

**SKRIPSI**

**APLIKASI PEMERIKSA KESALAHAN DOKUMEN SKRIPSI  
INFORMATIKA UNPAR**



**Marcell Trixie Alexander**

**NPM: 2014730003**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2019**



**UNDERGRADUATE THESIS**

**GENERAL ERROR CHECKER APPLICATION FOR  
INFORMATICS ENGINEERING UNPAR THESIS  
DOCUMENT**



**Marcell Trixie Alexander**

**NPM: 2014730003**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2019**



## **ABSTRAK**

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

**Kata-kata kunci:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»



## **ABSTRACT**

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

**Keywords:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»





## DAFTAR ISI

DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	1
1.3 Tujuan . . . . .	1
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi Penelitian . . . . .	2
1.6 Sistematika Pembahasan . . . . .	2
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 <i>Regular Expression</i> . . . . .	5
2.1.1 Teori <i>Regular Expression</i> . . . . .	5
2.1.2 <i>Metacharacters</i> . . . . .	5
2.1.3 <i>Character Classes</i> . . . . .	6
2.1.4 <i>Character Shorthands</i> . . . . .	6
2.1.5 <i>Quantifiers</i> . . . . .	6
2.2 <i>Library PDF Parser</i> . . . . .	7
<b>3 ANALISIS MASALAH</b>	<b>9</b>
3.1 Survei Kesalahan Umum . . . . .	9
3.1.1 Pengamatan Sidang . . . . .	9
3.1.2 Wawancara Personal . . . . .	11
DAFTAR REFERENSI	13
A KODE PROGRAM	15
B HASIL EKSPERIMEN	17



## DAFTAR GAMBAR

B.1 Hasil 1 . . . . .	17
B.2 Hasil 2 . . . . .	17
B.3 Hasil 3 . . . . .	17
B.4 Hasil 4 . . . . .	17



## DAFTAR TABEL

2.1	Tabel metacharacters . . . . .	6
3.1	Tabel rincian sidang skripsi yang diamati . . . . .	9



# BAB 1

## PENDAHULUAN

Pada bab pendahuluan ini dijelaskan mengenai latar belakang penulisan skripsi, rumusan masalah, tujuan penulisan skripsi, batasan masalah, metodologi penelitian, dan sistematika penulisan skripsi ini.

### 1.1 Latar Belakang

Skripsi merupakan karangan ilmiah yang wajib ditulis oleh mahasiswa sebagai bagian dari persyaratan akhir pendidikan akademiknya. Namun dalam penulisannya, peserta skripsi sering melakukan kesalahan kecil. Kesalahan sering terjadi dalam penggunaan imbuhan, kata keterangan, penulisan kata dan sebagainya. Hal-hal seperti ini seharusnya dapat diperiksa dan diminimalisir oleh diri sendiri. Pada saat bimbingan, waktu dosen pembimbing lebih baik dimanfaatkan untuk membahas konten dibanding memeriksa kesalahan-kesalahan tersebut.

Dari masalah tersebut dapat dibuat sebuah aplikasi untuk melakukan pemeriksaan pada dokumen skripsi. Kesalahan yang akan diperiksa berasal dari survei yang dilakukan kepada dosen-dosen Informatika Unpar. Hasil dari survei tersebut akan diseleksi untuk diimplementasikan ke dalam aplikasi. Aplikasi sederhana ini dapat dimanfaatkan oleh mahasiswa Informatika Unpar secara mandiri. Aplikasi ini dijalankan melalui terminal *command Windows*. Aplikasi menerima masukan berupa file *PDF* skripsi dan menampilkan laporan yang berisi kesalahan-kesalahan yang ditemukan pada dokumen skripsi.

### 1.2 Rumusan Masalah

Berdasarkan deskripsi topik yang sudah ditulis, dapat dirumuskan masalah sebagai berikut:

1. Bagaimana cara memeriksa kesalahan yang ada pada dokumen skripsi?
2. Bagaimana cara membuat perangkat lunak yang dapat memeriksa kesalahan pada dokumen skripsi?

### 1.3 Tujuan

Tujuan dari skripsi ini adalah sebagai berikut:

1. Dapat memeriksa kesalahan yang ada pada dokumen skripsi

2. Dapat membangun perangkat lunak untuk memeriksa kesalahan yang ada pada dokumen skripsi

## 1.4 Batasan Masalah

Batasan masalah skripsi ini adalah sebagai berikut:

1. Pemeriksaan menggunakan *pattern matching* tanpa analisis gramatikal
2. Tidak ada pemeriksaan kosakata

## 1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan pada skripsi ini adalah sebagai berikut:

1. Melakukan survei kepada dosen-dosen Informatika mengenai kesalahan-kesalahan penulisan yang ditemui dalam dokumen skripsi
2. Melakukan studi literatur *Regular Expression* untuk mendeteksi kesalahan-kesalahan dalam file *PDF* skripsi
3. Mempelajari *library PDF Parser* untuk mengekstraksi file *PDF* skripsi yang akan diperiksa
4. Melakukan perancangan perangkat lunak
5. Melakukan implementasi perancangan perangkat lunak
6. Melakukan pengujian terhadap perancangan perangkat lunak
7. Menulis dokumen skripsi

## 1.6 Sistematika Pembahasan

Sistematika penulisan pada skripsi ini terdiri dari 6 bab, yaitu:

1. Bab 1 Pendahuluan  
Bab 1 akan membahas latar belakang dibuatnya perangkat lunak untuk memeriksa kesalahan dokumen skripsi. Pada bab ini dibahas juga rumusan masalah, tujuan skripsi, batasan masalah dan metodologi penelitian yang digunakan pada skripsi.
2. Bab 2 Landasan Teori  
Bab 2 yang merupakan landasan teori akan berisi teori-teori yang menjadi dasar-dasar dalam penulisan skripsi ini. Teori yang akan dibahas pada bab 2, yaitu *Regular Expression* dan *library PDF Parser*.
3. Bab 3 Analisis Masalah  
Bab 3 berisi analisis masalah yang muncul dalam menyelesaikan masalah tersebut. Pada bab ini akan dianalisa masalah yang ditemukan pada saat melakukan pengamatan beberapa sidang



skripsi semester Ganjil 2018/2019 dan survei secara personal kepada dosen-dosen Informatika Unpar. Selain itu, akan dibahas mengenai hal-hal yang dibutuhkan oleh perangkat lunak yang akan dibuat.

#### 4. Bab 4 Perancangan

Bab 4 berisi rancangan perangkat lunak yang akan dibuat. Perangkat lunak akan dibangun menggunakan bahasa pemrograman *PHP*.

#### 5. Bab 5 Implementasi dan Pengujian

Bab 5 pada skripsi ini membahas implementasi perangkat lunak dan pengujian yang dilakukan terhadap perangkat lunak tersebut. Bab ini juga menjelaskan tentang spesifikasi perangkat lunak dan pengujian yang dilakukan pada skripsi ini.

#### 6. Bab 6 Kesimpulan dan Saran

Bab 6 berisi kesimpulan dari penulisan skripsi ini. Bab ini juga berisi saran untuk pengembangan perangkat lunak agar lebih baik lagi.



## BAB 2

### LANDASAN TEORI

Pada bab 2 akan diuraikan tentang landasan teori yang membahas *regular expression* dan *library PDF Parser*.

#### 2.1 *Regular Expression*

Pada subbab ini akan diuraikan tentang teori *regular expression*, *metacharacters*, *character classes*, *character shorthands* dan *quantifiers*.

##### 2.1.1 Teori *Regular Expression*

*Regular Expression* atau yang sering disebut dengan *Regex* adalah teks string yang disandikan khusus untuk digunakan sebagai pola untuk mencocokkan set string tertentu. *Regex* mulai muncul pada tahun 1940-an sebagai cara untuk menggambarkan bahasa biasa, tetapi *Regex* benar-benar mulai muncul di dunia pemrograman selama tahun 1970-an. Istilah *Regular Expression* berasal dari teori matematika dan komputer sains, dimana istilah tersebut mencerminkan sifat ekspresi matematika yang disebut keteraturan. Ekspresi tersebut dapat diimplementasikan dalam perangkat lunak menggunakan *deterministic finite automaton* (DFA). DFA adalah *finite state machine* (FSA) yang tidak menggunakan *backtracking*.

*Regex* terdiri dari kombinasi antara *normalcharacter* dan *metacharacter* atau *metasequences*. *Normal character* mewakili karakter itu sendiri, sedangkan *metacharacter* dan *metasequences* merupakan karakter atau urutan karakter yang mewakili ide seperti kuantitas, lokasi, atau jenis karakter. *Regex* dapat digunakan untuk memverifikasi kecocokan antara input dengan pola teks, untuk menemukan teks yang cocok dengan pola dalam teks yang lebih besar, untuk mengganti teks yang cocok dengan pola dengan teks lain atau menyusun ulang bit dari teks yang cocok dan untuk membagi sebuah blok teks menjadi beberapa subteks.

##### 2.1.2 *Metacharacters*

Ada 14 metakarakter yang digunakan dalam *Regex*, masing-masing memiliki makna khusus yang akan dijelaskan pada tabel:

Tabel 2.1: Tabel metacharacters

Metakarakter	Nama	Tujuan
.	Full Stop	Cocok untuk karakter apapun
\	Backslash	Escape karakter
	Vertical Bar	Alternasi
^	Circumflex	Awal dari line anchor
\$	Dollar Sign	Akhir dari line anchor
?	Question Mark	Nol atau satu kuantifier
*	Asterisk	Nol atau lebih kuantifier
+	Plus Sign	Satu atau lebih kuantifier
[	Left Square Bracket	Membuka kelas karakter
]	Right Square Bracket	Menutup kelas karakter
{	Left Curly Bracket	Membuka kuantifier atau blok
}	Right Curly Bracket	Menutup kuantifier atau blok
(	Left Parenthesis	Membuka grup
)	Right Parenthesis	Menutup grup

### 2.1.3 Character Classes

#### 1. Negated Character Classes

#### 2. Union and Difference

#### 3. POSIX Character Classes

### 2.1.4 Character Shorthands

Tabel 2.2: Tabel shorthands

Character Shorthand	Fungsi
\d	Karakter digit
\D	Karakter non-digit
\s	Karakter <i>space</i>
\S	Karakter <i>non-space</i>
\w	Karakter <i>word</i>
\W	Karakter <i>non-word</i>

### 2.1.5 Quantifiers

#### 1. Greedy Quantifiers

12      2. Lazy Quantifiers

13

1      3. Possessive Quantifiers

2

3    **2.2    *Library PDF Parser***



## BAB 3

### ANALISIS MASALAH

Pada bab 3 akan diuraikan tentang survei kesalahan umum.

#### 3.1 Survei Kesalahan Umum

Pada subbab ini, akan diuraikan tentang survei yang dilakukan untuk mengumpulkan informasi yang dibutuhkan dalam pengembangan perangkat lunak. Informasi yang dicari adalah tentang kesalahan-kesalahan umum yang sering terjadi pada penulisan dokumen skripsi. Untuk mengumpulkan informasi tersebut, metode yang dipilih adalah melakukan survei. Dalam pelaksanaannya, survei dibagi menjadi dua, yaitu pengamatan beberapa sidang skripsi dan wawancara secara personal kepada dosen-dosen Informatika Unpar.

##### 3.1.1 Pengamatan Sidang

Pengamatan dilakukan pada sidang skripsi semester Ganjil 2018/2019, yang berlangsung pada bulan Mei 2019. Tidak semua sidang skripsi yang berlangsung diamati, dari 42 sidang skripsi hanya diambil 7 sidang skripsi saja. Hal tersebut dilakukan dengan pertimbangan dari ke-7 sidang skripsi tersebut diuji oleh dosen Informatika yang berbeda-beda. Namun ada beberapa dosen Informatika yang tidak masuk dalam pengamatan, karena tidak dapat menghadiri sidang yang diuji oleh dosen tersebut. Berikut adalah rincian dari sidang skripsi yang telah diamati:

Tabel 3.1: Tabel rincian sidang skripsi yang diamati

No	Tanggal	Mahasiswa	Penguji 1	Penguji 2
1	15 Mei 2019	Osfaldo Mickael	Bu Vania Natali	Pak Elisati Hulu
2	16 Mei 2019	Ricky Wahyudi	Bu Cecilia Esti N.	Bu Veronica Sri M.
3	17 Mei 2019	Billy Adiwijaya	Pak Kristopher David H.	Pak Elisati Hulu
4	20 Mei 2019	Ihsan Fajari	Bu Rosa De Lima E. P.	Bu Veronica Sri M.
5	20 Mei 2019	Muhammad Adrian	Bu Rosa De Lima E. P.	Pak Pascal Alfian N.
6	23 Mei 2019	Ellena Angelica	Bu Natalia	Bu Veronica Sri M.
7	24 Mei 2019	Evelyn Wijaya	Pak Candra Wijaya	Pak Raymond Chandra P.

Dari ke-7 pengamatan yang dilakukan sesuai dengan tabel 3.1, didapatkan beberapa kesalahan-kesalahan yang terjadi dalam penulisan dokumen skripsi. Berikut adalah kesalahan-kesalahan yang disebutkan oleh dosen penguji pada sidang skripsi di atas beserta penjelasannya:

### 1. Penulisan Judul

Dalam penulisan judul, setiap huruf awal pada kata harus menggunakan huruf kapital. Hal ini berlaku untuk hampir semua jenis kata, seperti nama, tempat, sifat dan keterangan. Namun, ada beberapa pengecualian seperti preposisi (kata depan yang diikuti oleh kata lainnya), konjungsi (kata sambung), dan interjeksi (kata yang mengungkapkan seruan perasaan).

### 2. Penulisan kata

Mahasiswa paling sering melakukan kesalahan dalam penulisan kata, atau yang lebih sering disebut dengan *typo*. Kesalahan-kesalahan kecil seperti ini paling sering terjadi tanpa disadari.

### 3. Penggunaan imbuhan di- dan kata depan di

Kesalahan ini merupakan kesalahan yang sering terjadi dalam penulisan dokumen skripsi. Penulisan imbuhan di- disatukan antara imbuhan dengan kata dasarnya. Untuk kata depan, penulisannya dipisah antara kata depan dengan kata berikutnya. Pada umumnya diikuti oleh keterangan tempat atau waktu.

### 4. Tidak ada spasi setelah tanda baca titik

Sebuah kalimat akan diakhiri oleh tanda baca titik. Untuk melanjutkan kalimat baru, setelah tanda baca titik harus ada jarak 1 spasi untuk memisahkan kalimat sebelumnya dengan yang berikutnya. Terkadang mahasiswa lupa untuk memberikan jarak 1 spasi setelah tanda baca titik di akhir kalimat.

### 5. Awal kalimat tidak menggunakan huruf kapital

Setiap huruf pertama pada kata pertama dalam sebuah kalimat harus ditulis dengan huruf kapital.

### 6. Terdapat ruang kosong yang besar

Masalah ini sering ditemukan dalam penulisan dokumen skripsi, biasanya terjadi pada saat menyisipkan gambar atau tabel. Susunan atau ukuran gambar yang tidak tepat dapat mengakibatkan terciptanya ruang kosong yang besar.

### 7. Tidak ada spasi antar kata

Setiap kata dalam sebuah kalimat dipisahkan dengan jarak 1 spasi agar kalimat dapat dibaca dan dimengerti dengan baik.

### 8. Gambar tidak sesuai tempatnya

Pada PDF Latex, biasanya kesalahan ini karena mahasiswa tidak memberikan tag kepada gambar tersebut. Hal ini mengakibatkan posisi gambar tidak terletak pada tempat yang seharusnya.

### 9. Tidak ada keterangan untuk gambar dan tabel

Dalam penulisan dokumen skripsi, setiap gambar dan tabel perlu diberikan keterangan.

### 10. Jumlah subbab dalam 1 bab tidak boleh hanya 1

Dalam sebuah bab, biasanya jumlah subbab lebih dari 1. Kesalahan yang sering dilakukan oleh mahasiswa yaitu, hanya terdapat 1 subbab saja pada 1 bab. Apabila dalam bab tersebut hanya terdapat 1 subbab, lebih baik tidak perlu dibuat subbab.



### 3.1.2 Wawancara Personal

Survei dilanjutkan dengan melakukan wawancara secara personal kepada dosen-dosen Informatika Unpar. Berikut adalah hasil wawancara yang sudah dilakukan:

1. Wawancara pertama dilakukan pada tanggal 9 Juli 2019. Dosen yang menjadi narasumbernya adalah Keenan. Hasil wawancara:

- Cetak miring untuk bahasa asing  
Penggunaan kata dalam bahasa asing harus ditulis menggunakan cetak miring. Mahasiswa sering lupa untuk menulis cetak miring bahasa asing.
- Kalimat pengantar untuk setiap subbab  
Setiap penulisan bab dan subbab selalu diikuti dengan kalimat pengantar untuk memulai bab dan subbab tersebut. Kesalahan yang sering terjadi, yaitu mahasiswa seringkali lupa untuk menuliskan kalimat pengantar tersebut.
- Kelengkapan data skripsi  
Data skripsi harus diisi dengan lengkap sebagai bentuk identitas, seperti nama mahasiswa, NPM, dosen pembimbing, judul skripsi dan sebagainya. Hal-hal seperti seringkali lupa diisi karena terlalu fokus dalam mengerjakan konten-konten dalam skripsi.

2. Wawancara ke-2 dilakukan pada tanggal 9 Juli 2019 juga. Dosen yang menjadi narasumbernya adalah Pak Chandra Wijaya. Hasil wawancara:

- Letak keterangan untuk gambar dan tabel  
Kesalahan yang sering terjadi adalah letak dari penulisan keterangan tersebut. Keterangan pada gambar posisinya ada di bawah gambar, sedangkan keterangan pada tabel posisinya ada di atas tabel.
- Penggunaan bahasa yang benar  
KBBI menjadi kaidah dalam penulisan bahasa Indonesia. Mahasiswa terkadang salah memilih kata yang hendak ditulis dalam dokumen, padahal kata tersebut tidak sesuai dengan KBBI.

3. Wawancara ke-3 dilakukan pada tanggal 15 Juli 2019 juga. Dosen yang menjadi narasumbernya adalah Pak Husnul Hakim. Hasil wawancara:

- Rujukan untuk gambar dan tabel  
Setiap gambar dan tabel yang dimasukkan ke dalam dokumen skripsi, perlu dirujuk dalam sebuah paragraf. Mahasiswa sering lupa atau terlewat untuk merujuk gambar dan tabel tersebut.
- Penulisan pseudocode  
Dalam penulisan pseudocode hal-hal yang perlu diperhatikan antara lain nama method, masukan serta keluaran pada method dan no baris pada pseudocode.
- Penulisan kata hubung

1 4. Wawancara ke-4 dilakukan pada tanggal 16 Juli 2019 juga. Dosen yang menjadi narasumbernya  
2 adalah Bu Vania Natali. Hasil wawancara:

3 • Tahun skripsi pada cover skripsi

4 Penulisan tahun skripsi harus sama dengan tahun dimana mahasiswa mengambil skripsi  
5 tersebut. Kesalahan yang pernah terjadi, yaitu mahasiswa salah menuliskan tahun skripsi.  
6 Meskipun terlihat sepele, namun hal ini perlu diperhatikan.

7 • Konsistensi penggunaan kata

8 Mahasiswa harus konsisten dalam penulisan kata, misalnya kata *user* dan pengguna.  
9 Mahasiswa harus memilih antara memakai *user* atau pengguna.

10 • Penggunaan kata ganti orang

11 Dalam penulisan dokumen skripsi, tidak boleh ada kata ganti orang. Jika karya nonilmiah  
12 lebih santai karena memakai gaya bahasa nonformal, maka berbeda dengan karya ilmiah.  
13 Karya ilmiah memiliki aturan baku dan menggunakan bahasa formal.

14 5. Wawancara ke-5 dilakukan pada tanggal 16 Juli 2019 juga. Dosen yang menjadi narasumbernya  
15 adalah Bu Natalia. Hasil wawancara:

16 • Penulisan *equation*

17  
18 • Penulisan catatan kaki

19  
20 • Penulisan daftar pustaka  
21

## DAFTAR REFERENSI



# LAMPIRAN A

## KODE PROGRAM

Listing A.1: MyCode.c

```
1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf
```

Listing A.2: MyCode.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }
```



## LAMPIRAN B

### HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4