BAB 2
TINJAUAN PUSTAKA

2.1 Profil SDN Angkasa III

SDN Angkasa III beralamat di Jalan Albatros Blok D RT 02 RW 04 Kelurahan Sulaiman Kecamatan Margahayu Kabupaten Bandung Jawa Barat. Sekolah ini berstatus sekolah negeri dan berakreditasi B. SDN Angkasa III berdiri pada tahun 1978 yang berstatus bangunan milik pemerintah. Sekolah ini memiliki 12 kelas, dimana siswanya berjumlah 242 orang. Memiliki beberapa kantin, lapangan, beberapa kamar mandi, dan sebuah ruang guru. Disekolah ini media peraga mata pelajaran IPA tidak cukup banyak.

2.1.1 Visi Dan Misi

. Misi

Peserta Didik dan Personel yang :

- Iman dan Taqwa
- Cerdas
- Terampil
- Cinta Budaya
- Dan Profesional

b. Visi

1. Menumbuhkembangkan peserta didik yang taat beragama.
2. Mewujudkan tercapainya mutu pendidikan yang sesuai dengan tuntutan masyarakat dan perkembangan IPTEK.
3. Menciptakan anak didik berprestasi
4. Membudayakan hidup bersih, cinta lingkungan dan cinta almamater
5. Menumbuhkembangkan profesionalisme dan meningkatkan kinerja tenaga kependidikan
6. Memantapkan pelaksanaan tata tertib bagi siswa dan personel
7. Meningkatkan hubungan dengan orang tua siswa dan masyarakat sekitar untuk mencapai tujuan pendidikan.

2.2 Ilmu Pengetahuan Alam

Dalam bagian ini akan dibahas tentang pengertian ilmu pengetahuan alam dan pembelajaran ilmu pengetahuan alam di sekolah dasar.

2.2.1 Pengertian Ilmu Pengetahuan Alam

Ilmu pengetahuan alam adalah mata pelajaran yang berkaitan dengan mengetahui alam secara sistematis. IPA bukan hanya kumpulan-kumpulan pengetahuan yang berupa fakta-fakta, atau prinsip-prinsip saja, tetapi juga merupakan suatu proses penemuan (Devi, 2009:1). IPA adalah ilmu pengetahuan yang mempunyai objek menggunakan metode ilmiah sehingga perlu diajarkan di Sekolah Dasar [8].

2.2.2 Pembelajaran Ilmu Pengetahuan Alam di Sekolah Dasar

IPA diperlukan dalam kehidupan sehari-hari untuk memenuhi kebutuhan manusia melalui pemecahan masalah-masalah yang dapat diidentifikasikan. Penerapan IPA perlu dilakukan secara bijaksana agar tidak berdampak buruk terhadap lingkungan. Di tingkat Sekolah Dasar diharapkan ada penekanan pembelajaran Salingtemas (Sains, lingkungan, teknologi, dan masyarakat) yang diarahkan pada pengalaman belajar untuk merancang dan membuat suatu karya melalui penerapan konsep IPA.

Ada berbagai alasan yang menyebabkan satu mata pelajaran itu dimasuk ke dalam kurikulum suatu sekolah. Usman Samatowa (2006) menegemukakan empat Alasan IPA dimasukan dikurikulum Sekolah Dasar yaitu:

a. Bahwa IPA berfaedah bagi suatu bangsa, sebab IPA merupakan dasar teknologi, sering disebut-sebut sebagai tulang punggung pembangunan. Pengetahuan dasar untuk teknologi ialah IPA. Orang tidak menjadi Insinyur elektronika yang baik, atau dokter yang baik, tanpa dasar yang cukup luas mengenai berbagai gejala alam.
b. Bila IPA diajarkan menurut cara yang tepat, maka IPA merupakan suatu mata pelajaran yang memberikan kesempatan berpikir kritis; misalnya sains diajarkan dengan mengikuti metode "menemukan sendiri". Dengan ini anak dihadapkan pada suatu masalah; umpamanya dapat dikemukakan suatu

masalah demikian". Dapatkah tumbuhan hidup tanpa daun?" Anak diminta untuk mencari dan menyelidiki hal ini.

c. Bila IPA diajarkan melalui percobaan-percobaan yang dilakukan sendiri oleh anak. Maka IPA tidaklah merupakan mata pelajaran yang bersifat hafalan belaka.
d. Mata pelajaran ini mempunyai: nilai-nilai pendidikan yaitu mempunyai potensi yang dapat membentuk kepribadian anak secara keseluruhan.

2.3 Edukasi

Edukasi atau pendidikan ialah suatu usaha yang sadar yang teratur dan sitematis, yang dilakukan oleh orang – orang yang diserahi tanggung jawab untuk mempengaruhi anak agar mempunyai sifat dan tabiat yang sesuai dengan cita – cita pendidikan.Edukasi juga bisa diartikan sebagai bantuan yang diberikan secara sengaja kepada anak dalam pertumbuhan jasmani maupun rohani.

Secara teoritis pengertian mendidik dan mengajar tidaklah sama. Mengajar berarti menyerahkan atau manyampaikan ilmu pengetahuan atau keterampilandan lain sebagainya kepada orang lain, dengan menggunakan cara – cara tertentu sehingga ilmu – ilmu tersebut bisa menjadi milik orang lain.

Lain halnya mendidik, bahwa mendidik tidak hanya cukup dengan hanya memberikan ilmu pengetahuan ataupun keterampilan, melainkan juga harus ditanamkan pada anak didik nilai – nilai dan norma – norma susila yang tinggi dan luhur. Dari pengertian diatas diketahui bahwa mendidik lebih luas dari pada mengajar.Mengajar hanyalah alat atau sarana dalam mendidik .dan mendidik harus mempunyai tujuan dan nilai – nilai yang tinggi.

2.3.1 Unsur-Unsur Edukasi

Unsur-unsur edukasi diantaranya [11] :

1. Anak didik, yaitu pihak yang menjadi objek utama edukasi.
2. Pendidik, yaitupihak yang menjadi subjek dari pelaksanaan edukasi.
3. Materi, yaitubahan atau pengalaman belajar yang disusun menjadi kurikulum.
4. Alat edukasi, yaitu tindakan yang menjadi kelangsungan mendidik.
5. Lingkungan, yaitukeadaan yang berpengaruh terhadap hasil edukasi.

6. Dasar dan landasan edukasi, yaitulandasan yang menjadi fundamental dari segala kegiatan edukasi.

2.3.2 Faktor-faktor yang membatasi kemampuan edukasi

Adapun faktor-faktor yang membatasi kemampuan edukasi diantaranya :

1. Faktor anak

   Di dalam anak didik terdapat potensi-potensi yang butuh edukasi dari luar.

2. Faktor Pendidik

   Guru mempunyai metode penyampaian yang berbeda dan beragam.

3. Faktor Lingkungan

   Lingkungan sangat berpengaruh baik positif maupun negatif.

2.4 Aplikasi

Dalam bagian ini akan dibahas mengenai pengertian aplikasi, klasifikasi aplikasi berdasarkan kegunaannya, dan karakteristik aplikasi *mobile*.

2.4.1 Pengertian Aplikasi

Aplikasi menurut Jogiyanto *(2001:12)* adalah penggunaan dalam suatu komputer,instruksi *(instructiom)* atau pernyataan *(statement)* yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*. Menurut (shelly, cashman, vermant, 2009, p57), Aplikasi adalah seperangkat instruksi khusus dalam komputer yang dirancang agar kita menyelesaikan tugas-tugas tertentu.

Mobile dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat lain. Aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti *pager*, PDA, dan telepon seluler.

2.4.2 Klasifikasi Aplikasi Berdasarkan Kegunaan

Adapun klasifikasi aplikasi berdasarkan kegunaannya sebagai berikut [3] :

1. Aplikasi perusahaan (enterprise)
2. Aplikasi infrastruktur perusahaan
3. Aplikasi informasi kerja
4. Aplikasi media dan hiburan

5. Aplikasi pendidikan
6. Aplikasi pengembangan media
7. Aplikasi produk

2.4.3 Karakteristik Aplikasi *Mobile*

Aplikasi mobile memiliki beberapa karakteristik, diantaranya :

1. Ukuran yang kecil, perangkat mobile memiliki ukuran yang kecil.
2. Memori yang terbatas, perangkat mobile juga memiliki memory yang kecil, yaitu primary (RAM) dan secondary (disk).
3. Daya proses yang terbatas, sistem mobile tidaklah setangguh 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. Konektifitas yang terbatas, perangkat mobile memiliki bandwidth rendah, bahkan beberapa dari mereka bahkan tidak tersambung.
7. Masa hidup yang pendek, perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

2.5 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untk menciptakan aplikasi mereka[12].

2.5.1 Sejarah Android

Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. 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 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.

Sekitar September 2007, Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communication, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc.

Pada saat ini kebanyakan vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, Acer, Philips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi vendor *smartphone* di dunia yang memproduksi android. Hal ini karena android itu adalah sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

2.5.2 Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

1. *Applications* dan *Widgets*

Applications dan *Widgets* ini adalah *layer* di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut.

2. *Applications Frameworks*

Applications Frameworks adalah *layer* di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telepon.

3. *Libraries*

Libraries adalah *layer* di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya.

4. *Android Run Time*

Android Run Time adalah *layer* yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan Implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android.

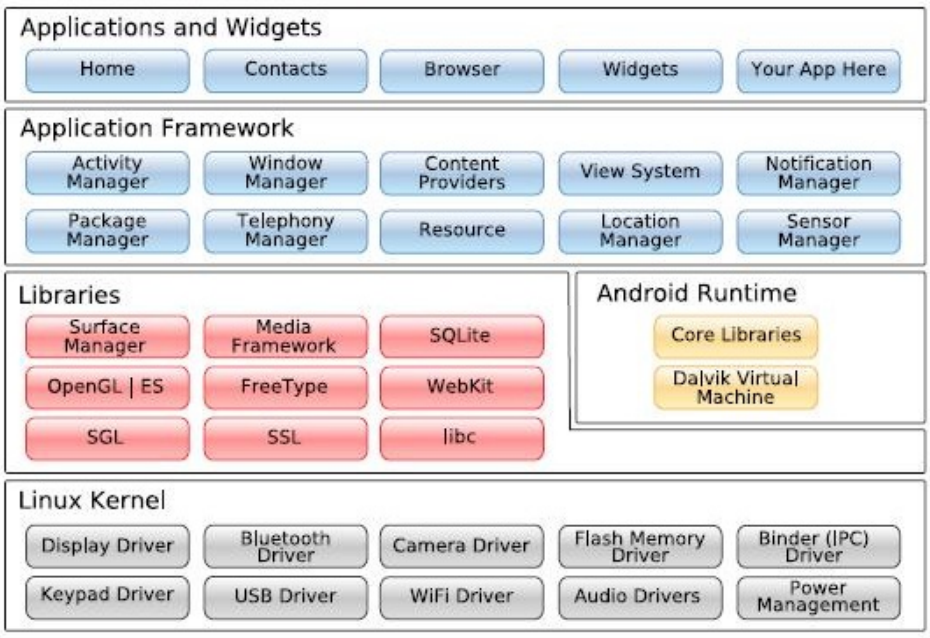
Di dalam *Android Run Time* dibagi menjadi dua bagian yaitu :

a. *Core Libraries*, Aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh *Core Libraries*.

b. *Dalvik Virtual Machine*, Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, di mana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

5. *Linux Kernel*

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



Gambar 2.1 Arsitektur Android [12]

2.5.3 Fitur Android

Beberapa fitur Android yang paling penting adalah [12] :

1. *Framework*, aplikasi yang mendukung penggantian komponen dan *reusable*.
2. Mesin Virtual Dalvik, dioptimalkan untuk perangkat *mobile*.
3. *Intergrated Browser*, berdasarkan engine open source WebKit.
4. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi *hardware*).
5. SQLite, untuk penyimpanan data (database).
6. Media Support yang mendukung audio, video, dan gambar (MPEG4, MP3, AAC, AMR, JPG, PNG, GIF). (tergantung hardware).
7. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware).
8. Kamera, GPS, kompas, dan i (tergantung hardware).
9. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat emulator, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE Eclipse.

2.5.4 Versi Android

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut [1] :

1. Android versi 1.1

   Dirilis pada 9 Maret 2009, versi ini dilengkapi dengan pembaruan estetis pada aplikasi jam, alarm, *voice search*, pengiriman pesan dengan *Gmail*, dan pemberitahuan *email*.
2. Android Versi 1.5 (cupcake)

   Dirilis pada pertengahan Mei 2009, adanya penambahan fitur seperti kemampuan merekam dan menonton video dengan modus kamera, meng-upload video ke *YouTube* dan gambar *Picasa* langsung dari telepon, animasi layar dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.
3. Android versi 1.6 (donut)

   Dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet

VPN. Fitur lainnya adalah kemampuan dial kontak, teknologi *text to change speech* (tergantung ponsel).

4. Android versi 2.0/2.1 (eclair)

   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 baru, dukungan *flash* untuk kamera 3.2MP, *digital zoom*, dan bluetooth 2.1.

5. Android versi 2.2 (froyo)

   Dirilis pada Mei 2010, dengan fitur kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia, Dalvik Virtual Machine dioptimalkan untuk perangkat mobile, grafik di 2D dan grafis 3D berdasarkan libraries opengl, SQLite sebagai penyimpanan data.

6. Android versi 2.3 (gingerbread)

   Dirilis pada Desember 2010, memiliki kemampuan seperti : *SIP-based VoIP*, *Near Field Communication* (NFC), *Gyroscope* dan sensor, *Multiple cameras support, Mixable audio effects, Download manager*.

7. Android versi 3.2 (honeycomb)

   Dirilis pada 22 Februari 2011[2] , terdiri dari beberapa fitur baru diantaranya : *New UI framework for creating great tablet apps, High-performance 2D and 3D graphics, Support for multicore processor architectures, Rich multimedia and connectivity, Enhancements for enterprise, Compatibility with existing apps*[1].

8. Android versi 4.0 (ice cream sandwich)

   Dirilis pada 19 Oktober 2012[2], memiliki beberapa fitur seperti : *Refined, evolved UI, Multitasking, Home screen folders and favorites tray, Resizable widgets, New lock screen actions, Quick responses for incoming calls, Improved text input and spell-checking, Powerful voice input engine, Control over network data, Live Effects for transforming video*[1].

9. Android versi 4.1/4.2/4.3 (jelly bean)

   Dirilis pada 9 Juli 2013[2], memiliki fitur seperti : *Faster, Smoother, More Responsive, OpenGL ES 3.0 for High-Performance Graphics, Enhanced*

Bluetooth Connectivity, Support for Restricted Profiles, Optimized Location and Sensor Capabilities, Modular DRM framework, VP8 encoder, Video encoding from a surface, Media muxer, Playback progress and scrubbing in remote control clients, Support for International Users[1].

2.5.5 Kelebihan dan Kekurangan Android

Kelebihan dan kekurangan sistem operasi Android adalah sebagai berikut:

1. Kelebihan Android [12]
   a. Lengkap (*Complete Platform*): para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyudiakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.
   b. Terbuka (*Open Source)*: *Platform* Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi.
   c. Free (*Free Platform*): Android adalah *platform*/aplikasi yang bebas untuk dikembangkan. Tidak ada lisesnsi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.
2. Kekurangan Android

a. Kurangnya ketersediaan

Berdasarkan peringkat harian agregat pada semester pertama di tahun ini, dari 50 aplikasi terlaris iPad baik yang berbayar maupun gratis di AppStore rupanya sebesar 30 persen diantaranya tidak tersedia di Google Play.

b. Ancaman dari malware/virus

Karena android adalah open source dan pengguna dapat menginstall aplikasi yang bukan berasal dari Google Play, perangkat Android jauh lebih rentan terkena malware dan aplikasi bajakan palsu.

c. Sering terdapat iklan

Karena mudah dan gratis, kadang sering diboncengi iklan. Secara tampilan memang tidak mengganggu kinerja aplikasi itu sendiri, karena memang kadang berada di bagian atas atau bawah aplikasi

d. Terhubung dengan internet

Android bisa dibilang sangat memerlukan koneksi internet yang aktif. Setidaknya harus ada koneksi internet GPRS di daerah anda, agar perangkat siap untuk online sesuai dengan kebutuhan kita.

2.6 Tools

Pada sub bab ini akan dijelaskan *tools* yang digunakan dalam pembangunan aplikasi pembelajaran ilmu pengetahuan alam ini.

2.6.1 Java

Bahasa Java dikembangkan oleh *Sun Microsystem* tahun 1991 sebagai bagian dari suatu proyek penelitian untuk mengembangkan software bagi konsumer barang – barang elektronik seperti televise, *VCR*, *toaster* dan mesin – mesin lainnya yang dapat dibeli di swalayan. Tujuan penciptaan Java pada waktu itu adalah menjadi suatu program yang berukuran kecil, efisien, dan *portable* di segala jenis *hardware*. Tujuan yang sama ini membuat Java menjadi satu bahasa yang ideal untuk mendistribusikan program – program yang dapat dijalankan melalui *www* dan juga suatu bahasa pemrograman untuk segala tujuan untuk mengembangkan program – program yang dapat digunakan dengan mudah dan *portable* di berbagai *platform* yang berbeda.

Sekarang, *Sun* telah mengeluarkan berbagai program Java yang dapat digunakan seperti *Java API,* atau JDK atau *JAVA Developer Kit* . Selain itu, banyak juga program-program lain yang dapat digunakan untuk membuat program Java, seperti *Eclipse, NetBeans, JBuilder, JCreator, J++,* dan sebagainya.

Saat ini distribusi Java dan kelas pendukungnya dibagi dalam tiga bagian yang masing-masing memiliki konsentrasi tersendiri yaitu [6]:

1. *Java 2 Standart Edition* (J2SE), untuk aplikasi *desktop*
2. *Java 2 Enterprise Edition* (J2EE), untuk aplikasi *server*

3. *Java 2 Micro Edition* (J2ME), untuk piranti dengan kemampuan terbatas.

2.6.2 Android SDK (Software Development Kit)

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk memulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan yang bukan merupakan aplikasi bawaan *smartphone*.

2.6.3 Eclipse

Pada bagian ini akan dijelaskan mengenai pengertian dan sifat Eclipse, berbagai versi Eclipse, serta Arstitektur Eclipse.

2.6.3.1 Pengertian dan Sifat Eclipse

Eclipse adalah sebuah IDE (Intergrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di platform. 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. 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.
3. Multi-role : Selain sebagai IDE untu 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 lain. Selain itu kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in.*

2.6.3.2 Versi Eclipse

Sejak tahun 2005, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama *Simlutaneous Release*. Tujuannya adalah untuk menyediaan distribusi Eclipse dngan fitur dan versi yang terstandarisasi.

Tabel 2.1 Versi Eclipse[5]

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

2.6.4 ADT/Plugin Eclipse

Android Development Tools (ADT) adalah *plugin* yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi project android, membuat GUI aplikasi, dan menambakan komponen-komponen yang lainnya, begitu juga kita dapat melakukan running aplikasi menggunakan Android SDK melalui Eclipse. Dengan ADT juga kita dapat melakukan pembuatan package android (.apk) yang digunakan untuk distribusi aplikasi android yang kita rancang.

Berikut adalah versi ADT untuk Eclipse yang sudah dirilis[10] : ADT 0.9.4 (Oktober 2009), ADT 0.9.5 (Desember 2009), ADT 0.9.6 (March 2010), ADT 0.9.7 (May 2010), ADT 0.9.8 (September 2010), ADT 0.9.9 (September 2010), ADT 8.0.0 (Desember 2010), ADT 8.0.1 (Desember 2010), ADT 9.0.0 (Januari 2011), ADT 10.0.0 (Februari 2011), ADT 10.0.1 (Maret 2011), ADT 11.0.0 (Juni 2011), dan ADT 12.0.0 (Juli 2011).

2.6.5 Android Virtual Device (AVD)

AVD adalah emulator untuk menjalankan program aplikasi android yang kita buat. Dengan AVD ini, kita bisa mengembangkan dan mencoba aplikasi Android tanpa harus menggunakan perangkat Android yang sebenarnya. Kita bisa menentukan karakteristik AVD, misalkan menentukan versi Android, jenis dan ukuran layar, besarnya memory, dan lain sebagainya. AVD bisa dibuat sebanyak yang kita inginkan.

2.6.6 Pemrograman Berorientasi Objek

Pada bagian ini dijelaskan tentang pengertian, bahasa pemrograman, dan konsep dasar pemograman berorientasi objek.

2.6.6.1 Pengertian Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengolalaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek [13].

2.6.6.2 Bahasa Pemrograman Berorientasi Objek

Berikut beberapa contoh bahasa pemrograman yang mendukung pemrograman berorientasi objek :

1. Bahasa Pemrograman Smalltalk, Smalltalk merupakan salah satu bahasa pemrograman yang dikembangkan untuk mendukung pemrograman berorientasi objek mulai tahun 1978.
2. Bahasa Pemrograman Eiffel, Eiffel merupakan bahasa pemrograman mulai tahun 1985 yang memiliki sintaks mirip dengan sintaks pada bahasa pemrograman pascal.
3. Bahasa Pemrograman C++, Bahasa pemrograman C++ merupakan pengembangan lebih lanjut dari bahasa pemrograman C.
4. Bahasa Pemrograman PHP, PHP 4 dengan versi-versi akhir menuju PHP5 sudah mendukung pemrograman berorientasi objek.

5. Bahasa Pemrograman Java, Java merupakan bahasa pemrograman objekm murni karena semua kode programnya dibungkus dalam kelas.

2.6.6.3 Konsep Dasar Metodologi Beriorentasi Objek

Berikut ini adalah konsep dasar yang harus dipahami tentang metodologi berorientasi objek :

1. Kelas (*class*)

   Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (relationship), dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, di mana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

2. Objek (*object*)

   Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berperngaruh pada status objeknya.

3. Metode (*method*)

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi

4. Atribut (*Atribute*)

   Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek.

5. Abstraksi (*abstraction*)

   Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederahana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

   Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tida mengetahui cara kerjanya.

7. Pewarisan (*inheritance*)

   Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya

8. Antarmuka (*interface*)

   Antarmuka atau interface sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Antarmuka biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas, mengakses antarmukanya.

9. *Reusability*

   Pemanfaatan kembali objek yang sudah didefinisikan untuk permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

    Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

11. Komunikasi Antarobjek

    Komunikasi antarobjek dilakukan lewat pesan (message) yang dikirim dan satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

    Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehinga menghemat baris program.

13. *Package*

    Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.6.7 UML

Pada bagian ini akan dijelaskan tentang pengertian UML dan diagram UML.

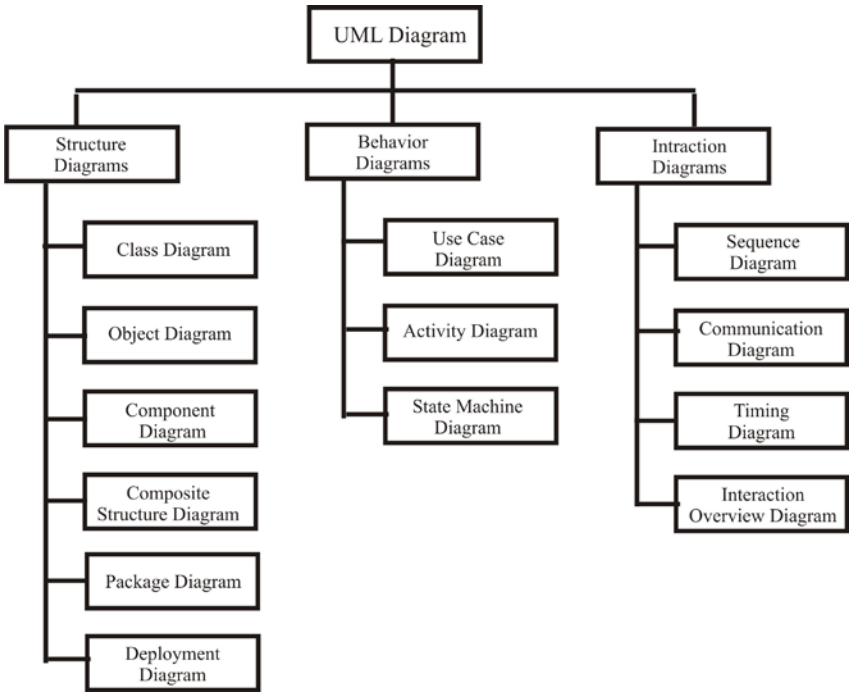
2.6.7.1 Pengertian UML

Unified Modeling Language (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 oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. 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 untukmemahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

2.6.7.2 Diagram UML



Gambar 2.2 Diagram UML[13]

Berikut ini penjelasan singkat dari pembagian kategori tersebut :

1. *Structure Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction Diagrams* yaitu kumpulan diagaram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

Berikut ini penjelasan dari masing-masing diagram :

1. *Class Diagram*

   Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi.

2. *Object Diagram*

   Diagram objek menggambarkan struktur sistem dari segi penamaan obje dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan.

3. *Component Diagram*

   Diagram komponen dibuat untuk menunjukan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

4. *Composite Structure Diagram*

   Composite structure diagram baru mulai ada pada UML versi 2.0. Diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*) dari *instance* yang terhubung.

5. *Package Diagram*

   Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML.

6. *Deployment Diagram*

   Diagram *deployment* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut: sistem tambahan yang menggambarkan rancangan, *device*, *node*, dan *hardware*, sistem *client/server*, sistem terdistribusi murni, serta rekayasa ulang aplikasi.

7. *Use Case Diagram*

   Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Ada dua hal utama pada *use case* yaitu pendefinisian aktor (orang, proses, atau sistem lain yang berinterkasi dengan sistem informasi yang akan dibuat) dan *use case* (fungsionalitas sebagai unit-unit yang saling bertukar pesan antarunit dan aktor).

8. *Activity Diagram*

   Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor.

9. *State Machine Diagram*

   State machine diagram digunakan untuk menggambarkan perubahan status atau transisi dari sebuah mesin atau sistem. Perubahan tersebut digambarkan dalam suatu graf berarah.

10. *Sequence Diagram*

    Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima objek. Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri.

11. *Communication Diagram*

    Communication diagram menggambarkan interaksi antarobjek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi merepresentasikan informasi yang diperoleh dari diagram kelas, diagram sekuen, dan diagram *use case* untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem.

12. *Timing Diagram*

    Timing diagram merupakan diagaram yang fokus pada penggambaran terkait batasan waktu. *Timing diagram* digunakan untuk menggambarkan tingkah laku sistem dalam periode waktu tertentu.

13. *Interaction Overview Diagram*

    Interaction overview diagram adalah bentuk aktivitas diagram yang setiap titik merepresentasikan diagram interaksi. Interaksi diagram dapat meliputi diagram sekuen, diagram komunikasi, dan *timing diagram.*