

BAB 2

LANDASAN TEORI

2.1 Algoritma

Algoritma adalah runtunan langkah-langkah untuk menyelesaikan suatu masalah. Untuk masalah dengan instansiasi yang kecil, kita dapat menyelesaikan masalah itu dengan mudah dan cepat. Akan tetapi bagaimana jika instansiasi masalah berukuran besar, misalnya pada masalah pengurutan, jika $n = 10000$ tentu saja tidak akan mudah untuk mengurutkan data sebanyak itu. Oleh karena itu kita perlu menuliskan langkah-langkah pengurutan sehingga dapat dijalankan oleh sebuah pemroses (komputer, robot dan sebagainya) untuk menghasilkan solusi instansiasi masalah pengurutan [11].

2.2 Error

Error adalah kejadian pada program yang tidak sesuai dengan yang diharapkan karena kesalahan dari pengguna program. Error sifatnya terprediksi dan dapat ditangani kejadiannya. Pengertian error secara umum adalah ‘kesalahan’, akan tetapi pengertian Error juga dapat di kelompokkan menurut penggunaan kata-katanya, contohnya Error dalam pengertian ilmu komputer yaitu tidak sanggup menampilkan data (tidak dapat melanjutkan ke system berikutnya) [12].

2.2.1 Perceptual Error

Perceptual Error yaitu adanya kesalahan yang disebabkan oleh ketidak jelasan keterangan dari petunjuk penggunaan yang menyebabkan kesalah tanggapan dari pihak pengguna atau *user*. Contohnya ada *icon* yang memiliki arti ambigu dalam sebuah desain antar muka [12].

2.2.2 Cognitive Error

Cognitive Error yaitu kesalahan yang diakibatkan oleh kemampuan memecahkan oleh *user* atau karena terlalu banyak konteks dan informasi status. Contohnya seperti argumen yang tidak konsisten pada pada sintak *command language* [12].

2.3 Error Handling

Error Handling adalah satu penanganan kesalahan (*error*) pada berbagai macam keadaan dalam pemrograman. Setiap ada kesalahan, maka eksekusi program tidak akan

dihentikan secara tiba tiba, tetapi akan diteruskan ke baris program yang terdapat *script* penanganan kesalahan [13].

2.3.1 Error Prevention

Error Prevention yaitu untuk mencegah kesalahan *user* sehingga skala kesalahan *user* dapat diminimalisir agar tidak terjadi eror yang lebih fatal disuatu sistem. Contohnya seperti munculnya *error message* untuk menghindari adanya kesalahan yang lebih fatal [13].

2.3.2 Error Recovery

Error Recovery yaitu untuk memperbaiki kesalahan terhadapap suatu kesalahan yang sudah terlanjur dilakukan oleh *user*. Contohnya *user* sering melakukan *undo* atau *cancel* dalam suatu sistem [13].

2.4 Natural Language Processing

Natural Language Processing atau yang dapat diartikan sebagai bahasa alami merupakan bahasa yang secara umum digunakan oleh manusia untuk berkomunikasi dalam kehidupan sehari-hari. *Natural Language Processing* sendiri merupakan salah satu bidang dari ilmu kecerdasan buatan (*Artificial Intelligence*). *Natural Language Processing* mengkaji tentang hubungan interaksi antara manusia dan komputer dengan menggunakan bahasa alami manusia. Ada beberapa alasan mengapa sebuah komputer agen harus dapat memroses bahasa alami, yang pertama adalah untuk berkomunikasi dengan manusia. Kedua, untuk mendapatkan informasi dari bahasa yang ditulis [14].

2.5 Grammar

Tata bahasa (grammar) dapat diidentifikasi secara formal sebagai sekumpulan dari himpunan-himpunan variabel, simbol-simbol terminal, simbol awal, yang dibatasi oleh aturan-aturan produksi [15]. Pada tahun 1959 Noam Chomsky melakukan penggolongan tingkatan bahasa menjadi empat, seperti yang ditunjukkan pada tabel berikut [15].

Tabel 2.1 Penggolongan Tingkat Bahasa

Bahasa	Mesin Otomata	Batasan Aturan Produksi
Reguler / Tipe 3	<i>Finite State Automata</i> (FSA) meliputi <i>Deterministic Finite Automata</i> (DFA) & <i>Non-</i>	α adalah simbol variabel β maksimal memiliki sebuah simbol variabel

	<i>deterministic Finite Automata</i> (NFA)	yang bila ada terletak diposisi paling kanan
Bebas Konteks / <i>Context Free</i> / Tipe 2	<i>Push Down Automata</i> (PDA)	α merupakan simbol variabel
<i>Context Sensitive</i> / Tipe 1	<i>Linier Bounded Automata</i>	$ \alpha \leq \beta $
<i>Unrestricted / Phase Structure / Natural Language</i> / Tipe 0	Mesin Turing	Tidak ada batasan

Aturan produksi merupakan pusat dari tata bahasa yang mengatur bagaimana tata bahasa melakukan perubahan suatu *string* ke bentuk lainnya, dan melalui aturan produksi tersebut didefinisikan suatu bahasa yang berhubungan dengan tata bahasa tersebut [15]. Aturan produksi dapat dinyatakan dengan bentuk ‘ $a \rightarrow b$ ’ (a menghasilkan b atau a menurunkan b), dimana a menyatakan simbol-simbol ruas kiri aturan produksi (sebelah kiri tanda ‘ \rightarrow ’), dan b menyatakan simbol-simbol pada ruas kanan dan bisa disebut juga hasil produksi [15]. Simbol-simbol tersebut dapat berupa simbol terminal atau simbol variabel/non terminal. Simbol terminal adalah simbol yang tidak dapat diturunkan lagi, sedangkan simbol variabel/non terminal adalah simbol yang masih bisa diturunkan lagi.

Dengan menerapkan aturan produksi, suatu tata bahasa dapat menghasilkan sejumlah *string*. Himpunan *string* tersebut adalah bahasa yang didefinisikan oleh tata bahasa tersebut [15].

2.6 Source Code

Source code pada ilmu komputer merupakan kumpulan-kumpulan perintah yang digunakan untuk menyelesaikan masalah yang ditulis dalam bahasa yang dapat dipahami oleh komputer [16].

2.7 Bahasa Pemrograman *Pascal*

Pascal adalah salah satu bahasa pemrograman komputer yang sering digunakan untuk pembelajaran algoritma dan pemrograman bagi pemula terutama di bidang akademis. Nama Pascal diambil dari ahli matematika Prancis abad pertengahan, Blaise Pascal. Bahasa Pemrograman Pascal dikembangkan oleh Niklaus Wirth pada tahun 1970 dan mencapai puncak kepopulerannya di pada era 1970 hingga awal 1990an (terutama dengan compiler Turbo Pascal) [17]. Bahasa pemrograman *Pascal* menerapkan konsep procedural dan structured programming yang baik, sehingga sesuai dalam konsep pembelajaran

pemrograman. Perintah *Pascal* memiliki beberapa perintah seperti *begin*, *end*, *write*, dan *read* [17].

Dalam Pascal, pengenalan ialah nama yang dapat diberikan pada suatu elemen program, dapat berupa konstanta, variabel, fungsi, suatu prosedur, maupun suatu program [18]. Pengenal dapat disusun dari karakter huruf maupun karakter bilangan, dengan beberapa aturan yang harus dipenuhi [18], yaitu:

1. Nama pengenalan harus diawali dengan karakter huruf.
2. Karakter kedua dan selanjutnya dapat berupa kombinasi angka dan huruf, tetapi tidak boleh menggunakan karakter khusus seperti *?*, *#*, dan sebagainya. Namun ada beberapa versi Pascal yang menerima garis bawah *'_'* sebagai karakter penyusun nama pengenalan.
3. Panjang karakter yang digunakan sebagai pengenalan bisa sembarang, tapi dalam beberapa versi Pascal hanya mengenal delapan karakter awal, karakter sembilan dan seterusnya diabaikan.
4. Beberapa karakter sudah digunakan oleh Pascal untuk tujuan tertentu, sehingga tidak dapat dipakai sebagai nama pengenalan. Nama-nama ini disebut kata khusus (reserved word).
5. Beberapa nama yang disebut pengenalan standar juga telah mempunyai arti khusus, tetapi jika didefinisikan ulang maka dapat menjadi nama pengenalan. Jika pengenalan standar digunakan sebagai pengenalan biasa maka arti khususnya tidak akan dipergunakan.

2.8 Case Folding

Case folding merupakan proses menyeragamkan huruf pada teks masukan menjadi huruf kecil (lowercase) atau huruf kapital (uppercase). Dalam penelitian ini penyeragaman huruf digunakan untuk mengubah semua huruf pada teks masukan menjadi huruf kecil.

2.9 Filtering

Filtering biasanya dilakukan pada dokumen untuk menghapus beberapa stop word [19]. Pada penelitian ini karakter yang diperbolehkan dalam penelitian ini yaitu *'a'* sampai *'z'*, *'0'* sampai *'9'*, koma (*' , '*), titik (*' . '*), garis bawah (*' _ '*), dan spasi.

2.10 Scanning

Scanning merupakan tahap memilah teks masukan menjadi token-token berdasarkan kelasnya. Pada penelitian ini, tahap scanning menerima masukan stream

kata yang kemudian memilah teks masukan menjadi satuan leksik atau token [20]. Scanner juga membaca karakter input satu per satu untuk menentukan unit kata primitif, atau "token", dari setiap perintah. Besaran pembangun token atau leksik meliputi hal-hal sebagai berikut.

1. Identifier

Identifier dapat berupa keyword. Keyword merupakan kata kunci yang sudah didefinisikan oleh suatu bahasa [20].

2. Nilai Konstanta

Nilai konstanta merupakan konstanta yang ada pada suatu teks. Dapat berupa nomor, string, dan sebagainya [20].

3. Operator dan Delimeter

Operator dapat berupa operator aritmetika atau operator logika. Delimiter adalah pemisah atau pembatas, misalnya titik, koma, spasi, dan sebagainya [20].

2.11 Parsing

Parsing adalah proses menentukan bagaimana string terminal dapat dihasilkan oleh grammer [21]. Metode parsing digolongkan menjadi 2 yaitu :

1. Top Down

Metode ini melakukan penurunan dari root/puncak menuju leaf/daun. Metode top down meliputi Backtrack/backup dan No backtrack.

2. Bottom Up

Metode ini melakukan penurunan dari leaf/daun menuju root.

2.12 Pohon Sintaks

Pohon sintaks atau pohon penurunan berguna untuk menggambarkan bagaimana memperoleh suatu string dengan cara menurunkan simbol-simbol variabel menjadi simbol-simbol terminal [22]. Setiap simbol variabel akan diturunkan sampai tidak ada lagi simbol variabel atau hanya menyisakan simbol terminal. Proses penurunan atau parsing bisa dilakukan dengan cara sebagai berikut [22].

1. Penurunan paling kiri (leftmost derivation): simbol variabel paling kiri yang akan diperluas terlebih dahulu. Misal, terdapat tata bahasa bebas konteks:

$$S \rightarrow aAS \mid a$$

$$A \rightarrow SbA \mid ba$$

Untuk memperoleh untaian 'aabbaa' dari tata bahasa bebas konteks diatas maka penurunannya sebagai berikut.

$S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbaa.$

2. Penurunan paling kanan (rightmost derivation): simbol variabel atau non terminal paling kanan yang akan diperluas terlebih dahulu.

Misal, terdapat tata bahasa bebas konteks:

$S \rightarrow aAS \mid aA \rightarrow SbA \mid ba$

Untuk memperoleh untaian 'aabbaa' dari tata bahasa bebas konteks diatas maka penurunannya sebagai berikut.

$S \Rightarrow aAS \Rightarrow aAa \Rightarrow aSbAa \Rightarrow aSbbaa \Rightarrow aabbaa.$

2.13 Perancangan Sistem

Perancangan sistem bertujuan untuk menggambarkan, merencanakan, dan pembuatan sketsa dari beberapa elemen terpisah menjadi satu kesatuan sistem yang akan dibangun.

2.13.1 Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem [23]. Diagram konteks merupakan level tertinggi dari Data Flow Diagram (DFD), yang dipakai untuk menggambarkan aliran data input dan output sistem. Dalam diagram konteks hanya memiliki satu proses, yaitu sistem secara keseluruhan. Pada diagram konteks tidak memiliki data store.

2.13.2 Data Flow Diagram (DFD)

Diagram aliran data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. DFD sering digunakan untuk penggambaran pada suatu sistem yang telah ada atau sistem yang baru yang akan dikembangkan secara logika dan menjelaskan aliran data dari mulai pemasukan data sampai dengan keluaran data tingkatan diagram arus data mulai dari diagram konteks yang menjelaskan secara umum suatu sistem atau batasan sistem dari level 0 diturunkan menjadi level 1 sampai sistem dapat tergambarkan secara terperinci [23].

1. Entitas Luar (Boundary)

Entitas luar adalah kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan masukan (input) atau menerima keluaran (output) dari sistem [23].

2. Arus Data (Data Flow)

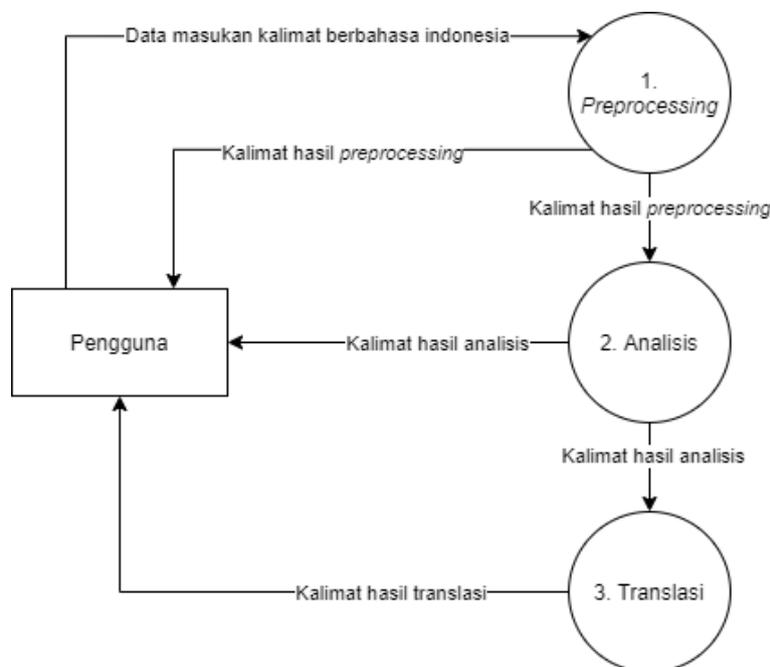
Arus data diberi simbol suatu panah. Arus data mengalir diantara proses (process), simpanan data (data store) dan kesatuan luar (external entity) [23].

3. Proses (Process)

Proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk menghasilkan arus data yang akan keluar dari proses [23].

4. Penyimpanan Data (Data Store)

Penyimpanan data adalah simpanan dari data yang dapat berupa file atau database di sistem komputer, arsip atau catatan manual, agenda atau buku. Simpanan data DFD dapat di simbolkan dengan sepasang garis horizontal [23].



Gambar 2.1 Contoh Data Flow Diagram

2.13.3 Spesifikasi Proses

Spesifikasi proses merupakan keterangan dari semua proses yang ada pada DFD. Dalam spesifikasi proses yang perlu dicantumkan yaitu nomor urut proses, nama proses, sumber, input, output, tujuan, dan logika proses. Penulisan logika proses dapat menggunakan bahasa deskriptif maupun pseudocode [23].

Tabel 2.2 Tabel Spesifikasi Proses

No.	Proses	Keterangan
-----	--------	------------

1	No. Proses	1
	Nama Proses	<i>Preprocessing</i>
	Sumber	Pengguna
	<i>Input</i>	Data masukan kalimat berbahasa indonesia
	<i>Output</i>	Kalimat hasil <i>Preprocessing</i>
	Logika Proses	<ol style="list-style-type: none"> 1. Pengguna memasukkan kalimat berbahasa indonesia ke sistem 2. Sistem melakukan proses <i>Case Folding</i> dan <i>Filtering</i>. 3. Sistem menampilkan kalimat hasil <i>Preprocessing</i>

2.13.4 Kamus Data

Kamus data berfungsi untuk memudahkan mengartikan aplikasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem secara presisi, sehingga pengguna dan penganalis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses [23].

2.14 PHP

PHP secara umum dikenal sebagai bahasa pemrograman script yang membuat dokumen HTML secara on the fly yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman server side. PHP dikembangkan menjadi bahasa pemrograman script yang dijalankan di atas platform sistem operasi secara langsung. PHP juga dapat digunakan dan dijalankan dari desktop, yang disebut PHP CLI (Command Line Interface). Dalam penelitian ini, PHP digunakan untuk pemrosesan bahasa alami ke source code, dari mulai preprocessing sampai proses translasi [24].