

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Chat Messenger**

*Chat Messenger* adalah suatu teknologi jaringan yang menjadikan penggunanya mengirimkan pesan secara *real time* ke pengguna lain yang tersambung dalam sebuah LAN atau *Local Area Network* ataupun internet. (Jumadi, 2004)

Pengertian *Chat Messenger* atau *Chating* mengobrol jika diterjemahkan langsung dari bahasa Inggris. Dalam dunia computer dan internet, pengertian *Chat Messenger* adalah suatu fasilitas dalam internet untuk berkomunikasi sesama pengguna internet yang sedang *On-Line*. Komunikasi dapat berupa teks. Sekarang banyak sekali *Software* atau aplikasi yang memfasilitasi pengguna untuk melakukan *Chat Messenger*, misalnya Yahoo Messenger, Google Talk, Skype, dan sebagainya.

#### **II.2. Perancangan**

Menurut Jogiyanto (1999,p179) perancangan mempunyai 2 maksud, yaitu untuk memenuhi kebutuhan kepada pemakai system dan untuk memberikan gambaran yang jelas kepada pemogram komputer dan ahli-ahli teknik lainnya yang terlibat. (Hariadi, 2013 : 2).

### II.3. Aplikasi

Menurut Abdul Kadir (2008:3) Program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu.

Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu. (Rahmatillah, 2011).

### II.4. Informasi

Informasi adalah data yang sudah diolah, dibentuk atau dimanipulasi sesuai dengan keperluan tertentu. Kualitas dari suatu sistem informasi (*quality of information*) tergantung dari 3 hal, yaitu informasi harus akurat (*accurate*), tepat pada waktunya (*timeliness*) dan relevan (*relevance*).

Yang dimaksud dengan akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Sedangkan tepat waktu berarti, informasi yang datang pada penerima tidak boleh terlambat, dan yang terakhir relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya.

## **II.5. Android**

Menurut situs resmi android (<http://www.android.com>) dan Lessard et.al (2010) serta Bharati et.al (2010) Android adalah system 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 standar terbuka pada perangkat seluler. Di lain pihak, google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Terdapat beberapa versi pada sistem operasi android, mulai dari versi 1.5 (CupCake), versi 1.6 (Donut), versi 2.1 (Éclair), versi 2.2 (Froyo), versi 2.3 (GingerBread), versi 3.0 (HoneyCumb), hingga versi yang terbaru yaitu versi 4.0 (Ice Cream Sandwich). (Darma, 2012 : 138-139).

## **II.6. Eclipse**

Menurut Nasruddin Safaat h (Pemrograman aplikasi mobeli smartphome dan tablet PC berbasis android 2012:16) 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:

- a. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- b. Multi-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.
- c. 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.

Eclipse dibuat dari kerja sama antara perusahaan-perusahaan anggota 'Eclipse Foundation' (beserta individu-individu lain). Banyak nama besar yang ikut dalam 'Eclipse Foundation', termasuk IBM, BEA, Intel, Nokia, Borland. Eclipse bersaing langsung dengan Netbeans IDE. Plugin tambahan pada Eclipse jauh lebih banyak dan bervariasi dibandingkan IDE lainnya.

1. IntelliJ IDEA (commercial, free 30-day trial)
2. Oracle JDeveloper (free)
3. Xinox JCreator (ada versi berbayar maupun free)

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US\$ 40 juta untuk pengembangannya. Semenjak itu konsorsium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya. Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah 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 yang terbuka

(open), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstall plug-in yang dibutuhkan.

Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools). Selain itu, 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 secara gratis seperti Eclipse downloads by project.

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous Release. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi.

Hal ini juga dimaksudkan untuk mempermudah Deployment dan maintenance untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya. (Hasmoro, 2014).

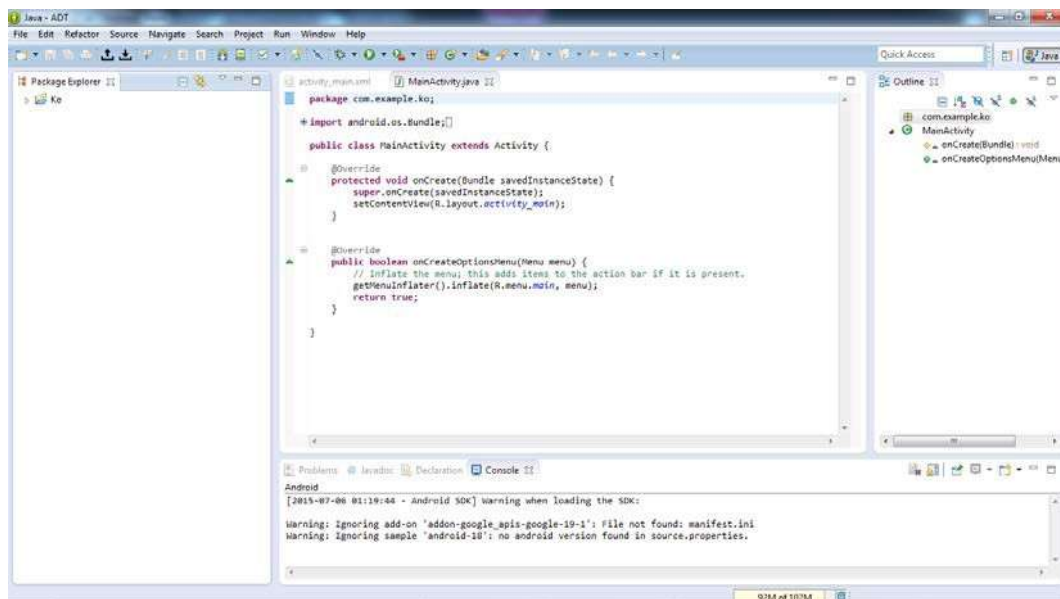
### **II.6.1. Area Kerja *Eclipse***

Saat akan menjalankan program aplikasi *Eclipse*, maka kita akan melihat tampilan awal pembuka *Eclipse* seperti yang terlihat pada gambar II.1.



**Gambar II.1. Tampilan Awal Pembuka *Eclipse***

Setelah proses *loading* pada aplikasi *Eclipse* selesai, maka akan tampil bagian area kerja *Eclipse*. Area kerja *Eclipse* dapat dilihat pada Gambar II.2.



**Gambar II.2. Tampilan Area Kerja *Eclipse***

## II.7. MySQL

MySQL merupakan salah satu DBMS yang gratis dan *open source*, dikembangkan, dan didistribusikan. SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengakses basis data yang tergolong relasional.

SQL sering digunakan sebagai bahasa *query*, dapat melakukan lebih hanya dari sekadar melakukan *query* pada basis data. SQL dapat digunakan untuk mendefinisikan struktur data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga ke pemeliharaan kinerja basis data dan pengaturan pemrosesan pada CPU parallel

Keunggulan dari MySQL adalah :

- a. MySQL merupakan program yang *multithreaded*, sehingga dapat dipasang pada *server* yang memiliki multi-CPU.
- b. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, TCL APIs.
- c. Bekerja pada berbagai *platform*.
- d. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi system *database*.
- e. Memiliki sistem keamanan yang cukup baik dengan verifikasi *host*.
- f. Mendukung ODBC untuk sistem operasi Microsoft Windows.
- g. Mendukung rekaman yang memiliki kolom dengan panjang tetap atau panjang bervariasi.
- h. MySQL merupakan *software* yang *free*. ( Firdayanti, (tt) : 4).

## II.8. UML (*Unified Modelling Language*)


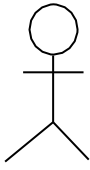


Penjadwalan telah lama diteliti, contohnya dalam penghasilan tenaga oleh *Windu Gata* dan *Grace Gata* (2013), UML (*Unified Modelling Language*) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML (*Unified Modelling Language*) merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

### II.8.1. *UseCase Diagram*

*UseCase Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Simbol-simbol yang digunakan dalam *UseCase Diagram*, yaitu:

**Tabel II.1. *UseCase Diagram***

Gambar	Keterangan
	<i>UseCase</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>UseCase</i> .
	<i>Actor</i> atau Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>usecase</i> , tetapi tidak memiliki kontrol terhadap <i>usecase</i> .
	Asosiasi antara aktor dan <i>usecase</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>usecase</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.

(Sumber : *Windu Gata dan Grace Gata ; 2013*)



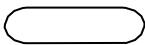
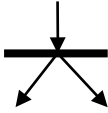

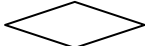
### II.8.2. Class Diagram (Diagram Kelas)

*Class Diagram* merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

### II.8.3. Activity Diagram (Diagram Aktivitas)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu:

**Tabel II.2. Activity Diagram (Diagram Aktivitas)**

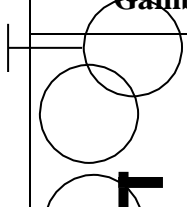
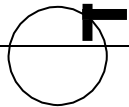
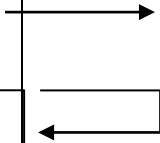


Gambar	Keterangan
	<i>Start Point</i> diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End Point</i> , akhir aktifitas.
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>Rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .



(Sumber : Windu Gata dan Grace Gata ; 2013)

#### II.8.4. *Sequence Diagram* (Diagram Urutan)

*Sequence Diagram* menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu:

**Tabel II.4. *Sequence Diagram* (Diagram Urutan)**

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak.
	<i>Control Class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. <i>Control object</i> mengkoordinir pesan antara <i>boundary</i> dengan entitas.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<p><i>Activation</i>, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Windu Gata dan Grace Gata ; 2013)