BAB II

TINJAUAN PUSTAKA

II.1 Kota Semarang

II.1.1 Sejarah Kota Semarang

Sejarah Kota Semarang berawal kurang lebih pada abad ke-8 M, tepatnya di daerah pesisir yang bernama Pragota (sekarang menjadi Bergota) dan merupakan bagian dari kerajaan Mataram Kuno. Daerah tersebut pada masa itu merupakan pelabuhan dan di depannya terdapat gugusan pulau-pulau kecil. Akibat pengendapan, yang hingga sekarang masih terus berlangsung, gugusan tersebut sekarang menyatu membentuk daratan.

Berdirinya Kota Semarang diawali dengan diangkatnya Ki Ageng Pandan Arang sebagai Bupati Semarang yang pertama oleh pemerintahan Kesultanan Demak. Sebagai pusat pemerintahan kadipaten saat itu adalah di sekitar Kanjengan, dengan peninggalan antara lain Masjid Kauman. Setelah Ki Ageng Pandan Arang wafat, digantikan oleh puteranya bernama Pandan Arang II yang diangkat oleh Kesultanan Demak pada tanggal 2 Mei 1547, yang dikemudian ditetapkan sebagai Hari Lahir Kota Semarang. Sejak saat itu Kota Semarang mengalami perkembangan yang pesat dengan difungsikannya pelabuhan Semarang sebagai pelabuhan dagang dan pusat penyiaran Agama Islam (Wikipedia, 2013).

II.1.2 Geografis Kota Semarang

Sebagai ibukota Provinsi Jawa Tengah, Secara geografis Kota Semarang terletak antara 6˚50’ - 7˚10’ Lintang Selatan dan 109˚35’ - 110˚50’ Bujur Timur, dengan batas-batas sebelah Utara dengan Laut Jawa, sebelah Timur dengan Kabupaten Demak, sebelah Barat dengan Kabupaten Kendal dan sebelah Selatan dengan Kabupaten Semarang.

Kota Semarang yang memiliki Luas 373,70 km atau 37.366.836 Ha terdiri dari 16 kecamatan dan 117 kelurahan. Penduduknya sangat heterogen terdiri dari campuran beberapa etnis, Jawa, Cina, Arab dan Keturunan. Juga etnis lain dari beberapa daerah di Indonesia yang datang di Semarang untuk berusaha, menuntut ilmu maupun menetap selamanya di Semarang. Mayoritas penduduk memeluk Agama Islam, kemudian berikutnya adalah Kristen, Katholik, Hindu dan Budha. Mata pencaharian penduduk beraneka ragam, terdiri dari pedagang, pegawai pemerintah, pekerja pabrik dan petani (Dinas Kebudayaan dan Priwisata Kota Semarang, 2013).

II.1.3 Pariwisata Kota Semarang

Menurut Undang-Undang No. 10 Tahun 2009 tentang Kepariwisataan Bab I Pasal 1 ; dinyatakan bahwa wisata adalah kegiatan perjalanan yang dilakukan oleh seseorang atau sekelompok orang dengan mengunjungi tempat tertentu untuk tujuan rekreasi, pengembangan pribadi atau mempelajari keunikan daya tarik wisata yang kunjungi dalam jangka waktu sementara (Undang-undang Kepariwisataan UU No. 10 tahun 2009, 2010).

Dalam hal pariwisata, Kota Semarang sedang melakukan pengembangan dan promosi wisata dengan dicanangkannya program “Ayo Wisata ke Semarang” yang dimulai sejak akhir tahun 2011 lalu. Tentunya Pemerintah Kota Semarang telah menyiapkan beberapa objek wisata unggulan yang tersebar di seluruh Kota Semarang. Objek wisata unggulan terdapat sekitar 27 objek wisata dengan beragam jenisnya, mulai dari objek wisata religi, objek wisata alam, objek wisata sejarah, objek wisata pendidikan, dan jenis wisata lainnya (Dinas Kebudayaan dan Pariwisata Kota Semarang, 2013).

II.2 Sistem Informasi Geografis

Sistem Informasi Geospasial atau juga dikenal sebagai Sistem Informasi Geografis (SIG) mulai dikenal pada awal tahun 1980-an. SIG adalah suatu sistem untuk memperoleh, menyimpan, menganalisis dan mengelola data spasial beserta data atribut terkait yang secara keruangan direferensikan pada bumi. Dangermond

mendefinisikan SIG sebagai kumpulan yang terorganisir dari perangkat keras komputer, perangkat lunak, data geografi dan personil yang didesain untuk memperoleh, menyimpan, memperbaiki, memanipulasi, menganalisis dan menampilkan semua bentuk informasi yang bereferensi geografi. Sedangkan pengertian lain dari SIG adalah serangkaian prosedur baik dengan komputer maupun manual yang digunakan untuk menyimpan dan memanipulasi data bereferensi geografis atau data geospasial. Pengertian SIG dapat beragam tetapi mempunyai satu kesamaan, yaitu bahwa SIG adalah suatu sistem yang berkaitan dengan informasi geografis . Dalam arti yang lebih sempit, SIG merupakan suatu sistem berbasis komputer yang digunakan untuk menyimpan dan menganalisis objek-objek dan fenomena-fenomena dengan lokasi geografis merupakan karakteristik yang penting untuk dianalisis.

SIG dapat diuraikan menjadi beberapa subsistem sebagai berikut (Prahasta, 2005):

a. Data *Input*

   Sub-sistem ini bertugas untuk mengumpulkan, mempersiapkan, dan menyimpan data spasial dan atributnya dari berbagai sumber. Sub-sistem ini pula yang bertanggung jawab dalam mengonversikan atau mentransformasikan format-format data aslinya ke dalam format yang dapat digunakan oeh perangkat SIG yang bersangkutan.

b. Data *Output*

   Sub-sistem ini bertugas untuk menampilkan atau menghasilkan keluaran (termasuk mengekspornya ke format yang dikehendaki) seluruh atau sebagian basis data (spasial) baik dalam bentuk *softcopy* maupun *hardcopy* seperti halnya tabel, grafik, *report*, peta, dan lain sebagainya.

c. Data *Management*

   Sub-sistem ini mengorganisasikan baik data spasial maupun tabel-tabel atribut terkait ke dalam sebuah sistem basis data sedemikian rupa hingga mudah dipanggil kembali atau di-*retrieve*, di-*update*, dan di-*edit*.

d. Data *Manipulation & Analysis*

Sub-sistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu sub-sistem ini juga melakukan manipulasi (evaluasi dan penggunaan fungsi-fungsi dan operator matematis & logika) dan pemodelan data untuk menghasilkan informasi yang diharapkan.

Komponen-komponen dalam SIG (Sistem Informasi Geografis) terdiri dari :

a. Perangkat Keras Komputer

SIG membutuhkan komputer untuk penyimpanan dan pemrosesan data. Ukuran dari sistem komputerisasi bergantung pada tipe SIG itu sendiri. *Hardware* yang digunakan dalam SIG memiliki spesifikasi yang lebih tinggi dibandingkan dengan sistem informasi lainnya.

b. Perangkat Lunak Komputer

Sebuah *software* SIG haruslah menyediakan fungsi dan *tool* yang mampu melakukan penyimpanan data, analisis, dan menampilkan informasi geografis. Sebagai inti dari sistem SIG adalah *software* dari SIG itu sendiri yang menyediakan fungsi-fungsi untuk penyimpanan, pengaturan, *link, queri,* dan analisis data geografi.

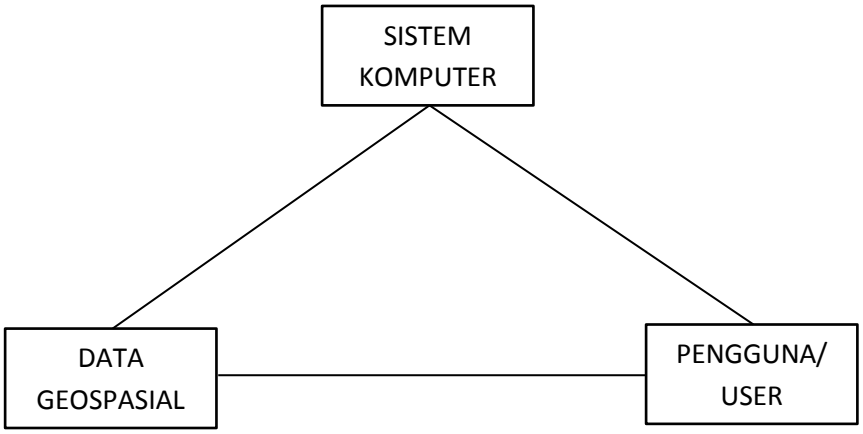
c. Data-data Geografis

SIG dapat mengumpulkan dan menyimpan data dan informasi yang diperlukan baik secara langsung maupun tidak langsung. Data yang dapat diolah dalam SIG merupakan fakta-fakta di permukaan bumi yang memiliki referensi keruangan baik referensi secara relatif maupun referensi secara absolut, dan disajikan dalam sebuah peta.

d. Sumberdaya Manusia

Sumberdaya manusia yang terlatih merupakan sebagai komponen terakhir dari SIG. Peranannya adalah sebagai pengoperasi perangkat keras dan perangkat lunak, serta menangani data geografis dengan kedua perangkat tersebut. Sumberdaya manusia juga merupakan sebagai sistem analis yang

menerjemahkan permasalahan riil di permukaan bumi dengan bahasa SIG, sehingga permasalahan dapat diidentifikasi dan dicari solusinya.

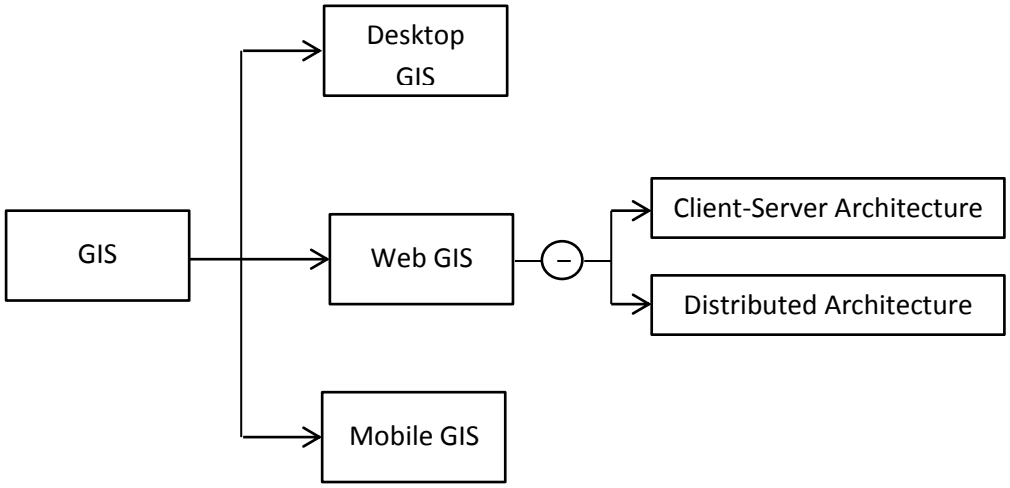


Gambar 2.1 Komponen Sistem Informasi Geografis (Prahasta, 2005)

Data yang diolah pada SIG adalah data geospasial, yang terdiri dari data spasial dan data non spasial. Pada diagram di atas data non spasial tidak digambarkan karena sebagian besar data yang akan ditangani dalam SIG merupakan data spasial yaitu sebuah data yang berorientasi geografis, memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang membuatnya berbeda dari data lain yaitu informasi lokasi (spasial) dan informasi deskriptif (atribut). Data spasial dapat diperoleh dari berbagai sumber seperti peta analog (seperti peta topografi, peta tanah , dan sebagainya), data sistem penginderaan jauh (citra satelit dan foto udara), data hasil pengukuran lapangan dan data GPS. Sedangkan data non spasial adalah data selain data spasial yaitu data yang berupa teks atau angka. Data non spasial ini akan menerangkan data spasial atau sebagai dasar untuk menggambarkan data spasial. Data non spasial ini nantinya dapat dibentuk data spasial.

Menurut Muehler dan McKee dalam bukunya "*OpenGIS Guide*", terdapat dua layanan utama dalam SIG yaitu layanan data geografis (*geodata service*) dan layanan pemrosesan data geografis *(geoprocessing service).* Berdasarkan teknologi dan implementasinya, sistem informasi geografis dapat dikategorikan dalam tiga aplikasi yaitu SIG berbasis *desktop* (*desktop* GIS), SIG berbasis *web*

(*web* GIS), dan SIG berbasis *mobile* (*mobile* GIS). Meskipun demikian, ketiganya saling berhubungan satu sama lain (Riyanto, 2010).



Gambar 2.2. Kategori SIG (Riyanto, 2010)

II.2.1 *Mobile* GIS

Mobile GIS merupakan integrasi antara beberapa teknologi, yaitu :

- Perangkat *Mobile*
- *Global Positioning System* (GPS)
- *Wireless communication* untuk mengakses *Internet* GIS.

Dengan kombinasi dari beberapa teknologi di atas membuat *mobile* GIS dapat digunakan untuk menangkap, menyimpan, update, manipulasi, analisis dan menampilkan informasi geografi secara tepat. Sehingga melalui teknologi tersebut juga dapat membuat basis data yang diakses oleh personil di lapangan secara langsung di segala tempat dan waktu. Sistem ini dapat menambah informasi secara *real-time* ke basis data dan aplikasinya dalam hal kecepatan akses, tampilan, dan penentuan keputusan.

Mobile GIS menawarkan fleksibilitas yang besar, memungkinkan pengguna memperoleh hasil secara cepat sesuai dengan kebutuhan mereka. *Mobile* GIS menyediakan akses data dari segala tempat dan di kapanpun keberadaan pengguna. Adapun beberapa komponen yang bergabung membentuk

mobile GIS, yaitu *mobile client*, jaringan tanpa kabel, dan *server*. *Mobile client* berupa perekam data posisi misalnya GPS, yang mana pergerakan *mobile* dengan GPS yang diperoleh dan dengan GSM dapat mengirimkan posisi geografis ke *server* atau dalam kondisi lain dimana orang yang membawa *smartphone* yang di dalamnya sudah terinstal sistem operasi tertentu seperti dengan dilengkapi GPS. *Smartphone* tersebut dapat menunjukkan peta digital beserta koordinatnya dengan mengkomunikasikan dengan *server* melalui jaringan tanpa kabel. Jaringan tersebut dapat melalui *Global System for Mobile Communication* (GSM), *General Pocket Radio System* (GPRS), *Code Division Multiple Access* (CDMA) yang mendukung transmisi digital.

II.3 GPS (*Global Positioning System*)

GPS (*Global Positioning System*) merupakan sebuah sistem satelit navigasi dan penentuan posisi dengan menggunakan satelit. GPS dapat memberikan informasi tentang posisi, kecepatan, dan waktu secara cepat, akurat, murah, dimana saja di bumi ini pada setiap saat tanpa tergantung cuaca (Abidin, 2007).

Pada dasarnya GPS terdiri atas tiga segmen utama, yaitu segmen angkasa (*space segment*) yang terdiri dari satelit-satelit GPS, segmen sistem kontrol (*control system segment*) yang terdiri dari stasiun-stasiun pengamat dan pengendali satelit, dan segmen pemakai (*user segment*) yang terdiri dari pemakai GPS termasuk alat-alat penerima dan pengolah sinyal dan data GPS (Abidin, 2006).

Berikut merupakan penjelasan mengenai tiga segmen utama pada GPS, yaitu segmen angkasa, segmen sistem kontrol, dan segmen pengguna.

1. Segmen Angkasa

   Satelit GPS dapat dianalogikan sebagai stasiun radio angkasa, yang dilengkapi dengan antena-antena untuk mengirim dan menerima sinyal-sinyal gelombang. Sinyal-sinyal ini selanjutnya diterima oleh *receiver* GPS di dekat

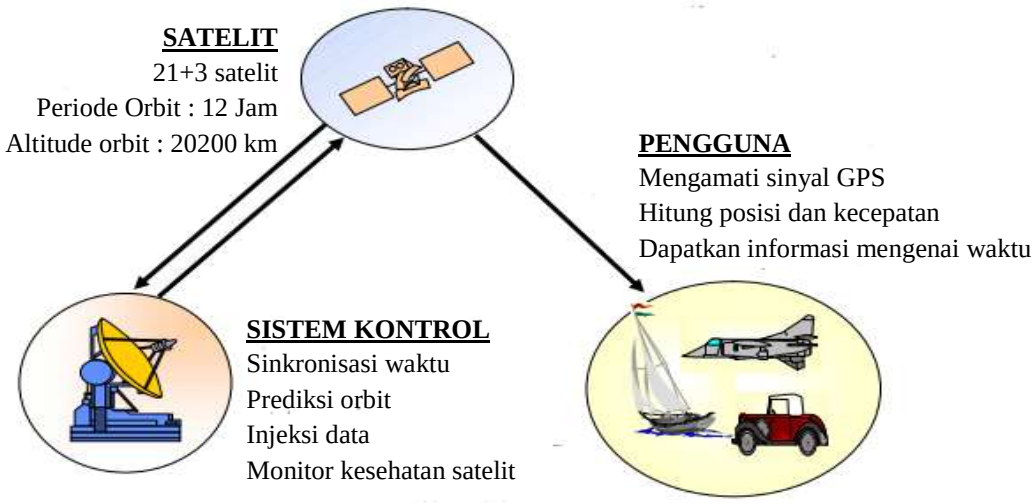
permukaan bumi, dan digunakan untuk menentukan informasi posisi, kecepatan maupun waktu.

2. Segmen Sistem Kontrol

   Segmen ini terdiri atas GCS (*Ground Control Station*), MS (*Monitor Station*), PCS (*Prelaunch Control Station*). Segmen ini berfungsi mengontrol dan memantau operasional satelit dan memastikan bahwa satelit berfungsi sebagai mana mestinya.

3. Segmen Pengguna

   Segmen pengguna terdiri dari para pengguna satelit GPS dimanapun berada. Dalam hal ini alat penerima sinyal GPS (GPS *receiver*) diperlukan untuk menerima dan memproses sinyal-sinyal dari satelit. Ada tiga macam tipe *receiver* GPS, dengan masing-masing memberikan tingkat ketelitian (posisi) yang berbeda-beda. Tipe alat GPS pertama adalah tipe navigasi (*handheld*) dengan ketelitian 3-6 meter. Tipe alat yang kedua adalah tipe geodetik *single* frekuensi (tipe pemetaan), yang biasa digunakan dalam survey dan pemetaan yang membutuhkan ketelitian posisi sekitar sentimeter sampai dengan beberapa desimeter. Tipe terakhir adalah tipe geodetik *dual* frekuensi yang dapat memberikan ketelitian posisi mencapai milimeter (Abidin, 2006).



Gambar 2.3 Segmen GPS (Abidin, 2006)

II.3.1 Sinyal GPS

Satelit GPS memancarkan sinyal, pada prinsipnya untuk memberi tahu si pengamat sinyal tersebut tentang posisi satelit GPS yang bersangkutan serta jaraknya dari pengamat lengkap dengan informasi waktunya, dan juga digunakan untuk menginformasikan kondisi satelit kepada pengamat, serta informasi pendukung lainnya seperti perhitungan jam satelit, model ionosfer pada satu frekuensi, transformsi dari waktu GPS ke UTC, serta status konstelasi satelit (Abidin, 2006). Sinyal GPS dibagi dalam tiga komponen yaitu :

1. Sinyal informasi jarak (*code*) yang berupa kode P dan kode C/A
   Sinyal ini terdiri dari dua kode *Pseudo Random Noise* (PSN) yaitu kode P (*private* atau *precise*) dan kode C/A (*Coarse Acquisition* atau *Clear Access*). Kode ini merupakan suatu rangkaian bilangan biner 0 dan 1. Setiap satelit GPS mempunyai struktur kode yang unik dan berbeda dengan satelit GPS lainnya. Hal ini memungkinkan GPS *receiver* untuk mengenali dan membedakan sinyal yang datang dari satelit GPS yang berbeda.
2. Sinyal informasi posisi satelit (*navigation message*)
   Disamping berisi kode-kode, sinyal GPS juga berisi pesan navigasi yang berisi informasi tentang koefisien koreksi jam satelit, parameter koreksi ionosfer, serta informasi spasial lainnya seperti status konstelasi dan kondisi satelit. Pesan navigasi ini ditentukan oleh segmen sistem kontrol dan dikirimkan ke pengguna dengan menggunakan satelit GPS.
3. Gelombang pembawa (*carrier wave*) L1/L2/L3
   Satelit GPS memancarkan sinyal dalam 3 frekuensi yang berbeda, dan setiap frekuensi membawa informasi atau kode.

II.3.2 Penentuan Posisi dengan GPS

Pada dasarnya penentuan posisi dengan GPS adalah pengukuran jarak secara bersama-sama ke beberapa satelit (yang koordinatnya telah diketahui) sekaligus. Untuk menentukan koordinat suatu titik di bumi, *receiver* setidaknya

membutuhkan 4 satelit yang dapat ditangkap sinyalnya dengan baik. Secara *default* posisi atau koordinat yang diperoleh bereferensi ke datum global yaitu *World Geodetic System* 1984 (WGS'84).

Secara garis besar penentuan posisi dengan GPS ini dibagi menjadi dua metode yaitu metode absolut dan metode relatif.

a. Metode absolut atau juga dikenal sebagai *point positioning*, menentukan posisi hanya berdasarkan pada 1 pesawat penerima (*receiver*) saja. Ketelitian posisi dalam beberapa meter (tidak berketelitian tinggi) dan umumnya hanya diperuntukkan bagi keperluan navigasi.
b. Metode relatif atau sering disebut *differential positioning*, menetukan posisi dengan menggunakan lebih dari satu *receiver*. Satu GPS dipasang pada lokasi tertentu dimuka bumi dan secara terus menerus menerima sinyal dari satelit dalam jangka waktu tertentu untuk dijadikan sebagai referensi bagi yang lainnya. Metode ini menghasilkan posisi berketelitian tinggi (umumnya kurang dari 1 meter) dan diaplikasikan untuk keperluan survei geodesi ataupun pemetaan yang memerlukan ketelitian tinggi.

II.3.3 *Assisted Global Positioning System* (A-GPS)

A-GPS (*Assisted* GPS) adalah sebuah teknologi yang menggunakan sebuah *server* bantu untuk mempercepat waktu yang diperlukan dalam menentukan sebuah posisi menggunakan perangkat GPS. A-GPS akan memberikan informasi mengenai satelit mana saja yang dapat digunakan dengan cepat tanpa harus mendeteksi seluruh satelit yang ada, sehingga dapat mengurangi waktu yang dibutuhkan secara signifikan untuk menentukan posisi saat ini yang disebut juga sebagai *Time to First Fix* (TTFF).

A-GPS didesain agar perangkat dapat terhubung ke satelit dengan lebih cepat dan lebih dapat diandalkan daripada menggunakan GPS tunggal. A-GPS menggunakan metode yang berbasis pada waktu. Pada metode ini, akan dilakukan pengukuran waktu tiba dari sebuah sinyal yang dikirimkan dari satelit GPS. Hal

ini berarti pada perangkat yang digunakan harus memiliki fasilitas untuk mengakses GPS. A-GPS seperti halnya GPS, juga menggunakan satelit yang memancarkan sinyal ke penerima.

Server bantuan penyedia data informasi satelit yang dibutuhkan oleh A-GPS biasanya di dukung oleh jaringan operator karena sering kali menara BTS memiliki unit penerima GPS dan secara terus-menerus akan men-*download* informasi data satelit yang ada diangkasa dan kemudian memprosesnya. Data dari *server* bantuan bisa diberikan kepada pelanggan telepon selular, bila diminta oleh perangkat A-GPS untuk mengindentifikasi lokasi pengguna berupa latitude dan longitude, lokasi dalam peta, dan lain-lain.

Dalam hal ini dibutuhkan 3 komponen dalam proses penentuan posisi yaitu satelit, *assistance server* (GSM), *receiver* A-GPS. Selain itu A-GPS berbeda dari reguler GPS dengan menambahkan elemen lain ke dalam proses pencarian posisi, yaitu *server* bantuan (*assistance server*). A-GPS memiliki beberapa kelebihan, antara lain :

- Dapat mengindentifikasi lokasi dengan lebih cepat.
- Membutuhkan power lebih kecil dalam proses komputasi data.
- Cocok untuk lokasi perkotaan atau lokasi yang kurang optimal dalam menangkap sinyal satelit seperti dalam gedung.

Sedangkan kekurangan A-GPS adalah :

- Tergantung pada coverage jaringan provider
- Membutuhkan biaya akses data

II.4 *Location Based Service* (LBS)

Location Based Service atau LBS memiliki kemampuan untuk mencari lokasi geografis dari *mobile device* dan menyediakan layanan berdasarkan lokasi yang diperolehnya. Konsep LBS ini menghasilkan layanan informasi mengenai lokasi keberadaan *user*. Hal ini menyebabkan peningkatan nilai informasi, dikarenakan penerima dapat mengasosiasikan pengetahuan atau informasi yang

didapat dengan keberadaannya. Peningkatan nilai informasi ini terkait secara erat dengan potensi nilai komersial bagi penyedia layanan (*provider*) layanan LBS. *Provider* dapat menyediakan informasi secara instan pada saat konsumen memerlukan dan relevan terhadap lokasi konsumen.

LBS dapat diklasifikasikan menjadi tiga jenis, yaitu *local information*, *traffic and tracking information* dan *general services*.

a. *Local Information*
Memungkinkan pengguna untuk mencari layanan di sekitar mereka.

b. *Traffic and Tracking Information*
Berfokus kepada pelacakan aset atau manusia.

c. *General Services*
Tidak menyediakan informasi ke pengguna, namun menggunakan data lokasi pengguna.

II.4.1 Komponen LBS

Terdapat empat komponen pendukung utama dalam teknologi Layanan Berbasis Lokasi atau LBS, antara lain :

1. Perangkat *Mobile*

   Perangkat *mobile* berfungsi sebagai alat bantu (*tool*) bagi pengguna untuk mendapatkan informasi. Informasi yang didapat dapat berupa teks, suara, gambar dan lain sebagainya. Perangkat *mobile* yang dapat digunakan bisa berupa PDA, *smartphone*, *laptop*.

2. Jaringan Komunikasi

   Komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari perangkat *mobile* untuk kemudian dikirimkan ke *provider* dan kemudian informasi tersebut dikirimkan kembali oleh *provider* kepada *user*.

3. Komponen *Positioning* (Penunjuk Posisi/Lokasi)

   Setiap layanan yang diberikan oleh *provider* biasanya akan berdasarkan pada posisi *user* yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah/pemroses yang akan menentukan posisi *user* saat itu. Posisi *user* tersebut bisa didapatkan melalui jaringan komunikasi *mobile* atau juga menggunakan *Global Positioning System* (GPS).

4. *Provider* dan Aplikasi

   Provider merupakan komponen LBS yang memberikan berbagai macam layanan yang bisa digunakan oleh *user*. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut.

5. Penyedia Data dan Konten

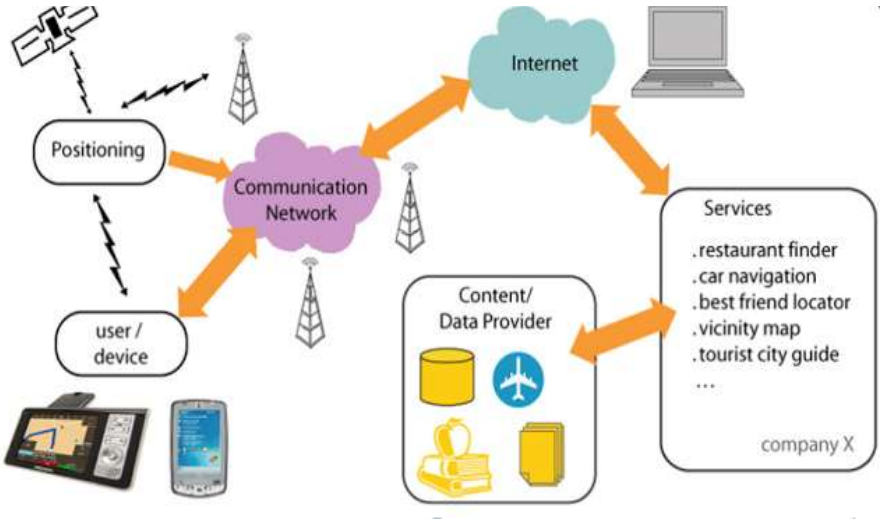
   Provider tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang/pihak ketiga yang memang memiliki otoritas untuk menyimpannya (Riyanto, 2010)

Gambar 2.4 Komponen pendukung utama teknologi LBS (Riyanto, 2010)

II.4.2 Cara Kerja *Location Based Service*

Berikut adalah penggambaran cara kerja LBS pada aplikasi mengenai pencarian objek wisata terdekat berdasarkan posisi *user*.

1. Pertama *smartphone* akan membuka aplikasi yang tentunya memanfaatkan layanan LBS yang sudah ter-*install*.
2. Kemudian aplikasi akan melakukan sambungan dengan jaringan *provider* (seperti telkomsel, xl, tri, dll) yang dipakai oleh *user* (pengguna).
3. Selanjutnya aplikasi akan mengambil informasi posisi *user* pada perangkat *mobile* yang diperoleh dari *Location Sensor*. Hal ini dapat dilakukan baik oleh perangkat menggunakan GPS sendiri atau layanan posisi jaringan yang berasal dari *provider*.
4. Setelah itu perangkat *mobile* pengguna akan mengirimkan permintaan informasi ke satelit untuk menentukan longitude (garis bujur) dan latitude (garis lintang) dari si pengguna aplikasi tersebut.
5. *Provider* menghubungkan aplikasi (di *smartphone*) dengan *server* LBS dan meminta data yang diinginkan *user* beserta informasi tentang jalan, jarak dan cara yang diperlukan dalam menjangkau lokasi tujuan.
6. Terakhir *user* mendapatkan data dan ditampilkan di *smartphone* melalui aplikasi.



Gambar 2.5 Cara Kerja *Location Based Sensor* (Riyanto, 2010)

II.5 *Smartphone*

Telepon cerdas *(smartphone)* adalah telepon genggam yang mempunyai kemampuan tingkat tinggi, bahkan hampir memiliki fungsi yang menyerupai komputer. Bagi beberapa orang, telepon pintar merupakan telepon yang bekerja menggunakan seluruh perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Dengan kata lain, telepon cerdas merupakan komputer kecil yang mempunyai kemampuan sebuah telepon.

Pertumbuhan permintaan akan alat canggih yang mudah dibawa ke mana-mana membuat kemajuan besar dalam pemroses, memori, layar dan sistem operasi yang di luar dari jalur telepon genggam sejak beberapa tahun ini. Dengan menggunakan telepon cerdas hanya merupakan sebuah evolusi dari jenjang-jenjang evolusi. Kebanyakan alat yang dikategorikan sebagai telepon cerdas menggunakan sistem operasi yang berbeda. Dalam hal fitur, kebanyakan telepon pintar mendukung sepenuhnya fasilitas dan fungsi pengatur personal yang lengkap. Fungsi lainnya dapat menyertakan miniatur *keyboard* QWERTY, layar sentuh (*touchscreen*) atau D-pad, kamera, pengaturan daftar nama, penghitung kecepatan, navigasi piranti lunak dan keras, kemampuan membaca dokumen bisnis, pemutar musik, penjelajah foto dan melihat klip video, penjelajah internet, dan lainnya. Fitur yang paling sering ditemukan dalam telepon cerdas adalah kemampuannya menyimpan daftar nama sebanyak mungkin, tidak seperti telepon genggam biasa yang mempunyai batasan maksimum penyimpanan daftar nama (Wikipedia, 2013).

II.5.1 Android

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

Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Android adalah sistem operasi dengan sumber terbuka (*open source*), dan Google merilis kodenya di bawah lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (*apps*) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java.

Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikustomisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Akibatnya, meskipun pada awalnya sistem operasi ini dirancang khusus untuk telepon pintar dan tablet, Android juga dikembangkan menjadi aplikasi tambahan di televisi, konsol permainan, kamera digital, dan perangkat elektronik lainnya. Sifat Android yang terbuka telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain (Wikipedia, 2013).

Android dianggap sebagai platform masa depan yang lengkap, terbuka dan bebas sebagai berikut (Safaat, 2012) :

- Lengkap (*Complete Platform*)

  Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam

membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.

- Terbuka (*Open Source*)

  Platform Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux kernel 2.6.

- Bebas (*Free Platform*)

  Android adalah *platform* atau aplikasi yang bebas develop. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Aplikasi untuk android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

Seiring dengan pengembangannya, Android memiliki berbagai macam versi, antara lain (newbiedroid.blogspot.com) :

a. Android 2.3 – 2.3.7 (Gingerbread)

   Versi ini dirilis pada 6 Desember 2010. Gingerbread adalah kue yang terbuat dari jahe. Gingerbread dipakai sebagai nama alias dari sistem operasi Android versi 2.3. Ada banyak peningkatan pada versi Android yang satu ini dibandingkan dengan versi sebelumnya, seperti memaksimalkan kemampuan aplikasi dan *game*, dukungan layar resolusi WXGA dan diatasnya ,serta mulai digunakannya *Near Field Communication* (NFC). Beberapa versi *update* yang dirilis antara lain v.2.3.3 hingga v.2.3.7.

b. Android versi 3.0 -3.2 (Honeycomb)

   Versi ini dirilis pada 22 Februari 2011. Honeycomb atau sarang madu, dipakai sebagai nama alias dari sistem Android versi 3.0 Android versi ini merupakan OS yang didesain khusus untuk pengoptimalan pengunaan pada tablet PC.

c. Android versi 4.0 – 4.0.4 (Ice Cream Sandwich)

   Versi ini dirilis pada 19 Oktober 2011. Ice Cream Sandwich dipakai sebagai nama alias dari Android versi 4.0. Secara teori semua perangkat seluler yang

menggunakan versi Android sebelumnya, Gingerbread, dapat di-*update* ke Android Ice Cream Sandwich.

d. Android versi 4.1 – 4-3 (Jelly Bean)

Android Jelly Bean memiliki penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui *Voice Search* yang lebih cepat. Terdapat pula aplikasi *Google Now* yang dapat memberikan informasi yang tepat pada waktu yang tepat pula. Selain itu dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 kini sudah mencapai versi Jelly Bean 4.3.

e. Android versi 4.4 (KitKat)

Android KitKat diluncurkan pada Oktober 2013. Penamaan pada versi ini didasarkan dari kerja sama antara Google dengan perusahaan Nestle. Android versi 4.4 ini diprediksikan akan kompatibel untuk digunakan pada *smartphone* Android mulai dari kelas *high-end* hingga *low-end.*

Tabel 2.1 Perkembangan Sistem Operasi Android (Wikipedia, 2013)

Versi	Nama Kode	Tanggal Rilis	Level API	Distribusi
4.4	KitKat	Oktober 2013	19	-
4.3	Jelly Bean	24 Juli 2013	18	-
4.2.x	Jelly Bean	13 November 2012	17	6,5%
4.1.x	Jelly Bean	9 Juli 2012	16	34.0%
4.0.3-4.0.4	Ice Cream Sandwich	16 Desember 2011	15	22,5%
3.2	Honeycomb	15 Juli 2011	13	0,1%
3.1	Honeycomb	10 Mei 2011	12	0,0%
2.3.3-2.3.7	Ginger Bread	9 Februari 2011	10	33,0%
2.3-2.3.2	Ginger Bread	6 Desember 2010	9	0,1%
2.2	Froyo	20 Mei 2010	8	2,5%
2.0-2.1	Eclair	26 oktober 2009	7	1,2%
1.6	Donut	15 September 2009	4	0,1%
1.5	Cupcake	30 April 2009	3	0%

II.5.2 Sinyal

Sinyal merupakan suatu jaringan yang digunakan pada *smartphone* atau modem untuk mengakses atau menghubungkan perangkat ke jaringan internet dengan kecepatan tertentu. Berikut ini adalah beberapa jenis sinyal yang digunakan dalam mengakses data melalui internet, antara lain :

a. GPRS (*Global Package Radio Service*)

   Suatu teknologi yang memungkinkan pengiriman dan penerimaan data dalam bentuk paket data yang berkaitan dengan *e-mail*, data gambar, dan penelusuran internet. GPRS yang juga disebut teknologi 2.5G merupakan evolusi dari teknologi 1G dan 2G sebelumnya. Idealnya jaringan GPRS memiliki kecepatan mulai dari 56 kbps sampai 115 kbps, namun hal tersebut tergantung dari faktor-faktor seperti konfigurasi dan alokasi *time slot* pada level BTS, *software* yang digunakan, dan dukungan fitur dan aplikasi *smartphone* yang digunakan.

b. EDGE (*Enhance Data rates for Global Evolution*)

   Merupakan kelanjutan evolusi dari GSM dan IS-136 dengan tujuan pengembangan teknologi untuk meningkatkan kecepatan transmisi data, efisiensi spektrum, dan memungkinkannya penggunaan aplikasi-aplikasi baru serta meningkatkan kapasitas. Jaringan EDGE juga disebut sebagai teknologi 2.75G. Jaringan EDGE pada idealnya memiliki kecepatan mencapai 236 kbps.

c. Teknologi 3G (*Third-Generation Technology*)

   Merupakan teknologi evolusi dari generasi sebelumnya yang memiliki kapasitas pengiriman dan penerimaan yang lebih besar dan lebih cepat. Maka teknologi ini dapat digunakan untuk melakukan *video call*. Teknologi 3G sering juga disebut dengan *mobile broadband* karena keunggulannya sebagai modem untuk internet yang bersifat *portable*. Idealnya teknologi ini memiliki kecepatan transfer data pada level minimum 2Mbps pada pengguna yang berada pada posisi diam ataupun berjalan kaki, dan 384 kbps pada pengguna yang berada di dalam kendaraan yang sedang berjalan.

d. HSDPA (*High-Speed Downlink Packet Access*)

   Merupakan teknologi yang disempurnakan dari teknologi sebelumnya yang juga dapat disebut 3.5G, 3G+ atau *Turbo* 3G yang memungkinkan jaringan berbasis *Universal Mobile Telecommunication System* (UMTS) memiliki kecepatan dan kapasitas transfer data yang lebih tinggi. Penggunaan HSDPA saat ini menyokong kecepatan penelusuran dari 1.8, 3.6, 7.2 hingga 14 Mpbs. Oleh karena itulah jaringan HSDPA ini sangat memungkinkan untuk digunakan sebagai modem internet pada komputer ataupun *notebook*.

e. Wi-Fi (WLAN)

   Merupakan sebuah teknologi yang memanfaatkan perangkat elektronik untuk bertukar data secara nirkabel melalui sebuah jaringan komputer yang terhubung dengan sebuah titik akses. Jangkauan titik akses (*hotspot*) sekitar 20 meter di dalam ruangan dan lebih luas lagi jika di luar ruangan. Wi-Fi atau WLAN memiliki kecepatan berkisar antara 11 Mbps hingga 100 Mbps dengan frekuensi band antara 2,4 GHz – 5 GHz, sehingga membuat kecepatan pada jaringan WLAN dapat lebih diandalkan (Wikipedia, 2013).

II.6 *Google Maps*

Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat popular. *Google Maps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. Kita dapat menambahkan fitur *Google Maps* dalam *web* atau aplikasi yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan *Google Maps* API.

Cara membuat *Google Maps* untuk ditampilkan pada suatu *web* atau *blog* sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *Java Script*, serta koneksi Internet yang sangat stabil. Dengan menggunakan *Google Maps* API, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat

suatu data sedangkan peta yang akan ditampilkan adalah milik Google sehingga kita tidak dipusingkan dengan mambuat peta suatu lokasi, bahkan dunia.

Pada *Google Maps* API terdapat empat jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah :

a. Roadmap

Menampilkan peta biasa dua dimensi pada tampilan *Google Maps.*

b. Satellite

Menampilkan foto satelit pada tampilan *Google Maps.*

c. Terrain

Menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai pada tampilan *Google Maps.*

d. Hybrid

Menunjukkan foto satelit yang diatasnya tergambar pula tampilan pada *roadmap* (jalan dan nama kota) pada tampilan *Google Maps.*

II.7 Pemrograman Aplikasi

II.7.1 UML

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP).

Bahasa UML digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek, namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Hal tersebut membuat penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang

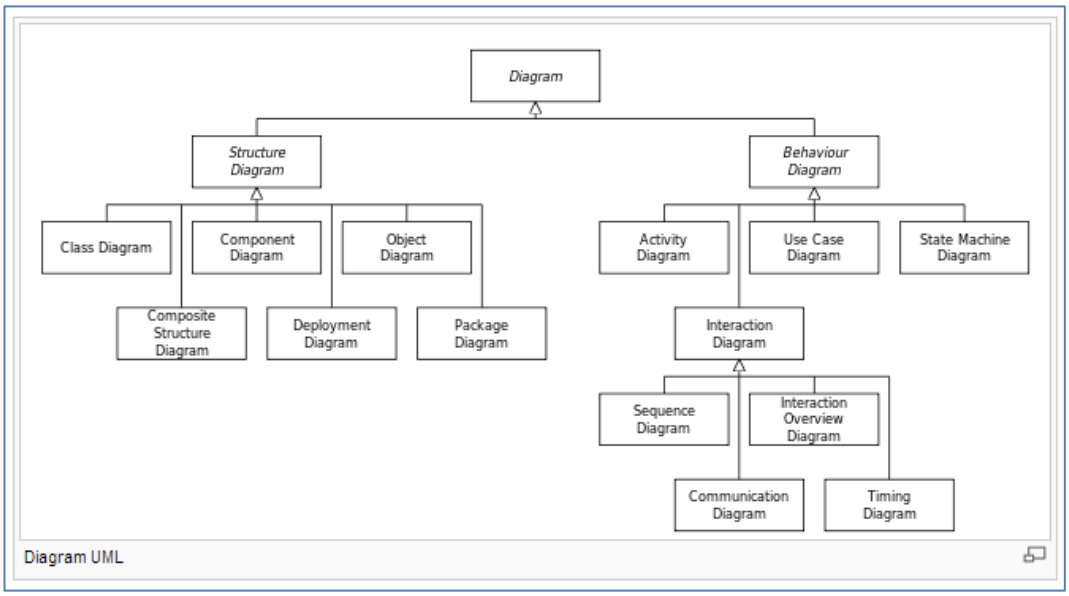
menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

UML menyediakan beberapa jenis diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

- *Use Case* Diagram untuk memodelkan proses bisnis. *Use case* diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

  Use case merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk *elips* dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.
- *Conceptual* Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
- *Sequence* Diagram untuk memodelkan pengiriman pesan (*message*) antar *objects*. *Sequence* diagram menjelaskan secara detail 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.
- *Collaboration* Diagram untuk memodelkan interaksi antar *objects*. *Collaboration* diagram dipakai untuk memodelkan interaksi antar objek di dalam sistem. Berbeda dengan *sequence* diagram yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration* diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh *object*.
- *State* Diagram untuk memodelkan perilaku *objects* di dalam sistem.
- *Activity* Diagram untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.
- *Class* Diagram untuk memodelkan struktur kelas.

- *Object* Diagram untuk memodelkan struktur *object*.
- *Component* Diagram untuk memodelkan komponen *object*.
- *Deployment* Diagram untuk memodelkan distribusi aplikasi (Wikipedia, 2013).



Gambar 2.6 Diagram UML (Wikipedia, 2013)

II.7.2 *App Inventor*

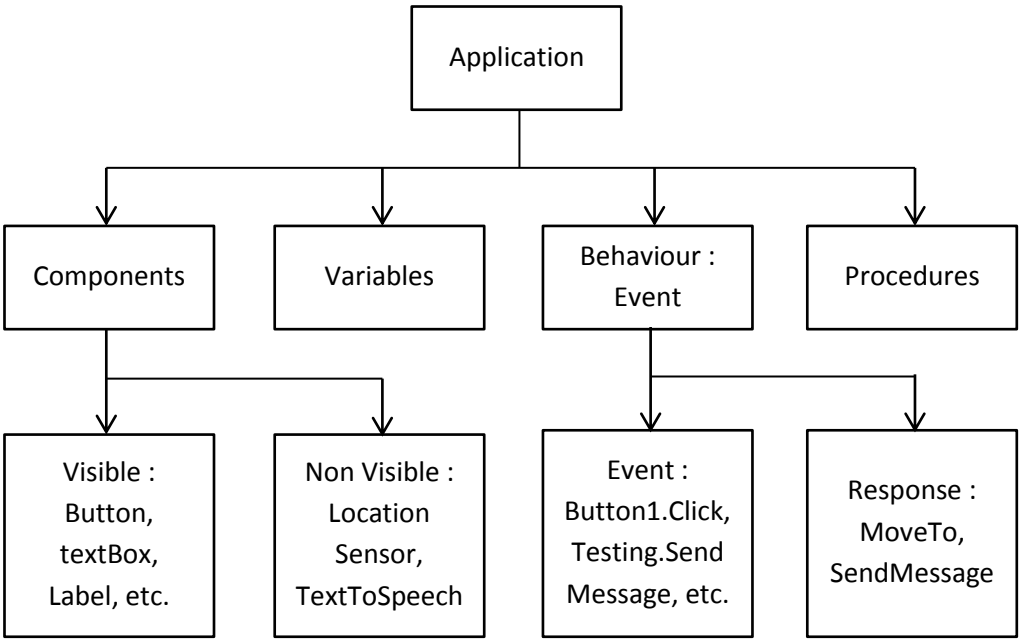
App Inventor for Android (*version* v.134) adalah aplikasi yang awalnya disediakan oleh Google dan sekarang di *maintenance* oleh *Massachusetts Institute of Technology (MIT). App Inventor* memungkinkan semua orang untuk membuat *software* aplikasi untuk sistem operasi Android. Pengguna dapat menggunakan tampilan grafis GUI dan tampilan *drag* and *drop* visual objek untuk membuat aplikasi yang akan dijalankan pada sistem operasi Android.

Dalam penggunaannya *App Inventor* dimulai melalui *web-based service* pada *browser* secara *online (http://beta.appinventor.mit.edu)* atau *offline (local host).* Dengan cara mengatur tampilan aplikasi (*user interface*) pada *web* GUI *(graphical user interface) builder*, kemudian menspesifikasikan *behavior* aplikasi yang ingin anda buat dengan menyusun *block* yang sesuai.

Gambar 2.7 Tampilan *App Inventor*

a. Arsitektur Aplikasi

Secara arsitektur aplikasi, *App Inventor* mempunyai struktur internal yang harus dipahami terlebih dahulu agar dapat dibuat secara efektif. Salah satu cara untuk memahami bagian dalam sebuah aplikasi adalah dengan memecahnya menjadi dua bagian yaitu komponen dan *behavior*. Kedua bagian tersebut adalah sebuah jendela yang saling berhubungan yang nantinya digunakan untuk mengembangkan aplikasi dengan *App Inventor*. Kemudian dengan menggunakan komponen *designer* dapat dibuat komponen dari sebuah aplikasi yang nantinya berjalan sesuai dengan perintah dari sistem *block* yang telah dirancang.



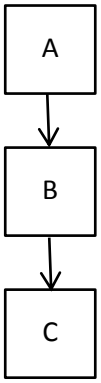
Gambar 2.8 Arsitektur aplikasi *App Inventor*

b. Komponen

Jenis komponen pada *App Inventor* dibagi menjadi 2 , yaitu komponen *visible* (tampak) dan komponen *non visible* (tak tampak). Komponen *visible* adalah komponen yang terlihat oleh pengguna saat aplikasi dijalankan. Sedangkan komponen *non visible* adalah komponen yang tidak dapat terlihat saat aplikasi dijalankan, karena komponen *non visible* memang menyediakan akses untuk membangun fungsionalitas dari perangkat ke aplikasi. Kedua komponen *visible* dan *non visible* didefinisikan oleh sebuah *properties. Properties* merupakan sebuah memori yang digunakan untuk menyimpan informasi mengenai sebuah komponen.

c. *Behavior*

Behavior pada sebuah aplikasi memiliki konsep yang terbilang rumit dan kompleks. Hal tersebut karena sebuah *behavior* harus dapat mendefinisikan aplikasi agar dapat merespon sebuah *event*. *App inventor* memiliki bahasa *visual block* yang dapat menspesifikasikan sebuah *behavior* dengan mengikuti alur instruksi secara sekuensial. Oleh karena itu sebuah aplikasi lebih dikonsepkan sebagai sebuah set komponen yang merespon *event*.



Gambar 2.9 Alur Program Sekuensial

Berdasarkan gambar, pada saat *event* terjadi, aplikasi akan bereaksi dengan memanggil fungsi secara berurutan. Fungsi adalah sesuatu yang akan dilakukan dengan atau tanpa komponen. *Event* dan beberapa set fungsi yang dikerjakan pada sebuah respon disebut dengan *event handler*.

d. *Event*

Event adalah pengeksekusian perintah atau pemanggilan *method* berdasarkan perintah tertentu. Ada beberapa jenis *event* pada *App Inventor*, yaitu :

- *User Initiated Event* : Adalah *event* yang paling sering terjadi. Pada *input form* biasa terdapat *button* yang memicu suatu respon terhadap sebuah aplikasi.
- *Initialization Event* : Adalah penggunaan *event* ketika aplikasi mulai dijalankan oleh pengguna.
- *Timer Event* : Adalah penggunaan *event* berdasarkan pada waktu, dimana dapat mengatur pergerakan dari sebuah *event* dengan waktu tertentu.
- *Animation Event* : Adalah pengaturan sebuah *event* dengan menggunakan objek grafis dengan perintah efek tertentu.
- *External Event* : Adalah sebuah *event* yang kita dapat dari luar aplikasi, seperti informasi lokasi dari satelit GPS.

e. IDE

App Inventor pada dasarnya bekerja secara *online* melalui *browser internet*, tetapi diperlukan beberapa *software* pendukung paket Java untuk membuka

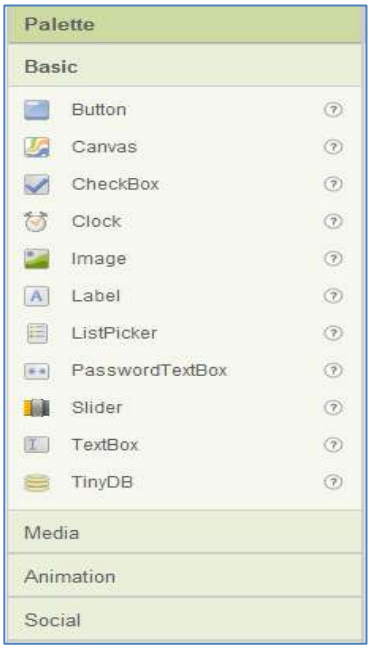
block designer, *emulator* untuk menjalankan aplikasi yang dibuat, dan lainnya. Lingkungan kerja *App Inventor* memiliki tiga bagian besar, yaitu :

- Komponen *Designer*

  Digunakan untuk memilih komponen yang akan dimasukkan pada aplikasi yang dibuat. Komponen *designer* terbagi menjadi beberapa bagian, yaitu *Palette*, *Viewer*, *Components*, *Properties*, dan *Main menu*.

Gambar 2.10 Tampilan komponen *Designer*

Palette digunakan untuk mengambil objek atau komponen yang akan digunakan oleh *block editor*. *Palette* terdiri dari beberapa komponen seperti *Basic, Media, Social, Animation, Sensor, Screen Arrangement, Lego mindstrom,* dan *Other stuff.*

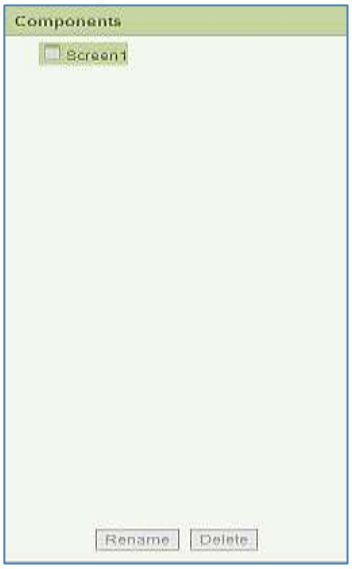


Gambar 2.11 Tampilan komponen *Palette* pada *Designer*

Viewer atau disebut juga *form designer* digunakan untuk mendesain tampilan *interface* dari aplikasi yang akan dibuat. Untuk perancangan tampilan atau *user interface* dapat diambil melalui komponen *palette.*

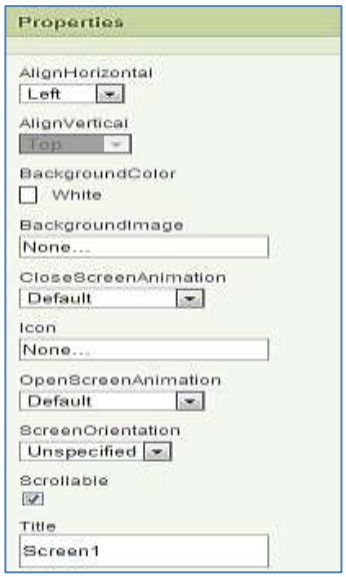
Gambar 2.12 Tampilan komponen *Viewer* pada *Designer*

Components digunakan untuk melihat daftar komponen yang terdapat pada suatu *screen* atau *form. Component* juga dapat digunakan untuk menghapus atau mengubah nama dari suatu komponen yang diletakkan pada *screen*.



Gambar 2.13 Tampilan komponen *Components* pada *Designer*

Properties digunakan untuk mengubah tampilan, teks atau kelengkapan sebuah komponen, sehingga dapat mengatur tampilan pada *interface* aplikasi.



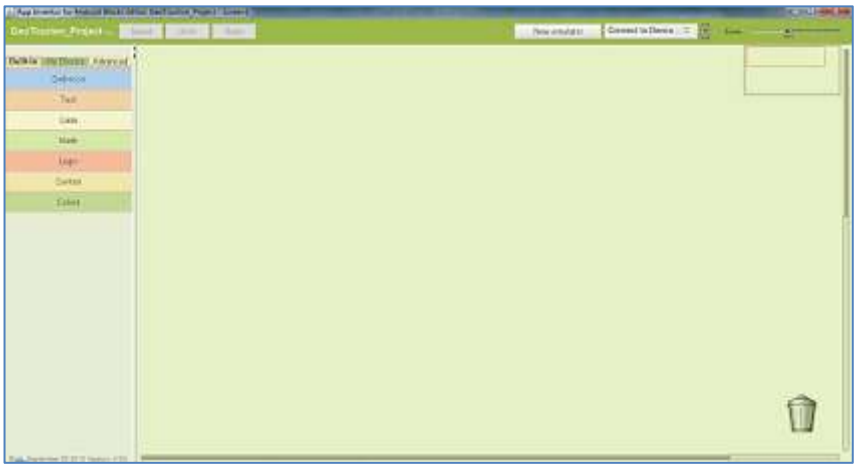
Gambar 2.14 Tampilan komponen *Properties* pada *Designer*

Main Menu digunakan sebagai antarmuka untuk mengakses menu utama pada *App Inventor*. Menu-menu yang terdapat pada *main menu* digunakan untuk menyimpan *project, debugging project* dan untuk memaket *project* menjadi sebuah aplikasi.

Gambar 2.15 Tampilan komponen *Main Menu* pada *Designer*

- *Block Editor*

  Block editor digunakan untuk merancang *behavior* pada aplikasi yang dibuat. *Block editor* terbagi menjadi beberapa bagian seperti *main menu, block palette, block designer,* dan *zoom panel.*

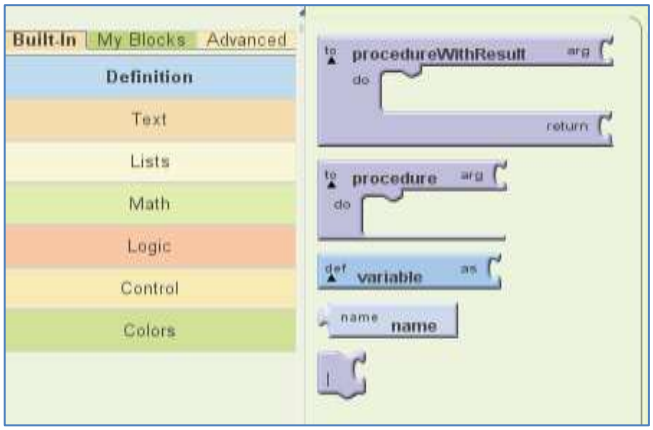


Gambar 2.16 Tampilan komponen *Block Editor*

Main Menu digunakan sebagai akses menu utama pada aplikasi *block editor*. *Main menu* dapat mengakses *emulator*, menyimpan *project* dan mengatur perangkat yang terhubung.

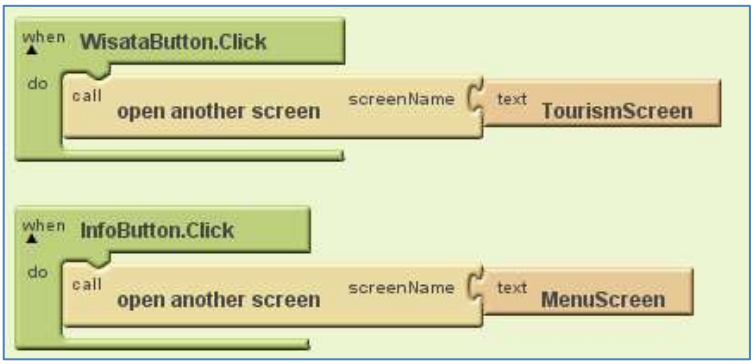
Gambar 2.17 Tampilan komponen *Main Menu* pada *Block Editor*

Block Palette digunakan untuk mengambil *part block* yang akan diletakkan pada *block designer*. Untuk meletakkan *part block* yang diinginkan dengan cara men-*drag* komponen *part block* ke *block designer*.



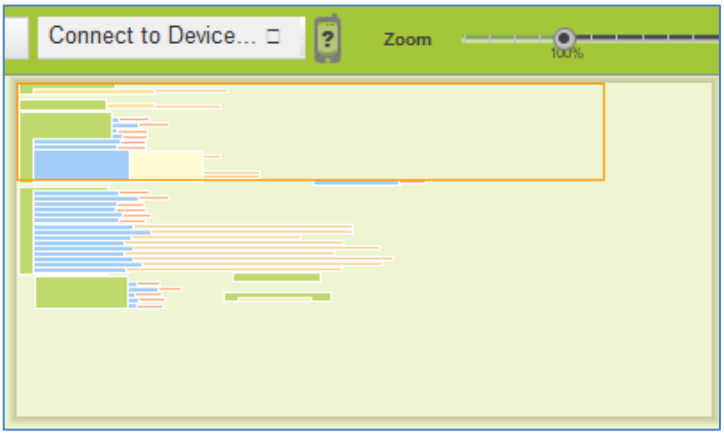
Gambar 2.18 Tampilan komponen *Block Palette* pada *Block Editor*

Block designer digunakan untuk meletakkan *part block* yang seterusnya akan digunakan sebagai program utama aplikasi. Penggunaan bahasa *visual block* membuat pemrograman menjadi tidak begitu sulit.



Gambar 2.19 Tampilan komponen *Block Designer* pada *Block Editor*

Zoom Panel digunakan untuk mengatur tampilan pada *block designer*, sehingga kita dapat memperbesar atau memperkecil tampilan dari *block editor* (Wahana Komputer, 2013).



Gambar 2.20 Tampilan komponen *Zoom Panel* pada *Block Editor*

II.7.3 Java

Java adalah suatu *platform* teknologi yang dikembangkan oleh Sun Microsystem sekitar tahun 90-an. Java bukan sekedar bahasa pemrograman, tetapi merupakan suatu sistem *platform* yang disediakan oleh Java untuk membangun suatu sistem baik dari yang berskala kecil (*game, workstation program, enterprise* sampai ke *mobile device*) dapat dibangun dengan *platform* yang disediakan oleh Sun tersebut. Java sebagai suatu platform dapat dibagi menjadi:

a. Bahasa pemrograman Java

   Bahasa pemrograman Java adalah suatu bahasa yang murni *Object Oriented Programming*. Semua kriteria OOP terhadap di dalam Java antara lain :

   - Abstraksi data dan enkapsulasi (*public, private, protected*).
   - *Inheritance.*
   - Polimorfism.

b. JVM (*Java Virtual Machine*).

   Dengan JVM ini maka semboyan Java yaitu "*write once run everywhere*" dapat direalisasikan dimana dengan JVM ini suatu program tidak lagi tergantung terhadap *platform* OS yang digunakan dan berinteraksi dengan OS. Kompilasi terhadap suatu *file* Java (*Source*) akan menghasilkan suatu *file byte code* (*extention class*) dimana *byte code java* adalah sama untuk

semua *platform*, sehingga ketika menjalankan sebuah program Java. JVM yang akan meng-*handle* segala sesuatu yang berhubungan dengan OS dan menjalankan *byte code* yang telah dihasilkan.

c. Java *Basic* API (J2SDK).

Java *Basic* API adalah sekumpulan *class* yang disediakan oleh java untuk mempermudah melakukan proses pengembangan terhadap aplikasi java . JDK (*Java Development Kit*) dapat diunduh dari *website* Sun (java.sun.com).

II.8 *Haversine Formula*

Haversine Formula merupakan sebuah persamaan yang penting dalam navigasi, dimana formula ini memberikan jarak di antara dua titik pada lingkaran bola dari setiap garis bujur *(longitude)* dan garis lintang *(latitude)*. Ini adalah kasus khusus dari sebuah formula yang lebih umum dalam trigonometri lingkaran bola, *haversine formula* menghubungkan sisi dan sudut dari sebuah segitiga bola.

Haversine formula nantinya akan digunakan dalam perhitungan jarak antara dua titik GPS. Dalam hal ini adalah titik GPS *user* dan titik GPS tujuan, sehingga dapat menjadi kunci utama dalam perbandingan jarak pada penentuan jarak terdekat. Rumus *Haversine* yang berlaku pada setiap titik pada lingkaran bola :

$$haversin\left(\frac{d}{r}\right) = haversin\,(\emptyset_2 - \emptyset_1) + cos(\emptyset_1)\,cos(\emptyset_2)\,haversin(\lambda_2 - \lambda_1) \ldots\ldots(2.1)$$

dimana haversin adalah *haversine formula* :

$$haversine(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1-\cos(\theta)}{2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.2)$$

Keterangan :

d = Jarak antara dua titik

r = radius dari lingkaran bulat

ϕ_1, ϕ_2 = latitude dari titik 1, latittude dari titik 2

λ_1, λ_2 = longitude dari titik 1, longitude dari titik 2

Pada sisi sebelah kiri tanda sama dengan, argumen fungsi haversin adalah dalam radian. Dalam derajat, haversin (d/R) dalam rumus akan menjadi haversin (180^o d / πR).

Kemudian untuk nilai d dapat diterapkan *haversin invers* (jika tersedia) atau dengan menggunakan fungsi *arcsin (Invers* sinus) :

$$d = r\ haversin^{-1}(h) = 2r \arcsin(\sqrt{h}\,) \quad \text{..... (2.3)}$$

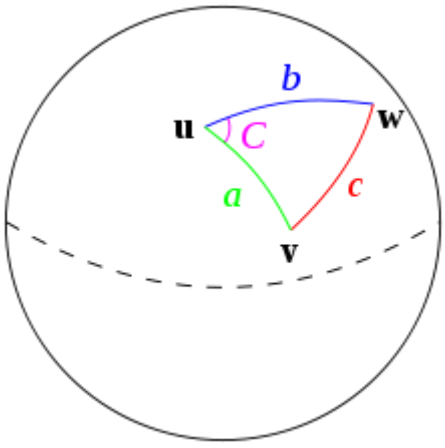
Dimana *h* adalah haversin (*d/R*)

II.8.1 Hukum Haversine

Dalam sebuah unit bola, sebah segitiga pada permukaan bola didefinisikan oleh lingkaran besar yang menghubungkan tiga titik yaitu u, v, dan w pada bola. Jika panjang dari tiga sisi adalah (dari u ke v), b (dari u ke w), dan c (dari v ke w), dan sudut sudut yang berlawanan dari c adalah C, maka hukum haversines adalah sebagai berikut :

$$haversin(c) = haversin(a - b) + \sin(a)\sin(b)\,haversin(C) \quad \text{..... (2.4)}$$

Karena ini adalah sebuah unit lingkaran bola, sehingga panjang a, b, dan c hanya sama dengan sudut (dalam radian) berdasarkan pada sisi-sisi dari pusat lingkaran bola (pada lingkaran tak penuh, masing-masing panjang busur sama dengan sudut pusat dikalikan dengan jari-jari bola).



Gambar 2.21 Segitiga bola yang diselesaikan dengan hukum haversine (Wikipedia, 2013)

Untuk mendapatkan rumus haversine dari bagian sebelumnya, secara sederhana kita mempertimbangkan sebuah kasus khusus di mana *u* adalah kutub utara, sementara *v* dan *w* adalah dua titik yang dipisahkan oleh *d* yang akan ditentukan. Dalam hal ini, a dan b adalah π/2 - φ1, 2 (yaitu, 90 ° - lintang), C adalah selisih bujur Δλ, dan c adalah d/R yang diinginkan. Dimana sin(π / 2 - φ) = cos (φ), maka rumus haversine dapat segera mengikuti.

Untuk menurunkan hukum haversines, saat dimulai dengan hukum bola cosinus:

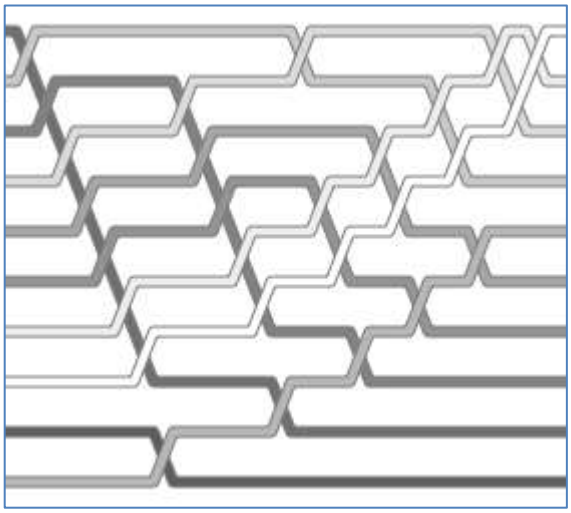
$$\cos(c) = \cos(a)\cos(b) + \sin(a)\sin(b)\cos(C) \quad \text{..... (2.5)}$$

Sebagaimana disebutkan di atas, rumus ini digunakan untuk mendapatkan nilai c. Sebaliknya, kita dapat mengganti persamaan $\cos(\theta) = 1 - 2\,\text{haversin}(\theta)$, dan juga melakukan persamaan tambahan cos (a − b) = cos(a) cos(b) + sin(a) sin(b), untuk memperoleh hukum haversine di atas (Wikipedia, 2013).

II.9 *Bubble Sort Algorithm*

Sebuah algoritma penyusun adalah sebuah algoritma yang menempatkan unsur-unsur dari sebuah daftar dalam urutan tertentu. Perintah yang paling sering digunakan adalah penyusunan pada angka (numerik) dan pengyusunan berdasarkan huruf. Penyusunan yang efisien sangatlah penting untuk mengoptimalkan penggunaan algoritma lain (seperti pencarian dan penggabungan algoritma) yang membutuhkan input data untuk berada dalam daftar yang telah diurutkan, melainkan juga sering berguna untuk menghindari duplikat data dan untuk menghasilkan *output* yang dapat dimengerti pengguna.

Algoritma *Bubble Sort*, adalah sebuah algoritma penyusunan sederhana yang bekerja dengan cara melakukan pembandingan berulang-ulang terhadap daftar yang akan di sortir (disusun), yaitu membandingkan setiap pasangan item yang berdekatan dan menukarnya jika berada di urutan yang tidak tepat, sehingga nantinya dapat menunjukkan bahwa daftar tersebut telah diurutkan. Karena hanya menggunakan perbandingan pada saat pengoperasiannya, maka ini adalah semacam penyusunan berdasarkan perbandingan dari data yang ada.



Gambar 2.22 Algoritma Penyusun *Bubble Sort* (Wikipedia, 2013)

II.9.1 Cara Kerja

Bubble sort memiliki *worst-case* dan rata-rata kompleksitas kedua O (n2), dimana n adalah jumlah item yang diurutkan. Ada banyak algoritma penyusun dengan kompleksitas *worst-case* atau rata-rata jauh lebih baik dari O (n log n). Bahkan algoritma pengurutan O (n2) lainnya, seperti *insertion sort*, cenderung memiliki kinerja yang baik seperti *bubble sort*.

Satu-satunya keuntungan yang signifikan adalah *bubble sort* memiliki lebih banyak implementasi dari yang lain, seperti *quicksort* yaitu sebuah kemampuan untuk mendeteksi bahwa daftar telah diurutkan secara efisien dan dibangun ke dalam sebuah algoritma. Kinerja *bubble sort* terhadap daftar yang sudah diurutkan (*best-case*) adalah O (n) . Sebaliknya , kebanyakan algoritma lain bahkan memberi kompleksitas *average-case* dengan baik, melakukan seluruh proses penyusunan pada sebuah daftar dan tetlihat lebih kompleks. Namun, *insertion sort* memiliki mekanisme yang sama, tetapi juga dapat melakukan kinarja lebih baik pada daftar yang diurutkan secara substansial.

Penyusunan pada *bubble sort* dapat dicontohkan sebagai berikut, misalkan terdapat urutan angka " **5 1 4 2 8** ". Dalam setiap langkah penyusunan, unsur-unsur angka yang ditulis dalam huruf tebal dan bergaris bawah akan dibandingkan satu sama lain.

Langkah Pertama :

- (**5 1** 4 2 8) menjadi (**1 5** 4 2 8)
  disini, algoritma membandingkan dua elemen pertama, dan menukarnya posisi keduanya, karena 5 > 1.
- (1 **5 4** 2 8) menjadi (1 **4 5** 2 8)
  menukarnya posisi keduanya, karena 5 > 4.
- (1 4 **5 2** 8) menjadi (1 4 **2 5** 8)
  menukarnya posisi keduanya, karena 5 > 2.
- (1 4 2 **5 8**) menjadi (1 4 2 **5 8**)
  dalam hal ini tidak terjadi pertukaran, karena (8 > 5) maka posisi keduanya tetap.

Langkah Kedua

- (**1 4** 2 5 8) menjadi (**1 4** 2 5 8)
- (1 **4 2** 5 8) menjadi (1 **2 4** 5 8) , menukarnya posisi keduanya, karena 4 > 2
- (1 2 **4 5** 8) menjadi (1 2 **4 5** 8)
- (1 2 4 **5 8**) menjadi (1 2 4 **5 8**)

Sekarang, barisan angka sudah disortir, tapi *bubble sort* algoritma belum mengetahui apakah proses penyusunannya telah selesai. Algoritma ini membutuhkan satu pengecekan kembali untuk memastikan apakah semua telah disusun sesuai sengan posisi seharusnya (Wikipedia, 2013).

Langkah Ketiga

- (**1 2** 4 5 8) menjadi (**1 2** 4 5 8)
- (1 **2 4** 5 8) menjadi (1 **2 4** 5 8)
- (1 2 **4 5** 8) menjadi (1 2 **4 5** 8)
- (1 2 4 **5 8**) menjadi (1 2 4 **5 8)**

II.10 *Review* Penelitian

Sistem operasi Android memiliki keunggulan tersendiri dibanding beberapa sistem operasi lainnya seperti *Blackberry* milik *Research In Motion* (RIM) dan juga *iOS* milik Apple.Inc. Keunggulan tersebut antara lain bebas dalam

pengembangan, *open source*, dan lengkap, sehingga membuat sistem operasi Android banyak diminati oleh pengembang aplikasi. Berikut adalah beberapa *review* mengenai aplikasi *smartphone* yang menggunakan sistem operasi Android.

1. ATLAS (*Alpha Version*)

   ATLAS (*Alpha Version*) merupakan aplikasi *culinary tour guide* berbasis *mobile* GIS yang terintegrasi dengan *Google Maps* dimana pengguna dapat mengakses informasi spasial maupun non spasial melalui peta dan fitur didalamnya.

   a. *Developer* : Arifah Trisnawati
   b. *Application* : *Eclipse Galileo*
   c. *OS Version* : *Eclair, Froyo (Frozen Yoghurt), Ginger Bread.*
   d. Fitur
      - Lokasi Terdekat
        Menampilkan list lokasi kuliner berserta jarak.
      - Peta Kuliner
        Menampilkan peta kuliner Semarang melalui *Google Maps.*
      - Menu Favorit
        Menampilkan informasi tempat makan favorit terdekat dari posisi pengguna.
      - Menu Kategori
        Menampilkan informasi tempat makan berdasarkan kategori yaitu restoran, rumah makan, dan kafe.
      - Informasi Lokasi
        Menampilkan informasi tempat secara detail yang mencakup nama, foto, *review*, alamat, nomor telepon, dan lain-lain.

2. GeoTrans

   GeoTrans merupakan aplikasi pedoman penggunaan bus Trans Jogja berbasis *Location Based Service* pada *smartphone* Android yang di intergrasikan dengan *Google Maps.*

a. *Developer* : Danang Budi Susetyo

b. Application : Eclipse Helios

c. OS Version : Ginger Bread (2.3.4), Ginger Bread (2.3.6), Ice Cream Sandwich (4.0.4)

d. Fitur

- Lokasi *shelter* Terdekat

  Menampilkan lokasi *shelter* terdekat dari posisi *user* dengan fitur *direction.*

- Menu *Destination*

  Menampilkan *list shelter-shelter* terdekat dari beberapa lokasi penting di Yogyakarta.

- Menu *Input*

  Menampilkan *screen* untuk meng-*input* lokasi *shelter* awal ke *shelter* tujuan.

- Menu *Shelter*

  Menampilkan keseluruhan jalur dari rute bus Trans Jogja dalam beberapa pilihan koridor.