BAB 2
LANDASAN TEORI

2.1 *Artificial Intelligence* (AI)

Artificial Intellegence atau AI dalam bahasa Indonesia artinya kecerdasan buatan yaitu, kecerdasan buatan yang di tunjukan oleh suatu entitas ilmiah. Kecerdasan dibuat untuk dimasukkan ke dalam suatu mesin/ komputer supaya bisa melakukan pekerjaan seperti yang biasa dikerjakan manusia [6]. Karena itu kecerdasan buatan harus didasarkan pada prinsip-prinsip teoretikal dan terapan yang menyangkut struktur data yang digunakan dalam representasi pengetahuan (*knowledge representation*), algoritma yang diperlukan dalam penerapan pengetahuan itu, serta teknik-teknik bahasa dan pemrograman yang dipakai dalam implementasinya.

Beberapa persoalan yang ditangani oleh kecerdasan buatan antara lain adalah permainan (*game*), pembuktian teorema, pemecahan problema umum, persepsi (*vision* dan percakapan), pemahaman bahasa alamiah, pemecahan problema pakar atau *expert* (matematika simbolik, diagnosis medis, rancang bangun rekayasa, analisis kimia). Hal – hal seperti itu telah menjadi disiplin ilmu tersendiri, yang memusatkan perhatian pada penyediaan solusi masalah di kehidupan yang nyata [6].

2.2 Bahasa Alami (*Natural Language*)

Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antar manusia. Bentuk utama representasinya adalah berupa suara/ucapan (*spoken language*), tetapi sering pula dinyatakan dalam bentuk tulisan. Bahasa dapat dibedakan menjadi (1) Bahasa Alami, dan (2) Bahasa Buatan. Bahasa alami adalah bahasa yang biasa digunakan untuk berkomunikasi antar manusia, misalnya bahasa Indonesia, Sunda, Jawa, Inggris, Jepang, dan sebagainya. Bahasa buatan adalah bahasa yang dibuat secara khusus untuk

memenuhi kebutuhan tertentu, misalnya bahasa pemodelan atau bahasa pemrograman komputer.

Chomsky adalah orang yang pertama kali merepresentasikan bahasa sebagai rangkaian simbol [7]. Chomsky berhasil memperlihatkan bahwa bahasa apapun dapat direpresentasikan dengan suatu cara yang universal. Pemikiran Chomsky yang merepresentasikan bahasa sebagai kumpulan simbol-simbol dan aturan yang mengatur susunan simbol-simbol tersebut telah membuka peluang untuk melakukan pemrosesan bahasa secara simbolik dengan teknologi komputer, sehingga melahirkan bidang ilmu *Natural Language Processing* (NLP).

2.2.1 Pengertian *Natural Language Processing*

Pengolahan bahasa alami (NLP) adalah bidang ilmu komputer dan linguistik berkaitan dengan interaksi antara komputer dan manusia. Seluler generasi bahasa sistem komputer yang mengubah informasi dari *database* ke dalam bahasa manusia yang dapat dibaca [7]. *Natural language processing*, biasanya disingkat dengan NLP, mencoba membuat komputer mampu memahami suatu perintah yang dituliskan dalam bentuk bahasa sehari-hari dan diharapkan komputer juga merespon dalam bahasa yang mirip dengan bahasa natural. Setelah komputer bisa memahami perintah dalam bahasa natural, maka diharapkan sistem komputer juga dapat memberikan respon dalam bahasa natural pula.

Pada proses NLP terdapat proses dimana teks masukan akan dipotong berdasarkan setiap kata ataupun karakter penyusunnya. Proses ini dinamakan proses *tokenizing*. Hasil setiap pemotongan ini dinamakan token.

2.3 *Structure Query Language* (SQL)

Structure Query Language atau SQL merupakan suatu bahasa yang digunakan untuk mengakses *database*. SQL juga sering disebut *query*. Secara umum SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML) [8].

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus *database* dan objek – objek yang diperlukan *database*. Secara umum, DDL yang

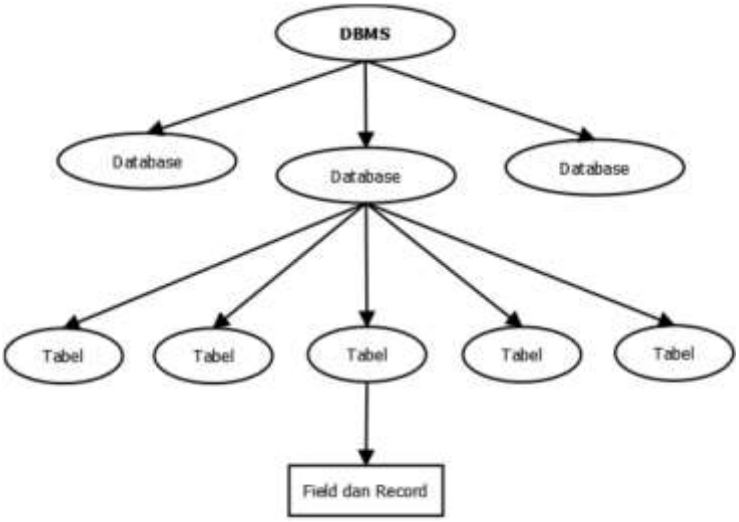
digunakan adalah *Create, Alter, Use, Drop* yang biasanya digunakan oleh administrator dalam pembuatan suatu aplikasi *database*. DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang dapat dilakukan adalah *Select, Insert, Update, Delete* [8].

Fungsi Agregat, yaitu fungsi khusus yang melibatkan sekelompok data, secara umum fungsi agregat adalah SUM, COUNT, AVG, MAX, dan MIN. Fungsi ini digunakan pada bagian *select*, syarat untuk fungsi ini adalah diletakkan pada bagian *having*, bukan *where. Sub query* adalah suatu *query* yang sudah menjadi kompleks terutama melibatkan lebih dari satu tabel atau fungsi agregat [8].

2.4 Database

Database atau basis data adalah kumpulan data yang disimpan secara sistematis di komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur, dan juga batasan – batasan dan yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghindari duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga ubah data yang rumit. Dalam aplikasi yang penulis buat, maka penulis menggunakan aplikasi dalam memanajemen *database* yaitu MySQL [8].

Dalam konsep *database*, urutan atau hierarki *database* sangatlah penting. Urutan atau hierarki *database* digambarkan dalam gambar sebagai berikut :



Gambar 2.1 Hierarki Database

2.5 Ekstraksi Informasi

Untuk mendapatkan informasi terstruktur dari teks yang tidak terstruktur, hal pertama yang perlu didefinisikan adalah informasi target sebagai informasi terstruktur yang akan di ekstrak [9]. Ekstraksi Informasi adalah pengambilan fakta dan informasi terstruktur dari isi koleksi teks yang besar. Pengertian fakta di sini adalah beragam entitas yang diperhitungkan. Secara singkat ekstraksi informasi adalah sebuah proses mendapatkan fakta-fakta terstruktur dari data yang tersedia [10]. Berdasarkan penjelasan dari kutipan di atas, penulis berpendapat bahwa ekstraksi informasi adalah proses penyaringan parameter-parameter informasi fakta dari data yang telah tersedia.

2.6 Sistem Berbasis Aturan

Sistem Berbasis Aturan (*Rule-based System*) adalah jalan untuk menyimpan atau memanipulasi pengetahuan untuk menintrepentasikan informasi yang berguna [11]. Biasanya sistem ini diimplementasikan dengan aplikasi *Artificial Intelligence*. Suatu Aturan terdiri dari 2 bagian, yaitu:

1. *Antacedent*, yaitu bagian yang mengekspresikan situasi atau premis (Pernyataan berawalan IF)

2. *Consequent*, yaitu bagian yang menyatakan suatu tindakan tertentu atau konklusi yang diterapkan jika situasi atau premis bernilai benar (Pernyataan berawalan THEN).

Umumnya, sebuah aturan dapat mempunyai gabungan beberapa *antecedent* dengan kata kunci AND (konjungsi), OR (disjungsi), atau kombinasi keduanya. Metode Aturan dapat dilihat pada ilustrasi berikut.

```
IF <antecedent 1>
AND <antecedent 2>
        ...
AND <antecedent n>
THEN <consequent>
IF <antecedent 1>
OR <antecedent 2>
        ...
OR <antecedent n>
THEN <consequent>
```

Gambar 2.2 Contoh Algoritma *Rule-Based Antacedent* dan *Consequent*

2.7 *Flowchart* Diagram

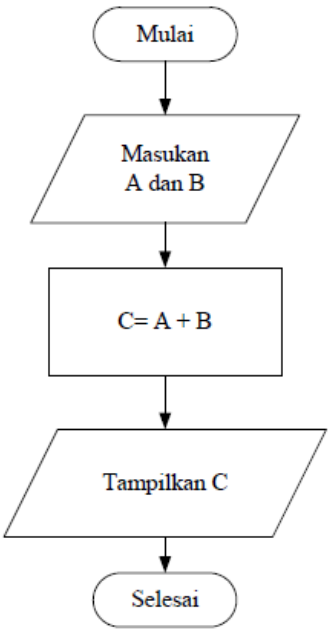
Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah [12]. *Flowchart* program merupakan langkah-langkah (instruksi-instruksi) program yang menceritakan kejadian suatu proses satu dengan proses lainnya dalam suatu program secara mendetail yang di wakilkan dalam bentuk simbol atau bagan.

Flowchart memiliki kriteria sebagai berikut

1. Tidak ada kaidah yang baku.
2. *Flowchart* = gambaran hasil analisa suatu masalah
3. *Flowchart* dapat bervariasi antara satu pemrogram dengan pemrogram
4. lainnya.
5. Secara garis besar ada 3 bagian utama yaitu *input*, proses dan *output*
6. Hindari pengulangan proses yang tidak perlu dan logika yang berbelit
7. sehingga jalannya proses menjadi singkat.

8. Jalannya proses digambarkan dari atas ke bawah dan diberikan tanda
9. panah untuk memperjelas.

Contoh *Flowchart* Program:



Gambar 2.3 Contoh *Flowchart* Program

2.8 Data Flow Diagram

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data,baik secara manual maupun komputerisasi [13]. DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa

maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Penelitian ini menggunakan acuan simbol DFD yang dikembangkan oleh Tom Demarco dan Edward Yourdon [13].

2.9 *Tools* Pembangunan Sistem

2.9.1 MYSQL

MySQL adalah sebuah sistem manajemen *database* relasi (*relation database management system*) yang bersifat *open source*. MySQL merupakan buah pikiran dari Michael "Monty" Widenius, Davied axmark dan Allan Larson yang dimulai tahun 1995. Mereka bertiga kemudian mendirikan perusahaan bernama MySQL AB di swedia. MySQL merupakan aplikasi *database* server. Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah *database* beserta isinya. Kita dapat memanfaatkan MySQL untuk menambah, mengubah, dan menghapus data yang berada dalam *database* [8]. MySQL versi 1.0 rilis pada Mei 1996 dan penggunaannya hanya terbatas di kalangan perusahaan saja. Barulah pada bulan Oktober 1996, MySQL versi 3.11.0 dirilis ke masyarakat luas. MySQL menggunakan bahasa standar SQL (*Structure Query Language*) sebagai bahasa interaktif dalam mengelola data. MySQL memiliki kinerja, kecepatan, proses dan ketangguhan yang tidak kalah dibanding *database* manajemen *system* lainnya seperti *Oracle, Sysbase, Unify*, dan sebagainya. MySQL dapat berjalan di atas banyak sistem operasi Linux, Windows, Solaris, FreeBSD, Mac OS X dan lain sebagainya [8].

2.9.2 PHP

PHP adalah singkatan dari "PHP: Hypertext Prepocessor", yaitu bahasa script open source yang sederhana namun kuat untuk menciptakan konten web dinamis. Jutaan situs web yang didukung oleh PHP adalah bukti popularitas dan kemudahan penggunaannya. PHP digunakan oleh kedua pemrogram, yang menghargai fleksibilitas dan kecepatan, dan perancang web, yang menghargai

aksesibilitas dan kenyamanannya [14]. PHP diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya PHP adalah singkatan dari "Personal Home Page Tools". Selanjutnya diganti menjadi FI ("Forms Interpreter"). Sejak versi 3.0, nama bahasa ini diubah menjadi "PHP: Hypertext Prepocessor" dengan singkatannya "PHP". PHP versi terbaru adalah versi ke-5. Berdasarkan survey Netcraft pada bulan Desember 1999, lebih dari sejuta website menggunakan PHP, di antaranya adalah NASA, Mitsubishi, dan RedHat.

2.9.3 HTML

Hyper Text Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (Standard Generalized Markup Language), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa) [15].

2.9.4 CSS

CSS adalah kependekan dari Cascading Style Sheet. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih elegan dan menarik.CSS adalah sebuah teknologi internet yang direkomendasikan oleh World Wide Web Consortium atau W3C pada tahun 1996. Awalnya, CSS dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. CSS telah mendukung banyak bahasa

markup seperti HTML, XHTML, XML, SVG (Scalable Vector Graphics) dan Mozilla XUL (XML User Interface Language) [15].

2.10 Pengujian Sistem

Pengujian menyajikan anomali yang menarik bagi perekayasa perangkat lunak. Pada proses perangkat lunak, perekayasa pertama–tama berusaha membangun perangkat lunak dari konsep abstrak ke implementasi yang dapat dilihat, baru kemudian dilakukan pengujian. Pengujian diperlukan tidak hanya untuk meminimalisasi kesalahan secara teknis tapi juga kesalahan non teknis [13].

2.10.1 Sasaran-sasaran Pengujian

Terdapat sejumlah aturan yang berfungsi sebagai sasaran pengujian :

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. Test *case* yang baik adalah test case yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

2.10.2 Pola Pengujian

Secara umum pola pengujian pada perangkat lunak adalah sebagai berikut [13].

1. Pengujian dimulai dari level komponen hingga integrasi antar komponen menjadi sebuah sistem
2. Teknik pengujian berbeda –beda sesuai dengan berbagai sisi sesuai kebutuhan
3. Pengujian dilakukan oleh pengembang perangkat lunak atau tim uji perangkat lunak
4. Pengujian dan *debugging* merupakan aktivitas yang berbeda tetapi saling melengkapi satu sama lain.

2.10.3 Pengujian *Black-box*

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk *suatu* program [13].

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi – fungsi yang tidak benar atau hilang,
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminal

2.10.4 Pengujian Unit

Strategi pengujian perangkat lunak dimulai dengan unit testing, *integration* testing, *validation* testing, dan sistem testing. Salah satu strategi pengujian perangkat lunak adalah pengujian unit, yaitu berfokus pada usaha verifikasi pada inti terkecil dari desain perangkat lunak yang disebut modul [13]. Modul diuji untuk memastikan bahwa informasi secara tepat mengalir masuk dan keluar dari inti program yang diuji. Pengujian modul didesain untuk mengungkap kesalahan sehubungan dengan komputasi yang salah. Kesalahan umum dalam komputasi adalah:

1. Inisialisasi yang tidak benar
2. Akurasi ketelitian
3. Mengakses database yang tidak dipilih

Dalam pengujian unit juga harus mengungkap kesalahan-kesalahan yang terjadi seperti:

1. Tidak ditemukan kata perintah
2. Kondisi yang tidak memiliki isi
3. Tidak ditemukan nama tabel
4. Tidak dapat terhubung ke DBMS

5. Tidak dapat terhubung ke database pendukung

Pengujian unit ini berkonsentrasi pada verifikasi fungsional dari sebuah modul dan gabungan modul-modul ke dalam struktur program. Setelah melakukan pengujian unit diharapkan secara modul tidak menemukan lagi kesalahan.

2.10.5 Pengujian *F-Measure*

F-Measure adalah representasi ukuran keakuratan dari sebuah tes dengan menggunakan nilai *precision* dan *recall* dari tes tersebut [16]. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Sedangkan *recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. *Precision* dan *recall* memanfaatkan tabel *confusion* tes untuk mengetahui nilainya. Hubungan tabel *confusion precision*, recall dan F-Measure dapat dilihat pada gambar berikut Gambar 2.4.

		Nilai sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP (True Positive) *Corect result*	FP (False Positive) *Unexpected result*
	FALSE	FN (False Negative) *Missing result*	TN (True Negative) *Corect absence of result*

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Gambar 2.4 Tabel *Confusion, Precission, Recall, Accuracy*