BAB II

LANDASAN TEORI

2.1 Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabungkan bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*suite application*). Contohnya adalah Microsoft Office dan OpenOffice.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.2 Fasilitas Umum

Fasilitas umum adalah barang yang dikuasai negara, dibiayai sebagian atau seluruhnya oleh anggaran dan belanja negara yang pemakaiannya atau peruntukkan-nya oleh pemerintah atau negara bagi umum. Dalam kamus besar bahasa Indonesia, fasilitas umum adalah fasilitas yang disediakan untuk kepentingan umum, seperti jalan dan alat penerangan umum. Fasilitas umum dibangun untuk menunjang kebutuhan masyarakat pada suatu daerah, seperti rumah sakit, ATM, stasiun pengisian bahan bakar umum (SPBU), dan lain sebagainya.

2.3 Data dan Informasi

Hubungan antara data dan informasi sangatlah erat sebagaimana hubungan antara sebab dan akibat. Bahwa data merupakan bentuk dasar dari sebuah informasi, sedangkan informasi merupakan elemen yang dihasilkan dari suatu bentuk peng-olahan data.

2.3.1 Data

Secara konseptual, data adalah deskripsi tentang benda, kejadian, aktivitas dan transaksi, yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. Data sering kali disebut sebagai bahan mentah informasi.

Berikut adalah kutipan pengertian data dari sudut pandang yang berbeda :

1. Menurut kamus bahasa Inggris-Indonesia, data diterjemahkan sebagai istilah yang berasal dari kata *"datum"* yang berarti fakta atau bahan-bahan keterangan.

2. Dari sudut pandang bisnis, terdapat pengertian data bisnis sebagai berikut: *"Bussines data is an organization's description of things (resources) and events (transactions) that it faces"*. Jadi data, dalam hal ini disebut sebagai data bisnis, merupakan deskripsi organisasi tentang sesuatu (*resource*) dan kejadian (*transaction*) yang terjadi.
3. Menurut (Jogiyanto, HM, 1989 : 8) data merupakan bentuk jamak dari datum (kenyataan) yang berupa fakta-fakta, angka-angka, gambar-gambar yang dapat ditarik kesimpulannya.
4. Gordon B. Davis dalam bukunya *Management Informations System : Conceptual Foundations, Structure, and Development* menyebut data sebagai bahan mentah dari informasi, yang dirumuskan sebagai sekelompok lambang-lambang tidak acak yang menunjukkan jumlah atau tindakan atau hal-hal lain.

Dari keempat pengertian diatas, dapat diambil kesimpulan bahwa data adalah bahan baku informasi, didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, benda, dan sebagainya. Data terbentuk dari karakter, dapat berupa alphabet, angka, maupun simbol khusus seperti *, $, dan /. Data disusun untuk diolah dalam bentuk struktur data, struktur file, dan basis data.

2.3.2 Informasi

Informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi ada suatu proses transformasi data menjadi suatu informasi (input – proses – output).



Gambar II.1 Pemrosesan data menjadi informasi

Definisi umum untuk informasi dalam sistem informasi menurut Jogiyanto H.M (1990; 11) : "Informasi adalah bentuk data yang dapat diolah yang lebih berguna dan berarti bagi yang menerimanya". Menurut Robert G. Munik (1973 ; 12) : "Informasi adalah data yang telah diolah menjadi suatu bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.

Berikut pengertian informasi dari berbagai sumber :

1. Menurut Gordon B. Davis dalam bukunya "*Management Informations System: Conceptual Foundations, Structures, and Development* menyebut informasi sebagai data yang telah diolah menjadi bentuk yang berguna bagi penerimanya dan nyata, berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan.
2. Menurut Berry E. Cushing dalam bukunya*"Accounting Information System and Business Organization"* dikatakan bahwa

informasi merupakan sesuatu yang menunjukkan hasil pengolahan data yang diorganisasi dan berguna kepada orang yang menerimanya.

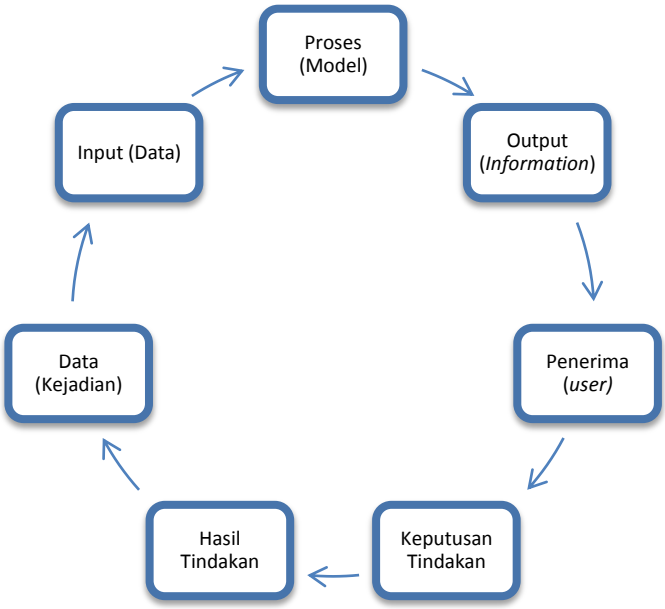
3. Menurut Robert N. Anthony dan John Dearden dalam bukunya *"Managemet Control Systems"*, menyebut informasi sebagai suatu kenyataan, data, item, yang menambah pengetahuan bagi penggunanya.
4. Menurut Stephan A. Moscove dan Mark G. Simkin dalam bukunya *"Accounting Information Systems: Concepts and Practise"* mengatakan informasi sebagai kenyataan atau bentuk-bentuk yang bergunay yang dapat digunakan untuk pengambilan keputusan bisnis.

Dari keempat pengertian seperti tersebut di atas dapat disimpulkan bahwa informasi hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan.

2.3.3 Siklus Informasi

Untuk memperoleh informasi yang bermanfaat bagi penerimanya, perlu untuk dijelaskan bagaimana siklus yang terjadi atau dibutuhkan dalam menghasilkan informasi. Pertama-tama data dimasukkan ke dalam model yang umumnya memiliki urutan proses tertentu dan pasti, setelah diproses akan dihasilkan informasi tertentu yang membuat suatu keputusan atau melakukakn tindakan tertentu. Dari keputusan atau tindakan tersebut akan

menghasilkan atau diperoleh kejadian-kejadian tertentu yang akan digunakan kembali sebagai data yang nantinya akan dimasukkan ke dalam model (proses), begitu seterusnya. Dengan demikian akan membentuk suatu siklus informasi (*information cycle*) atau siklus pengolahan data (*data processing cycles*), seperti gambar berikut :



Gambar II.2 Siklus Informasi

2.4 Augmented Reality

Realitas tertambah atau yang dikenal dengan *augmented reality* (AR), adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata. Tidak seperti realitas maya (*virtual reality*) yang sepenuhnya menggantikan kenyataan, realitas tertambah sekedar menambahkan atau melengkapi kenyataan.

Benda-benda maya menampilkan informasi yang tidak dapat diterima oleh pengguna dengan inderanya sendiri. Hal ini membuat realitas tertambah sesuai sebagai alat untuk membantu persepsi dan interaksi penggunanya dengan dunia nyata. Informasi yang ditampilkan oleh benda maya membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata.

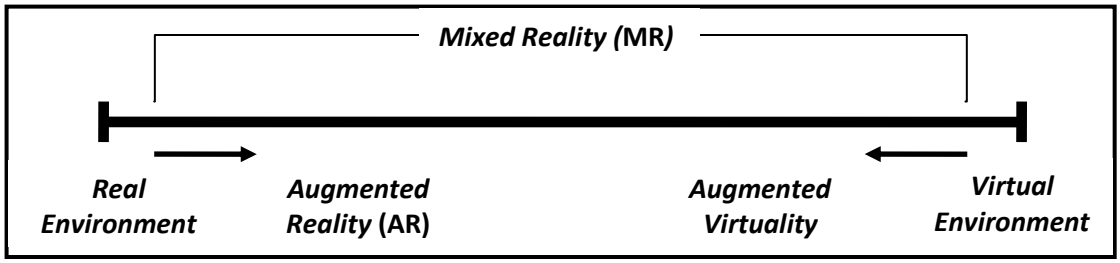
Realitas tertambah dapat diaplikasikan untuk semua indera, termasuk pendengaran, sentuhan, dan penciuman. Selain digunakan dalam berbagai bidang seperti bidang kesehatan, militer, industri manufaktur, realitas tertambah juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada telepon genggam.

2.4.1 Pengertian *Augmented Reality*

Ronald T. Azuma (1997) mendefinisikan *augmented reality* sebagai penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata dan terdapat integrasi antar-benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat input tertentu, dan integrasi yang baik memerlukan penjejakan yang efektif. Selain menambahkan benda maya dalam lingkungan nyata, realitas tertambah juga berpontensi menghilangkan benda-benda yang sudah ada. Menambahkan sebuah lapisan gambar maya dimungkinkan untuk menghilangkan atau menyembunyikan lingkungan nyata dari pandangan pengguna.

2.4.2 *Virtual Continuum*

Milgram dan Kishino (1994) merumuskan kerangka kemungkinan peng-gabungan dan peleburan dunia nyata dan dunia maya ke dalam sebuah *continuum virtualitas*. Sisi yang paling kiri adalah lingkungan nyata yang hanya berisi benda nyata, dan sisi paling kanan adalah lingkungan maya yang berisi benda maya.



Gambar II.3 Milgram*'s Reality-Virtuality Continuum*

Dalam realitas tertambah (*augmented reality*), yang lebih dekat ke sisi kiri, lingkungan bersifat nyata dan benda bersifat maya, sementara dalam *augmented virtuality* atau virtualitas tertambah, yang lebih dekat ke sisi kanan, lingkungan bersifat maya dan bersifat nyata. Realitas tertambah dan virtualitas tertambah digabungkan menjadi *mixed reality* atau realitas campuran.

2.4.3 Sejarah *Augmented Reality*

1936

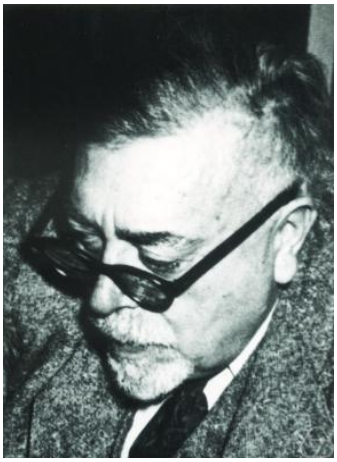
Konrad Zuse menciptakan komputer digital pertama dikenal dengan nama Z1 (*computer*).



Gambar II.4. Konrad Zuse

1948

Nobert Wiener menciptakan ilmu *cybernetics* : mentransmisikan pesan diantara manusia dan mesin



Gambar II.5. Nobert Wiener

1962

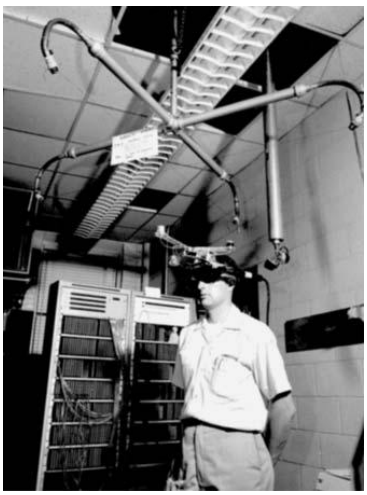
Morton Heileg, seorang *cinematographi*, menciptakan kendaraan simulator dengan nama *Sensorama*, memiliki kemampuan visual, suara, getaran (*vibrate*), dan penciuman (*smell*).



Gambar II.6 Morton Heileg

1966

Ivan Sutherland menciptakan perangkat *head-mounted display*, sebagai jendela terhadap dunia maya (*virtual world*).



Gambar II.7 Ivan Sutherland

1968

Ivan Sutherland menciptakan sistem *augmented reality* pertama, sekaligus menjadi sistem *virtual reality* pertama, yang dikenal sebagai "*Sword of Damocles*". Sistem ini menggunakan *optical see-through* HMD (*head-mounted display)* yang melacak menggunakan satu dari dua jenis

pelacakan 6DOF : pelacak mekanik dan ultrasonik. Karena keterbatasan kemampuan komputer pemroses pada saat itu, hanya sebuah gambar *frame* sederhana yang dapat ditampilkan dalam waktu nyata (*realtime*).

1975

Myron Krueger menciptakan *Videoplace*, pengguna mampu atau dapat berinteraksi dengan obyek maya (*virtual object*) untuk pertama kali.

1982

Komputer jinjing pertama (laptop), bernama Grid Compass 1100, komputer pertama yang menggunakan desain *clamshell*.



Gambar II.8 Grid Compass 1100

Grid Compass 1100 dilengkapi Intel 8086 CPU, memori 350 KB, dan sebuah layar dengan resolusi 320x240 *pixel*, menjadikannya komputer yang sangat bertenaga (*powerfull*) pada saat itu, dengan harga yang sangat besar, $10.000 USD. Beratnya hanya lima (5) kg sehingga dapat dibawa kemana-mana.

1989

Jaron Lanier memperkenalkan *virtual reality* dan menciptakan bisnis komersial yang pertama kali dalam dunia *virtual*.

1992



Gambar II.9 Tom Caudell

Tom Caudell dan David Mizell memperkenalkan istilah *augmented reality* ketika Boeing membantu pekerja memasang kabel dalam pesawat terbang. Penggunaan istilah ini berdasarkan pelapisan material hasil komputerisasi kedalam dunia nyata. Caudell dan Mizell mendiskusikan keuntungan dari *augmented reality* dibandingkan dengan *virtual reality* seperti mengurangi tenaga pemrosesan saat pixels dirender. Mereka juga mengakui adanya peningkatan kebutuhan registrasi (*registration*) ketika menyelaraskan dunia nyata dan maya (*virtual*).

1993

Loomis dkk, mengembangkan sebuah *prototype* dari sistem navigasi outdoor untuk gangguan visualiasasi. Mereka mengkombinasikan notebook dengan alat penerima GPS dan *compass electronic*. Pengaplikasiannya menggunakan data dari database GIS (*Geographics Information System*) dan menyediakan sisten navigasi yang menggunakan

sebuah layar *virtual acoustic* (*acoustic virtual display*) : label berbicara menggunakan pensintesis suara (*speech synthesizer*) dan diputar kembali pada lokasi yang tepat didalam ruang pendengaran pengguna.

1994

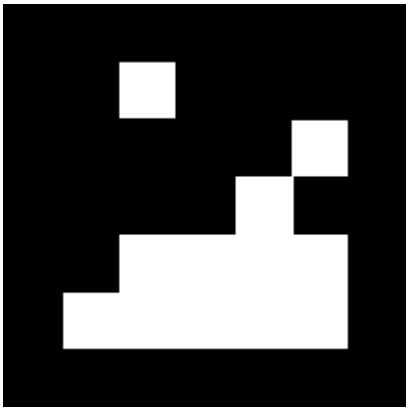
Steve Mann mulai menggunakan sebuah *webcam* selama dua tahun. Dari tahun 1994-1996, Mann memakai kamera beserta sebuah layar untuk setiap menitnya. Kedua alat tersebut terhubung ke website membuat setiap orang yang berkunjung ke website tersebut dapat melihat apa yang Steve lihat dan mengirimi pesan bahwa video tersebut berasal dari perangkat mobile.

1995

Jun Rekimoto dan Katashi Nagao menciptakan NaviCam, sebuah pengaturan tambatan (*tethered*), mirip *Fitzmaurice's Chameleon*. NaviCam juga menggunakan tenaga *workstation* terdekat, memiliki kamera disisi atas layar *mobile* yang digunakan untuk pelacakan *optical*. Komputer mendeteksi penanda kode warna dalam gambar yang tertangkap kamera dan menampilkan konteks informasi penting secara langsung yang melapisi video yang tertangkap oleh kamera.

1996

Jun Rekimoto memperkenalkan penanda (*marker*) matrix 2D (barcode berbentuk kotak), salah satu dari sistem penanda (*marker system*) pertama yang menggunakan pelacak kamera dengan 6DOF (*six degrees of freedom*).



Gambar II.10 Marker Matrix 2D

1997

Ronald Azuma memperkenalkan survei pertama pada *augmented reality*. Dalam publikasinya, Azuma memberikan definisi luas untuk AR, seperti di-identifikasikan dalam tiga karakter: menggabungkan dunia nyata dan maya; interaktif dalam waktu nyata (*realtime*); terdaftar (*registered*) dalam 3D.

Steve Feiner dkk, memperkenalkan *touring machine*, sistem *mobile* pertama dengan *augmented reality* (MARS = *Mobile Augmented Reality System*).

Thad Starner dkk, mengeksplorasi kemungkinan pengaplikasian dari *augmented reality*, membentuk sebuah komunitas kecil pengguna komputer yang dapat saling berhubungan melalui jaringan.

Philippe Kahn menciptakan kamera telepon, sebuah telepon genggam yang mempunyai kemampuan fotografi. Pada tahun 1997, Kahn menggunakan inovasinya untuk membagikan (*share*) gambar kelahiran putrinya kepada lebih dari 2000 relasi dan teman-temannya di seluruh

dunia. Sekarang, lebih dari setengah dari keseluruhan telepon genggam dilengkapi kamera.

1998

Bruce Thomas dkk, memperkenalkan "*Map-in-the-hat"*, sebuah tas punggung dilengkapi komputer dan GPS, kompas elektronik, dan *head mounted display*. Sistem ini digunakan untuk penuntun (*guidance*) navigasi, kemudian melibatkannya kedalam "Tinmith", platform AR yang telah digunakan pada beberapa projek AR sebelumnya.

1999

Hirokazu Kato dan Mark Billinghurst memperkenalkan ARToolkit, sebuah *library* pelacakan gerakan dengan 6DOF (*six degrees of freedom*), menggunakan *fiducial* persegi dengan pendekatan berbasis template untuk pengenalan. ARToolkit tersedia bebas (*open source*) dibawah lisensi GPL dan sampai sekarang masih popular digunakan.

2000

Bruce Thomas dkk, memperkenalkan AR-Quake, sebuah permainan komputer desktop yang popular. AR-Quake adalah aplikasi pertama kali yang berbasis sistem pelacakan 6DOF dilengkapi GPS, *digital compass*, dan pelacak berbasis penglihatan (*vision-based*) dari penanda *fiducial*.

2001

Joseph Newman dkk, memperkenalkan BatPortal, sebuah PDA berbasis *wireless* sistem AR. Penentuan lokasi dilakukan dengan mengukur waktu perjalanan *pulse ultrasonic* diantara perangkat yang

digunakan pengguna (disebut Bats), dan instalasi penerima yang terdapat pada langit-langit bangunan. Sistem ini mendukung sistem berbasis HMD, yang lebih dikenal sebagai BatPortal digunakan pada perangkat genggam. Berdasarkan pada konfigurasi akhir dari PDA dan Bat, arah pandangan pengguna diukur, dan model dari objek yang terlihat dilengkapi informasi tambahan tentang objek tersebut kemudian direnderkedalam layar PDA.

2002

Michael Kalkusch dkk, memperkenalkan sistem *mobile augmented reality* untuk membimbing (*guide*) pengguna dalam suatu bangunan menuju ke suatu ruangan. Sistem ini menyajikan model bangunan yang telah terdaftar dengan label informasi penuntun arah yang terlihat dalam layar, dan peta miniatur dunia tiga dimensi (*world in miniature*) pada pada pergelangan tangan yang dikenakan bertindak sebagai perangkat input. Pelacakan berhasil dilakukan dengan menggunakan kombinasi penanda ARToolkit pada dinding bangunan melalui kamera (*head-mounted camera*) dan pelacak *inertial*.

2003

Adrian David Cheok dkk, mengenalkan "*Human Pacman*". *Human Pacman* adalah lingkungan interaktif dengan sistem hiburan *mobile* yang terbentuk berdasarkan posisi dan perspektif melalui GPS dan sensor inertia; dan berwujud manusia hasil komputerisasi (*human-computer*) diinteraksikan dengan menggunakan *Bluetooth* dan sensor *capacitive*. Pacmen dan Ghost adalah permainan nyata dalam pengalaman dunia nyata yang menggabungkan dunia nyata dan grafik fantasi hasil komputerisasi,

dimainkan dengan menggunakan komputer yang dilengkapi GPS dan sensor inertia untuk posisi pemain dan pelacakan perspektif. Objek maya dan objek fisik yang nyata dengan perangkat Bluetooth dan sensor *capacitive* yang dimasukkan kedalam *game* untuk memberikan pengalaman dari transisi antara dunia nyata dan dunia maya.

2004

Mathias Mohring dkk, mengenalkan sebuah sistem untuk pelacakan penanda tiga dimensi pada telepon genggam. Karya ini menunjukkan video dengan sistem *augmented reality* pada telepon genggam pengguna. Sistem ini mendukung deteksi dan deferensiasi dari perbedaan penanda tiga dimensi, dan ketepatan integrasi dari grafik tiga dimensi yang dirender kedalam video streaming.

2005

Anders Henrysson menghubungkan ARToolkit dengan Symbian. Berdasarkan teknologi ini Anders memperkenalkan permainan AR-Tennis, kolaborasi pertama aplikasi AR yang berjalan pada telepon genggam. AR-Tennis mendapat penghargaan terbaik pada kontes *mobile gaming* pada tahun 2005, dan penghargaan prestasi teknikal.

2006

Reitmayr dkk, mengenalkan model berbasis sistem pelacakan *hybrid* untuk *augmented reality outdoor* pada lingkungan perkotaan dengan akurasi, pelapisan waktu nyata (*realtime*) pada perangkat genggam. Sistem ini mengkombinasikan berdasarkan sisi pelacakan untuk akurasi penentuan lokasi, pengukuran *gyroscope* untuk pergerakan,

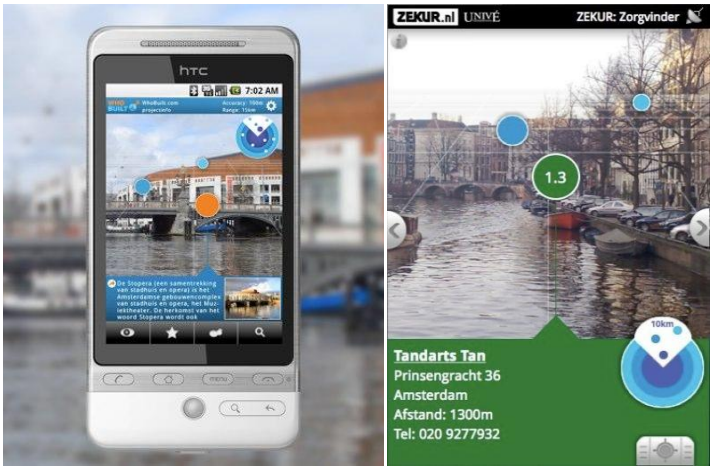
pengukuran gravitasi dan medan magnet untuk mencegah gangguan, dan mereferensi *frame* dengan seleksi frame online untuk menginisialisasi ulang setelah dinamis okulasi atau kegagalan.

2007

Klein dan Murray mengenalkan sistem yang mampu melacak secara *realtime* dan pemetaan secara paralel dengan kamera *monocular* dalam lingkungan yang kecil. Sistem ini diadopsi dari pendekatan SLAM dimana proses tugas pelacakan dan pemetaan dipisahkan menjadi dua bagian.

2009

Morrison dkk, mengenalkan MapLens, sebuah peta *mobile augmented reality* yang menggunakan lensa *magic*. Mereka melakukan studi pada pengguna yang luas dalam bentuk permainan berbasis lokasi. Penemuan utama mereka adalah fitur AR memfasilitasi pembuatan tempat dengan menciptakan kebutuhan konstan untuk referensi terhadap bentuk fisik objek. Dalam percobaannya menunjukkan potensial dari keberadaan peta *augmented reality* dalam penggunaanya sebagai *collaborative tool*.

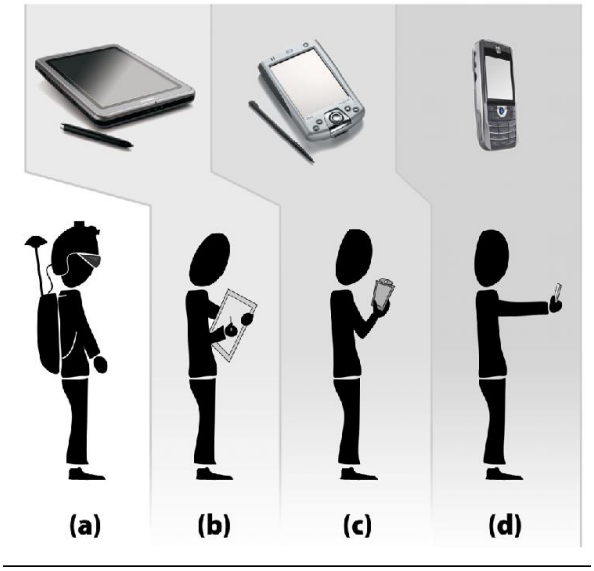


Gambar II.11 Tampilan aplikasi Wikitude (kiri) dan Layar (kanan)

SPRXmobile memperkenalkan *layar,* aplikasi tingkat lanjut dari *wikitude*. *layar* menggunakan metode mekanisme *registrasi* seperti pada *wikitude* (GPS + kompas), dan menggabungkannya kedalam platform *server* berbasis *open-client*. Konten dalam *layar* sama pada konten dalam halaman web yang digunakan dalam browser. Keberadaan *layar* termasuk w*ikipedia, twitter, brighkite* pada layanan lokal seperti *Yelp, Trulia, Store locators, nearby bus stops, mobile coupons, mazda dealers and tourist, nature and culture guides.* Tanggal 17 agustus, aplikasi *layar* men-global dengan layanan mencapai seratus (100) konten.

2.4.4 Perangkat Keras (*hardware*) *untuk Augmented Reality*

Pada dasarnya, terdapat tiga komponen utama yang dibutuhkan untuk menjalankan AR. Pertama, kamera untuk menangkap lingkungan nyata atau suatu objek nyata. Kedua, layar untuk menampilkan hasil akhir, dan yang terakhir adalah perangkat komputasi atau pemroses. Pada gambar 2.12 menunjukkan faktor-faktor yang berbeda pada perkembangan perangkat AR. Perangkat dari kiri ke kanan: Sebuah tas backpack dengan komputer dan HMD, tablet PC, PDA dan smartphone. Perbedaannya bervariasi bukan hanya dalam ukuran tapi juga dalam kemampuan pemrosesan.



Gambar II.12 (a) PC + *Head Mounted Display* (HMD), (b) Tablet PC, (c) PDA, (d) *Smartphone*

2.4.5 *Mobile Augmented Reality*

Rohr dan Gfeller mengenalkan konsep dari *mobile augmented reality*. Dibanding menggunakan perangkat khusus untuk membangun sebuah aplikasi AR, mereka mengusulkan untuk menggunakan generasi baru dari perangkat *portable*. Peningkatan pada kemampuan komputasi, resolusi kamera pada perangkat *portable* memungkinkan pengimplementasian aplikasi AR.

Dalam proses membangun aplikasi AR, Rohs menyederhanakan tugas pengenalan tempat pada suatu gambar dengan menggunakan penanda *fiducial*. Penanda fiducial berbentuk persegi 2 dimensional yang terdiri dari area hitam dan putih. Selanjutnya, aplikasi mencari bentuk spesifik pada layar perangkat. Dari penanda *fiducial*, aplikasi dapat menentukan keberadaan pada layar, orientasi dan skala.

Mobile augmented reality adalah area penting dalam lingkungan penelitian yang mempunyai potensi dan kemungkinan untuk dibangun aplikasi komersial. Menggunakan *hardware* yang umum, sehingga murah bagi pengguna akhir.

2.4.6 Teknik Tampilan *Augmented Reality*

Terdapat tiga teknik tampilan pada *augmented reality* yaitu *head-mounted display*, *handheld displays*, dan *spatial display*.

1. *Head-mounted display*

   *Head-mounted display (*HMD*)* menempatkan gambar diantara dunia nyata dan objek grafik *virtual* melalui pandangan user terhadap dunia nyata. *Head-mounted display* terbagi menjadi dua bagian yaitu *optical see-through* dan *video see-through. Optical see-through* biasanya menempatkan sebuah *semi-silvered mirror* sebelum mata pengguna. Pengguna dapat melihat dunia nyata melalui *mirror* (cermin), dan juga melihat grafik komputer digambarkan pada layar miniatur yang tampak pada refleksi cermin. Proses ini mempunyai efek grafik seperti munculnya objek hitam transparan terhadap pengguna, memberikan pandangan tanpa modifikasi dari objek nyata pada tempat yang sama.

   Video see-through, pandangan pengguna tidak secara langsung terhadap dunia nyata tetapi hanya sebuah miniatur hasil komputerisasi yang nampak penuh dalam layar. HMD harus melacak dengan sensor yang menyediakan 6DOF (six degrees of

freedom). Pelacakan ini membuat sistem dapat menyelaraskan virtual informasi ke dunia nyata.

2. *Handheld display*

Handheld display bekerja dengan sebuah layar kecil yang pas atau sesuai dengan genggaman pengguna. *Handheld augmented reality* merupakan solusi untuk *video-see through*. Mulanya, teknik ini bekerja dengan penanda *fiducial*, dan kemudian GPS, dan sensor MEMS (*Microelectromechanical systems*) seperti kompas digital, *accelerometer*, dan *gyroscope*. Saat ini, pelacakan tanpa marker, yaitu SLAM (*Simultaneous localization and mapping*) seperti PTAM yang mulai digunakan. Keuntungan utama dari handheld AR adalah mudah digunakan, dapat dibawa kemana-mana (*portable*) dan telah dilengkapi kamera. Kekurangannya adalah untuk mendapat efek pelapisan yang baik dibutuhkan kestabilan perangkat untuk mencegah efek distorsi.

3. *Spatial display*

Spatial augmented reality (SAR) menggunakan proyektor digital untuk menampilkan informasi digital pada objek fisik (objek nyata). Perbedaan dari SAR dengan teknik lainnya adalah tampilan (*display*) dipisahkan dari pengguna sistem, karena tampilan tidak berasosiasi dengan setiap pengguna, melainkan kumpulan pengguna, memungkinkan menempatkan kolaborasi pengguna. Terdapat beberapa keuntungan penggunaan teknik SAR dibanding *head-mounted* dan *handheld display*. Pengguna tidak lagi

membawa atau menggunakan alat yang dikenakan pada mata. Sistem ini dapat digunakan oleh banyak orang dalam waktu bersamaan tanpa menggunakan HMD.

2.4.7 Komponen Augmented Reality

1 Kamera Telepon

Kamera telepon adalah telepon genggam (ponsel) yang mempunyai kemampuan untuk menangkap foto dan merekam video. Sejak awal abad 21, kamera banyak digunakan atau diimplementasikan pada telepon genggam.

Pada umumnya kamera telepon lebih simple dibandingkan dengan kamera *digital*. Kamera telepon biasanya dilengkapi dengan *fixed focus* (lensa tanpa pengaturan focus) dan sensor kecil yang membatasi kemampuannya dalam pencahayaan yang sedikit. Tidak memiliki flash atau pembesaran optic atau buat dudukan tripod.

Keuntungan dari pengguna kamera telepon adalah biaya dan bentuknya; terutama bagi orang yang sering membawa telepon genggam, perubahan ukuran dan tambahan biaya bukanlah menjadi persoalan besar. *Smartphone* dilengkapi kamera dapat menjalankan aplikasi yang mempunyai kemampuan *geotagging* dan *image stitching* (menggabungkan beberapa foto menjadi satu gambar panaroma).

2 GPS Navigation Device

Perangkat navigasi GPS adalah perangkat yang dapat menerima sinyal GPS (*Global Positioning System*) yang bertujuan untuk menentukan lokasi perangkat di bumi. Perangkat GPS memberikan informasi garis lintang (*latitude*) dan garis bujur (*longitude*), dan beberapa diantaranya menyertakan kalkulasi *altitude* (ketinggian), walaupun hal ini tidak selamanya akurat atau secara kontinue tersedia (terbatas pada kemungkinan sinyal terhalangi dan faktor lainnya) secara ekslusif diandalkan pada pilot pesawat terbang. Perangkat GPS telah digunakan dalam bidang militer, penerbangan, kelautan, dan produk aplikasi consumer.

Berdasarkan aturan terhadap pelacakan perangkat *mobile* (*mobile phone tracking*), termasuk E911 (*Enhanced 911*, evolusi modern dalam bidang telekomunikasi), mayoritas perangkat penerima GPS ditanam pada perangkat *mobile*, dengan variasi derajat jangkauan dan akses pengguna. Perangkat lunak navigasi komersial telah tersedia pada *smartphone* abad ke-21 serta perangkat telepon berbasis Java yang mengijinkan pelacakan perangkat mobile digunakan sebagai internal atau eksternal penerima GPS (dalam kasus ini, terhubung melalui serial atau *Bluetooth*). Beberapa telepon dengan kemampuan GPS bekerja hanya menggunakan A-GPS (*assisted GPS*), dan tidak berfungsi ketika perangkat berada diluar jangkauan tower penyedia jaringan.

3 Accelerometer

Accelerometer adalah sebuah tranducer yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan gravitasi bumi. *Accelerometer* juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan, bangunan, mesin, dan juga dapat digunakan untuk mengukur getaran yang terjadi dalam bumi, getaran mesin, jarak yang dinamis, dan kecepatan dengan ataupun tanpa pengaruh gravitasi bumi.

4 Digital Compass

Kompas adalah alat navigasi untuk menentukan arah berupa sebuah panah penunjuk magnetis yang bebas menyelaraskan dirinya dengan medan magnet bumi secata akurat. Kompas memberikan rujukan arah tertentu, sehingga sangat membantu dalam bidang navigasi. Arah mata yang ditunjuknya adalah utara, selatan, timur, dan barat. Apabila digunakan bersama-sam dengan jam dan sekstan, maka kompas akan lebih akurat dalam menunjukkan arah. Alat ini membantu perkembangan perdagangan maritime dengan membuat perjalanan jauh lebih aman dan efisien dibandingkan saat manusia masih berpedoman pada kedudukan bintang untuk menentukan arah.

Kompas dibedakan menjadi dua jenis, yaitu kompas analog dan kompas digital. Kompas analog adalah kompas yang biasa kita lihat dalam kehidupan sehari-hari. Misalnya saja kompas yang

dipakai ketika acara pramuka. Sedangkan kompas digital merupakan kompas yang telah menggunakan proses digitalisasi. Dengan kata lain, cara kerja kompas ini menggunakan komputerisasi.

2.5 Geotagging

Geotagging adalah proses penambahan identifikasi metadata geografis terhadap berbagai media contohnya, *geotagged photografi*, atau video, websites, pesan SMS, *QR code*, atau *RSS feed* dan merupakan bentuk dari metada *geospatial*. Data tersebut biasanya terdiri dari koordinat garis lintang (*latitude*) dan garis bujur (*longitude*), dan mungkin juga termasuk ketinggian (*altitude*), bantalan poros (*bearing*), jarak, data akurasi, dan nama tempat.

Geotagging dapat membantu pengguna untuk menemukan informasi posisi keberadaan spesifik dari lingkungan yang luas. Contohnya, seseorang dapat menemukan gambar terdekat dengan memberikan informasi koordinat *latitude* dan *longitude* ke dalam mesin pencari gambar yang sesuai. *Geotagging* dapat memberikan informasi layanan yang potensial digunakan untuk menemukan lokasi berdasarkan berita, website, dan sumber lainnya. *Geotagging* dapat memberitahukan kepada pengguna lokasi dari konten dari gambar atau media lainnya atau titik pandangan (*point of view*), dan sebaliknya pada beberapa platform media menunjukkan media yang relevan ke lokasi tertentu.

Terkait dengan istilah *geocoding* yang mengacu pada proses menentukan geografis *non* kordinat berdasarkan pengidentifikasi geografi, seperti alamat jalan, dan penemuan yang diasosiakan dengan koordinat geografi. banyak teknik yang dapat digunakan bersama dengan *geotagging* untuk menyediakan teknik pencarian alternatif.

2.6 Metode Pelacakan *Augmented Reality*

1. *Elektromagnetic tracking system*: mengukur medan magnet yang dihasilkan melalui arus listrik yang secara simultan melewati tiga kumparan kabel yang disusun secara tegak lurus satu dengan yang lain. Setiap kumparan kecil bersifat elektromagnet. Sensor sistem mengkalkulasikan bagaimana medan magnet terbentuk dan pengaruhnya terhadap kumpuran lainnya. Pengukuran tersebut menunjukkan posisi atau orientasi dan arah dari *emitter*. Reponsibilitas dari efisiensi sistem pelacakan elektromagnet sangat baik dan tingkat latensinya cukup rendah. Satu kekurangan dari sistem ini adalah apapun yang dapat menghasilkan medan magnet dapat mempengaruhi sinyal yang dikirim ke sensor.
2. *Acoustic tracking system*: sistem pelacakan ini menangkap dan menghasilkan gelombang suara *ultrasonic* untuk mengidentifikasi orientasi dan posisi dari target. Sistem ini mengkalkulasi waktu yang digunakan suara *ultrasonic* untuk mencapai sensor. Sensor biasanya selalu menjaga kestabilan dalam lingkungan dimana pengguna menempatkan *emitter*. Bagaimanapun, kalkulasi dari orientasi serta posisi target bergantung pada waktu yang digunakan oleh suara untuk mencapai sensor

adalah dilakukan oleh sistem. Terdapat banyak kekurangan pada sistem pelacakan *acoustic*. Suara yang lewat sangat lambat, sehingga tingkat update posisi target juga menjadi lambat. Efesiensi sistem dapat menjadi tidak efektif dengan kecepatan suara melewati udara karena sering kali berubah, bergantung pada kelembaban, temperatur atau tekanan barometer dalam lingkungan.

3. *Optical tracking system*: perangkat ini menggunakan cahaya untuk menghitung orientasi dan posisi target. Sinyal emitter dalam perangkat *optical* secara khusus terdiri atas sekumpulan LED inframerah. Sensor kamera dapat menangkap cahaya inframerah yang dipancarkan. LED menyala dalam *pulse* secara sekuensial. Kamera merekam sinyal *pulse* dan mengirim informasi kepada unit pemrosesan sistem. Unit tersebut kemudian dapat menghitung kemungkinan data untuk menentukan posisi dan orientasi target. Sistem optical mempunyai tingkat upload data yang cepat, sehingga latensi dapat diminimalisir. Kekurangan sistem ini adalah penglihatan antara kamera dan LED dapat menjadi gelap, bertentangan dengan proses pelacakan. Radiasi inframerah juga dapat membuat sistem kurang efektif.
4. *Mechanical tracking system*: sistem pelacakan ini bergantung pada physical link antara target dan referensi titik tetap. Salah satu contohnya adalah sistem pelacakan mekanikal dalam lingkungan virtual reality (VR), yaitu BOOM display. BOOM *display*, sebuah *head-mounted display* (HMD), dipasang di bagian belakang dengan yang terdiri atas 2 poin artikulasi. Deteksi orientasi dan posisi dari sistem dilakukan melalui

lengan. Tingkat update cukup tinggi dengan sistem pelacakan mekanikal, tetapi sistem ini memiki kekurangan yaitu membatasi pergerakan dari pengguna (*user*)

5. *Inertial navigation system*: navigasi bantuan yang menggunakan komputer, sensor gerak (*accelerometer*), sensor rotasi (*gyroscopes*) secara continue dikalkulasi melalui posisi dead reckoning (proses pengukuran posisi sekarang seseorang, dengan menggunakan posisi yang telah ditentukan sebelumnya, atau memperbaikinya, dan tingkatan posisi berdasarkan kecepatan rata-rata dari waktu-waktu), orientasi, dan kecepatan perpindahan objek tanpa membutuhkan referensi luar. Sistem ini digunakan dalam bidang transportasi seperti, kapal, pesawat, kapal selam, dan pesawat ruang angkasa.
6. GPS *Tracking*: teknologi AVL (*Automated Vehicle Locater*) yang memungkinkan pengguna untuk melacak suatu objek bergerak seperti kendaraan, armada ataupun mobil secara realtime. GPS *tracking* memanfaatkan kombinasi teknologi GSM dan GPS untuk menentukan koordinat sebuah objek, lalu menerjemahkan dalam bentuk peta digital.
7. *Hybrid Tracking*: sistem pelacakan yang merupakan gabungan dari dua atau lebih teknik pelacakan, *hybrid tracking* digunakan untuk menciptakan sistem pelacakan yang lebih baik. Teknik ini secara sinergis dapat meningkatkan kesegaran (*robustness*), kecepatan pelacakan (*tracking speed*) dan akurasi, dan mengurangi *jitter* dan *noice.* Hybrid tracking telah banyak digunakan dengan gabungan beberapa teknik pelacakan (misalnya,

GPS *electronic compass* dan sensor *inertial* dan sensor *optical* [Azuma et al., 1999]. *Magnetic* dan *vision-based* [Auer dan Pinz, 1999]).

2.7 Metode Penentuan Lokasi

Sejak tahun 1978, terdapat 59 satelit GPS yang berhasil di orbitkan mengelilingi bumi, walaupun pada tahun 2010, hanya 30 satelit yang dinyatakan masih beroperasi dengan normal. Kemampuan manusia untuk menentukan lokasi dengan tepat telah dikenal lama dengan menggunakan asap sebagai sinyal untuk menandakan suatu tempat atau lokasi.

GPS dirancang khusus untuk sistem navigasi di bumi. Jika digambarkan dalam peta dunia, maka posisi kita dapat ditandai dengan sebuah titik koordinat lokasi tempat dimana kita berada, seperti pada gambar II.13**,** titik yang ditandai terdiri atas dua komponen, yaitu *latitude* dan *longitude*, yang menginformasikan keberadaan kita di bumi melalui perangkat lunak GPS. Ketika titik koordinat lokasi diketahui, maka informasi ini dapat digunakan oleh program GPS untuk mendapatkan informasi lebih banyak untuk pengguna, seperti bisnis terdekat, dan informasi kemacetan lalu lintas. Ketika titik koordinat telah ditentukan, aplikasi akan menggunakan proses *geolocation* untuk mendapatkan informasi mengenai lokasi disekitar pengguna.

Tentunya penentuan lokasi tidak hanya berasal dari sistem GPS. Informasi yang digunakan dan bagaimana proses yang terjadi pada perangkat dan mem-*parsing* informasi lokasi adalah berdasarkan pada jenis perangkat yang digunakan. Terdapat beberapa metode untuk

menentukan lokasi pada perangkat komputasi dan tidak semua berdasarkan pada satelit GPS. Berikut beberapa cara penentuan lokasi :



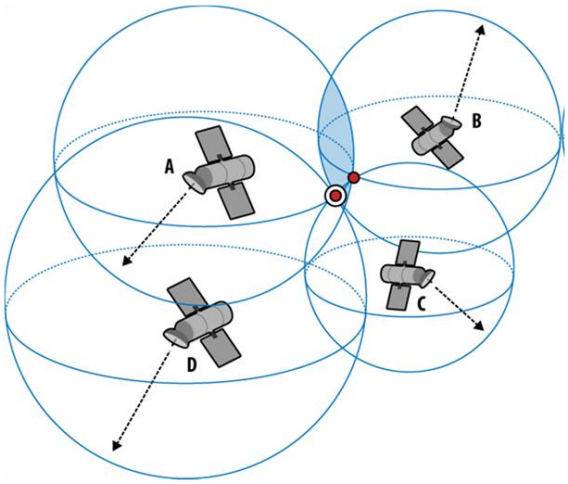
Gambar II.13 Pinpoint Lokasi keberadaan dimana pun di bumi

1. *Global Positioning Service* (GPS)

   Satelit GPS secara terus-menerus mengirim informasi sehingga perangkat GPS dapat mengurainya (*parse*), contohnya: susunan GPS yang masih berfungsi dengan baik, dimana semua satelit berada dalam orbit yang tepat atau jalur transmisi satelit, dan waktu transmisi. Perangkat *receiver* akan mengkalkulasikan posisinya dengan waktu pengiriman sinyal dengan beberapa satelit dalam *array* yang terlihat (tanpa hambatan ke angkasa).

   Alat penerima menentukan waktu yang dibutuhkan untuk menerima setiap pesan kemudian menghitung jarak dari setiap satelit berdasarkan informasi ini. Jarak pada setiap satelit dari alat penerima, orbit, dan perhitungan *trilateration* menginformasikan alat penerima mengenai posisi sekarang. Sementara tiga transmitter radio *triangulation* cukup untuk menentukan lokasi yang masuk akal (*reasonable location*), terdapat faktor waktu yang menjadi

perbandingan dengan satelit yang sedang orbit. Proses ini membutuhkan waktu, mungkin beberapa detik agar sinyal satelit menjangkau bumi – setiap kesalahan kecil jam pada satelit, dikalikan dengan waktu sekarang dapat menjadi kesalahan besar penentuan lokasi. Menggunakan keempat satelit mengurangi kesalahan dalam persamaan (gambar II.14). Dalam kasus yang paling sering terjadi, sebuah alat *receiver* akan menggunakan empat satelit atau lebih untuk menghitung posisinya.



Gambar II.14 Sistem GPS terhubung ke satelit

2. *IP Address*

IP (*Internet Protocol*) *address* merupakan angka unik yang diberikan pada setiap perangkat yang terhubung dengan internet sehingga dapat berkomunikasi dengan perangkat lainnya. Ketika perangkat terhubung dengan internet, perangkat dapat mengirim data kepada perangkat lain dan kemudian menerima respon kembali. Meskipun sebuah perangkat menerima alamat yang lebih atau selalu berubah-ubah, *IP address* dapat berubah statis

(permanen) atau dinamis (temporar). Tanpa memperhatikan jenis alamat yang dimiliki oleh sebuah perangkat, alamat tersebut selalu akan terdiri dari empat kumpulan angka yang dipisahkan oleh titik, misalnya : 123.123.123.123.

Umumnya, sebuah *IP Address* ditentukan oleh ISP (*Internet Service Provider*) dalam bagian-bagian berdasarkan wilayah oleh lembaga *registry* lokal. Oleh karena itu, negara, wilayah, dan bahkan sebuah kota dapat dengan mudah di identifikasi dengan sebuah *IP Address*. Selanjutnya, kemajuan saat ini dalam pengumpulan data dan pengelolaannya oleh ISP, perangkat lokasi geografis mampu menentukan sebuah lokasi dengan tingkat akurasi tinggi. Yang menjadi tantangan terbesar dengan penggunaan IP Address untuk menentukan sebuah lokasi adalah bahwa terdapat ratusan institusi regional yang akan diproses untuk mendapatkan data.

3. GSM/CDMA Cell ID

Cell ID merupakan angka unik yang digunakan untuk mengidentifikasi setiap perangkat *mobile* pada jaringan seluler, umumnya seperti *IP Address* mengidentifikasikan sebuah perangkat dalam jaringan. Ada dua tipe jaringan yang paling popular digunakan yaitu *Global System for Mobile Communication* (GSM) dan *Code Division Multiple Access* (CDMA). Jenis layanan seluler yang dibutuhkan didasarkan pada daerah cakupan sinyal.

GSM merupakan teknologi jaringan *mobile* yang paling tertua dan oleh karena itu peningkatan pelayanan dari teknologi GSM tersedia luas dibanding teknologi lainnya. GSM adalah teknologi 2G yang tersedia lebih dari 200 negara, dan jumlah *transfer* data pada jaringan ini lebih dari 75% dari seluruh dunia. Kemampuan jaringan GSM untuk dilakukan migrasi ke layanan jaringan 3G dan 4G (*Evolved High Speed Packet Access* (HSPA+), *Long Term Evolution* (LTE), dan *Service Architecture Evolution* (SAE) juga sangat cepat, sehingga dapat melayani hampir empat juta pelanggan. Meskipun teknologi GSM tidak dapat menangani jumlah data yang sama seperti teknologi lainnya, jaringan GSM memiliki tingkat kualitas jaringan yang tinggi karena kemampuannya untuk menempatkan repeater didalam dan diluar bangunan bahkan jumlah penggunaan repeater yang semakin dikurangi. Hal ini menjadikan jaringan GSM sebagai operator dengan layanan jaringan yang baik dengan tingkat gangguan jaringan yang kecil.

CDMA merupakan teknologi terbaru setelah GSM, baik CDMA dan GSM sama-sama merupakan layanan jaringan 2G (cdmaOne(TM)), dan 3G (CDMA2000(R) dan WCDMA). Keuntungan dari teknologi CDMA adalah dapat mengijinkan banyak pengguna dalam waktu yang bersamaan dengan porsi frekuensi yang diberikan oleh *band*, dimana tidak dapat dilakukan pada jaringan GSM. Jaringan CDMA, seperti GSM juga dapat

dilakukan migrasi ke teknologi LTE untuk mendukung jaringan 4G.

Tanpa memperhatikan jenis teknologi dari perangkat *mobile* yang digunakan, prinsip dasarnya adalah mempunyai *identifier unique* dalam jaringan. Menggunakan *triangulation*, *latitude* dan *longitude* Cell ID dapat diidentifikasi, dengan dilakukan *geolocated*. Penggunaan beberapa tower untuk melakukan triangulation posisi dari perangkat mobile, mempunyai tingkat akurasi tinggi. Hal ini mengapa *geolocation* yang bergantung pada jenis teknologi, bekerja lebih baik dalam lingkungan urban (perkotaan) – dimana keberadaan tower yang satu dengan lainnya berdekatan dibanding pada area pedesaan.

4. *Wifi* dan *Bluetooth MAC Address*

Wifi dan *Bluetooth MAC address* mempunyai kinerja seperti *IP Address. Media Access Control (MAC) address* merupakan pemberian angka unik pada suatu perangkat, biasanya dilakukan oleh pabrik pembuat perangkat tersebut. Angka ini sifatnya permanen dan sebagai penanda global, namun pada perangkat modern, angka ini dapat diganti secara manual; dikenal dengan *MAC spoofing* (teknik untuk mengganti default *MAC address* dari antarmuka jaringan pada perangkat jaringan). Contoh sebuah MAC Address : 12-34-56-78-9A-BC. *MAC address* untuk router wifi adalah alamat yang mudah untuk ditemukan atau ditandai pada perangkat nirkabel (*wireless*), begitupun dengan

MAC *address* untuk perangkat Bluetooth. Menggunakan alamat ini dengan cara yang sama sebagai alamat IP, *latitude* (garis lintang) dan *longitude* (garis bujur), dan lokasi fisik, dapat ditemukan.

2.8 Layanan Berbasis Lokasi

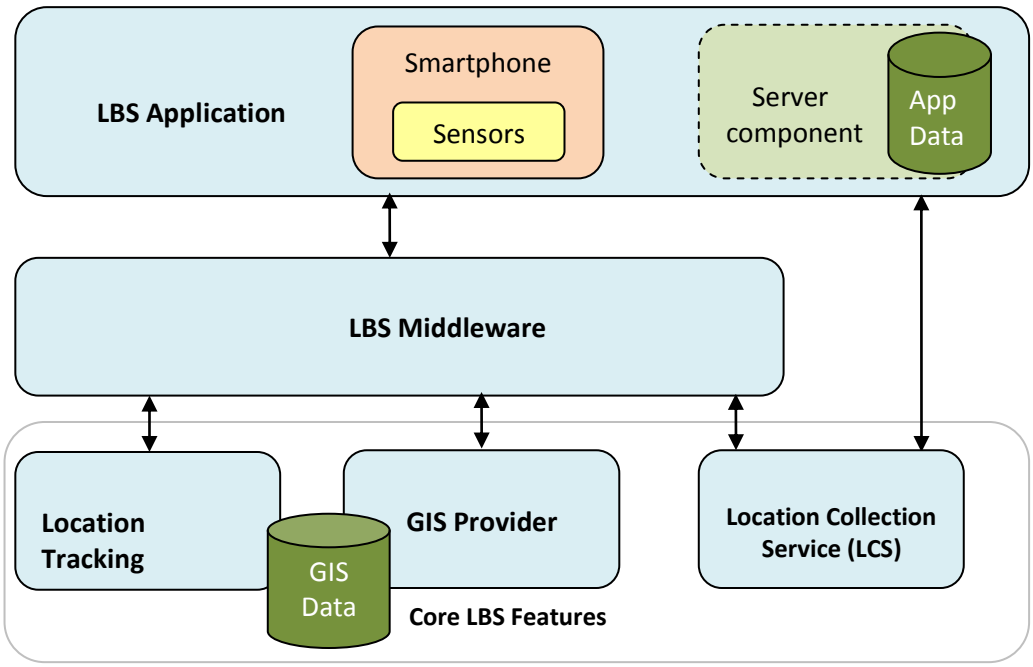
Layanan berbasis lokasi atau yang dikenal LBS (*location-based service*) adalah aplikasi komputasi *mobile* yang memberikan layanan kepada pengguna berdasarkan lokasi geografis. Contohnya, "temukan stasiun terdekat", sebuah aplikasi yang memberikan informasi stasiun kereta terdekat.

Generasi pertama LBS, reaktif dan fokus pada *client-server*; pengguna akan bertanya pada aplikasi atau sistem untuk sebuah informasi dan kemudian mendapatkan respon. Dengan kemampuan mekanisme *push notification*, peningkatan akses internet mobile dan dikolaborasikan dengan kemampuan Web 2.0, kemampuan LBS akan lebih proaktif dan interaktif. Contohnya :

1. Informasi yang dikirim kepada pengguna berdasarkan lokasi keberadaan sehingga pengguna dapat mencari informasi sesuai kebutuhannya.
2. Orang-orang dapat berbagi (*sharing*) foto dan ulusan informasi dengan model *peer-to-peer*. Sumber data lebih bervariasi, dengan layanan provider yang banyak dibandingkan jika hanya satu sumber.

2.8.1 Teknologi dan Komponen-komponen LBS

Layanan Berbasis lokasi adalah layanan informasi yang dapat diakses melalui *mobile device* dengan mengunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut. LBS terdiri dari beberapa komponen seperti peta (*maps*) dan sistem informasi geografis (GIS), layanan kolektor lokasi (*Location Collective Service*), dan subkomponen spesifik dari aplikasi LBS. Arsitektur dari sebuah LBS dapat digeneralisasi pada gambar 2.15. Deskripsi dari setiap komponen juga ditabulasikan pada tabel



Gambar II.15 Diagram komponen LBS

Tabel II.1 Komponen – komponen LBS

Komponen	Deskripsi
LBS Application	Mendeskripsikan sebuah aplikasi yang spesifik seperti aplikasi pencarian teman ("*find my friend*"). Aplikasi ini terdiri dari komponen

	smartphone, mempunyai beberapa sensor, dan komponen *server* yang potensial termasuk data spesifik aplikasi (seperti informasi lokasi keberadaan).
LBS middleware	Bagian inti dari fitur-fitur LBS (pelacakan lokasi, layanan GIS, dan layanan pengumpulan lokasi (*location collective services*) untuk memberikan asistensi antarmuka pada aplikasi LBS
Location Tracking	Komponen ini menyimpan lokasi pelacakan dari pengguna. Merepresentasikan komponen dasar dalam next-generasi LBS seperti data yang memungkinkan rute pengguna ditentukan dan diprediksi. Secara khusus, komponen ini mendukung fungsionalitas sebagai berikut : • Tetap merekam posisi pengguna baik lokasi sekarang maupun lokasi sebelumnya. • Memberikan notifikasi pada komponen lainnya ketika pengguna berpindah dari suatu tempat ke tempat yang lain. Mendukung lokasi berdasarkan notifikasi yang dikirim ke pengguna. • Menentukan lokasi ketika pengguna berada dalam lokasi yang telah ditentukan. Mendukung fitur *geocasting*. • Queri dari jejak lokasi untuk men-generasi model perpindahan pengguna.
GIS Provider	Komponen ini menyediakan fungsionalitas *geospatial* untuk beberapa LBS termasuk peta informasi, peta visualisasi, dan layanan direktori. Google maps dengan API nya dapat dianggap sebagai penyedia GIS. Contohnya,

	deCarta dan OpenRouteService.org
Location Collection Service (LCS)	Komponen ini mengumpulkan lokasi untuk mendapatkan garis lintang (*latitude*) dan garis bujur (*longitude*) untuk lokasi spesifik pengguna. Berdasarkan pada teknologi, komponen ini mungkin diakses melalui *middleware* LBS (misalnya, triangulation jaringan *mobile* melalui penyedia layanan) atau secara langsung (contohnya, via GPS *receiver* dalam *smartphone*).

2.9 Geolocation

Geolocation adalah identifikasi lokasi geografis suatu objek pada dunia nyata. Contohnya, radar, telepon genggam, atau terminal koneksi internet komputer. *Geolocation* mempunyai kaitan erat dengan *positioning*, perbedaannya adalah *geolocation* lebih spesifik dalam menentukan sebuah lokasi (misalnya alamat jalan) dibandingkan dengan *positioning* yang hanya mencakup sekumpulan koordinat geografis. Secara spesifik melibatkan penggunaan sistem lokasi frekuensi radio (RF), misalnya, perbedaan waktu kedatangan (TDOA = *Time Different of Arrival*) dimana memungkinkan spesifikasi lokasi yang baik. Sistem TDOA memanfaatkan pemetaan (*mapping*) atau sistem informasi geografis lainnya. Hal ini berbeda dengan teknologi *radiolocation* (pencarian lokasi dengan menggunakan gelombang radio) konvensional, misalnya pencarian langsung (*direct finding*) dimana sebuah garis perpindahan (*line of bearing)* ke pemancar dicapai dan bukan spesifik lokasi.

Internet dan komputer memungkinkan *geolocation* dapat dilakukan dengan menggabungkan lokasi geografis dengan *internet protocol* (IP) *address*, MAC *Address*, RFID (*Radio-frequency identification*), perangkat keras dengan nomor produksinya, angka seri perangkat lunak (contohnya, UUID, Exif/IPTC/XMP atau modern Steganography), *invoice*, *wifi*, GPS, dan masih banyak lagi. *Geolocation* umumnya bekerja secara otomatis mencari alamat IP pada layanan WHOIS (query atau respon protokol yang sangat luas digunakan dalam query database untuk menyimpan pengguna terdaftar atau bertindak sebagai sumber data internet, seperti nama domain, IP address, atau sistem *autonomous*) dan mengambil *physical address* terdaftar.

Data geolocation alamat IP mencakup informasi seperti negara, wilayah, kota, kode pos, *latitude*, *longitude*, dan zona waktu. Sekumpulun data yang lebih khusus dapat menentukan parameter lainya seperti nama domain, kecepatan koneksi, ISP, bahasa, *proxy*, nama perusahaan, US DMA/MSA, NAICS codes, dan rumah ataupun bisnis.

2.9.1 *Latitude* (Garis Lintang)

Dalam geografi, garis lintang adalah garis khayal yang digunakan untuk menentukan lokasi di bumi terhadap garis khatulistiwa (utara atau selatan). Posisi lintang biasanya dinotasikan dengan simbol huruf Yunani (φ). Posisi lintang merupakan perhitungan sudut dari 0^0 di khatulistiwa sampai ke $+90^0$ di kutub utara dan -90^0 di kutub selatan.

2.9.2 Longitude (Garis Bujur)

Bujur kadangkala dinotasikan oleh huruf abjad Yunani (λ), menggambarkan lokasi sebuah tempat di timur atau barat daya Bumi dari sebuah garis utara-selatan yang disebut Meridian Utama. Longitude diberikan berdasarkan pengukuran sudut yang berkisar dari 0^0 di Meridian Utama ke $+180^0$ arah timur dan -180^0 arah barat. Tidak seperti lintang yang memiliki equator sebagai posisi awal alami, tidak ada posisi awal alami, tidak ada posisi awal untuk bujur. Oleh karena itu, sebuah dasar meridian harus dipilih. Meskipun kartografer Britania Raya telah lama menggunakan meridian Observatorium Greenwich di London, referensi lainnya digunakan di tempat yang berbeda, termasuk Ferro, Roma, Kopenhagen, Yerusalem, Saint Petersburg, Pisa, Paris, Philadelphia, dan Washington, D.C.. Pada tahun 1884, Konferensi Meridian Internasional mengadopsi meridian Greenwich sebagai Meredian utama universal atau titik nol bujur.

2.10 JSON (*Javascript Object Notation*)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl,

Phyton, dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data.

Berikut contoh format penulisan JSON :

```
[{
"menu" : "File",
"commands" : [
{
    "value" : "New",
    "action" : "CreateDoc"
},
    "value" : "Open",
    "action" : "OpenDoc"
},
    "value" : "Close",
    "action" : "CloseDoc"
}]
}]
```

Perbandingan XML dan JSON

1. *Simplicity*

   XML lebih sederhana dari SGML (*Standard Generalized Markup Language*), tetapi JSON lebih sederhana daripada XML. JSON mempunyai banyak grammer singkat dan dipetakan langsung kedalam struktur data yang digunakan dalam bahasa pemrograman terkini (*modern*).

2. *Extensibility*

   JSON tidak dapat diperluas (*is not extensible*) karena tidak dibutuhkan dalam JSON. JSON bukanlah dokumen bahasa *markup*, sehingga tidak diperlukan mendefinisikan *tags* atau atribut untuk representasi data didalamnya.

3. *Interoperability*

   JSON dan XML, mempunyai potensial *interoperability* yang sama.

4. *XML is human readable*

   JSON lebih mudah dibaca dan ditulis oleh manusia dibandingkan dengan XML. Serta mudah diterjemahkan dan dihasilkan (*generate*) oleh komputer.

5. XML dan JSON, sama-sama mendukung pertukaran data sehingga me-mungkinkan pengguna memindahkan data diantara aplikasi.
6. XML dan JSON, sama-sama menyediakan data terstruktur untuk meningkatkan atau memperkaya informasi.
7. XML mudah untuk diproses karena struktur datanya sederhana dan standard. JSON, proses ini lebih mudah karena struktur datanya lebih sederhana.

2.11 Google Maps

Google maps adalah teknologi dan aplikasi layanan web pemetaan yang disediakan oleh Google, sifat bebas (tidak untuk penggunaan komersial), yang banyak digunakan layanan berbasis peta, mencakup website Google Maps, Google Ride Finder, Google Transit, dan peta yang digunakan oleh penyedia layanan pihak ketiga melalui Google Maps API. Google maps menawarkan peta jalan (*street maps*), sebuah perencana rute (*route planner*) untuk perjalanan dengan menggunakan mobil, pejalan kaki, sepeda atau angkutan umum dan pencari bisnis perkotaan untuk beberapa negara di seluruh dunia. Satelit Google Maps, gambar tidak diperbaharui dalam waktu nyata (realtime); mungkin beberapa minggu atau tahun lamanya.

Google Maps menggunakan varian yang mirip dengan *Mercator projection* (proyeksi peta silender diperkenalkan oleh *geographer* Belgian dan *cartographer* Gerardus Mercator, pada tahun 1569) sehingga tidak dapat menampilkan wilayah kutub. Produk yang berelasi dengan Google Maps adalah Google Earth, program yang berdiri sendiri (*stand-alone*) yang menawarkan fitur pandangan globe, termasuk area kutub.

2.10.1 Google Maps API

Diperkenalkan oleh Google pada Juni 2005, Google Maps API adalah fitur yang memungkinkan para pengembang meng-integrasikan Google Maps ke dalam website mereka. Google Maps API adalah layanan gratis (free). Penggunaan Google Maps API, dimungkinkan untuk menanamkan Google Maps kedalam website luar, untuk itu data spesifik dapat ditambahkan. Walaupun pada awalnya hanya JavaScript API, Maps API telah dikembangkan termasuk API untuk aplikasi *adobe flash*, layanan untuk mengumpulkan gambar peta static, dan layanan web *(web service*) untuk melakukan *geocoding*, menghasilkan arah kemudi (*driving directions*), dan informasi elevasi.

2.10.2 *Google Maps for Mobile*

Pada tahun 2006, Google memperkenalkan aplikasi Java yang diberi nama *Google Maps for mobile*, ditujukan untuk telepon genggam berbasis Java atau perangkat mobile lainnya. Pada tanggal, 28 november 2007, *Google Maps for Mobile* 2.0 dirilis. Fitur baru dalam versi ini adalah layanan lokasi *non*-GPS, layanan yang tidak membutuhkan perangkat penerima GPS. Fitur "*my location*" bekerja dengan memanfaatkan lokasi

GPS dari perangkat *mobile,* jika tersedia. Informasi ini dilengkapi dengan perangkat lunak untuk menentukan jaringan nirkabel (*wireless*) dan *cell site* (tower BTS) terdekat. Perangkat lunak akan mencari lokasi dari tower BTS menggunakan database pengenal jaringan *wireless* dan *cell site*. Metode *cell side* menggunakan triangulasi perbedaan kekuatan sinyal dari *cell transmitter* yang berbeda dan selanjutnya menggunakan properti lokasi (diperoleh dari database cell site online) untuk membantu "My Location" menentukan lokasi pengguna. Metode jaringan *wireless* (jaringan nirkabel) mengkalkulasikan dengan menjangkau hotspot wifi terdekat, dan menggunakan property lokasi (diperoleh dari database wifi online, cara yang sama dengan database cell site) untuk kemudian menemukan lokasi pengguna. Berikut proses urutannya :

1. *GPS-based Services*
2. *WLAN, Wifi-based Services*
3. *Cell Transmitter-based Services*

2.10.3 Google Maps Android 2.0

Saat ini, penggunaan telepon genggam sangat meningkat sebagai alat bantu navigasi. Google *Navigation for* Android 2.0 adalah layanan gratis.

Fitur-fitur yang terdapat dalam aplikasi adalah sebagai berikut :

1. Pencarian dalam bahasa Inggris
2. Pencarian dengan suara
3. Pandangan lalu lintas
4. Pencarian sepanjang rute

5. Pandangan satelit
6. Pandangan jalan
7. *Car dock mode*

2.12 *Relational Unified Process* (RUP)

Rational Unified Process merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practise* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan *iterative* untuk siklus pengembangan perangkat lunak.

2.13 Konsep Perancangan Berorientasi Objek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang di dominasi oleh objek. Di dalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah mempermudah *programmer* dalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata. Suatu perusahaan *software* yaitu *Rational Software*, telah membentuk konsorsium dengan berbagai organisasi untuk meresmikan pemakaian *Unifed Modelling Language* (UML*)* sebagai bahasa standar dalam *Object Oriented Analysist Design* (OOAD*).*

2.12.1 Unified Modeling Language (UML)

UML dalam sebuah bahasa untuk menentukan visualisasi, konstruksi, dan mendokumentasikan *artifact* dari sistem *software*, untuk memodelkan bisnis, dan sistem *non-software* lainnya. UML merupakan sistem arsitektur yang bekerja dalam OOAD (*Object Oriented Analysis and Design*) dengan satu bahasa yang konsisten untuk menentukan, visualisasi, konstruksi dan mendokumentasikan *artifact* yang terdapat dalam sistem. *Artifact* adalah potongan informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*. *Artifact* dapat berupa model, deskripsi atau *software*.

2.12.2 Use Case Diagram

Use Case Diagram menjelakan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (aktor). Diagram ini menunjukan fungsionalitas suatu sistem yang berinteraksi dengan dunia luar. *Use Case Diagram* dapat digunakan selama proses analisis untuk menangkap *requirement* sistem dan untuk memahami bagaimana sistem bekerja.

2.12.3 Class Diagram

Class Diagram menjelaskan dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class Diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain dari suatu sistem.

Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan prilaku sistem. Selama tahap desain, *class diagram* berperan dalam menagkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

2.12.4 *Behaviour Diagram*

Behaviour diagram dapat dikelompokkan menjadi tiga diagram, yaitu :

1. *Statechart Diagram*

   Statechart Diagram berfungsi untuk memodelkan perilaku dinamis satu kelas satu objek

2. *Activity Diagram*

   Activity diagram memodelkan alur kerja (*workflow*) sebuah bisnis dan urutan aktifitas dalam suatu proses.

3. *Interaction Diagram*

   Interaction diagram dibagi menjadi dua model diagram, yaitu :

   1) *Sequence Diagram*, menjelaskan interaksi objek yanag disusun dalam suatu urutan waktu. Diagram ini secara khsus bersosialisasi dengan *use case*. *Sequence diagram*, memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu dalam *use case*.
   2) *Collaboration Diagram*, melihat pada interaksi dan hubungan terstruktur antar objek. Tipe diagram ini menekankan pada hubungan (*relationship*) antar objek, sedangkan *sequence* diagram menekankan pada urutan kejadian. Dalam

collaboration diagram terdapat beberapa objek, *link*, dan *message*.

2.12.5 *Implementation Diagram*

Implementation diagram dibagi menjadi dua diagram, yaitu :

1. Component diagram, menggambarkan alokasi semua kelas dan objek kedalam komponen-komponen dalam desain fisik sistem *software*. Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software*, seperti *source code*, *binary code*, dan komponen tereksekusi (*execute components*).
2. Deployment diagram, memperlihatkan pemetaan *software* kepada *hardware*. Dimana akan berjalan (di server/multitier, *standalone*, atau lainnya), dan menggambarkan model koneksi dan kemampuan jaringan dan hal lainnya yang bersifat fisik.

2.14 Android

2.14.1Sejarah Android

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. Hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh

kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. versi android terbaru yaitu versi 3.0. Android juga sudah bergabung dengan beberapa smart *Mobile* seperti Nokia, Sony Ericsson, dan lainnya. Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android ARM Holdings*, *Atheros Communications*, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*mobile*) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

Berikut adalah berbagai versi android yang telah diluncurkan atau telah dirilis :

1. *Android versi 1.1*

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

2. *Android versi 1.5 (Cupcake)*

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

3. *Android versi 1.6 (Donut)*

   Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai *indicator* dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus pada kamera, *camcorder* dan galeri yang dintegrasikan pada CDMA / EVDO, 802.1x, VPN,

Gestures, dan *Text-to-speech engine*. Kemampuan dial kontak teknologi *text to change speech* (tidak tersedia pada semua ponsel).

4. *Android versi 2.0/2.1 (Eclair)*

   Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan *Google Maps* 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan Bluetooth 2.1. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi m*obile* terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

   Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset* Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah *Shazam*, *Backgrounds*, dan *WeatherBug*. Sistem operasi Android dalam situs internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

5. *Android versi 2.2 (Froyo : Frozen Yoghurt)*

   Pada 20 Mei 2010, Android versi 2.2 (*Froyo*) diluncurkan. Perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada browser, pemasangan aplikasi dalam SD *Card*, kemampuan *WiFi Hotspot portabel*, dan kemampuan *auto update* dalam aplikasi Android Market.

6. *Android versi 2.3 (Gingerbread)*

   Pada 6 Desember 2010, Android versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang di dapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru *(reverb, equalization, headphone virtualization, dan bass boost)*, dukungan kemampuan *Near Field Communication (NFC)*, dan dukungan jumlah kamera yang lebih dari satu.

7. *Android versi 3.0/3.1 (Honeycomb)*

   Android *Honeycomb* dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk *tablet*. Honeycomb juga mendukung *multiprocessor* dan juga akselerasi

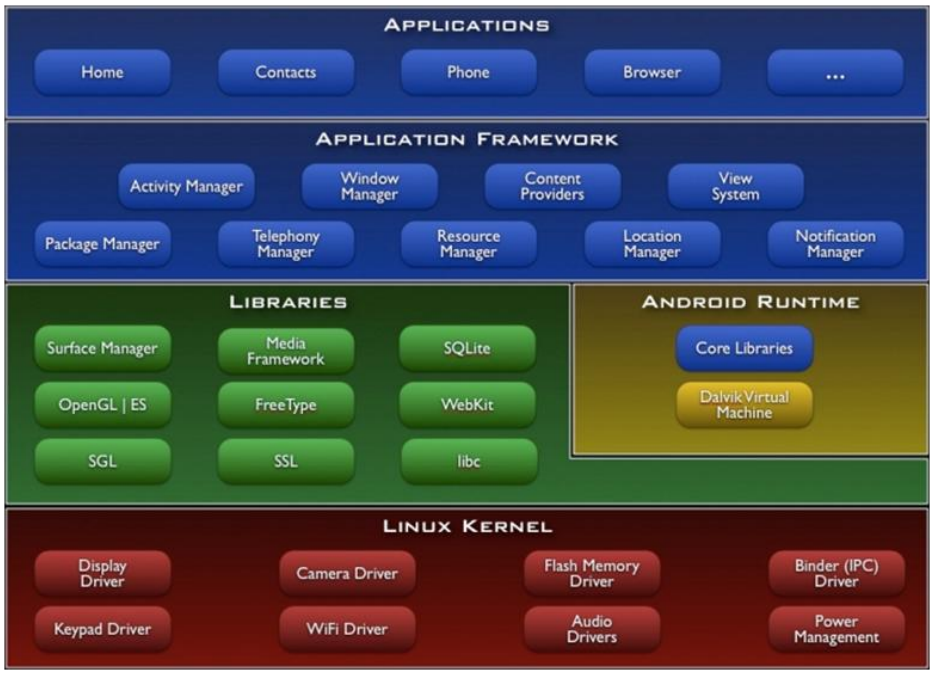
perangkat keras (*hardware*) untuk grafis. Tablet pertama yang dibuat dengan menjalankan *Honeycomb* adalah Motorola Xoom.

8. *Android versi 4.0 (Ice Cream Sandwich)*

   Ice Cream Sandwich telah diperkenalkan dalam acara Google I/O pada bulan Mei 2011, dan resmi dirilis pada tanggal, 19 Oktober 2011. Dengan fitur *Honeycomb* yang telah disempurnakan seperti pengunci dengan pengenal muka (*facial recognition unlock*), control dan pengawasan penggunaan data jaringan (network data usage monitoring), penggabungan kontak dengan jaringan sosial (*social networking contacts*), peningkatan kualitas fotografi, pencarian email offline (*offline email searching*), dan berbagi informasi menggunakan NFC (*Near Field Communication*).

2.14.2 Anatomi Aplikasi Android

Dalam paket sistem operasi android tediri dari beberapa unsur seperti tampak pada gambar 2.8. Secara sederhana arsitektur android merupakan sebuah kernel Linux dan sekumpulan pustaka C/C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi.



Gambar II.16 Arsitektur Linux kernel

2.14.2.1 Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa *service* yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan *driver*. Kernel linux menyediakan driver layar, kamera, *keypad*, *WiFi*, *Flash Memory*, *audio*, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

2.14.2.2 *Library*

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari yang

aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

1. *Media Library* untuk memutar dan merekam berbagai macam format audio

   dan video.
2. *Surface Manager* untuk mengatur hak akses layer dari berbagai aplikasi.
3. *Graphic Library* termasuk didalamnya *SGL* dan *OpenGL,* untuk tampilan 2D dan 3D.
4. *SQLite* untuk mengatur relasi database yang digunakan pada aplikasi.
5. *SSI* dan Webkit untuk browser dan keamanan internet.

Pustaka-pustaka tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan *Native Development Toolkit* (NDK).

2.14.2.3 Android *Runtime*

Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. *Android Runtime* merupakan mesin virtual yang membuat aplikasi

android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam Android Runtime terdapat 2 bagian utama, diantaranya:

1. Pustaka Inti, android dikembangkan melalui bahasa pemrograman Java, tapi *Android Runtime* bukanlah mesin *virtual* Java. Pustaka inti android menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus android.
2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin *virtual* yang mengeksekusi file dalam format *Dalvik Executable* (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

2.14.2.4 *Aplication Framework*

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan *User Interface* dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut:

1. *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan ”*Backstack“* untuk navigasi penggunaan.
2. *Content Providers*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.
3. *Resource Manager*, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan *file layout*.
4. *Location Manager,* berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.
5. *Notification Manager,* mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada *status bar.*

2.14.2.5 *Application Layer*

Puncak dari diagram arsitektur android adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam *Android runtime* dengan menggunakan kelas dan *service* yang tersedia pada *framework* aplikasi.

Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada android semua aplikasi, baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (*Application Programming Interface*) yang sama.

2.14.3 Komponen Aplikasi Android

Fitur penting android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur *scroller* dan aplikasi lain telah mengembangkan fitur *scroller* yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup menggunakan *scroller* yang telah ada.

Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi ”*main()“* pada aplikasi lain. Sebaliknya, aplikasi memiliki komponen penting yang memungkinkan sistem untuk memanggil dan menjalankan ketika dibutuhkan.

2.14.3.1 *Activity*

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena a*ctivity* menyajikan tampilan visual program yang sedang

digunakan oleh pengguna. Setiap *activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari *Views* dan respon terhadap *Event*. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi dijalankan.

Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack.* Keadaan suatu *activity* ditentukan oleh posisinya dalam tumpukan *acitivity*, LIFO (*Last In First Out*) dari semua aplikasi yang sedang berjalan. Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *back*, atau menutup *activity* yang sedang digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory manager* android menggunakan tumpukkan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut.

1. ***Active,*** setiap *activity* yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukkan dari pengguna. Android akan berusaha untuk membuat *activity* aplikasi ini untuk untuk tetap hidup dengan segala cara, bahkan akan menghentikan *activity* yang berada dibawah tumpukkannya jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.

2. ***Paused,*** dalam beberapa kasus *activity* akan terlihat tapi tidak terfokus pada kondisi inilah disebut *paused.* Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat *active* namun tidak dapat menerima masukkan dari pengguna. Dalam kasus ekstrim, android akan menghentikan *activity* dalam keadaan *paused* ini, untuk menunjang sumber daya bagi *activity* yang sedang aktif.
3. ***Stopped,*** ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped. Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh karenanya ketika suatu *activity* dalam kondisi *stopped* maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika *activity* telah keluar atau ditutup, maka dia akan menjadi *inactive.*
4. ***Inactive,*** kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. *Inactive activity* telah ditiadakan dari tumpukan *activity* sehingga perlu *restart* ulang agar dapat tampil dan digunakan kembali.

Kondisi transisi ini sepenuhnya ditangani oleh manajer memori android. Android akan memulai menutup aplikasi yang mengandung

activity inactive, kemudian *stopped activity*, dan dalam kasus luar biasa *paused activity* juga akan di tutup.

2.14.3.2 *Services*

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.

2.14.3.3 *Intents*

Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman web, dan lain sebagainya. *Intents* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi suatu hal, misal pesan masuk. *Intents* dapat eksplisit atau implisit, contohnya jika suatu aplikasi ingin menampilkan *URL*, sistem akan menentukan komponen apa yang dibutuhkan oleh *Intents* tersebut.

2.14.3.4 *Broadcast Receiver*

Broadcast Receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar *broadcast* berasal dari sistem misalnya, baterai sudah hampir habis, informasi zona waktu telah berubah, atau

pengguna telah merubah bahasa *default* pada perangkat. Sama halnya dengan *service, Broadcast Receivers* tidak menampilkan antarmuka pengguna. Namun, *Broadcast Receivers* dapat menggunakan *notification manager* untuk memberitahukan sesuatu kepada pengguna.

2.14.3.5 *Content Providers*

Content Providers digunakan untuk mengelola dan berbagi database. Data dapat disimpan dalam file sistem, dalam database *SQLite*, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *Content Provider* memungkinkan antar aplikasi untuk saling berbagi data. Komponen ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain, sehingga mudah dalam penerapannya.

2.14.4.5 Tipe Aplikasi Android

Terdapat tiga kategori aplikasi pada android :

1. *Foreground Activity*

   Aplikasi yang hanya dapat dijalankan jika tampil pada layar dan tetap efektif walaupun tidak terlihat. Aplikasi dengan tipe ini pasti mempertimbangkan siklus hidup *activity*, sehingga perpindahan antar *activity* dapat berlangsung dengan lancar.

2. *Background Service*

   Aplikasi yang memiliki interaksi terbatas dengan user, selain dari pengaturan konfigurasi, semua dari prosesnya tidak tidak tampak pada layar. Contohnya aplikasi penyaringan panggilan atau sms *auto respon*.

3. *Intermittent Activity*

   Aplikasi yang masih membutuhkan beberapa masukkan dari pengguna, namun sebagian sangat efektif jika dijalankan di *background* dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu. Contohnya pemutar musik.

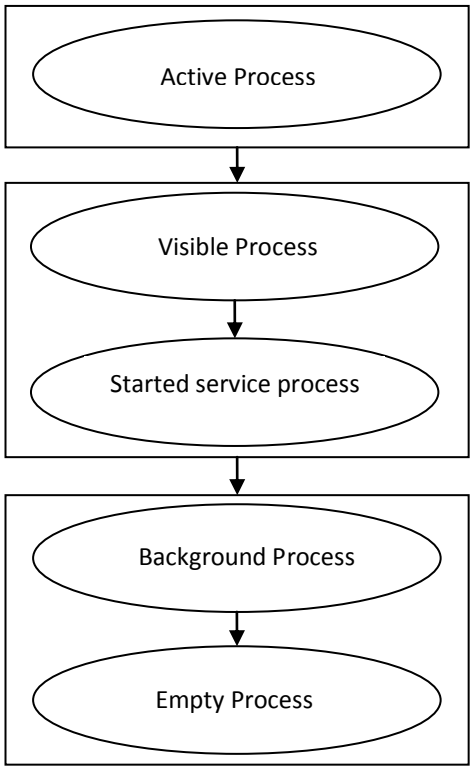
Untuk aplikasi yang kompleks akan sulit untuk menentukan kategori aplikasi tersebut apalagi memiliki ciri-ciri dari semua kategori. Oleh karena itu, perlu pertimbangan bagaimana aplikasi tersebut digunakan dan menentukan kategori aplikasi yang sesuai.

2.14.5 Android *Life Cycle*

Life cycle aplikasi android dikelola oleh sistem, berdasarkan kebutuhan pengguna, sumberdaya yang tersedia, dan sebagainya. Misalnya Pengguna ingin menjalankan browser web, pada akhirnya sistem yang akan menentukan menjalankan aplikasi. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika pengguna ketika itu sedang menjalankan sebuah *Activity*, maka sistem akan memberikan perioritas utama untuk aplikasi yang tersebut. Sebaliknya, jika suatu *Activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *Activity* yang prioritas rendah akan ditutup.

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin virtual pengolah sendiri, dengan ini melindungi penggunaan memori pada aplikasi. Selain itu juga android

dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karenanya android sangat sensitive dengan siklus hidup aplikasi dan komponen-komponennya. Perlu adanya penanganan terhadap setiap kondisi agar aplikasi menjadi stabil. Gambar II.17 menunjukkan prioritas dari aplikasi.



Gambar II.17 Prioritas aplikasi berdasarkan *activity*

2.14.6 Kelebihan Android

Sudah banyak *platform* untuk perangkat selular saat ini, termasuk di dalamnya Symbian, iOS, Windows *Mobile*, BlackBerry, Java *Mobile* Edition, Linux *Mobile* (LiM), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada

telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut :

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar royaliti. Sementara pengembang *software* menyukai karena android dapat digunakan diperangkat manapun dan tanpa terikat oleh *vendor* manapun.
2. Arsitektur komponen dasar android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu khawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.

5. Dukungan grafis dan suara terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun *game* yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin *virtual* Dalvik, sehingga kode program *portabel* antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard,* layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.15 MySQL

MySQL adalah implementasi dari sistem menejemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, dan pemakaiannya tidak untuk kepentingan komersial. MySQL merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (*Structured Query Language*), adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

2.15.1 Sejarah MySQL

MySQL diciptakan oleh MySQL AB, sebuah perusahaan yang didirikan oleh Michael Widenius (Monty), David Axmark, dan Allan Larsson, di Swedia pada tahun 1995. Kemudian pada tahun 2008, Sun Microsystem,Inc. mengakusisi perusahaan tersebut dengan perkiraan harga $1 billion US dollar. Pada tanggal, 27 januari 2010, Sun Microsystem, Inc diakusisi oleh Oracle Corporation, setelah kedua perusahaan tersebut menandatangi persetujuan pada tanggal 20 april 2009, Sun Microsystem, Inc kemudian bergabung (*merged*) dengan Oracle USA, Inc. kemudian perusahaan tersebut berubah nama menjadi Oracle America,Inc.

2.15.2 Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Microsoft Windows, Linux, FreeBSD, Mac OS X server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-*user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. Peningkatan performa (*performance tuning*). MySQL memiliki kecepatan yang menakjubkan dalam manangani query sederhana,

dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed/unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. MySQL memiliki lapisan keamanan seperti level subnetmask, nama *host*, dan izin akses user dengan sistem perizinan yang mendatail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan *protokol* TCP/IP, Unix *socket* (UNIX), atau *Named Pipes* (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antar muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API *(Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tools*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

Struktur Tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.16 PHP (*Hypertext Preprocessor*)

Hypertext Preprocessor atau disingkat PHP, adalah bahasa *script server side* didesain untuk pengembangan web untuk menghasilkan sebuah web dinamis. Untuk itu, kode PHP disisipkan kedalam skrip HTML dan diproses atau diterjemahkan oleh web server dengan sebuah modul processor PHP untuk kemudian menghasilkan halaman web.

2.16.1 Sejarah PHP

Pada awalnya PHP merupakan singkatan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengola data formulis dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP.FI. dengan perilisan kode sumber ini menjadi sumber

terbuka, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Dalam tahun yang sama, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih., lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP, *Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

2.16.2 Kelebihan PHP

Beberapa kelebihan PHP dari bahasa pemrogram web, antara lain :

1. Bahasa pemrograman PHP adalah sebuah skrip yang tidak melakukan sebuah kompilasi dalam penggunaannya.

2. Web Server yang mendukung PHP dapat ditemukan dimana-mana dari mulai Apache, IIS, Lightpd hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai sistem operasi (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.17 XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdari dari program Apache, HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), A (Apache), M (MySQL), P (PHP), P (Perl). Program ini tersedia dibawah lisensi GNU Public Lisence dan bertindak sebagai web server yang mengelola halaman web dinamis. XAMPP tersedia untuk sistem operasi Microsoft Windows, Linux, Solaris, dan Mac OS. Umumnya perangkat lunak ini digunakan untuk projek pengembangan web.

2.18 Eclipse

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

1. *Multi-platform*: Dapat berjalan pada berbagai platform seperti, Microsoft Windows, Linux, Solaris, AIX, HP-UX, dan Mac OS X.
2. *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Phyton, Perl, PHP, dan lain sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya

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

2.18.1 Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM *Visual Age for Java* 4.0. Produk ini diluncurkan oleh

IBM pada tanggal, 5 november 2001, yang menginvestasikan sebanyak US$40 juta untuk pengembangannya. Semenjak itu konsursium Eclipse *Foundation* mengambil alih untuk pengembangan lebih lanjut dan pengaturan organisasinya.

2.18.2 Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkut plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan *Rich Client Platform* (RCP). Berikut ini adalah komponen yang membentuk RCP :

1. *Core Platform*
2. OSGi
3. SWT (*Standard Widget Toolkit*)
4. JFace
5. *Eclipse Workbench*

Standard pengoperasian Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in* nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal

plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (C/C++ *Development Tools*). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya.

2.18.3 Perkembangan Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama *Simultaneous Release*. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah pengembangan (*deployment*) dan perawatan (*maintenance*) untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya.

Tabel II.2 Perkembangan Versi Eclipse

Kode Peluncuran	Tanggal Peluncuran	Platform	Nama Proyek
Eclipse 3.0	21 Juni 2004	3.0	-
Eclipse 3.1	28 Juni 2005	3.1	-
Callisto	30 Juni 2006	3.2	Callisto projects
Europa	29 Juni 2007	3.3	Europa projects
Ganymede	25 Juni 2008	3.4	Ganymede projects
Galileo	24 Juni 2009	3.5	Galileo projects
Helios	23 Juni 2010	3.6	Indigo projects
Juno	27 Juni 2012	4.2	Juno projects

2.19 *Software Development Kit* (SDK)

Software Development Kit (SDK atau *devkit*) adalah sekumpulan alat pengembangan yang memungkinkan untuk menciptakan sebuah aplikasi untuk paket perangkat lunak tertentu (*software package*), framework perangkat lunak, platform perangkat keras, komputer sistem, konsol video game, sistem operasi, atau platform yang sama.

2.20 *Java Development Kit* (JDK)

Java Development Kit atau disingkat JDK merupakan produk dari Oracle Corporation yang ditujukan untuk pengembangan perangkat lunak berbasis Java. Sejak Java diperkenalkan, sampai saat ini Java SDK telah banyak digunakan. Pada tanggal, 17 November 2006, Sun memperkenalkan JDK dibawah lisensi GNU *General Public License* (GPL), sehingga penggunaannya bebas (gratis).

JDK juga dilengkapi dengan JRE (*Java Runtime Environment*), biasanya disebut *private*. JRE terdiri dari JVM (*Java Virtual Machine*) dan semua *library class* yang terdapat lingkungan produksi, sama baiknya dengan penambahan hanya library-library yang berguna bagi *developers*, seperti *library internationalization* dan *library* IDL.

2.21 *Android Development Tools* (ADT)

Android Development Tools (*ADT*) adalah *plugin* untuk Eclipse *Intergrated Development Environment* (*IDE*) yang dirancang untuk

memberikan lingkungan yang terpadu di mana untuk membangun aplikasi Android.

ADT memperluas kemampuan Eclipse untuk membiarkan para *developer* lebih cepat dalam membuat proyek baru Android, membuat aplikasi UI, menambahkan komponen berdasarkan Android *Framework* API, *debug* aplikasi dalam pengunaan Android SDK, dan membuat file APK untuk mendistribusikan aplikasi.

Mengembangkan aplikasi di Eclipse dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi android, karena banyak kemudahan-kemudahan sebagai tools yang terintegrasi seperti, custom XML editor, dan *debug panel ouput*. Selain itu ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.

2.22 Augmented Reality Browser

Augmented reality browser (*user agent*) adalah sebuah aplikasi perangkat lunak yang bertujuan untuk mengambil, menampilkan, dan melintaskan sumber informasi melalui antarmuka *augmented reality*.

Fitur-fitur pada mobile AR browser (*user agent*) adalah sebagai berikut :

1. Mengumpulkan dan mengelola kedua sumber baik aplikasi maupun penyedia data pihak ketiga.
2. Menghubungkan *web service* atau fungsi ponsel seperti komunikasi suara, pesan atau email.
3. Merender objek 2 dimensi dan 3 dimensi.

4. Terhubung dengan kamera ponsel untuk menampilkan hasilnya (*output*).
5. Mencari sumber informasi yang mengikat data sensor dengan konten digital

Berikut daftar AR browser yang ada saat ini :

Tabel II.3 Daftar AR browser

No	Mobile AR Browser	Vendor	Lokasi	Homepage	Platform pendukung
1	Argon (KHARMA)	Georgia Tech	USA	https://research.cc.gatech.edu/polaris	iPhone
2	Acrossair	Acrossair	UK	http://www.acrossair.com	iPhone
3	Google Goggles	Google Inc.	USA	http://www.google.com/mobile/goggles	iPhone, Android
4	InstantReality	FraunhoferlGD	Germany	http://www.instantreality.org/	-
5	Junaio	Metaio Gmbh	Germany	http://www.junaio.com/	iPhone, Android
6	Kooaba	Kooaba	Swistzerland	http://www.kooaba.com/	iPhone
7	Layar	Layar B.V	Netherlands	http://www.layar.com/	iPhone, Android, Symbian, Bada
8	LibreGeoSocial	-	Spain	http://www.libregeosocial.org/	Android
9	Mixare	-	Italy	http://www.mixare.org	Android, iPhone
10	Ovjet	Kiwiple Inc.	Korea	http://ovjet.com/	iPhone, Android
11	Point and Find	Nokia	Finland	http://pointandfind.nokia.com/main_publisher	Symbian

12	ScanSearch	OlaWorks	Korea	http://www.scansearch.com/	iPhone, Android
13	Sekai Camera	Tonchidot Corp.	Japan	http://sekaicamera.com	iPhone, Android
14	Wikitude	Mobilizy	Austria	http://www.wikitude.org	iPhone, Android

Sumber. AR Landscape / Browser (www.w3.org/2010/POI/wiki/AR_Landscape/Browsers)

Mixare – *Open Source Augmented Reality Engine*

Mixare adalah browser *augmented reality* yang sifatnya bebas (*open source*), dipublikasikan dibawah lisensi GPLv3 (GNU *General Public Lisence, Free Software Foundation*, Inc). Mixare tersedia untuk perangkat yang menggunakan sistem operasi Android dan iPhone. Sistem ini bekerja sebagai *completely autonomous application* dan tersedia bebas bagi para pengembang.