

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**

LEMBAR PENGESAHAN

APLIKASI PEMERIKSA KESALAHAN DOKUMEN SKRIPSI INFORMATIKA UNPAR

Marcell Trixie Alexander

NPM: 2014730003

Bandung, «tanggal» «bulan» 2019

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

«pembimbing utama/1»

«pembimbing pendamping/2»

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

APLIKASI PEMERIKSA KESALAHAN DOKUMEN SKRIPSI INFORMATIKA UNPAR

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» 2019

Meterai Rp. 6000

Marcell Trixie Alexander
NPM: 2014730003

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»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Bandung, «bulan» 2019

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
2 LANDASAN TEORI	3
2.1 <i>Regular Expression</i>	3
3 ANALISIS MASALAH	5
3.1 Survei Kesalahan Umum	5
4 PERANCANGAN	7
4.1 Perancangan Kelas	7
5 IMPLEMENTASI DAN PENGUJIAN	9
5.1 Implementasi	9
5.1.1 Lingkungan Implementasi	9
6 KESIMPULAN DAN SARAN	11
6.1 Kesimpulan	11
6.2 Saran	11
DAFTAR REFERENSI	13
A KODE PROGRAM	15
B HASIL EKSPERIMEN	17

DAFTAR GAMBAR

4.1	Diagram kelas Aplikasi Pemeriksa Kesalahan Dokumen Skripsi	7
B.1	Hasil 1	17
B.2	Hasil 2	17
B.3	Hasil 3	17
B.4	Hasil 4	17

DAFTAR TABEL

BAB 1

PENDAHULUAN

Pada bab 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 yang tidak dapat diabaikan. 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

BAB 2

LANDASAN TEORI

Pada bab ini akan dibahas mengenai landasan teori yang membahas *regular expression*, *library PDF Parser* dan kamus bahasa Indonesia *LibreOffice*.

2.1 *Regular Expression*

Regular expression (regex) [1] adalah jenis pola teks tertentu yang dapat digunakan pada banyak aplikasi modern dan bahasa pemrograman. *Regex* biasanya dimanfaatkan 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.

Regex sudah banyak digunakan dalam pencocokan pola, misalnya untuk validasi beberapa string seperti *username* dan *password*, alamat *e-mail*, alamat *IP* ataupun nomor telepon. Pemanfaatan *regex* dengan baik, dapat menyederhanakan banyak tugas pemrograman dan pemrosesan teks dalam kehidupan sehari-hari. Istilah *regex* berasal dari teori matematika dan komputer sains, yang mencerminkan sifat ekspresi dalam matematika yang disebut keteraturan. Ekspresi tersebut dapat diimplementasikan dalam perangkat lunak, dengan menggunakan *Deterministic Finite Automaton* (DFA). DFA adalah *finite state machine* yang tidak menggunakan *backtracking*.

Regex dapat digunakan dalam berbagai bahasa pemrograman, salah satunya yaitu, *Perl Compatible Regular Expression* (PCRE). PCRE [?] adalah serangkaian fungsi yang menerapkan pencocokan pola *regex* dengan menggunakan sintaks dan semantik yang sama dengan bahasa pemrograman *Perl* 5, meskipun ada beberapa sedikit perbedaan. Pada saat ini, implementasi yang digunakan sesuai dengan *Perl* versi 5.005.

2.1.1 Metakarakter

Metakarakter pada *regex* dibedakan menjadi 2 jenis berdasarkan dari posisinya, yaitu metakarakter *outside square brackets* dan metakarakter *inside square brackets*. Meskipun ada beberapa simbol metakarakter yang sama, namun fungsinya agak berbeda. Pada metakarakter *outside square brackets* terdapat 14 simbol, sedangkan metakarakter *inside square brackets* terdapat 3 simbol.

BAB 3

ANALISIS MASALAH

Pada bab ini akan dibahas survei kesalahan umum dan keputusan implementasi hasil survei.

3.1 Survei Kesalahan Umum

Pada bagian ini akan dijelaskan 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, melainkan 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. Data dari sidang yang akan diamati akan disajikan pada tabel ?? dan ??:

Tabel 3.1: Tabel informasi sidang skripsi yang diamati

Tanggal	Mahasiswa	Judul Skripsi	Penguji
15-05-2019	Osfaldo Mickael Oktavianus Naibaho	Sistem Informasi Penjualan Barang Pada Apotek	-Vania Natali, S.Kom, M.T. -Elisati Hulu, M.T.
16-05-2019	Ricky Wahyudi	Temu Kembali Gambar Menggunakan Fitur Surf dan Warna	-Dr.rer.nat. Cecilia Esti Nugraheni, ST, MT -Dr. Ir. Veronica Sri Moer- tini, MT.

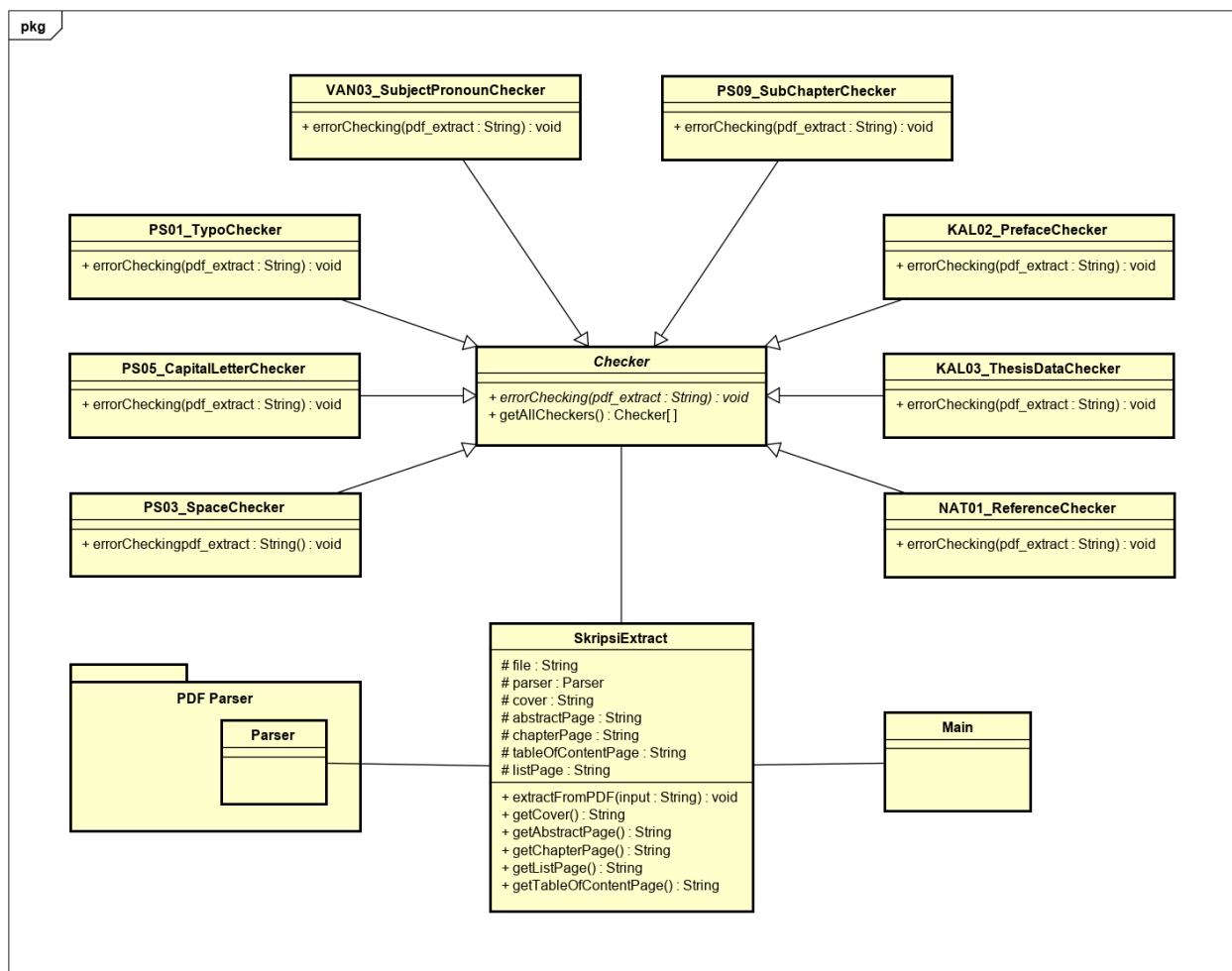
BAB 4

PERANCANGAN

Pada bab ini dibahas mengenai perancangan perangkat lunak yang dibangun, meliputi perancangan kelas dan algoritma pengecekan dokumen skripsi.

4.1 Perancangan Kelas

Pada bagian ini akan dijelaskan rancangan kelas yang akan digunakan pada perangkat lunak. Rancangan kelas tersebut akan ditunjukkan oleh diagram kelas di bawah ini:



Gambar 4.1: Diagram kelas Aplikasi Pemeriksa Kesalahan Dokumen Skripsi

BAB 5

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini dibahas mengenai implementasi perangkat lunak dan pengujian yang dilakukan terhadap perangkat lunak tersebut. Lingkungan implementasi, yang meliputi perangkat keras dan perangkat lunak, serta hasil implementasi akan dijelaskan pada bab ini. Selain Pengujian yang dilakukan pada skripsi ini, yang meliputi pengujian fungsional dan eksperimental akan dijelaskan pada bab ini.

5.1 Implementasi

Pada bagian ini akan dijelaskan mengenai lingkungan yang digunakan untuk membangun perangkat lunak beserta hasil implementasinya.

5.1.1 Lingkungan Implementasi

Berikut spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam pembangunan pada skripsi ini:

1. Spesifikasi Perangkat Keras

- Perangkat: Laptop
- Processor: AMD Bristol Ridge Quad Core FX-9830P 3GHz
- RAM: 8GB
- GPU: Radeon RX 460
- Storage: Harddisk 1TB

2. Spesifikasi Perangkat Lunak

- Sistem Operasi Windows 10 64-bit
- PHP 7.3.5 (cli)
- Composer versi 1.8.5
- Sublime Text versi 3.2.1

BAB 6

KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dari pembangunan aplikasi dan saran untuk pengembangan aplikasi ini.

6.1 Kesimpulan

6.2 Saran

DAFTAR REFERENSI

- [1] Goyvaerts, J. dan Levithan, S. (2012) *Regular Expressions Cookbook*, 2nd edition. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

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