

BAB II

LANDASAN TEORI

Dasar dari sebuah penelitian adalah landasan teori. Selain untuk mendukung pelaksanaan pada penelitian, landasan teori akan membuktikan bahwa penelitian tersebut berkualitas, dan memiliki pengetahuan yang cukup untuk melanjutkan penelitian sebelumnya, memperbaiki teori-teori pada penelitian sebelumnya, atau bahkan mematahkan teori sebelumnya.

2.1 Konsep Dasar Sistem Informasi

2.1.1 Konsep Dasar Sistem

Menurut Tata Sutabri (2012,p.3) Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum pertama kali diuraikan oleh Kenneth Boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem. Model umum sebuah sistem adalah input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu, sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem.

2.1.2 Konsep Dasar Informasi

Menurut Tata Sutabri (2012, p.22) Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari tak berguna menjadi berguna bagi

penerimanya. Nilai informasi berhubungan dengan keputusan maka informasi menjadi tidak diperlukan keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks sebuah keputusan.

2.1.3 Konsep Sistem Informasi

Menurut Tata Sutabri (2012, p.38) Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan- laporan yang diperlukan oleh pihak luar tertentu dengan.

2.2 Pengertian Anatomi

Menurut (Wikipedia, 2016), Anatomi berasal dari bahasa Yunani yang berarti memotong. Ilmu anatomi merupakan cabang ilmu pengetahuan mengenai struktur tubuh pada manusia, hewan, dan makhluk lainnya. Pemahaman tentang ilmu anatomi secara keseluruhan merupakan sesuatu yang sulit bahkan para ahli sekalipun yang mengkhususkan diri mempelajari sebuah bidang anatomi juga mengalami kesulitan untuk memahami semuanya.

Tubuh manusia tersusun oleh serangkaian sistem yang kompleks, dimulai dari sel, jaringan, organ, dan gabungan dari beberapa organ sehingga menjadi sebuah sistem organ yang mempunyai fungsi dan peran tersendiri dalam tubuh manusia. Peran dan fungsi dari organ tubuh saling berhubungan dengan beberapa organ tubuh lainnya.

Pengertian atau definisi fisiologi secara umum memiliki pengertian yaitu mempelajari hal yang berkaitan dengan kehidupan makhluk hidup walaupun dalam setiap bidang memiliki pengertian masing-masing. Ilmu fisiologi menggunakan tahapan dan langkah serta berbagai macam metode untuk dapat mempelajari sebuah sel, lalu biomolekul, kemudian organ dan jaringan. Selain itu, fisiologi juga mempelajari organisme dan sebuah sistem organ secara merata dan keseluruhan untuk menjalankan fungsi fisik serta zat kimianya agar mendukung

sebuah kehidupan. Ilmu fisiologi ialah bidang ilmu yang wajib untuk di pelajari karena merupakan salah satu bidang ilmu yang sudah menjadi salah satu objek untuk pemberian penghargaan tertinggi Nobel.

Secara umum pengertian fisiologi memiliki makna lain karena berdasarkan sebuah objek kajiannya fisiologi dikenal juga ilmu fisiologi manusia kemudian ilmu fisiologi tumbuhan dan ilmu tentang fisiologi hewan, walaupun pada dasarnya prinsip sebuah ilmu fisiologi sangat bersifat luas atau universal, hal tersebut sangat tidak terlalu bergantung pada salah satu jenis dan tipe suatu organisme yang dipahami dan dipelajari. Ada beberapa contoh tentang apa yang dapat dipelajari dan dipahami pada ilmu fisiologi tentang sel khamir serta dapat juga diterapkan dan dilakukan sebagian maupun secara ke seluruhan terhadap sel organ manusia.

Banyak ilmu-ilmu lain yang sudah berkembang dan maju dari ilmu fisiologi karena mengingat bahwa ilmu ini sudah cukup lama dan terbilang tua saat awal dipelajari hingga kini. Beberapa sebuah turunan yang bisa dibilang sangat penting adalah ilmu biokimia kemudian ilmu biofisika lalu ilmu tentang biomekanika dan ilmu yang mempelajari tentang genetika sebuah sel serta ilmu tentang farmakologi selain itu ada ilmu tentang ekofisiologi. Sebuah perkembangan ilmu biologi tentang molekuler sangat memengaruhi arah dan tujuan kajian ilmu fisiologi.

Fisiologi sangat banyak diminati oleh para peneliti karena ilmu ini sangat bermanfaat bagi umat manusia karena pengertian fisiologi adalah sebuah ilmu yang bisa dibilang ilmu ini mempelajari berbagai macam teori dan rangkaian tentang seluruh kehidupan baik alam maupun penghuninya secara keseluruhan dan manfaat ilmu ini juga dapat mempelajari tentang masa lalu maupun yang akan datang terhadap revolusi makhluk hidup.

2.3 Pengertian Analisis dan Desain Berorientasi Objek

2.3.1 Pengertian OOAD

OOAD adalah metode analisis yang memeriksa requirements dari sudut pandang kelas kelas dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek system atau subsistem. OOAD merupakan cara baru

dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas.

2.3.2 Konsep Dasar OOAD

OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memeriksa requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. Sedangkan OOD adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

Terdapat beberapa konsep dalam OOAD, yaitu :

1. Kelas (*Class*)

Kumpulan objek-objek dengan karakter yang sama. Sebuah kelas mempunyai sifat (*atribut*), kelakuan (*operasi/metode*), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dari kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Kelas adalah sebuah struktur tertentu dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*Object*)

Abstraksi dan sesuatu yang mewakili dunia nyata. Objek merupakan suatu entitas yang mampu menyimpan informasi (*status*) dan mempunyai operasi (*kelakuan*) yang dapat diterapkan.

3. Metode (*Method*)

Operasi atau metode pada kelas hampir sama dengan fungsi atau prosedur pada metodologi terstruktur. Operasi merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan objek.

4. Atribut

Variabel global yang dimiliki kelas. Atribut dapat berupa nilai atau elemen – elemen data yang dimiliki oleh objek dalam kelas. Atribut dipunyai secara individu oleh suatu objek misalnya berat, jenis.

5. Abstraksi

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi suatu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan masalah.

6. Enkapsulasi

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

7. Pewarisan

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

8. Antar muka

Biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas.

9. *Reusability*

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

10. Generalisasi & spesialisasi

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

11. Komunikasi antar objek

Dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

12. *Polimorfisme*

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

13. *Package*

Sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompok kelas-kelas yang bernama sama disimpan dalam package yang berbeda.

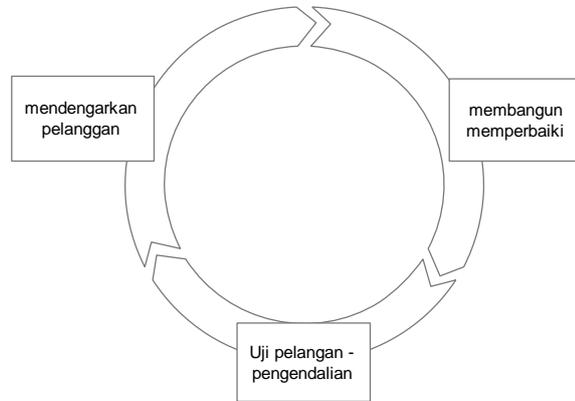
2.4 Metode Pengembangan sistem

Menurut Rosa A.S M. Shalahudin (2014, p.26) proses pengembangan mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan dalam mengembangkan sistem-sistem perangkat lunak sebelumnya.

2.4.1 Metodologi *Prototyping*

Menurut Rosa A.S M. Shalahudin (2016, p.31) Model prototipe dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Model prototipe (*prototyping model*) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program prototipe ini dievaluasi

oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau user.



Gambar 2.1 *Prototype* Paradigma

Proses pada model *prototyping* dapat dijelaskan sebagai berikut :

1. Pengumpulan kebutuhan : *developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detil kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan.
2. Perancangan : perancangan dilakukan cepat dan rancangan mewakili semua aspek perangkat lunak yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.
3. Evaluasi *prototype* : klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan perangkat lunak. Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype-prototype* dibuat untuk memuaskan kebutuhan klien dan untuk membangun perangkat lunak lebih cepat, namun tidak semua *prototype* bisa dimanfaatkan. Demi kebutuhan klien lebih baik *prototype* yang dibuat diusahakan dapat dimanfaatkan.

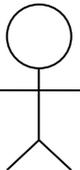
2.4.2 Definisi *Unified Modeling Language* (UML)

Menurut Rosa A.S M. Shalahudin (2016, p.137) UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. Perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya, dalam penggunaan UML adalah suatu yang salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin divisualkan. Menurut Rosa A.S M. Shalahudin (2016, p.140) Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori.

a. Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat . Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami.

Tabel 2.1 Simbol *Use Case*

Simbol	Keterangan
	<p>Aktor : Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.</p>
	<p>Use case : perangkat tertinggi dari fungsionalitas yang dimiliki sistem.</p>

	Association : adalah relasi antara actor dan <i>use case</i> .
	Generalisasi : untuk memperlihatkan struktur pewaris yang terjadi.
	Include : menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i>
	Collaboration : interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi)
	Extend : menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan

b. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.2 Simbol *Activity Diagram*

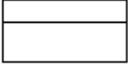
No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi

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.

c. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Tabel 2.3 Simbol *Class Diagram*

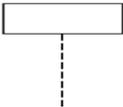
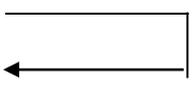
No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan 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 memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

d. 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 respons 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. Masing-masing objek, termasuk aktor, memiliki lifeline message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk object boundary, controller dan persistent entity. Komponen – komponen dari Sequence Diagram dapat dilihat pada tabel 2.4 berikut :

Tabel 2.4 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 to self</i>	Menggambarkan pesan yang menuju dirinya sendiri
4		<i>Return Message</i>	Menggambarkan pengembalian dari pemanggilan prosedur

2.5 Basis Data

Pendapat dari (Indrajani, 2009), “Basis data merupakan kumpulan terpadu dari elemen data logis yang saling berhubungan. Basis data mengkonsolidasi banyak catatan yang sebelumnya disimpan dalam *file* terpisah”. Sedangkan (Janner simarmata, 2006) berpendapat bahwa basisdata adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Dari beberapa pendapat tersebut dapat disimpulkan bahwa basisdata adalah sekumpulan informasi yang berhubungan dengan subjek atau tujuan tertentu seperti melacak pesanan konsumen, *maintaining inventory* atau pada umumnya mengumpulkan semua data yang berhubungan pada satu tempat. Data tersebut dapat disimpan, dimanipulasi dan digunakan dalam banyak cara.(Linda Marlinda, 2008) berpendapat bahwa terdapat istilah-istilah yang dipergunakan di dalam basisdata yakni sebagai berikut.

- a. Entitas
Suatu objek yang dapat dibedakan dengan objek lainnya yang dapat diwujudkan di dalam basis data.
- b. *Attribute/Field*
Karakteristik entitas tertentu.
- c. *Data Value*
Merupakan data *actual* atau informasi yang di simpan di tiap data elemen atau *attribute*. Isi *attribute* disebut nilai data.
- d. *Record/Tuple*
Kumpulan isi elemen data yang saling berhubungan menginformasikan tentang suatu entitas secara lengkap.
- e. *File*
Kumpulan *record* sejenis yang mempunyai panjang elemen dan *attribute* yang sama, namun berbeda-beda data *value*-nya.
- f. Kunci elemen data
Sebagai tanda pengenal yang secara unik mengidentifikasi entitas dari suatu kumpulan entitas.
- g. *Database management system*
File yang saling berkaitan bersama dengan program untuk pengelolaannya.

Kamus data adalah suatu daftar elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analisis sistem mempunyai pengertian yang sama tentang *input*, *output* dan komponen *data store*. Pembentukan kamus data didasarkan pada alur data yang terdapat pada DFD. Alur data pada DFD bersifat global (hanya menunjukkan alur datanya tanpa menunjukkan struktur dari alur data). Untuk menunjukkan struktur dari alur data secara rinci maka dibentuklah kamus data (Jogiyanto, 2005). Berikut contoh tampilan format kamus data pada tabel 2.5 :

Nama Database	:
Nama Tabel	:
Primary Key	:

Foreign Key :

Nama Field	Type	Size	Kondisi	Keterangan

Keterangan kondisi berisi (contoh : NULL/NOT NULL)

Tabel 2.5 Format Kamus Data

Selain itu, dalam perancangan *database* terdapat sistem kode yang digunakan untuk mengklasifikasikan data, memasukan data kedalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya. Kode dapat dibentuk dari kumpulan angka, huruf dan karakter-karakter khusus. Menurut (Jogiyanto, 2005) beberapa tipe kode yang digunakan diantaranya sebagai berikut.

a. Kode mnemonik (*mnemonic code*)

Kode mnemonik digunakan untuk tujuan supaya mudah diingat dengan dasar singkatan.

b. Kode urut (*sequential code*)

Kode urut disebut juga dengan kode seri, merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya.

c. Kode blok (*block code*)

Kode blok mengklasifikasikan *item* kedalam kelompok blok tertentu yang mencerminkan suatu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

d. Kode group (*group code*)

Kode group merupakan kode yang berdasarkan *field-field* dan tiap-tiap *field* kode mempunyai arti.

e. Kode desimal (*decimal code*)

Kode desimal mengklasifikasikan kode atas dasar 10 unit angka desimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari banyaknya kelompok.

2.6 Telepon Seluler

2.6.1 Pengertian Telepon Seluler

Telepon merupakan alat komunikasi yang digunakan untuk menyampaikan pesan suara (terutama pesan yang berbentuk percakapan). Kebanyakan telepon beroperasi dengan menggunakan transmisi sinyal listrik dalam jaringan telepon sehingga memungkinkan pengguna telepon untuk berkomunikasi dengan pengguna lainnya. Telepon Seluler atau yang sering disebut Hand Phone atau HP merupakan paduan perpaduan antara Teknologi Telepon dengan Teknologi Radio. Tetapi dalam perkembangannya Teknologi Komputer juga masuk dengan mulus pada telepon seluler ini.

2.6.2 Sejarah Perkembangan Telepon Seluler

Sejarah telepon seluler atau yang kita kenal HP, ternyata sudah ada dari jaman penjajahan, yaitu kira-kira tahun 1947 di negara paman sam alias Amrik dan Eropa sana. Pada tahun 1910 adalah cikal bakal telepon seluler yang ditemukan oleh Lars Magnus Ericsson, yang merupakan pendiri perusahaan Ericsson yang kini di kenal dengan perusahaan Sony Ericsson. Pada tahun 1921 pertama kalinya Departemen Kepolisian Detroit Michigan menggunakan telepon mobile yang terpasang di semua mobil polisi dengan menggunakan frekuensi 2 MHz. Pada tahun 1960, di Finlandia sebuah perusahaan bernama *Fennis Cable Works* yang semula berbisnis dibidang kabel, melakukan ekspansi dengan mendirikan perusahaan elektronik yang bernama Nokia sebagai handset telepon seluler. Tahun 1970-an perkembangan telepon mobile menjadi pesat dengan di dominasi oleh 3 perusahaan besar yaitu di Eropa dengan perusahaan Nokia dan Motorola. Pada tahun 1969, sistem telekomunikasi seluler dikomersialkan. Setelah tahun 1970, telekomunikasi seluler semakin sering dibicarakan orang. Motorola mengenalkan telepon genggam tiga tahun kemudian. Ukurannya memang cukup besar dengan antena pendek. Ada pula ponsel dengan ukuran sekoper. Dr Cooper yang menjadi manajer proyek inovasi Motorola itu

memasang base station di New York. Untuk proyek ini Motorola bekerja dengan Bell Labs. Penemuan ini sekaligus diklaim sebagai penemuan ponsel pertama. Di suatu pagi 3 April 1973, Cooper, saat itu menjabat sebagai general manager pada *Divisi Communication Systems Motorola* mempertunjukkan cara berkomunikasi aneh dari terminal telepon portable. Dia mencoba ponsel ‘raksasanya’ sambil berjalan–jalan di berbagai lokasi di New York. Itulah saat pertama ponsel ditampilkan dan digunakan di depan publik. Dalam pertunjukan itu, Cooper menggunakan ponsel seberat 30 ounce sekitar (800 gram) atau sepuluh kali lipat dibandingkan rata – rata ponsel yang beredar saat ini.

2.7 Android

2.7.1 Pengertian Android

Android adalah sistem operasi mobile yang dikembangkan oleh Google. Sistem ini digunakan oleh beberapa smartphone, seperti Motorola Droid, Samsung Galaxy dan Google sendiri melalui perangkat yang dinamakan Nexus. Sistem operasi Android (OS) didasarkan pada kernel Linux terbuka. Berbeda dengan OS iPhone, Android adalah open source, yang berarti pengembang dapat memodifikasi dan menyesuaikan OS untuk setiap telepon. Oleh karena itu, ponsel berbasis Android yang berbeda mungkin memiliki berbagai antarmuka pengguna grafis UI yang beragam meskipun mereka menggunakan OS yang sama.

2.7.2 Sejarah Perkembangan Android

Android dikembangkan secara pribadi oleh Google sampai perubahan terbaru dan pembaruan siap untuk dirilis, dan informasi mengenai kode sumber juga mulai diungkapkan kepada publik. Kode sumber ini hanya akan berjalan tanpa modifikasi pada perangkat tertentu, biasanya pada seri Nexus. Ada binari tersendiri yang disediakan oleh produsen agar Android bisa beroperasi. Logo Android yang berwarna hijau awalnya dirancang untuk Google pada

tahun 2007 oleh desainer grafis Irina Blok. Tim desain ditugaskan dengan sebuah proyek untuk membuat sebuah ikon universal yang mudah dikenali dengan menyertakan ikon robot secara spesifik dalam desain akhir. Setelah sejumlah perkembangan desain yang didasarkan pada tema-tema fiksi ilmiah dan film luar angkasa, tim akhirnya mendapat inspirasi dari simbol manusia yang terdapat di pintu toilet, dan memodifikasi bentuknya menjadi bentuk robot. Karena Android adalah perangkat lunak sumber terbuka, disepakati bahwa logo tersebut juga harus terbuka, dan sejak diluncurkan, logo hijau tersebut telah didesain ulang kembali dalam berbagai variasi yang tak terhitung jumlahnya.

2.7.3 Macam-Macam OS (*Operating System*) Android

1. Tanpa nama (Android versi 1.0.1.1)

Sistem android versi 1.0 ini pertama kali hadir di tahun 2008, tepatnya pada oktober 2008. Ponsel pertama yang menggunakan system android adalah HTC. Pada bulan februari 2009 rilis update pertama android versi 1.1 yang masih belum memiliki nama.

Dan untuk android versi selanjutnya google memuskan untuk memberi nama versi android dengan nama makanan ringan, tujuannya adalah agar mudah di ingat oleh para pengguna dan pencinta android.

2. *Cupcake* (Android versi 1.2 - 1.5)



Gambar 2.2 versi android *cupcake*

Nama *Cupcake* diambil dari sebuah makanan ringan berupa kue kecil yang di kemas dalam sebuah wadah yang biasanya di sajikan dengan frosting di atasnya. Dengan penamaan *Cupcake* ini maka penamaan dari versi Android dimulai.

2. *Donut* (Android versi 1.6)



Gambar 2.3 versi android *donut*

Versi selanjutnya adalah *Donut*, versi ini dirilis pada bulan september 2009 tentu donut sudah tak asing lagi bagi telinga orang Indonesia. Versi android ini memiliki fitur foto dan video dari kamera antarmuka dan integrasi pencarian yang lebih baik. Selain itu ditambahkan juga dukungan untuk ukuran layar yang lebih besar, dan diberi versi awal fitur navigasi turn-by-turn besutan Google.

3. *Eclair* (Android versi 2.0 - 2.1)



Gambar 2.4 versi android *eclair*

Eclair adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya. Di rilis pada bulan oktober 2009, dengan rilisnya android versi ini google menambahkan fitur *Bluetooth 2.1*, flash dan kamera dengan digital *zoom*, *multi-touch*, *live wallpaper* dan lainnya.

4. *Froyo* (Android versi 2.2 - 2.2.3)



Gambar 2.5 versi android *froyo*

Nama *Froyo* diambil dari kependekan dari *Frozen Yoghurt* yaitu yoghurt yang telah mengalami proses pendinginan sehingga terlihat seperti es krim. Versi ini dirilis pada bulan Mei 2010 dengan update memperbaiki segi kecepatan dan pengadopsian *Javascript* dari browser *Google Chrome* dengan fitur-fiturnya.

5. *Gingerbread* (Android versi 2.3 - 2.4)



Gambar 2.6 versi android *gingerbread*

Nama *Gingerbread* diambil dari jahe atau *cookie* dengan rasa khas jahe yang biasanya berbentuk boneka mirip manusia. versi ini dirilis pada bulan desember 2010. Smartphone pertama yang memakai versi android ini adalah Nexus S yang dikeluarkan oleh produsen Samsung. Android 2.3 *Gingerbread* ini merupakan OS Android yang paling lama berkuasa bahkan sampai sekarangpun beberapa vendor masih mengeluarkan ponsel dengan versi ini. Adapun perbaikan di versi *Gingerbread* ini adalah tambahan fitur dukungan untuk SIP internet calling, kemampuan nirkabel NFC, dukungan untuk dual kamera, dukungan untuk sensor *giroskop* dan sensor lainnya, *fitur download manager*, sejumlah *tweak* untuk penggunaan di Tablet, dan lainnya.

6. *Honeycomb* (Android versi 3.0 - 3.2)



Gambar 2.7 versi android *honeycomb*

Nama *Honeycomb* diambil dari nama sereal manis yang terbuat dari jagung dengan rasa madu yang berbentuk sarang lebah, makanan ini yang populer sejak tahun 1965. Untuk versi ini merupakan versi yang ditujukan untuk gadget Tablet. Android *Honeycomb* rilis pada Februari 2011, kemudian *upgrade* ke versi 3.1 dan 3.2.

7. *Ice Cream Sandwich* (Android versi 4.0)



Gambar 2.8 versi android *ice cream sandwich*

Android *Ice Cream Sandwich* atau biasa dikenal dengan nama Android 4.0 *ICS* adalah versi terbaru Android yang sangat mendukung baik untuk *smartphone*, tablet, dan lainnya. Android ini rilis pada 19 October 2011. Yang baru dalam android ini adalah perubahan *interface* dari android sebelumnya, antara lain pengoptimalan multitasking, variasi layar beranda yang bisa disesuaikan dan interaktivitas mendalam serta cara baru yang ampuh untuk berkomunikasi dan berbagi konten. Nama Ice Cream Sandwich diambil dari makanan dimana lapisan es krim yang biasanya berupa vanila yang terjepit antara dua *cookies* coklat, dan biasanya berbentuk persegi panjang.

8. Jelly Bean (Android versi 4.1 - 4.3)



Gambar 2.9 versi android *jelly bean*

Android 4.1 Jelly Bean diumumkan pada 27 Juni 2012 pada konferensi Google I/O yang secara resmi dikenalkan ke publik sekitar Oktober 2012. Versi ini adalah yang tercepat dan terhalus dari semua versi Android. Fitur baru yang terapat di versi ini adalah meningkatkan kemudahan dan keindahan tampilan dari *Ice Cream Sandwich* dan memperkenalkan pengalaman pencarian Google yang baru di Android. Android 4.2 Jelly Bean juga menawarkan peningkatkan kecepatan dan kemudahan Android 4.1 serta mencakup semua fitur baru seperti Photo Sphere dan desain baru aplikasi kamera, keyboard *Gesture Typing*, *Google Now* dan lainnya. Untuk nama *Jelly Bean* pasti semua tahu kan, ya nama ini diambil dari sejenis permen yang juga populer disebut kacang jeli.

9. ***KitKat*** (Android versi 4.4)



Gambar 2.10 versi android *kitkat*

Awalnya android versi ini di isukan bernama *Key Lime Pie*. Namun pada tanggal oktober 2013 google merilis kitkat sebagai generasi android berikutnya. Android versi ini memiliki banyak fitur & semakin memanjakan para pengguna android. Diantaranya : *Immersive mode*, Akses kontak langsung dari aplikasi telepon, *google now launcher*, dan pastinya memiliki *interface UI* yang baru.

10. ***Lollipop*** (Android versi 5.0)



Gambar 2.11 versi android *lollipop*

Android 5.0 merupakan versi paling baru dari sistem operasi android. Android 5.0 sendiri dianggap membawa update yang fantastis, banyak perubahan yang disertakan Google di dalamnya.

11. *Marshmallow* (Android versi 6.0)



Gambar 2.12 versi android *marshmallow*

Google secara resmi telah mengumumkan seri OS terbarunya yang diawali dengan inisial M itu. Dan akhirnya Android M atau Android 6.0 ini diberi nama Android Marshmallow. Jika Android Lollipop muncul dengan fokus terhadap update dari sisi desain, *Android Marshmallow* ini akan lebih memfokuskan peningkatan terhadap fitur yang telah ada sebelumnya dan menghilangkan bug yang muncul pada seri android sebelumnya.

2.8 Pengertian Bahasa Pemrograman Java

Menurut Gosling et al. (2008, p1), Java adalah bahasa yang konkuren, berbasis kelas, dan bahasa yang berorientasi objek, yang dirancang sederhana yang banyak

programmer dapat mencapai kefasihan dalam bahasa. Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi seperti desktop, web, dsb. Aplikasi dengan teknologi Java secara umum adalah aplikasi serba guna yang dapat dijalankan pada seluruh mesin yang memiliki Java Runtime Environment (JRE).

2.9 Pengertian *Extensible Markup Language* (XML)

Menurut Hunter et al. (2007, p3), Extensible Markup Language (XML) merupakan teknologi dengan aplikasi dunia nyata, khususnya untuk manajemen, tampilan, dan organisasi data. XML bekerja dengan tujuan markup dari setiap jenis data tetapi dengan kompleksitas yang di eliminasi, XML tidak benar – benar merupakan bahasa, tetapi lebih pada sintaks yang digunakan untuk menjelaskan markup lain. Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur, serta menyediakan format tertentu untuk dokumen – dokumen yang mempunyai data terstruktur.

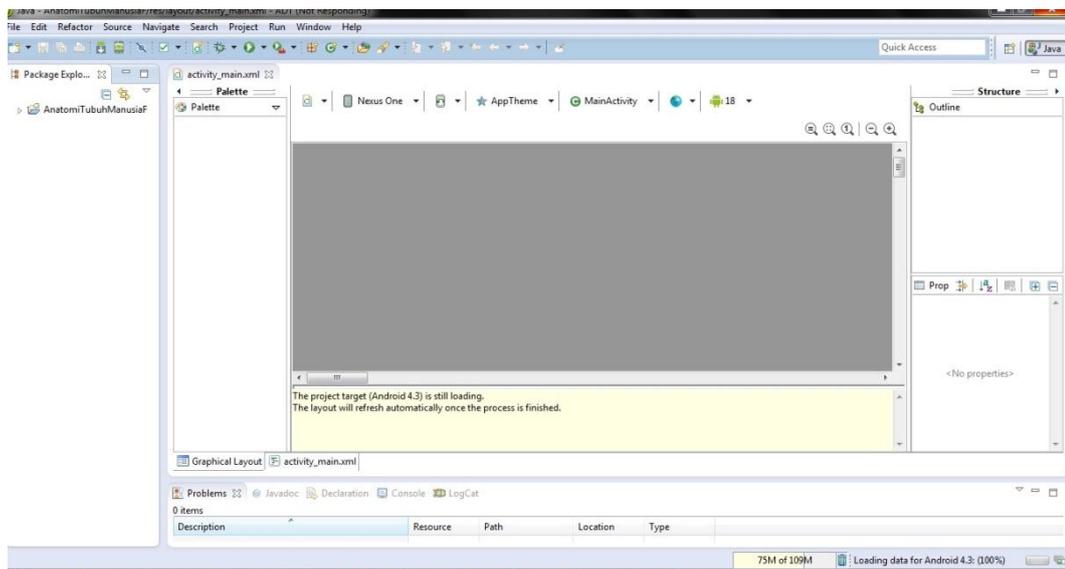
2.10 Perangkat Lunak Pendukung

- *JDK (Java Development Kit)*

Menurut Abdul Kadir (2013, p.4) *Java Development Kit* (JDK) adalah perangkat pengembangan aplikasi java yang bisa diunduh secara gratis di www.oracle.com/technetwork/java/javanes/download/. Perangkat ini mutlak diperlukan untuk membuat aplikasi android, mengingat aplikasi android itu berbasis java. Sebagaimana diketahui, java adalah salah satu bahasa pemrograman yang biasa digunakan untuk membuat aplikasi. Namun perlu diketahui, tidak semua pustaka dalam java digunakan di android.

- Eclipse

Menurut Abdul Kadir (2013, p.5) Eclipse adalah perangkat pengembangan aplikasi yang tergolong sebagai IDE (integrated development environment), karena menyediakan berbagai fasilitas untuk membuat aplikasi. Perangkat lunak ini dapat digunakan sebagai peranti pengembangan aplikasi yang menggunakan bahasa seperti java, C++ dan python. Dengan menggunakan IDE inilah aplikasi android dibangun.



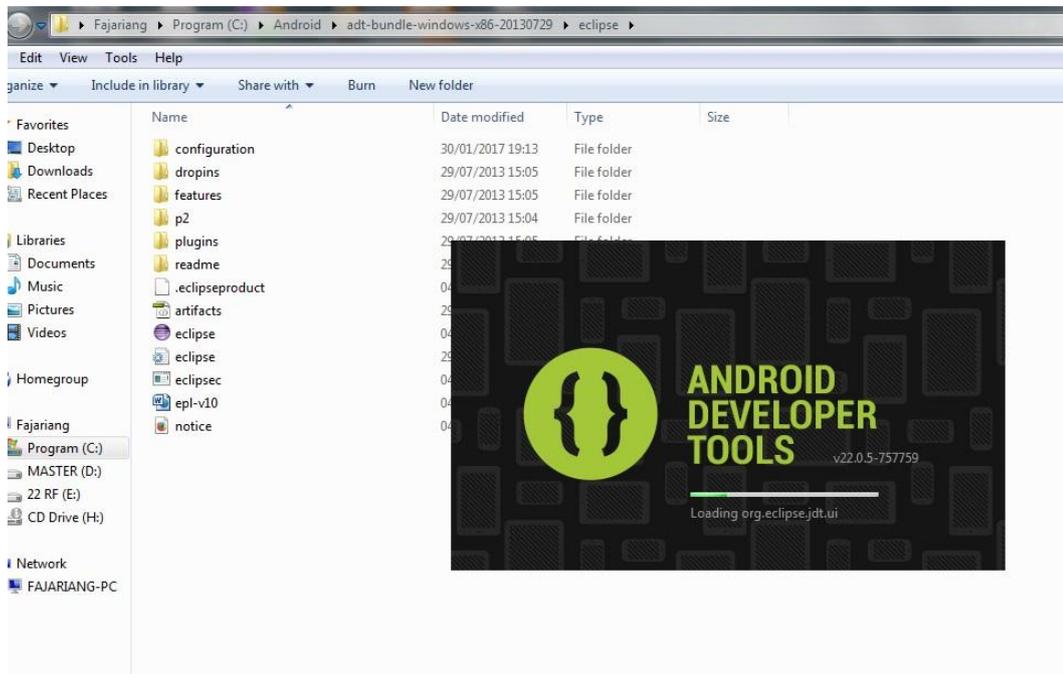
Gambar 2.13 Tampilan Awal Eclipse

- Android SDK

Menurut Abdul Kadir (2013, p.5) Android SDK adalah kumpulan software yang berisi mengenai pustaka, debugger (alat pencari kesalahan program), emulator (peniru perangkat bergerak), dokumentasi, kode contoh, dan panduan.

- *Android Development Tools*

Menurut Abdul Kadir (2013, p.6) *Android Development Tools* (ADT) adalah plugin untuk eclipse IDE yang memungkinkan eclipse digunakan untuk mengembangkan aplikasi android. ADT inilah yang membuat pengembangan aplikasi android dapat dilakukan dengan mudah.



Gambar 2.14 Tampilan ADT *Android Developer Tools*.

- *SQLite*

Menurut Abdul Kadir (2013, p.264) digunakan untuk pengaksesan database yang menggunakan *Structured Query Language* (SQL) yang bersifat relasional.