BAB 2
TINJAUAN PUSTAKA

2.1 Algoritma

Algoritma adalah kata serapan dari kata "*Algorithm*", algoritma adalah suatu langkah-langkah atau urutan dalam menyelesaikan suatu masalah yang disusun secara sistematis dan logis[3]. Dalam penulisan algoritma dapat menggunakan standar penulisan berupa Notasi algoritmik, standar ini digunakan sebagai alat untuk merancang algoritma yang dapat dibaca dan mudah dimengerti. Ada beberapa cara dalam menulis notasi algoritmik, yaitu:

1. Deskriptif

   Langka dalam penulisan notasi algoritmik dapat ditulis dengan kalimat bahasa sederhana yang digunakan sehari-hari. Notasi ini akan sulit untuk diterjemahkan ke *source code,* namun baik untuk orang awam.

2. Dengan bagan-alir (*flowchart*)

   Langkah dalam penulisan notasi algoritmik dapat ditulis dengan menggunakan *flowchart* dimana langkah atau urutan suatu masalah digambarkan menggunakan simbol-simbol khusus tertentu. *Flowchart* cocok untuk masalah yang kecil karena masalah dapat digambarkan secara visual, namun akan menimbulkan masalah jika langkah algoritma dianggap besar sehingga berakibat membutuhkan lembar halaman yang besar untuk menggambar *flowchart*.

3. Dengan *Pseudo-code*

   Langkah dalam penulisan notasi algoritmik selanjutnya dapat ditulis dengan menggunakan *Pseudo-code*. Notasi ini mirip dengan penulisan bahasa pemrograman tingkat tinggi, hanya saja tidak menggunakan aturan khusus seperti indeks, titik koma, dan sebagainya, asalkan tidak membingungkan pembaca.

2.2 Matriks

Matriks merupakan suatu bilangan (real atau kompleks) yang tersusun dalam baris dan kolom yang membentuk persegi panjang[14]. Matriks dinyatakan dengan suatu kurung siku ([…]) dan memiliki *m* baris dan *n* kolom sehingga matriks tersebut disebut matriks orde *m* x *n*. Pada gambar 2.1 contoh matriks (*a)* dengan orde 3x3 dan (*b)* dengan orde 2x3.

$$\begin{bmatrix} 4 & 2 & 0 \\ 6 & 6 & 6 \\ 9 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 5 & -1 & 2 \\ 3 & 0 & 9 \end{bmatrix}$$

(a) (b)

Gambar 2.1 Orde pada matriks

Dalam matriks untuk menyatakan suatu elemen dapat dinyatakan dengan koordinat *x* dan *y*, pernyataan ini disebut dengan notasi akhir ganda[14]. Pada gambar 2.2 , dapat dinyatakan bahwa a_{21} adalah elemen dalam baris kedua dan kolom kesatu.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Gambar 2.2 Notasi akhir ganda matriks

2.2.1 Operasi Matriks

2.2.1.1 Penambahan Matriks

Agar dua buah matriks dapat dilakukan operasi penambahan, dua matriks tersebut harus memiliki orde yang sama. Dalam melakukan penambahan dapat dilakukan seperti halnya penjumlahan operasi aritmatika dasar. Berikut contoh operasi penambahan matriks dapat dilihat pada gambar 2.3.

$$\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 7 & 10 \\ 8 & 12 \end{bmatrix}$$

Gambar 2.3 Operasi penambahan matriks

Dapat dilihat dari contoh Gambar 2.3, dilakukan penambahan dua matriks dimana penambahan seperti aritmatika dasar, dapat dilihat dari dua matriks elemen baris satu dan kolom satu dengan nilai 2 dan 5 ditambah menghasilkan nilai 7, begitu juga dengan elemen-elemen lainnya.

2.2.1.2 Pengurangan Matriks

Dalam operasi pengurangan matriks sama seperti penambahan matriks, dua buah matriks harus memiliki orde yang sama untuk dilakukan operasi perhitungan. Berikut contoh operasi pengurangan matriks dapat dilihat pada Gambar 2.4.

$$\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix} - \begin{bmatrix} 5 & 6 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} -3 & -2 \\ 4 & 4 \end{bmatrix}$$

Gambar 2.4 Operasi pengurangan matriks

2.2.1.3 Perkalian Matriks

Dalam operasi perkalian matriks dibagi menjadi dua, yaitu operasi perkalian skalar matriks dan perkalian matriks[14]. Perkalian skalar matriks dilakukan dengan cara suatu matriks dikalikan dengan bilangan tunggal, perkalian ini dapat ditulis seperti $a1(b_{21}) = (ab_{21})$. Berikut contoh perkalian skalar matriks dapat dilihat pada Gambar 2.5.

$$3 \times \begin{bmatrix} 1 & 2 \\ 4 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 12 & 3 \end{bmatrix}$$

Gambar 2.5 Perkalian skalar matriks

Dalam operasi perkalian matriks berbeda dengan perkalian skalar matriks, perkalian matriks memiliki aturan dimana dua matriks dapat dikalikan jika jumlah kolom matriks pertama sama dengan jumlah baris pada matriks kedua, hasil matriks perkalian akan terbentuk matriks dengan orde baru dimana nilai baris pada hasil perkalian berasal dari nilai baris matriks pertama dan nilai kolom pada hasil perkalian berasal dari nilai kolom matriks kedua. Contoh perkalian antara dua matriks dapat dilihat pada Gambar 2.6.

$$\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} (2 * 1) + (4 * 2) \\ (6 * 1) + (8 * 2) \end{bmatrix}$$

$$= \begin{bmatrix} 10 \\ 22 \end{bmatrix}$$

Gambar 2.6 Perkalian matriks

Dapat dilihat dari Gambar 2.6, matriks pertama memiliki orde 2x2 dan matriks kedua memiliki orde 2x1. Dilihat juga dalam perkalian matriks, setiap elemen baris matriks pertama dikalikan dengan elemen yang berkorespons dalam kolom pertama matriks kedua, kemudian hasil kalinya ditambahkan. Hasil dari perkalian terbentuk matriks baru dengan orde 2x1 dimana orde baris berasal dari orde baris matriks pertama dan orde kolom berasal dari orde baris matriks kedua.

2.3 Bahasa Pemrograman Pascal

Bahasa pemrograman *Pascal* merupakan salah satu bahasa pemrograman tingkat tinggi (*high level language*), bahasa *Pascal* dikembangan dari bahasa ALGOL dimana pada masanya bahasa ALGOL sulit untuk dipelajari sehingga lahir bahasa *Pascal* yang dikembangkan oleh professor Niklaus Wirth dari Switzerland pada tahun 1970[1]. Bahasa *Pascal* diciptakan dengan notasi yang mirip dengan standar bahasa Inggris sehingga pemrograman *Pascal* mudah untuk dibaca dan dipelajari hingga saat ini.

Pada bahasa *Pascal*, konstanta, variabel, fungsi, prosedur, merupakan elemen program yang harus memiliki suatu pengenalan (*identifier*)[15]. Pengenalan pada bahasa *Pascal* memiliki beberapa aturan yang harus terpenuhi, yaitu:

1. Nama pengenal 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 pengenal.
3. Panjang karakter yang digunakan sebagai pengenal 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 pengenal. Nama-nama ini disebut kata khusus (*reserved word*).

5. Beberapa nama yang disebut pengenal standar juga telah mempunyai arti khusus, tetapi jika didefinisikan ulang maka dapat menjadi nama pengenal. Jika pengenal standar digunakan sebagai pengenal biasa maka arti khususnya tidak akan dipergunakan.

2.4 Natural Language Processing

Natural Language Processing atau dalam bahasa indonesia disebut pemrosesan bahasa alami adalah suatu proses dimana komputer memproses atau untuk memahami bahasa alami dengan tujuan tertentu[16]. Bahasa alami sendiri adalah bahasa umum yang digunakan oleh manusia sebagai sarana berkomunikasi. Dalam pemrosesan bahasa alami berhubungan dengan lebih dikembangkan fasilitas interaksi antara komputer dan manusia, sehingga komputer harus memahami terlebih dahulu maksud dari masukan dari manusia.

2.5 Grammar

Grammar (tata bahasa) merupakan kumpulan dari suatu himpunan variabel, simbol terminal, simbol awal, yang memiliki batasan dan diatur oleh aturan produksi [17]. Dalam penggolongannya tingkatan bahasa, Noam Chomsky membagi menjadi empat hirarki berdasarkan aturan produksinya. Berikut hirarki Chomsky pada Tabel 2.1.

Tabel 2.1 Penggolongan Tingkat Bahasa

Bahasa	Mesin Automata	Aturan Produksi
Regular / Tipe 3	Finite State Automata (FSA) meliputi Deterministic Finite Automata (DFA), Non-deterministic Finite Automata (NFA)	α adalah sebuah simbol variabel. β maksimal memiliki sebuah simbol variabel yang bila ada terletak diposisi paling kanan.
Context Free / Bebas Konteks / Tipe 2	*Push Down Automata* (PDA)	α berupa sebuah simbol variable
Context Sensitive/ Tipe 1	Linear Bounded Automata	$\lvert \alpha \rvert \leq \lvert \beta \rvert$
Natural Language / Tipe 0	Mesin Turing	Tidak ada batasan

Aturan produksi sendiri digunakan pada suatu tata bahasa sebagai aturan dalam melakukan perubahan suatu string ke bentuk lainnya, sehingga bahasa dapat didefinisikan dengan tata bahasa tersebut[17]. Aturan atau himpunan produksi dapat dinyatakan dalam bentuk 'X → xa' (X menghasilkan xa atau X menurunkan xa) dimana X merupakan suatu aturan produksi atau nonterminal dan xa sebagai hasil produksi atau terminal.

Pada penelitian ini, penerjemahan dilakukan dari bahasa Indonesia ke bahasa *Pascal* secara bahasa bebas konteks (*Context Free Grammar*). Bahasa bebas konteks merupakan tipe 2 dimana merupakan dasar pembentukan parser, adapun dalam penulisan *syntax* secara formal dengan notasi BNF(*Backus Naur Form* atau *Backus Normal Form*)[17]. Simbol dalam notasi BNF dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol notasi BNF

Simbol	Keterangan
::=	Sama identik dengan simbol →
|	Sama dengan atau
<>	Pengapit simbol terminal
{}	Mengulang 0 sampai n kali

2.6 Case Folding

Case Folding merupakan proses menyeragamkan suatu kalimat atau teks menjadi case yang sama sesuai dengan kebutuhan, baik menyeragamkan kalimat ke huruf kecil (*lowercase*) atau huruf kapital (*uppercase*)[18]. Pada penelitian ini, semua kalimat masukan diubah atau diseragamkan menjadi *lowercase*. Contoh penerapan case folding dapat dilihat di Tabel 2.3.

Tabel 2.3 Contoh case folding

Data Masukan	Hasil case folding
Buat aplikasi coba1!!. Buat variabel x dengan tipe data bilangan bulat. Baca nilai x, jika x lebih dari 5 maka Tampilkan nilai x.	buat aplikasi coba1!!. buat variabel x dengan tipe data bilangan bulat. baca nilai x, jika x lebih dari 5 maka tampilkan nilai x.

2.7 Filtering

Filtering adalah proses menghapus karakter yang tidak diperlukan dan dihilangkan pada suatu kalimat masukan[19].Pada penelitian ini, filtering digunakan untuk menerima karakter masukan ‘a’ sampai ‘z’, ‘0’ sampai ‘9’, garis bawah (‘_’), titik (‘.’), koma (‘,’), tambah (‘+’), kurang (‘-’), kali (‘*’), bagi (‘/’), dan spasi. Contoh penerapan filtering dapat dilihat di Tabel 2.4.

Tabel 2.4 Contoh filtering

Data Masukan	Hasil filtering
buat aplikasi coba1!!. buat variabel x dengan tipe data bilangan bulat. baca nilai x, jika x lebih dari 5 maka tampilkan nilai x.	buat aplikasi coba1. buat variabel x dengan tipe data bilangan bulat. baca nilai x, jika x lebih dari 5 maka tampilkan nilai x.

2.8 Scanning

Scanning adalah proses dimana teks masukan diubah menjadi token dan dipilih berdasarkan kelasnya[17]. Pada tahap penelitian ini, teks masukan akan dipecah menjadi satuan leksikal atau token yang kemudian akan diproses ke tahap selanjutnya. Berdasarkan leksikal atau token terdapat hal-hal yang meliputi sebagai berikut.

1. Identifier

   Identifier dapat berupa keyword atau nama. Pada penelitian ini keyword digunakan sebagai kata kunci yang telah didefinisikan oleh suatu

bahasa, contohnya ‘tampilkan’, ‘masukan’, ‘buat’, dan sebagainya. Sedangkan nama dideklarasikan oleh pemakai, seperti nama variabel.

2. Nilai Konstanta

Nilai konstanta merupakan konstanta yang ada pada suatu teks, contohnya dapat berupa string, nomor, dan sebagainya.

3. Operator dan Delimiter

Operator dapat berupa operator aritmatika dan logika, sedangkan delimiter berupa pemisah dan pembatas, contoh titik, koma, dan sebagainya.

2.9 Parsing

Parsing adalah proses dimana token-token akan diperiksa sesuai dengan aturan sintak yang telah dibentuk atau ditetapkan. Ketika urutan token tidak sesuai dengan aturan sintak mengakibatkan token menjadi tidak valid atau ditolak. Pada proses parsing dapat digolongkan sebagai berikut[20].

1. *Top Down*

   Metode *Top Down* dengan cara penurunan dari *root* (puncak) menuju *leaf* (daun). Metode ini meliputi:

   a. *Backtrack/backup : Brute Force;*
   b. *No. backtrack : Recursive Descent Parser.*

2. *Bottom Up*

   Metode *Bottom Up* dengan cara penurunan dari *leaf* (daun) menuju *root*.

2.10 Pohon Sintaks

Pohon sintaks atau disebut juga pohon penurunan digunakan sebagai gambaran untuk memperoleh suatu string dengan cara menurunkan simbol variabel menjadi simbol terminal[20]. Penurunan pada simbol variabel dilakukan

hingga tidak ada lagi simbol variabel tersisa atau hanya menyisakan simbol terminal. Terdapat dua cara dalam penurunan, diantaranya:

1. Penurunan paling kiri (*leftmost deviation*)

   Penurunan ini dilakukan dengan cara simbol variabel yang terletak paling kiri akan diperluas terlebih dahulu.

   Contoh, terdapat tata bahasa :

   $$S \rightarrow aAS \mid a$$
   $$A \rightarrow SbA \mid ba$$

   untuk memperoleh turunan akhir ‘aabbaa’ dari tata bahasa diatas, maka dilakukan penurunan:

   S => aAS => aSbAS => aabAS => aabbaS => aabbaa

2. Penurunan paling kanan (*rightmost deviation*)

   Penurunan ini dilakukan dengan cara simbol variabel yang terletak paling kanan akan diperluas terlebih dahulu.

   Contoh, terdapat tata bahasa:

   $$S \rightarrow aAS \mid a$$
   $$A \rightarrow SbA \mid ba$$

   untuk memperoleh turunan akhir ‘aabbaa’ dari tata bahasa diatas, maka dilakukan penurunan:

   S=> aAS => aAa => aSbAa => aSbbaa => aabbaa

2.11 PHP

PHP adalah singkatan dari *Hypertext Preprocessor* dimana bahasa ini adalah bahasa *script* secara *server-side* yang dirancang khusus untuk pengembangan web secara statis maupun dinamis[21]. PHP merupakan bahasa yang dapat disematkan ke dalam kode HTML dan bahasa PHP harus dijalankan atau diproses melalui server terlebih dahulu sebelum dikirim ke *client.*

2.12 Software Pendukung

2.12.1 XAMPP

XAMPP adalah sebuah program *all-in-one* dimana software tersebut sudah berupa web server Apache, database server MySQL, dan mendukung bahasa pemrograman PHP[21]. XAMPP digunakan sebagai pengembangan web berbasis lokal komputer dimana software ini gratis, mudah untuk digunakan, dan tersedia pada Windows, MacOS, dan Linux.

2.12.2 MySQL

MySQL merupakan software database relasi atau Relational Database Management System (RDBMS) dimana software ini digunakan sebagai pengakses database untuk aplikasi multi user[21]. MySQL terdistribusi gratis dibawah lisensi GPL (General Public License) sehingga tidak bisa dijadikan produk turunan closed source atau komersial[22].

2.12.3 Code Igniter

Code Igniter merupakan sebuah framework PHP yang dikembangkan oleh Rick Ellis, CEO Ellislab, Inc, dimana framework ini menggunakan metode MVC (Model, View, Controller) untuk memudahkan pengembangan suatu website tanpa membuat dari awal[22]. Kelebihan dari framework codeigniter adalah ukuran yang relatif lebih kecil yang hanya sebesar 2 MB dan memiliki lisensi dibawah Apache/BSD *open source* sehingga bersifat gratis.

2.12.4 Visual Studio Code

Visual Studio Code adalah software teks editor yang dikembangkan oleh Microsoft dan kontributor di dunia dengan framework Electron sehingga dapat dijalankan pada lintas sistem operasi seperti Windows, Linux, dan macOS[23]. Software visual studio code ini memiliki fitur *debugger* Node.js beserta *debug* lainya seperti *.Net* dan *Mono*, serta mendukung banyak bahasa pemrograman

dengan dukungan bahasa tambahan melalui ekstensi (bahasa seperti Go, Python, Java, C++, C#).

2.13 Pengujian Akurasi

Proses pengujian dilakukan untuk mengetahui nilai akurasi dari sistem yang dikembangkan, dimana pengujian dilakukan dengan cara membanding hasil translasi *source code* menggunakan sistem dengan *source code* yang diharapkan. Adapun nilai akurasi yang didapatkan dihitung dengan rumus berikut [24].

$$Akurasi = \frac{Jumlah\ hasil\ source\ code\ sistem\ benar}{Jumlah\ sampel\ data\ uji} \times 100\%$$