BAB 2
TINJAUAN PUSTAKA

2.1 Kota Bandung

Kota Bandung adalah ibu kota provinsi Jawa Barat. Kota ini pada zaman dahulu dikenal sebagai Paris Van Java (bahasa Belanda) atau "Paris dari Jawa". Karena terletak di dataran tinggi, Bandung dikenal sebagai tempat yang berhawa sejuk. Hal ini menjadikan Bandung sebagai salah satu kota tujuan wisata [2].

Bandung terletak pada koordinat 107° BT and 6° 55' LS. Luas Kota Bandung adalah 16.767 hektar. Kota ini secara geografis terletak di tengah-tengah provinsi Jawa Barat, dengan demikian, sebagai ibu kota provinsi, Bandung mempunyai nilai strategis terhadap daerah-daerah di sekitarnya. Kota Bandung terletak pada ketinggian ±768 m di atas permukaan laut, dengan di daerah utara pada umumnya lebih tinggi daripada di bagian selatan. Ketinggian di sebelah utara adalah ±1050 dpl, sedangkan di bagian selatan adalah ±675 dpl. Bandung dikelilingi oleh pegunungan, sehingga Bandung merupakan suatu cekungan (Bandung Basin).

Untuk urusan pariwisata, Bandung memang belum bisa menandingi Yogyakarta, apalagi Bali. Namun banyak jenis wisata unik dan menarik yang ditawarkan, mulai dari wisata jajanan dan wisata belanja yang paling menarik wisatawan domestik maupun mancanegara, sampai wisata alam, wisata budaya, wisata sejarah, bahkan wisata loak. Pokoknya, ada seabrek yang bisa diubek-ubek di kota ini. Yang dibutuhkan hanya cukup uang, dan yang penting cukup jeli.

Untuk menuju kota Bandung saat ini sudah dapat ditempuh dengan menggunakan pesawat terbang dari Jakarta, Bali, Medan atau bahkan dari Kuala Lumpur Malaysia dan Singapura. Untuk wisatawan yang menggunakan kendaraan pribadi, bisa menggunakan jalan Tol Purbaleunyi dan masuk melalui bberapa akses gerbang tol yang ada di kota bandung yaitu di antaranya adalah Pintu gerbang Tol Baros, Pintu Gerbang Tol Pasir Koja, Pintu Gerbang Tol Kopo, Pintu Gerbang Tol

Moh Toha, Pintu Gerbang Tol Buah Batu, Pintu Gerbang Tol Cileunyi, Pintu Gerbang Tol Pasteur.

2.2 Potensi Oleh-Oleh Bandung

Banyaknya potensi wisata oleh-oleh dan tempat perbelanjaan oleh-oleh di Kota Bandung yang ada. Membuat warga luar kota bandung tertarik untuk datang ke kota kembang untuk berwisata serta membeli oleh-oleh yang ada di bandung, dari yang biasa maupun yang ternama. Sehingga banyak warga luar bandung yang berdatangan dari daerah masing-masing utuk berkunjung ke kota bandung.

Luasnya kota bandung dan banyaknya tempat wisata serta tepat oleh-oleh membuat warga pendatang dan warga asli kota bandung kesulitan untuk menemukan informasi mengenai tempat oleh-oleh maupun wisata yang ada di kota Bandung, dan seringkali membuat masyarakat kesulitan untuk menemukan rute atau jalan untuk menuju lokasi wisata dan oleh-oleh yang ada di kota bandung.

Sektor pariwisata merupakan salah satu sektor yang memiliki potensi cukup besar bila dikembangkan dan dikelola secara optimal. Jika membahasa mengenai pariwisata tentu saja tidak bisa lepas dengan souvenir, cinderamata, atau oleh-oleh. Setiap daerah yang menjadi tujuan wisata memiliki berbagai macam oleh-oleh yang banyak dicari oleh wisatawan. Selama ini di Bandung tidak semua orang tahu informasi mengenai tempat untuk memperoleh oleh-oleh tersebut.

Perkembangan teknologi saat ini lebih mengedepankan pada penggunaan perangkat *mobile* seperti *smartphone*. Pada dasarnya *smartphone* diciptakan tidak hanya untuk menelpon ataupun mengirim pesan saja, tetapi *smartphone* dapat melakukan beberapa hal yang sebelumnya hanya bisa dilakukan pada perangkat PC (*Personal Computer*), seperti mengakses internet, menonton video, mendengarkan musik, dan juga membaca dokumen. Saat ini *smartphone* juga telah dilengkapi dengan fasilitas-fasilitas canggih yang memungkinkan *user* melakukan apapun yang dia inginkan sesuai dengan kebutuhannya seperti fasilitas kamera berkualitas tinggi dan

juga *pixel* yang besar sehingga dapat memungkinkan *user* untuk mengambil gambar dengan hasil yang maksimal baik untuk foto maupun video, adapun fitur lain seperti GPS (*Global Positioning System*), *Wi-Fi hotspot*, dan juga *tethering.*

Di sisi lain perkembangan teknologi akhir-akhir ini cenderung mengarah kepada teknologi mobile. Sejalan dengan perkembangan teknologi *mobile* semakin meningkat pesat yang ditandai dengan semakin banyaknya fungsi pada perangkat *mobile* tersebut selain fungsi utamanya sebagai alat komunikasi. Sistem operasi pada telepon selular juga terus mengalami perkembangan, diantaranya adalah android. Pengguna ponsel berbasis android juga terus bertambah seiring semakin terjangkaunya harga ponsel berbasis android. Sebenarnya akses informasi mengenai tempat oleh-oleh di Bandung sudah dapat di akses di website. Namun kebanyakan wisatawan tidak mengakses situs tersebut. Biasanya situs baru diakses setelah mengalamai kebingungan saat mencari suatu tempat oleh-oleh di Bandung. Wisatawan tidak tahu dimana lokasi oleh-oleh yang terdekat dari posisi mereka dan mereka akan sungkan jika bertanya lokasi oleh-oleh terdekat pada warga sekitar apalagi jumlah lokasi oleh-oleh di Bandung sangat banyak. Untuk itu sangat diperlukan suatu aplikasi yang mampu merencanakan dan menyediakan rute terpendek mencapai tempat tujuan akhir. Jarak terpendek merupakan suatu permasalahan yang sering timbul pada pengguna transportasi, karena pengguna transportasi dalam melakukan perjalanan membutuhkan solusi bagaimana rute yang akan dilalui.

2.2.1 Kategori Potensi Oleh-oleh Bandung

Kategori potensi oleh-oleh Bandung merupakan salah satu kategori yang membedakan antara tempat dan lokasi oleh-oleh di kota Bandung, adapun pengkategorian tempat oleh-oleh Bandung diantaranya seperti macam-macam oleh-oleh makanan, oleh-oleh barang seperti fashion dan tempat perbelanjaan seperti mall dan pasar tradisional yang terdapat di Kota Bandung. Beberapa contoh oleh-oleh makanan yang cukup terkenal di Bandung seperti : gorengan oncom/tempe, peyeum

Bandung, Batagor, Baso Tahu, Gepuk, Keripik dan lain-lain. Selain makanan terdapat juga beberapa contoh oleh-oleh berupa barang / fashion di Bandung seperti : Jeans, kaos bandung, jaket, sepatu kulit, tas, dan lain-lain. Dan juga terdapat tempat perbelanjaan seperti mall dan pasar tradisional yang didalamnya menjual aneka macam-macam oleh-oleh, adapun beberapa contoh tempat perbelanjaan seperti : Mall Bandung Indah Plasa, Mall Istana Plasa, Mall Cihampelas Walk, serat Pasar Tradisional seperti Pasar Baru, dan ITC.

2.2.1.1 Oleh-oleh Makanan Bandung

Adapun macam-macam kategori oleh-oleh jenis makanan di kota bandung yang terkenal yaitu diantaranya seperti :

1. Pisang Bolen yang dapat di beli di Outlet Kartika Sari Di Daerah :
    - Jl. Dago
    - Jl. Buah Batu,
    - Jl. Kopo
    - Jl. Antapani
    - Jl. Jakarta
    - Jl. Kebon Kawung
    - Jl. Kebon Jukut
    - Jl. Cimahi.
2. Gepuk Nyonya Ong yang terletak di kawasan daerah :
    - Jl. Pasteur
    - Jl. Pasir Kaliki
    - Jl. Setiabudi
    - Jl. Buah Batu.
3. Outlet Brownis Kukus Amanda yang terletak di kawasan daerah :
    - Jl. Ranca Bolang
    - Jl. Lodaya

- Jl. Antapani
- Jl. Dr. Oten
- Jl. Pasteur
- Jl. Cihampelas.

4. Surabi Bandung yang terkenal dapat di beli di kawasan daerah :
    - Jl. Setiabudi (ENHAI)
    - Jl. Cisangkuy, The Kiosk (Cihampelas, Braga)
    - Jl. Kopo
    - Jl. Abdurahman Saleh (Husen).
5. Kerikip Singkong merupakan salah satu oleh-oleh yang cukup dikenal di kota Bandung dan cukup banyak terdapat di toko oleh-oleh Bandung seperti keripik Karuhun, Ma'Icih yang dapat ditemukan dan dibeli di gerai Mini Market seperti :
    - Indomart
    - Alfamart
    - Yomart

    serta terdapat juga di gerai toko oleh-oleh yang berada di kawasan :
    - Jl. Pasteur
    - Jl. Cihampelas
    - Jl. Leuwipanjang
6. Baso Tahu Siomay makanan satu ini termasuk banyak terdapat dimana mana, namun terdapat juga Baso Tahu Tulen yang cukup terkenal di Bandung yaitu Baso Tahu Saboga yang terdapat di daerah :
    - Baso Tahu Saboga Jl. Burangrang, Jl. Pasteur (Mall BTC), Jl. Pasir Kaliki (Mall IP), Jl. Purnawarman (Mall BEC), Jl. Gatot Subroto (Mall TSB), Jl. Peta (Mall Citylink).
7. Batagor Cemilan/jajanan/oleh-oleh khas Bandung yang satu ini memang banyak ditawarkan di Bandung. Beberapa yang terkenal sebagai tempat

memborong oleh-oleh Bandung yang satu ini adalah *Batagor Kingsley*, *Batagor Riri*, *Batagor H. Isan*, *Batagor Abuy*. Oleh-oleh yang satu ini dapat di dapatkan di kawasan daerah :

- Outlet Batagor Riri, Jl. Burangrang Jl. Pasir Kalili (Batagor Riri - Hyper Square), Jl. Kebon Kawung, Jl. Setiabudi.
- Batagor Kingsley terletak di Jl. Veteran.
- Batagor Isan terletak di Jl. Panjunan. Sementara Batagor Abuy terletak di daerah Jl. Lengkong Besar.

8. Martabak Bandung Kota Kembang Bandung juga terkenal dengan *martabak*nya, terutama *martabak manis*nya. Beberapa yang terkenal adalah :
    - *Martabak San Fransisco* di Jl. Burangrang, Jl. Dago
    - *Martabak Good* di Jl. Cibadak
    - *Martabak Intisari* di Jl. Sudirman.
9. Oncom / Tempe Bandung *Oncom dan Tempe* merupakan salah satu makanan khas yang terkenal di Bandung. Selain dibuat keripik Oleh-oleh Bandung yang satu ini tersedia di banyak tempat, terutama keripik oncomnya, daerah sekitar :
    - Jl. Pasteur
    - Jl. Cihampelas
    - Jl. Lewipanjang, juga di beberapa terminal bus, banyak menyediakan keripik oncom dan tempe Bandung.
    - Jl. Cicaheum (Terminal)
10. Peyeum Bandung merupakan salah satu makanan yang cukup terkenal di Bandung. Oleh-oleh Bandung yang satu ini tersedia di toko oleh-oleh di daerah sekitar :
    - Jl. Pasteur
    - Jl. Leuwipanjang

- Jl. Cihampelas
- Jl. Cicaheum (Terminal)

11. Manisan Buah merupakan salah satu makanan khas yang terkenal di Bandung. Selain dibuat keripik Oleh-oleh Bandung manisan buah yang satu ini tersedia di toko oleh-oleh di daerah sekitar :
    - Jl. Pasteur
    - Jl. Cihampelas
    - Jl. Lewipanjang, juga di beberapa terminal banyak menyediakan Manisan Buah Bandung.
    - Jl. Cicaheum (Terminal)
12. Dodol dan Wajit merupakan salah satu makanan khas yang terkenal di Bandung. Dodol dan Wajit yang satu ini tersedia di toko oleh-oleh di daerah sekitar :
    - Jl. Pasteur
    - Jl. Cihampelas
    - Jl. Lewipanjang, juga di beberapa terminal bus, banyak menyediakan Dodol dan Wajit Bandung.
    - Jl. Cicaheum (Terminal)
13. Rangginang merupakan salah satu makanan khas yang terkenal di Bandung. Rangginang yang satu ini tersedia di toko oleh-oleh di daerah sekitar :
    - Jl. Pasteur
    - Jl. Cihampelas
    - Jl. Lewipanjang, juga di beberapa terminal bus, banyak menyediakan Rangginang Bandung.
    - Jl. Cicaheum (Terminal)
14. Tahu Bandung memang terkenal empuk, lembut di mulut dan gurih. Tahu Bandung sendiri banyak dijadikan oleh-oleh, Beberapa Tahu

Bandung yang terkenal adalah Tahu Lembang yang terletak di daerah kawasan Lembang

- Tahu Tauhid terletak di Jl. Seskoau Lembang
- Tahu Talaga/Tahu Yun Sen terletak di Jl. Jendral Sudirman
- Tahu Yun Yi terletak di Jl. Jendral Sudirman, Jl. Kebon Kawung, Jl. Laswi, Jl. Buah Batu, Jl. Raya Timu (Batunuggal).

2.2.1.2 Oleh-oleh Barang *(Fashion)* Bandung

Adapun macam-macam kategori oleh-oleh barang (fashion) di kota bandung yang terkenal yaitu diantaranya seperti :

1. Tas dapat di temukan di Mall, Pasar Tradisional seperti Pasar Baru serta di Outlet-Outlet seperti Terminal Tas Kawasan Daerah Jl. Riau
2. Sepatu dapat di temukan di Mall serta di Outlet-Outlet seperti Kawasan Daerah Jl. Riau dan kawasan khusus seperti di Jl. Cibaduyut.
3. Baju dapat di temukan di Mall, Pasar Tradisional seperti Pasar Baru serta di Outlet-Outlet di Bandung.
4. Jeans dapat di temukan di Mall, Cihampelas, Pasar Tradisional seperti Pasar Baru serta di Outlet-Outlet di Bandung.

2.2.2 Tempat Perbelanjaan Oleh-oleh Banudng

Adapun macam-macam kategori oleh-oleh tempat perbelanjaan di kota bandung yang terkenal yaitu diantaranya seperti :

1. Mall :
    - PVJ (Paris Van Java)
    - TSM (Trans Studio Mall)
    - Cihampelas Walk
    - BIP (Bandung Indah Plasa)
    - BTC (Bandung Trade Center)

- Braga City Walk
- FCL (Festival City Link)
- BEC (Bandung Elektronik Center)
- MIKO Mall
- Paskal 23

2. Pasar Tradisonal :
    - Pasbar (Pasar Baru)
    - ITC (Istana Trade Center)
    - PP (Plasa Parahyangan)

2.3 Navigasi

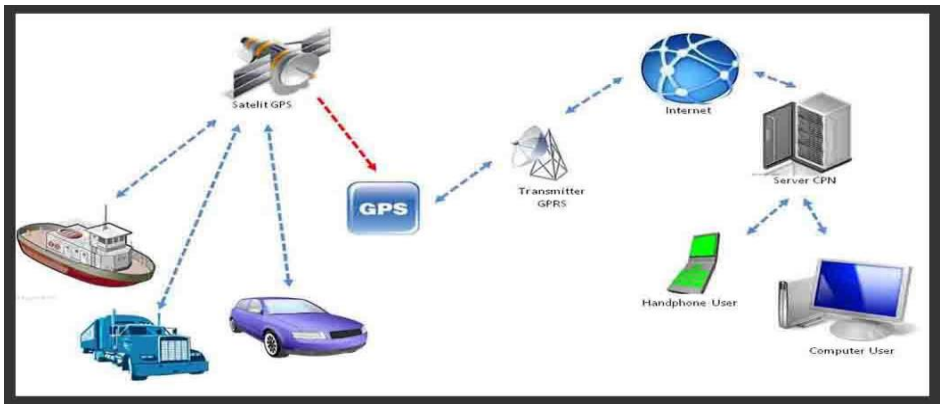
Navigasi berasal dari bahasa Yunani, yaitu dari kata navis yang berarti perahu kapal dan kata agake yang berarti mengarahkan. Arti secara harfiah yaitu mengarahkan sebuah kapal dalam melakukan pelayaran. Pada perkembangan selanjutnya kata navigasi tidak hanya diperuntukkan lagi dalam dunia pelayaran, akan tetapi juga digunakan dalam perjalanan darat (navigasi darat) dan udara (navigasi udara) [14].

Navigasi adalah suatu teknik untuk menentukan kedudukan dan arah lintasan secara tepat dengan menggunakan peralatan navigasi, personil yang menggunakannya biasa disebut navigator.

Istilah navigasi pada umumnya digunakan untuk keperluan pelayaran dan penerbangan, penambahan kata darat pada navigasi lebih ditekankan pada penggunaannya di daratan antara lain meliputi gunung, hutan, lembah, sungai, rawa, pantai dan sebagainya. Navigasi merupakan suatu pengetahuan yang sangat penting dan harus dikuasai oleh orang yang melakukan kegiatan di alam terbuka (out door activities).

2.4 GPS

Populasi aplikasi yang dibuat pada sistem operasi Android semakin meningkat. Salah satunya adalah aplikasi yang memanfaatkan fungsi dari lokasi dengan menggunakan GPS (*Global Positioning System*) atau yang biasa disebut *location based services* (LBS). Bagi para pembaca yang belum memahami pengertian *location based services*, mari simak penjelasan di bawah ini [14].



Gambar 2.1 Cara Kerja GPS

Cara kerja GPS secara logik ada 5 langkah:

1. Memakai perhitungan "triangulation" dari satelit.
2. Untuk perhitungan "triangulation", GPS mengukur jarak menggunakan travel time sinyal radio.
3. Untuk mengukur travel time, GPS memerlukan memerlukan akurasi waktu yang tinggi.
4. Untuk perhitungan jarak, kita harus tahu dengan pasti posisi satelit dan ketingian pada orbitnya.
5. Terakhir harus menggoreksi delay sinyal waktu perjalanan di atmosfer sampai diterima reciever.

GPS dapat melakukan perhitungan dan menentukan posisi *user* dan menampilkan dalam *maps*. Jika sudah dapat menyimpan posisi *user* selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah tuju, rute, tujuan perjalanan serta jarak tujuan. Perhitungan waktu yang akurat sangat menentukan informasi lokasi *user*.

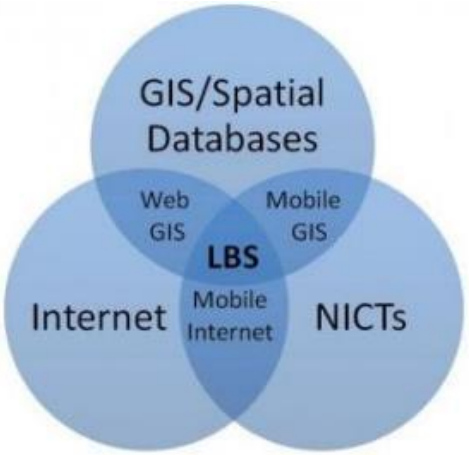
2.4.1 *LBS (Location based services*)

Location based services adalah layanan berbasis lokasi atau istilah umum yang sering digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi g*lobal positioning service* (GPS) dan *cell-based location* dari Google [5].

Selain itu, LBS tersebut terdiri dari beberapa komponen di antaranya *mobile devices, communication network, position component,* dan *service and content provider*. *Mobile devices* merupakan komponen yang sangat penting. Piranti *mobile* tersebut diantaranya adalah *smartphone*, PDA, dan lainnya yang dapat berfungsi sebagai alat navigasi atau seperti halnya alat navigasi berbasis GPS.

Komponen *communication network* ini berupa jaringan telekomunikasi bergerak yang memindahkan data pengguna dari perangkat ke penyedia layanan. *Position component* yang dimaksud adalah posisi pengguna harus ditentukan. Posisi ini dapat didapatkan dengan jaringan telekomunikasi atau dengan GPS. Sedangkan *service and content provider* adalah penyedia layanan yang menyediakan layanan berbeda ke pengguna seperti pencarian rute, kalkulasi posisi, dan lainnya. Sederhananya, dengan layanan LBS kita dapat mengetahui posisi dimana kita berada, posisi teman, dan posisi rumah sakit atau pom bensin yang jaraknya dekat dengan kita. Dalam mengukur posisi, digunakan lintang dan bujur untuk menentukan lokasi geografis tetapi, Android menyediakan *geocoder* yang mendukung *forward* dan *reverse geocoding*. Menggunakan *geocod*, kita dapat mengkonversi nilai lintang bujur menjadi alamat dunia nyata atau sebaliknya. LBS

dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu, *Geographic Information System*, *Internet Service*, dan *Mobile Devices*. Hal ini dapat dilihat pada gambar dibawah ini:



Gambar 2.2 LBS Sebagai Simpang Tiga

2.4.2 Unsur Utama LBS

LBS atau layanan berbasis lokasi adalah istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan. Dua unsur utama dari Location Based Services (LBS) adalah :

1. Location Manager (API Maps): Menyediakan perangkat bagi sumber atau *source* untuk LBS.API (*Application Programming Interface*)Maps menyediakan fasilitas untuk menampilkan atau memanipulasi peta. Paket ini berada pada "com.goole.android.maps;".
2. Location Providers (API Location): Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. PAI Location berhubungan dengan dara GPS (*Global Positioning System*) dan data lokasi *real-time*. API Location berada pada

paket Android yaitu dalam paket “android.location”. Lokasi, perpindahan, serta kedekatan dengan lokai tertentu dapat ditentukan melalui *Location Manager*.

2.4.3 Komponen Utama LBS

Terdapat lima komponen pendukung utama dalam teknologi layanan berbasis lokasi, antara lain:



Gambar 2.3 Komponen LBS

1. Piranti *Mobile*, adalah satu komponen penting dalam LBS. Piranti ini berfungsi sebagai alat bantu bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebgainya. Piranti mobile yang dapat digunakan bisa berupa PDA, *smartphone*, laptop Selain itu, piranti *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya navigasi berbasis GPS.
2. Jaringan Komunikasi, komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data oleh pengguna dari piranti mobile-nya untuk kemudian ditujukan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

3. Komponen *Positioning* (penujuk posisi/lokasi), setiap layanan yang di berikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah/pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi mobile atau juga menggunakan Global Positioning System (GPS).
4. Penyedia layanan dan aplikasi, merupakankomponen LBS yang memberikan berbagai macam layanan yang bsia digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menetukan posisi pengguna, menentukan rute jalan, mencari Yellow Pages sesuai dengan permintaan, dan masih banyak lagi yang lainnya.
5. Penyedia data adan konten, penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang atau pihak ketiga yang memang memiliki otoritas untuk menyimpannya.

2.5 Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer,instruksi (*instructiom*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output*. Menurut Kamus Kamus Besar Bahasa Indonesia, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini [6].

Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instructionset*. Program aplikasi merupakan program siap pakai yang dirancang untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh-contoh aplikasi ialah program pemproses kata dan *Web Browser*. Aplikasi akan menggunakan *system* operasi (OS) komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app.

Secara historis, aplikasi adalah software yang dikembangkan oleh sebuah perusahaan. App adalah sofware yang dibeli perusahaan dari tempat pembuatnya. Industri PC tampaknya menciptakan istilah ini untuk merefleksikan medan pertempuran persaingan yang baru, yang parallel dengan yang terjadi antar sistem operasi yang dimunculkan.

2.5.1 Klasifikasi Aplikasi

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain:

1. Perangkat lunak perusahaan (enterprise)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk

2.6 Sistem

Sistem adalah seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan untuk mencapai tujuan.

2.6.1 Syarat Sistem

1. Sistem harus dibentuk untuk menyelesaikan masalah.
2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan diantara elemen sistem.
4. Unsur dasar dari proses lebih penting dari pada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen

2.6.2 Klasifikasi Sistem

Suatu sistem dapat diklasifikasikan sebagai berikut :

1. Sistem abstrak

   Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (sistem teologia).
2. Sistem fisik

   Merupakan sistem yang ada secara fisik (sistem komputer, sistem akuntansi, sistem produksi)
3. Sistem alamiah

   Sistem yang terjadi melalui proses alam. (sistem matahari, sistem luar angkasa, sistem reproduksi)
4. Sistem Buatan Manusia

   Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut human-machine system contoh : sistem informasi.
5. Sistem Tertentu (*deterministic system*)

Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan (contoh : sistem komputer)

6. Sistem Tak Tentu (*probabilistic system*)
   Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
7. Sistem tertutup (*close system*)
   Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* atau secara relatif tertutup, tidak benar-benar tertutup .
8. Sistem terbuka (*open system*)
   Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.
9. Sistem sederhana dan Sistem kompleks

2.6.3 Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, yaitu: tujuan, masukan, proses, keluaran, batas, mekanisme pengendalian dan umpan balik serta lingkungan. Berikut penjelasan mengenai elemen-elemen yang membentuk sebuah sistem:

1. Tujuan (*Goal*)
   Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.
2. Masukan (*Input*)

Masukan(input) sistem yaitu segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa hal-hal yang berwujud atau tampak secara fisik maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi, misalnya permintaan jasa pelanggan.

3. Proses

   Proses merupakan bagian yang melakukan perubahan atau ransformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

4. Keluaran (*output*)

   Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistemnformasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dansebagainya.

5. Batas(*Boundary*)

   Batas sistem adalah pemisah antara sistem dan daerah di luar sistem atau lingkungan. Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh, tim sepak bola mempunyai aturan permainan dan keterbatasan kemampuan pemain.

6. Mekanisme Pengendalian dan Umpan Balik

   Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*). Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

7. Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. Lingkungan yang merugikan tentu saja harus ditahan dan dikendalikan agar tidak mengganggu kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus terus dijaga, karena akan memacu terhadap kelangsungan hidup sistem.

2.7 Informasi

Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai yang nyata yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan - keputusan yang akan datang, sedangkan menurut *Burch* dan *Strater*, informasi adalah pengumpulan atau pengolahan data untuk memberikan pengetahuan atau keterangan.

Jadi, secara umum informasi adalah data yang sudah diolah menjadi suatu bentuk lain yang lebih berguna yaitu pengetahuan atau keterangan yang ditujukan bagi penerima dalam pengambilan keputusan, baik masa sekarang atau yang akan datang. Informasi yang dihasilkan dari aplikasi yang dibuat adalah navigasi rute terpendek dan tercepat dalam menentukan tujuan untuk dapat dipilih oleh pengguna.

2.7.1 Kegunaan Informasi

Ada 4 (empat) factor utama yang berhubungan dengan kegunaan informasi yaitu :

1. Kualitas informasi (*information quality*) Kualitas dari suatu informasi tergantung dari 4 hal informasi yaitu :
   a. Akurat (*accurate*) dan presisi (*precision*)
      Akurat dalam menampilkan informasi dan presisi dalam detail informasi yang diberikan.
   b. Kelengkapan (*completeness*)
      Informasi yang tersedia cukup lengkap untuk setiap user dan situasi.

c. Umur (*age*) dan ketepatan waktu (*timeliness*)

Umur berarti lamanya waktu dalam meng-update informasi dan ketepatan waktu berarti menyediakan informasi secepat mungkin pada saat dibutuhkan sehingga berguna.

d. Sumber (*source*)

Orang atau organisasi yang menghasilkan informasi.

2. Aksesibilitas informasi (*information accessibility*)
   a. Ketersediaan (*availability*)

   Memberikan informasi kepada yang membutuhkan. Informasi dapat diakses oleh yang membutuhkan.

   b. Keabsahan (*admissibility*)

   Keabsahan (boleh atau tidak boleh dipakai) informasi tergantung pada hukum, peraturan atau budaya pada saat tertentu.
3. Presentasi informasi (*information presentation*)
   a. Tingkatan (*level of summarization*)

   Perbandingan antara data asli dengan yang ditampilkan. Manipulasi data hingga tingkatan yang sesuai, semakin sederhana semakin baik.

   b. Format

   Bentuk dimana informasi ditampilkan ke *user*. Manipulasi data dalam bentuk yang sesuai.
4. Keamanan informasi (*information security*)
   a. Batasan akses (*access restriction*) Prosedur dan teknik mengontrol *user* yang boleh atau tidak mengakses data pada situasi tertentu. Penggunaan password atau teknik lain untuk mencegah *user* yang tidak berhak.
   b. Enkripsi (*encryption*) Konversi data ke bentuk tertentu sehingga tidak dapat dibaca oleh *user* yang tidak berhak.

2.7.2 Nilai Informasi

Nilai dari informasi (*value of information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Sebagian besar informasi tidak dapat persis ditaksir keuntungannya dengan satuan nilai uang, tetapi dapat ditaksir nilai efektivitasnya. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *costeffectiveness* atau *cost benefit*.

2.8 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer *tablet*. Antarmuka pengguna Android umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yanag serupa dengan tindakan nyata, misalnya menggeser, mengetuk untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan untuk digunakan oleh bermacam peranti bergerak (*mobile device*). Hal ini memungkinkan para pengembang menulis kode terkelola (*managed code*) dalam bahasa pemrograman Java, mengontrol peranti via perpustakaan Java yang dikembangkan Google [9].

Pada saat perilisan perdananya, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan akan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode – kode Android di bawah lisensi Apache, sebuah lisensi piranti lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services* (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution* (OHD).

2.8.1 Android Life Cycle

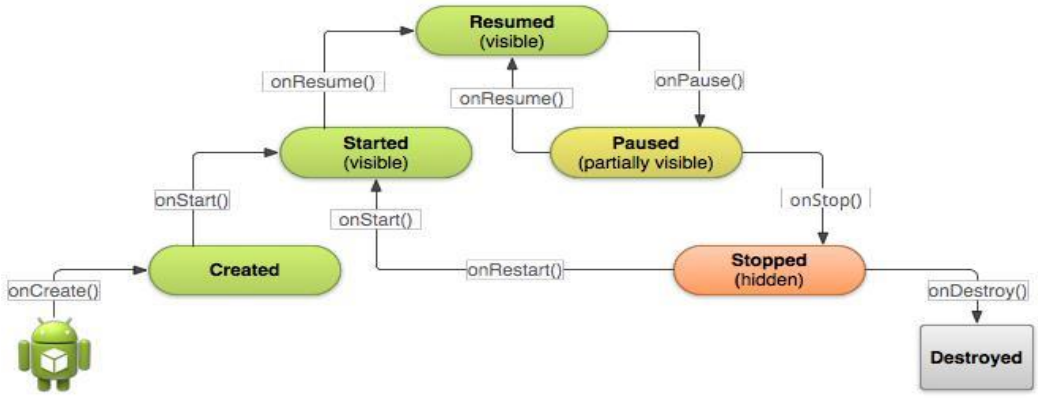
Aplikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang berbeda seperti ditunjukkan pada , klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan [9].

Tabel 2.1 Komponen Aplikasi Android

Functionality	***Java Base Class***	***Examples***
Focused thing a user can do	*Activity*	*Edit a note, play a game*
Background process	*Service*	*Play music, update weather icon*
Receive messages	*BroadcastReceiver*	*Trigger alarm upon event*
Store and retrieve data	*ContentProvider*	*Open a phone contact*

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan specified permission sebelum digunakan seperti komponen *Service, Broadcast Receiver, Content Provider.*

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 h*andling* panggilan masuk aplikasi diharuskan menghentikan proses sementara, penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai pengguna.



Gambar 2.4 Siklus Hidup Android

Terdapat beberapa state dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gambar 2.1 akan tetapi hanya beberapa dari state tersebut yang menjadi statis diantaranya :

- ***Resumed***

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

- ***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.

- ***Stopped***

  Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan *service* di *background*. 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*().

2.8.2 Fitur Android

Android memiliki beberapa fitur utama yang sering digunakan dalam proses pembangunan aplikasi diantaranya adalah [9]:

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 *keyboard* virtual, dan tombol navigasi aplikasi.

2.8.3 Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework) [25].

Visual Studio kini telah menginjak versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19 November2007, yang ditujukan untuk platform Microsoft .NET Framework 3.5. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform .NET Framework 2.0 dan 3.0. Visual Studio 2003 ditujukan untuk .NET Framework 1.1, dan Visual Studio 2002 ditujukan untuk .NET Framework 1.0. Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio .NET, karena memang membutuhkan Microsoft .NET Framework. Sementara itu, sebelum muncul Visual Studio .NET, terdapat Microsoft Visual Studio 6.0 (VS1998).

2.9 C#

C# atau yang dibaca C sharp adalah bahasa pemrograman sederhana yang digunakan untuk tujuan umum, dalam artian bahasa pemrograman ini dapat digunakan untuk berbagai fungsi misalnya untuk pemrograman server-side pada website, membangun aplikasi desktop ataupun mobile, pemrograman game dan sebagainya. Selain itu C# juga bahasa pemrograman yang berorientasi objek, jadi C# juga mengusung konsep objek seperti *inheritance, class, polymorphism*dan *encapsulation*. Dalam prakteknya C# sangat bergantung dengan framework yang disebut .NET Framework, framework inilah yang nanti digunakan untuk mengcompile dan menjalankan kode C#. C# dikembangkan oleh Microsoft dengan merekrut Anders Helsberg. Tujuan dibangunnya C# adalah sebagai bahasa pemrograman utama dalam lingkungan .NET Framework (lihat C#). Banyak pihak juga yang menganggap bahwa Java dengan C# saling bersaing, bahkan ada juga yang menyatakan jika pernah belajar Java maka belajar C# akan sangat mudah dan begitu juga sebaliknya. Anggapan tersebut sebenarnya tidak salah karena perlu diketahui sebelum adanya C# Microsoft mengembangkan J++ dengan maksud mencoba membuat Java agar berjalan pada platform Windows, karena adanya masalah dari pihak luar maka Microsoft menghentikan proyek J++ dan beralih untuk mengembangkan bahasa baru yaitu C# [23].

2.10 Xamarin

Xamarin adalah perusahaan software yang terletak di San Francisco, california dan berdiri semenjak Mei 2011. Engginer yang bergerak di perusahaan ini adalah yang membuat Mono, MonoTouch dan Mono for Android yang mana cross-platform dan dapat di implementasikan ke Common Language Infrastructure (CLI) dan Common Language Specifications (atau yang biasa disebut Microsoft .NET). Dengan based nya adalah bahasa pemograman C# dan mengimplemetasikan metode shared codebase, developers bisa menggunakan Xamarin untuk menggembangkan aplikasi native iOS,

Android, and Windows dengan native user interfaces dan share code across multiple platforms [22].

Berdasarkan sumber dari website resmi xamarin, didefinisikan bahwa xamarin digunakan untuk pengembangan aplikasi mobile. Dengan menggunakan xamarin, user dapat mengembangkan aplikasi mobile pada 3 sistem operasi yang berbeda dengan menggunakan 1 kode inti. Xamarin menggunakan bahasa pemograman C# atau kombinasi dengan java untuk beberapa sistem operasi tertentu, sehingga dapat menghemat waktu dan SDM(Sumber Daya Manusia). Kemudian dikombinasikan dengan user interface yang menggunakan bahasa pemograman sesuai dengan system operasi yang dituju, setelah itu semua akan diproses oleh xamarin untuk menjadi aplikasi mobile dengan masing-masing format sesuai dengan system operasi yang diinginkan.

Xamarin adalah unik karena menggabungkan semua kekuatan platform dan menambahkan sejumlah fitur canggih sendiri, yaitu:

a. Complete Binding for the Indigenous SDKs

   Xamarin mengandung binding untuk hampir seluruh SDK platform yang mendasari baik iOS dan android. Selain itu, binding ini kuat dalam implementasinya, yang berarti mudah dinavigasi dan digunakan, dan menyediakan waktu kompilasi memeriksa jenis

   code selama pengembangan. Hal ini menyebabkan kesalahan runtime yang lebih sedikit dan kualitas yang lebih tinggi dari aplikasi yang dibuat.

b. Objective-C, Java, C, and C++ Interop

   Xamarin menyediakan fasilitas untuk langsung memanggil Objective-C, Java, C, dan C++ library, dan memberikan user kesempatan untuk menggunakan beragam code. Hal ini memungkinkan user mengambil keuntungan dari iOS yang ada dan library android ditulis di dalam Objective-C, Java atau C/C++. Selain itu, xamarin menawarkan binding

project yang memungkinkan user untuk dengan mudah mengaitkan native Objective-C dan Java library menggunakan sintaks deklaratif.

c. Modern Language Constructs

Aplikasi xamarin ditulis dalam C #, bahasa modern yang mencakup perbaikan yang signifikan atas Objective-C dan Java seperti fitur Dynamic Language, konstruktur fungsional seperti lambdas, LINQ, fitur pemrograman paralel, generik canggih, dan banyak lagi.

d. Amazing Base Class Library (BCL)

Aplikasi xamarin menggunakan. NET BCL, koleksi besar dari kelas yang memiliki fitur yang lengkap dan efisien seperti XML yang kuat, database, serialisasi, IO, string, dan jaringan dukungan, hanya untuk beberapa nama. Selain itu, C # kode yang ada dapat dikompilasi untuk digunakan dalam aplikasi user, yang menyediakan akses ke ribuan library yang akan membiarkan user melakukan hal-hal yang belum tercakup dalam BCL.

e. Modern Integrated Development Environment (IDE)

Xamarin menggunakan Xamarin Studio di Mac OSX, dan juga Xamarin Studio atau Visual Studio 2010 pada Windows. Ini adalah kedua IDE modern yang mencakup fitur seperti kode auto completion, Proyek canggih dan sistem manajemen Solution,sebuah project template library yang komprehensif, kontrol sumber terintegrasi, dan banyak lainnya.

f. Mobile Cross Platform Support

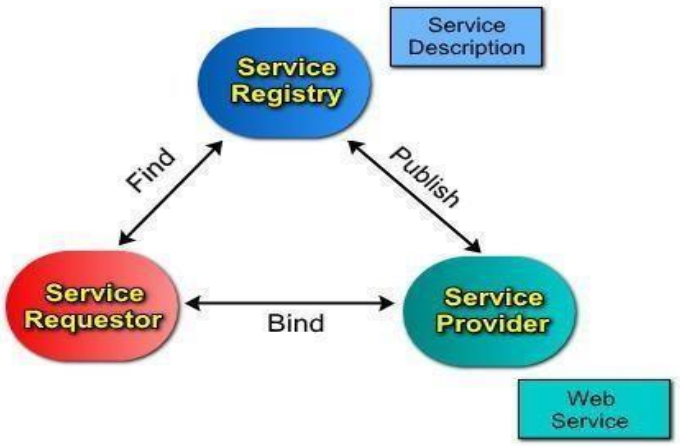
Xamarin menawarkan dukungan cross-platform canggih untuk tiga platform mobile utama iOS, Android, dan Windows Phone. Aplikasi dapat ditulis untuk berbagi hingga 90% dari kode, dan library Xamarin. Xamarin menyediakan API untuk mengakses sumber daya umum di dalam semua tiga platform. Hal ini dapat secara signifikan mengurangi baik biaya pengembangan dan waktu, untuk mobile pengembangan ditargetkan tiga platform mobile yang paling populer. Karena sangat berguna dan

komprehensif, dapat mengisi kemudahan untuk pengembang aplikasi yang ingin menggunakan bahasa modern dan platform untuk mengembangkan aplikasi mobile cross platform.

2.11 Web Service

Web service adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan. *Web service* memiliki *interface* yang dideskripsikan dalam format yang dapat dibaca oleh mesin. Sistem-sistem lainnya berinteraksi dengan web service menggunakan pesan SOAP yang umumnya dikirim melalui HTTP dalam bentuk XML [19].

Definisi diatas diberikan oleh *World Wide Web Consortium*(W3C) yang merupakan badan yang menciptakan dan mengembangkan standar *web service*. Tetapi secara umum, *web service* tidak terbatas hanya pada standar SOAP saja. Salah satu pustaka yang mengulas lengkap tentang *web service* menyebutkan definisi yang lebih umum: *web service* adalah aplikasi yang diakses melalui internet menggunakan protokol standar internet dan menggunakan XML sebagai format pesannya.



Gambar 2.5 Arsitektur Web Service

Pada gambar diatas, ada tiga komponen yang membuat *web service*

berjalan.

Ketiga komponen itu adalah [12]:

1. *Service provider*, merupakan pemilik *web service* yang berfungsi menyediakan kumpulan operasi dari *web service*.
2. *Service requestor*, merupakan aplikasi yang bertindak sebagai klien dari *web service* yang mencari dan memulai interaksi terhadap layanan yang disediakan.
3. *Service registry*, merupakan tempat dimana *service provider* mempublikasikan layanannya. Pada arsitektur *web service*, *service registry* bersifat optional. Teknologi *web service* memungkinkan kita dapat menghubungkan berbagai jenis software yang memiliki platform dan sistem operasi yang berbeda.

2.12 Database

Database atau basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur, dan juga batasan-batasan data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghidari duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga update yang rumit [16].

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (*database management system* | DBMS). DBMS merupakan sistem perangkat lunak yang memungkinkan user untuk memelihara, mengontrol, dan mengakses data secara praktis dan efisien. Dengan kata lain semua akses ke basis data akan ditangani oleh DBMS. Ada beberapa fungsi yang harus ditangani DBMS yaitu mengolah

pendefinisian data, dapat menangani permintaan pemakai untuk mengakses data, memeriksa sekuriti dan integriti data yang didefinisikan oleh DBA (*Database Administrator*), menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun disk, dan menangani unjuk kerja semua fungsi secara efisien.

Tujuan utama dari DBMS adalah untuk memberikan tinjauan abstrak data kepada user (pengguna). Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara, dan tetap dapat diambil (akses) secara efisien. Pertimbangan efisien disini adalah bagaimana merancang struktur data yang kompleks tetapi masih tetap bisa digunakan oleh pengguna awam tanpa mengetahui kompleksitas strukturnya. Database dibagi menjadi dua jenis yaitu :

1. Basis Data Flat File

Basis data flat-file ideal untuk data berukuran kecil dan dapat dirubah dengan mudah. Pada dasarnya, mereka tersusun dari sekumpulan string dalam satu atau lebih file yang dapat diurai untuk mendapatkan informasi yang disimpan. Basis data flat-file baik digunakan untuk menyimpan daftar atau data yang sederhana dan dalam jumlah kecil. Basis data flat-file akan menjadi sangat rumit apabila digunakan untuk menyimpan data dengan struktur kompleks walaupun dimungkinkan pula untuk menyimpan data semacam itu. Salah satu masalah menggunakan basis data jenis ini adalah rentan pada korupsi data karena tidak adanya penguncian yang melekat ketika data digunakan atau dimodifikasi.

2. Basis Data Relasional

Basis data ini mempunyai struktur yang lebih logis terkait cara penyimpanan. Kata "relasional" berasal dari kenyataan bahwa tabel-tabel yang berada di basis data dapat dihubungkan satu dengan lainnya. Basis data relasional menggunakan sekumpulan tabel dua dimensi yang masing-masing tabel tersusun atas baris (*tupel*) dan kolom (atribut). Untuk membuat hubungan antara dua atau lebih tabel, digunakan key (atribut kunci) yaitu *primary key* di salah satu tabel dan *foreign key* di tabel yang lain. Saat ini,

basis data relasional menjadi pilihan karena keunggulannya. Beberapa kelemahan yang mungkin dirasakan untuk basis data jenis ini adalah implementasi yang lebih sulit untuk data dalam jumlah besar dengan tingkat kompleksitasnya yang tinggi dan proses pencarian informasi yang lebih lambat karena perlu menghubungkan tabel-tabel terlebih dahulu apabila datanya tersebar di beberapa tabel.

2.13 XAMPP

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU General Public License dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Jika ingin mendapatkan xampp dapat mendownload langsung dari situs resminya [16].

1. Bagian penting XAMPP

Memahami XAMPP :

- htdoc adalah folder di mana Anda meletakkan file yang akan dijalankan, seperti file PHP, HTML dan script lainnya.
- phpMyAdmin adalah bagian untuk mengelola database MySQL yang dikomputer.
- Untuk membukanya, membuka browser dan ketik alamat http: // localhost / phpMyAdmin, halaman phpMyAdmin akan muncul.
- Control Panel yang berfungsi untuk mengelola layanan (service) XAMPP. Seperti stop service (berhenti), atau mulai (mulai).

2. Komponen Xampp

XAMPP 1.8.3 untuk Windows, Termasuk :

- Apache 2.4.4
- MySQL 6.5.11
- PHP 5.5.0
- phpMyAdmin 4.0.4
- FileZilla FTP Server 0.9.41
- Tomcat 7.0.41 (with mod_proxy_ajp as connector)
- Strawberry Perl 5.16.3.1 Portabel
- XAMPP Control Panel 3.2.1 (dari hackattack142)

2.14 HTTPS (Hypertext Transfer Protocol)

HTTPS (*Hypertext Transfer Protocol Secure*) adalah protokol komunikasi internet yang melindungi integritas dan kerahasiaan data pengguna antara komputer pengguna dan situs. Data yang dikirim menggunakan HTTPS diamankan melalui protokol *Transport Layer Security* (TLS), yang memberikan tiga lapis perlindungan kunci [15] :

1. Enkripsi: mengenkripsi data pertukaran untuk menjaga keamanannya dari penyadap. Artinya, saat pengguna menjelajahi situs web, tidak ada yang dapat "menguping" percakapan, melacak aktivitas di berbagai laman, atau mencuri informasi mereka.
2. Integritas data: data tidak dapat diubah atau dirusak selama transfer, dengan sengaja atau tidak, tanpa terdeteksi.
3. Autentikasi: membuktikan bahwa pengguna Anda berkomunikasi dengan situs web yang diinginkan. Hal tersebut melindungi dari serangan *Man In The Middle* (MITM) dan membangun kepercayaan pengguna, yang dapat memberikan keuntungan lain untuk bisnis Anda.

2.14.1 Fungsi dan Tujuan HTTPS

Fungsi khusus HTTPS adalah untuk menjaga kemanan data dari hacker yang berniat membajak sebuah dokumen secara ilegal. Sedangkan tujuan HTTPS adalah untuk mengantisipasi terjadinya eror program sebuah data dan kerusakan server akibat kegagalan koneksi yang disebabkan perusahaan yang di buat oleh pakar hacker yang tidak bertanggung jawab.

2.14.2 Kelebihan dan Kelemahan HTTPS

a. Kelebihan HTTPS

Beberapa kelebihan HTTPS yang membuatnya masih digunakan hingga saat ini :

1. Jalur yang digunakan untuk transfer data sangat aman karena terenkripsi.
2. Sertifikat yang didapatkan dari badan pemberi sertifikat dijamin keamanannya oleh badan tersebut.
3. Dapat menangkal serangan *eavesdroppers,* dan *man in the middle attacks*.
4. Banyak situs-situs besar yang menggunakan metode ini, sehingga komunitas pengguna HTTPS cukup banyak.

b. Kelemahan HTTPS

Beberapa kelemahan yang dimiliki dari menggunakan metode HTTPS, adalah sebagai berikut :

1. Tidak semua *web browser* mempunyai kemampuan untuk menginstall sertifikat dengan baik. Oleh karena itu sertifikat harus mempunyai *browser recognition rate*.
2. Kerja komputer semakin tinggi karena melibatkan enkripsi, maka *client* dan *server* membutuhkan waktu, *resource*, dan *bandwidth* lebih untuk melakukan kegiatannya.

3. Kebutuhan *bandwidth* bertambah karena data yang dikirim lebih besar daripada data yang tidak dienkripsi.
4. Proses mendapatkan sertifikat yang cukup rumit, sehingga membuat para pembuat situs individu sulit untuk membuat sertifikat yang sesuai dengan kerja dari situsnya.

2.15 Google Maps API

Google Maps API adalah sebuah layanan yang memungkinkan Google Maps agar dapat diintegrasikan ke dalam aplikasi yang memang membutuhkan fitur dari Google Maps ini. Google Maps API adalah sebuah *library Javascript*. Dengan menggunakan Google Maps API ini dapat memudahkan pengembang dalam membuat aplikasi yang membutuhkan teknologi peta di dalamnya dan tidak perlu memikirkan cara membuatnya secara manual karena dengan mengunakan Google Maps API ini sudah semua yang dibutuhkan terkait teknologi peta digital sudah cukup lengkap dan dapat dimanfaatkan untuk aplikasi yang membutuhkannya [20].

Saat ini Google Maps API telah mengalami perkembangan, hingga saat ini Google Maps API telah mencapai versi 3. Namun, Google Maps API versi 3 ini masih hanya bisa digunakan oleh pemrograman berbasis web dengan menggunakan Javascript. Untuk penggunaan Google Maps API pada perangkat mobile seperti misalnya android Google Maps API yang bisa digunakan saat ini telah mencapai versi 2.

Perkembangan pada Google Maps Versi 2 ini adalah dengan memiliki fitur–fitur yang lengkap dari versi sebelumnya salah satunya adalah dapat menampilkan peta dengan sudut yang di inginkan dan berupa 3D view. Selain itu pada Google Maps API terdapat beberapa pilihan model peta yang ditampilkan seperti Roadmap, Satellite, Terrain dan Hybrid.

2.16 Google Maps Distance Matrix API

Google Maps Distance Matrix API Adalah layanan yang menyediakan waktu dan jarak perjalanan untuk matriks tempat asal dan tujuan, berdasarkan rute yang direkomendasikan di antara titik awal dan akhir [20].

Dokumen ini ditujukan bagi developer yang ingin menghitung jarak dan waktu tempuh perjalanan antara sejumlah titik dalam peta yang disediakan oleh salah satu Google Maps API. Dokumen ini menyediakan pengantar untuk menggunakan API dan materi referensi mengenai parameter yang tersedia. Google Maps Distance Matrix API mengembalikan informasi berdasarkan rute yang disarankan antara titik awal dan titik akhir, sesuai perhitungan Google Maps API, dan terdiri dari baris-baris berisi nilai duration dan distance untuk setiap pasangan

a. Parameter yang diperlukan

- origins — Titik awal untuk menghitung jarak dan waktu perjalanan. Anda bisa memberikan satu atau beberapa lokasi yang dipisahkan dengan karakter pipa (|), dalam bentuk alamat, koordinat garis lintang/bujur, atau ID tempat:
- Jika Anda meneruskan sebuah alamat, layanan akan melakukan geocode atas string itu dan mengubahnya menjadi koordinat garis lintang/bujur untuk menghitung jarak. Koordinat ini mungkin berbeda dengan yang dikembalikan oleh Google Maps Geocoding API, misalnya pintu masuk bangunan dan bukan pusatnya.

  origins=Bobcaygeon+ON|24+Sussex+Drive+Ottawa+ON

- Jika Anda meneruskan koordinat garis lintang/bujur, koordinat akan digunakan tanpa diubah untuk menghitung jarak. Pastikan tidak ada spasi di antara nilai garis lintang dan garis bujur.

  origins=41.43206,-81.38992|-33.86748,151.20699

- Jika Anda memberikan ID tempat, Anda harus mengawalinya dengan place_id. Anda hanya bisa menetapkan ID tempat jika permintaan menyertakan kunci API atau ID klien Google Maps APIs Premium Plan. Anda bisa mengambil ID tempat dari Google Maps Geocoding API dan Google Places API (termasuk Place Autocomplete). Untuk contoh menggunakan ID tempat dari Place Autocomplete, lihat Place Autocomplete dan Directions. Untuk informasi selengkapnya tentang ID tempat, lihat ringkasan ID tempat.

  origins=place_id:ChIJ3S-JXmauEmsRUcIaWtf4MzE

- destinations Satu atau beberapa lokasi untuk digunakan sebagai titik akhir untuk menghitung jarak dan waktu perjalanan. Opsi untuk parameter destinations sama dengan opsi untuk parameter originsyang dijelaskan di atas.

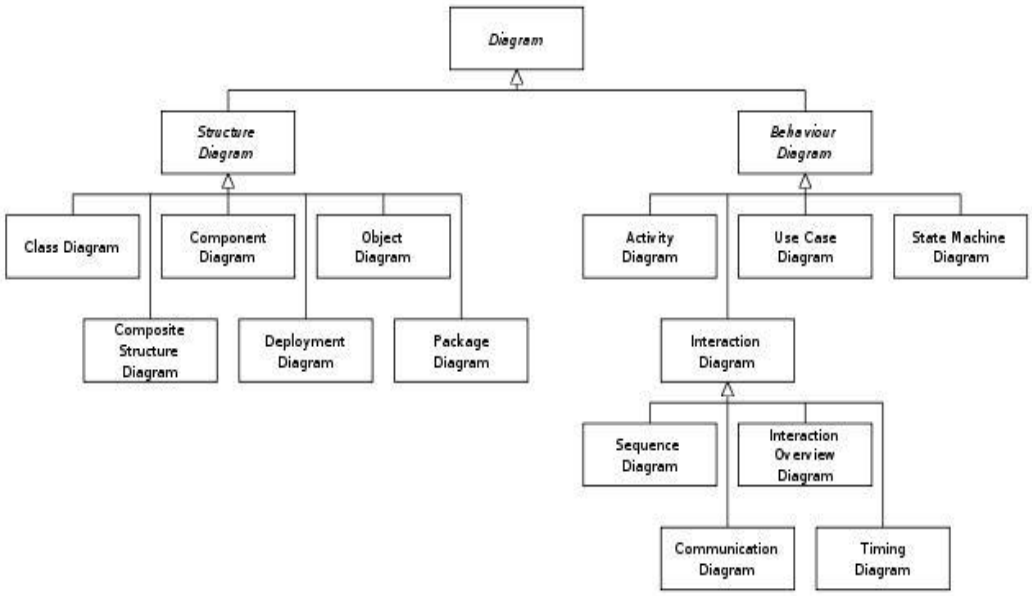

```
{
   "destination_addresses" : [ "New York, NY, USA" ],
   "origin_addresses" : [ "Washington, DC, USA" ],
   "rows" : [
      {
         "elements" : [
            {
               "distance" : {
                  "text" : "225 mi",
                  "value" : 361715
               },
               "duration" : {
                  "text" : "3 hours 49 mins",
                  "value" : 13725
               },
               "status" : "OK"
            }
         ]
      }
   ],
   "status" : "OK"
}
```


Gambar 2.6 Contoh Respons Google Maps Distance Matrix API

2.17 UML (Unifed Modeling Language)

Unified Modeling Language (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh meta-model tunggal, Pemodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan *object-oriented* (OOP).UML merupakan sintaks umum untuk membuat model logika dari suatu sistem dan digunakan untuk menggambarkan sistem agar dapat dipahami sela fase analisis dan desain. UML merupakan suatu bahasa pemodelan yang terdiri banyak model diantaranya adalah [17] :

Diagram UML dikelompokan menjadi 2 yaitu Diagram Struktur atau *Statis Diagram* dan Diagram Prilaku Sistem atau *Behaviour Diagram*.

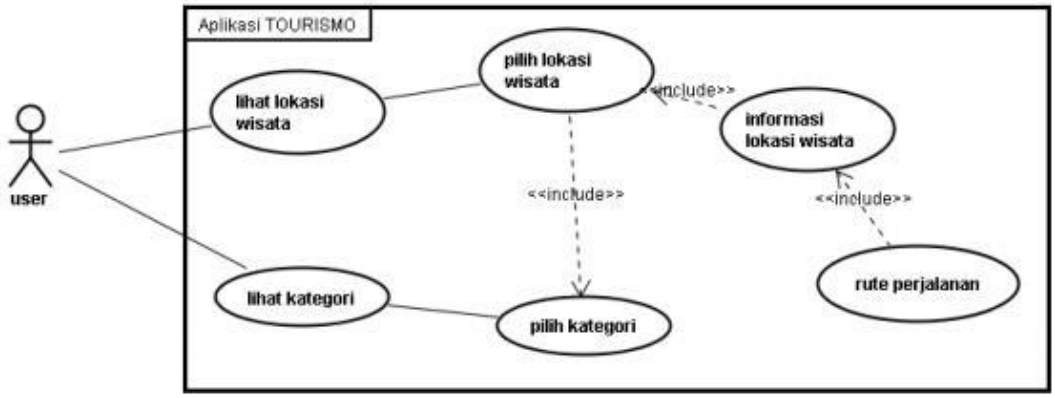


Gambar 2.7 Diagram UML

UML terdiri dari bermacam-macam diagram yang digunakan untuk pemodelan dalam pengembangan sistem mulai dari tahap analisi sampai implementasi. Pada saat melakukan desain sistem tidak harus semua diagram pada UML diimplementasikan akan tetapi perlu adanya kekonsistenan rancangan diagram yang satu dengan lainya.

Pada penelitian ini diagram yang digunakan yaitu *Use case, Class, Sequence dan Activity*.

1. **Diagram *Use Case* (*Use Case Diagram*)**



Gambar 2.8 Contoh *Use Case Diagram*

Use case diagram merupakan suatu pemodelan yang menitik beratkan pada fungsionalitas utama dalam suatu konteks sistem. *Use case* diagram juga menekankan tingkah laku fungsional utama dalam sistem berinteraksi dengan objek diluar sistem tersebut. Selain itu, *use case* diagram juga telah menitik beratkan jenis hubungan diantara fungsi utama. Adapun komponen-komponen dalam *use case* diagram antaranya.

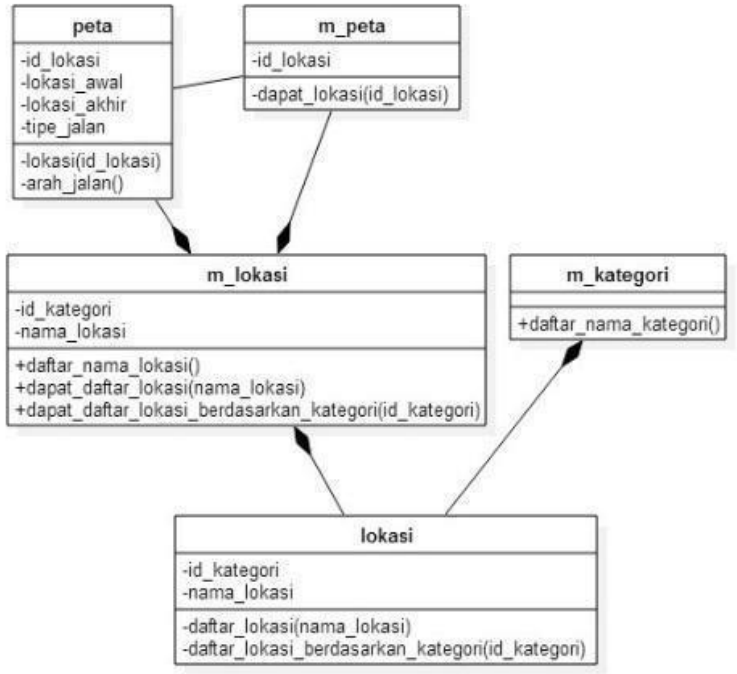
a. Aktor

Aktor merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. Aktor berada diluar sistem namun berkaitan erat dengan fungsionalitas didalamnya. Aktor dapat memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas utama. Aktor juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. Aktor biasanya dapat berupa pengguna atau database yang secara pandang berada dalam suatu ruang lingkup sistem tersebut.

b. *Use Case*

Use case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada use case, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. Use case menitik beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem.

2. **Diagram Kelas (*Class Diagram*)**

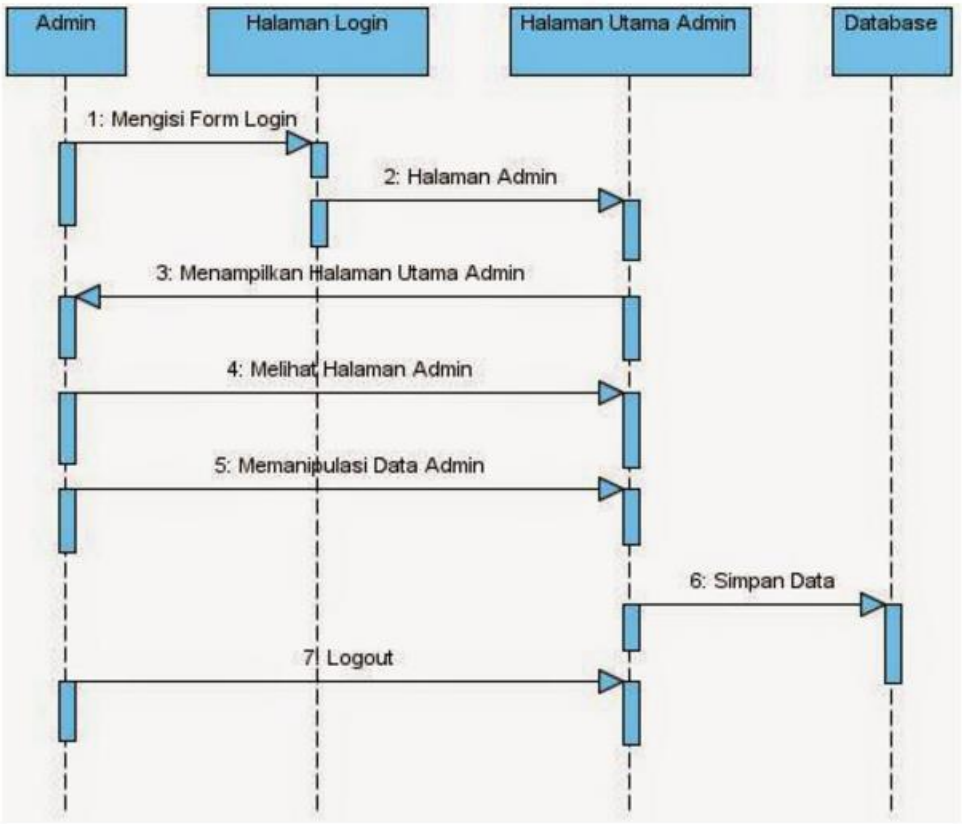


Gambar 2.9 Contoh *Class Diagram*

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. Kelas pada kelas diagram

terdiri dari 3 bagian utama yaitu nama kelas, isi properti dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif, dependensi, agregasi, komposisi, spesifikasi dan generalisasi. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing-masing komponen penyusun kelas memiliki hak akses seperti *public, private*, dan *protected.*

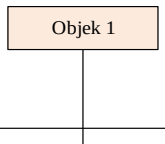
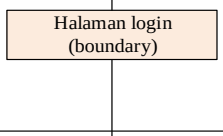
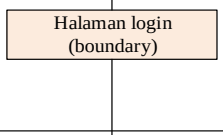
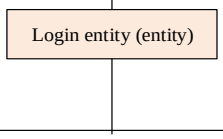
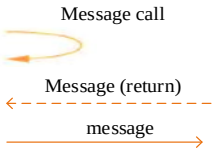
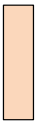
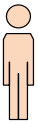
3. Diagram Sekuensial (*Sequence Diagram)*



Gambar 2.10 Contoh *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.

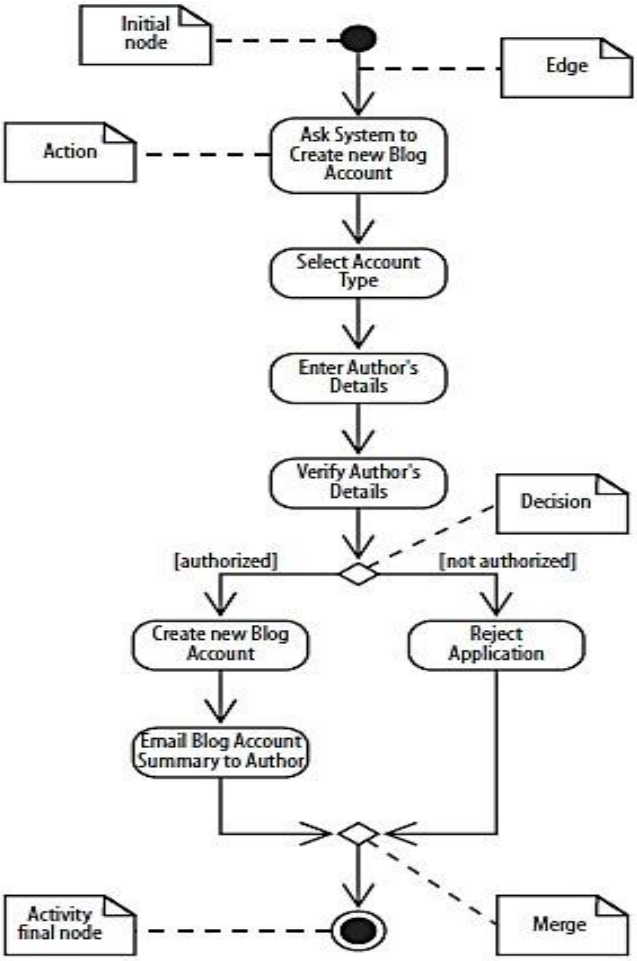
Tabel 2.2 Elemen-elemen Sequence Diagram

No	Nama	Penjelasan	Gambar
1	Objek Lifeline	Menggambarkan batasan objek	
2	Boundary	Berhubungan dengan proses input output interface	
3	Controller	Berhubungan dengan proses	
4	Entity	Berhubungan dengan input – output data	
5	Message Arrow	Menggambarkan alir proses, perintah atau pengiriman data	
6	Aktivasi	Menggambarkan aktivasi objek	
7	Aktor	Menggambarkan aktor suatu objek	

4. Aktiviti Diagram *(Activity Diagram)*

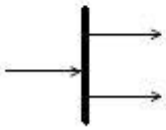
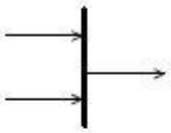
Diagram aktivitas adalah diagram *flowchart* yang diperluas yang menunjukan aliran kendali satu aktivitas ke aktivitas lain di sistem. Diagram aktivitas ini digunakan untuk memodelkan aspek dinamis sistem. Diagram aktivitas mendeskripsikan aksi-aksi dan hasilnya. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokkan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak. Berikut Contoh *Activity* Diagram dapat dilihat pada Gambar 2.9.



Gambar 2.11 Contoh *Activity* Diagram

Tabel 2.3 Elemen *Activity Diagram*

Simbol	Keterangan
	Start Point
	End Point
	Activities
	Fork (Percabangan)
	Join (Penggabungan)
	Decision
Swimlane	Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama)