol *QR code* semakin besar [19]. Penggunaan *QR code* dalam penelitian ini diharapkan dapat mempermudah petugas maupun peserta dalam melakukan proses verifikasi dari tiket *event* yang diikuti.



Sumber gambar: https://www.qrcode.com
Gambar 2.9 Versi Simbol *QR Code*

2.2.16 PHP

PHP merupakan suatu bahasa pemrograman sisi server yang dapat kita gunakan untuk membuat halaman Web dinamis. Contoh bahasa yang lain adalah *Microsoft Active Server Page* (ASP) dan *Java Server Page* (JSP). Dalam suatu halaman HTML kita dapat menanamkan kode PHP yang akan dieksekusi setiap kali halaman tersebut dikunjungi [20].

PHP merupakan produk *open source* sehingga kita dapat mengakses *source code*, menggunakan, dan mengubahnya. Karena kekayaan akan fitur yang mempermudah perancangan dan pemrograman Web, PHP memiliki tingkat popularitas yang cukup tinggi dikalangan *developer*.

```
<?php
 $a = 1;
 $b = "Ini Adalah Syntax Php";
 $c = 27,0895;

      Echo $b;
?>
```

2.2.17 *CodeIgniter*

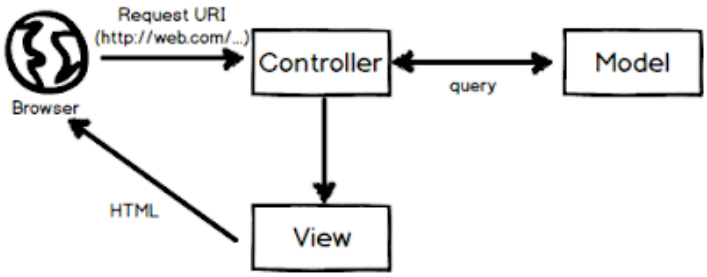
CodeIgniter adalah sebuah *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan bahasa pemrograman PHP, *CodeIgniter* menawarkan kemudahan serta standarisasi dalam proses pengembangan *website* dan aplikasi berbasis web. Dengan *CodeIgniter* proses pengembangan *website* menjadi lebih cepat dan terstandar. *CodeIgniter* juga telah menyediakan *library* dan *helper* yang berguna dan mempermudah *development* [21].

Ada beberapa kelebihan *CodeIgniter* dibandingkan dengan *framework* PHP lain, diantaranya:

1. Performa sangat cepat: salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scracth*, tapi *Codeigniter* sangat cepat bahkan mungkin bisa dibilang *Codeigniter* merupakan *framework* yang paling cepat dibanding *framework* yang populer di Indonesia seperti *Yii, Cake, Symfony* dan *Laravel*.
2. Konfigurasi yang sangat minim: untuk menyesuaikan dengan *database* dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa *file configurasi* seperti *database.php* atau *autoload.php*, dengan *CodeIgniter* kita hanya perlu mengubah sedikit *file* pada *folder config.*
3. Banyaknya komunitas: dengan banyaknya komunitas *CodeIgniter* maka dapat memudahkan kita untuk saling berinteraksi satu sama lain.

4. Dokumentasi yang sangat lengkap: instalasi *CodeIgniter* sudah disertai *guide* yang bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

MVC merupakan sebuah konsep arsitektural yang membagi suatu pengembangan *website* menjadi 3 bagian komponen logika yakti *Model*, *View*, dan *Controller*, jadi proses dasarnya model menghantarkan data dari *database* dapat dikelola oleh *Controller* kedalam *View*, sehingga *View* dapat menampilkan data yang sudah akan di proses.

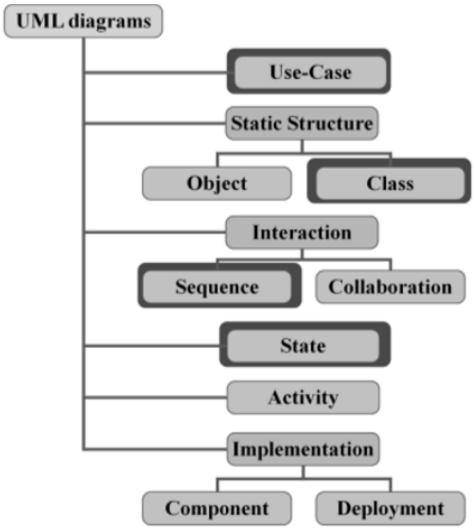


Gambar 2.10 Konsep MVC *CodeIgniter*

2.2.18 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk menspesifikasikan, menvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak (Rumbaugh, Jacobson, & Booch, 2005) [22].

Pemodelan memberikan gambaran yang jelas mengenai sistem yang akan dibangun baik dari sisi struktural ataupun fungsional. UML dapat diterapkan pada semua model pengembangan, tingkatan siklus sistem, dan berbagai macam *domain* aplikasi. Dalam UML terdapat konsep semantik, notasi, dan panduan masing-masing diagram. UML bertujuan menyatukan teknik-teknik pemodelan berorientasi objek-objek menjadi terstandarisasi [22]. Diagram UML dapat digambarkan sebagai berikut :

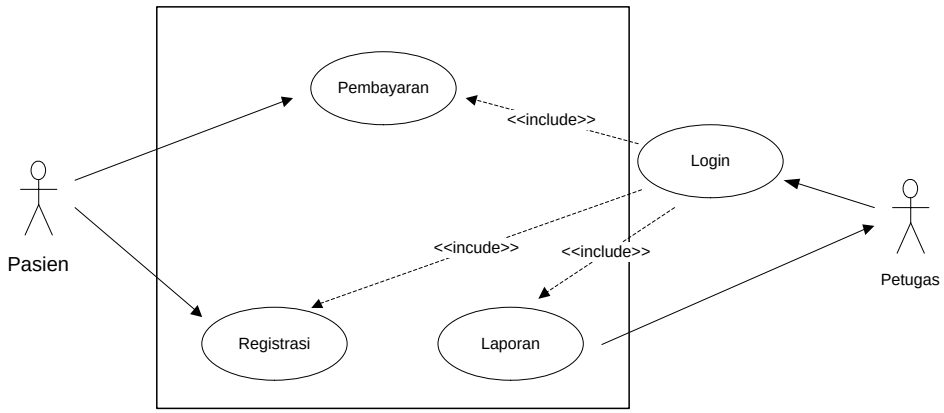


Sumber gambar : JT-IBSI [23]
Gambar 2.11 Diagram UML

Untuk mendapatkan banyak pandangan terhadap sistem yang akan dibangun, UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Ada beberapa diagram yang disediakan dalam UML antara lain :

a. Use case Diagram

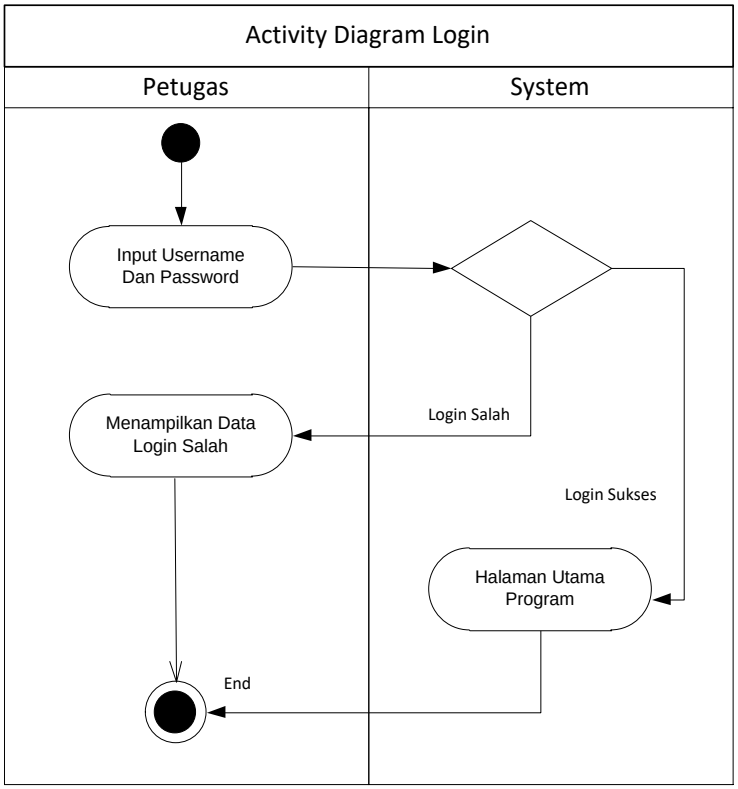
Use case diagram menyajikan interakasi antara *use case* dan aktor. Dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.



Gambar 2.12 Contoh *Use case Diagram*

b. Activity Diagram

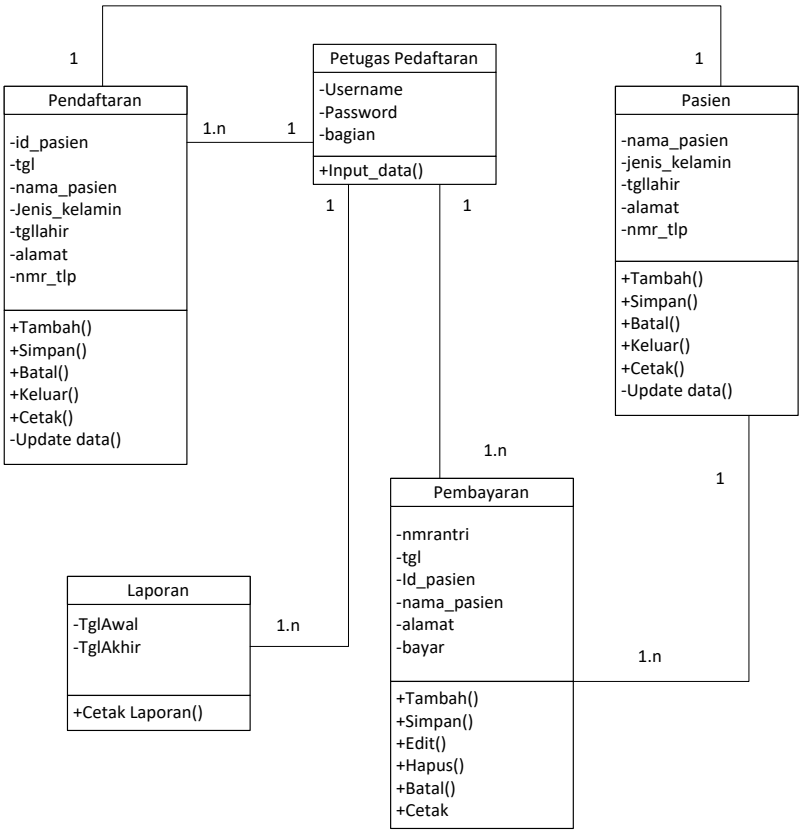
Activity diagram merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. *Activity diagram* juga digunakan untuk mendefinisikan urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan serta rancang menu yang ditampilkan pada perangakat lunak.



Gambar 2.13 Contoh *Activity Diagram*

c. Class Diagram

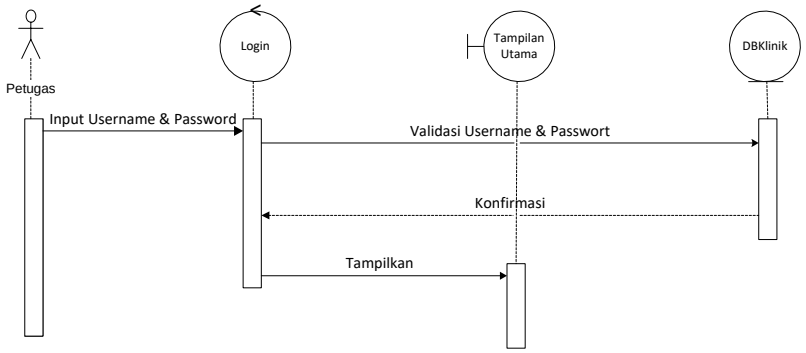
Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* memiliki apa yang disebut atribut dan metode atau operasi. *Class diagram* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.



Gambar 2.14 Contoh *Class Diagram*

d. Sequence Diagram

Sequence diagram merupakan salah satu diagram *interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu dan objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.



Gambar 2.15 Contoh *Sequence Diagram*