BAB 2
LANDASAN TEORI

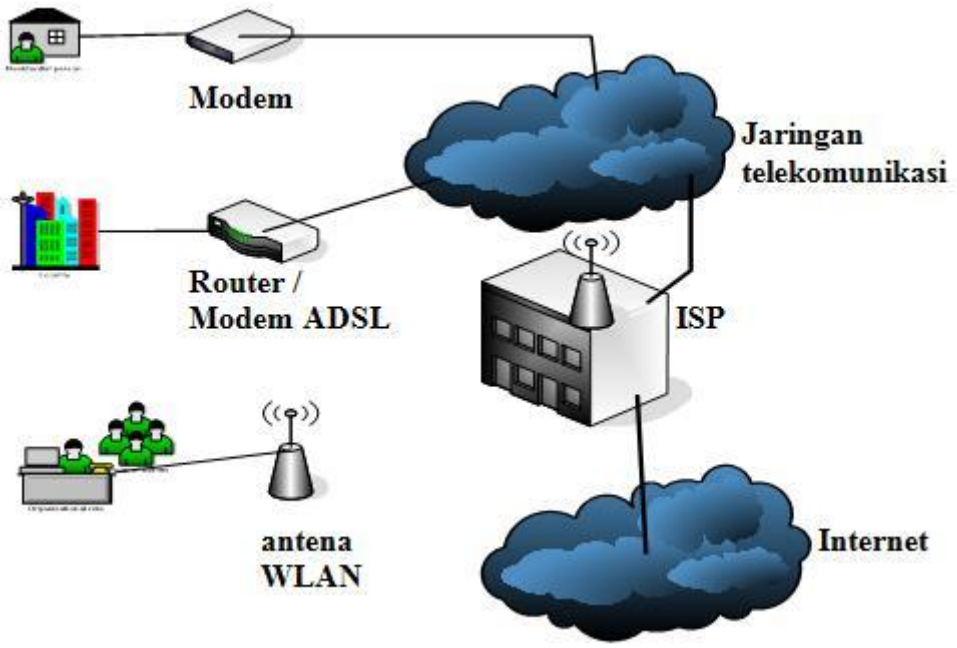
2.1 Jaringan

Internet adalah jaringan komputer yang bisa dikategorikan sebagai WAN, menghubungkan berjuta komputer diseluruh dunia, tanpa batas negara, dimana setiap orang yang memiliki komputer dapat bergabung ke dalam jaringan ini hanya dengan melakukan koneksi ke penyedia layanan internet (*Internet Service Provider* / ISP) seperti Telkom Speedy, atau IndosatNet. *Internet* dapat diterjemahkan sebagai *international networking* (jaringan internasional), karena menghubungkan komputer secara internasional, atau sebagai *internetworking* (jaringan antar jaringan) karena menghubungkan berjuta jaringan diseluruh dunia.

Internet dimulai ketika Departemen Pertahanan Amerika Serikat (*Department of Defense USA*) membangun sebuah jaringan komputer di tahun 1969, yang diberi nama ARPANET (*Advanced Research Project Agency NETwork*) dengan tujuan untuk menghubungkan beberapa komputer yang berada dibeberapa universitas melakukan riset militer, terutama untuk membangun jaringan komunikasi komputer yang mampu bertahan terhadap serangan nuklir. Jaringan ini berkembang terus, semakin banyak komputer yang terlibat, dan riset disisi pengembangan perangkat lunak juga berkembang. Pada bulan Mei tahun 1974, Vinton G.Cerf dari Stanford University dan Robert E.Kahn dari Departemen Pertahanan USA, mempublikasi sebuah paper di *IEEE Transaction on Communication* berjudul "*A Protocol for Packet Network Intercommunication*", konsep ini kemudian populer sebagai protokol TCP/IP, ketika ARPANET meng-adopsi protokol menjadi protokol standard untuk ARPANET pada tahun 1983.

Badan usaha komersil kemudian menyediakan layanan akses dengan menyediakan koneksi dari komputer pengguna ke internet, dan badan ini disebut sebagai penyedia akses internet atau ISP. Beberapa ISP terkenal di dunia adalah America On Line (AOL), Australia OnLine, CompuServe, GEnie, dan Prodigy. Di Indonesia ada TelkomNet, IndosatNet, Wasantara Net, InterNux, dan

sebagainya. ISP menyediakan koneksi dial-up melalui modem-telepon, koneksi wireless melalui antena WLAN, atau koneksi ADSL melalui telepon. Protokol koneksi yang digunakan adalah SLIP (*Serial Line Interface Protocol*) atau PPP (*Point-to-Point Protocol*), dimana koneksi SLIP biasanya lebih lambat dari PPP.



Gambar 2. 1 Koneksi Internet

Secara logis jaringan internet dibagi kedalam beberapa domain, yang menurut standar IPv4 (*Internet Protocol version 4*) di-identifikasi melalui nomer IP 32 bit atau 4 angka biner yang dipisahkan dengan titik (seperti 192.168.10.25).

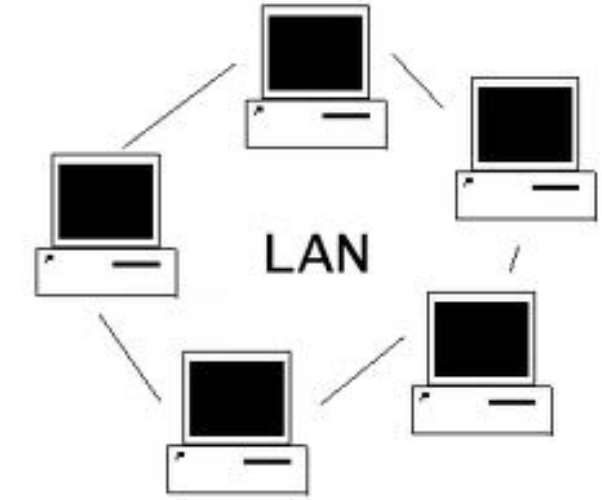
Jenis-jenis jaringan

a. Jaringan berdasarkan area jaringan

   1. *Local Area Network* (LAN)

      LAN adalah jaringan komputer yang jaringannya hanya mencakup wilayah kecil; seperti jaringan komputer kampus, gedung, kantor, dalam rumah, sekolah atau yang lebih kecil. Saat ini, kebanyakan LAN berbasis pada teknologi IEEE 802.3 Ethernet menggunakan perangkat *switch*, yang

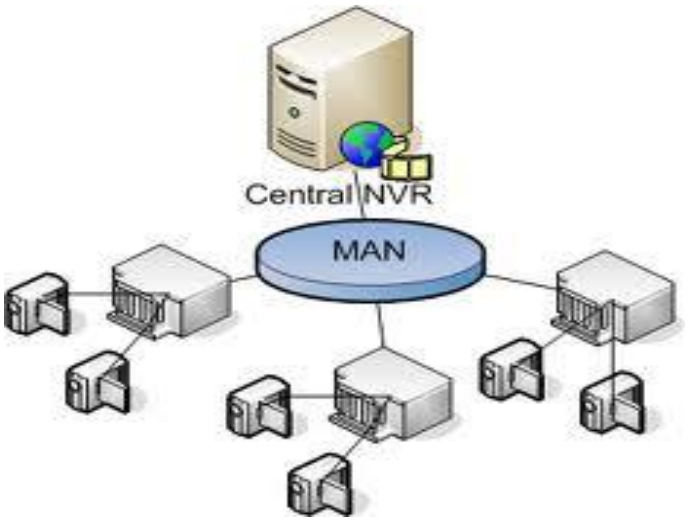
mempunyai kecepatan transfer data 10, 100, atau 1000 Mbit/s. Selain teknologi *Ethernet*, saat ini teknologi 802.11b (atau biasa disebut *Wi-fi*) juga sering digunakan untuk membentuk LAN. Tempat-tempat yang menyediakan koneksi LAN dengan teknologi *Wi-fi* biasa disebut *hotspot*.



Gambar 2. 2 LAN (*Local Area Network*)

2. *Metropolitan Area Network* (MAN)

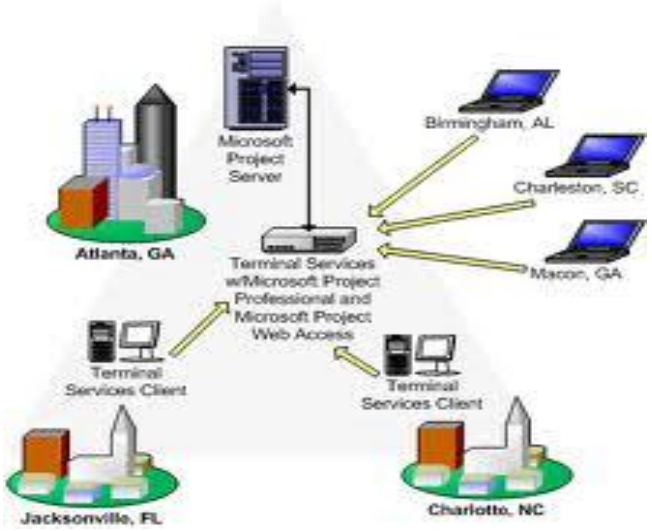
Suatu jaringan dalam suatu kota dengan transfer data berkecepatan tinggi, yang menghubungkan berbagai lokasi seperti kampus, perkantoran, pemerintahan, dan sebagainya. Jaringan MAN adalah gabungan dari beberapa LAN. Jangkauan dari MAN ini antar 10 hingga 50 km, MAN ini merupakan jaringan yang tepat untuk membangun jaringan antar kantor-kantor dalam satu kota antara pabrik/instansi dan kantor pusat yang berada dalam jangkauannya.



Gambar 2. 3 MAN (*Metropolitan Area Network*)

3. Wide Area Network (WAN)

Jaringan ini merupakan jaringan yang terbesar karena mencakup radius antar negara bahkan benua tanpa batasan geografis seperti jaringan lain. Dalam beberapa hal, WAN dapat dikatakan Internet

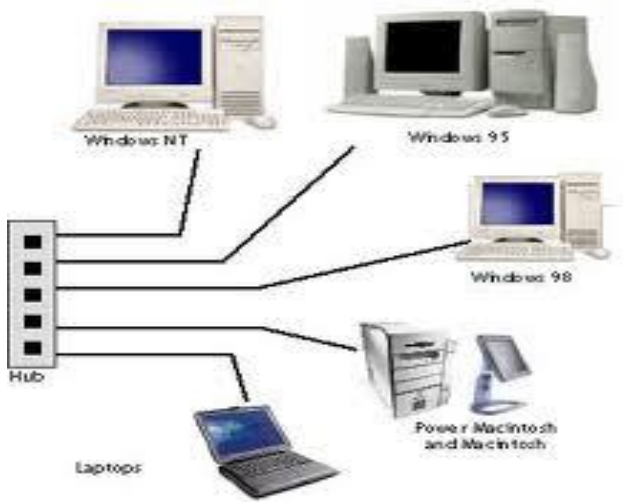


Gambar 2. 4 WAN (*Wide Area Network*)

b. Jaringan berdasarkan metode koneksi

1. Jaringan Berkabel

Jaringan yang menggunakan kabel untuk menghubungkan anggota anggota jaringan



Gambar 2. 5 Jaringan Berkabel

2. Jaringan Nirkabel

   Jaringan yang menggunakan gelombang radio untuk koneksi antar anggota jaringan. Jenis jaringan yang populer dalam kategori jaringan nirkabel ini meliputi: Jaringan kawasan lokal nirkabel (*wireless LAN/WLAN*), dan *Wi-Fi*.



Gambar 2. 6 Jaringan Nirkabel

2.2 Aplikasi *Mobile*

Pengertian aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain lain tanpa terjadipemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA. Adapun karakteristik perangkat *mobile* yaitu[9]:

1. Menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang terbatas : Perangkat *mobile* juga memiliki memory yang kecil, yaitu *primary (RAM)* dan *secondary (disk)*.
3. Daya proses yang terbatas : Sistem *mobile* tidaklah setangguh rekan mereka yaitu *desktop.*
4. Mengkonsumsi daya yang rendah : Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin desktop
5. Kuat dan dapat diandalkan : Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.
6. Konektivitas yang terbatas : Perangkat *mobile* memiliki bandwith rendah, beberapa dari mereka bahkan tidak tersambung.
7. Masa hidup yang pendek : Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

2.3 Android

Android adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan

digunakan untuk bermacam peranti bergerak. Android merupakan subset perangkat lunak untuk perangkat meliputi sistem operasi, *middleware* dan aplikasi. Sedangkan Android SDK (dan API yang diperlukan untuk mengembangkan aplikasi pada platform Android dengan menggunakan bahasa pemrograman Java. Dikembangkan bersama antara Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, NVIDIA yang tergabung dalam OHA (*Open Handset Alliance*) dengan tujuan membuat sebuah standar terbuka untuk perangkat bergerak (*mobile device*). Berikut ini adalah jenis dari android os[2]:

1. Android versi 1.1
2. Android versi 1.5 (Cupcake)
3. Android versi 1.6 (Donut)
4. Android versi 2.0/2.1 (Eclair)
5. Android versi 2.2 (Froyo: Frozen Yoghurt)
6. Android versi 2.3 (Gingerbread)
7. Android versi 3.0 (Honeycomb)
8. Android versi 4.0 (ICS: Ice Cream Sandwich)
9. Android versi 4.1 (Jelly Bean)

a. Kelebihan Android

a) Multitasking – Kalau anda pernah merasakan keunggulan dari Symbian yang bisa membuka beberapa aplikasi sekaligus, begitu juga Android yang mampu membuka beberapa aplikasi sekaligus tanpa harus menutup salah satunya.
b) Kemudahan dalam Notifikasi – Setiap ada SMS, Email, atau bahkan artikel terbaru dari RSS Reader, akan selalu ada notifikasi di *Home Screen Ponsel Android*, tak ketinggalan Lampu LED Indikator yang berkedip-kedip, sehingga Anda tidak akan terlewatkan satu SMS, *Email* ataupun *Misscall* sekalipun.
c) Akses Mudah terhadap Ribuan Aplikasi Android lewat *Google* Android *App Market* – Kalau Anda gemar *install* aplikasi ataupun games, lewat *Google Android App Market* Anda bisa men*download* berbagai aplikasi

dengan gratis. Ada banyak ribuan aplikasi dan *games* yang siap untuk Anda download di ponsel Android.

d) Pilihan Ponsel yang beranekaragam – Bicara ponsel Android, akan terasa „beda" dibandingkan dengan iOS, jika iOS hanya terbatas pada iPhone dari Apple, maka Android tersedia di ponsel dari berbagai produsen, mulai dari Sony Ericsson, Motorola, HTC sampai Samsung. Dan setiap pabrikan ponsel pun menghadirkan ponsel Android dengan gaya masing-masing, seperti Motorola dengan Motoblur-nya, Sony Ericsson dengan TimeScape-nya. Jadi Anda bisa leluasa memilih ponsel Android sesuai dengan „merk" favorite.

e) Bisa menginstal ROM yang dimodifikasi – tak puas dengan tampilan standar Android,

f) *Widget* – benar sekali, dengan adanya *Widget* di *homescreen*, Anda bisa dengan mudah mengakses berbagai setting dengan cepat dan mudah.

g) *Google* Maniak – Kelebihan Android lainnya jika Anda pengguna setia layanan *Google* mulai dari *Gmail* sampai *Google Reader*, ponsel Android telah terintegrasi dengan layanan *Google*, sehingga Anda bisa dengan cepat mengecek email dari Gmail.

b. Kelemahan Android

a) Koneksi Internet yang terus menerus – Yups, kebanyakan ponsel berbasis system ini memerlukan koneksi internet yang simultan alias terus menerus aktif.Koneksi internet GPRS selalu aktif setiap waktu, itu artinya Anda harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan.

b) Iklan – Aplikasi di Ponsel Android memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut akan selalu ada Iklan yang terpampang, entah itu bagian atas atau bawah aplikasi.

2.4 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. 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[10].

JSON terbuat dari dua struktur:

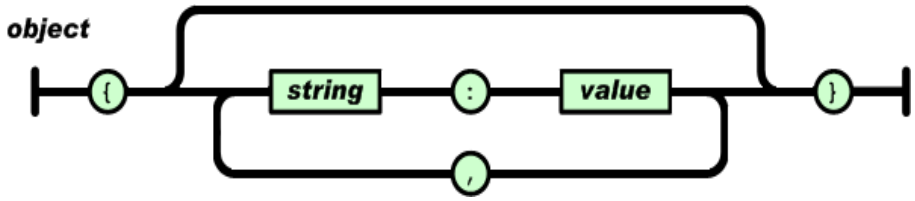
a) Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
b) Daftar nilai terurutkan (*an ordered list of values*).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

JSON menggunakan bentuk sebagai berikut:

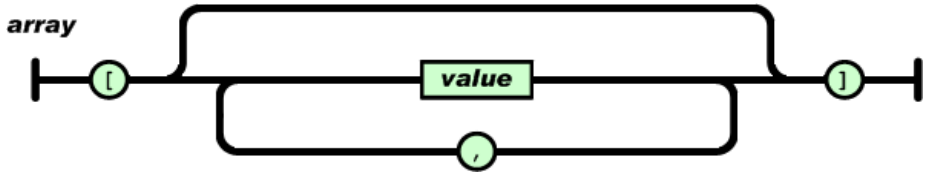
1. **Objek** adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal

tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



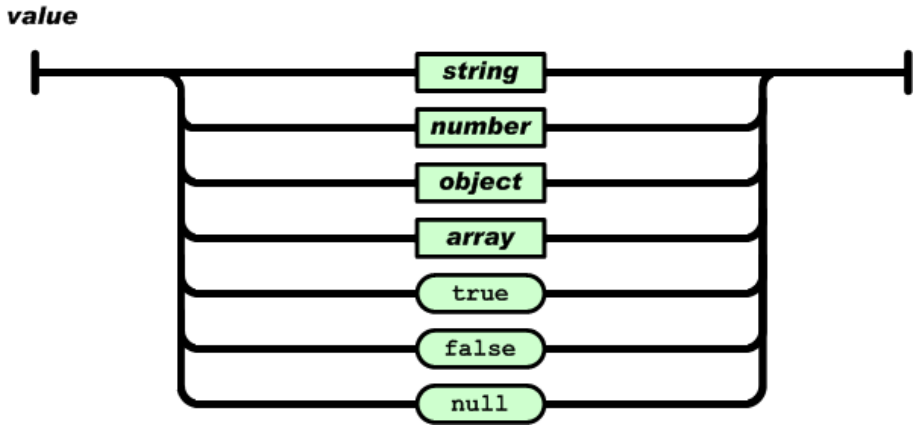
Gambar 2. 7 JSON Berbentuk *Object*

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



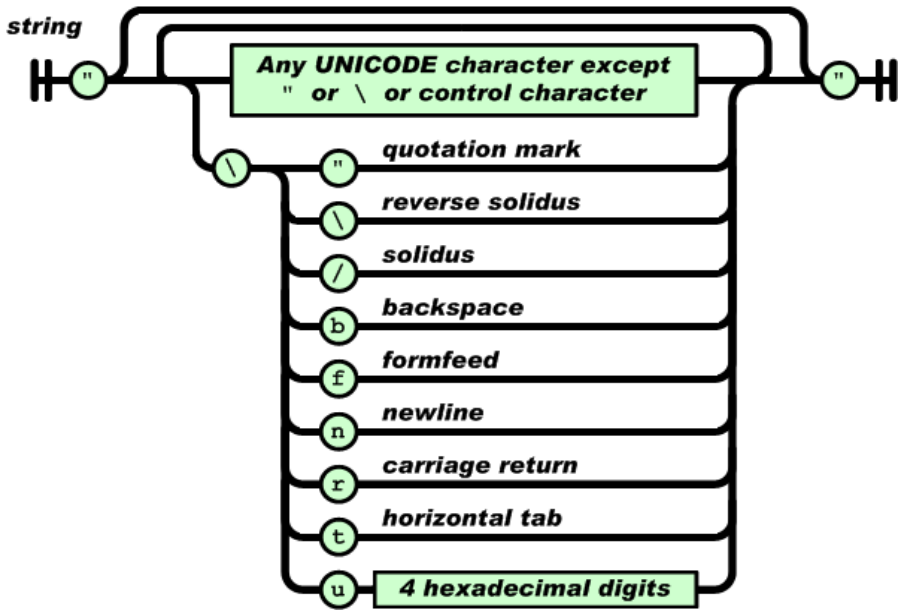
Gambar 2. 8 JSON Berbentuk *Array*

3. **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.



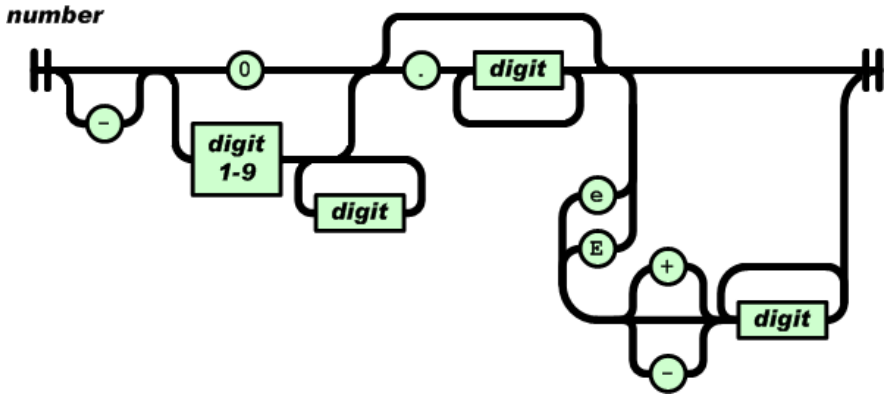
Gambar 2. 9 JSON Berbentuk *Value*

4. **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.



Gambar 2. 10 JSON Berbentuk *String*

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



Gambar 2. 11 JSON Berbentuk *Number*

2.5 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari Oracle dan dirilis tahun 1995[4]. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan *C++* namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis *Java* umumnya dikompilasi ke dalam *p-code* (*bytecode*) dan dapat dijalankan pada berbagai Mesin *Virtual Java* (JVM). *Java* merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi *Java* mampu berjalan di beberapa platformsistem operasi yang berbeda, *Java* dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini *Java* merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web (Budi Raharjo , Imam Heryanto, Arif haryono: Mudah Belajar Java. 2010).

2.5.1 Kelebihan dan Kekurangan Java

Berikut ini adalah kelebihan dan kekurangan dari bahasa pemrograman Java :

1. Kelebihan Java
   a) *Multiplatform*. Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja.
   b) OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek) yang artinya semua aspek yang terdapat di Java adalah Objek. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir.

c) Perpustakaan Kelas yang Lengkap, Java terkenal dengan kelengkapan library/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya.

d) Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java.

e) Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2. Kekurangan Java

a) Tulis sekali, perbaiki di mana saja - Masih ada beberapa hal yang tidak sesuai antara *platform* satu dengan *platform* lain. Untuk J2SE, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.

b) Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET *Platform*. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/di-reverseengineer.

c) Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal.

2.5.2 Tahap Kompilasi Java

1. Tulis / Ubah. Pemrogram menulis program dan menyimpannya di media dalam bentuk berkas '.java'.
2. Kompilasi. Pengkompilasi membentuk *bytecodes* dari program menjadi bentuk berkas '.class'.
3. Muat. Pemuat kelas memuat *bytecodes* ke memori.

4. Verifikasi. Peng-verifikasi memastikan *bytecodes* tidak mengganggu sistem keamanan Java.
5. Jalankan. Penerjemah menerjemahkan *bytecodes* ke bahasa mesin tidak bisa di pakai.

2.6 UML (*Unified Modeling language*)

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.6.1 Pengertian UML

UML adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an[5].

Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. UML adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*).

UML adalah sebuah bahasa standard untuk pengembangan sebuah software yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan software. UML tidak hanya merupakan sebuah bahasa pemograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah object-oriented database. Begitu juga mengenai pendokumentasian dapat dilakukan seperti; *requirements*, *arsitektur*, *design*, *source code*, *project plan*, *tests*, dan *prototypes*. Adapun tujuan dari penggunaan UML, sebagai berikut:

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahsa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
4. UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bias diketahui informasi secara detail tentang coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (*reserve enginering*).

2.6.2 Bagian-Bagian UML

Bagian-bagian utama dari UML adalah *view* dan diagram[7].

a. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *view* dalam UML antara lain: *use case view, logical view, component view, concurrency view,*dan *deployment view.*

1. Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya. *View* ini digambarkan dalam *use case diagram* dan kadang-kadang dengan *activity diagrams*. *View* ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

2. *Logical view*

   Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class*, *object*, dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu. *View* ini digambarkan dalam *class* diagrams untuk struktur statis dan dalam *state*, *sequence*, *collaboration*, dan *activity diagram* untuk model dinamisnya. *View* ini digunakan untuk perancang (*designer*) dan pengembang (*developer)*.

3. *Component view*

   Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya. *View* ini digambarkan dalam *component view* dan digunakan untuk pengembang (*developer*).

4. *Concurrency view*

   Membagi sistem ke dalam proses dan prosesor.View ini digambarkan dalam diagram dinamis (*state*, *sequence*, *collaboration*, dan *activity diagrams*) dan diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

5. *Deployment view*

   Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. *View* ini digambarkan dalam *deployment* diagramsdan digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

b. Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain :

1. *Use Case Diagram*

   Use case adalah abstraksi dari interaksi antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara aktor sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem agar terlihat di mata aktor. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*[5].

2. *Class Diagram*

   Class adalah dekripsi kelompok objek-objek dengan properti, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat memberikan pandangan *global* atas sebuah sistem. Hal tersebut tercermin dari *class- class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat membantu dalam visualisasi struktur *class* dari suatu sistem[5].

3. *Sequence Diagram*

   Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

4. *Activity Diagram*

   Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dibangun. Diagram ini tidak hanya memodelkan

software melainkan memodelkan model bisnis juga. Diagram aktivitas menunjukan aktivitas sistem dalam bentuk kumpulan aksi-aksi[5].

5. *Component Diagram*

   Component software merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada di benak para analis. Komponent merupakan implementasi *software* dari sebuah atau lebih *class*. Komponent dapat berupa *source code*, komponent biner, atau *executable component*. Sebuah komponent berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.Sehingga *component diagram* merepresentasikan dunia *real* yaitu *component software* yang mengandung *component*, *interface* dan *relationship*.

6. *Deployment Diagram*

   Menggambarkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

2.7 Eclipse

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

1. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Mulit-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa

pemrograman lainnya, seperti C/C++, Cobol, Python, 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 favorit dikarenakan gratis dan *opensource*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya popular adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in.

Tabel 2. 1 Peluncuran Eclipse

Kode Peluncuran	Tanggal Peluncuran	Platform	Nama Proyek
Eclipse 3.0	28 Juni 2004	3.0	
Eclipse 3.1	28 Juni 2005	3.1	
Callisto	30 Juni 2006	3.2	Callisto projects
Eroupa	29 Juni 2007	3.3	Europa projects
Ganymade	25 Juni 2008	3.4	Ganymede projects
Galileo	24 Juni 2009	3.5	Galileo projects
Helios	23 Juni 2010	3.6	

2.7.1 Arsitektur

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah dipasang (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

Secara standar 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 adalah terbuka (*open*), mudah diperluaskan (*extensible*) untuk apa saja, tidak untuk sesuatu yang spesifik. Eclipse tidak saja untuk mengembangkan program Java, tetapi juga untuk berbagai macam keperluan. Perluasan apapun cukup dengan menginstal *plug-in* yang dibutuhkan. Apabila ingin mengembangkan program C/C++ maka telah terdapat *plug-in* CDT (*C/C++ Development Tools*) yang dapat dipasang di Eclipse untuk Eclipse menjadi perangkat untuk pengembangan C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*).

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. Salah satu situs yang menawarkan *plug-in* yang gratis adalah Eclipse *downloads by project.*

2.8 Dreamweaver

Macromedia Dreamweaver adalah sebuah HTML *editor* profesional untuk mendesain secara visual dan mengelola situs *web* maupun halaman *web*. Bilamana kita menyukai untuk berurusan dengan kode-kode HTML secara manual atau lebih menyukai bekerja dengan lingkungan secara visual dalam melakukan *editing*, Dreamweaver mambuatnya menjadi lebih mudah dengan menyediakan *tool-tool* yang sangat berguna dalam peningkatan kemampuan dan pengalaman dalam mendesain *web*.

2.9 SDK (*Software Development Kit*)

Software Development Kit (*SDK*) adalah suatu *kit* atau *library* dari bahasa pemrograman untuk pengembangan atau pembangunan suatu perangkat lunak dan biasanya *SDK* terdiri dari kumpulan *tools* yang dibutuhkan. Misalnya bahasa pemrograman java, mempunyai *SDK* yang berisi suatu *library* yang dapat digunakan untuk membuat suatu aplikasi berbasis java.

2.10 JDK (*Java Development Kit*)

JDK(*Java Development Kit*) adalah sekumpulan perangkat lunak yang dapat digunakan untuk mengembangkan perangkat lunak yang berbasis Java, Sedangkan JRE adalah sebuah implementasi dari *Java Virtual Machine* yang benar-benar digunakan untuk menjalankan program java.Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber kompiler java, *bundling, debuggers, development libraries* dan lain sebagainya. Perbedaan JDK dengan SDK (*Software Development Kit*) yaitu JDK adalah sebuah SDK tetapi sebuah SDK tidak harus menjadi sebuah JDK.

2.11 ADT (*Android Development Tools*)

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 projek 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.

2.12 Basis Data

Basis data (*database*) adalah kumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya. Basis data tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. *Database* merupakan salah satu komponen yang

penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau *user*.

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamannya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Basis data relasional yang diimplementasikan dengan tabel-tabel yang saling memiliki relasi.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS). Kebutuhan basis data dalam sistem informasi meliputi[6] :

1. Memasukan, menyimpan dan memasukan data.
2. Membuat laporan berdasarkan data yang telah disimpan.

Penyusunan basis data meliputi proses memasukkan data kedalam media penyimpanan data dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data (*Database Management System DBMS*). Manipulasi basis data meliputi pembuatan pernyataan (*query*) untuk mendapatkan informasi tertentu, melakukan pembaharuan atau penggantian (*update*) data, serta pembuatan *report data*.

Tujuan utama *DBMS* adalah untuk menyediakan tinjauan abstrak dari data bagi *user*. Jadi sistem menyembunyikan informasi mengenai bagaimana data disimpan dan dirawat, tetapi data tetap dapat diambil dengan efisien. Pertimbangan efisien yang digunakan adalah bagaimana merancang struktur data yang kompleks, tetapi tetap dapat digunakan oleh pengguna yang masih awam, tanpa mengetahui kompleksitas struktur data. Basis data menjadi penting karena munculnya beberapa masalah bila tidak menggunakan data yang terpusat, seperti adanya duplikasi data, hubungan antar data tidak jelas, organisasi data dan *update* menjadi rumit. Jadi tujuan dari pengaturan data dengan menggunakan basis data adalah :

1. Menyediakan penyimpanan data untuk dapat digunakan oleh organisasi saat sekarang dan masa yang akan datang.

2. Kemudahan pemasukan data, sehingga meringankan tugas operator dan menyangkut pula waktu yang diperlukan oleh pemakai untuk mendapatkan data serta hak-hak yang dimiliki terhadap data yang ditangani.
3. Pengendalian data untuk setiap siklus agar data selalu *up-to-date* dan dapat mencerminkan perubahan spesifik yang terjadi di setiap sistem.
4. Pengamanan data terhadap kemungkinan penambahan, pengubahan, pengerusakan dan gangguan-gangguan lain.

Oprasi-oprasi dasar pada basis data :

1. Pembuatan basis data baru (*Create DataBase*)
2. Penghapusan basis data (*Drop DataBase*)
3. Pembuatan Tabel Baru (*Create Table*)
4. Penghapusan Tabel (*Drop Table*)
5. Pengisian atau penambahan data baru (*Insert Data*)
6. Pengambilan data dari suatu table (*Retrieve Data*)
7. Pengubahan Data dari suatu table (*update Data*)
8. Penghapusan data dari suatu table (*Delete Data*)

2.12.1 Elemen Basis Data

1. Entitas

Entitas adalah sekumpulan objek yang terdefinisikan yang mempunyai karakteristik sama dan bisa dibedakan satu dengan lainnya. Objek dapat berupa barang, orang, tempat atau suatu kejadian. Contoh entitas : Seseorang yang menjadi siswa di sebuah sekolah, barang yang menjadi inventaris suatu perusahaan**.** Keterangan: siswa merupakan entitas dan barang juga di sebut entitas

2. Atribut

Atribut adalah deskripsi data yang bisa mengidentifikasi entitas yang membedakan entitas tersebut dengan entitas yang lain. Seluruh atribut harus cukup untuk menyatakan identitas obyek, atau dengan kata lain, kumpulan atribut dari setiap entitas dapat mengidentifikasi keunikan suatu individu. Contoh siswa memiliki atribut : no siswa, alamat siswa, dll.

3. Data *Value* (*Nilai Data*)

Data *Value* (*Nilai Data*) adalah data aktual atau informasi yang disimpan pada tiap data, elemen, atau atribut. Atribut nama mahasiswa menunjukan tempat dimana informasi nama mahasiswa disimpan, nilai datanya misalnya adalah Andra, Andri Andro dan lain-lain yang merupakan isi data nama mahasiswa tersebut.

4. File/Table

File/Table adalah kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.

5. Record/Tuple

Record/Tuple adalah kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu *entitas* secara lengkap. Satu *record* mewakili satu data atau informasi.

2.12.2 Bahasa Basis Data

DBMS merupakan perantara bagi pemakai dengan basis data dalam disk. Cara berkomunikasi/berinteraksi antara pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS. Bahasa tersebut dapat disebut sebagai bahasa basis data yang terdiri atas sejumlah perintah yang diformulasikan oleh *user* dan diproses oleh DBMS untuk melakukan suatu aksi atau pekerjaan tertentu.

Ada 3 bahasa yang digunakan dalam basis data yaitu :

1. DDL (*Data Definition Language*)

   Merupakan bahasa definisi data yang digunakan untuk membuat dan mengelola objek *database* seperti *database*, tabel dan *view*.

2. DML (*Data Manipulation Language*)

   Merupakan bahasa manipulasi data yang digunakan untuk memanipulasi data pada objek *database* seperti tabel.

3. DCL (*Data Control Language*)

   Merupakan bahasa yang digunakan untuk mengendalikan pengaksesan data.

2.13 Rumus

Menurut kamus besar bahasa Indonesia, rumus adalah ringkasan hukum, patokan ilmu matematika atau kimia yang dilambangkan oleh huruf, angka, atau tanda (*nomina*) [3]. Rumus juga bisa diartikan sebagai pernyataan atau simpulan asas, pendirian, ketetapan yang disebut dengan kalimat ringkas dan tepat.

2.14 Matematika

Pengertian matematika sangat sulit didefinsikan secara akurat. Pada umumnya orang awam hanya akrab dengan satu cabang matematika elementer yang disebut aritmatika atau ilmu hitung yang secara informal dapat didefinisikan sebagai ilmu tentang berbagai bilangan yang bisa langsung diperoleh dari bilangan-bilangan bulat 0, 1, -1, 2, – 2, …, dst, melalui beberapa operasi dasar: tambah, kurang, kali dan bagi [3]. Akan tetapi, penulis mencoba memberikan pengertian dari matematika. Menurut bahasa kata "matematika" berasal dari kata μάθημα(máthema) dalam bahasa Yunani yang diartikan sebagai "s*ains*, ilmu pengetahuan, atau belajar" juga μαθηματικός (*mathematikós*) yang diartikan sebagai "suka belajar".

Sedangkan menurut istilah, ada yang berpendapat bahwa matematika itu bahasa simbol, matematika itu adalah bahasa numrik, matematika itu adalah bahasa yang menghilangkan sifat kabur,majemuk, dan emosional, matematika adalah metode berpikir logis , matematika adalah saran berpikir, matematika adalah logika pada masa dewasa , matematika adalah ratunya ilmu dan sekaligus menjadi pelayannya, matematika adalah sains mengenai kuantitas dan besaran, matematika adalah sains yang bekerja menarik mkesimpulan-kesimpulan yang perlu, matematika adalah sains formal yang murni, matematika adalah sains yang memanipulsi simbol, matematika adalah ilmu tentang bilangan dan ruang, matematika adalah ilmu yang mempelajari hubungan pola, bentuk dan struktur , matematika adalah imu yang abstrak dan deduktif .

Selain itu juga, beberapa pendapat para ahli tentang matematika yang telah menyinggung muatan materi yang terdapat dalam ruang lingkup matematika dan karakteristik matematika itu sendiri, yakni :

1. **James dan James** yang mengatakan bahwa matematika adalah ilmu tentang logika, mengenai bentuk,susunan, besaran dan konsep-konsep yang berhubungan lainnya dengan jumlah banyak yang terbagi kedalam tiga bidang, yaitu aljabar, analisis dan geometri.
2. **Jhonson dan Rising** bahwa matematika adalah pola berpikir,pola mengorganisasikan, pembuktian yang logik, matematika itu bahasa yang menggunakan istilah yang didefinisikan dengan cermat, jelas dan akurat, representasinya dengan simbol dan padat, lebih berupa bahasa simbol mengenai ide daripada mengenai bunyi.
3. **Reys** mengatakan bahwa matematika adalah telaahan tentang pola dan hubungan , suatu jalan atau pola pikir, suatu seni, suatu bahasa dan suatu alat.
4. **Kline** mengatakan bahwa matematika bukanlah pengetahuan menyendiri yang dapat sempurna karena dirinya sendiri, tetapi adanya matematika itu terutama untuk membantu manusia dalam memahami dan menguasai permasalahan sosial , ekonomi dan alam.

Jadi dari seluruh pendapat para ahli di atas dapat disimpulkan bahwa adanya matematika itu karena kemampuan proses berpikir manusia tentang pengalaman permasalahan yang ditemui dan dipecahkan, yang kemudian pengalaman pemecahan masalah tersebut menjadi suatu yang terkonstruksi sebagai suatu konsep matematika yang kemudian dapat digunakan sebagai alat pemecahan masalah yang sama atau yang baru.