BAB 2
LANDASAN TEORI

2.1. Profil Dinas

Kedudukan dan struktur Dinas Bina Marga dan Pengairan telah diatur dalam Peraturan Daerah Kota Bandung No.13 Tahun 2007 tentang pembentukan dan susunan Organisasi Dinas Daerah Kota Bandung.

Dinas Bina Marga dan Pengairan Kota Bandung mempunyai tugas pokok untuk melaksanakan kewenangan daerah dibidang pekerjaan umum lingkup kebinamargaan dan sumberdaya air.

2.1.1. Sejarah Dinas

Bina Marga dan Pengairan merupakan bagian dari Pekerjaan Umum. Istilah "Pekerjaan umum" adalah terjemahan dari istilah bahasa belanda "Openbare Werken" yang pada jaman belanda disebut " Waterstaat werken". Dilingkungan pusat Pemerintahan dibina oleh Dep.Van Verkeer &Waterstaat (Dep V&W) Yang sebelumnya terdiri dari 2 Dept. Van Guovernements Bedri Jven dan Dept. Van Burgerlijke Openbare Werken. Dep. V dan W dikepalai oleh seorang Direktur, yang membawahi beberapa Afdelingen dan Diensten sesuai dengan tugas / wewenang Departemen ini. Yang meliputi bidang PU (openbare werken) termasuk afdeling Waterstaat, dengan onder afdelinger. : 1. Lands gebouwen, 2.Wegen, 3. Irrigate & Assainering, 4.Water Kracht, 5. Constructie bureau (untuk jembatan).

Disamping yang tersebut di atas, yang meliputi bidang PU (Openbare Werken) juga afd. Havenwezen (Pelabuhan), afd. Electricitswezen (kelistrikan) dan afd. Luchtvaart (Penerbangan sipil). Organisasi PU (Openbare werken) Di daerah-daerah adalah sebagai berikut : Jawa Barat, Jawa Tengah, dan Jawa Timur urusan Waterstaat/openbare werken diserahkan pada pemerintahan Provinsi yang disebut : Provinciale Waterstaatdienst" dan dikepalai oleh seorang Hoofd Provinciale Waterstaatdienst (H.P.W).

Ketika Bandung ingin mengembalikan kekuasaan pemearintahan di Hindia Belanda sebelum perang, datang mengikuti tentara sekutu masuk ke Indonesia akibat dari keinginan pemerintahan belanda ini, terjadilah pertentangan fisik dengan pemuda Indonesia yang ingin mempertahankan tanah air berikut gedung-gedung yang telah didudukinya, antara lain "Gedung Sate" yang telah menjadi Gedung Departemen Pekerjaan Umum pada waktu itu. (Peristiwa bersejarah itu dikenal dengan peristiwa "3 Desember 1945"). Pasa waktu revolusi fisik dari tahun 1945 s/d 1949, Pemerintah Pusat RI di Jakarta terpaksa mengungsi ke Purworejo untuk selanjutnya ke yogjakarta, begitu juga kementrian PU. Sesudah pemerintahan belanda tahun 1949 mengakui kemerdekaan Republik Indonesia maka pusat pemerintahan RI di Yogyakarta, di pindahkan ke Jakarta.

Dalam masa prolog G 30 S /PKI terjadilah dalam sejarah Pemerintahan RI suatu cabinet yang besar disebut dengan nama Kabinet DwiKora Atau Kabinet 100 Menteri, dimana pada masa itu ikut mengalami perubahan orgasisasi menjadi 5 Dept. dibawah kompartemen PUT ketika membawahi, antara lain :

1. Departemen Listrik dan Ketenagakerjaan
2. Departemen Bina Marga
3. Departemen Cipta Karya Konstruksi
4. Departemen Pengairan Dasar
5. Departemen Jalan Raya Sumatera

2.1.2. Visi dan Misi

Berikut merupakan visi dan misi dari Dinas Bina Marga dan Pengairan Kota Bandung:

Visi

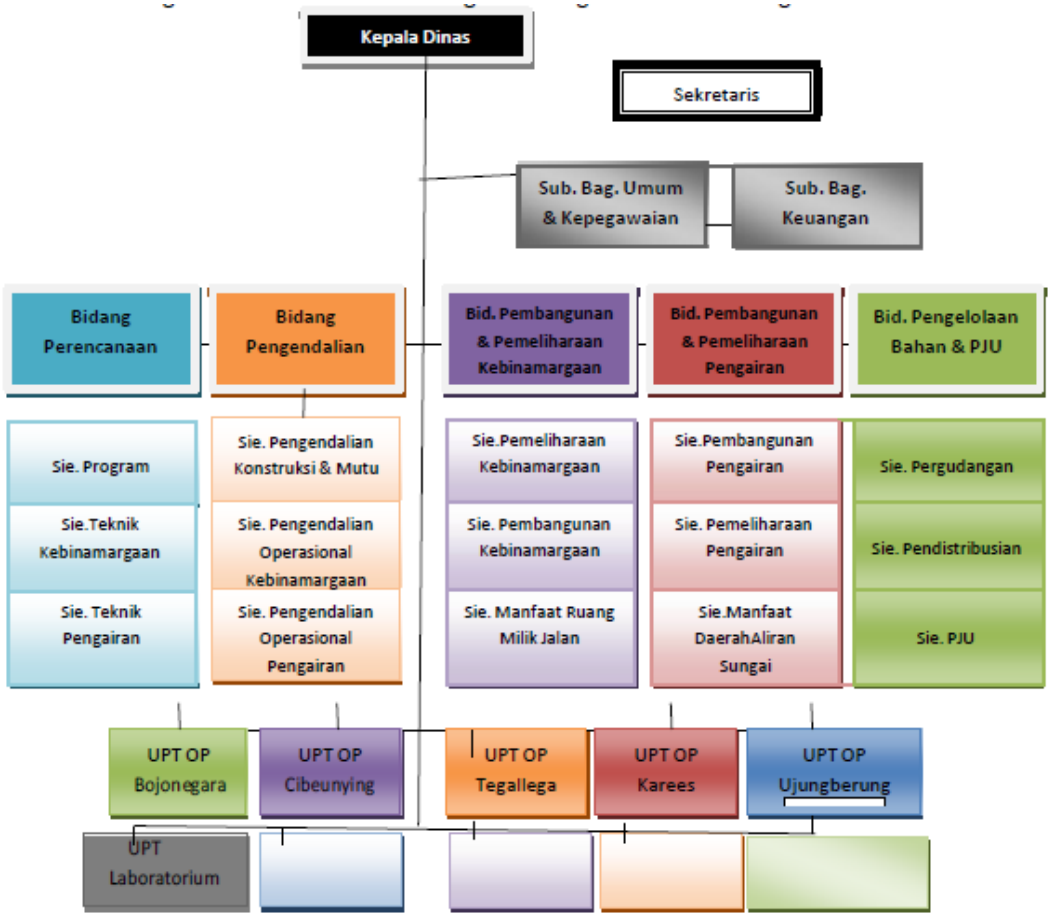
"Terwujudnya Infrastruktur Kebinamargaan, Pengairan, Dan Penerangan Jalan Umum Yang Unggul, Nyaman dan Andal Untuk Kota Bandung Yang Bermartabat".

Misi

1. Meningkatkan aksesibilitas antar wilayah kota dan mobilitas warga kota;

2. Membangun jalan, trotoar dan drainase jalan kota dengan struktur yang berkualitas dan menjamin keselamatan pengguna jalan;
3. Menata sungai sebagai sistem drainase alami pengendali banjir dan bagian muka pembangunan kota *(river-front city*);
4. Mengembangkan sistem penerangan jalan umum kota yang hemat energi, ramah lingkungan dan berestetika;
5. Meningkatkan akuntabilitas dan kinerja pelayanan SKPD Dinas Bina Marga dan Pengairan.

2.1.3. Struktur Organisasi



Sumber : Dinas Bina Marga dan Pengairan Kota Bandung

Gambar 2.1 Struktur Organisasi

2.2. Landasan Teori

Landasan terori bertujuan memberikan gambaran dari teori yang terkait dalama pembangunan aplikasi. Landasan teori yang dibahas yaiitu pengertian *Mobile*, Android, peta, GPS, *Geotagging*, *Database*, Metode yang digunakan dan bahasa pemrograman yang digunakan dalam pembangunan aplikasi.

2.2.1. Mobile

Menurut Turban, *Mobile application* juga biasa disebut dengan *mobile apps*, yaitu istilah yang digunakan untuk medeskripsikan aplikasi internet yang berjalan pada *smartphone* atau piranti *mobile* lainnya. Aplikasi *mobile* biasanya membantu para penggunanya untuk terkoneksi dengan layanan internet yang biasa diakses pada PC atau mempermudah mereka untuk menggunakan aplikasi internet pada piranti yang bisa dibawa [2].

2.2.2. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh *Android Inc*, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh *Google Inc*. Untuk pengembangannya, dibentuklah Open Handset Alliance (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [3].

2.2.2.1. Android Life Cycle

Aplikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang

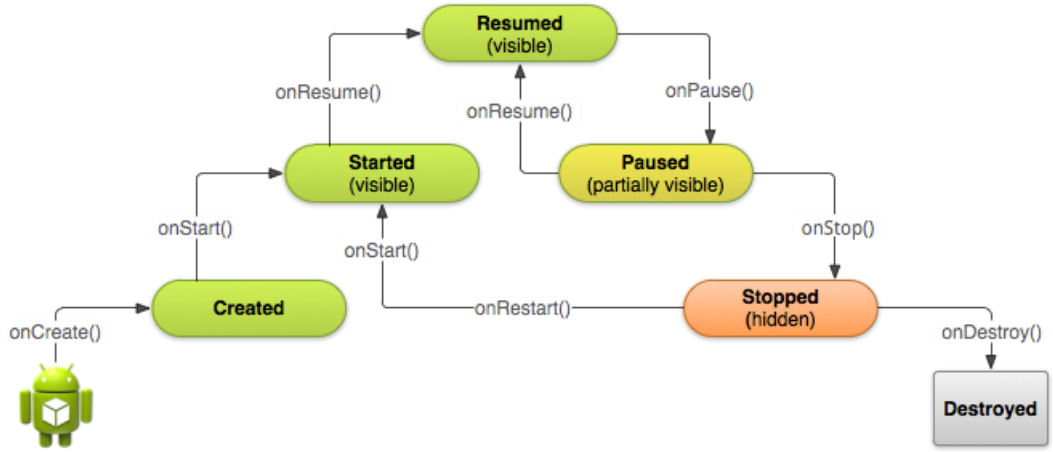
berbeda seperti ditunjukkan pada , klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan [4].

Tabel 2.1 Komponen Apliksi Android

Functionality	Java Base Class	Examples
Focused thing a user can do	Activity	Edit a note, play a game
Background process	Service	Play music, update weather icon
Reveice messages	Broadcast Receiver	Trigger alarm upon event
Functionality	**Java Base Class**	**Examples**
Store and retrive data	Content Provider	Open a phone contact

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan specified permission sebelum digunakan seperti komponen Service, BroadcastReceiver. ContentProvider [4].

Android memiliki paradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi main(), sistem android menjalankan kode dalam method Activity dengan menerapkan metode callback tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi android memiliki siklus hidup yang berbeda dengan aplikasi desktop atau web. Hal ini dikarenakan aplikasi mobile memiliki tingkat interupsi proses yang lumayan tinggi seperti ketika handling panggilan masuk aplikasi diharuskan menghentikan proses sementara. Penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai pengguna [5].



Sumber : http://developer.android.com/images/training/basic/basiclifecycle.png

Gambar 2.2 Android Life Cycle

Terdapat beberapa state dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gambar II.4 Siklus Hidup Android [5], akan tetapi hanya beberapa dari state tersebut yang menjadi statis diantaranya :

1. Resumed

Resumed terjadi ketika aplikasi berjalan setelah state *paused* . State ini akan menjalankan perintah program yang ditulis pada method onResume() [5].

2. Paused

Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karena terdapat proses yang memiliki prioritas lebih tinggi seperti panggilan telepon. Aplikasi tidak dapat menjalankan perintah apapun ataupun menampilkan apapun dalam state ini [5].

3. Stopped

Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan service dibackground [5].

State lain seperti Created dan Started bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode *life cycle callback*

berikutnya. Artinya, setelah sistem OnCreate() dipanggil, dengan cepat sistem akan memanggil method OnStart(), krmudian diikuti oleh onResume().

2.2.3. Geotagging

GPS *Photo Tagging* juga dikenal sebagai *geotagging*, merupakan proses penambahan informasi posisi data GPS (*Latitude*, *Longitude*, *Altitude*) dalam sebuah foto digital. Ponsel-ponsel berkamera yang memiliki GPS *receiver* internal umumnya memiliki fitur ini. Mekanisme GPS *PhotoTagging* adalah pada saat sebuah foto diambil menggunakan kamera (digital atau ponsel) yang memiliki fitur *geotagging*, kamera atau ponsel tersebut mencatat lebih banyak informasi/data dibandingkan dengan sebuah foto yang diambil dengan kamera biasa. Informasi tersebut termasuk waktu dan data ketika sebuah foto diambil, orientasi dari kamera (*portrait* atau *landscape*), apakah menggunakan lampu flash dan detil kamera lainnya yang digunakan seperti Apertur, Exposure, dan Local Length. Semua data ini disimpan pada suatu tempat yang disebut EXIF Headers.

Ada 2 pilihan untuk melakukan geotagging foto; menangkap informasi GPS pada saat foto itu diambil atau melampirkan foto itu ke peta setelah foto diambil.

EXIF (*Exchangeable Image File Format*) headers berisi petunjuk foto dengan data yang dapat dibaca oleh aplikasi manajemen foto atau sebuah situs foto tertentu. Selain itu, EXIF *Headers* juga menyediakan sebuah ruang untuk mengisi koordinat *Longitude*, *Latitude*, dan *Altitude* [6].

2.2.4. Global Positioning System (GPS)

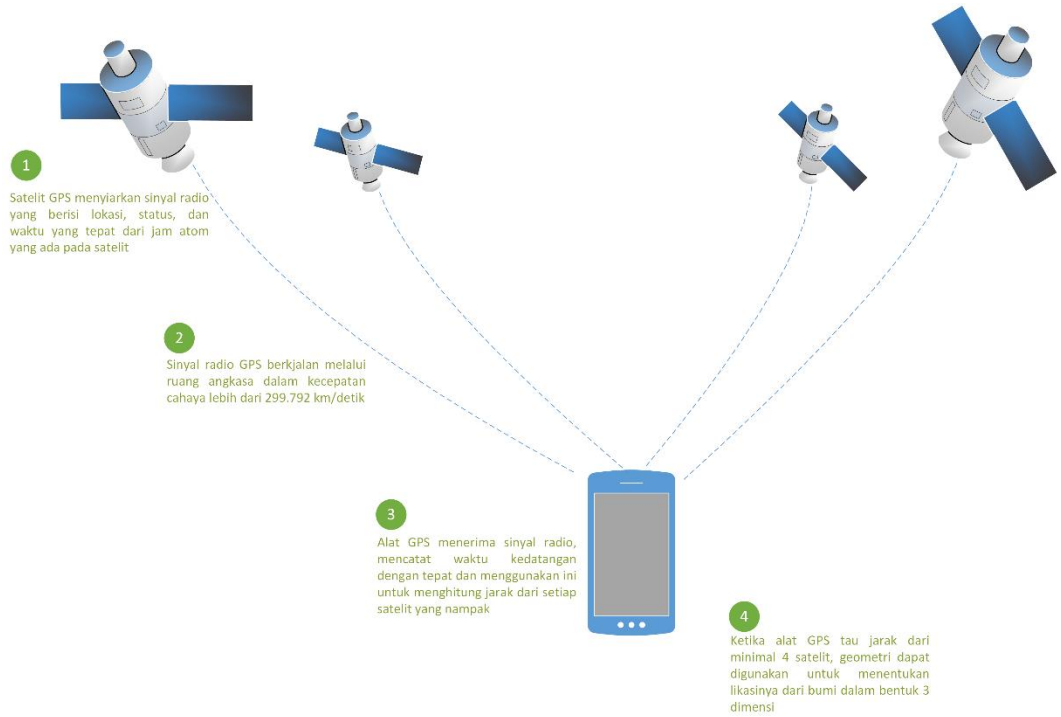
Global Positioning System adalah sistem navigasi berbasis satelit terdiri dari jaringan 24 satelit ditempatkan ke orbit oleh Departemen Pertahanan Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978. Layanan GPS dahulu hanya dipergunakan untuk keperluan militer namun mulai terbuka untuk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

penguat kecil pada masing-masing satelit agar dapat mengorbit tepat pada tempatnya.

Satelit-satelit GPS harus selalu berada pada posisi orbit yang tepat untuk menjaga akurasi data yang dikirim ke GPS *receiver*, sehingga harus selalu dipelihara agar posisinya tepat. Posisi satelit-satelit tersebut selalu dipantau oleh stasiun pengendali. Stasiun-stasiun pengendali di bumi ada di Hawaii, Ascension Islan, Diego Garcia, Kwajalein dan Colorado Spring. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS *receiver* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. GPS *receiver* mengambil informasi tersebut dan melakukan perhitungan triangulation untuk menentukan lokasi pengguna dengan tepat. GPS receiver membandingkan waktu sinyal dikirim dengan waktu sinyal tersebut diterima untuk mengetahui jarak satelit. Dengan mengetahui jarak tersebut, GPS receiver dapat melakukan perhitungan dan menentukan posisi pengguna dan menampilkan dalam peta elektronik. Setelah menentukan posisi pengguna, selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan masih banyak lagi.

Satelit GPS sangat presisi dalam mengirim informasi karena satelit tersebut menggunakan jam atom. Jam atom yang ada pada satelit merupakan partikel atom yang diisolasi, sehingga dapat menghasilkan jam yang akurat dibandingkan dengan jam biasa. Keistimewaan GPS adalah mampu bekerja dalam berbagai kondisi cuaca, siang atau malam. Keakuratan sebuah perangkat GPS bisa mencapai 15 meter, bahkan model terbaru yang dilengkapi teknologi Wide Area Augmentation System (WAAS) keakuratannya sampai 3 meter. Karena GPS bekerja mengandalkan satelit, maka penggunaannya disarankan di tempat terbuka. Penggunaan di dalam ruangan, atau di tempat yang menghalangi arah satelit (di angkasa), maka GPS tidak akan bekerja secara akurat dan maksimal. Perhitungan waktu yang akurat sangat menentukan akurasi perhitungan untuk menentukan informasi lokasi. Semakin banyak sinyal satelit yang dapat diterima maka akan semakin presisi data posisi yang dihasilkan. Selain itu, ketinggian juga mempengaruhi proses kerja GPS, karena semakin tinggi maka semakin bersih atmosfer, sehingga gangguan semakin sedikit [6].

Berikut adalah gambar 2.3 yang menjelaskan cara kerja GPS.



Sumber : http://www.gps.gov

Gambar 2.3 Cara Kerja GPS

2.2.5. JavaScript Object Notation (JSON)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [7].

JSON terbuat dari dua struktur [7]:

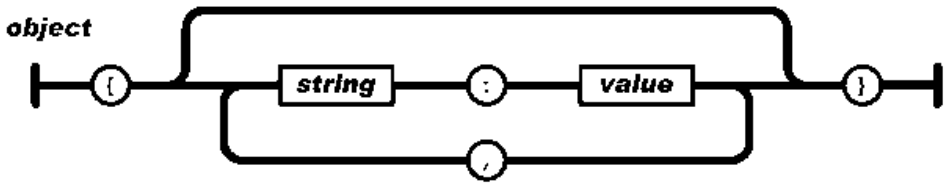
1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau associative array.

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*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini [7].

JSON menggunakan bentuk sebagai berikut:

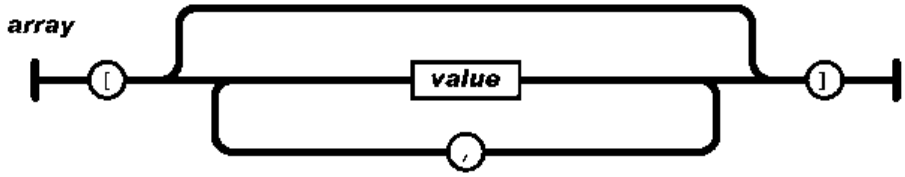
Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



Sumber : http://json.org

Gambar 2.4 Objek JSON

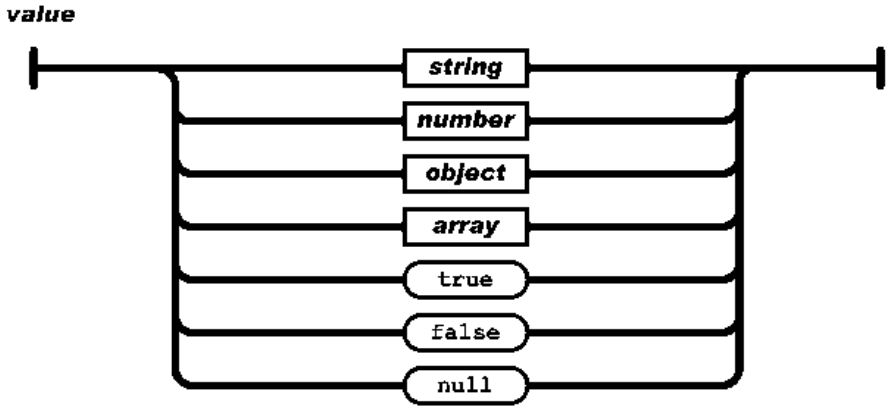
Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).



Sumber : http://json.org

Gambar 2.5 Array JSON

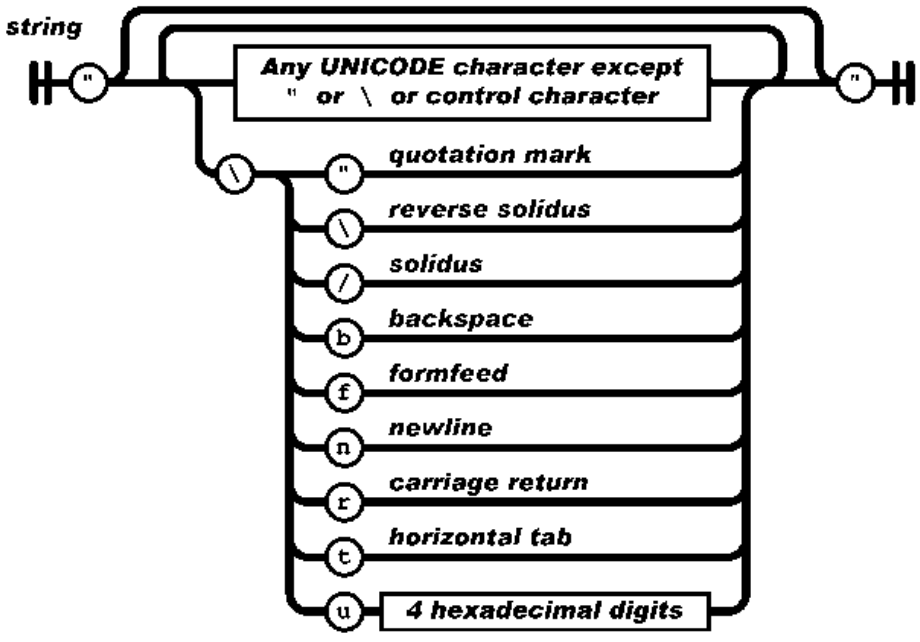
Nilai (value) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau true atau false atau null, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



Sumber : http://json.org

Gambar 2.6 Value JSON

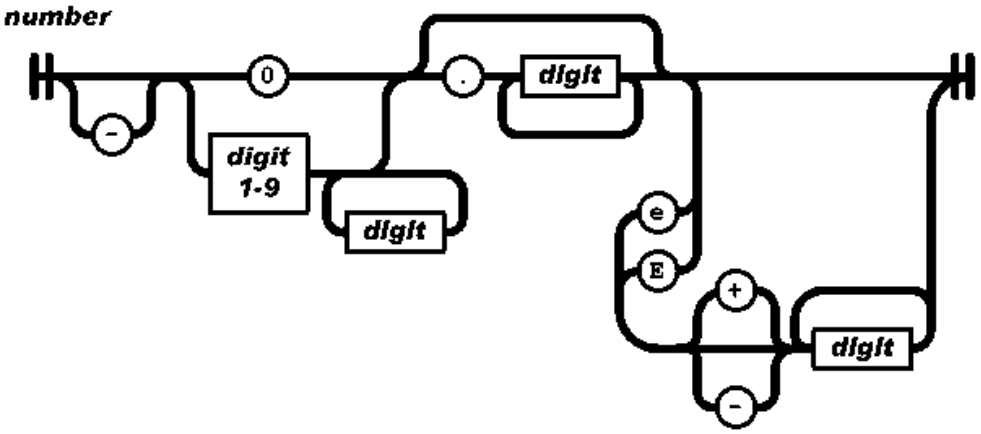
String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan backslash escapes "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.



Sumber : http://json.org

Gambar 2.7 String JSON

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Sumber : http://json.org

Gambar 2.8 Number JSON

Spasi kosong (whitespace) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detil encoding yang secara lengkap dipaparkan oleh bahasa pemprograman yang bersangkutan.

2.2.6. Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Java dikembangkan pada bulan Agustus 1991, dengan nama semula Oak. Pada Januari 1995, karena nama Oak dianggap kurang komersial, maka diganti menjadi Java. Pada Desember 1998, Sun memperkenalkan nama "Java 2" (J2) sebagai generasi kedua dari java platform. Konvensi nama baru ini diterapkan untuk semua edisi Java yaitu Standard Edition (J2SE), Enterprise Edition (J2EE), dan Micro Edition (J2ME) [8].

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

1. Standard Edition (J2SE) : J2SE merupakan inti dari bahasa pemrograman Java. J2SE didesain untuk jalan pada komputer desktop dan komputer workstations.
2. Enterprise Edition (J2EE): Dengan built-in mendukung untuk servlets. JSP, dan XML, edisi ini ditujukan untuk aplikasi berbasis server.

3. Micro Edition (J2ME) : Didesain untuk piranti dengan memori terbatas, layar display terbatas dan power pemrosesan yang juga terbatas.

2.2.6.1. Kelebihan Java

Adapun beberapa kelebihan dari bahasa pemrograman java adalah sebagai berikut [9]:

1. Sederhana dan Ampuh

Java dirancang untuk mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++ akan mudah sekali untuk berpindah ke Java. Pemakai dapat belajar membuat program dengan Java secara cepat jika telah memahami konsep dasar pemrograman berorientasi objek. Java tidak memiliki hal-hal yang mengejutkan dan aneh. Java memberi anda kemampuan untuk menuangkan semua ide, karena bahasa pemrograman ini bukan merupakan scripting language (bahasa naskah) yang menghilangkan kemampuan kita untuk berinovasi, tetapi dengan cara berorientasi objek yang mudah dan jelas.

2. Aman

Java dirancang sebagai bahasa pemrograman yang handal dan aman. Aplikasi-aplikasi yang dibangun dengan bahasa Java sangat handal dengan manajemen memori yang bagus. Aplikasi Java juga dikenal sangat secure, yaitu kasus-kasus seperti buffer everflow yang umumnya menjadi lubang keamanan aplikasi-aplikasi berbasis C/C++ tidak terjadi di Java, karena pengaturan securitynya yang bagus.

3. Berorientasi Objek

Paradigma pemrograman berorientasi objek merupakan paradigma pemrograman masa depan. Java merupakan bahasa pemrograman berorientasi objek. Java bukan turunan langsung dari bahasa pemrograman manapun, juga sama sekali tidak kompetibel dengan semuanya. Java memiliki keseimbangan, Java secara cepat jika telah memahami konsep dasar pemrograman berorientasi objek. Java tidak memiliki hal-hal yang mengejutkan dan aneh. Java memberi anda kemampuan untuk

menuangkan semua ide, karena bahasa pemrograman menyediakan mekanisme peng-class-an sederhana, dengan model antar muka dinamik yang intuitif hanya jika diperlukan.

4. Kokoh

Java membatasi anda dari beberapa hal kunci supaya anda dapat menemukan kesalahan lebih cepat saat mengembangkan program. Java langsung memeriksa program saat anda menuliskannya, dan sekali lagi ketika program di jalankan. Karena Java adalah bahasa yang sangat ketat dalam hal tipe data dan deklarasi, banyak kesalahan umum terjadi saat kompilasi. Hal ini akan lebih menghemat waktu jika dibandingkan dengan keharusan menjalankan program terlebih dahulu dan memeriksa semua bagian program untuk melihat ketidakcocokan dinamis selama program berjalan. Ini adalah contoh di mana Java lebih luwes dan kokoh dari beberapa bahasa lain, tetapi dengan imbalan yang layak untuk kelebihan itu.

5. Interaktif

Java memiliki beberapa kemampuan yang memungkinkan program melakukan beberapa hal pada saat bersamaan, tanpa harus kesulitan menangani proses yang akan terjadi selanjutnya. Jalinan program-program Java yang mudah digunakan memungkinkan kita untuk memikirkan pembuatan perilaku khusus, tanpa harus mengintegrasikan perilaku tersebut dengan model pemrograman global yang mengatur perulangan kejadian.

6. Netral Terhadap Berbagai Arsitektur

Java telah mengambil beberapa keputusan yang sulit dalam pembuatan bahasa Java dan bagaimana program dijalankan, jadi anda dapat sepenuhnya percaya “tulis sekali, jalan di mana saja, kapan saja, selamanya”.

7. Terintrepetasi dan Berkinerja Tinggi

Java dilengkapi keajaiban lintas platform yang luar biasa dengan kompilasi ke dalam representasi langsung yang disebut kode-byte Java (Java byte-code), yang dapat diterjemahkan oleh sistem manapun yang memilki program Java didalamnya. Java, bagaimanapun dirancang untuk tetap berkinerja baik pada CPU yang tidak terlalu kuat. Walaupun Java merupakan bahasa terinterpretasi, kodekode Java telah dirancang dengan hati-hati sehingga mudah diterjemahkan ke dalam bahasa asli suatu mesin untuk menghasilkan kinerja yang tinggi.

Sistem program Java yang melakukan optimasi tepat waktu tersebut tidak kehilangan keuntungan dari program yang netral terhadap platform. "lintas platform berkinerja tinggi" bukan sekedar omong-kosong. Dalam aplikasi Java (*.class) merupakan Java bytecode yang berjalan di atas jvm (Java Virtual Machine), yang kemudian jvm-lah yang akan menginterpresentasikan kode-kode tersebut ke kode native atau kode mesin dari arsitektur yang bersangkutan. Hal sangat menarik karena urusan arsitektur mesin bukan jadi masalah bagi programmer tapi menjadi urusan kompiler pada bahasa pemrograman Java.

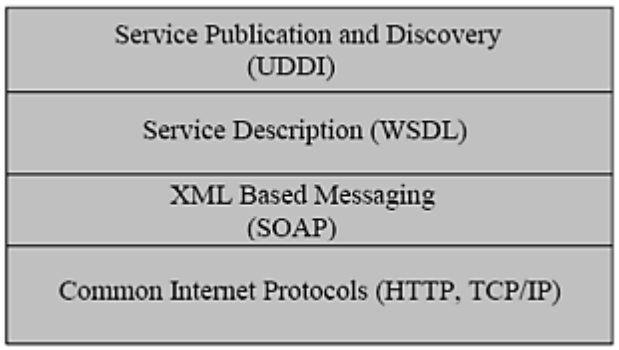
2.2.7. Web services

Web service merupakan suatu komponen software yang merupakan selfcontaining, aplikasi modular self-describing yang dapat dipublikasikan, dialokasikan, dan dilaksanakan pada web. Web service adalah teknologi yang mengubah kemampuan internet dengan menambahkan kemampuan transactional web, yaitu kemampuan web untuk saling berkomunikasi dengan pola program-to-program (P2P). Fokus web selama ini didominasi oleh komunikasi program-to-user dengan interaksi business-to-consumer (B2C), sedangkan transactional web akan didominasi oleh program-to-program dengan interaksi business-to-business.

Gambar 2.3 merupakan blok bangunan web service yang mana menyediakan fasilitas komunikasi jarak jauh antara dua aplikasi yang merupakan layer arsitektur web service.

1. Layer 1 : Protokol internet standar yang digunakan sebagai sarana transportasi adalah HTTP dan TCP/IP.

2. Layer 2 : Simple Object Access Protocol (SOAP) berbasiskan XML dan digunakan untuk pertukaran informasi antar sekelompok layanan.
3. Layer 3 : Web service Definition Language (WSDL) digunakan untuk mendiskripsikan attribute layanan.
4. Layer 4 : Universal Description, Discovery and Integration, yang mana merupakan direktori pusat untuk deskripsi layanan.



Gambar 2.9 Blok Bangunan *Web Service*

2.2.8. Processor Hypertext Protocol (PHP)

PHP adalah bahasa server-side –scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan server-side-scripting maka sintaks dan perintah-perintah PHP akan diesksekusi diserver kemudian hasilnya akan dikirimkan ke browser dengan format HTML.

Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web [9].

2.2.9. MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya. MySQL dikembangkan oleh perusahaan

swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya Tcx merupakan perusahaan pengembang software dan konsultan database, dan saat ini MySQL sudah diambil alih oleh Oracle Corp.

Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakan, kinerja query cepat, dan mencukupi untuk kebutuhan database perusahaan-perusahaan yang berskala kecil sampai menengah, MySQL juga bersifat open source (tidak berbayar). MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pembangun aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman script PHP [9].

2.2.10. Unified Modelling Language (UML)

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

UML merupakan standar yang relatif terbuka yang dikontrol oleh Object Management Group (OMG), sebuah konsorsium terbukayang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar CORBA (Common Object Request Broker Architecture) [10].

2.2.10.1. Class Diagram

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antaranya. Class diagram juga menunjukan properti dan operasi sebuah class dan batasan-batasan yang terdapat dalam

hubungan-hubungan objek tersebut. Kotak-kotak yang terdapat di dalam diagram merupakan class, yang dibagi menjadi tiga bagian : nama class (cetak tebal), atributnya dan operasinya [10].

1. Properti

   Properti mewakili fitur-fitur struktural dari sebuah class. Properti merupakan sebuah konsep tunggal, tetapi tampak seperti dua notasi yang sedikit berbeda: atribut dan asosiasi.

2. Atribut

   Notasi atribut mendeskripsikan properti dengan sebaris teks di dalam kotak class tersebut.

3. Operasi

   Operasi merupakan suatu kegiatan yang dimengerti sebuah class untuk dilakukan. Operasi secara jelas berhubungan dengan metode dalam sebuah class.

2.2.10.2. Use Case Diagram

Use case diagram menggambarkan fungsional yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat oleh sistem, dan bukan “bagaimana”. Sebuah use case mempresentasikan antara aktor dengan sistem.

2.2.10.3. Sequence Diagram

Sebuah sequence diagram secara khusus menjabarkan *behaviour* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam use case.

2.2.10.4. Activity Diagram

Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. dalam beberapa hal, diagram ini seperti sebuah diagram alir, tetapi perbedaan prinsip antar diagram ini dan notasi diagram alir adalah diagram ini mendukung *behaviour* paralel.

2.2.11. Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas (entity) serta hubungan (relationship) antar entitas-entitas tersebut dengan menggunakan beberapa notasi [10]. Komponen-komponen pembentuk ERD dapat di lihat pada Tabel 2.4 Komponen-komponen ERD [10].