

BAB III

LANDASAN TEORI

3.1. Aplikasi

Pengertian aplikasi menurut para ahli tidaklah sulit untuk ditemukan. Istilah aplikasi yang telah menjamur dan semakin banyak digunakan membuatnya menjadi bahan penelitian yang menarik. Istilah ini berasal dari bahasa Inggris "*application*" yang bermakna penerapan, lamaran, atau penggunaan. Bila dimaknai secara istilah, aplikasi adalah suatu program yang siap untuk digunakan oleh suatu sasaran yang akan dituju. Menurut sebuah kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

Sebelum membicarakan mengenai pengertian aplikasi menurut para ahli, ketahui dulu sejarahnya. Awalnya sebelum aplikasi menjadi secanggih sekarang, perangkat lunak menggunakan *aljabar Boolean* yang direpresentasikan sebagai *binary digit* (bit), yaitu 1 (benar/*on*) atau 0 (salah/*off*). Cara ini sangat menyulitkan sehingga orang mulai mengelompokkan bit tersebut menjadi *nible* (4 bit), *byte* (8 bit), *word* (2 *byte*), *double word* (32 bit). Kelompok-kelompok bit ini disusun ke dalam struktur instruksi seperti penyimpanan, transfer, operasi aritmatika, operasi logika, dan bentuk bit ini diubah menjadi kode-kode yang dikenal sebagai

assembler. Kode-kode mesin masih cukup sulit karena tuntutan untuk dapat menghafal kode tersebut dan format penulisannya yang membingungkan. Dari masalah ini kemudian lahir bahasa pemrograman tingkat tinggi (bahasa Inggris). Saat ini pembuatan perangkat lunak sudah menjadi suatu proses produksi yang sangat kompleks, dengan urutan proses yang panjang dengan melibatkan puluhan bahkan ratusan orang dalam pembuatannya.

Sejarah perkembangan aplikasi diatas memang cukup menarik untuk diikuti. Mengetahui perkembangan aplikasi komputer memang tidak akan ada habisnya. Untuk semakin menambah pemahaman, lebih lengkap bila mengetahui definisi aplikasi yang diutarakan oleh para pakar. Berikut ini adalah pengertian aplikasi menurut para ahli.

Menurut Ali Zaki dan *Smitdev Community*, aplikasi adalah komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data.

Menurut Jogiyanto, aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.

Menurut Hengky W. Pramana, aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, *game*, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia.

Menurut Sri Widianti, aplikasi adalah sebuah perangkat lunak yang menjadi *front end* dalam sebuah sistem yang digunakan untuk mengolah

data menjadi suatu informasi yang berguna orang-orang dan sistem yang bersangkutan.

Menurut Harip Santoso, aplikasi adalah suatu kelompok file (*form*, *class*, *report*) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi *payroll*, aplikasi *fixed asset*, dan lain - lain.

Menurut Yuhfizar, aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu.

Menurut Rachmad Hakim S, aplikasi merupakan perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur *Windows* dan permainan (*game*), dan sebagainya.

Menurut R. Eko I. dan Djokopran, aplikasi merupakan proses atau prosedur aliran data dalam infrastruktur teknologi informasi yang dapat dimanfaatkan oleh para pengambil keputusan yang sesuai dengan jenjang dan kebutuhan (relevan).

3.2. Media Pembelajaran

Pembelajaran disekolah pada saat ini mulai disesuaikan dengan perkembangan teknologi informasi, sehingga terjadi perubahan dan pergeseran paradigma pendidikan. Perkembangan pesat dibidang teknologi khususnya internet, mempercepat aliran ilmu pengetahuan yang menembus batas-batas dimensi ruang, birokrasi, kemapanan, dan waktu. Program-program di internet bukan hanya menampilkan data dan informasi yang

dapat ditransmisikan dengan kecepatan tinggi, tetapi juga ilmu pengetahuan yang dapat diakses secara cepat oleh penggunanya dan tentu saja kondisi ini berpengaruh pada kebiasaan dan budaya pendidikan yang dikelola selama ini.

Kemajuan dan perkembangan teknologi sudah demikian menonjol, sehingga penggunaan alat-alat bantu mengajar seperti alat-alat audio, visual serta perlengkapan sekolah disesuaikan dengan perkembangan zaman tersebut. Dan juga harus disesuaikan dengan tuntutan kurikulum sesuai dengan materi, metode, dan tingkat kemampuan belajar siswa agar dapat mencapai tujuan pembelajaran dengan baik disekolah.

Untuk itu, para pengajar mulai berusaha membiasakan diri untuk menggunakan peralatan-peralatan seperti OHP, LCD, CD, VCD, video, komputer dan internet dalam pembelajaran dikelas, dengan program pembelajaran yang dikembangkan ini patut dipelajari pengajar harus mempelajarinya agar mempermudah proses pembelajaran dan pendidikan, sehingga memudahkan pembelajaran untuk berjalan dengan baik dikelas.

3.2.1. Pengertian Media Pembelajaran

Media adalah sebuah alat yang mempunyai fungsi menyampaikan pesan (Bovee, 1997). Media pembelajaran adalah sebuah alat yang berfungsi dan digunakan untuk menyampaikan pesan pembelajaran. Pembelajaran adalah proses komunikasi antara pembelajar, pengajar, dan bahan ajar. Banyak batasan atau pengertian yang dikemukakan para ahli tentang media, diantaranya

adalah: Asosiasi Teknologi dan Komunikasi Pendidikan (*Association of Education and Communication Technology* (AECT)).

Dari pengertian diatas, secara umum dapat dikatakan bahwa substansi dari media pembelajaran adalah bentuk saluran, yang digunakan untuk menyalurkan pesan, informasi atau bahan pelajaran kepada penerima pesan atau pembelajar dapat pula dikatakan bahwa media pembelajaran adalah berbagai jenis komponen dalam lingkungan dalam lingkungan pembelajar yang dapat merangsang pembelajar untuk belajar.

3.2.2. Tujuan dan Manfaat Media Pembelajaran

1. Tujuan Media Pembelajaran

Tujuan media pembelajaran sebagai alat bantu pembelajaran, adalah sebagai berikut:

- a. Mempermudah proses pembelajaran dikelas
- b. Meningkatkan efisiensi proses pembelajaran
- c. Menjaga relevansi antara materi pelajaran dengan tujuan belajar
- d. Membantu konsentrasi pembelajar dalam proses pembelajaran

2. Manfaat Media Pembelajaran

Manfaat media pembelajaran sebagai alat bantu dalam proses pembelajaran adalah sebagai berikut:

- a. Pengajaran lebih menarik perhatian pembelajar sehingga dapat menumbuhkan motivasi belajar
- b. Bahan pengajaran akan lebih jelas maknanya, sehingga dapat menumbuhkan motivasi belajar
- c. Metode pembelajaran bervariasi, tidak semata-mata hanya komunikasi verbal melalui penuturan kata-kata lisan pengajar, pembelajar tidak bosan, dan pengajar tidak kehabisan tenaga
- d. Pembelajar lebih banyak melakukan kegiatan belajar, sebab tidak hanya mendengarkan penjelasan dari pengajar saja, tetapi juga aktivitas lain yang dilakukan seperti mengamati, melakukan, mendemonstrasikan dan lain-lainnya.

3.2.3. Pertimbangan Pemilihan Media

Pertimbangan media yang akan digunakan dalam pembelajaran menjadi pertimbangan utama, karena media yang dipilih harus sesuai dengan:

1. Tujuan pengajaran
2. Bahan pelajaran
3. Metode mengajar
4. Alat yang dibutuhkan
5. Pribadi mengajar
6. Minat dan kemampuan mengajar

Keterkaitan antara media pembelajaran dengan tujuan, materi, metode, dan kondisi pembelajar, harus menjadi perhatian dan pertimbangan pengajar untuk memilih dan menggunakan media dalam proses pembelajaran dikelas, sehingga media yang digunakan lebih efektif dan efisien untuk mencapai tujuan pembelajaran. Sebab media pembelajaran tidak dapat berdiri sendiri, tetapi terkait dan memiliki hubungan secara timbal balik dengan empat aspek tersebut. Dengan demikian, alat-alat, sarana, atau media pembelajaran yang digunakan harus disesuaikan dengan empat aspek tersebut, untuk mencapai tujuan pembelajaran secara efektif dan efisien.

3.2.4. Fungsi Media Pembelajaran

Media pembelajaran berfungsi untuk merangsang pembelajaran dengan:

1. Menghadirkan obyek sebenarnya dan obyek yang langka
2. Membuat duplikasi dari obyek yang sebenarnya
3. Membuat konsep abstrak ke konsep konkret
4. Memberi kesamaan persepsi
5. Mengatasi hambatan waktu, tempat, jumlah, dan jarak
6. Menyajikan ulang informasi secara konsisten
7. Memberi suasana belajar yang tidak tertekan, santai, dan menarik.

3.3. Android

3.3.1. Pengertian Android

Android adalah suatu sistem operasi yang berjalan pada *smartphone* saat ini dan menyesuaikan spesifikasi di kelas *low-end* hingga *high-end*. Hampir semua *vendor* saat ini mengembangkan produknya dengan sistem operasi *Android*, karena peminatnya yang semakin meningkat tajam.



Gambar 3.1 Logo Android

3.3.2. Sejarah Android

Android adalah sistem operasi berbasis *kernel Linux* yang pada awalnya dikembangkan oleh *Android, Inc*, yang didukung *Google financial* dan kemudian dibeli pada tahun 2005.

Android ini diresmikan pada tahun 2007 seiring dengan berdirinya *Open Handset Alliance* konsorsium *hardware, software*, dan perusahaan telekomunikasi yang ditujukan untuk memajukan standar perangkat seluler. *Smartphone* yang tersedia untuk publik pertama kalinya yang menjalankan *Android* adalah *HTC Dream*, yang dirilis pada 22 Oktober 2008.

3.3.3. Perkembangan Android

Perkembangan android dari awal hingga sekarang ini sudah mencapai 19 kali, yaitu:

- 2.1 Android 1.0
- 2.2 Android 1.1
- 2.3 Android 1.5 Cupcake
- 2.4 Android 1.6 Donut
- 2.5 Android 2.0 Eclair
- 2.6 Android 2.0.1 Eclair
- 2.7 Android 2.1 Eclair
- 2.8 Android 2.2-2.2.3 Froyo
- 2.9 Android 2.3-2.3.2 Gingerbread
- 2.10 Android 2.3.3-2.3.7 Gingerbread
- 2.11 Android 3.0 Honeycomb
- 2.12 Android 3.1 Honeycomb
- 2.13 Android 3.2 Honeycomb
- 2.14 Android 4.0-4.0.2 Ice Cream Sandwich
- 2.15 Android 4.0.3-4.0.4 Ice Cream Sandwich
- 2.16 Android 4.1 Jelly Bean
- 2.17 Android 4.2 Jelly Bean
- 2.18 Android 4.3 Jelly Bean
- 2.19 Android 4.4 Jelly Bean

Bukan tidak mungkin jika android menguasai pasaran saat ini, karena *developer* atau pengembang android sendiri terus berupaya dalam hal peningkatan kinerja android dengan performa yang maksimal.

3.3.4. Kelebihan atau Keunggulan Android

1. *User Friendly* Kalimat ini sangat melekat pada system operasi Windows miliknya Microsoft, ibaratnya dengan sangat mudah mengoperasikan komputer hanya dengan belajar beberapa hari bahkan beberapa jam saja, dan ini juga melekat pada Android yang berjalan pada *Smartphone*.
2. *Notification* dengan sangat mudah mendapatkan notifikasi dari *smartphone* android dengan mengatur beberapa akun *Email, SMS, Voice, Dial, Update* dan lain sebagainya.
3. Tampilan android tidak kalah bagusnya dari iOS miliknya Apple, karena memang dari awal android hampir mengungung teknologi iOS, hanya saja ini versi murahannya.
4. *Open Source Operating system* ini memang dibuat *open source* oleh penciptanya, karena memang berbasis *kernel Linux*. Jadi jangan kaget jika diluar sana ada banyak sekali *Custom Rom* untuk masing-masing perangkat *android*.
5. Aplikasi akan disajikan jutaan pilihan aplikasi yang menarik dari yang gratis hingga berbayar, dan bisa mendownloadnya di *Google Play*.

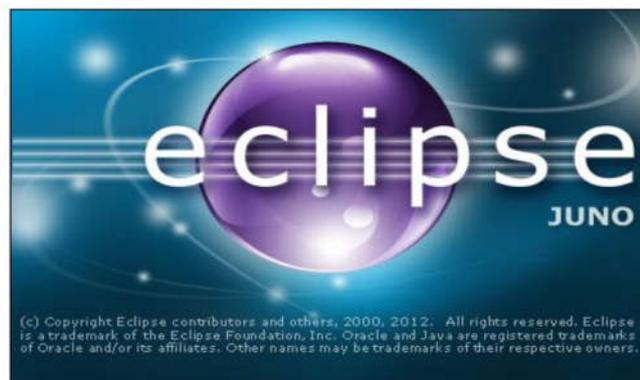
6. Dan banyak lagi banyak kelebihan dan fitur yang dimiliki Android, bisa mengembangkannya sesuai dengan keinginan anda, karena Sistem operasi ini *Open Source* dan *User Friendly*.

3.3.5. Kekurangan atau Kelemahan Android

1. *Update System* untuk melakukan *update system* tidaklah mudah, harus menunggu dari masing-masing *Vendor* untuk merilis Update Versi yang terbaru. Tentu ini sangat tidak menyenangkan.
2. Baterai Cepat Habis Ini sering terjadi jika menyalakan paket data dan menggunakan *widget* serta aplikasi yang berjalan secara berlebihan, untuk mengatasinya harus mengurangi aktivitas aplikasi pada *smartphone*.
3. Lemot atau *Lag* Sebenarnya kalau ini ada kaitnya dengan spesifikasi dari masing-masing perangkat, namun ada kalanya Android ini tidak bersahabat dengan beberapa aplikasi dikarenakan Ram ataupun Prosesornya yang kurang memadai, jadi sesuaikan aplikasi sesuai dengan perangkatnya.

3.4. IDE Eclipse

IDE adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Dengan menggunakan Java IDE (*Integred Development Environment*) tertentu, semua kebutuhan pemrograman akan dijadikan menjadi satu tempat. Mulai *text editor*, *compiler/interpreter*, *system help* dan terkadang juga terdapat fitur lain yang sangat bermanfaat dalam penulisan kode (seperti: *code auto-complete* dan *syntax highlight*). Contoh dari aplikasi IDE adalah Eclipse.



Gambar 3.2 Logo Eclipse Juno

Eclipse adalah sebuah IDE (*Integred 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 OSX*.
- b. *Multilanguage*: 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 merupakan komunitas *open source* yang bertujuan menghasilkan platform pemrograman terbuka. Eclipse terdiri dari framework yang dapat dikembangkan lebih lanjut, peralatan bantu untuk membuat dan memmanage *software* sejak awal hingga diluncurkan. Platform Eclipse didukung oleh ekosistem besar yang terdiri dari vendor teknologi, *strat-up* inovatif, universitas, riset intitusi serta individu. Banyak orang mengenal Eclipse sebagai IDE (*Integred Development Environment*) untuk bahasa Java, tapi Eclipse lebih dari sekedar IDE untuk Java. Komunitas Eclipse memiliki lebih dari 60 proyek *open source*. Proyek-proyek ini secara konsep terbagi menjadi 7 kategori:

- *Enterprise Development*
- *Embedded and Device Development*
- *Rich Client Platform*
- *Rich Internet Aplication*

- *Application Framework*
- *Application Lifecycle Management (ALM)*
- *Service Oriented Architecture (SOA)*

Secara umum Eclipse digunakan untuk membangun *software* inovatif berstandar industri dan alat bantu beserta *framework* nya membantu pekerjaan menjadi lebih mudah. Lisensi Eclipse menggunakan EPL (*Eclipse Public License*), yaitu lisensi yang memungkinkan organisasi untuk menjadikan Eclipse sebagai produk komersialnya, dan pada saat yang sama meminta orang yang melakukan perubahan untuk mengkontribusikan hasilnya kembali kepada komunitas.

3.4.1. Android SDK

Android SDK merupakan *tool* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan tutorial. Saat ini android sudah mendukung arsitektur x86 pada *Linux* (distribusi *Linux* apapun untuk desktop modern), *Mac OS X 10.4.8* atau lebih, *Windows XP* atau *Vista*. Persyaratan mencakup *JDK*, *Apache Ant* dan *Python 2.2* atau yang lebih baru. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan *plugin Android Development Tools* (ADT), dengan ini pengembang dapat menggunakan teks editor untuk mengedit *file* Java dan *XML* serta menggunakan peralatan

Command Line untuk menciptakan, membangun, melakukan *debug* aplikasi Android dan pengendalian perangkat Android (misalnya, *reboot*, menginstal paket perangkat lunak dengan jarak jauh).

Android SDK telah dirilis pada tanggal 12 November 2007. Dan pada tanggal 15 Juli 2008 tim *Android Developer Challenge* sengaja mengirimkan *email* ke semua pendaftar di *Android Developer Challenge* untuk mengumumkan bahwa rilis SDK terbaru telah tersedia pada halaman download pribadi. *Email* tersebut juga ditujukan kepada pemenang *Android Developer Challenge* putaran pertama. Sebuah pernyataan bahwa Google telah menyediakan rilis SDK terbaru untuk beberapa pengembang dan bukan untuk orang lain.

Pada tanggal 18 Agustus 2008, Android SDK 0,9 beta dirilis. Rilis ini menyediakan API yang diperbarui dan diperluas, perbaikan pada alat-alat pengembangan dan desain terbaru untuk layar awal. Petunjuk untuk meng-upgrade SDK sudah tersedia pada rilis sebelumnya. Pada tanggal 23 September 2008, Android 1.0 SDK telah dirilis. Pada tanggal 9 Maret 2009, Google merilis versi 1.1 untuk telepon seluler Android. Rilis terbaru tersebut termasuk dukungan untuk pencarian dengan suara, harga aplikasi, perbaikan jam alarm, perbaikan pengiriman gmail, perbaikan surat pemberitahuan dan peta.

Pada pertengahan Mei 2009, Google merilis versi 1.5 (Cupcake) pada sistem operasi Android dan SDK. Pembaruan ini termasuk banyak fitur baru seperti perekaman video, dukungan untuk *Bluetooth*, *system keyboard* pada layar dan pengenalan suara. Rilis ini juga membuka *AppWidget framework* kepada para pengembang yang memungkinkan orang untuk membuat *widget* sendiri pada halaman *home*. Pada September 2009 versi 1.6 (*Donut*) dirilis yang menampilkan hasil pencarian yang lebih baik dan penggunaan indikator baterai.

Aplikasi Android dipaketkan ke dalam format.apk dan disimpan pada *folder / data/ app*. Pengguna dapat menjalankan perintah *adb root* untuk mengakses *folder* tersebut karena *root* memiliki izin untuk mengakses *folder* tersebut.

3.4.2. Java Deveopment Kit (JDK)

Bahasa pemrograman java sangat populer dikalangan programmer dunia. Memiliki beberapa keunggulan yang tak dimiliki bahasa pemrograman lain membuat java semakin kokoh menduduki peringkat teratas sebagai bahasa pemrograman terpopuler di dunia. Java dapat dijalankan pada berbagai *device computer* dan *smartphone*. Bahasa ini banyak mengadopsi sintak dan beberapa bahasa c dan c++, namun dengan sintak model objek yang lebih sederhana.

Java Development Kit atau biasa disingkat JDK merupakan *software* yang digunakan untuk melakukan proses kompilasi dari *java code* ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh JRE (*Java Runtime Envirotment*). JDK berisi sekumpulan *command line tool* untuk menciptakan program java. JDK harus terinstall pada komputer yang akan melakukan pembuatan aplikasi berbasis java, namun tidak harus terinstall di komputer yang akan menjalankan aplikasi berbasis java. Sekarang, JDK lebih mengacu pada *Java Standar Edition* atau biasa disingkat Java SE.

Berikut adalah beberapa komponen utama JDK:

1. *Kompulator (Javac)*
2. *Interpreter program java (Java)*
3. *Applet viewer (appletviewer)*
4. *Debugger (jdb)*
5. *Class file disassemble (javap)*
6. *Java Archive (jar)*
7. *Documentation generator (javadoc)*
8. *Applet demo*
9. *Kode sumber Java API*

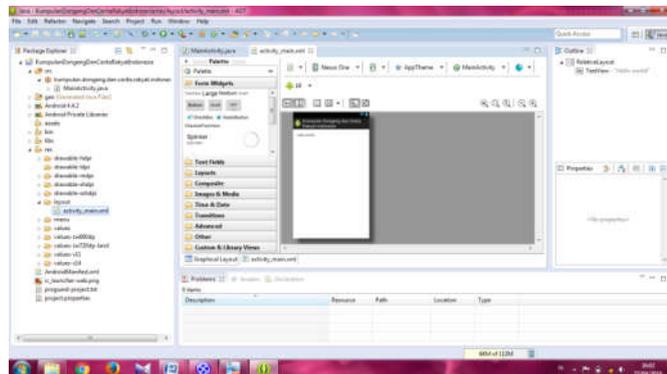
3.4.3. ADT (Android Development Tools)

Android Development Tools (ADT) adalah *plug-in* yang didesain untuk IDE *Eclipse* yang memberikan kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE

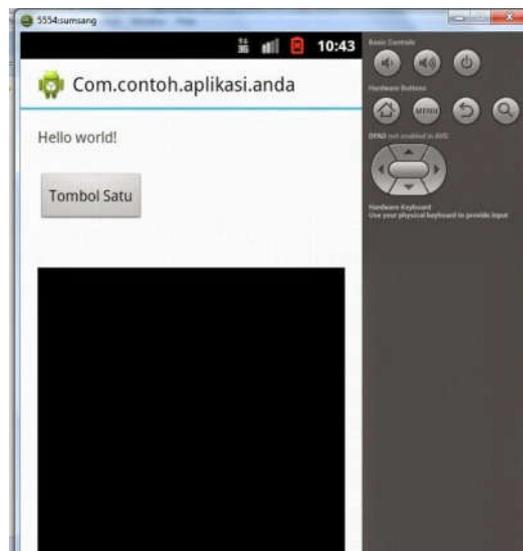
Eclipse. Dengan menggunakan ADT untuk *Eclipse* akan memudahkan dalam membuat aplikasi *project* android, membuat GUI, dan menambahkan komponen-komponen yang lainnya, begitu juga dapat melakukan *running* aplikasi menggunakan Android SDK melalui *Eclipse*. Dengan ADT juga dapat melakukan *package* android (.apk) yang digunakan untuk distribusi aplikasi android yang dirancang.



Gambar 3.3 Tampilan Awal Software Android Eclipse



Gambar 3.4 Area Kerja Eclipse



Gambar 3.5 Emulator Android Eclipse

3.5. UML (Unified Modelling Language)

Unified Modeling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah *system software* yang terkait dengan objek (Whitten L. Jeffery et al, 2004).

Henderi (2007: 4) Unified Modelling Language (UML) adalah sebuah bahasa pemodelan yang telah menjadi standar dalam *industry software* untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Bahasa Pemodelan UML lebih cocok untuk pembuatan perangkat lunak dalam bahasa pemrograman berorientasi objek (C+, Java, VB.NET), namun demikian tetap dapat digunakan pada bahasa pemrograman procedural.

3.5.1. Use Case Diagram

Use Case Diagram adalah model fungsional sebuah sistem yang menggunakan actor dan use case. Use Case adalah layanan (*services*) atau fungsi-fungsi yang disediakan oleh sistem untuk pengguna-penggunanya (Henderi et al, 2008). Use Case adalah suatu pola atau gambaran yang menunjukkan kelakuan atau kebiasaan sistem. Setiap Use Case adalah suatu urutan (*sequence*)

transaksi yang saling berhubungan dan dilakukan oleh sebuah *actor* dan sistem dalam bentuk sebuah dialog (Henderi, 2007). Use Case Diagram dibuat untuk memvisualisasikan atau menggambarkan hubungan antara *Actor* dan Use Case. Use Case Diagram mempresentasikan kegunaan atau fungsi-fungsi sistem dari pengguna.

Tabel 3.1 Simbol Use Case Diagram

No	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

No	GAMBAR	NAMA	KETERANGAN
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

3.5.2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain

Tabel 3.2 Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).

NO	GAMBAR	NAMA	KETERANGAN
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

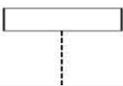
3.5.3. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan

perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan

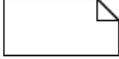
Tabel 3.3 Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
4		<i>Actor</i>	Dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebaai kolom.
5		<i>Control</i>	Berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, proses terdistribusi, dan penanganan kesalahan.
6		<i>Entity</i>	Digunakan menangani informasi yang mungkin akan disimpan secara pemanen. <i>Entity</i> bisa juga merupakan sebuah tabel pada struktur basis data.

3.5.4. State Chart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya). Pada umumnya state chart diagram menggambarkan cliass tertentu (satu *class* dapat memiliki lebih dari satu statechart diagram).

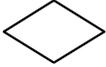
Tabel 3.4 Simbol State Chart Diagram

No	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

3.5.5. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Tabel 3.5 Simbol Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6		<i>Decision</i>	Pilihan untuk mengambil keputusan
		<i>Fork/Join</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
		<i>Rake</i>	Menunjukkan adanya dekomposisi.
		<i>Time</i>	Tanda waktu
		<i>Send</i>	Tanda pengiriman