BAB II

LANDASAN TEORI

2.1 Aplikasi

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

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi *(application suite)*. Contohnya adalah Microsoft Office dan OpenOffice.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu

lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.2 Pengertian Sistem

Pada dasarnya sistem informasi bukanlah sistem yang selalu terkait dengan komputer. Adapun beberapa penjelasan terkait dengan sistem informasi adalah pengertian sistem, pengertian data, pengertian informasi, dan pengertian sistem informasi itu sendiri.

Menurut Jogiyanto (2008:34) sistem dapat didefinisikan melalui pendekatan prosedur dan pendekatan komponen. Dalam pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. Dalam pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu.



Gambar 2.1 *Bentuk dasar suatu sistem*

Sumber Jogiyanto (2004 :689)

2.3 Bahasa

Bahasa adalah sistem lambang bunyi ujaran yang digunakan untuk berkomunikasi oleh masyarakat pemakainya. Bahasa yang baik berkembang berdasarkan suatu sistem, yaitu seperangkat aturan yang dipatuhi oleh pemakainya. Bahasa sendiri berfungsi sebagai sarana komunikasi serta sebagai sarana integrasi dan adaptasi.

2.3.1 Bahasa Arab

Bahasa Arab dikenal juga sebagai bahasa Al Quran. Sekarang ini bahasa Arab tidak hanya menjadi bagian dari kurikulum sekolah Islam namun pada beberapa sekolah negeri juga telah memasukkan bahasa Arab dalam kurikulum. Selain bisa belajar bahasa Arab di sekolah, kita juga bisa menemukan lembaga kursus yang mengajarkan bahasa Arab.

Bagi umat Muslim, menguasai bahasa Arab adalah hal yang sangat esensial karena dengan bisa berbahasa Arab, maka kita bisa lebih memahami kitab suci kita, Al Quran, serta hal-hal penting lainnya.

2.3.2 Bahasa Inggris

Bahasa Inggris kini menjadi salah satu bahasa internasional. Penggunaan bahasa Inggris banyak dijumpai dimana-mana. Penggunaan bahasa Inggris juga kerap dilakukan dalam kehidupan sehari-hari. Terlebih lagi sekarang banyak dijumapi sekolah-sekolah yang menggunakan dua bahasa dan salah satunya adalah bahasa Inggris.

2.4 Konsep Dasar *Translate*

Translate dalam bahasa Indonesia adalah Terjemahan yang berarti interpretasi makna suatu teks dalam suatu bahasa ("teks sumber") dan penghasilan teks yang merupakan padanan dalam bahasa lain ("teks sasaran" atau "terjemahan") yang mengkomunikasikan pesan serupa. Terjemahan harus mempertimbangkan beberapa batasan, termasuk konteks, aturan tata bahasa, konvensi penulisan, idiom, serta hal lain antar kedua bahasa. Orang yang melakukan terjemahan disebut sebagai penerjemah.

Secara tradisional terjemahan merupakan suatu kegiatan manusia, walaupun banyak upaya telah dilakukan untuk mengotomatisasikan penerjemahan teks bahasa alami (terjemahan mesin, *machine translation*) atau menggunakan komputer sebagai alat bantu penerjemahan (penerjemahan berbantuan komputer, *computer-assisted translation*)

(http://id.wikipedia.org/wiki/Terjemahan/6-Januari-2013)

2.5 Konsep Aplikasi *Mobile*

Mobile Application adalah perangkat lunak berbasis mobile (*PDA, handphone, smartphone, Pocket PC*).

Aplikasi *software mobile* dapat terintegrasi dengan perangkat *handphone* dan dipakai klien sebagai aplikasi mini yang dapat dibawa kemanapun untuk membantu berbagai aktifitas klien.

(http://www.e-scapemedia.co.id/mobile-application.html)

2.6 Android

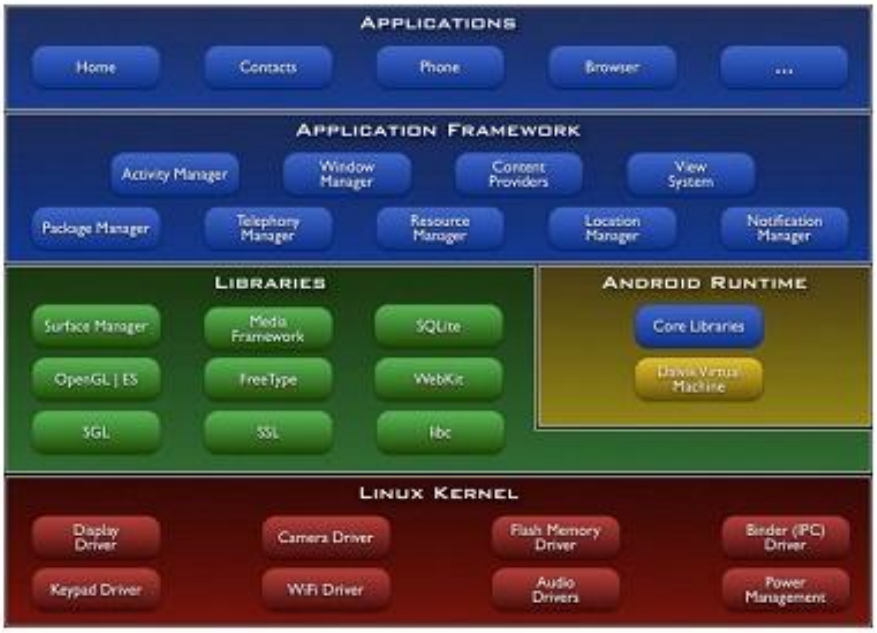
Android atau Sistem Operasi *Android* adalah- sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc*. membeli *Android Inc*., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung, pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode - kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Android is an operating system (OS) developed by the Open Handset Alliance (OHA). The Alliance is a coalition of more than 50 mobile technology companies ranging from handset manufactures and service providers to semiconductor manufacturers and software developers, including Acer, ARM, Google, eBay, HTC, Intel, LG Electronics, Qualcomm, Sprint, and T-Mobile. The stated goal of the OHA is to "accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience" (OHA, 2009, n.p.).

2.6.1 Arsitektur Android

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



Gambar 2.4 *Detail Arsitektur Android*

2.6.2 Android SDK

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android 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 untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi netral, Android memberi kesempatan untuk membuat aplikasi yang berbasis *Handphone/Smartphone*. Beberapa fitur - fitur Android yang paling penting adalah:

a. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*
b. Mesin *virtual Dalvik* dioptimalkan untuk perangkat *mobile*
c. *Integrated browser* berdasarkan *engine open source WebKit*
d. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasrkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware)
e. SQLite untuk penyimpanan data (*database*)
f. Media Support yang mendukung audio, video, dan gambar.
g. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
h. Kamera, GPS, kompas, dan *accelerometer* (tergantung hardware)
i. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

2.7 Tinjauan Perangkat Lunak Pendukung

Perangkat pendukung yang penulis pergunakan adalah Mozilla Firefox, bahasa pmrograman PHP, Java, Eclipse, Adobe Photoshop, XAMPP serta Photoscape.

2.7.1 JAVA

Java adalah bahasa pemograman tingkat tinggi yang berorientasi objek, diedarkan oleh Sun Microsystem1 pada awal tahun 1996. Sejarah awal Java berawal di tahun 1991 ketika satu group insinyur-insinyur Sun, yang dipimpin oleh *Patrick Naughton* dan *James Gosling,* ingin mendesain sebuah bahasa pemograman komputer yang berukuran kecil yang dapat digunakan untuk

peralatan elektronika konsumen sepertu *switchboxes* TV kabel.

Dikarenakan peralatan-peralatan ini menggunakan konsumsi daya dan *memory* yang rendah, maka bahasa pemograman tersebut harus berukuran sangat kecil. Juga karena setiap vendor menggunakan CPUs (*Central Processing Unit*) yang berbeda, maka bahasa tersebut harus bersifat multiplatfrom, tidak terikat hanya pada satu arsitektur (*Architecture Neutral*). Proyek ini diberi nama *Green Project.*

(sumber : blog.trisakti.ac.id/labkomputer/files/2010/05/MODULDKP/27-juni-2012)

2.7.2 Karakteristik JAVA

1. Sederhana

Inti dari Java terutama terletak pada kemiripannya dengan C dan C++. Karena programmer-programmer masa kini khususnya yang tertarik menggunakan Java telah memiliki pengalaman setidaknya dengan C dan mungkin dengan C++, Java tentu saja terlihat sederhana dan akrab bagi programmer-programmer ini.

Java menyederhanakan bahasa C++ dengan menambahkan fitur-fitur pendukung yang belum terdapat dalam C++ dan membuang beberapa fitur yang membuat C++ menjadi bahasa yang rumit dan sulit untuk dikuasai. Java sederhana karena hanya ia memiliki tiga tipe angka data primitif, tipe *Boolean* dan *array*. Selebihnya, semua dalam Java adalah sebuah kelas. Salah satu fitur penting yang ditawarkan Java (yang belum ada pada C++) adalah pengumpulan sampah

(*garbage collection*). Dengan mekanisme ini, user tidak perlu membebaskan memori yang dialokasikan, karena semua dilakukan oleh Mesin *Virtual Java*. Java juga mendukung penulisan program multi jalinan, yaitu suatu program yang dapat melakukan lebih dari satu pekerjaan dalam waktu yang bersamaan.

2. Berorientasi Objek

Dalam pendekatannya pada orientasi obyek, Java lebih merujuk pada SmallTalk daripada C++. Selain tipe data primitivenya, semua yang ada pada Java adalah kelas. Sebaliknya dalam C++ semuanya serba campur, dimana programmer dapat secara bebas (benar-benar bebas)dalam mencampur kode-kode orientasi obyek (kelas) (kelas) dengan kode prosedural (fungsi). Dalam Java semua ini tidak diperbolehkan. Tidak ada fungsi global dalam Java, semua fungsi harus dipanggil melalui sebuah obyek.

Dukungan Java terhadap orientasi-obyek tidak termasuk penurunan (*inheritansi*) ganda. Para perancang Java merasa kerumitan yang dimunculkan *inheritansi* ganda tidak sebanding dengan keuntungan yang dihasilkan. Sebagai ganti dari inheritansi ganda, Java kemudian menyediakan antarmuka (*interface*).

3. Terdistribusi

Java memudahkan pembuatan aplikasi terdistribusi dengan sekumpulan kelas yang digunakan pada aplikasi-aplikasi jaringan. Dengan menggunakan kelas URL (*Uniform Resource Locator*) Java, suatu aplikasi dapat dengan mudah mengakses server-jauh (remote server).

4. Terinterpretasi

Karena Java adalah bahasa interpretasi, sekali Interpreter Java terpasang, terinstal pada suatu mesin tertentu, mesin tersebut dapat secara langsung menjalankan aplikasi-aplikasi Java (tidak peduli platform atau sistem operasi apa yang terpasang pada mesin tersebut). Ketika menggunakan bahasa interpreter, programmer juga terbebas dari kekhawatiran berkaitan dengan ketergantungan antar modul.

Keuntungan lain adalah waktu yang digunakan untuk lingkaran *edit-compile-link-test* dapat dipotong. Tanpa adanya langkah kompile dan link, maka bekerja dalam lingkungan interpreter lebih sederhana dan lebih hemat waktu karena hanya melalui lingkaran edit-test.

5. Kokoh

Membuat suatu program yang terdistribusi, mendukung multi-jalinan yang dapat dijalankan pada berbagai sistem operasi dan berbagai prosesor bukanlah suatu pekerjaan yang mudah. Berdasarkan pemikiran ini, maka Java diciptakan sebagai bahasa yang sangat ketat dalam penulisan *(strongly typed language*). Dalam Java management memori telah disederhanakan dengan dua cara. Pertama Java tidak mendukung manipulasi *pointer* atau aritmatik secara langsung, sehingga mustahil bagi program Java untuk menumpuk (meng-*overwrite*) memori atau mengkorupsi data. Kedua Java menggunakan mekanisme pengumpulan sampah saat program berjalan (*runtime*) daripada pembebasan memori secara *eksplisit*.

6. Aman

Karena Java tidak menggunakan pointer yang secara langsung merujuk pada lokasi memori seperti yang terdapat pada C atau C++, Java memiliki kendali penuh terhadap semua kode yang ada pada lingkungan Java. Sebelumnya telah diantisipasi bahwa aplikasi-aplikasi Java akan dijalankan di internet dan akan secara dinamis dapat dijalankan bersama atau mengeksekusi program lain di tempat yang berlainan melalui internet, maka para pengembang Java berpendapat tentang perlu adanya kompiler Java yang akan menghasilkan kode bite Java yang akan melewati standar keamanan *runtime* Java

7. Arsitektur Netral

Sebelum dijalankan, program Java harus dikompile terlebih dulu menggunakan kompiler Java. Proses kompilasi ini kemudian akan menghasilkan suatu kode bite tertentu yang serupa dengan *file* kode (bahasa) mesin yang dapat dieksekusi pada mesin apapun yang memiliki interpreter Java.

8. *Portable*

Salah satu tujuan penting pembuatan Java adalah kode-kode Java haruslah *Portable*, sehingga ketika arsitektur baru (baik itu perangkat keras ataupun sistem operasi atau keduanya) berkembang, lingkungan Java dapat diterapkan dan dipindahkan pada mereka.

Pada Java, semua tipe data *primitif* (*integer, long, float, double* dan sebagainya) memiliki ukuran tertentu, tidak bergantung pada mesin atau sistem

operasi dimana program Java dijalankan. Hal ini sangat berlawanan dengan bahasa seperti C atau C++ yang menyerahkan ukuran tipe data primitif pada kompiler dan mesin (serta sistem operasi). Java *portable* karena kompiler Java sendiri ditulis menggunakan Java.

9. Berkinerja Tinggi

Sebuah aplikasi Java tidak akan menyamai kinerja aplikasi bahasa terkompilasi penuh seperti C atau C++. Namun demikian untuk sebagian besar aplikasi termasuk pengolahan grafis, satu dari berbagai macam hal yang dapat ditemui secara umum pada *World Wide Web*, kinerja Java lebih dari cukup.

Banyak pengadopsi awal C++ khawatir aanya kemungkinan penurunan kinerja saat mereka mengubah program mereka dari C ke C++. Namun para pengadopsi (pengguna) awal C++ menemukan walaupun program dalam C akan melebihi suatu program C++ dalam banyak kasus, waktu untuk pengembangan lebih lanjut dan biaya tidak sesuai dengan kinerja yang dihasilkan.

Walupun aplikasi Java mungkin tidak dapat menyamai aplikasi C++, namun secara normal aplikasi Java akan cukup cepat, dan Java memungkinkan user melakukan banyak hal yang tidak dapat dilakukan dengan C++.

10. Multi-Jalinan

Menulis suatu bahasa yang hanya bisa melakukan satu perkerjaan adalah hal yang biasa dalam bahasa pemrograman. Program-program Java dapat terdiri atas beberapa jalinan yang memungkinkan program untuk melakukan beberapa

pekerjaan sekaligus. Suatu contoh, sebuah program multi jalinan dapat me-render sebuah image di layar pada satu jalinan disamping menerima masukan dari keyboard user pada jalinan utama-nya.

Dukungan terhadap beberapa jalinan tersinkronisasi sekaligus telah terbangun (secara internal) pada Java dan lingkungan *runtime*-nya. Jalinan tersinkronisasi sangat berguna dalam membuat aplikasi yang terdistribusi dan berbasis jaringan. Aplikasi semacam ini dapat berkomunikasi dengan server jauh pada satu jalinan dan berinteraksi dengan user pada jalinan yang lain.

11. Dinamis

Karena terinterpretasi, Java adalah bahasa yang benar-benar dinamis. Saat *runtime*, lingkungan Java dapat mengembangkan dirinya dengan terhubung pada kelas-kelas yang mungkin terletak pada server-jauh pada suatu jaringan (misalnya, lewat internet). Pada C++ setiap kali anggota variabel atau fungsi ditambahkan pada suatu kelas, maka kelas tersebut dan semua kode tambahan yang merujuk pada kelas tersebut perlu dikompile ulang. Java menyederhanakan masalah ini dengan menyerahkan pada *runtime*. Saat *runtime* interpreter Java melakukan resolusi nama ketika terhubung dengan kelas-kelas yang bersangkutan. Interpreter Java juga bertanggung-jawab dalam menentukan penempatan obyek dalam memori. Dua fitur pada interpreter Java ini memecahkan masalah berkaitan degan perubahan definisi kelas ketika digunakan kelas-kelas yang lain.

2.7.3 JDK (Java Development Kit) jdk-6u5-windows-i586-p

Java Development Kit (JDK) merupakan perlengkapan tempur yang mendasar dalam pengembangan aplikasi dengan Java. Dua program utama yang disediakan dalam JDK adalah :

1. javac, yaitu program untuk meng-*compile* kode sumber
2. java, yaitu program untuk meluncurkan aplikasi

Seperti namanya, JDK adalah semacam kotak peralatan (kit) yang digunakan untuk development. JDK ini berguna saat Anda menulis code program. Seperti halnya JRE, JDK juga memiliki JVM di dalamnya. Jadi kesimpulannya JDK digunakan untuk "menulis" program Java.

2.7.4 Eclipse Galileo

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 *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari *Eclipse* yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in.*

(http://id.wikipedia.org/wiki/Eclipse_%28perangkat_lunak%29/12/September/2012)

2.7.5 Adobe Photoshop

Adobe Photoshop, atau biasa disebut Photoshop, adalah perangkat lunak editor citra buatan Adobe Systems yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (market leader) untuk perangkat lunak pengolah gambar/foto, dan, bersama Adobe Acrobat, dianggap sebagai produk terbaik yang pernah diproduksi oleh Adobe Systems. Versi kedelapan aplikasi ini disebut dengan nama Photoshop CS (Creative Suite), versi sembilan disebut Adobe Photoshop CS2, versi sepuluh disebut Adobe Photoshop CS3 , versi kesebelas adalah Adobe Photoshop CS4 dan versi yang terakhir (keduabelas) adalah Adobe Photoshop CS5.

2.7.6 Photoscape

Photoscape adalah perangkat lunak editor citra yang dikembangkan oleh MOOII Tech, Korea. Konsep dasar PhotoScape adalah "Mudah dan Menyenangkan" Sehingga memungkinkan pengguna untuk dengan mudah mengedit foto yang diambil dari kamera digital atau kamera ponsel. PhotoScape menyediakan antarmuka pengguna sederahan untuk melakukan tambahan foto umum termasuk penyesuaian warna, memotong, mengubah ukuran, percetakan dan animasi GIF.

(Sumber :http://id.wikipedia.org/wiki/Photoscape/10/ Oktober /2012).

2.7.7 XML

XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet. Teknologi XML adalah teknologi keturunan dari SGML (*Standard Generalized Markup Language, ISO 8879*) yang dikembangkan pada tahun 1980-an. XML adalah suatu bahasa Markup, yaitu bahasa yang berisikan kode-kode berupa tanda-tanda tertentu dengan aturan tertentu untuk memformat dokumen teks dengan tag sendiri agar dapat dimengerti. XML adalah language untuk mengidentifikasi dan menandai data terstruktur

XML didesain untuk mempu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah

bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang view database, tetapi dengan suatu cara yang standar.

XML memiliki tiga tipe file, diantaranya:

1. XML, merupakan standar format dari struktur berkas (file).
2. XSL, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
3. XSD, merupakan standar yang mendefinisikan struktur database dalam XML.

Keunggulan **XML** bisa diringkas sebagai berikut :

1. Pintar *(Intelligence)*. XML dapat menangani berbagai tingkat (level) kompleksitas.
2. Dapat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri. Seperti *Microsoft* membuaat bahasa MSXML atau Macromedia mengembangkan MXML.
3. Sederhana. XML lebih sederhana.
4. Mudah dipindah-pindahkan *(Portability)*. XML mempunyai kemudahan perpindahan *(portabilitas)* yang lebih bagus.

(Sumber: http://id.wikipedia.org/wiki/XML /10/ Oktober /2012)