

## **BAB 2 LANDASAN TEORI**

### **2.1 Penerjemah**

Penerjemah merupakan orang yang meneliti sebuah penerjemahan yaitu proses dimana terdapat upaya mengalihkan pesan dari satu bahasa (bahasa sumber) ke bahasa yang lain (bahasa sasaran) atau upaya mengungkapkan kembali dalam suatu bahasa (bahasa sasaran) pesan dalam bahasa lain (bahasa sumber). Dengan demikian, bahasa sasaran merupakan “turunan” dari bahasa sumber. Tanpa bahasa sumber, tidak akan ada bahasa sasaran. Tanpa teks sumber, tidak akan pernah ada teks sasaran. Namun, ada kalanya penerjemah belum tentu membuat akurasi tinggi dalam pengalihan bahasa dikarenakan bahasa yang dialihkan pun terdapat pemikiran atau unsur kebudayaan tertentu. [11]

### **2.2 Bahasa Alami**

Bahasa Alami merupakan bahasa sehari-hari manusia yang mudah dipahami oleh manusia itu sendiri. Saat ini bahasa alami tengah dikembangkan dengan berbagai teknik serta metode dengan tujuan agar komputer mampu memahami instruksi manusia melalui bahasa alami tersebut. [12]

### **2.3 Source Code**

Source code adalah sekumpulan deklarasi atau pun pernyataan dari bahasa pemrograman komputer yang ditulis dan dapat dibaca oleh manusia. Source code membuat programmer dapat berkomunikasi dengan komputer menggunakan beberapa perintah yang telah terdefinisi. [13]

### **2.4 PASCAL**

PASCAL adalah bahasa tingkat tinggi yang dirancang oleh Prof Niklaus Wirth dari Techival University di Zurich, Switzerland. Bahasa Pascal sendiri, berorientasi pada segala tujuannya yang dibangun oleh Prof. Niklaud. Pascal diambil dari Blaise Pascal seorang ahli matematika dan filosofi yang terkenal pada abad 17 berasal dari Perancis. Pascal mulai dipublikasikan pertama kali pada tahun 1971 dengan tujuan untuk mengajar komputer secara sistematis. Karena pascal merupakan pemrograman terstruktur, maka dalam waktu singkat Pascal telah

menjadi bahasa pemrograman yang populer. Pascal mempunyai beragam *compiler* yang digunakan, antara lain Turbo Pascal, GNU Pascal, Free Pascal, USCD Pascal dan Apple Pascal.

Pascal adalah bahasa pemrograman terstruktur. Bahasa Pascal merupakan bahasa tingkat tinggi, instruksinya ditulis dengan menggunakan kata dalam bahasa Inggris. Bahasanya mudah dipahami karena bentuk programnya terstruktur yang tersusun atas sejumlah blok (Sub Program). Blok-blok yang kecil selanjutnya dapat dipakai untuk membuat blok yang lebih besar. Suatu permasalahan dapat dipecah menjadi bagian-bagian kecil sehingga mudah dikodekan. Kesalahan yang terjadi dalam program akan mudah ditelusuri. Selain itu, program dapat dimodifikasi tanpa menimbulkan efek sampingan terhadap bagian lain dari program [14]

## **2.5 Manipulasi String PASCAL**

String adalah sebuah bentuk runtunan karakter yang dapat membentuk sebuah kalimat, dalam bahasa pemrograman String dapat dikategorikan sebagai sebuah variabel, variabel ini dapat di manipulasi sedemikian rupa dengan perintah-perintah yang ada dalam suatu bahasa pemrograman. Manipulasi Variabel String dapat dicapai dengan hasil yang berbeda-beda. Variabel tersebut dapat di rubah panjangnya, disisipkan sebuah karakter baru, dikonversikan kedalam tipe data lain, dan masih banyak hal yang dapat dilakukan dengan Manipulasi String. [15]

Pada umumnya setiap bahasa pemrograman tingkat tinggi memiliki fungsi untuk melakukan manipulasi terhadap variabel string, Suatu String dalam Bahasa Pascal dapat dioperasikan dengan berbagai macam tujuan. Pascal menyediakan berbagai prosedur standar dan fungsi standar untuk operasi string. Dalam Operasinya String hanya mempunyai sebuah operator, yaitu operator '+'. Bila operator ini digunakan untuk dua buah elemen string, maka akan menjadi penggabungan dua buah string menjadi satu. Panjang Maksimal suatu string yang diijinkan oleh Pascal adalah 255 karakter.

Manipulasi string pada pascal pada umumnya menggunakan beberapa set perintah seperti DELETE, INSERT, STR, VAL, CONCAT, COPY, POS, LENGTH. String pada bahasa pascal hanya memiliki satu operator yaitu operator

'+', operator '+' pada kasus manipulasi string pascal akan menggabungkan 2 atau lebih string menjadi satu string yang dapat dijadikan sebuah output. Misalnya string A = 'rajin' dan string B = 'belajar' dan kemudian dilakukan operasi string A + string B maka akan menghasilkan sebuah string 'rajinbelajar' (tanpa spasi, spasi pada pascal harus di definisikan manual). [16]

Fungsi manipulasi string yang terdapat pada bahasa pemrograman pascal dan implementasi dari fungsi tersebut diantaranya:

### 2.5.1 Pos

Mencari urutan atau posisi dari sebuah substring yang terletak di dalam sebuah string, bentuk umum dari penggunaan fungsi ini : POS(substring, string).

Contoh implementasi :

```
var
    Kata,S : String[20];
begin
    S := 'ABCDEFGHJKLMNOPQRST';
    Kata := 'FGH';
    WriteLn(Kata, ' di posisi ', POS(Kata,S));
Readln;
end.

Output:   FGH di posisi 6
```

### 2.5.2 Insert

Menyisipkan string ke dalam suatu string lainnya, pada posisi I, bentuk umum dari penggunaan fungsi ini adalah : INSERT (string1, string2, i)

Contoh implementasi:

```
uses crt;
var
  s1, s2 : string;
begin
  clrscr;
  s1:='AS';
  s2:='PCAL';
  Insert(s1,s2,2);
  writeln('setelah disisipi=',s2);
  readln;
end.
```

Output:

```
setelah disisipi=PASCAL
```

### 2.5.3 Copy

Meng-*copy* atau menyalin sebuah substring dari suatu *string*, dimulai pada posisi (i) sebanyak (n) karakter, bentuk umum dari penggunaan fungsi tersebut adalah: COPY(*string*, i, n)

Contoh implementasi:

```
uses crt;
Var
  s,r : string;
BEGIN
  clrscr;
  s := 'informatika';
  r := copy (s,3,6);
  WriteLn ('Nilai r(COPY) adalah ', r);
```

```
readln;
END.
Output:
Nilai r(COPY) adalah format
```

#### 2.5.4 Length

Memberikan nilai panjang atau jumlah karakter dari suatu string, dengan kata lain memberikan output berupa nilai integer dari panjang string, misal pada kata PASCAL terdapat 6 huruf yang membentuk kata tersebut.

Contoh implementasi:

```
VAR
    s : string[20];
BEGIN
    s := 'Pascal';
    WriteLn('Length : ', ''',s, ''', ' adalah ',
length(s));
Readln;
END.
Output : Length : "Pascal" adalah 6
```

#### 2.5.5 Concat

Menggabungkan dua atau lebih *string* secara berturut-turut, fungsionalitas dari fungsi ini hampir sama dengan operasi *string* dasar dengan operator penjumlahan (+), bentuk umum dari fungsi tersebut adalah:

CONCAT(*string1*, *string2*, *string3*, ... *stringN*)

Contoh implementasi:

```
uses crt;
VAR
S1, S2 : string[8] ;
S3 : string[20];
BEGIN
```

```

clrscr;
    S1 := 'infor';
    S2 := 'matika';
    S3 := CONCAT(S1,S2);
    Write ('Setelah digabung = ', S3);
    readln;
END.
Output:
Setelah digabung = informatika

```

## 2.6 Natural Language Processing

Natural Language Processing (NLP) merupakan bagian dari salah satu ilmu komputer, NLP merupakan cabang dari kecerdasan buatan, dan bahasa (linguistik). NLP berkaitan dengan interaksi antara komputer dan bahasa alami manusia, seperti bahasa Inggris atau bahasa Indonesia. Tujuan utama dari studi NLP adalah menciptakan mesin yang mampu mengerti dan memahami makna bahasa manusia lalu memberikan respon yang sesuai. [17]

## 2.7 Case Folding

Case folding adalah proses dimana membuat data masukan berbahasa Indonesia yang bercampur antara huruf-huruf alphabet besar dan juga kecil, menjadi sama rata yaitu kecil (lowercase). Contoh dari case folding pada penelitian ini adalah:

**Tabel 2.1 Contoh Case Folding**

Data <i>Input</i> Bahasa Indonesia	Setelah <i>Case Holding</i>
Panggil fungsi luas bervariabel panjang dan lebar.	panggil fungsi luas bervariabel panjang dan lebar.

## 2.8 Filtering

Pada penelitian ini filtering dilakukan untuk menghilangkan karakter yang tidak diperlukan dalam tahap selanjutnya yaitu scanning, karakter yang diperlukan pada penelitian ini mencakup huruf a-z, 0-9, ‘\_’, ‘,’, ‘.’, ‘-’, ‘+’, ‘-’, ‘/’, ‘\*’, dan spasi. Apabila terdapat karakter yang tidak sesuai dengan karakter yang telah disebutkan maka akan dihilangkan pada proses filtering. Contoh filtering yang akan dilakukan pada penelitian ini adalah:

**Tabel 2. 1 Contoh Filtering**

Hasil Sebelumnya Dari <i>Case Holding</i>	Setelah <i>Case Filtering</i>
buat program persegi panjang, buat “panjang” dan “lebar” bertipe data integer.	buat program persegi panjang, buat panjang dan lebar bertipe data integer.

## 2.9 Scanning

Tahap pemilihan teks masukan supaya menjadi token-token menurut masing-masing kelasnya [18]. Di penelitian ini, proses scanning akan menerima masukan berupa runtunan kata yang kemudian akan dipilih dan dipisahkan menjadi satuan leksikal atau biasa disebut token. Token tersebut lantas akan menjadi data masukan pada tahap parsing. Proses scanning pada penelitian ini didasarkan pada Finite State Automata yang terkandung di dalam suatu bahasa reguler [18], adapun token yang berhasil dideteksi oleh aplikasi akan dibandingkan terhadap token-token yang telah terdefinisi di penelitian sebelumnya dan token baru yang akan disimpan pada basis data. Contoh scanning pada penelitian ini dapat dilihat pada tabel:

**Tabel 2. 2 Contoh Scanning**

Input	Output	
Hitung karakter kehidupan	Hitung karakter	keyword
	kehidupan	string

Adapun notasi leksikal kebahasaan yang digunakan pada penelitian ini adalah:

### **2.9.1 Identifier**

Pada penelitian ini identifier akan dibagi menjadi dua jenis yaitu keyword dan penamaan. Keyword sendiri adalah sebuah kata yang telah dipesan atau ditempati kedudukannya dalam suatu aturan kebahasaan pada kasus ini bahasa pemrograman. Keyword akan memicu sebuah perintah, contoh yang akan dilakukan pada penelitian ini adalah keyword 'program' yang akan memicu inisialisasi pada bahasa pemrograman Pascal. Penamaan merupakan kata yang dapat di definisikan sendiri oleh pengguna, misalnya penamaan suatu variabel akan bebas ditentukan karena kedudukannya tidak ditempati oleh kata kunci manapun contohnya variabel x yang artinya sebuah variabel bernama x.

### **2.9.2 Operator**

Operator mencakup operator pada operasi aritmetik (+-\*/) dan operasi logika (TRUE,FALSE).

### **2.9.3 Delimiter**

Delimiter mencakup tanda baca yang berfungsi untuk memisahkan atau membatasi sebuah ekspresi, beberapa tanda baca yang sering digunakan sebagai delimiter meliputi ('.' ',' ';').

### **2.9.4 Nilai Konstanta**

Nilai konstanta meliputi konstanta yang dapat dimasukkan kedalam suatu bahasa pemrograman pada kasus ini bahasa Pascal, konstanta ini dapat pula disebut 'nilai yang dimasukkan dan tidak akan berubah' pada bahasa pemrograman, salah satu contoh:  $X := 3$ , 3 merupakan konstanta karena nilainya telah didefinisikan terlebih dahulu di dalam suatu program, nilai konstanta memiliki jenis yang beragam mulai dari integer berupa angka bulat sampai string atau susunan karakter yang membentuk kalimat/kata.

## 2.10 Parsing

Proses parsing bertujuan untuk memberikan analisis sintaksis terhadap kemunculan token-token. Terdapat dua metode parsing yang digolongkan berdasarkan alur penurunannya, Metode top-down yang akan melakukan penurunan dari atas/root menuju bawah/leaf. Metode top-down dibagi lagi berdasarkan tipe algoritma yang digunakan, brute force dan recursive descent parser. Penelitian ini menggunakan metode top-down dengan algoritma brute force, dimana aturan produksi yang akan diturunkan dipilih dari ujung kiri, aturan produksi akan diturunkan simbol variabelnya sampai simbol yang tersisa hanya tinggal simbol terminalnya. Metode top-down dengan algoritma brute force akan mencoba berbagai kombinasi untuk terhadap simbol non-terminal (simbol variabel sebelum simbol terminal) apabila ditemukan kesalahan atau ketidakcocokan simbol akan dilakukan *backtrack* sampai kombinasi simbol non-terminal yang benar ditemukan [18].

Contoh dari metode parsing top-down terdapat pada aturan produksi:

$$S \rightarrow aAd \mid aB$$

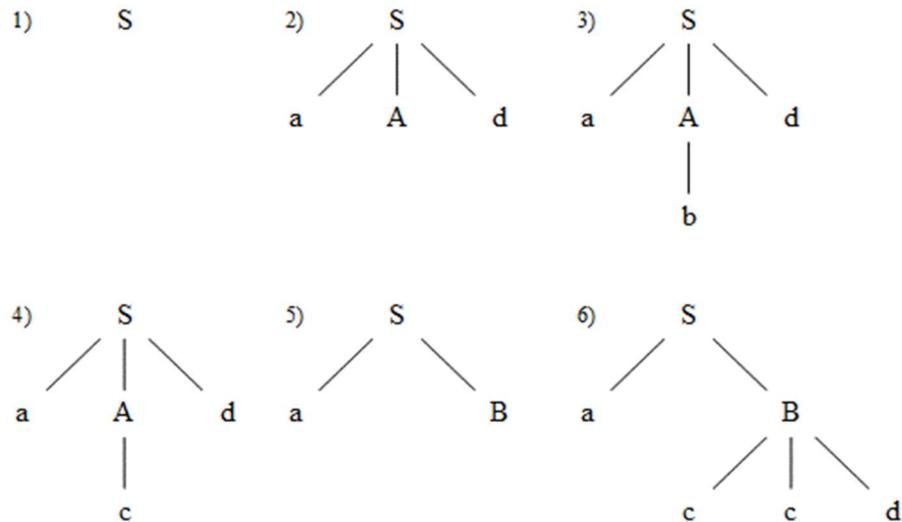
$$A \rightarrow b \mid c$$

$$B \rightarrow ccd \mid ddc$$

Misalnya dari aturan produksi diatas dilakukan parsing untuk *string* 'accd'.

$S \rightarrow aAd \mid aB$

$A \rightarrow b \mid c$



**Gambar 2.1 Pohon Penurunan**

Tahap yang ada pada Gambar 2.1, terlihat pada penurunan nomor 3 terdapat kesalahan, maka dilakukan *backtrack* pada penurunan 4. Namun masih terdapat kesalahan, maka akan melakukan *backtrack* lagi sampai semua *string* bisa diturunkan.

## 2.11 Hypertext Preprocessor (PHP)

PHP merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah server-side embedded scriptlanguage artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web browser tetapi prosesnya secara keseluruhan dijalankan di server, pada prinsipnya server akan bekerja apabila ada permintaan dari client. Dalam hal ini sistem menggunakan kode-kode PHP untuk mengirimkan permintaan ke server [19].

## 2.12 Metode Pengujian

Metode pengujian dilakukan dengan dikalkulasikan dengan cara membandingkan data uji masukan yang memiliki keluaran yang sesuai harapan, dengan seluruh data uji masukan, hasil dari kalkulasi tersebut kemudian di persentase, persentase tersebut akan dijadikan sebagai hasil akhir tingkat akurasi dari penerjemah, formula dari metode penelitian dapat dilihat pada gambar dibawah ini.

$$\text{Rata - rata Akurasi} = \frac{\text{Jumlah hasil translasi benar}}{\text{jumlah seluruh data uji}} \times 100\%$$

**Gambar 2.2 Formula Hitung Uji Akurasi Akhir**

## 2.13 Unified Modeling Language (UML)

*Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [20]. Dalam pengembangan suatu perangkat lunak, UML digunakan untuk memodelkan suatu sistem yang menggunakan konsep berorientasi object agar lebih bisa dipahami oleh banyak pihak yang terlibat dalam pengembangan.

Terdapat beberapa diagram yang biasanya digunakan untuk memodelkan analisis fungsional dalam rangka pengembangan perangkat lunak. Berikut diantaranya diagram yang umum digunakan :

### 2.13.1 Use Case Diagram

Menggambarkan sejumlah external actors dan hubungannya ke use case yang diberikan oleh sistem. Use case adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari use case symbol. Use case digambarkan hanya yang dilihat dari luar oleh actor dan bukan bagaimana fungsi yang ada di dalam sistem.

### **2.13.2 Activity Diagram**

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi. Activity diagram dibuat sebanyak aktivitas yang digambarkan pada use case diagram.

### **2.13.3 Class Diagram**

Menggambarkan struktur statis class di dalam sistem. Class merepresentasikan sesuatu yang ditangani oleh sistem. Class dapat berhubungan dengan yang lain melalui berbagai cara: associated (terhubung satu sama lain), dependent (satu class tergantung/menggunakan class yang lain), specialized (satu class merupakan spesialisasi dari class lainnya), atau package (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa class diagram.

### **2.13.4 Sequence Diagram**

Menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.