

BAB II

LANDASAN TEORI

2.1 Pengertian Aplikasi *Mobile*

Menurut Wikipedia, pengertian aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA. Adapun karakteristik perangkat *mobile* yaitu:

1. Ukuran yang kecil : Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang terbatas : Perangkat *mobile* juga memiliki memory yang kecil, yaitu *primary (RAM)* dan *secondary (disk)*.
3. Daya proses yang terbatas : Sistem *mobile* tidaklah setangguh rekan mereka yaitu *desktop*.

4. Mengonsumsi 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. Konektivitas yang terbatas : Perangkat *mobile* memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung.
7. Masa hidup yang pendek : Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

2.2 Pengertian Android

Menurut Nasruddin Safaath (Pemrograman aplikasi *mobile smartphone* dan tablet PC berbasis android 2012:1) android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk 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 di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

a. Kelebihan Android

- a) Multitasking – Kalau anda pernah merasakan keunggulan dari Symbian yang bisa membuka beberapa aplikasi sekaligus, begitu juga Android yang mampu membuka beberapa aplikasi sekaligus tanpa harus menutup salah satunya.
- b) Kemudahan dalam Notifikasi – Setiap ada SMS, Email, atau bahkan artikel terbaru dari RSS Reader, akan selalu ada notifikasi di *Home Screen Ponsel Android*, tak ketinggalan Lampu LED Indikator yang berkedip-kedip, sehingga Anda tidak akan terlewatkan satu SMS, *Email* ataupun *Misscall* sekalipun.
- c) Akses Mudah terhadap Ribuan Aplikasi Android lewat *Google Android App Market* – Kalau Anda gemar *install* aplikasi ataupun games, lewat *Google Android App Market* Anda bisa *mendownload* berbagai aplikasi dengan gratis. Ada banyak ribuan aplikasi dan *games* yang siap untuk Anda download di ponsel Android.
- d) Pilihan Ponsel yang beranekaragam – Bicara ponsel Android, akan terasa ‘beda’ dibandingkan dengan iOS, jika iOS hanya terbatas pada iPhone dari Apple, maka Android tersedia di ponsel dari berbagai produsen, mulai dari Sony Ericsson, Motorola, HTC sampai Samsung.

Dan setiap pabrikan ponsel pun menghadirkan ponsel Android dengan gaya masing-masing, seperti Motorola dengan Motoblur-nya, Sony Ericsson dengan TimeScape-nya. Jadi Anda bisa leluasa memilih ponsel Android sesuai dengan ‘merk’ favorite.

- e) Bisa menginstal ROM yang dimodifikasi – tak puas dengan tampilan standar Android, jangan khawatir ada banyak *Costum ROM* yang bisa Anda pakai di ponsel Android.
- f) *Widget* – benar sekali, dengan adanya *Widget* di *homescreen*, Anda bisa dengan mudah mengakses berbagai setting dengan cepat dan mudah.
- g) *Google Maniak* – Kelebihan Android lainnya jika Anda pengguna setia layanan *Google* mulai dari *Gmail* sampai *Google Reader*, ponsel Android telah terintegrasi dengan layanan *Google*, sehingga Anda bisa dengan cepat mengecek email dari Gmail.

b. Kelemahan Android

- a) Koneksi Internet yang terus menerus – Yups, kebanyakan ponsel berbasis system ini memerlukan koneksi internet yang simultan alias terus menerus aktif. Koneksi internet GPRS selalu aktif setiap waktu, itu artinya Anda harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan.
- b) Iklan – Aplikasi di Ponsel Android memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut,

akan selalu Iklan yang terpampang, entah itu bagian atas atau bawah aplikasi.

2.3 Pengertian *Java*

Menurut Budi Raharjo , Imam Heryanto, Arif haryono (Mudah Belajar Java 2010) *java* adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis *Java* umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual *Java* (JVM). *Java* merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi *Java* mampu berjalan di beberapa platform sistem operasi yang berbeda, *Java* dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini *Java* merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web. Adapun kelebihan dari *Java* yaitu :

1. Multiplatform. Kelebihan utama dari *Java* ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program *Java* dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis *Java* dikerjakan diatas *operating system Linux* tetapi dijalankan dengan baik di atas *Microsoft Windows*. Platform yang didukung sampai saat ini adalah *Microsoft Windows*, *Linux*, *Mac OS* dan *Sun Solaris*. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs *Java*) untuk meninterpretasikan *bytecode* tersebut.
2. OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek)
3. Perpustakaan Kelas Yang Lengkap, *Java* terkenal dengan kelengkapan library/perpustakaan (kumpulan program program yang disertakan dalam pemrograman *Java*) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas *Java* yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
4. Bergaya *C++*, memiliki sintaks seperti bahasa pemrograman *C++* sehingga menarik banyak pemrogram *C++* untuk pindah ke *Java*. Saat ini pengguna *Java* sangat banyak, sebagian besar adalah pemrogram *C++*

yang pindah ke *Java*. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan *Java* kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.

5. Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa *C++* yang dipakai secara luas).

Sedangkan kekurangan *Java* yaitu :

1. Tulis sekali, jalankan di mana saja - Masih ada beberapa hal yang tidak kompatibel antara *platform* satu dengan *platform* lain. Untuk *J2SE*, misalnya *SWT-AWT bridge* yang sampai sekarang tidak berfungsi pada *Mac OS X*.
2. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi *Java* merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada *Microsoft .NET Platform*. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/*direverse-engineer*.
3. Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis *Java* jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti *C/C++* dan *Pascal* (lebih spesifik lagi, *Delphi* dan

Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutut dengan mesin komputer berumur lebih dari 4 tahun.

2.4 Pengertian *UML*

Menurut (Adi Nugroho : 2005). “*Unified Modeling Language (UML)* adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek”. Menurut (Joomla dari <http://soetrasoft.com> : 2007). “*Unified Modeling Language (UML)* merupakan standard modeling language yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan *software* agar bisa menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan testing serta Dokumentasi”.

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa “*Unified Modeling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OO (*Object Oriented*)”. *UML* merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. *UML* tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. Bagian-bagian utama dari *UML* adalah *View*, *Diagram*, *model element*, dan *general mechanism*.

a. *View*

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah *Diagram*. Beberapa jenis *View* dalam *UML* antara lain: *Use case View*, *Logical View*, *Component View*, *concurrency View*, dan *Deployment View*.

1. *Use case View*

Use case View mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa *user* atau sistem lainnya. *View* ini digambarkan dalam *Use case Diagrams* dan kadang-kadang dengan *Activity Diagrams*. *View* ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

2. *Logical View*

Logical View mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*Class*, *object*, dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu. *View* ini digambarkan dalam *Class Diagrams* untuk struktur statis dan dalam *State*, *Sequence*, *Collaboration*, dan *Activity Diagram* untuk model dinamisnya. *View* ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

3. *Component View*

Component View mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya. *View* ini digambarkan dalam *Component View* dan digunakan untuk pengembang (developer).

4. *Concurrency View*

Concurrency View membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam *Diagram* dinamis (*State*, *Sequence*, *Collaboration*, dan *Activity Diagrams*) dan *Diagram* implementasi (*Component* dan *Deployment Diagrams*) serta digunakan untuk pengembang (developer), pengintegrasi (integrator), dan penguji (tester).

5. *Deployment View*

Deployment View mendeskripsikan fisik dari sistem seperti komputer dan perangkat (nodes) dan bagaimana hubungannya dengan lainnya. *View* ini digambarkan dalam *Deployment Diagrams* dan digunakan untuk pengembang (developer), pengintegrasi (*integrator*), dan penguji (*tester*).

6. *Diagram*

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah *Diagram* merupakan bagian dari suatu *View* tertentu dan

ketika digambarkan biasanya dialokasikan untuk *View* tertentu. Adapun jenis *Diagram* antara lain :

1) *Use case Diagram*

Use case adalah abstraksi dari interaksi antara system dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata user. Sedangkan *Use case Diagram* memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

2) *Class Diagram*

Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *Class Diagram* dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari *Class- Class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *Class Diagram*. *Class Diagram* sangat membantu dalam visualisasi struktur kelas dari suatu system.

3) *Component Diagram*

Component software merupakan bagian fisik dari sebuah system, karena menetap di komputer tidak berada di benak para analis. *Component* merupakan implementasi software dari sebuah atau lebih *Class*. *Component* dapat berupa *source code*, *Component*

biner, atau *executable Component*. Sebuah *Component* berisi informasi tentang *logic Class* atau *Class* yang diimplementasikan sehingga membuat pemetaan dari *Logical View* ke *Component View*. Sehingga *Component Diagram* merepresentasikan dunia riil yaitu *Component* software yang mengandung *Component*, interface dan relationship.

4) *Deployment Diagram*

Deployment Diagram menggambarkan tata letak sebuah system secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable Component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

5) *State Diagram*

State Diagram menggambarkan semua *State* (kondisi) yang dimiliki oleh suatu *object* dari suatu *Class* dan keadaan yang menyebabkan *State* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan. *State Class* tidak digambarkan untuk semua *Class*, hanya yang mempunyai sejumlah *State* yang terdefinisi dengan baik dan kondisi *Class* berubah oleh *State* yang berbeda.

6) *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

7) *Collaboration Diagram*

Collaboration Diagram menggambarkan kolaborasi dinamis seperti *Sequence Diagrams*. Dalam menunjukkan pertukaran pesan, *Collaboration Diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *Sequence Diagrams*, tapi jika penekanannya pada konteks gunakan *Collaboration Diagram*.

8) *Activity Diagram*

Activity Diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *Use case* atau interaksi.

Adapun tujuan penggunaan *UML* yaitu :

- a. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
- b. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
- c. Memberikan model yang siap pakai, bahsa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

d. *UML* bisa juga berfungsi sebagai sebuah (blue print) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bias diketahui informasi secara detail tentang coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk *Diagram* (*reverse engineering*).

Sedangkan perangkat lunak yang mendukung pembuatan diagram *UML* yaitu

a) *StarUML* (<http://StarUML.sourceforge.net/en/>)

StarUML adalah sebuah proyek *Staropen source* untuk mengembangkan cepat, *fleksibel*, *extensible*, *featureful*, dan bebas-tersedia *UML / platform MDA* berjalan pada *platform Win32*. Tujuan dari proyek *StarUML* adalah untuk membangun sebuah alat pemodelan perangkat lunak dan juga platform yang menarik adalah pengganti alat *UML* komersial seperti *Rational Rose*, *Bersama* dan sebagainya.

b) *Acceleo* (<http://www.acceleo.org/pages/home/en>)

Acceleo adalah generator kode yang mengubah model menjadi kode. *Acceleo* mudah digunakan dan menyediakan “dari rak” generator (Jee, Bersih, Php ...) dan *template editor* untuk *Eclipse*.

c) *ArgoUML* (<http://ArgoUML.tigris.org/>)

ArgoUML adalah *Staropen source UML modeling tool* terkemuka dan termasuk dukungan untuk semua *Diagram UML* standar 1,4. Ini berjalan pada setiap *platform Java* dan tersedia dalam bahasa sepuluh. *ArgoUML* ditulis seluruhnya di Jawa dan menggunakan *Java Kelas Foundation*. Hal ini memungkinkan *ArgoUML* untuk berjalan di hampir semua *platform*.

2.5 Pengertian *Eclipse*

Menurut Nasruddin Safaat h (Pemrograman aplikasi mobeli smartphone 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. *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* 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 menginstal *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.

2.6 Pengertian Shalat

Menurut bahasa shalat artinya adalah berdoa, sedangkan menurut istilah shalat adalah suatu perbuatan serta perkataan yang dimulai dengan takbir dan diakhiri dengan salam sesuai dengan persyaratan yang ada.

Menurut Moh.Yusuf Haji Ismail (Panduan Lengkap Shalat Sunnah 2005 : 3) Shalat Sunnah adalah shalat yang dilakukan muslim dengan kerelaan hatinya, baik shalat sunnah itu dikerjakan pada waktu malam ataupun siang. Tujuan shalat adalah untuk mencegah perbuatan keji dan munkar. Untuk melakukan shalat ada syarat-syarat yang harus dipenuhi dulu, yaitu :

1. Beragama Islam
2. Memiliki akal yang waras alias tidak gila atau autis
3. Berusia baligh (bagi perempuan : telah haid, bagi laki-laki: telah bermimpi)
4. Telah sampai dakwah islam kepadanya
5. Bersih dan suci dari najis, haid, nifas, dan lain sebagainya
6. Sadar atau tidak sedang tidur

Syarat sah pelaksanaan sholat adalah sebagai berikut ini :

1. Masuk waktu sholat
2. Menghadap ke kiblat
3. Suci dari najis baik hadas kecil maupun besar
4. Menutup aurat

Dalam sholat ada rukun-rukun yang harus kita jalankan, yakni :

1. Niat
2. Posisis berdiri bagi yang mampu
3. Takbiratul ihram
4. Membaca surat al-fatihah
5. Ruku / rukuk yang tumakninah
6. I'tidal yang tuma'ninah
7. Sujud yang tumaninah
8. Duduk di antara dua sujud yang tuma'ninah
9. Sujud kedua yang tuma'ninah
10. Tasyahud
11. Membaca salawat Nabi Muhammad SAW
12. Salam ke kanan lalu ke kiri

Dalam melaksanakan ibadah salat, sebaiknya kita memperhatikan hal-hal yang mampu membatalkan shalat kita, contohnya seperti :

1. Terdapat hadas / najis baik pada tubuh, pakaian maupun lokasi
2. Berkata-kata kotor
3. Melakukan banyak gerakan di luar sholat bukan darurat
4. Gerakan sholat tidak sesuai rukun shalat dan gerakan yang tidak tuma'ninah.

2.7 Pengertian Do'a

Menurut bahasa do'a berasal dari Bahasa Arab الدعاء yang merupakan bentuk masdar dari mufrad داعى yang memiliki bermacam-macam arti. Dalam kamus Bahasa Arab di bawah judul huruf د, ع, و disebutkan sebagai berikut:

- 1) داعي, يدعو, دعوة artinya menyeru, memanggil.
- 2) داعي, يدعو, دعاء artinya memanggil, mendo'a, memohon, meminta.
- 3) Dalam bentuk jama'nya ادعية artinya do'a, permohonan, permintaan.
- 4) دعاء له artinya mendo'akan kebaikan kepadanya.
- 5) دعاء عليه artinya mendo'akan keburukan atau kejahatan kepadanya.
- 6) داع artinya orang yang memanggil, orang yang menyeru, orang yang memohon.
- 7) الدعاء adalah bentuk masdarnya, yang pada umumnya diartikan sebagai suatu keinginan yang besar kepada Allah SWT dan pujian kepadaNya.

Adapun lafadz do'a yang ada dalam al Qur'an bisa bermakna sebagai berikut:

- a) Ibadah, seperti firman Allah “ Dan janganlah kamu menyembah apa-apa yang tidak memberi manfaat dan tidak memberi madharat kepadamu selain Allah sebab jika kamu berbuat demikian maka, kamu termasuk orang-orang yang zhalim”. (Yunus: 106)
- b) Perkataan atau Keluhan. Seperti pada firman Allah: Maka tetaplah demikian keluhan mereka, sehingga kami jadikan mereka sebagai tanaman yang telah dituai, yang tidak dapat hidup lagi. (al Anbiya: 15).
- c) Panggilan atau seruan. Allah berfirman: Maka kamu tidak akan sanggup menjadikan orang-orang yang mati itu dapat mendengar, dan menjadikan orang-orang yang tuli dapat mendengar seruan, apabila mereka itu berpaling ke belakang. (ar- Rum: 52)
- d) Meminta pertolongan. Allah berfirman: Dan jika kamu (tetap) dalam keraguan tentang at Qur'an yang Kami wahyukan kepada hamba Kami

(Muhammad) buatlah satu surat yang semisal at Qur'an itu dan ajaklah penolong-penolongmu selain Allah, jika kamu orang-orang yang benar. (al Baqarah: 23).

- e) Permohonan. Seperti firman Allah: Dan orang-orang yang berada dalam neraka berkata kepada penjagapenjaga jahannam: "Mohonkanlah kepada Tuhanmu supaya Dia meringankan azab dari kami barang sehari." (al Mukmin: 49).

Sedangkan Fadhilah (Keutamaan) Berdo'a yaitu: Do'a merupakan ibadah dan sebuah ketaatan alas perintah Allah. Allah ber_rman: Dan Tuhanmu berfirman: Berdo'alah kepadaKu, niscaya akan Kaperkenankan bagimu, sesungguhnya orang-orang yang menyombongkan diri dari menyembahKu akan masuk neraka jahannam dalam keadaan hina dina. (al Mukmin: 60). Di dalam sebuah hadits disebutkan: Dari an-Nu'man bin Basyir berkata, bahwa Rasulullah bersabda: Do'a adalah ibadah.

- 1) Merupakan perbuatan yang paling mulia dan dicintai oleh Allah. Dari Abu Hurairah Rasulullah bersabda: Tidak ada sesuatu yang lebih mulia di sisi Allah dari do'a.
- 2) Do'a menghalangi kemurkahan Allah. Karena orang yang tidak berdo'a kepada Allah, Allah akan marah kepadanya sebagaimana yang disabdakan oleh Rasulullah, Barang siapa yang tidak meminta kepada Allah, Allah akan marah padanya.
- 3) Do'a menunjukkkan kecerdasan dan kekuatan batin seseorang. Dari Abu Hurairah bahwasanya Rasulullah bersabda: "Selemah-lemahnya manusia

adalah orang lemah (jarang) berdo'a dan sebakhlbakhilnya manusia adalah orang yang bakhil dalam mengucapkan salam.

- 4) Do'a bisa mencegah bencana yang belum terjadi dan menghilangkannya bila sudah terjadi. Rasulullah bersabda: Tidak ada yang bisa menolak al Qadar (takdir) itu kecuali do'a.
- 5) Do'a menjadi perekat tali cinta dan kasih sayang sesama mukmin. Karena seseorang yang mendo'akan saudaranya yang lain yang jauh, maka akan dikabulkan. Allah ber_rman: Sesnngguhnya orang-orang yang beriman dan beramal shalih, Allah akan menanamkan dalam (hati) mereka rasa kasih sayang. (Maryam: 96). Dan sudah disepakati bahwa do'a termasuk dalam iman dan amal shalih.
- 6) Berdo'a merupakan sifat orang-orang muttaqin. Allah berfirman: Dan orang-orang yang datang sesudah mereka (Muhajirin dan Anshar) mereka berdo'a: Ya Tahan kami, beri ampunlah kami dan saudara-saudara Kami yang telah beriman lebih dahulu dari kami, dan janganlah engkau menjadikan kedengkian dalam hati kami terhadap orang-orang yang beriman, Ya Tuhan kami sesungguhnya Engkau Maha Penyantun lagi Maha Penyayang. (al Hasyr: 10). Dan juga merupakan sifat para Nabi-nabi terdahulu, kalau mendapatkan permasalahan mereka segera berdo'a kepada Allah, sebagaimana yang diceritakan oleh Allah dalam firmanNya: "Maka Kami berkenankan do'anya, dan Kami menganugerahkan Yahya kepadanya. Dan Kami jadikan isterinya dapat mengandung. Sesungguhnya mereka adalah orang-

orang yang selalu bersegera dalam (mengerjakan) perbuatan-perbuatan yang baik dan mereka berdo'a kepada Kami dengan harap dan cemas, dan mereka adalah orang-orang yang khusyuk kepada Kami." (al Anbiya: 90).

7) Do'a menjadikan seseorang tsabat (teguh/kokoh) dalam menghadapi musuh dan sarana untuk mendapatkan pertolongan. Sebagaimana kisah Thalut dengan pasukan seadanya mampu mengalahkan Jalut dengan bala tentaranya. Allah ber_rman: Tatkala Jalut dan tentaranya telah nampak oleh mereka, merekapun (Thalut dan tentaranya) berdo'a. "Ya Tuhan kami, berilah kesabaran pada diri kami, dan kokohkanlah pendirian kami dan tolonglah kami terhadap orang-orang Kafr." (al-Baqarah: 250).

8) Do'a sebagai pelarian dan tumpuan harapan orang-orang yang teraniaya. Seperti yang di contohkan oleh Nabi-nabi terdahulu ketika dimusuhi oleh kaumnya, bahkan akan dibunuh dan dianiaya. Mereka berdo'a kepada Allah. Seperti yang diceritakan oleh Allah tentang Nabi Nuh. Sebelum mereka (kaum musyrikin Quraishy), kaum Nuh telah mendustakan (Nuh). Maka mereka mendustakan hamba Kami (Nuh) dan mengatakan) "Dia seorang gila dan dia sudah pernah diberi ancaman." Maka dia mengadu kepada Tuhannya "bahwasannya aku ini adalah orang yang dikalahkan, oleh sebab itu menangkanlah aku. (al Qamar: 9-10). Begitu juga yang dilakukan oleh Nabi Adam sebelumnya, jaga oleh Nabi Ibrahim, Nabi Yunus, Nabi Musa dan Nabi Muhammad.