



BAB II

LANDASAN TEORI

BAB II

LANDASAN TEORI

II.1. Kecerdasan Buatan

Kecerdasan buatan atau *artificial intelligence* adalah salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. Lebih detailnya, kecerdasan buatan dapat dipandang dari berbagai sudut pandang, antara lain :

a. Sudut pandang kecerdasan

Kecerdasan buatan akan membuat mesin menjadi cerdas (mampu berbuat seperti yang dilakukan oleh manusia).

b. Sudut pandang penelitian

Kecerdasan buatan adalah suatu studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia.

c. Sudut pandang bisnis

Kecerdasan buatan adalah kumpulan peralatan yang sangat powerful dan metodologis dalam menyelesaikan masalah-masalah bisnis.

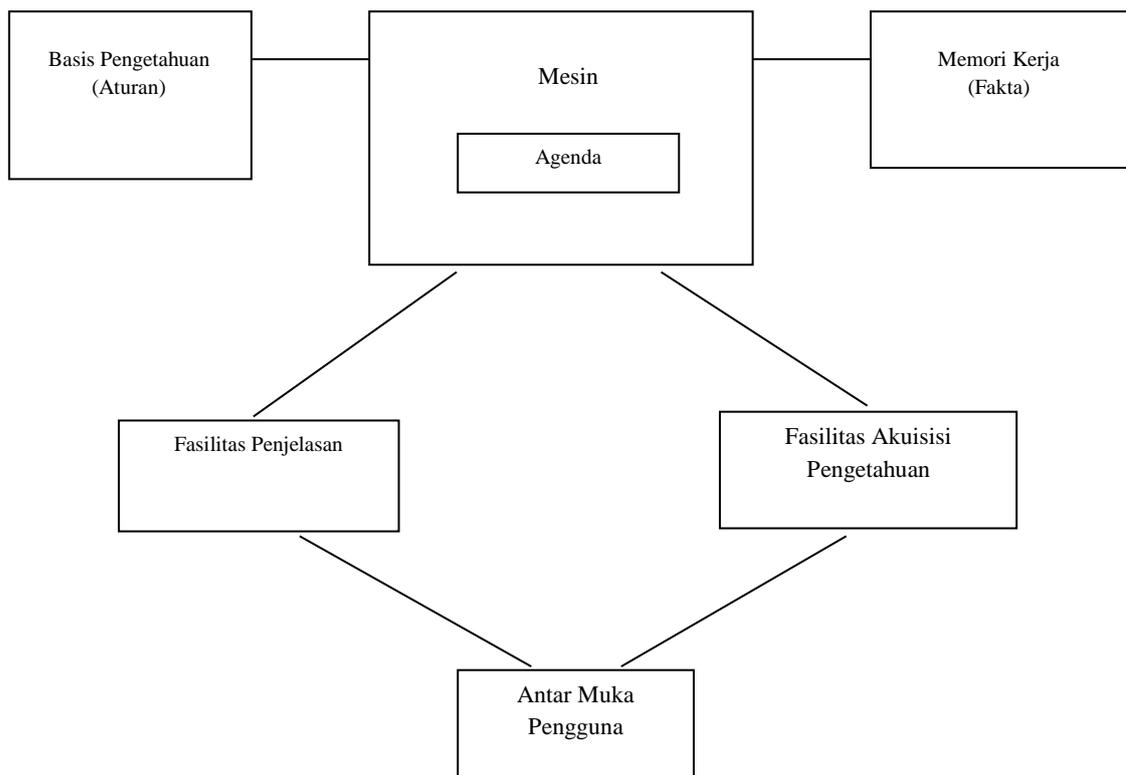
d. Sudut pandang pemrograman

Kecerdasan buatan meliputi studi tentang pemrograman simbolik, penyelesaian masalah dan pencarian (sri kusuma;2003:3).

II.2. Sistem Pakar

Sistem pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek (*emulates*) kemampuan pengambilan keputusan (*decision making*) seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar untuk memecahkan masalah (Rika Rosnelly;2011:2).

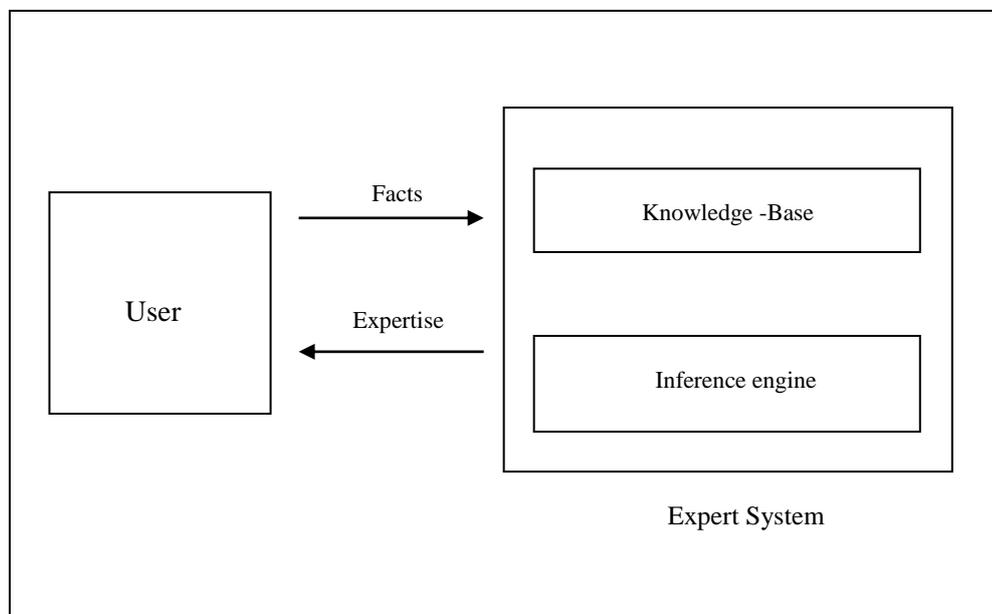
Dapat dilihat dari memori kerja dalam arsitektur sistem pakar merupakan bagian dari sistem pakar yang berisi fakta – fakta masalah yang ditemukan dalam suatu sesi, berisis fakta – fakta tentang suatu masalah yang ditemukan dalam proses konsultasi. Berikut gambar arsitektur sistem pakar. Dapat dilihat pada gambar II.1.



Gambar II.1. Arsitektur Sistem Pakar
(Sumber : Kusri; 2007:19)

Menurut Muhammad Arhami dalam bukunya yang berjudul Konsep Dasar Sistem Pakar, ada beberapa definisi sistem pakar, antara lain (Muhammad Arhami, 2005) :

1. Sistem pakar adalah salah satu cabang dari *Artificial Intelligence* (AI) yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar.
2. Suatu sistem pakar adalah suatu sistem komputer yang menyamai (*emulates*) kemampuan pengambilan keputusan dari seorang pakar.
3. Sistem pakar (*expert system*) merupakan paket perangkat lunak atau paket program komputer yang ditujukan sebagai penyedia nasihat dan sarana bantuan dalam memecahkan masalah di bidang – bidang spesialisasi tertentu.



Gambar II.2. Konsep Dasar Fungsi Sistem Pakar Berbasis Pengetahuan
(Sumber : Rika Rosnelly; 2011:4)

Pada gambar II.2 mengilustrasikan konsep dasar sistem pakar berbasispengetahuan (*knowledge based expert system*). User memberikan informasi atau fakta kepada sistem dan menerima respon berupa saran ahli (*advice/ expertise*).

Secara *internal*, sistem terdiri dari dua komponen utama yaitu basis pengetahuan (*knowledge based*), berisi pengetahuan yang akan digunakan oleh komponen lainnya yaitu mesin infrensi (*inference engine*) untuk menghasilkan kesimpulan sebagai respon terhadap kueri yang dilakukan user (sumber).

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihanannya, adapun kelebihan dari sistem pakar tersebut, seperti:

- Meningkatkan ketersediaan kepakaran atau keahlian menjadi tersedia dalam sistem computer.
- Menurangi biaya. Biaya yang diperlukan untuk menyediakan keahlian per satu orang user menjadi berkurang.
- Mengurangi bahaya. Sistem pakar dapat digunakan dilingkungan yang mungkin berbahaya bagi manusia.
- Permanen. sistem pakar dan pengetahuan yang terdapat didalam bersifat lebih permanen,dibandingkan pengetahuan yang hilang saat sang pakar mninggal dunia.
- Keahlian multiple. Bekerja secara simultan dan kontinyu dalam menyelesaikan masalah setiap saat.
- Meningkatkan kehandalan. Sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternative.

- Penjelasan detail proses penalaran yang dilakukan hingga mencapai suatu kesimpulan.
- Dapat memberikan respon yang cepat.
- Stabil, tidak emosional, dan memberikan respon yang lengkap setiap saat
- Pengaksesan basis data secara cerdas (sumber).

II.3. UML

Menurut Windu Gata (2013 : 4) hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

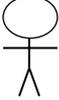
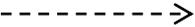
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *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. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di

dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.1. Simbol Use Case

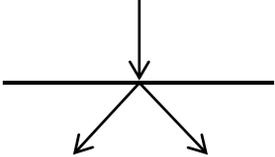
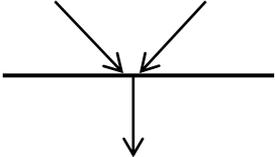
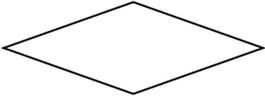
Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(Sumber : Windu Gata; 2013: 4)

2. Activity Diagram (Diagram Aktivitas)

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

Tabel II.2. Simbol Activity Diagram

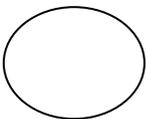
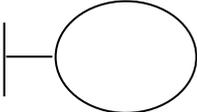
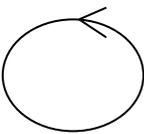
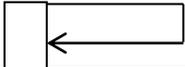
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata; 2013: 6)

3. *Sequence Diagram* (Diagram Urutan)

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

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata; 2013: 7)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata; 2013: 9)

II.4. Database

Terkait dengan database terdapat istilah dasar yang disebut data. Awalnya data menyatakan fakta yang dapat direkam dan disimpan pada media computer, misalnya: hardisk. Dari pernyataan tersebut, muncul beberapa definisi terkait database. Berikut pengertian dari database tersebut pada tabel II.5.

Tabel II.5. Definisi Database

Definisi Database	Sumber
Kumpulan data yang terkait	Elmazri dan Navathe, 1994
Kumpulan yang terorganisasi dari data – data yang secara nalar terkait	Hoffer,dkk,2005
Kumpulan data yang umumnya menjabarkan aktivitas – aktivitas dari satu atau lebih dari satu organisasi terkait	Ramakrishnan dan Gerke,2000

(Sumber : Abdul Kadir;2008:9)

II.5. Sql Lite

Sql Lite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard hip. Tidak seperti paradigma client server sebelumnya, inti sql lite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagian integral dari sebuah program secara keseluruhan. Sehingga protocol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemograman (Dimas Radityo,2012).

II.5. Naïve Bayes

Naïve Bayes Classifier adalah sebuah metode *information retrieval* yang menggunakan pendekatan probabilistik dalam menginfrensi, yakni berbasis pada teorema bayes secara umum. Aplikasi yang paling banyak digunakan yaitu untuk klasifikasi teks. Tujuan dari asumsi independensi posisi kata adalah untuk mengurangi kompleksitas dalam perhitungan nilai-nilai probabilitas bersyarat pada teorema bayes secara umum (Ricky Yuda,2012).

Sebuah *bayes classifier* adalah *classifier* probabilistik sederhana berdasarkan penerapan teorema bayes (dari *statistic Bayesian*) dengan asumsi indepeden(naïf) yang kuat. Sebuah istilah yang lebih deskriptif untuk model

probabilistik yang digaris bawahi adalah “model fitur independen”. Dalam terminologi sederhana, sebuah mbc mengasumsikan bahwa kehadiran (atau ketiadaan) fitur tertentu dari suatu kelas tidak berhubungan dengan kehadiran (atau ketiadaan) fitur lainnya.

Sebagai contoh, buah mungkin dianggap apel jika, merah, bulat dan berdiameter sekitar 4 inchi. Bahkan, jika fitur inibergantung satu sama lain atau atas keberadaan fitur lain, sebuah NBC menganggap bahwa seluruh sifat-sifat berkontribusi mandiri untuk probabilitas bahwa buah ini adalah apel.

Dibalik desain naifnya dan asumsi yang tampak disederhanakan, NBC telah bekerja cukup baik dalam banyak situasi dunia nyata yang kompleks. Sebuah keuntungan dari NBC adalah bahwa ia hanya memerlukan sejumlah kecil data pelatihan untuk mengestimasi parameter (rata-rata dan variabel dari variable) yang diperlukan untuk klasifikasi. Variabel diasumsikan independe, hanya varian dari variabel – variabel untuk setiap kelas yang perlu ditentukan dan bukan keseluruhan covariance matrix (Ni Wayan Sumartini;2011:37)

II.5.1. Model Probabilitas Naïve Bayes Clasifier

Model probabilitas untuk clasifier adalah model kondisional

$$p(C | F_1, \dots, F_n) \dots\dots\dots (1)$$

Terhadap variable kelas depende C dengan sejumlah kecil hasil atau kelas, tergantung pada beberapa variable fitur F1 sampai Fn. Masalahnya adalah bahwa jika sejumlah fitur n besar atau bila fitur bisa mengambil sejumlah nilai besar, maka membuat sebuah model pada table probabilitas adalah tidak mungkin. Maka

diformulasikan model untuk yang lebih fleksibel dengan menggunakan teorema bayes, sebagai berikut:

$$p(C | F_1, \dots, F_n) = \frac{p(C)p(F_1, \dots, F_n | C)}{p(F_1, \dots, F_n)} \dots\dots\dots (2)$$

Dalam bahasa inggris dapat ditulis sebagai berikut:

$$posterior = \frac{Prior \times Likelihood}{Evidence} \dots\dots\dots (3)$$

Pada hakikatnya, pembilang ini setara dengan model probabilitas gabungan $p(C|F_1, \dots, F_n)$ yang dapat ditulis seperti berikut menggunakan penggunaan berulang dari definisi probabilitas bersyarat:

$$\begin{aligned} & p(C | F_1, \dots, F_n) \\ & p(C)p(F_1, \dots, F_n | C) \\ & p(C)p(F_1 | C)p(F_2, \dots, F_n | C, F_1) \\ & p(C)p(F_1 | C)p(F_2 | C, F_1)p(F_3, \dots, F_n | C, F_1, F_2) \\ & p(C)p(F_1 | C)p(F_2 | C, F_1)p(F_3 | C, F_1, F_2)p(F_4, \dots, F_n | C, F_1, F_2, F_3) \\ & p(C)p(F_1 | C)p(F_2 | C, F_1) \dots p(F_n | C, F_1, F_2, F_3, \dots, F_{n-1}) \dots\dots\dots (4) \end{aligned}$$

Sekarang asumsi kemandirian bersyarat yang naif memegang peranan , bahwa setiap fitur F_i adalah secara kondisi independen terhadap setiap fitur lainnya F_j untuk j tidak sama dengan i , hal ini menyatakan bahwa :

$$\begin{aligned} & p(F_i | C, F_j) = p(F_i | C) \\ & i \neq j \dots\dots\dots (5) \end{aligned}$$

Untuk sehingga gabungan model dapat dinyatakan sebagai berikut.

$$\begin{aligned}
 p(C | F_1, \dots, F_n) &= p(C)p(F_1 | C)p(F_2 | C)p(F_3 | C)\dots \\
 &= p(C) \prod_{i=1}^n p(F_i | C) \dots \dots \dots (6)
 \end{aligned}$$

Ini berarti bahwa dibawah asumsi independen diatas, distribusi bersyarat dari variable kelas C dapat dinyatakan sebagai berikut :

$$p(C | F_1, \dots, F_n) = \frac{1}{z} p(C) \prod_{i=1}^n p(F_i | C) \dots \dots \dots (7)$$

dimana z (bukti) adalah factor skala tergantung hanya padayaitu, sebuah konstanta jika nilai dari variable diketahui (Ni Wayan Sumartini;2011:40-41).

Modal dari bentuk ini jauh lebih mudah dikelola, karena mereka memecah menjadi kelas prior $p(C)$ dan distribusi probabilitas independence $p(F_i|C)$. jika ada kelas K dan jika model untuk masing – masing $p(F_i|C=C)$ dapat dinyatakan dalam bentuk parameter, maka model naïf Bayes yang sesuai memiliki $(K-1)+n$ K parameter. Dalam prakteknya $K=2$ (Klasifikasi Biner) dan $r=1$ (Variabel bernaulli sebagai fitur).

Sehingga jumlah parameter model bayes naïf adalah $2n+1$, dimana n adalah jumlah fitur biner yang digunakan untuk klasifikasi dan prediksi (Ni Wayan Sumartini;2011:42).

II.6. Kanker Mulut

Kanker mulut adalah kanker yang menyerang bagian – bagian mulut. Kanker ini dapat tumbuh di gusi, bibir, dinding mulut, lidah, langit- langit mulut atau dasar mulut. Kanker mulut adalah kanker yang berasal dari jaringan epitel, baik mukosa atau kelenjar liur pada dinding rongga mulut dan organ dalam

mulut.karsinoma (kanker) mulut merupakan kanker nomor enam paling banyak didunia dan mencakup 30% dari kanker kepala dan leher.

Di amerika serikat pada tahun 2005, di perkirakan sedikitnya terdapat 20.000 kasus kanker mulut, dan sekitar 5000 diantaranya menyebabkan kematian. Sementara insiden kanker mulut di Indonesia belum diketahui dengan pasti, frekuensi relatif diperkirakan 1,5%-5% dari seluruh kanker.

Terdapat beberapa jenis kanker mulut yang sering ditemukan, yaitu :

1. Kanker Lidah

Sebagian besar (40%) dari kanker mulut adalah kanker lidah.Tanda awal kanker jenis ini umumnya berupa ulkus tanpa nyeri yang tidak sembuh-sembuh. Kemudian membesar dan menekan atau menginfiltrasi jaringan sekitar yang mengakibatkan nyeri lokal , *otalgia ipsilateral*, dan nyeri *mandibula*.

2. Kanker Dasar Mulut (*Fom*)

Kanker dasar mulut merupakan 10 % -15% dari rongga mulut.lesi umumnya adalah karsinoma sel skuamosa dengan differensiasi ringan sampai sedang. Terkadang juga muncul dari kelenjar liur berupa *adenaokarsinoma*, *adenoid*, *kististik*, *karsinoma* dan *mucoepidermoid karsinoma*(Suyatno; 2009:101-107). (Suyatno; 2009:101).

3. Kanker Bibir

Kanker yang tersering ini adalah karsinoma sel skuamous dan basalioma, melanoma maligna dan kanker kelenjar liur minor. Terdapat tiga tipe karsinoma sel skuamose bibir yakni: *eksofitik*, *endofitik* dan *verrucous* (jarang).

4. Kanker *Palatum Durum*

Tampilan klinis karsinoma sel skuamos palatum adalah berupa lesi ulkus, umumnya *asimptomatis* pada stadium dini dan sangat nyeri pada stadium lanjut. Gambaran lain adalah masa dipalatum, berdarah, *foul*, *odor* (bau mulut) dan gigi tanggal.

5. Kanker *Gingiva*

Kanker *gingiva* sering terlihat sebagai perubahan mukosa yang disertai leukoplakia. Tumor lebih *extensive* akan mengakibatkan gigi goyang, berdarah atau nyeri, kemudian menginvasi tulang disekitarnya (Suyatno; 2009:101-103).

6. Kanker *Trigonum Retromolar*

Kanker *trigonum retromolar* adalah sejenis kanker mulut yang berada di area bawah mulut. Kanker ini dapat menyebabkan *trismus* karena terlibatnya *pterygomandibular space*, *pterygoid* dan otot-otot bukinator.

7. Kanker *Mukosabukal*

Pada stadium dini *asimtopatis* atau teraba oleh lidah. Ulserasi dapat menyebabkan nyeri lokal. *Opstruksi duktus stenson's* dapat menyebabkan pembesaran kelenjar parotis (Suyatno; 2009:101-107).

II.7. Android

Android adalah sebuah nama untuk sistem operasi pada suatu gadget seperti computer tablet, smartphome, dan telepon selular. Sistem operasi yang digunakan berbasis linux.

Simbol atau logo android berbentuk robot dengan dua antena dikepalanya, ini melambangkan bahwa android merupakan simbolisasi dari sistem operasi kelas atas untuk gadget dan smartphome.



Gambar II.3. Logo Android
Sumber : Google.com

II.7.1. Sejarah android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google Inc.* membeli *Android Inc.* yang merupakan pendatang baru membuat piranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan android, dibutuhkan *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi (Nazruddin; 2012: 1).

II.7.2. Android SDK (Software Development Kit)

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mulai mengembangkan bahasa pemrograman Java. Saat ini disediakan android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform android* menggunakan bahasa pemrograman Java. Beberapa fitur-fitur android yang paling penting adalah:

- *Framework* aplikasi yang mendukung penggantian komponen dan reusable
- Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile*
- Integrated browser berdasarkan engine open source *WebKit*
- *Grafis* yang dioptimalkan dan didukung oleh *libraries grafis 2D, grafis 3D*.
- *SQLite* untuk penyimpanan data
- *Media support* yang mendukung *audio, video*, dan gambar
- Bluetooth, EDGE, 3G, dan WiFi.

- Lingkungan Development yang lengkap (Nazruddin;2012:5).

II.6.3. ADT (Android Development Tools)

ADT adalah *plug in* yang di desain 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 menambahkan komponen – komponen yang lainnya. Dengan ADT juga kita dapat melakukan pembuatan *package android* (.apk) yang digunakan untuk distribusi aplikasi *android* yang kita rancang (Nazruddin;2012:6).

II.7.4. Arsitektur Android

Arsitektur Android dibangun dengan menggunakan asas object oriented, dimana elemen-elemen penyusun sistem operasinya berupa objek yang dapat kita gunakan kembali/reusable. Agar bisa membuat aplikasi dengan baik, tentunya kita harus mengetahui arsitektur OS Android beserta elemen elemennya.

Secara garis besar arsitektur android dapat dijelaskan sebagai berikut:

1. *Application and widgets*

Application and widgets ini adalah *layer* dimana kita berhubungan dengan aplikasi apa saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut.

2. *Application Framework*

Adalah “*Open Development Platform*” yaitu *android* menawarkan kepada pengembang untuk membangun aplikasi yang bagus dan *inovatif*. Komponen yang termasuk didalam *application frameworks* adalah sebagai berikut:

- *Views*
- *Content Provider*
- *Resource Manager*
- *Notification Manager*
- *Activity Manager*

3. *Libraries*

Adalah *layer* dimana fitur-fitur *android* berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

4. *Android runtime*

Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi linux.

5. *Linux kernel*

Adalah *layer* dimana inti dari operating sistem dari android itu berada. Berisi *file-file* sistem yang mengatur sistem *prossesing*, *memory*, *resource*, *drivers* dan sistem operasi android lainnya (Nazruddin;2012:8).

II.7.5. Versi Android

1. Android versi 1.1

Pada 9 maret 2009, google merilis android versi 1.1. android 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*)

Pada pertengahan mei 2009, google kembali merilis dengan menggunakan android dan SDK terdapat beberapa fitur dalam versi ini yaitu kemampuan menonton video, merekam, mengupload video ke *Youtube* dan lain- lain.

3. Android versi 1.6 (*Donut*)

Pada android versi ini fitur baru yang dapat ditemukan yaitu *camcorder*, galeri yang diintegrasikan, *CDMA/EVDO*, *VPN*, *Gestures*, *text to speech engine* dan kemampuan dial kontak.

4. Android versi 2.0/2.1 (*Éclair*)

Pada versi ini, perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan google map 3.1.2, perubahan UI dengan *browser* baru, *digital zoom* dan *Bluetooth 2.1*.

5. Android versi 2.2 (*Froyo*)

Fitur yang tersedia pada android *frozen yogurt* ini diantaranya, adalah:

Kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia.

6. Android versi 2.3 (*gingerbread*)

Berikut hal – hal yang direvisi dari versi sebelumnya adalah kemampuan seperti: *SIP-based VoIP*, *Gyroscope* dan *Sensor*, *Multiple Camera Support*, *Download Manager*.

7. Android versi 3.0 (*Honey comb*)

Android versi ini adalah versi android yang di khususkan bagi pengguna tablet. Sistem ini disesuaikan bagi pengguna tablet agar tablet memiliki inerja yang maksimal.

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

Android yang disingkat dengan ICS ini adalah android jenis pertama yang mempunyai fitur baru membuka kunci dengan pengenalan wajah. Dan android jenis ini juga diklaim menghasilkan interface yang lebih halus dan bersih.

9. Android versi 4.1 (*Jelly Bean*)

Jenis android jelly bean diketahui memiliki fitur terbaru yng menawan antara lain, pencarian dengan voice search lebih cepat, informasi cuaca, lalu lintas dan mempunyai fitur keyboard virtual yang lebih baik.

10. Android 4.2 (*Jelly Bean*)

Android versi 4.2 ini merupakan versi terbaru dari versi versi android sebelumnya. Jelly bean v.4.2 diklaim lebih pintar dan inovatif dibandingkan dengan pendahulunya. Beberapa fitur yang diperbaharui dalam system informasi operasi ini , antara lain : notification, google assist, face unlock, barell roll, smart widgets dan lain – lain.

11. Android 4.3 (Jelly Bean)

12. Android versi 4.4 (KitKat)

Fitur terbarunya antara lain SMS yang terintegrasi langsung kedalam google hangouts, terdapat aplikasi *cloud printing*, mendengarkan perintah suara dari Google Now tanpa menguras baterai, interface yang sangat halus dan bisa mengakses aplikasi kamera dari layar terkunci.

II.6.6. Bahasa Pemrograman pada Android

Berbeda dengan Java ME (*Micro Edition*) *device* yang hanya mungkin di program menggunakan Java. Sedangkan pada program *Android* bisa dibuat dengan beberapa macam bahasa pemrograman, bahasa resmi yang didukung adalah Java dan XML.

- a. XML (*Extensible Markup Language*) dipakai untuk membuat *interface* aplikasi, yang seperti bahasa *markup* HTML yang dirancang untuk membawa data, bukan untuk menampilkan data. *Tag* XML tidak ditentukan sebelumnya melainkan didefinisikan sendiri. XML dirancang untuk menjadi *self-descriptive*
- b. *Java* dipakai untuk menangani *Logic dan event*. *Java* juga bisa dipakai untuk membuat *User interface*, namun tidak disarankan, kecuali untuk membuat komponen yang digenerate secara dinamis.