BAB II

LANDASAN TEORI

2.1 Memasak

Memasak adalah suatu proses menciptakan sebuah makanan dengan mengurutkan konsep yang ada yaitu langkah-langkah dalam memasak, yang berarti juga memasak adalah mencipta sebuah karya yg bisa kita nikmati dan juga orang lain. Makanan hasil olahan tersebut menjadi sesuatu yang mempunyai cita rasa tersendiri tergantung pada cara pengolahannya

(http://kualicom.blogspot.com/2009/11/pengertian-memasak.html)

2.1.1 Sejarah Memasak

Cara memasak tertua dikenal sampai saat ini adalah membakar. Tidak ada yang tahu persis bagaimana manusia purba menemukan teknik ini, namun para ahli menduga bahwa mungkin manusia purba menemukannya ketika mereka memakan daging hewan yang mati dalam kebakaran hutan. Teknik pembakaran dan pemanggangan di atas bara api sendiri bisa ditemukan mulai dari masyarakat yang paling terisolir dan primitif sampai manusia modern. Kunci dalam memasuki fase variasi memasak yang lainnya, muncul ketika manusia menggunakan tanah liat untuk alat memasak (pot masak). Dengan penemuan tanah liat, dan juga dengan teknik memasak dengan batu, manusia

menemukan cara - cara lain seperti merebus makanan, mengasap daging, sampai membuat oven tanah liat.

Seiring waktu, variasi memasak juga menjadi semakin maju dengan munculnya dan berkembangnya cara-cara bercocok tanam. Pada masa-masa ini, manusia juga belajar mengolah tumbuhan untuk pelengkap makanan, khususnya biji-bijian, yang kemudian menjadi sumber karbohidrat di meja makan.

(http://resepmasakanindonesia.info/info-kuliner-sejarah-masakan/)

2.1.2 Bahan Pangan

Bahan pangan atau disebut juga bahan pokok makanan berasal dari hewan maupun tumbuhan yang dikomsumsi oleh makhluk hidup untuk memberikan tenaga dan nutrisi.

2.1.2.1 Bahan Pangan Nabati

Bahan pangan nabati adalah bahan-bahan makanan yang berasal dari tanaman (bisa berupa akar, batang, dahan, daun, bunga, buah atau beberapa bagian dari tanaman bahkan keseluruhannya) atau bahan makanan yang diolah dari bahan dasar dari tanaman

(http://cai-sl.blogspot.com/2012/06/pengertian-bahan-pangan-hewani-dan.html)

2.1.2.2 Bahan Pangan Hewani

Bahan pangan hewani merupakan bahan-bahan makanan yang berasal dari hewan atau olahan yang bahan dasarnya dari hasil hewan. Kedua bahan pangan ini memiliki karakteristik yang berbeda sehingga memerlukan penanganan dan pengolahan yang berbeda pula. Bahan pangan hewani meliputi susu, telur, daging dan ikan serta produk-produk olahannya yang bahan dasarnya berasal dari hasil hewani.

Bahan pangan hewani memiliki karakteristik sebagai berikut :

a. Bahan pangan hewani memiliki daya simpan yang jauh lebih pendek daripada bahan pangan nabati bila dalam keadaan segar (kecuali telur). Pendeknya daya simpan ini terkait dengan struktur jaringan hasil hewani dimana bahan pangan hewani tidak memiliki jaringan pelindung yang kuat dan kokoh sebagaimana pada hasil tanaman.

b. Bahan pangan hewani bersifat lunak dan lembek sehingga mudah terpenetrasi oleh faktor tekanan dari luar.

c. Karakteristik masing-masing bahan pangan hewani sangat spesifik sehingga tidak bisa digeneralisasi. Sifat pada daging sangatlah berbeda dengan sifat telur. Berbeda dengan pangan nabati yang memiliki kesamaan dalam hal jaringan-jaringan atau komponen-komponen penyusunnya. Pada bahan pangan hewani, lemak pada daging terletak pada jaringan lemak,

pada susu terletak pada globula-globula lemak dan pada telur terdapat pada kuning telur.

d. Bahan pangan hewani pada umumnya merupakan sumber protein dan lemak dan bahan pangan nabati merupakan sumber karbohidrat, vitamin, mineral, lemak dan protein.

(http://cai-sl.blogspot.com/2012/06/pengertian-bahan-pangan-hewani-dan.html)

2.2 Pengertian Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi.

Android merupakan generasi baru *platform mobile* yang memberikan kesempatan kepada pengembang untuk melakukan pengembangan sesuai dengan yang diharapkan. Sistem operasi yang mendasari Android merupakan lisensi di bawah naungan GNU, General *Public License* Versi 2(GPLv2), yang biasa dikenal dengan istilah *Copyleft*. Istilah *copyleft* ini merupakan lisensi yang setiap perbaikan oleh pihak ketiga harus terus jatuh di bawah *terms*.

Distribusi Android berada di bawah lisensi Apache *Software* (ASL/Apache2), yang memungkin untuk distribusi kedua atau seterusnya. Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan.

Pengembang memiliki beberapa pilihan dalam membuat aplikasi yang berbasis Android. Namun kebanyakan pengembang menggunakan Eclipse sebagai IDE untuk merancang aplikasi mereka. Hal ini diikarenakan Eclipse mendapat

dukungan langsung dari Google untuk menjadi IDE pengembangan aplikasi Android. Aplikasi Android dapat dikembangkan pada berbagai sistem operasi, diantaranya adalah:

a. Windows XP/Vista/7

b. Mac OS X (Mac OS X 10.48 atau yang lebih baru)

c. Linux

(http://id.wikipedia.org/wiki/Android_(sistem_operasi))

2.2.1 Sejarah Android

Pada saat perilisan perdana Android pada tanggal 5 November 2007, Android bersama *Open Handset Alliance* mendukung pengembangan standar terbuka pada perangkat seluler tersebut. Di sisi lain, Google merilis kode-kode Android di bawah lisensi Apache. Sehingga terdapat dua jenis distributor sistem operasi Android yaitu yang mendapat dukungan penuh dari Google dan yang mendapat dukungan penuh dari Open Handset Distribution (OHD).

Telepon selular pertama yang menggunakan sistem operasi Android adalah HTC Dream yang dirilis pada 22 Oktober 2008. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM *Holdings, Atheros Communication* yang diproduksi oleh *Asustek Computer* Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp dan Vodafone Group Plc.

Hingga saat ini terdapat beberapa versi dari sistem operasi Android, antara lain:

a. Android versi 1.1

Dirilis pada 9 Maret 2009. Android versi ini dilengkapi dengan adanya jam, alarm, *voice search*, pengiriman pesan dengan Gmail dan pemberitahuan *email*.

b. Android versi 1.5 (*Cupcake*)

Dirilis pada Mei 2009. Terdapat pembaruan dari versi 1.1 diantaranya adalah fitur *upload* video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan *bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

c. Android versi 1.6 (*Donut*)

Dirilis pada September 2009. Pembaruan yang terdapat pada versi ini diantaranya adalah proses pencarian yang lebih baik, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, *camcorder* dan galeri yang diintegrasikan, CDMA/EVDO, 802.1x, VPN, *Gestures*, *Text-to-speech engine*.

d. Android versi 2.1 (*Éclair*)

Dirilis pada 3 Desember 2009. Perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3.2 MP, digital zoom dan *bluetooth* 2.1.

e. Android versi 2.2 (Froyo)

Dirilis pada 20 Mei 2010. Versi Android inilah yang sekarang banyak digunakan sebagai standar sistem operasi mereka. Terdapat perubahan yang cukup signifikan dari versi sebelumnya diantaranya adalah kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia, *Dalvik Virtual Machine* (DVM) yang dioptimalkan untuk perangkat *mobile*, grafik di 2D dan 3D berdasarkan *libraries* OpenGL, SQLite, mendukung berbagai format audio dan video, GSM, *bluetooth*, EDGE, 3G, Wifi, kamera, *Global Positioning System* (GPS), kompas dan *accelerometer*.

f. Android versi 2.3 (*GingerBread*)

Dirilis pada 6 Desember 2010. Beberapa perbaikan fitur dari versi sebelumnya adalah SIP-*based* VoIP, *Near Field Communications* (NFC), *gyroscope* dan sensor, *multiple cameras support*, *mixable audio effect* dan *download manager*.

g. Android versi 3.0 (*Honeycomb*)

Dirilis tahun 2011. Android versi ini dirancang khusus untuk tablet, sehingga terdapat perbedaan dari fitur UI (*User Interface*). *Honeycomb* sengaja dibuat untuk layar yang lebih besar dan juga dapat mendukung *multiprocessor*.

h. Android versi 4.0 (*Ice Cream*)

Versi ini masih dalam pengembangan. Dari berbagai informasi menyebutkan bahwa versi *Ice Cream* merupakan gabungan antara versi *Gingerbread* dengan *Honeycomb*. Sehingga bisa digunakan untuk ponsel maupun tablet. kemungkinan dirilis pada quarter ke 4 tahun 2011.

(http://id.wikipedia.org/wiki/Android_(sistem_operasi))

2.2.2 Pengertian *Smartphone* (Telepon Pintar)

Smartphone adalah sebuah telepon yang kegunaan dasarnya sama dengan telepon biasa yang dapat dibawa kemana-mana dan tidak perlu disambungkan dengan kabel, namun memiliki kemampuan tingkat tinggi dengan fungsi yang menyerupai komputer.

(http://id.wikipedia.org/wiki/Telepon_pintar/)

2.2.3 Pengertian Java

Java adalah bahasa pemrograman serbaguna. Java dapat digunakan untuk membuat suatu program sebagaimana anda membuatnya dengan bahasa seperti Pascal atau C++. Yang lebih menarik, Java juga mendukung sumber daya internet yang saat ini populer, yaitu *World Wide Web* atau yang sering disebut web. Java juga mendukung aplikasi klient/server, baik dalam jaringan local (LAN) maupun jaringan berskala luas (WAN).

Java dikembangkan oleh *Sun Microsystems* pada Agustus 1991, dengan nama semula Oak. Konon Oak adalah pohon semacam Jati yang terlihat dari jendela tempat pembuatnya, James Gosling, bekerja. Ada yang mengatakan bahwa Oak adalah singkatan dari "*Object Application Kernel*", tetapi ada yang menyatakan hal itu muncul setelah nama Oak diberikan. Pada Januari 1995, karena nama Oak dianggap kurang komersial, maka diganti menjadi Java.

Dalam sejumlah literatur disebutkan bahwa Java merupakan hasil perpaduan sifat dari sejumlah bahasa pemrograman, yaitu C, C++, Object-C, *SmallTalk*, dan *Common* LISP. Selain itu Java juga dilengkapi dengan unsur keamanan. Yang tak kalah penting adalah bahwa Java menambahkan paradigma pemrograman yang sederhana. Jika telah mengenal C atau C++, yang mengandalkan *pointer*, Java justru meninggalkannya sehingga anda akan memperoleh kemudahan saat menggunakannya.

Program java bersifat tidak bergantung pada *platform*, artinya, Java dapat dijalankan pada sembarang komputer dan bahkan pada sembarang sistem operasi. Ketidak bergantungan pada platform sering dinyatakan dengan istilah portabilitas. Yang menarik, tingkat portabilitas java tidak hanya sebatas pada program sumber (*source code*), melainkan juga pada tingkat kode biner yang disebut *bytecode*. Dengan demikian bila telah mengkompilasi program java pada komputer berbasis operasi Windows, dapat juga menjalankan hasil kompilasi pada Macintosh secara langsung, tanpa perlu mengkompilasi ulang.

Java merupakan sebuah teknologi yang diperkenalkan oleh *Sun Microsystems* pada pertengahan tahun 1990. Menurut definisi dari *Sun*, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan.

Java2 adalah generasi kedua dari Java platform (generasi awalnya adalah Java *Development Kit*). Java berdiri di atas sebuah mesin interpreter yang diberi nama JVM (*Java Virtual Machine*). JVM inilah yang akan membaca *bytecode* dalam *file* .class dari suatu program sebagai representasi langsung program yang

berisi bahasa mesin. Oleh Karena itu bahasa Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Platform Java terdiri dari kumpulan *library*, JVM, kelas-kelas yang dipaket dalam sebuah lingkungan rutin Java, dan sebuah *compiler*, *debugger*, dan perangkat lain yang dipaket dalam Java *Development Kit* (JDK). Java2 adalah generasi yang sekarang sedang berkembang dari *platform* Java. Agar sebuah program Java dapat dijalankan, maka *file* dengan ekstensi ".java" harus dikompilasi menjadi file *bytecode*. Untuk menjalankan *bytecode* tersebut dibutuhkan JRE (*Java Runtime Environment*) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi. JRE berisi JVM dan *library* Java yang digunakan.

(http://ilmucomputer2.blogspot.com/2010/02/pengertian-java.html/)

2.2.4 *Java Software Development Kit* (SDK Java)

SDK Java merupakan kebutuhan utama bagi programmer untuk membuat dan menjalankan java. Komponen JDK antara lain *compiler* (javac), *interpreter* (java) disebut juga java *virtual machine* atau java *runtime environment*, *applet viewer* (*appletviewer*), *debugger* (jdb), java class *library* (jcl), *header* dan *stub generator* (javah), dan yang paling penting yaitu java *documentation* (javadoc).

(http://en.wikipedia.org/wiki/Java_Development_Kit)

2.2.5 *Eclipse Software Development Kit* (SDK Eclipse)

SDK Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*).

(http://id.wikipedia.org/wiki/Eclipse_(perangkat_lunak))

2.2.6 *Android Software Development Kit* (SDK Android)

SDK Android adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java.

(http://developer.android.com/guide/index.html/Android SDK Developer Guide/)

2.2.7 *Android Eclipse Plugin* (AEP)

Definisi *Android Eclipse Plugin* adalah suatu perangkat tambahan koneksi Eclipse dengan Android SDK, sehingga bisa lebih memudahkan membuat program untuk android di Eclipse.

(http://developer.android.com/tools/sdk/eclipse-adt.html)

2.2.8 *Android Package* (APK)

APK adalah paket aplikasi Android (*Android PacKage*). APK umumnya digunakan untuk menyimpan sebuah aplikasi atau program yang akan dijalankan pada perangkat Android. APK pada dasarnya seperti zip *file*, karena berisi dari kumpulan *file*, dapat diperoleh melalui berbagai metode, seperti menginstal

sebuah aplikasi melalui m*arket*, download dari sebuah situs web, atau membuat sendiri dengan bahasa Java.

(http://en.wikipedia.org/wiki/APK_(file_format))

2.2.9 *Android Virtual Devices* (AVD)

AVD adalah konfigurasi dari emulator sehingga kita dapat menjalankan perangkat Android sesuai model yang dipilih, misalkan Android 1.5 atau 2.2. Untuk dapat menjalankan emulator.

(http://developer.android.com/tools/devices/index.html)

2.2.10 Emulator

Emulator atau lebih tepatnya piranti lunak emulator memungkinkan suatu program atau piranti lunak yang dibuat pada awalnya oleh suatu sistem komputer (arsitektur dan sistem operasi) dan untuk dijalankan dalam sistem itu (atau dijalankan dalam suatu sistem yang didedikasikan), dapat dijalankan dalam sistem komputer yang sama sekali berbeda. Sebagai contoh suatu program Windows dapat dijalankan di sistem operasi Linux dengan menggunakan piranti lunak emulator Wine. Ada pula program yang mengemulasikan suatu komputer dalam komputer, misalnya VMware. Contoh lain adalah program-program emulator untuk menjalankan permainan komputer yang awalnya hanya bisa dijalankan pada konsolnya masing-masing.

(http://id.wikipedia.org/wiki/Emulator)

2.3 Definisi Perangkat Yang Digunakan

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*). 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 popular adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

Karakteristik Eclipse Indigo :

1. *Components*

   Eclipse terdiri dari model komponen-komponen (*plug-ins*) yang dapat digunakan lebih dari satu aplikasi.

2. *Middleware and Infrastructure*

   Komponen utama dalam eclipse dibuat berdasarkan *framework* dan fasilitas, sehingga mempermudah penulisan kode oleh pengguna. Fasilitas yang diperoleh, seperti: paradigma UI yang fleksibel, aplikasi yang dapat diekstensi, *help support*, *scalable* UI, *context-sensitive help*, *network updates*, *error handling*, dan lain-lainnya.

3. *Native User Experience*

   Eclipse Standard Widget Toolkit menyediakan *toolkit* GUI untuk java yang efisien dan akses yang portable ke fasilitas bawaan (*native*) UI di OS.

4. *Portability*

   Eclipse memiliki aplikasi yang fleksibel di berbagai OS dan *client environments* (syaratnya: *hardware* itu dapat diinstal *Java Runtime Environment*).

5. *Intelligent Install and Update*

   Aplikasi di eclipse memiliki fitur *update plugins* melalui HTTP, *Java Web Start, Update Site, copy file,* atau system manajemen perusahaan yang canggih.

6. *Disconnected Operation*

   Tidak membutuhkan koneksi internet, karena aplikasi eclipse berjalan di local komputer.

7. *Development Tooling Support*

   Eclipse menyediakan class pertama di IDE java yang dapat diintegrasikan untuk *develop*, *test*, dan *package* aplikasi *rich clients*.

8. *Component Libraries*

   Komponen di *framework* tidak akan lengkap tanpa set *plugins* yang komprehensif. Sehingga eclipse akan memproduksi *plugins* yang dibutuhkan dalam membuat aplikasi secara utuh.

(http://www.ekowahyudi.com/2009/12/eclipse-rcp-bagian-1.html)

2.4 *Location Based Service* (LBS)

Menurut Nazaruddin Safaat *Location Based Service* (LBS) atau layanan berbasis lokasi adalah istilah umum yang digunakan untuk menggambarkan

teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan (Nazaruddin:2011).

Dua unsur utama LBS adalah:

1. *Location Manager* (API Maps)

   Menyediakan *tools/source* untuk LBS, *Application Programming Interface* (API) Maps menyediakan fasilitas untuk menampilkan, memanipulasi *maps*/peta beserta *feature-feature* lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya. Paket ini berada pada com.google.android.maps

2. *Location Providers* (API Location)

   Menyediakan teknologi pencarian lokasi yang digunakan oleh device/perangkat. API Location berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi real-time. API *Location* berapa pada paket android yaitu dalam paket *android.location.* Dengan Location Manager, kita dapat menentukan lokasi kita saat ini, *Track* gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

2.5 Arsitektur Andorid

Menurut Nazaruddin Syafaat secara garis besar arsitektur android dapat dijelaskan dan digambarkan sebagai berikut (Nazaruddin:2011):

1. *Applications dan Widgets*

   Applications dan *Widgets* ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien *email*, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

2. *Applications Frameworks*

   Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notifications*, dan sebagainya.

3. *Libraries*

   Libraries ini adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, *layer* ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL, serta *libraries* lainnya.

4. *Android Run Time*

   Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. *Dalvik Virtual*

Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android.

5. *Linux Kernel*

   Linux Kernel adalah *layer* dimana inti dari operating sistem dari Android itu berada. Berisi *file-file* sistem yang mengatur *system processing, memory, resource, drivers,* dan system-sistem operasi android lainnya.

2.6 Android Lifecycle

Menurut Nazaruddin Safaat setiap aktifitas yang ada di dalam aplikasi pasti akan melalui *lifecycle.* Misalnya jika menjalankan aplikasi fungsi *onCreate* akan digunakan untuk membuat aplikasi tampil di layar Android, jika keluar dari aplikasi fungsi *OnDestroy* akan digunakan (Nazaruddin:2011).

2.7 Database SQLite

Menurut Nazaruddin Safaat android memiliki fasilitas untuk membuat *database* yang dikenal dengan *SQLite* yaitu salah satu *software* yang *embedded* yang sangat popular, kombinasi *SQL interface* dan penggunaan memory yang sangat sedikit dengan kecepatan yang sangat cepat. *SQLite* di android termasuk dalam *Android runtime,* sehingga setiap versi dari android dapat membuat *database* dengan *SQLite* (Nazaruddin:2011).

Dalam sistem android terdapat beberapa teknik untuk melakukan penyimpanan data. Teknik yang umum digunakan adalah sebagai berikut:

1. *Shared prefences* yaitu menyimpan data beberapa nilai (*value*) dalam bentuk *groups key* yang dikenal dengan *prefences*.
2. *Files* yaitu menyimpan data dalam *file*, dapat berupa menulis ke *file* atau membaca dari file.
3. *SQLite Database*, yaitu menyimpan data dalam bentuk *database.*
4. *Content Providers*, yaitu menyimpan data dalam bentuk *content providers service.*

2.8 Eclipse

Menurut Nazaruddin Safaat eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dijalankan di semua *platform*. Eclipse sendiri juga merupakan sebuah komunitas *open source*, yang memiliki proyek yang berfokus pada membangun sebuah *platform* pengembangan terbuka dari extensible *Framework, tools* dan runtime untuk membangun, menyebarkan dan mengelola perangkat lunak di seluruh siklus hidup perangkat lunak tersebut (Nazaruddin:2011).

Umumnya Eclipse digunakan untuk membuat sebuah program yang menggunakan bahasa pemrograman Java. Namun, Eclipse juga bisa digunakan untuk penggunaan bahasa pemrograman lainnya seperti C, C++, COBOL, Perl, PHP, Python, dan sebagainya.

2.8.1 Versi Eclipse

Berikut ini adalah versi Eclipse yang telah dirilis:

Tabel 2.2 Versi *Eclipse*

Nama	**Tanggal Rilis**	**Versi**
Eclipse 3.0	21 Juni 2004	3.0
Eclipse 3.1	28 Juni 2008	3.1
Callisto	30 Juni 2006	3.2
Nama	**Tanggal Rilis**	**Versi**
Europa	29 Juni 2007	3.3
Ganymade	25 Juni 2008	3.4
Galileo	24 Juni 2009	3.5
Helios	23 Juni 2010	3.6
Indigo	22 Juni 2011	3.7

(Sumber:Nazaruddin. 2011)

2.9 *Extensible Markup Language* (XML)

Extensible Markup Language (XML) adalah bahasa untuk keperluan umum yang disarankan oleh W3C *(World Wide Web Consortium/* suatu konsorsium yang bekerja untuk mengembangkan standar-standar untuk *World Wide Web)* untuk membuat dokumen keperluan pertukaran data antarsistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet .

(http://id.wikipedia.org/wiki/XML/)

XML didesain untuk mempu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data 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.