

BAB 2

LANDASAN TEORI

2.1 Landasan Teori

Landasan teori bertujuan memberikan gambaran dari teori yang terkait dalam pembangunan aplikasi. Landasan teori yang dibahas yaitu pengertian, Android, GPS, Geotagging, Java, Web Service, PHP, MySQL, Metode yang digunakan dan bahasa pemrograman yang digunakan dalam pembangunan aplikasi.

2.2 Kosep *Event*

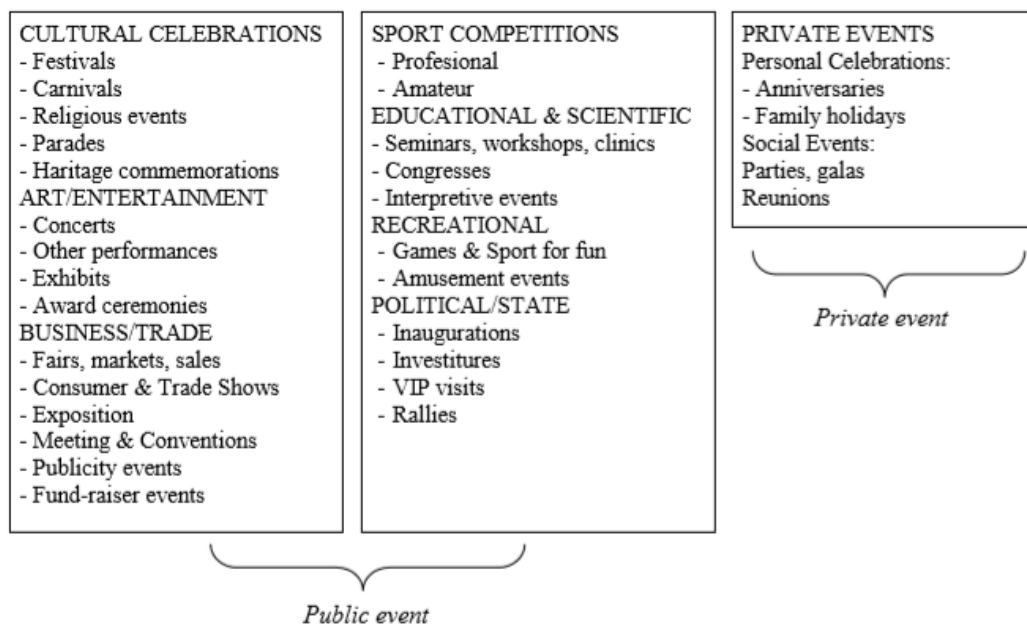
Definisi *Event* adalah sebagai berikut: “*Events are transient, and every event is a unique blending of its duration, setting, management, and people.*” *Event* adalah fana, tidak abadi, dan setiap event merupakan suatu campuran unik dari durasi, pengaturan, pengurus dan orang-orangnya. Dari pengertian di atas dapat dikatakan bahwa *event* merupakan media komunikasi yang bertujuan untuk mengenalkan sebuah brand kepada masyarakat. Namun tidak hanya itu, *event* secara tidak langsung juga menjadi media komunikasi untuk menyampaikan informasi dan pesan kepada publik. Informasi dan pesan yang coba disampaikan melalui event dengan mengandalkan interaksi antara perusahaan dengan khalayak atau konsumen. Dari interaksi tersebut perusahaan atau pemasar juga dapat secara langsung mendapatkan respon dari khalayak atau konsumen yang menjadi target. Selain sebagai media komunikasi dalam menyampaikan pesan, event juga dapat menjadi media publisitas yang efektif dalam jenis *special event* [10].

Sedangkan definisi *special event* adalah sebagai berikut: “*Special events are defined as specific rituals, presentation, performances or celebrations that are conciously planned and created to mark special occasions or to achieve particular social, cultural, or corporate objectives.*” *Special event* adalah suatu ritual istimewa, penunjukan, penampilan, atau perayaan yang pasti direncanakan dan dibuat untuk menandai acara-acara khusus atau untuk mencapai tujuan sosial, budaya atau tujuan bersama-sama [10].

Event dibedakan menjadi public event dan private event, yaitu :

1. *public event* adalah : perayaan budaya, seni atau hiburan, bisnis atau perdagangan, kompetisi olahraga, pendidikan dan ilmu pengetahuan, rekreasi, serta politik atau kenegaraan [10].
2. *private event* adalah : meliputi perayaan pribadi seperti peringatan hari jadi atau *anniversaries*, liburan keluarga, pesta pernikahan, dan pesta ulang tahun, serta *event-event* sosial seperti pesta-pesta, gala, dan acara reuni [10].

Typology of planned event menurut Getz dapat digambarkan sebagai berikut :



Gambar 2.1 *Typologi of planned event*

Sumber : Getz (1997, p. 7)

Sedangkan event dalam bidang ekonomi politik dan pariwisata adalah sebagai penarik minat turis, pembuat image, alat politik, dan sebagai alat untuk menggalang dana. Dalam perayaan umum dapat berupa acara peringatan, pesta, upacara agama atau ritual-ritual, dan event-event santai [10].

2.3 Pengertian dan Prosedur Skala Likert

Skala Likert adalah skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti. Skala

ini merupakan suatu skala psikometrik yang biasa diaplikasikan dalam angket dan paling sering digunakan untuk riset yang berupa survei, termasuk dalam penelitian survei deskriptif [11].

Mengutip dari buku Nazir M. “Metode Penelitian”, Ghalia Indonesia; Bogor; tahun 2005, dalam membuat skala Likert, ada beberapa langkah prosedur yang harus dilakukan peneliti, antara lain [12]:

1. Peneliti mengumpulkan item-item yang cukup banyak, memiliki relevansi dengan masalah yang sedang diteliti, dan terdiri dari item yang cukup jelas disukai dan tidak disukai [12].
2. Kemudian item-item itu dicoba kepada sekelompok responden yang cukup representatif dari populasi yang ingin diteliti [12].
3. Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan lima pilihan skala dengan format seperti [12]:
 - a. Pertanyaan Positif (+)
 - Skor 1. Sangat (tidak setuju/buruk/kurang sekali)
 - Skor 2. Tidak (setuju/baik/) atau kurang
 - Skor 3. Netral / Cukup
 - Skor 4. (Setuju/Baik/suka)
 - Skor 5. Sangat (setuju/Baik/Suka)
 - b. Pertanyaan Negatif (-)
 - Skor 1. Sangat (setuju/Baik/Suka)
 - Skor 2. (Setuju/Baik/suka)
 - Skor 3. Netral / Cukup
 - Skor 4. Tidak (setuju/baik/) atau kurang
 - Skor 5. Sangat (tidak setuju/buruk/kurang sekali)
4. Total skor dari masing-masing individu adalah penjumlahan dari skor masing-masing item dari individu tersebut.
5. Respon dianalisis untuk mengetahui item-item mana yang sangat nyata batasan antara skor tinggi dan skor rendah dalam skala total. Misalnya,

responden pada upper 25% dan lower 25% dianalisis untuk melihat sampai berapa jauh tiap item dalam kelompok ini berbeda. Item-item yang tidak menunjukkan beda yang nyata, apakah masuk dalam skor tinggi atau rendah juga dibuang untuk mempertahankan konsistensi internal dari pertanyaan [12].

2.3.1 Perhitungan Skala Likert

Sekelompok tim mahasiswa gizi sedang melakukan uji organoleptik (pengujian terhadap bahan makanan berdasarkan kesukaan) sebuah produk dengan menggunakan skala Likert. Aspek yang akan diukur dalam uji organoleptik tersebut adalah cita rasanya. Ada 100 responden atau panelis yang memberikan jawaban dari angket yang diberikan. Berikut rangkuman hasil penilaian 100 responden tersebut [13].

- a. Responden yang menjawab sangat suka (skor 5) berjumlah 8 orang
- b. Responden yang menjawab suka (skor 4) berjumlah 14 orang
- c. Responden yang menjawab netral (skor 3) berjumlah 21 orang
- b. Responden yang menjawab tidak suka (skor 2) berjumlah 31 orang
- c. Responden yang menjawab sangat tidak suka (skor 1) berjumlah 26 orang

2.3.1.1 Rumus $T \times P_n$

T = Total jumlah responden yang memilih

P_n = Pilihan angka skor Likert

- a. Responden yang menjawab sangat suka (5) = $8 \times 5 = 40$
- b. Responden yang menjawab suka (4) = $14 \times 4 = 56$
- c. Responden yang menjawab netral (3) = $21 \times 3 = 63$
- d. Responden yang menjawab tidak suka (2) = $31 \times 2 = 62$
- d. Responden yang menjawab sangat tidak suka (1) = $26 \times 1 = 26$

Semua hasil dijumlahkan, total skor = 247

2.3.2 Interpretasi Skor Perhitungan

mendapatkan hasil interpretasi, terlebih dahulu harus diketahui skor tertinggi (X) dan skor terendah (Y) untuk item penilaian dengan rumus sebagai berikut:

$Y = \text{skor tertinggi likert} \times \text{jumlah responden}$

$X = \text{skor terendah likert} \times \text{jumlah responden}$

Jumlah skor tertinggi untuk item “Sangat Suka” adalah $5 \times 100 = 500$, sedangkan item “Sangat Tidak Suka” adalah $1 \times 100 = 100$. Jadi, jika total skor penilaian responden diperoleh angka 247, maka penilaian interpretasi responden terhadap cita rasa produk tersebut adalah hasil nilai yang dihasilkan dengan menggunakan rumus Index % [13].

2.3.2.1 Rumus Index

Sebelum menyelesaikannya kita juga harus mengetahui interval (rentang jarak) dan interpretasi persen agar mengetahui penilaian dengan metode mencari Interval skor persen.

2.3.2.2 Rumus Interval

$1 = 100 / \text{Jumlah Skor (Likert)}$

Maka $= 100 / 5 = 20$

Hasil (1) $= 20$

(Ini adalah intervalnya jarak dari terendah 0 % hingga tertinggi 100%)

Berikut kriteria interpretasi skornya berdasarkan interval:

- a. Angka 0% – 19,99% = Sangat (tidak setuju/buruk/kurang sekali)
- b. Angka 20% – 39,99% = Tidak setuju / Kurang baik)
- c. Angka 40% – 59,99% = Cukup / Netral
- d. Angka 60% – 79,99% = (Setuju/Baik/suka)
- e. Angka 80% – 100% = Sangat (setuju/Baik/Suka)

Penyelesaian Akhir

$= \text{Total skor} / Y \times 100$

$= 247 / 500 \times 100$

$= 49.4 \%$, berada dalam kategori “Cukup/Netral”

2.4 Jejaring Sosial Facebook

Facebook merupakan salah satu jejaring sosial yang sangat digemari dikalangan masyarakat. Pengertian Facebook menurut wikipedia berbahasa Indonesia adalah sebuah situs web jejaring sosial populer yang diluncurkan pada 4 Februari 2004. Facebook dapat juga diartikan sebuah web jejaring sosial yang didirikan oleh Mark Zuckerberg dan diluncurkan pada 4 Februari 2004. Facebook memungkinkan para pengguna menambahkan profil dengan foto, kontak, ataupun informasi personal lainnya dan dapat bergabung dalam komunitas untuk melakukan koneksi dan berinteraksi dengan pengguna lainnya [14].

Sejarah Facebook berawal ketika Mark Zuckerberg, seorang mahasiswa Harvard kelahiran 14 Mei 1984 dan mantan murid Ardsley High School membuat situs jejaring sosial Facebook. Awalnya, penggunaannya hanya diperuntukkan bagi mahasiswa dari Harvard College. Dalam dua bulan selanjutnya, keanggotaannya diperluas ke sekolah lain di wilayah Boston, Rochester, Stanford, NYU, Northwestern, dan semua sekolah yang termasuk dalam Ivy League. Banyak perguruan tinggi lain yang selanjutnya ditambahkan berturut-turut dalam kurun waktu satu tahun setelah peluncurannya [14]. Hadirnya situs jejaring sosial menurut Melha (2012) adalah suatu kemudahan bagi para penggunanya untuk berinteraksi dengan orang-orang di seluruh dunia dengan biaya yang murah dibandingkan menggunakan telepon dan penyebaran informasi dapat berlangsung secara cepat. Namun, selain memberikan kemudahan bagi penggunanya, kemunculan situs jejaring sosial Facebook juga menyebabkan interaksi interpersonal secara tatap muka (*face to face*) cenderung menurun. Jejaring sosial ini menurut J.A Barnes (1954) dalam Alela (2012) adalah situs yang menunjukkan jalan dimana mereka berhubungan karena kesamaan sosialitas, mulai dari mereka yang dikenal sehari-hari sampai dengan keluarga [14].

Facebook menawarkan keprivasian dan beragam fitur yang sangat lengkap bila dibandingkan dengan situs jejaring sosial lainnya. Facebook menyediakan fitur gabungan antara aplikasi *social networking*, *chatting*, *blogging*, multimedia, *photo sharing* dan bahkan email. Beberapa bagian dalam Facebook adalah profile,

news feed, wall, application, photo, video, poke, group, events, marketplace, post, notes, dan gifts [14].

2.5 API (Application programming Interface)

API adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan *system* operasi. API atau *Application Programming Interface* juga merupakan suatu dokumentasi yang terdiri dari antar muka, fungsi, kelas, struktur untuk membangun sebuah perangkat lunak [15]. Dengan adanya API, maka memudahkan seorang programmer untuk membongkar suatu software untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya. Suatu rutin standar yang memungkinkan developer menggunakan *system function*. Proses ini dikelola melalui operating system. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya untuk saling berinteraksi [15].

API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call*. Hal ini sangat penting dalam hal editing dan pengembangan. *System call interface* ini berfungsi sebagai penghubung antara API dan *system call* yang dimengerti oleh sistem operasi. *System call interface* ini akan menerjemahkan perintah dalam API dan kemudian akan memanggil *system calls* yang diperlukan. Untuk membuka suatu file tersebut user menggunakan program yang telah dibuat dengan menggunakan bantuan API, maka perintah dari user tersebut diterjemahkan dulu oleh program menjadi perintah `open()` [15]. Perintah `open()` ini merupakan perintah dari API dan bukan perintah yang langsung dimengerti oleh kernel sistem operasi. Oleh karena itu, agar keinginan pengguna dapat dimengerti oleh sistem operasi, maka perintah `open()` tadi diterjemahkan ke dalam bentuk *system call* oleh *system call interface*. Implementasi perintah `open()` tadi bisa bermacam-macam tergantung dari sistem operasi yang digunakan. Cara menggunakan API :

1. Dilakukan dengan mengimpor package/kelas

2. Ada beberapa kelas bernama sama *dipackage* yang berbeda, yaitu :

- Import salah satu dan gunakan nama lengkap untuk yang lain
- Gunakan nama lengkap semua kelas.

Kebanyakan sistem operasi seperti Windows, menyediakan fasilitas API sehingga programmer dapat melakukan aktivitas programming dengan lebih konsisten. Meskipun API didesain untuk programmer, namun API juga baik untuk *user* karena setidaknya dapat menjamin bahwa program tersebut memiliki *interface* yang sama, sehingga lebih mudah untuk dipelajari [15].

API dapat diklasifikasikan menjadi beberapa kategori, hal ini dilihat dari abstraksi apa yang dideskripsikan di dalam system. Kategori-kategori ini diantaranya [16]:

Tabel 2.1 Kategori API [16]

Kategori API	Deskripsi	Contoh
<i>Operating System</i>	API yang digunakan untuk fungsi dasar yang dapat dilakukan oleh computer seperti proses I/O, eksekusi program	API for MS Windows
<i>Programming Languages</i>	API yang digunakan untuk fungsi dasar yang dapat dilakukan oleh computer seperti proses I/O, eksekusi program	Java API
<i>Application Services</i>	API yang digunakan untuk mengakses data dan layanan yang disediakan dari suatu aplikasi.	API for mySAP (BAPI/ <i>Business Application Programming Interface</i>)
<i>Infrastructure Services</i>	Digunakan untuk mengakses infrastruktur dari suatu komputer beserta peripheral seperti storage,	Amazon EC2 (<i>Elastic Compute Cloud</i>) untuk akses pada virtual computing dan amazon S3

	aplikasi, dan lain-lain.	(<i>Simple Storage Service</i>) untuk menyimpan data dalam jumlah yang besar.
<i>Web Services</i>	API yang digunakan untuk mengakses content dan layanan yang disediakan oleh suatu <i>web application</i> .	Facebook Graph API yang digunakan untuk mengakses informasi yang dapat dibagikan

2.6 Facebook API

Graph API merupakan inti dari Facebook platform yang memungkinkan pengembang untuk membaca dan menulis data pada Facebook. Graph API merupakan perkembangan dari *Old Rest* API. Paradigma membaca dan menulis data dari yang berorientasi pada fungsi berubah menjadi berorientasi *obyek*. Objek Facebook contohnya seperti *Users*, *Photos*, *Albums*, dan koneksi diantaranya. Pendekatan membuat *Facebook* API lebih sederhana dan konsisten saat bekerja dengan *obyek* [4].

2.7 API EventBrite

Eventbrite adalah platform teknologi acara terbesar di dunia, yang memotori jutaan acara di lebih dari 180 negara dan wilayah. Tim pengoptimalan mesin telusur (SEO) mereka mulai menggunakan data terstruktur di situs mereka pada 2015 untuk meningkatkan visibilitas dan kemudahan dalam menemukan acara kreator di internet. Karena itu, mereka sudah sangat siap ketika Google meluncurkan pendekatan baru dalam pengalaman penelusuran acara di platform seluler pada Mei 2017 [17]. "Ketertarikan pengguna pada layanan kami lumayan tinggi berkat upaya SEO kami", kata Allen Jilo, product manager *Eventbrite*. "Saat Google mulai memanfaatkan markup schema.org untuk acara, kami melihat bahwa acara muncul pada hasil penelusuran dengan cara berbeda. Kami tertarik untuk bekerja sama lebih intensif dengan Google untuk memperluas kapabilitas baru tersebut [17].

Sama halnya dengan bisnisnya terdahulu, *Eventbrite* juga memiliki jangkauan pasar yang sangat luas. Hal ini tentu akan berbanding lurus dengan peluang sukses yang dimiliki oleh layanan yang satu ini. Di dalam *Eventbrite*, para pemilik *even organizer* dapat melakukan kegiatan mereka dengan lebih

leluasa dan nyaman, di mana mereka memiliki kesempatan untuk melakukan berbagai hal, seperti: melakukan perencanaan, mengoptimalkan kegiatan promosi, dan juga menjual tiket untuk berbagai acara yang akan mereka gelar. Bukan hanya itu saja, platform ini bahkan memungkinkan *even organizer* tersebut untuk melakukan kegiatan publikasi melalui Facebook, Twitter, dan juga berbagai sosial media lainnya [18].

Dalam hal ini Integrasi *Eventbrite* dengan Facebook kini masuk pada tahap baru. *Eventbrite* menambahkan fasilitas yang memungkinkan para pengguna yang telah memiliki akun Facebook untuk mendapatkan rekomendasi sosial yang berhubungan dengan acara dari teman Facebook mereka [6].

Eventbrite menggunakan Facebook *Connect* API untuk mendorong aktivitas *social event discovery* serta memberikan fasilitas dimana para pengguna *Eventbrite* bisa menemukan rencana acara yang akan dihadiri oleh teman-teman mereka [6]. Dengan berbagai fasilitas ini, maka peluang *even organizer* untuk bisa sukses dalam menggelar sebuah acara akan semakin besar, sebab mereka dimudahkan oleh keberadaan *Eventbrite*. Hal ini juga akan memudahkan orang-orang untuk bisa lebih mudah dalam menemukan dan membeli berbagai tiket yang mereka butuhkan untuk menyaksikan pertunjukan / acara yang mereka minati. Berbagai kemudahan inilah yang kemudian mengantarkan *Eventbrite* menjadi salah satu layanan terbesar yang ada saat ini [6].

Jadi API *Eventbrite* memungkinkan komunikasi antara *Eventbrite* dan aplikasi atau situs *web* Anda menggunakan permintaan HTTP. Konten data permintaan dan respons yang ditransmisikan diformat sebagai JSON , menggunakan pengkodean UTF-8 dan nilai-nilai yang disandikan URL.

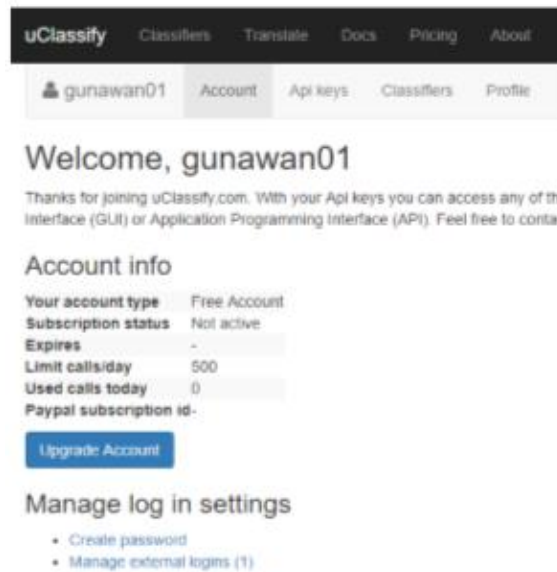
2.8 Uclassify API

Uclassify adalah layanan API yang melakukan banyak fungsi seperti deteksi Bahasa, teks gender dan mendeteksi usia, spam filter, analisis sentimen, dokumen tagging dan dapat mendeteksi emosi. Berikut adalah demo dari *uclassify* disini bisa membuat *classifier* baru. Akan disediakan sebuah *class* yaitu untuk menentukan kata atau kalimat bersifat negatif atau positif. Dengan cara

melakukan pelatihan pada data atau kata yang akan digunakan dalam *classifier* baru. API *Uclassify* dapat diakses dengan pilihan gratis dan berbayar. *Uclassify* telah banyak dikembangkan dalam analisis sentimen terutama pada media sosial. Berikut adalah proses pembuatan classifier baru pada *WEB API Uclassify* [16].

1. Pembuatan *Classifier* Baru

Berikut adalah halaman utama *Uclassify*.



Gambar 2.2 Halaman Utama Uclassify [16]

2.9 Android

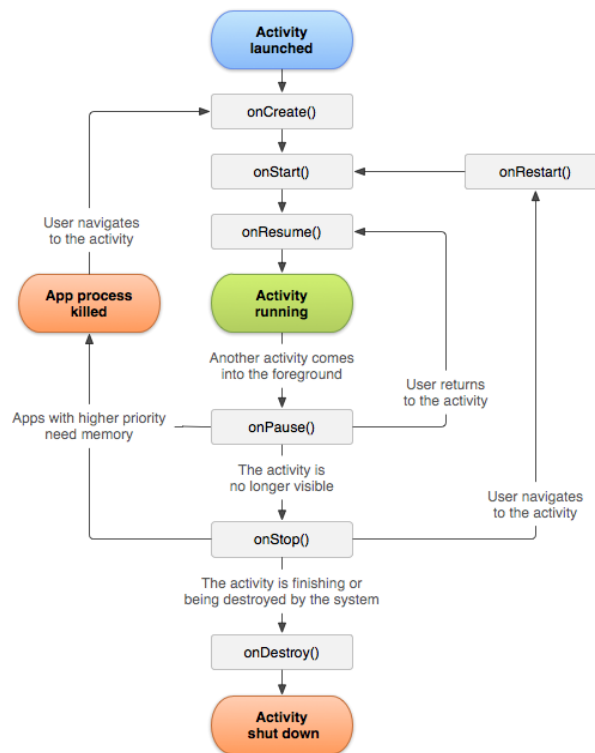
Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh *Android Inc*, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh *Google Inc*. Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [19].

Pada saat perilisannya pertama Android, 5 november 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan open source pada

perangkat mobile. Di lain pihak, Google merilis kode – kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler [19].

2.9.1 Android Life Cycle

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.



Gambar 2.3 Android Life Cycle

Sumber : https://developer.android.com/guide/components/images/activity_lifecycle.png

Terdapat beberapa state dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gambar 2.1 Siklus Hidup Android, 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*. 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()*, kemudian diikuti oleh `onResume()`.

2.9.2 Android SDK (Software Development Kit)

Android SDK adalah tool API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman java. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk memulai mengembangkan aplikasi pada platform android. Beberapa fitur-fitur android yang paling penting adalah

- a. Framework aplikasi yang mendukung penggantian komponen dan reusable.

- b. *Intergrated browser* berdasarkan *engine open source webkit*
- c. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (opsional akselerasi hardware)
- d. SQLite untuk menyimpan data
- e. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephong (tergantung *hardware*)
- f. Bluetooth,EDGE, 3G, 4G, dan WiFi. (tergantung *hardware*)
- g. Kamera, GPS, kompas, dan accelerometer. (tergantung *hardware*)
- h. Lingkungan Deuelopment yang lengkap dan kaya termasuk perangkat emulator, fools untuk debugging, profil dan kinerja memori, danplugin untuk IDE Eclipse [19].

2.9.3 Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut:

1. *Applications* dan *Widgets*

Applications dan *Widgets* ini adalah layer di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita download aplikasi kemudian kita lakukan instlasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java [19].

2. *Applications Framworks*

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status notifi.cations, dan sebagainya. pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang kategori inti.

Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*) [16].

Sehingga bisa kita simpulkan *Applications Frameworks* ini adalah layer di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *contentproviders* yang berupa sms dan panggilan telepon [19].

Komponen-komponen yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut [19]:

- a. *Views*
- b. *Content Provider*
- c. *Resource manager*
- d. *Notification manager*
- e. *Activity manager* [19].

3. Libraries

Libraries ini adalah layer di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, Layer ini meliputi berbagai library C/C++ ini seperti *Libc* dan *SSL*, serta [19]:

1. *libraries media* untuk pemutaran media audio dan video.
2. *libraries* untuk manajemen tampilan
3. *libraries Graphics* mencakup *SGL* dan *OpenGL* untuk grafis 2D dan 3D
4. *libraries SQLite* untuk dukungan database.
5. *libraries SSL* dan *WebKit* terintegrasi dengan web browser dan security.
6. *libraries LiveWebcore* mencakup modern web browser dengan engine embeded web view.
7. *libraries 3D* yang mencakup implementasi *OpenGL ES 1.0 API's* [19].

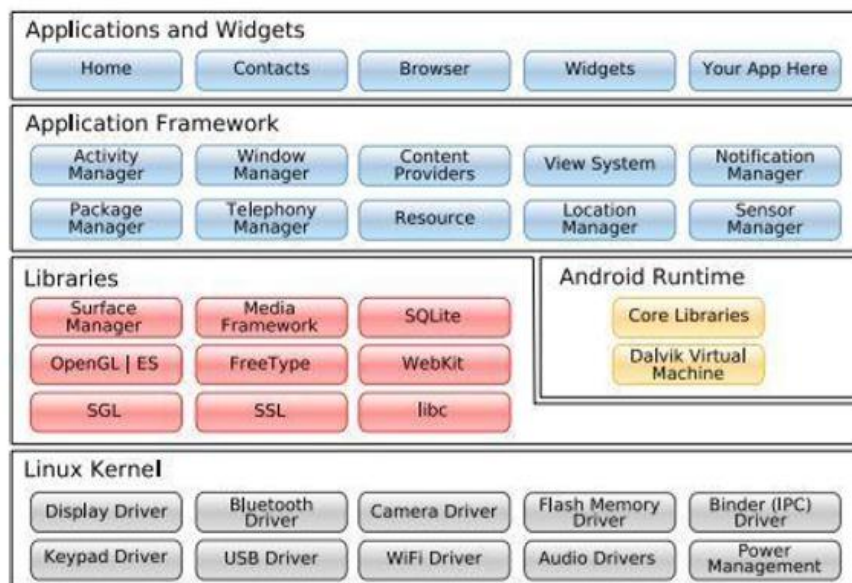
4. Android *Run Time*

Layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan Implementasi *Linux. Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam *Android Run Time* dibagi menjadi dua bagian yaitu [19]:

1. *Core Libraries*: Aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh *Core Libraries* [19].
 2. *Dalvik Virtual Machine*: Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, di mana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah [19].
5. *Linux Kernel*

Linux kernel adalah *layer* di mana inti dari operating sistem dari Android itu berada. Berisi "*file-file system* yang mengatur sistem *processing, memory, resource, drivers*, dan sistem-sistem operasi android lainnya [19].

Linux kernel yang digunakan android adalah linux kernel *release 2.6*.



Gambar 2.4 Linux Kernel [19].

2.9.4 Versi Android

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2010 diperkirakan hampir semua vendor seluler di dunia menggunakan Android sebagai operating sistem. Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut [19]: .

a. Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam, alarm, voice search (pencarian suara), pengiriman pesan dengan gmail, dan pemberitahuan email [19].

b. Android versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, meng-upload video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem [19].

c. Android versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan, CDMA/EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine, kemampuan diol kontak, teknologi text to change speech (tidak tersedia pada semua ponsel, pengadaan resolusi WVGA [19].

b. Android versi 2.0/2.1

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan;flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

c. Android versi 2.2 (Froyoz FrozenYoghurt)

Pada bulan mei zero Android versi 2.2 Rev 1 diluncurkan. Android inilah yang sekarang sangat banyak beredar di pasaran, salah satunya adalah dipakai di Samsung FX tab yang sudah ada di pasaran. Fitur yang tersedia di Android versi ini sudah kompleks di antaranya adalah [19]: .

- Kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- DalvikVirtual Machine dioptimalkan untuk perangkat mobile.
- Grafik: grafik di 2D dan grafis 3D berdasarkan libraries OpenGL.
- SQLite: untuk penyimpanan data.
- Mendukung media: audio, uideo, dan berbagai format gambar (MPEG4, H.264, MP3,AAC, AMR, JPG, PNG, GIF) .
- GSM, Bluetooth, EDGE, 3G, dan WiFi (hardware independent) .
- Kamera, Global Positioning System (GPS), kompas, dan accelerometer (tergantung hardware) [19].

d. Android versi 2.3 (Gingerbread)

Android versi 2.3 diluncurkan pada Desember 2010, hal-hal yang direvisi dari versi sebelumnya adalah kemampuan seperti berikut:

- S/P-basedVoIP .
- Near Field Communications (NFC) .
- Gyroscope dan sensor .
- Multiple cemas support .
- Mixable audio effects .
- Download manager

- e. Android versi 3.0 (Honeycomb) dirilis Februari 2011 sebagai android 3.0 revisi 1 serta android versi 3.0 revision 2 telah dirilis pada juli 2011.
- f. Android versi 3.1 dirilis Mei 2011, sedangkan android 3.1 revisi 2 juga dirilis mei 2011, serta android 3.1 revision 3 dirilis pada juli 2011.
- g. Android versi 3.2 dirilis Juli 2011.
- h. Android versi 4.0 dirilis November 2011 [19].

Android versi 3.0 ke atas adalah generasi platform yang digunakan untuk tablet pc. Sementara versi 4.0 sudah merupakan platform yang bisa dipakai di smartphone dan tablet pc. Demikian beberapa versi android yang sudah dirilis sampai dengan buku ini ditulis, kemungkinan besar versi tersebut akan terus berkembang seiring dengan kebutuhan yang sangat kompleks di bidang penggunaan smartphone. Untuk informasi detail versi platform atau untuk mendapatkan informasi versi terbaru android Anda dapat melihat di www.developer.android.com [19].

2.10 Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Bila seorang analis dan programmer akan membuat flowchart, ada beberapa petunjuk yang harus diperhatikan, seperti :

1. *Flowchart* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja..

5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada *flowchart* yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
7. Gunakan simbol-simbol flowchart yang standar.

2.11 Global Positioning System (GPS)

Global Positioning System adalah sistem navigasi berbasis satelit terdiri dari jaringan 24 satelit ditempatkan ke orbit oleh Departemen Pertahanan Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978. Layanan GPS dahulu hanya dipergunakan untuk keperluan militer namun mulai terbuka untuk publik. 24 satelit GPS tersebut berada sekitar 12.000 mil di atas bumi bergerak mengelilingi bumi 12 jam dengan kecepatan 7.000 mil per jam. Satelit GPS berkekuatan energi sinar matahari, memiliki baterai cadangan untuk menjaga agar tetap berjalan pada saat gerhana matahari atau pada saat tidak ada energi matahari dan memiliki roket penguat kecil pada masing-masing satelit agar dapat mengorbit tepat pada tempatnya.

GPS adalah singkatan dari Global Positioning System, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, dimana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS, seperti [10]:

- a. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
- b. Lokasi. GPS memberikan informasi lokasi dalam tiga dimensi:
 1. *Latitude*
 2. *Longitude*
 3. *Elevasi*

- c. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
- d. Arah perjalanan. GPS dapat menunjukkan arah tujuan.
- e. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.
- f. Komulasi data. GPS *receiver* dapat menyimpan informasi *track*, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya [10].

2.12 JavaScript Object Notation (JSON)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [20]. JSON terbuat dari dua struktur [20]:

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 [20].
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*) [20].

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini [20].

2.13 Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Java dikembangkan pada bulan Agustus 1991, dengan nama semula Oak. Pada Januari 1995, karena nama Oak dianggap kurang komersial, maka diganti menjadi Java. Pada Desember 1998, Sun memperkenalkan nama “Java 2” (J2) sebagai generasi kedua dari java *platform*. Konvensi nama baru ini diterapkan untuk semua edisi Java yaitu Standard Edition (J2SE), *Enterprise Edition* (J2EE), dan *Micro Edition* (J2ME).

Ada tiga platform Java yang masing-masing diarahkan untuk tujuan tertentu dan untuk lingkungan komputasi yang berbeda-beda:

1. *Standard Edition* (J2SE) : J2SE merupakan inti dari bahasa pemrograman Java. J2SE didesain untuk jalan pada komputer desktop dan komputer *workstations*.
2. *Enterprise Edition* (J2EE): Dengan built-in mendukung untuk *servlets*, JSP, dan XML, edisi ini ditujukan untuk aplikasi berbasis server.
3. *Micro Edition* (J2ME) : Didesain untuk piranti dengan memori terbatas, layar display terbatas dan power pemrosesan yang juga terbatas.

2.13.1 Processor Hypertext Protocol (PHP)

PHP merupakan salah satu pemrograman *server-side* di antara beberapa pemrograman yang ada. Sejak di luncurkan , PHP mendapat respon yang sangat baik dari kalangan pengembang aplikasi web kemudahannya untuk di pahami, serta sintaksnya yang mirip dengan Bahasa C menjadikan pemrograman ini cepat di kenal di kalangan luas. Tanpa mengurangi kemampuan-kemampuan yang sudah ada, PHP 5 hadir dengan fitur lebih kompleks , dan merupakan rilis terbaru yang di keluarkan oleh pengembangnya secara besar besaran pengembang PHP merombak bug yang sering di jumpai daam PHP 4, hal ini akan sangat jelas terasa ketika anda membuat aplikasi berbasis objek. Dengan tujuan untuk lebih meningkatkan kinerja aplikasi, tanpa bermaksud membuang fitur yang ada , PHP

5 menambahkan fitur – fitur baru seperti dukungan terhadap MYSQL. Dukungan ini di wujudkan dalam bentuk fungsi- fungsi lama dan di tulis ulang [22].

2.14 Android Studio

Android Studio adalah *Integrated Development Environment* resmi untuk pengembangan aplikasi Android dan bersigat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada *event* Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android. Android Studio digunakan pada pembangunan aplikasi ini di karenakan IDE resmi untuk pembangunan aplikasi android.

2.15 HTML

Hyper Text Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web. *Weibsite* yang dibuat dengan HTML ini, dapat di lihat oleh semua orang yang terkoneksi dengan internet . Tentunya dengan menggunakan aplikasi penjelajah atau browser.

Tiap kali kita mengakses dokumen web, maka sesungguhnya kita mengakses dokumen seseorang yang ditulis dengan menggunakan format HTML. Beberapa orang merasa keberatan jika dikatakan HTML adalah sebuah bahasa pemrograman karena struktur yang dimilikinya dianggap terlalu sederhana, kode-kode dibaca oleh *browser* baris per baris, dari atas ke bawah. HTML juga tidak memiliki '*looping*' seperti bahasa pemrograman lain. Pada HTML dipergunakan *hypertext link* atau hubungan antara teks dan dokumen lain. Dengan demikian pembaca dokumen bisa melompat dari satu dokumen ke dokumen yang lain dengan mudah.

2.16 MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya. MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979.

Awalnya Tcx merupakan perusahaan pengembang *software* dan konsultan database, dan saat ini MySQL sudah diambil alih oleh Oracle Corp.

Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan database perusahaan-perusahaan yang berskala kecil sampai menengah, MySQL juga bersifat *open source* (tidak berbayar). MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pembangun aplikasi *web* yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman script PHP [22].

2.17 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek. Pada pengembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang dibuat dan perlu adanya standarisasi agar orang diberbagai negara dapat mengerti pemodelan perangkat lunak [21].

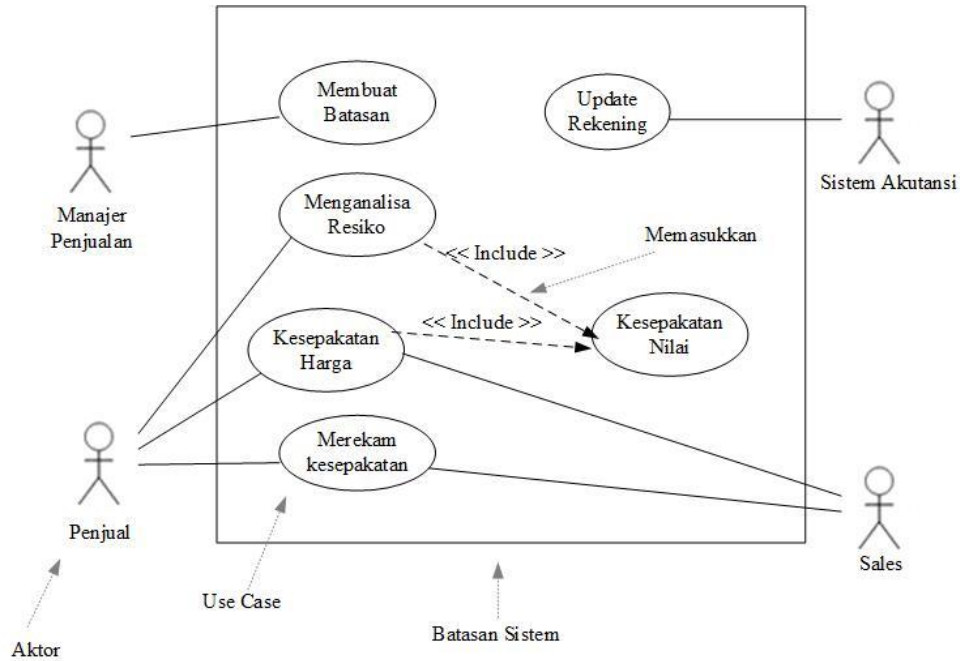
2.17.1 Use Case Diagram

Use case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Sebuah *use case* adalah serangkaian skenario yang dikemas menjadi satu oleh tujuan pengguna umum. Dalam bahasa *use case*, para pengguna disebut sebagai aktor. Aktor merupakan sebuah peran yang dimainkan oleh seorang pengguna dalam kaitannya dengan sistem. aktor tidak harus manusia. Jika sebuah sistem melakukan layanan untuk sebuah sistem komputer lain, sistem lain tersebut merupakan aktor.

Use case diagram menampilkan aktor, *use case*, dan hubungan antara mereka:

- a. Aktor mana yang menggunakan *use case* mana.
- b. *Use case* mana yang memasukkan *use case* lain.

Berikut ini adalah contoh dari diagram *use case* :

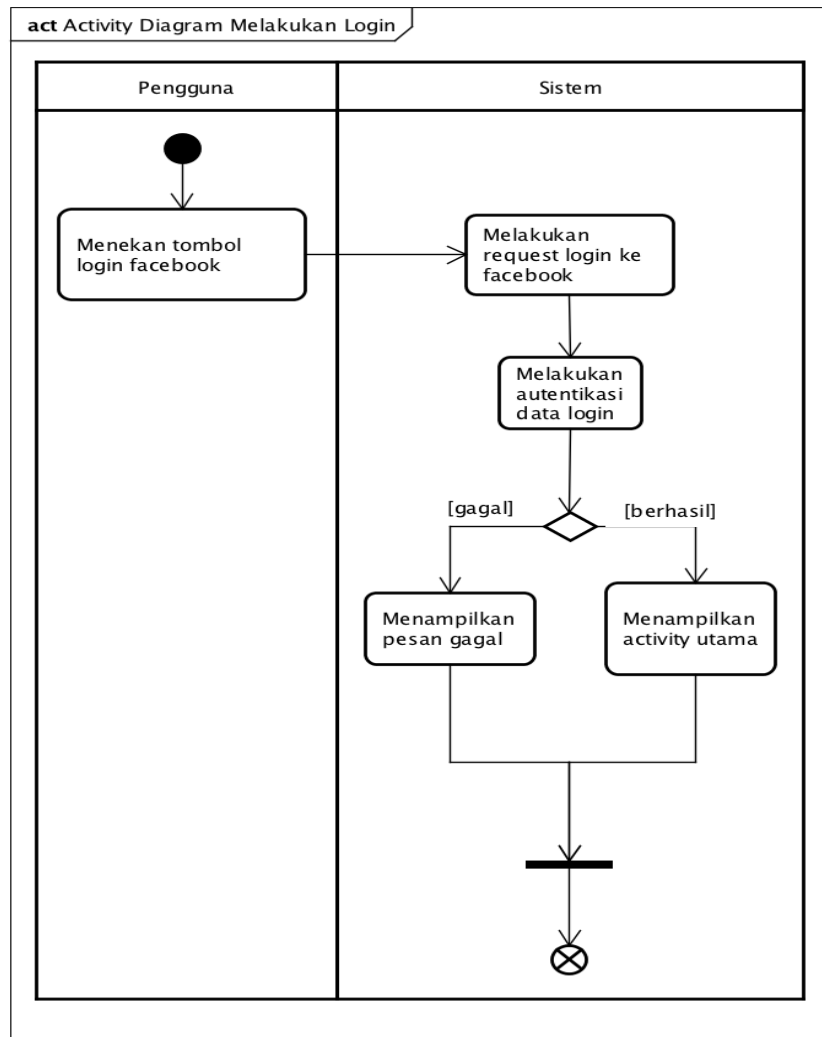


Gambar 2.5 Contoh Diagram Use Case

[Cockburnm *use case*] menjelaskan sebuah skema tingkatan *use case*. Inti *use case* berada pada tingkat sea level. *Use case sea level* khususnya mewakili sebuah interaksi diskrit antara aktor utama dan sistem. *Use case* yang ada disana hanya karena mereka dimasukkan oleh *use case sea level* adalah *fish level*. Lebih tinggi, *use case kite level* menampilkan bagaimana *use case sea level* sesuai dengan interaksi bisnis yang lebih luas. *Use case kite level* biasanya merupakan *use case bisnis*, sedangkan *use case sea* dan *fish level* merupakan *use case sistem*. Kebanyakan *use case* Anda berada pada tingkatan *sea level*

2.17.2 Activity Diagram

Diagram *activity* adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur

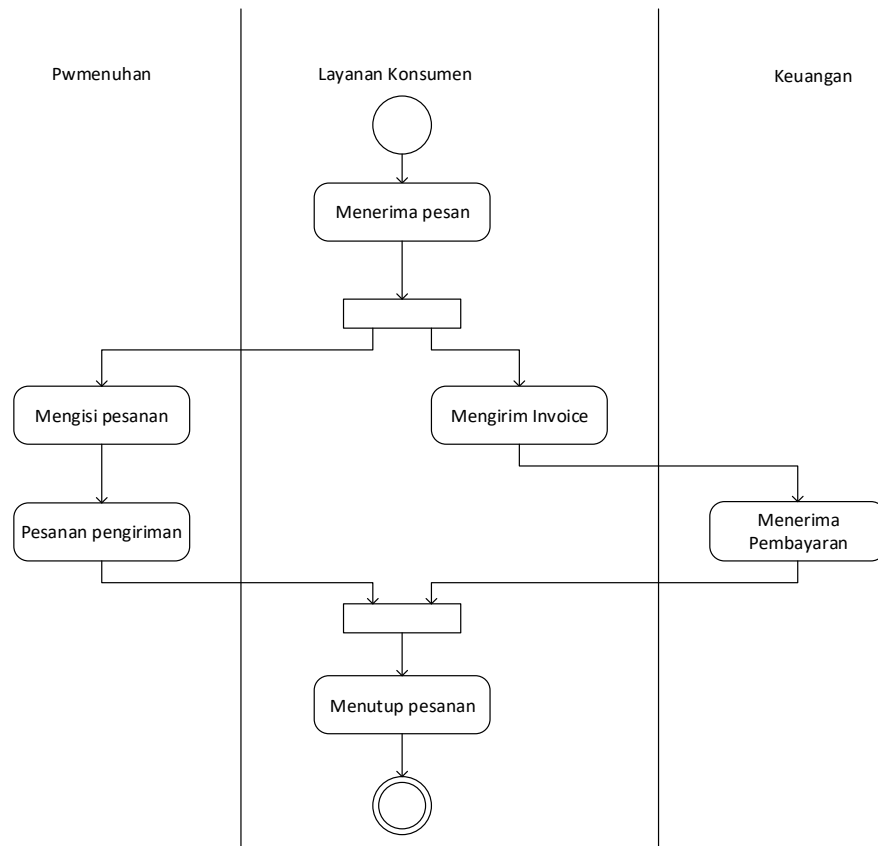


kerja.

Gambar 2.6 Contoh Diagram Aktivitas

Activity diagram memberi tahu anda tentang apa yang terjadi, tetapi diagram ini tidak memberi tahu anda tentang siapa yang melakukan apa. Jika anda ingin menunjukkan siapa melakukan apa, anda dapat memisahkan activity diagram ke dalam partisi-partisi yang menampilkan action mana yang dilakukan oleh sebuah class atau organisasi. Pembuatan partisi pada Gambar 2.7 merupakan

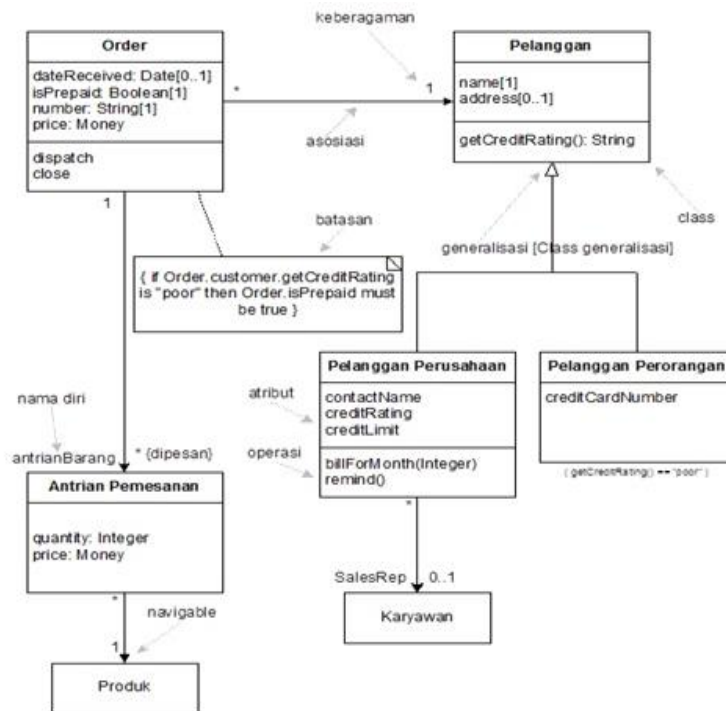
pembuatan partisi sederhana satu dimensi. Model ini sering disebut sebagai swim lanes.



Gambar 2.7 Partisi Pada Activity Diagram

2.17.3 Diagram Kelas

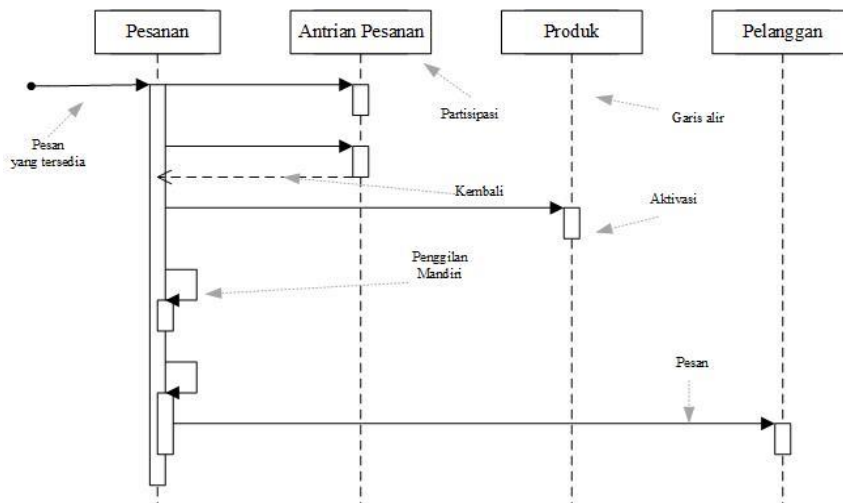
Diagram *class* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antara mereka. Diagram *class* juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat pada sebuah hubungan-hubungan objek. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah *class*. Didalam class dibagi menjadi tiga bagian. Nama *class*, atributnya, dan operasinya.



Gambar 2.8 Contoh Diagram Class

2.17.4 Sequence Diagram

Diagram *sequence* secara khusus menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek di dalam *use case*. Diagram *sequence* menunjukkan interaksi dengan menampilkan setiap partisipan dengan garis alir secara vertikal dan pengurutan pesan dari atas ke bawah.



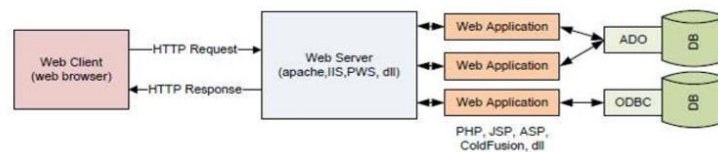
Gambar 2.9 Contoh Diagram Sequence

2.18 Web Browser

Web browser atau yang lebih dikenal browser adalah suatu program atau aplikasi untuk mencari sebuah informasi dari suatu halaman web/blog. Browser yang sering digunakan oleh para pengguna internet yaitu *Internet Explorer*, *Mozilla Firefox*, *Opera*, *Google Chrome*, *Browser* dan lain-lain.

2.19 Web Server

Web server adalah suatu perangkat lunak yang berfungsi untuk melayani aktifitas *request and reply file-file web*. Salah satu web server yang paling banyak digunakan saat ini adalah *Apache Web Server*. Keunggulan *Apache* antara lain : bersifat open source, gratis, memiliki dukungan luas terhadap bahasa pemrograman web, antara lain PHP, JSP, Perl, dan lain sebagainya. [17]



Gambar 2.12 Arsitektur Web Server

(sumber : [http://programing.infogue.com/Web Server/07-04-2011](http://programing.infogue.com/Web%20Server/07-04-2011))

