

BAB II

LANDASAN TEORI

2.1 Perancangan Sistem

(Fitria, Kenali, & Sari, 2008) Perancangan sistem adalah suatu fase dimana diperlukan suatu keahlian perancangan untuk element-element komputer yang akan menggunakan sistem yaitu pemilihan peralatan dan program komputer untuk sistem yang baru. Adapun tujuan yang hendak dicapai dari tahap perancangan sistem mempunyai maksud atau tujuan utama, yaitu untuk memenuhi kebutuhan pemakaian sistem atau (user), untuk memberikan gambaran yang jelas dan menghasilkan rancangan bangun yang lengkap kepada pemrograman komputer dan ahli-ahli teknik lainnya yang terlibat dalam pengembangan atau pembuatan sistem.

2.2 Kecerdasan Buatan

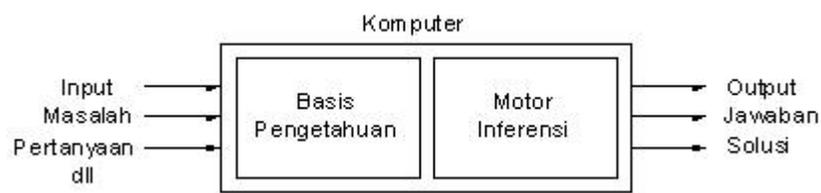
(Dahria, 2008) mendefinisikan Kecerdasan buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia.

Manusia cerdas (pandai) dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu akan lebih mampu menyelesaikan permasalahan.

Demikian juga agar mesin bisa cerdas (bertindak seperti dan sebaik manusia) maka harus diberi bekal pengetahuan, sehingga mempunyai kemampuan untuk menalar. Untuk membuat aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan, yaitu :

1. Basis Pengetahuan (*Knowledge Base*), bersifat fakta-fakta, teori, pemikiran dan hubungan antar satu dengan yang lainnya.
2. Motor Inferensi (*Inference Engine*), kemampuan menarik kesimpulan berdasarkan pengetahuan dan pengalaman.

Penerapan konsep kecerdasan buatan pada komputer adalah sebagai berikut :



Gambar 2.1 Penerapan Konsep Kecerdasan Buatan di Komputer

2.3 Sistem Pakar

(Fitria, Sriyanto, & Chairani, 2012) menjelaskan Sistem pakar merupakan salah satu bagian dari kecerdasan buatan. Sistem ini dirancang untuk menirukan keahlian seorang pakar dalam menjawab pertanyaan dan menyelesaikan suatu permasalahan baik di bidang kesehatan atau kedokteran, bisnis, ekonomi dan sebagainya. Sistem pakar sangat membantu untuk pengambilan keputusan, dimana sistem pakar ini dapat mengumpulkan dan menyimpan pengetahuan dari seseorang atau beberapa orang pakar dalam suatu basis pengetahuan (knowledge base) dan menggunakan sistem penalaran yang menyerupai seorang pakar dalam memecahkan masalah. Jadi, sistem pakar ini dapat memecahkan suatu masalah tertentu karena sudah menyimpan pengetahuan secara keseluruhan.

(Veradani, 2014) mendefinisikan, istilah sistem pakar (*Expert System*) berasal dari istilah sistem pakar berbasis pengetahuan. Sistem pakar adalah suatu sistem yang menggunakan pengetahuan untuk memecahkan persoalan yang biasanya

membutuhkan keahlian manusia. Sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah.

2.3.1 Keuntungan Sistem Pakar

Ada beberapa keunggulan sistem pakar, diantaranya dapat :

1. Menghimpun data dalam jumlah yang sangat besar.
2. Menyimpan data tersebut untuk jangka waktu yang panjang dalam suatu bentuk tertentu.
3. Mengerjakan perhitungan secara cepat dan tepat dan tanpa jemu mencari kembali data yang tersimpan dengan kecepatan tinggi.

2.3.2 Kekurangan Sistem Pakar

Selain keuntungan-keuntungan diatas, sistem pakar sama halnya seperti sistem lainnya, juga memiliki kekurangan, diantaranya adalah :

1. Masalah dalam mendapatkan pengetahuan dimana pengetahuan tidak selalu bisa didapatkan dengan mudah. Karena kadang kala pakar dari masalah yang kita buat tidak ada, dan walaupun ada kadang-kadang pendekatan yang dimiliki oleh pakar berbeda-beda.
2. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk biaya pengembangan dan pemeliharannya.
3. Boleh jadi sistem tak dapat membuat keputusan.
4. Sistem pakar tidaklah 100% menguntungkan walaupun seorang tetap tidak sempurna atau tidak selalu benar. Oleh karena itu perlu diuji ulang serta teliti sebelum digunakan.

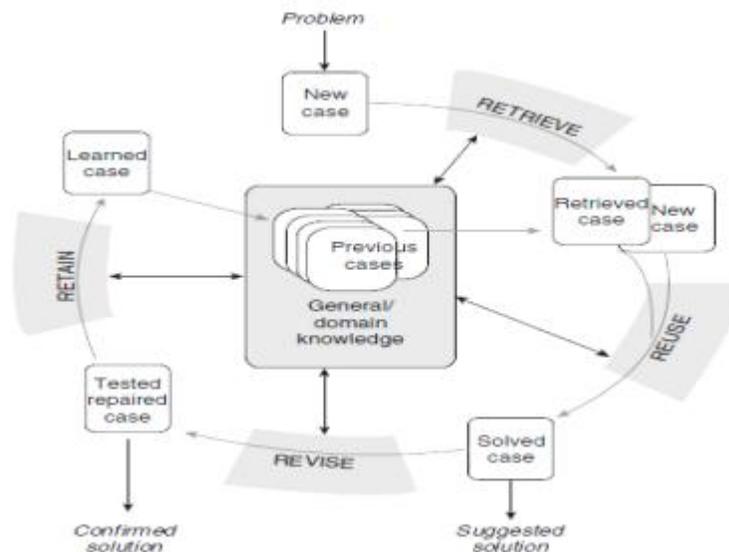
2.4 Case Based Reasoning (CBR)

Sistem *Case Base Reasoning* (CBR) harus melakukan tahapan proses untuk menghasilkan solusi suatu masalah diantaranya adalah mencari tingkat kemiripan kasus baru dengan kasus yang tersimpan.

Menurut (Tursina, 2014) menyatakan. Tahapan proses pada CBR dalam mencari nilai kemiripan dibutuhkan empat (4) tahap, yaitu :

1. *Retrieve* (penelusuran) adalah menemukan kembali kasus yang sama atau yang paling mirip dengan kasus baru.
2. *Reuse* adalah menggunakan kembali informasi dan pengetahuan dari basis kasus untuk memecahkan masalah kasus baru.
3. *Revise* adalah merevisi atau memperbaiki solusi yang diusulkan.
4. *Retain* adalah menyimpan pengalaman untuk memecahkan masalah yang akan datang kedalam basis kasus.

Tahapan proses CBR dapat dilihat pada Gambar 2.2:



Gambar 2.2 Tahapan CBR

Menurut (Tursina, 2014) menyatakan. Berdasarkan tahapan yang ada dalam CBR, diperlukan dua langkah utama dalam menentukan solusi, yaitu:

1. Membangun basis kasus, yang digunakan sebagai tempat penyimpanan.
2. Menentukan fungsi kemiripan (*similarity*), langkah ini digunakan untuk mengenali kesamaan atau kemiripan antara kasus-kasus yang tersimpan dalam basis kasus dengan kasus yang baru.

2.5 Algoritma *Nearest Neighbor*

Menurut Kusriani (2009), "*Nearest Neighbor* adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pencocokan bobot dari sejumlah fitur yang ada".

2.6 Penelusuran Kasus (*Retrieve*)

Menurut Kusriani (2009), Proses *retrieve* (penelusuran kasus) pada sistem penalaran computer berbasis kasus untuk mendeteksi penyakit maag menggunakan algoritma *nearestneighbor*. Algoritma *nearest neighbour* adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur-fitur yang ada.

Adapun rumus untuk melakukan perhitungan kedekatan antara 2 kasus adalah:

$$similarity(T, S) = \frac{\sum_{i=1}^n f(T_i, S_i) \times w_i}{w_i} \dots\dots (1)$$

Keterangan:

T = Kasus baru

S = Kasus yang ada dalam penyimpanan

n = jumlah atribut dalam masing-masing kasus

i = atribut individu antara 1 s/d n

f = fungsi similarity atribut i antara kasus T dan kasus S

w = bobot yang diberikat pada atribut ke i

2.7 MAAG

Menurut (Wardaniati, A, & Dahlan, 2016) Gastritis adalah inflamasi pada lapisan mukosa dan submukosa lambung. Gastritis kronis tingkat ringan samapi sedang sering di temukan pada masyarakat, terutama sekali pada orang dewasa. Inflamasi ini kadang-kadang terjadi superficial atau di permukaan mukosa lambung saja sehingga tidak begitu nyeri, jadi tadak begitu mengganggu. Akan tetapi, bila inflamasi telah mengenai samapi kedalam mukosa lambung, maka akan timbul nyeri di daerah epigastrium. Bila gastritis khronis berlangsung dalam jangka waktu yang lama, maka dapat menyebabkan atropi mukosa lambung beserta kelenjar-kelenjar yang terdapat didalammnya. Namun, kadang- kadang gastritis bisa pula menjadi sanagat akut dan berat dengan ekskoriiasi ulseratif (luka bertukak) mukosa lambung yang disebabkan oleh aktifitas sekresi sel peptik dari lambung sendiri, yaitu berupa enzim pepsin.

2.7.1 Penyebab Penyakit Maag

Beberapa hal penyebab timbulnya penyakit maag, yaitu adalah sebagai berikut :

1. Pengeluaran asam lambung yang berlebihan.
2. Pertahanan dinding lambung yang lemah.
3. Infeksi helicobacter pylori (sejenis bakteri yang hidup di dalam lambung, dalam jumlah kecil) ketika asam lambung yang dihasilkan lebih banyak kemudian pertahanan dinding lambung menjadi lemah.
4. Gangguan gerakan saluran cerna.
5. Stres psikologis.

2.7.2 Jenis Penyakit Maag

Gastritis atau penyakit maag dapat dikelompokkan menjadi tiga kelompok, yaitu adalah sebagai berikut :

1. Gastritis / Dyspepsia.
2. Gastritis Akut, terdiri dari 2 jenis yaitu gastritis eksogenus dan gastritis endogenus.
3. Gastritis Kronis, terdiri dari 3 jenis yaitu gastritis supersifialis, gastritis atropikan dan gastritis hypertropikan.

2.8 Perangkat Lunak Pengembangan Sistem

Agar Perangkat lunak dapat dioperasikan secara optimal maka ada perangkat lunak yang sangat diperlukan antara lain :

2.8.1 *Personal Home Page (PHP)*

Menurut (Ramadhani, Anis, & Masruro, 2013) PHP adalah bahasa pemrograman script yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

Menurut (Ariesna, 2014) menyatakan PHP dikenal sebagai bahasa pemrograman yang dapat menyatu dengan sintak-sintak HTML, dieksekusi di server dan dapat digunakan untuk membuat halaman website yang dinamis.

2.8.2 *Cascading Style Sheet (CSS)*

Menurut (Binarso, Sarwoko, & Bahtiar, 2012) CSS (*Cascading Style Sheet*) adalah *stylesheet language* yang digunakan untuk mendeskripsikan penyajian dari dokumen yang dibuat dalam *mark up language*. CSS merupakan sebuah dokumen yang berguna untuk melakukan pengaturan pada komponen halaman *web*, inti dari dokumen ini adalah memformat

halaman *web* standard menjadi bentuk *web* yang memiliki kualitas yang lebih menarik.

2.8.3 Hypertext Markup Language (HTML)

Menurut (Sasongko, 2017) HTML merupakan file text murni yang dapat dibuat dengan editor text apa saja. Dokumen ini dikenal sebagai *web page*. File-file HTML berisi instruksi-instruksi yang kemudian diterjemahkan oleh browser yang ada dikomputer client sehingga isi formasinya dapat ditampilkan secara visual di komputer pengguna.

2.8.4 MySQL

Menurut (Ramadhani et al., 2013) MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). MySQL juga merupakan sebuah perangkat lunak sistem manajemen basis data SQL yang multithread, multi-user dengan sekitar 6 juta instalasi diseluruh dunia. MySQL bebas digunakan oleh setiap orang, tetapi tidak boleh dijadikan produk turunan yang bersifat komersial.

2.9 Data Flow Diagram (DFD)

(Fitria, Mustika, D., 2011) *Data Flow Diagram* (DFD) dalam bukunya (*Pressman:2002*), yaitu sebuah teknis grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output.

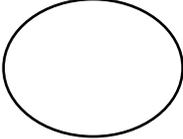
DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level yang lebih detail untuk merepresentasikan 16 aliran informasi lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Oleh karena itu, DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi pada perangkat lunak yang

akan diimplementasikan menggunakan pemrograman terstruktur dengan fungsi dan prosedur.

2.9.1 Context Diagram

(Fitria & Iriyanto, Y.S., 2016) *Contexts diagram* adalah diagram tingkat atas yang digunakan untuk menggambarkan sistem secara garis besar (top level) dan juga merupakan diagram yang tidak detail dari sistem informasi yang menggambarkan kesatuan luar sistem.

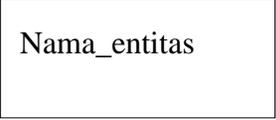
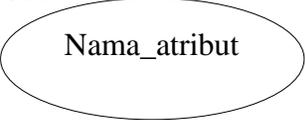
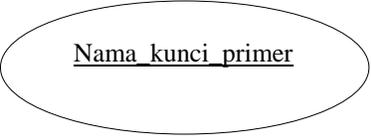
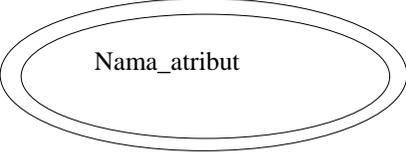
Tabel 2.1 Simbol-simbol DFD

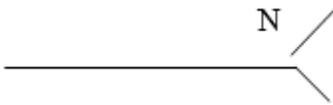
Komponen	Keterangan
	Process merupakan sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi
	Eclips mewakili atribut
	Garis mewakili relasi dan entitas atau relasi dan entitas dengan atribut
	Penyimpanan data dalam database, biasanya berupa tabel

2.10 Entity Relational Diagram (ERD)

Menurut (Edi & Betshani, 2014) Entity Relationship Diagram (ERD) adalah sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas (entity) serta hubungan (relationship) antar entitas-entitas tersebut dengan menggunakan beberapa notasi.

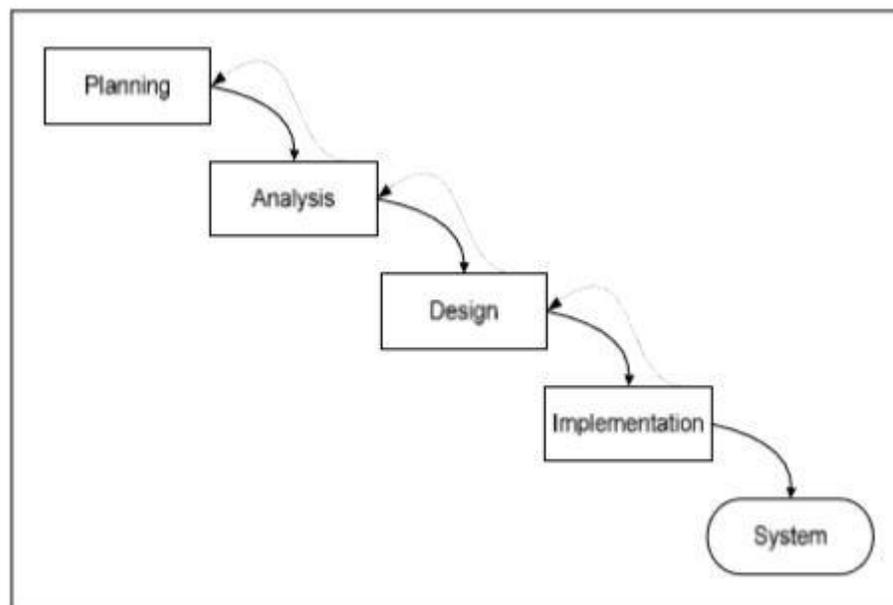
Tabel 2.2 Notasi pada Model ERD

Simbol	Keterangan
<p data-bbox="391 317 570 348">Entitas /entity</p> 	<p data-bbox="831 317 1308 730">Entitas/entity merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.</p>
<p data-bbox="391 783 483 814">Atribut</p> 	<p data-bbox="831 783 1308 919">Atribut atau field; atau kolom data yang butuh disimpan dalam suatu entitas.</p>
<p data-bbox="391 978 667 1010">Atribut Kunci Primer</p> 	<p data-bbox="831 978 1308 1230"><i>Field</i> atau kolom yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id.</p>
<p data-bbox="391 1278 773 1310">Atribut Multi nilai/<i>multivalue</i></p> 	<p data-bbox="831 1278 1308 1415"><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.</p>
<p data-bbox="391 1560 472 1591">Relasi</p> 	<p data-bbox="831 1560 1308 1707">Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.</p>

<p>Asosisasi / Association</p> 	<p>Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian, Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.</p>
--	---

2.11 Metode Pengembangan *Waterfall*

(Fitria, 2015) Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak yang berurutan, di mana proses pengerjaannya terus mengalir dari atas ke bawah (seperti air terjun) melewati fase-fase *Requirements* (analisis kebutuhan), *Design* (perancangan dan pemodelan), *Implementation* (penerapan), *Verification* (pengujian), dan *Maintenance* (pemeliharaan).



Gambar 2.3 Tahapan Metode *Waterfall*

1. **Planning (Perencanaan)**, Pada tahap ini langkah-langkah yang dilakukan adalah dengan menginventarisir kebutuhan data yang akan dibuat menjadi sebuah sistem dan menentukan bagaimana cara membangun sistem tersebut. Langkah langkah tersebut meliputi perlunya dilakukan dan dikembangkan sebuah sistem yang dapat memberikan kemudahan bagi pengguna.
2. **Analysis (Analisis)**, Analisis sistem dilakukan untuk memberikan jawaban pertanyaan siapa yang akan menggunakan sistem yang meliputi kemudahan akses data oleh individu/lembaga/institusi terkait, kemudahan akses data update dan perubahan data dimanapun karena berbasis mobile, validasi data laporan.
3. **Design (Perancangan)**, Tahap perancangan dilakukan untuk menetapkan bagaimana sistem akan dioperasikan. Hal ini berkaitan dengan menentukan perangkat keras, perangkat lunak, jaringan, tampilan program dan laporan yang akan dipakai. Selain itu perlu juga menspesifikasi program, database dan file yang dibutuhkan. Desain pada penelitian ini dilakukan dengan merancang bentuk-bentuk sistem yang dibuat seperti, tampilan sistem, cara kerja sistem dan paketan sistem.
4. **Implementation (Implementasi)**, Implementasi merupakan tahap untuk menerjemahkan data atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman komputer yang telah ditentukan. Pada tahap ini desain perangkat lunak sudah menjadi sebuah program lengkap yang terdiri dari unit program dimana konstruksi sistem dan instalasi sistem telah disesuaikan sehingga program dapat berjalan.
5. **Maintenance (Pemeliharaan)**, Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung dan pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada tapi untuk membuat perangkat lunak baru.

2.12 Penelitian Terdahulu

Tabel penelitian yang terkait dengan aplikasi pembelajaran berbasis android, diantaranya :

Tabel 2.3 Penelitian Terdahulu

No	Nama	Judul	Keterangan
1.	Yulmaini & Muhamad Arifin	Perancangan Sistem Pendeteksi Pencemaran Air Menggunakan Case Base Reasoning	IBI DARMAJAYA 2012
2.	Suhendro Yusuf Irianto & Winarsih	Pemanfaatan Aplikasi Sistem Pakar untuk Diagnosa Penyakit Paru-paru	IBI DARMAJAYA 2011
3.	Tursina	Case Base Reasoning untuk menentukan daerah berpotensi demam berdarah (Studi kasus kota Pontianak	IBI DARMAJAYA 2012