

BAB 2

TINJAUAN PUSTAKA

2.1 Fotografi

Fotografi berasal dari kata foto yang berarti cahaya dan grafis yang berarti gambar. Dengan berkembangnya teknologi digital yang sangat pesat saat ini bahkan hampir semua orang.

Secara harfiah fotografi bisa diartikan sebagai teknik melukis dengan cahaya. Fotografi merupakan gabungan ilmu, teknologi, dan seni. Perpaduan yang harmonis antara ketiganya bisa menghasilkan sebuah karya yang mengagumkan. Tentunya dengan *skill* serta sentuhan seni sang fotografer, sebuah foto bisa menjadi berarti.

2.2 Sejarah Perkembangan Fotografi di Indonesia

Kassian Cephas, lahir di Yogyakarta, 15 Januari 1845. Olehbanyak pihak diakui sebagai fotografer pertama Indonesia. Fotografer lainnya yang ada di Indonesia sebagian adalah keturunan Belanda. Kassian Cephas yang tinggal dan punya studio di Yogyakarta juga merupakan “pemotret resmi” Kraton Yogyakarta.

Selain itu, ada pula **Ansel Adam** seorang “*fine art photographer*” Amerika terbesar dari abad ke-20. Ansel adam tidak hanya dihargai dari karya foto-fotonya saja, juga dari dedikasinya dalam dunia pendidikanfotografi. Ansel bersama **Fred Archer** pada 1940-an memperkenalkan suatu metode yang dikenal dengan nama *zone system* (ZS).

Metode temuan Ansel ini secara umum adalah proses terencana dalam pembuatan foto, mulai dari pra-visualisasi kemudian mengkalkulasi pencahayaan secara tepat sampai memproses film secara akurat. Hasil akhirnya adalah negative foto yang prima sebagai pondasi utama membuat cetakan foto yang berkualitas juga maksimal.

Era reformasi yang dimulai pada tahun 1998 di Indonesia telah membawa dampak yang besar bagi segala aspek di masyarakat.[3]

Berikut ini adalah jenis-jenis fotografi :

1. *Portrait Photography*

Yang menjadi kekuatan utama dari jenis fotografi yang satu ini adalah karakteristik dan kepribadian yang unik dari setiap orang. Bukan semata menampilkan foto manusia saja, *portrait photography* yang baik juga harus mampu menangkap ekspresi, mimik, kepribadian maupun suasana hati seseorang agar foto yang dihasilkan lebih berkesan. Untuk mendapatkan itu, maka wajah seseorang menjadi fokus utama agar kesan emosionalnya dapat dimunculkan.

2. *Landscape Photography*

Tuhan menciptakan semesta yang terlalu indah untuk dilewatkan. Maka dari itu, *landscape photography* memfokuskan objeknya pada pemandangan alam seperti pantai ataupun gunung. Agar hasil dari *landscape photography* bagus, diperlukan moment yang tepat untuk memotret. Misalnya saat matahari terbit atau terbenam saat cuacanya cerah.

3. *Human Interest Photography*

Hampir sama dengan *portrait photography*, objek utama dari *human interest photography* adalah manusia. Namun, tentunya ada yang menjadi pembeda diantara keduanya.

Jika *portrait photography* fokus untuk menangkap ekspresi manusia yang menjadi objeknya, *human interest photography* lebih menonjolkan bagaimana sisi kehidupan dan interaksi manusia dengan lingkungan sekitar dalam kesehariannya dengan tujuan membangkitkan perasaan simpati maupun empati dari penikmat foto tersebut.

4. *Aerial Photography*

Foto-foto yang dihasilkan oleh jenis fotografi ini membuat seolah-olah manusia memiliki mata seekor burung yang melihat segalanya dari atas.

Aerial photography memiliki ciri yang unik dari pengambilannya, yaitu dilakukan dari atas (*high angel*).

5. *Stage Photography*

Jenis fotografi ini menjadikan orang diatas panggung sebagai objeknya. Gerakannya yang sulit diprediksi ditambah dengan tata lampu (*lighting*) yang berubah-ubah menjadi tantangan jika ingin menggeluti jenis *stage photography*. Kecepatan dan ketepatan mengambil momen menjadi hal yang penting untuk dikuasai.

6. *Wildlife Photography*

Jenis fotografi yang satu ini umumnya hanya dilakukan oleh para fotografer profesional yang menjadi kontributor di sebuah media. Fotografi ini fokus untuk mengabadikan kehidupan alam liar di habitat aslinya dan lebih memacu adrenalin ketika yang ditemui adalah hewan-hewan yang buas.

7. *Macro Photography*

Jenis fotografi dari *macro photography* adalah membuat objek berukuran kecil untuk dapat tampak sangat dekat dengan menampilkan detail yang apik. Untuk melakukan jenis fotografi ini, diperlukan kamera dengan fitur **zoom** yang bagus agar bias mendapatkan detil-detail objek yang kecil tersebut.

8. *Photojournalism / Journalism Photography*

Jenis fotografi ini hampir sama dengan *wildlife photography*, biasanya dilakukan oleh fotografer profesional yang menjadi kontributor dari sebuah media. Tujuan utama dari *photojournalism* ini adalah menyajikan foto yang memiliki cerita atau nilai berita. Kemudian, foto tersebut dapat digunakan untuk keperluan penanyangan berita atau publikasi di media massa. Objek-objek yang diambil oleh jenis fotografi ini merupakan hal-hal atau kejadian

yang terjadi disekitar dengan prinsip utama yaitu menghasilkan foto yang merupakan kejadian sesungguhnya tanpa rekayasa dan memihak.

9. *Fashion Photography*

Keindahan desain pakaian seperti gaun maupun aksesoris *fashion* lainnya menjadi titik fokus utama dari jenis fotografi ini. Peran dari seorang model peraga *fashion* memang tidak dapat dipungkiri kehadirannya, sehingga terkadang *fashion photography* menjadi bias dengan *portrait photography*, tetapi *fashion photography* akan menekankan fokusnya pada desain pakaian agar orang-orang bias tertarik untuk membelinya.

10. *Street Photography*

Realitas yang terjadi di jalan merupakan prinsip utama dari jenis *street photography* ini. Sekilas akan tampak sama dengan jenis-jenis sebelumnya, yaitu *human interest photography* atau *photojournalism*. Namun, ada ciri khas yang ditawarkan oleh *street photography*, yaitu suatu pendekatan yang berusaha menampilkan realitas sesungguhnya yang terjadi di ruang publik secara spontan.

11. *Architectural Photography*

Jenis fotografi ini menampilkan keindahan bentuk bangunan atau gedung. Fotografer untuk *architectural photography* harus jeli dalam memerhatikan setiap sudut dan celah bangunan agar dapat memperoleh komposisi yang ritmis.

Selain itu, angle foto juga bias menjadi faktor yang penting agar foto yang dihasilkan tersebut tidak tampak statis. *Architectural photography* ini sering dipakai untuk keperluan komersil, seperti promosi hotel, apartemen maupun *real estate*.

12. *Sport Photography*

Jenis fotografi dari *sport photography* adalah berusaha untuk mengabadikan setiap kejadian yang menarik di tengah lapangan atau lintasan. Untuk menangkap objek dengan baik dalam jenis fotografi ini, fotografer lebih baik menggunakan kamera yang dilengkapi dengan lensa tele.

13. *Food Photography*

Fotografi jenis ini menuntut fotografernya agar mampu menciptakan komposisi yang bagus dan memperlihatkan detail serta tekstur makanan yang dibidiknya.

14. *Still Life Photography*

Jenis fotografi ini menawarkan tantangan besar yaitu bagaimana membuat objek benda mati yang diambil dapat tampak lebih hidup. Dengan kata lain, objek yang nantinya tidak sekedar menampilkan benda mati saja. Namun, lebih dari itu, *still lifephotography* mampu bercerita kepada penikmat foto lewat komposisi, properti dan tentunya pencahayaan yang bagus.[4]

2.3 *Global Positioning System (GPS)*

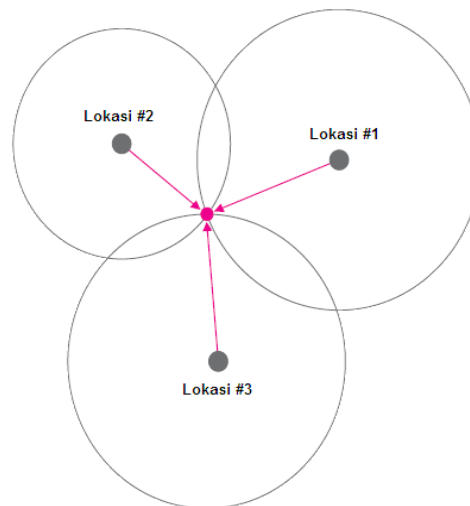
Global Positioning Sytem (GPS) merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunanya dimana dia berada (secara *global*) di permukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal *radio* dengan data *digital*.

2.3.1 *Definisi Global Positioning System (GPS)*

GPS (*Global Positioning System*) adalah sistem navigasi yang berbasiskan satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan (*Department of Defense*) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit. Untuk mengetahui posisi seseorang maka diperlukan alat yang bernama GPS *receiver* yang berfungsi untuk menerima sinyal yang dikirim dari satelit

GPS. Posisi diubah menjadi titik yang dikenal dengan nama *Way-point* nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi, kemudian di layar pada peta elektronik.

GPS adalah satu-satunya sistem satelit navigasi global untuk penentuan lokasi, kecepatan, arah, dan waktu yang telah beroperasi secara penuh di dunia saat ini. GPS menggunakan konstelasi 27 buah satelit yang mengorbit bumi, dimana sebuah GPS *receiver* menerima informasi dari tiga atau lebih satelit tersebut seperti terlihat dalam gambar 2.1 dibawah, untuk menentukan posisi. GPS *receiver* harus berada dalam *line-of sight* (LoS) terhadap ketiga satelit tersebut untuk menentukan posisi, sehingga GPS hanya ideal untuk digunakan dalam *outdoor positioning*.



Gambar 2.1 Trilaterasi Dalam GPS

Aplikasi yang berada disisi target (*client*) setelah mendapatkan *request* dari pelacak (*server*) maka *client* akan meminta koordinat posisinya pada GPS (*Global Positioning System*) , yang kemudian akan dikirimkan ke pelacak (*server*).

Sejak tahun 1980, layanan GPS yang dulunya hanya untuk keperluan militer mulai terbuka untuk publik. Meskipun satelit-satelit tersebut berharga ratusan juta dollar, namun setiap orang dapat menggunakannya secara gratis. Satelit-satelit ini mengorbit pada ketinggian sekitar 12.000 mil dari

permukaan bumi. Posisi ini sangat ideal karena satelit dapat menjangkau *area coverage* yang lebih luas. Satelit-satelit ini akan selalu berada di posisi yang menjangkau semua area di atas permukaan bumi sehingga dapat meminimalkan terjadinya *blank spot* (area yang tidak terjangkau oleh satelit).

Setiap satelit hanya mampu mengelilingi bumi hanya dalam waktu 12 jam. Sangat cepat, sehingga mereka selalu bisa menjangkau dimana pun posisi anda di atas permukaan bumi. GPS *receiver* sendiri berisi beberapa *intergrated circuit (IC)* sehingga murah dan teknologinya mudah untuk digunakan oleh semua orang. GPS dapat digunakan untuk berbagai kepentingan, misalnya mobil, kapal, pesawat terbang, pertanian dan diintegrasikan dengan komputer maupun laptop.[5]

Berikut beberapa contoh perangkat GPS *receiver* :



Gambar 2.2 Macam-macam Perangkat GPS

2.3.2 Cara Kerja *Global Positioning System (GPS)*

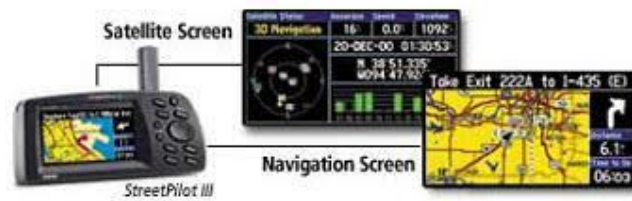
Setiap daerah di atas permukaan bumi ini minimal terjangkau oleh 3-4 satelit. Pada prakteknya, setiap GPS terbaru bisa menerima sampai dengan 12 *channel* satelit sekaligus. Kondisi langit yang cerah dan bebas dari halangan membuat GPS dapat dengan mudah menangkap sinyal yang dikirimkan oleh satelit. Semakin banyak satelit yang diterima oleh GPS, maka akurasi yang diberikan juga akan semakin tinggi. Cara kerja GPS secara sederhana ada 5 langkah, yaitu :

1. Memakai perhitungan “*triangulation*” dari satelit.
2. Untuk perhitungan *triangulation*, GPS mengukur jarak menggunakan *travel time* sinyal *radio*.
3. Untuk mengukur *travel time*, GPS memerlukan akurasi waktu yang tinggi.
4. Untuk pengukuran jarak, kita harus tahu dengan pasti posisi satelit dan ketinggian pada orbitnya.
5. Terakhir harus mengoreksi *delay* sinyal waktu perjalanan di atmosfer sampai diterima *receiver*.



Gambar 2.3 Cara Kerja Satelit GPS

Satelit *GPS* berputar mengelilingi bumi selama 12 jam di dalam orbit yang akurat dan mengirimkan sinyal informasi ke bumi. *GPSreceiver* mengambil informasi itu dan dengan menggunakan perhitungan “*triangulation*” menghitung lokasi *user* dengan tepat. *GPSreceiver* membandingkan waktu sinyal di kirim dengan waktu sinyal tersebut diterima. Dari informasi itu dapat diketahui berapa jarak satelit. Dengan perhitungan jarak *GPSreceiver* dapat melakukan perhitungan dan menentukan posisi *user* dan menampilkan dalam peta elektronik.



Gambar 2.4 Tampilan GPS Receiver

Sebuah *GPS receiver* harus mengunci sinyal minimal tiga satelit untuk menghitung posisi 2D (*latitude* dan *longitude*) dan *track* pergerakan. Jika *GPS receiver* dapat menerima empat atau lebih satelit, maka dapat menghitung posisi 3D (*latitude*, *longitude* dan *altitude*). Jika sudah dapat menentukan posisi *user*, selanjutnya *GPS* dapat menghitung posisi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan masih banyak lagi.

Satelit *GPS* dalam mengirim informasi waktu sangat presisi karena satelit tersebut memakai jam atom. Jam atom yang ada pada satelit jalan dengan partikel atom yang di isolasi, sehingga dapat menghasilkan jam yang akurat dibandingkan dengan jam yang biasa. Perhitungan waktu yang akurat sangat menentukan akurasi perhitungan untuk menentukan informasi lokasi kita. Selain itu semakin banyak sinyal satelit yang dapat diterima maka akan semakin presisi data yang diterima karena ketiga satelit mengirim *pseudo-random code* dan waktu yang sama. Ketinggian itu menimbulkan keuntungan dalam mendukung proses kerja *GPS*, bagi kita karena semakin tinggi maka semakin bersih atmosfer, sehingga gangguan semakin sedikit dan orbit yang cocok dan perhitungan matematika yang cocok. Satelit harus tetap pada posisi yang tepat sehingga stasiun di bumi harus terus memonitor setiap pergerakan satelit, dengan bantuan radar yang presisi selalu di cek tentang *altitude*, *posision* dan kecepatannya.

2.3.3 Cara Satelit Menentukan Posisi Lokasi

Sinyal yang dikirimkan oleh satelit ke *GPS* akan digunakan untuk menghitung waktu perjalanan (*travel time*). Waktu perjalanan ini sering juga disebut sebagai *Time of Arrival (TOA)*. Sesuai dengan prinsip fisika, bahwa untuk mengukur jarak dapat diperoleh dari waktu dikalikan dengan cepat rambat sinyal. Maka, jarak antara satelit dengan *GPS* juga dapat diperoleh dari prinsip fisika tersebut. Setiap sinyal dikirimkan oleh satelit akan juga berisi informasi yang sangat detail, seperti orbit satelit, waktu, dan hambatan di atmosfer. Satelit menggunakan jam atom yang merupakan satuan waktu paling presisi. Untuk dapat menentukan posisi dari sebuah *GPS* secara dua dimensi (jarak), dibutuhkan minimal tiga buah satelit. Empat buah satelit dibutuhkan agar didapatkan lokasi ketinggian (secara tiga dimensi). Setiap satelit akan memancarkan sinyal yang akan diterima oleh *GPS receiver*. Sinyal ini akan dibutuhkan untuk menghitung jarak dari masing-masing satelit ke *GPS*. Dari jarak tersebut, akan diperoleh jari-jari lingkaran jangkauan setiap satelit. Lewat perhitungan matematika yang cukup rumit, interseksi (perpotongan) setiap lingkaran jangkauan satelit tadi akan dapat digunakan untuk menentukan lokasi dari *GPS* di permukaan bumi.[5]

2.3.4 Manfaat Penggunaan *Global Positioning System (GPS)*

Dengan menggunakan *GPS*, anda dapat menandai semua lokasi yang pernah dikunjungi. Misalnya, lokasi Politeknik Negeri Sriwijaya diberi *waypoint* dan tempat-tempat lainnya. Sebenarnya, ada banyak manfaat yang bisa diambil jika mengetahui *waypoint* dari suatu tempat. Pertama, *user* dapat memperkirakan jarak lokasi yang dituju dengan lokasi asal *user*. *GPS* keluaran terakhir dapat memperkirakan jarak *user* ke tujuan, sampai estimasi lamanya perjalanan dengan kecepatan aktual yang sedang *user* tempuh. Kedua, lokasi daratan memang cukup mudah untuk dikenali dan diidentifikasi. Namun, jika *user* kebetulan berada di tempat memancing yang terletak ditengah lautan ataupun tempat melihat matahari terbenam yang berada di puncak gunung. Disaat seperti inilah *GPS* akan menunjukkan manfaatnya.

Dengan teknologi *GPS* dapat digunakan untuk beberapa keperluan sesuai dengan tujuannya. *GPS* dapat digunakan oleh peneliti, olahragawan, petani, tentara, pilot, petualang, pendaki, pengantar barang, pelaut, penebang pohon dan dengan berbagai kepentingan untuk meningkatkan produktivitas, keamanan, dan untuk kemudahan. Dari beberapa pemakaian diatas dikategorikan menjadi :

1. Lokasi

Digunakan untuk menentukan dimana lokasi suatu titik dipermukaan bumi berada

2. Navigasi

Membantu mencari lokasi suatu titik di bumi.

3. *Tracking*

Membantu untuk memonitoring pergerakan objek dan membantu memetakan posisi tertentu, dan perhitungan jaingan terdekat.

4. *Timing*

Dapat dijadikan dasar penentuan jam seluruh dunia, karena memakai jam atom yang jauh lebih presisi di banding dengan jam biasa.[5]

2.3.5 Model DanInterkoneksi *Global Positioning System (GPS)*

Sebuah *GPS* juga memiliki *firmware* yang bisa di upgrade. *Upgradefirmware* ini biasanya disediakan pada site produsen *GPS* tersebut. *Upgrade firmware* biasanya menggunakan kabel yang dibundel ataupun tersedia sebagai aksesoris. Kabel ini juga ternyata bisa digunakan untuk menghubungkan *GPS* ke komputer. *SoftwareGPS* yang tersedia untuk beberapa *platform* tersebut juga cukup banyak. Dengan *software* tersebut, *user* dapat dengan mudah *download* informasi dari *GPS*. Memori dari sebuah *GPS* memang relatif terbatas, sehingga kemampuan ekstra untuk menyimpan informasi yang pernah *user* tempuh ke *PC* (yang biasanya memiliki memori lebih besar) tentu akan sangat menyenangkan. Untuk media komunikasi *GPS* dengan *hardware* lain selain kabel, model *GPS* sekarang juga ada yang dilengkapi dengan *Bluetooth*, *Infrared*.

Berdasarkan fisik, model *GPS* dibagi menjadi beberapa tipe antara lain model *portable/handheld* (ukurannya menyerupai ponsel), ada yang lebih besar (biasanya di mobil/kapal), ada pula yang menggunakan interface khusus untuk di koneksikan ke notebook maupun PDA(*Palm, pocket PC* maupun *Nokia Communicator*). *GPS* untuk keperluan outdoor biasanya juga dilengkapi dengan perlindungan anti air dan tahan benturan. Beberapa *GPS* keluaran terakhir bahkan sudah menyediakan layar warna dan kemampuan komunikasi radio jarak pendek (*FRS/Family Radio Service*). [6]

2.4 Android



Gambar 2.5 Logo Android

Android adalah sebuah kumpulan perangkat lunak untuk perangkat mobile untuk mencakup system informasi, middleware dan aplikasi utama dan aplikasi utama mobile. Android memiliki karakteristik sebagai berikut :

1. Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera dan lain-lain. Android menggunakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan *open source*, dapat dengan bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat

teknologi itu muncul. *Platform* ini akan terus berkembang untuk membangun aplikasi *mobile* yang inovatif.

2. Semua aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi yang utama dari telepon dan aplikasi pihak ketiga (*third party application*). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan aplikasi yang luas terhadap para pengguna.

3. Memecahkan hambatan pada aplikasi

Android memecahkan hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang di peroleh dari *web* dengan data pada ponsel seseorang seperti kontak pengguna, kalender atau lokasi geografis.

4. Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan *library* yang di perlukan dan *tools* yang dapat digunakan untuk membangun aplikasi yang semakin baik. Android memiliki sekumpulan *tools* yang dapat digunakan sehingga membantu pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

Google Inc. sepenuhnya membangun Android dan menjadikannya bersifat terbuka (*open source*) sehingga para pengembang dapat menggunakan Android tanpa mengeluarkan biaya untuk lisensi dari *Google* dan dapat membangun Android tanpa adanya batasan-batasan.

Android Software Development Kit (SDK) menyediakan alat dan *Application Programming Interface (API)* yang diperlukan untuk mulai mengembangkan aplikasi pada *Platform* android menggunakan bahasa pemrograman *Java*. [7]

2.4.1 Sejarah Singkat Android

Telepon seluler menggunakan berbagai macam system operasi seperti *Symbian OS*, *Microsoft's Windows Mobile*, *Mobile Linux*, *Iphone OS*, *Moblin* dan berbagai macam sistem operasi lainnya. *API* yang tersedia untuk mengembangkan aplikasi *mobile* terbatas dan oleh karena itu *Google* mulai mengembangkan dirinya. *Platform Android* menjanjikan keterbukaan, kemudahan untuk menjangkau, *sourcecode* yang terbuka dan pengembang *framework* yang *high end*.

Google membeli perusahaan *Android Inc.* yang merupakan sebuah perusahaan kecil yang berbasis mengembangkan perangkat lunak untuk ponsel, pada tahun 2005 untuk memulai pengembangan pada *platform android*. Tokoh utama yang meliputi *Android Inc.* *Andy Rubin*, *Rich Miner*, *Nick Sears* dan *Chris White*.

Pada tanggal 5 November 2007, kelompok pemimpin industri bersama-sama membentuk *Open Handset Alliance* (OHA) yang diciptakan untuk mengembangkan standar terbuka bagi perangkat mobile. *OHA* terdiri dari 34 anggota besar dan beberapa anggota yang terkemuka diantaranya adalah *Sprint Nextel*, *T-Mobile*, *Motorola*, *Samsung*, *Sony Ericson*, *Toshiba*, *Vodafone*, *Google*, *Intel*, dan *Texas Instrument*. [8]

2.4.2 Perkembangan Sistem Operasi Android

Sejak pertama kali diluncurkan *Android* masih banyak kekurangan, tetapi kekurangan tersebut dengan cepat diperbaiki. Bahkan dengan penambahan fitur-fitur yang lebih menarik pada versi *Android* selanjutnya. *Mobile phone* pertama yang menggunakan system operasi *Android* versi pertama adalah *HTC*, dan sekarang sudah banyak *mobile* yang menggunakan *Android*. Berikut perkembangan versi operasi sistem *Android*.

1. *AndroidVersion 1.0 beta*

Android beta pertama kali dirilis tanggal 5 november 2007 dan versi *SDK (Software Development Kit)* beta dirilis pada tanggal 12 November 2007 oleh *Google*

2. Android Version 1.0

Android 1.0 merupakan *software* android komersial yang dirilis pada tanggal 23 September 2008. Sebenarnya Android versi pertama ini akan dinamai dengan nama “*Astro*”, tetapi karena alasan hak cipta dan *trademark*, nama “*Astro*” tidak jadi disematkan pada versi dari OS Android ini. Kemudian perangkat Android pertama yang menggunakan dan memperkenalkan Android 1.0 adalah ponsel *HTC Dream (G1)*.

3. AndroidVersion 1.1

Pada 9 maret 2009, versi Android 1.1 diluncurkan. Versi android ini memiliki masalah penamaan yang sama dengan versi sebelumnya. Pada awalnya Android ini akan dinamakan “*Bender*” tetapi karena alasan *trademark*, nama “*Bender*” tidak jadi disematkan pada versi Android ini. Awalnya hanta untuk *T-Mobile G1*, akan tetapi aplikasi ini guna memecahkan masalah (*bug*), merubah API dan menambahkan sejumlah fitur dari versi sebelumnya. Versi ini belum begitu populer sejak awal peluncurannya. Android versi ini dilengkapi dengan tampilan baru pada aplikasi, GUI (*Graphical User Interface*), jam alarm, voice search(pencarian suara) yang memungkinkan pencarian melalui suara manusia, pengiriman *e-mail* menggunakan *Google Mail* dan *push e-mail* atau pemberitahuan *e-mail* yang masuk ke alamat *e-mail* yang kita miliki.

4. AndroidVersion 1.5 (CupCake)

Versi Android perama yang mempunyai nama, yaitu Android 1.5, *CUPCAKE*. Mulai versi Android ini, penamaan menggunakan nama makanan pencuci mulut (*dessert*) mulai digunakan. Versi ini dirilis pada pertengahan Mei 2009. Android *CupCake* diluncurkan dengan Android dan SDK (*Software Development Kit*) sehingga memudahkan antar PC dan Telepon seluler. Versi ini berbasis *Linux Kernel* 2.6.27. Pembaruan pada versi ini berupa sejumlah fitur baru dan perubahan *interface*. Penambahan beberapa fitur pada versi ini, diantaranya

penambahan kemampuan merekam *video* dan *audio*, serta menontonnya pada aplikasi kamera, dan juga bisa langsung meng-*upload* atau mengunggahnya ke media sosial seperti *Youtube*, sedangkan foto bisa diunggah ke *Pisaca* secara langsung melalui telepon. Kemampuan *Bluetooth A2DP* memungkinkan secara otomatis terhubung ke *headset Bluetooth*. Penyesuaian tampilan, animasi yang ada, dan perbaikan *keyboard* bisa diubah melalui pengaturan yang disediakan. Versi Android selanjutnya, mengikuti urutan abjad huruf, mulai dari C, D, E, F, G, H, I.

5. AndroidVersion 1.6 (*Donut*)

Pada tanggal 16 September 2009, SDK *Android 1.6* yang bernama *Donut* dirilis. Android tipe ini berbasis *Linux Kernel 2.6.29*. Menampilkan proses pencarian yang lebih baik dibandingkan sebelumnya, penggunaan baterai indikator yang bisa lebih memudahkan pembacaan status baterai kita, sehingga kita bisa mengetahui apabila baterai sudah mendekati habis, maka kita pun siap untuk men-*charging* telepon, dan kontrol *applet VPN* sudah bisa digunakan pada android versi ini. Didalamnya terdapat sejumlah fitur baru, diantaranya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus pada kamera, *camcorder* dan galeri yang diintegrasikan di *CDMA/EVDO*, *802.1x*, *VPN (Virtual Private Network)*, *gestures* dan *text-to-speech engine* (kemampuan dial kontak), teknologi *text-to-change speech* (teknologi ini tidak terdapat pada semua jenis telepon seluler), pengadaan resolusi *VWGA*.

6. AndroidVersion 2.0/2.1 (*Éclair*)

Pada Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Éclair*).*Éclair* adalah kue sus yang berbentuk panjang dengan topping coklat diatasnya. Pada saat itu *Motorolla Droid* merupakan perangkat yang menggunakan *Éclair*. Perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan *Google Maps 3.1.2*, perubahan *UI* dengan *browser* baru dan dengan dukungan dari

HTML 5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2MP, *Digital Zoom* dan *Bluetooth* 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, *Google* melakukan investasi dengan mengadakan kompetisi aplikasi *mobile* terbaik (*killer apps*). Kompetisi ini berhadiah \$25000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap, yang tiap tahapnya dipilih 50 aplikasi terbaik.

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

7. AndroidVersion2.2 (Froyo : Frozen Yoghurt)

Android versi 2.2 yang berbasis *Linux Kernel* 2.6.32, diluncurkan pada tanggal 20 mei 2010. Sistem operasi Android 2.2 ini diberi nama *Froyo*. *Froyo* (*Frozen Yoghurt*) adalah *yoghurt* (susu yang dibuat melalui fermentasi bakteri) yang dibekukan mirip seperti *ice cream*. Perubahan-perubahan umumnya pada versi sebelumnya antara lain dukungan *Adobe Flash* 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat dari versi sebelumnya, juga terdapat fitur *Task Manager* yang memudahkan pengguna mengatur aplikasi-aplikasi yang terinstal, integrasi *V8 Java Script Engine* dipakai oleh *Google Chrome* yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi pada *SD Card*, kemampuan *Wi-fi Hotspot Portable*, dan kemampuan *auto update* dalam aplikasi *Android Market*. Selain itu, dukungan *multitouch* dan *pinch to zoom* sudah ada di *Android Froyo*. *User interface*-nya pun menjadi lebih menarik untuk dilihat, serta *Launcher* yang digunakan pada *Froyo* ini juga lebih baik dari sebelumnya. Versi *Froyo* sudah banyak digunakan

pada beberapa merk telepon seluler terkenal. Ponsel pintar yang pertama menggunakan Android *Froyo* adalah *Google Nexus One*.

8. *AndroidVersion 2.3 (Ginger Bread)*

Pada tanggal 6 desember 2010, Android 2.3 (*Ginger Bread*) diluncurkan. *Gingerbread* adalah kue yang terbuat dari jahe, biasanya berbentuk boneka sering disajikan sebagai teman minum kopi. Versi ini berbasis *Linux Kernel 2.6.35*. Perubahan-perubahan umum yang didapat pada 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, handphone virtualization*, dan *bass booster*), dukungan kemampuan *Near Field Communication (NFC)*, dukungan kamera yang lebih dari satu, yaitu *Front Camera* dan *Camera Standar*, perbaikan terhadap dukungan layar resolusi WXGA dan di atasnya. Beberapa versi *update* yang dirilis antara lain v.2.3.3 hingga v.2.3.7. Sampai saat ini Android *Gingerbread* merupakan versi Android yang memiliki pengguna terbanyak dibandingkan dengan seri Android lainnya, yaitu mencapai 65% dari seluruh versi Android yang dirilis. *Google NexusS* merupakan perangkat *mobile* pertama yang hadir dengan Android *Gingerbread* ini.

9. *AndroidVersion3.0 (Honeycomb)*

Pada tanggal 22 februari 2011, Android 3.0 (*Honeycomb*) yang merupakan versi android pertama yang diperuntukkan untuk perangkat tablet diluncurkan, Android ini berbasis *Linux Kernel 2.6.36*. Android versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah di desain untuk tablet. *Honeycomb* juga sudah mendukung *multiprocessor* dan juga meningkatkan kinerja *hardware* serta akselerasi *hardware* untuk grafis. Tablet pertama yang dibuat untuk menjalankan *Honeycomb* adalah *Motorolla Xoom* dirilis pada 24 februari 2011.

Perangkat tablet dengan *platform* Android 3.0 saat itu hadir di Indonesia. Perangkat tersebut bernama *Eee Pad Transformer* produksi dari Asus. Saat itu masuk pasar Indonesia pada Mei 2011. *Photo browser 3D* merupakan aplikasi yang ada pada Android *Honeycomb*, dimana sangat memukau karena kemampuannya menampilkan *slide show* foto 3 dimensi.

10. Android Version 4.0 (*Ice Cream Sandwich*)

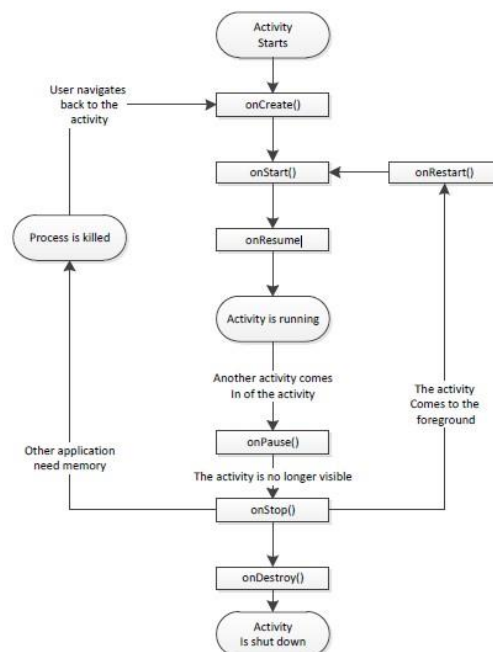
Android 4.0 atau *Ice cream sandwich* dirilis untuk publik pada 19 Oktober 2011 dan berbasis *Linux Kernel* 3.0.1. Gabe Cohan dari *Google* menegaskan bahwa android 4.0 secara teoritis cocok dengan perangkat android 2.3 ke atas dalam produksi masa itu. Android ICS 4.0 membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru berupa membuka kunci atau *password* dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari *e-mail* secara *offline*, dan berbagi fungsi dengan menggunakan NFC. *Source code* untuk Android 4.0 tersedia secara publik pada tanggal 14 November 2011. Ponsel pertama yang menggunakan sistem operasi ini adalah *Samsung Galaxy Nexus*.

11. Android Version 4.1 (*Jelly Bean*)

Android ini diluncurkan pada acara *Google I/O* lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya menambahkan *input keyboard*, desain baru fitur pencarian, UI yang baru dan fitur pencarian berupa *voice search* yang lebih tepat. Tak ketinggalan pula *Google Now* juga menjadi bagian yang diperbaharui. *Google Now* memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu lintas, ataupun hasil olahraga. Sistem operasi tipe ini muncul pertama kali pada produk tablet Asus, yakni *Google Nexus 7*.

2.4.3 Siklus Hidup Android

Siklus hidup aplikasi Android dikelola oleh sistem, berdasarkan kebutuhan *user* dan sumber daya yang tersedia. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika *user* sedang menjalankan sebuah *activity*, maka sistem akan memprioritaskan aplikasi tersebut. Sebaliknya, jika suatu *activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *activity* yang prioritas rendah akan ditutup. Berikut adalah siklus hidup *activity* Android pada gambar 2.6:



Gambar 2.6 Siklus Hidup Activity Android

1. *onCreate()*

Kondisi awal saat *Activity* baru diciptakan, biasanya dilakukan inisialisasi pada tahapan ini.

2. *onStart()*

Kondisi ketika *Activity* dimulai.

3. *OnResume()*

Saat *Activity* dibuka kembali, biasanya dieksekusi setelah *onPause()*

4. *onPause()*

Akan dipanggil ketika ada *Activity* lain yang terbuka.

5. *onStop()*

Kondisi saat *Activity* tidak ditampilkan dilayar (biasanya saat pengguna menekan tombol *Home*).

6. *onRestart()*

Kondisi saat *Activity* dibuka kembali oleh pengguna.

7. *onDestroy()*

Kondisi saat *Activity* dihancurkan pada memori.

8. *onSaveInstanceState()*

Android akan memanggil method ini untuk mengizinkan *Activity* menyimpan setiap contoh keadaan, seperti posisi kursor dalam sebuah *text field*.

9. *OnRestoreInstanceState()*

Dipanggil ketika *Activity* di inisialisasi ulang sejak keadaan sebelum disimpan dari *onSaveInstanceState()*

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin virtual pengolah sendiri, dengan melindungi penggunaan memori pada aplikasi. Selain itu juga Android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karena itu Android menjadi sangat sensitif dengan siklus hidup aplikasi dan komponen-komponennya.[9]

2.4.4 Kekurangan Android

1. Koneksi internet yang terus menerus, kebanyakan ponsel berbasis sistem ini memerlukan koneksi internet yang simultan atau terus menerus aktif. Koneksi internet GPRS selalu aktif setiap waktu, itu artinya anda harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan.

2. Iklan – Aplikasi di ponsel Android memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap aplikasi tersebut, akan selalu iklan yang terpampang, entah itu bagian atas atau bawah aplikasi.[9]

2.4.5 Android SDK(Software Development Kit)

Android SDK (Software Development Kit) merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API diperlukan untuk memulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java.[10]

2.4.6 Android Studio

Android *studio* adalah Lingkungan Pengembangan Terpadu – *Integrated Development Environment (IDE)* untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA*. Selain merupakan *editor* kode *IntelliJ* dan alat pengembang yang berdaya guna, Android *Studio* menawarkan fitur yang lebih banyak untuk meningkatkan produktivitas anda saat membuat aplikasi Android, misalnya :

1. Sistem versi berbasis *Gradle* yang fleksibel.

2. Emulator yang cepat dan seperti fitur baru.
3. Lingkunganyang menyatu untuk pengembangan bagi semua perangkat Android.
4. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
5. *Template* kode dan integrasi *GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
6. Alat pengujian dan kerangka kerja yang ekstensif.
7. Alat *Lint* untuk meningkatkan kerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*. [11]

2.5 Web Servis

Web service adalah aplikasi sekumpulan data (*database*), perangkat lunak (*software*) atau bagian dari perangkat lunak yang dapat diakses secara *remote* oleh berbagai piranti dengan sebuah perantara tertentu. Secara umum, *web service* dapat diidentifikasi dengan menggunakan URL seperti hanya *web* pada umumnya. Namun yang membedakan *web service* dengan *web* pada umumnya adalah interaksi yang diberikan oleh *web service*. Berbeda dengan URL *web* pada umumnya, URL *web service* hanya mengandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi.

Web service dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah *database* ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan *platform* apa sebuah data itu dikonsumsi. *Web service* mampu menunjang interoperabilitas. Sehingga *web service* mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada.

Menurut W3C *Web service Architecture Working Group*, pengertian *Web service* adalah sebuah sistem *software* yang di desain untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. *Interface service* dideskripsikan dengan format yang mampu di proses oleh mesin (khususnya WSDL). Sistem lain yang akan berinteraksi dengan *web service* lainnya hanya memerlukan SOAP (*Simple Object Access Protocol*), yang biasanya disampaikan dengan HTTP dan XML sehingga mempunyai korelasi dengan standar *Web* (*Web Service Architecture Working Group*, 2004).

Web pada umumnya digunakan untuk melakukan respon dan *request* yang dilakukan antara *client* dan *server*. Sebagai contoh, seorang pengguna layanan *web* tertentu mengetikkan alamat url *web* untuk membentuk sebuah *request*. *Request* akan sampai pada *server*, diolah kemudian disajikan dalam bentuk sebuah respon. Dengan singkat kata terjadilah hubungan *client-server* secara sederhana.

Sedangkan pada *web service*, hubungan antara *client* dan *server* tidak terjadi secara langsung. Hubungan antara *client* dan *server* dijumpai oleh *file webservice* dalam format tertentu. Sehingga akses terhadap *database* akan ditangani tidak secara langsung oleh *server*, melainkan melalui perantara yang disebut *web service*. Peran dari *web service* ini akan mempermudah distribusi sekaligus integrasi *database* yang tersebar di beberapa server sekaligus.

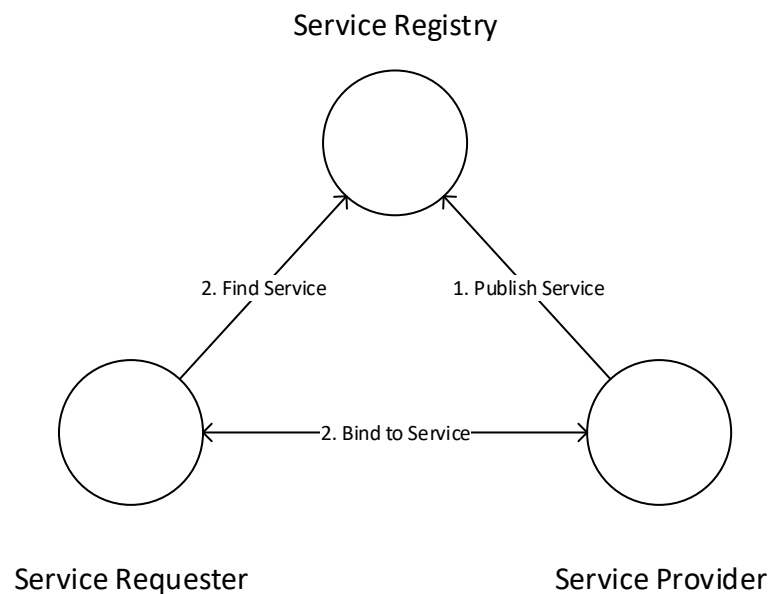
Web Service dibangun di atas beberapa teknologi *web* seperti XML, SOAP, WSDL, dan UDDI.

1. XML (*Ekstensible Markup Language*) digunakan untuk sumber daya data yang memiliki interoperabilitas tinggi dapat diakses dan didukung oleh berbagai jenis aplikasi dan teknologi)
2. SOAP (*Simple Object Access Protocol*), protokol ringan untuk XML sehingga dapat digunakan untuk menganalisa informasi dari *request* dan *response* pada *Web service* sebelum dikirimkan melalui jaringan.
3. WSDL (*Web Service Description Language*), sebuah bahasa berbetuk XML yang digunakan untuk mendeskripsikan kemampuan dari *Web service* sebagai kumpulan dari komunikasi-komunikasi yang saling bertransaksi pesan.
4. UDDI (*Universal Description, Discovery, and Integration*), sebuah direktori yang di distribusikan secara *web based* sehingga dapat mendaftarkan diri ke internet yang dapat dijelajahi.[12]

2.5.1 Arsitektur Web Service

Web service memiliki 3 entitas yaitu :

1. *Service provider*, merupakan pemilik *Web service* yang berfungsi menyediakan kumpulan operasi dari *Web service*.
2. *Service requestor*, merupakan aplikasi yang bertindak sebagai client dari *Web service* yang mencari dan memulai interaksi terhadap layanan yang disediakan.
3. *Service registry*, merupakan tempat dimana *Service provider* mempublikasikan layanannya. Pada arsitektur *web service*, *Service registry* bersifat optional. Teknologi *web service* memungkinkan kita dapat menghubungkan berbagai jenis *software* yang memiliki *platform* dan sistem operasi yang berbeda.[13]



Gambar 2.7 Arsitektur Web Service

2.6 Processor Hypertext Protocol (PHP)

PHP merupakan salah satu pemrograman server-side di antara beberapa pemrograman yang ada. Sejak di luncurkan, PHP membuat respon yang sangat baik dari kalangan pengembang aplikasi web kemudahannya untuk di pahami, serta sintaksnya yang mirip dengan bahasa C menjadikan pemrograman ini cepat dikenal dikalangan luas.

Tanpa mengurangi kemampuan-kemampuan yang sudah ada, PHP 5 hadir dengan fitur yang lebih kompleks, dan merupakan rilis terbaru yang di keluarkan oleh pengembangnya secara besar-besaran. Pengembang PHP merombak *bug* yang sering di jumpai dalam PHP 4, hal ini akan sangat jelas terasa ketika anda membuat aplikasi berbasis objek.

Dengan tujuan untuk lebih meningkatkan kinerja aplikasi, tanpa bermaksud membuang fitur yang ada, PHP 5 menambahkan fitur-fitur baru seperti dukungan terhadap MySQL. Dukungan ini di wujudkan dalam bentuk fungsi-fungsi lama dan di tulis ulang.[14]

2.6.1 JavaScript Object Notations (JSON)

JSON (*JavaScript Object Notations*) adalah format pertukaran data yang ringan , dan mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa perograman *JavaScript*, standar ECMA-262 edisi ke-3 – Desember 1999. JSON merupakan format teks yng tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* dan lain-lain. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data.

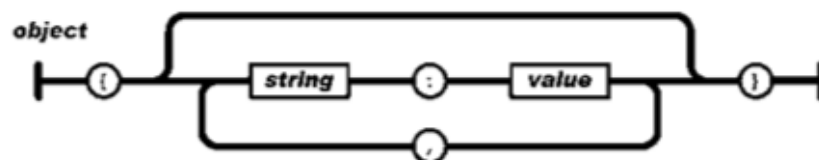
JSON terbuat dari dua struktur :

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel *hash* (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

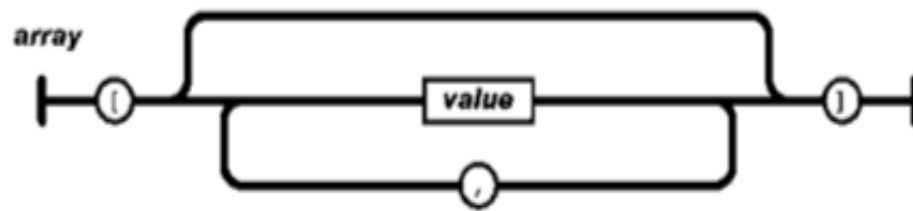
JSON menggunakan bentuk sebagai berikut :

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek di mulai dengan { (kurung kurawal buka) dan di akhiri dengan } (kurung kurawal tutup). Setiap nama di ikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



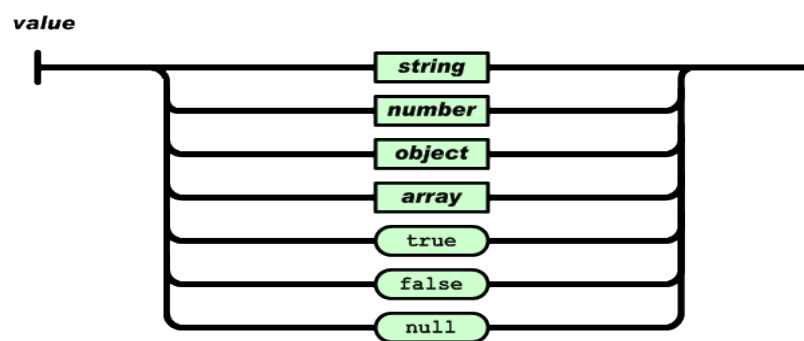
Gambar 2.8 Objek JSON

Larik adalah kumpulan nilai yang terurutkan. Larik di mulai dengan [(Kurung kurawal buka) dan di akhiri dengan] (kurung kurawal tutup). Setiap nilai di pisahkan oleh , (koma).



Gambar 2.9 Array JSON

Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau larik. Struktur-struktur tersebut dapat disusun bertingkat.



Gambar 2.10 Value JSON

String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* “\” untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. *String* sangat mirip dengan *string* C atau Java.

programmer menggunakan sistem *function*. Proses ini dikelola melalui sistem operasi. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi.

2.7 Database dan DBMS (Database Management System)

2.7.1 Pengertian Database

Database adalah sekumpulan koleksi data yang berhubungan secara logikal, dan sebuah deskripsi dari data tersebut, didesain untuk menemukan keperluan informasi pada sebuah perusahaan (Conolly, p15). *Database* merupakan tempat penyimpanan data yang besar yang dapat digunakan secara bersamaan oleh banyak pengguna dan berisi deskripsi dari data itu sendiri selain data operasional milik perusahaan.

Menurut Hoffer, Prescott, dan McFadden, *database* adalah sekumpulan organisasi data yang berelasi secara logikal. *Database* dapat memiliki banyak ukuran dan tingkat kompleksitas.

2.7.2 Pengertian DBMS (Database Management System)

Menurut James A. Hall, DBMS adalah sebuah sistem perangkat lunak khusus yang diprogram untuk mengetahui elemen data mana yang dapat diakses (didapatkan otorisasinya) oleh pemakai.

Menurut Connolly, DBMS atau *Database Management System* merupakan sebuah perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, mengambil data, dan mengontrol akses, kepada *database*. DBMS merupakan sebuah perangkat lunak yang menginterasikan *database* dengan aplikasi program pada pengguna.

Biasanya, DBMS menyediakan fasilitas sebagai berikut :

1. *Data Definition Language (DDL)* memperbolehkan pengguna untuk mendeskripsikan *database*, misalnya merinci tipe dan batasan data yang akan disimpan dalam *database*.

2. *Data Manipulation Language (DML)* memperbolehkan pengguna untuk memanipulasi data, msalnya memasukkan data, menghapus data, dan mendapatkan data dari *database*.
3. Menyediakan akses terkontrol ke *database*, misalnya *security system*, *integrity system*, *concurrency control system*, *recovery control system*, *user-accessible catalog*.

2.7.3 Komponen DBMS

Ada 5 komponen utama pada DBMS, yaitu :

1. *Hardware* (Perangkat Keras)

DBMS dan aplikasi membutuhkan perangkat keras untuk dapat berjalan. Perangkat kerasnya dapat berupa satu *personal computer*, satu *mainframe*, maupun jaringan yang terdiri dari banyak komputer. Perangkat keras yang dibutuhkan bergantung dari permintaan dari organisasi dan DBMS yang digunakan.

2. *Software* (Perangkat Lunak)

Komponen dari perangkat lunak terdiri dari perangkat lunak DBMS itu sendiri dan program aplikasi, bersamaan dengan sistem aplikasi, termasuk perangkat lunak jaringan jika DBMS digunakan melalui jaringan.

3. Data

Mungkin komponen yang terpenting pada DBMS, terutama dari sudut pandang pengguna, adalah data. Data berperan sebagai jembatan antara komponen mesin (*hardware dan software*) dan komponen manusia (prosedur dan manusia). *Database* berisi baik data, maupun meta data, yaitu tentang data. Struktur *database* disebut skema.

4. Prosedur

Prosedur menunjuk pada instruksi dan aturan yang mempengaruhi desain dan penggunaan dari *database*. Para pengguna sistem dan para *staff* yang mengatur dokumen prosedur *database* yang

dibutuhkan dan bagaimana cara menggunakan atau menjalankan sistem.

2.8 Algoritma Association Rule (Apriori)

Algoritma apriori (*association rule*) merupakan salah satu algoritma klasik data *mining*. Algoritma apriori digunakan agar komputer dapat mempelajari aturan asosiasi, mencari pola hubungan antar satu atau lebih *item* dalam suatu *dataset*.

Algoritma apriori banyak digunakan pada data transaksi atau biasa disebut *market basket*, misalnya sebuah swalayan memiliki *market basket*, dengan adanya algoritma apriori, pemilik swalayan dapat mengetahui pola pembelian seorang konsumen, jika seorang konsumen membeli item A, B, punya kemungkinan 50% dia akan membeli item C, pola ini sangat signifikan dengan adanya data transaksi selama ini.[16]



Gambar 2.13 Algoritma Association Rule

2.9 Object Oriented Analysis and Design

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memeriksa *requirement* (syarat atau keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD

adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem.[17]

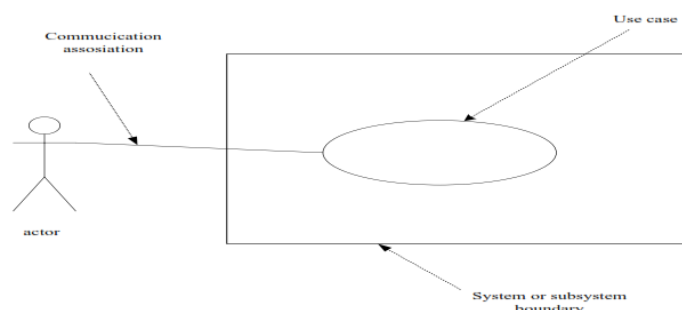
2.9.1 UML (Unfied Modeling Language)

Unfied Modeling Language (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh meta-model tunggal. Pemodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan *object-oriented* (OO). UML merupakan standar terbuka yang dikelola oleh *Open Management Group* (OMG) yang berada dibawah naungan perusahaan-perusahaan konsorsium terbuka. UML merupakan bahasa pemodelan yang terdiri banyak model diantaranya yaitu :

1. *Usecase Diagram*

Menurut Booch (2005), suatu *usecase diagram* menampilkan sekumpulan *usecase* dan aktor, dan hubungan diantara *usecase* dan aktor tersebut. *Usecase diagram* digunakan untuk *usecase* statik dari suatu sistem. *Usecase diagram* penting dalam mengatur dan memodelkan kelakuan dari suatu sistem.

Usecase menjelaskan apa yang dilakukan sistem (atau subsistem) tetapi tidak menspesifikasikan cara kerjanya. *Flow of event* digunakan untuk menspesifikasikan kelakuan dari *usecase*. *Flow of event* menjelaskan *usecase* dalam bentuk tulisan dengan sejelas-jelasnya, diantara bagaimana, kapan *usecase* dimulai dan berakhir, ketika *usecase* berinteraksi dengan aktor, objek apa yang digunakan, alur dasar dan alur alternatif.

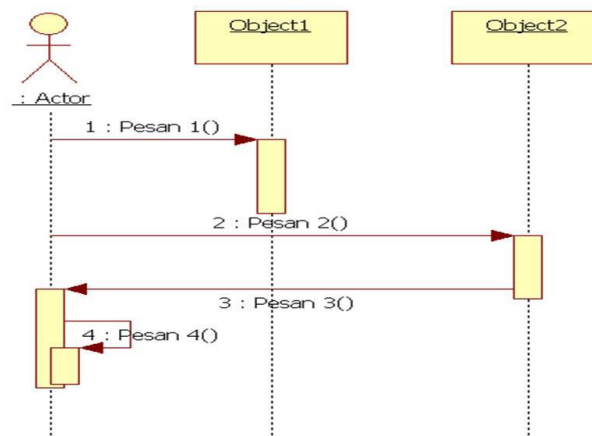


Gambar 2.14 Notasi Use Case Diagram

2. Sequence Diagram

Menurut Booch (2005), suatu *sequence diagram* adalah suatu diagram interaksi yang menekankan pada pengaturan waktu dari pesan-pesan. Diagram ini menampilkan sekumpulan peran dan pesan-pesan yang dikirim dan diterima oleh instansi yang memegang peranan tersebut.

Sequence diagram menangkap objek dan *class* yang terlibat dalam skenario dan urutan-urutan pesan yang ditukar antara objek diperlukan untuk melaksanakan fungsionalitas skenario. *Sequence diagram* berasosiasi dengan *usecase* selama proses pengembangan. Dalam *Unified Modeling Language (UML)*, objek dalam *sequence diagram* dengan segiempat yang berisi nama objek yang diberi garis bawah. Objek dapat diberi nama dengan tiga cara : (nama objek), (nama objek dan *class*) atau (hanya nama *class* (*anonymous object*)). Berikut notasi *sequence diagram* seperti terlihat di gambar 2.15 dibawah ini :



Gambar 2.15 Notasi Sequence Diagram

3. Class Diagram

Menurut Booch (2005), *class diagram* menunjukkan sekumpulan kelas, antarmuka, dan kerjasama serta hubungannya. *Class diagram* digunakan untuk memodelkan perancangan statik dari sistem. Biasanya meliputi pemodelan *vocabulary* dari sistem, pemodelan kerjasama, atau pemodelan skema. *Class diagram* dapat digunakan untuk membangun sistem yang dapat dieksekusi

melalui teknik *forward and reverse*, selain untuk penggambaran, penspesifikasian, dan pendokumentasian struktur model. *Class diagram* terdiri dari Nama Class, Atribut dan Operasi/Metode.

Tabel 2.1 Class Diagram

Nama Class
Atribut
Operasi/Metode

Atribut dan Operasi/metode dapat memiliki tiga sifat berikut :

1. *Public*, dapat dipanggil oleh *class* apa saja.
2. *Protected*, hanya dapat dipanggil atau diakses oleh *class* yang bersangkutan dan *class* turunannya.
3. *Private*, hanya dapat dipanggil oleh dirinya sendiri (tidak dapat diakses dari luar *class* yang bersangkutan).
4. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

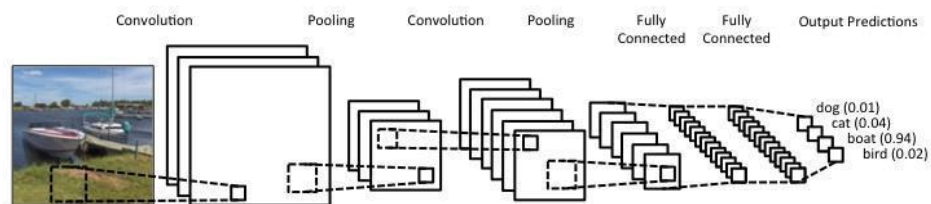
Activity diagram merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *Action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activitydiagram* tidak menggambarkan *behavior* internal sebuah sistem.

Dalam penelitian ini, *activity diagram* menjelaskan tata cara pengunjung mall Asia Plaza untuk menggunakan aplikasi dengan sebuah penggambaran proses yang terjadi di suatu aplikasi.[18]

2.10 Clarifai

Clarifai adalah perusahaan kecerdasan buatan yang unggul dalam pengenalan visual (Visual Recognition). Clarifai didirikan pada tahun 2013 oleh Matthew Zeiler, seorang ahli terkemuka dalam Machine Learning, Clarifai telah menjadi pemimpin pasar sejak memenangkan lima tempat teratas dalam klasifikasi citra pada kompetisi ImageNet 2013 The ‘Food Model’. [19]

Clarifai adalah alat yang dapat mengidentifikasi atau mengenali gambar atau video yang dimasukkan sebagai inputan yang dapat memberikan hasil berupa prediksi tentang apa yang ada di dalam gambar atau video berupa besaran probabilitas kemungkinan nya. [20]



Gambar 2.16 Skema Proses Clarifai API

