

SKRIPSI

**APLIKASI PEMERIKSA KESALAHAN DOKUMEN SKRIPSI
TEKNIK 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 UNPAR
INFORMATICS ENGINEERING 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 TEKNIK INFORMATIKA UNPAR

Marcell Trixie Alexander

NPM: 2014730003

Bandung, 12 Desember 2019

Menyetujui,

Pembimbing

Pascal Alfadian, M.Comp.

Ketua Tim Penguji

Anggota Tim Penguji

Dr. Veronica Sri Moertini

Husnul Hakim, M.T.

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 TEKNIK 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 12 Desember 2019

Meterai Rp. 6000

Marcell Trixie Alexander
NPM: 2014730003

ABSTRAK

Skripsi merupakan karangan ilmiah yang wajib ditulis oleh mahasiswa sebagai bagian dari persyaratan akhir pendidikan akademiknya di perguruan tinggi. Namun, mahasiswa sering melakukan kesalahan-kesalahan dalam penulisan dokumen, misalnya dalam penggunaan imbuhan, kata keterangan, penulisan kata dan sebagainya. Mahasiswa seharusnya dapat memeriksa dan meminimalisir kesalahan sendiri, sehingga waktu bimbingan dapat dimanfaatkan untuk membahas konten skripsi. Dari permasalahan tersebut akan dikembangkan sebuah perangkat lunak yang dapat memeriksa kesalahan dokumen skripsi.

Perangkat lunak ini hanya dapat digunakan untuk memeriksa dokumen skripsi Informatika UNPAR. Perangkat lunak menerima sebuah masukan berupa lokasi dan nama dokumen skripsi. Perangkat lunak akan mengeluarkan sebuah laporan kesalahan yang ditemukan di dokumen skripsi. Perangkat lunak tidak menggunakan *Graphical Unit Interface*, sehingga masukan dan keluaran akan ditampilkan pada terminal.

Dokumen skripsi diekstrak menggunakan *Library PdfParser*, yang akan dipotong menjadi kalimat-kalimat. Kalimat tersebut akan diperiksa menggunakan metode pencocokan pola. Pola dibuat dengan *Regular Expression*. Kesalahan yang dapat diperiksa yaitu kesalahan yang bersifat tekstual.

Perangkat lunak telah diuji fungsionalitasnya dengan menggunakan dua kasus uji yang dibuat menggunakan template skripsi yang digunakan Fakultas Teknologi Informasi dan Sains. Dua kasus uji tersebut serupa, tetapi salah satu kasus uji disisipkan kesalahan-kesalahan. Pengujian dengan cara menggunakan kasus uji sebagai masukan dan membandingkan laporan kesalahan yang dikeluarkan dengan hasil yang diharapkan.

Kata-kata kunci: Informatika UNPAR, Pemeriksa dokumen, *PdfParser*, *Regular Expression*

ABSTRACT

Thesis is a scientific paper that must be written by students as part of the final requirement of academic education in university degree. But, students often make a gramatical mistakes in the process of document writing, e.g the usage of affix, adverb, word typing and etc. Students should be able to check and minimize their own mistakes, so that the consultation time can be used to discuss the content of the thesis. Based on the issue the will be developed a software that be able to check the errors of thesis document.

This software can only be used to check UNPAR Informatics thesis document. The software receives an input such as location and name of thesis document. The software will issue an error report that found in the thesis document. The software does not use the Graphical Unit Interface, so the input and output will be displayed on the terminal.

The thesis document is extracted by PdfParser Library, that will be parsed into the sentences. The sentence will be checked by the pattern matching method. Pattern made with Regular Expression. Errors that can be checked are textual errors.

The software has been tested for its functionality by two test cases, that created using a thesis template of Faculty of Information Technology and Science. The two test cases are similar, but one of the test cases are an error test. Testing by test cases as input and comparing the error report issued with the expected results.

Keywords: Document checker, *PdfParser*, *Regular Expression*, UNPAR Informatics

*Dipersembahkan kepada Tuhan YME, keluarga, teman-teman, dan
diri sendiri*

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas berkat yang telah diberikan kepada penulis sehingga dapat menyelesaikan skripsi dengan judul Aplikasi Pemeriksa Kesalahan Dokumen Skripsi Teknik Informatika UNPAR dengan baik. Penulis ingin menyampaikan rasa terima kasih kepada:

1. Keluarga penulis yaitu orang tua yang selalu mendukung dalam proses perkuliahan dan skripsi ini dengan baik. Serta kepada adik yang selalu mendukung dan menjadi teman bertukar pendapat dalam menyelesaikan skripsi ini.
2. Bapak Pascal Alfadian Nugroho, S.Kom, M.Comp sebagai dosen pembimbing yang telah membantu dan membimbing penulis dalam menyelesaikan skripsi ini.
3. Bapak Husnul Hakim, M.T. dan Ibu Dr. Ir. Veronica Sri Moertini, MT. sebagai dosen penguji yang telah menguji dan memberikan saran pada skripsi ini.
4. Ibu Mariskha Tri Adithia, P.D.Eng sebagai koordinator skripsi, yang memberikan informasi dan arahan pada mata kuliah skripsi 1 dan skripsi 2.
5. Seluruh dosen Informatika UNPAR yang telah memberikan ilmu kepada penulis selama masa perkuliahan.
6. Hendri Susanto, Michael Stevanus Valentino, Vincent Eka dan Krisogonus Ferdie Rendragraha sebagai teman seperjuangan dalam menyelesaikan mata kuliah skripsi 2.
7. Teman-teman yang selalu memberikan dukungan, menanyakan kabar, dan memberikan motivasi kepada penulis, yaitu Albert, Keenan, Stilmen, Reza, Hereza, Kalas, Andre, Fedrian, serta teman-teman penulis lain yang tidak dapat disebutkan satu persatu.

Akhir kata, semoga skripsi ini dapat bermanfaat bagi pembaca dan dapat menjadi dasar untuk penelitian yang terkait dengan skripsi ini.

Bandung, Desember 2019

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
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
2 LANDASAN TEORI	5
2.1 <i>Regular Expression</i>	5
2.1.1 Metakarakter	5
2.1.2 Kelas Karakter	8
2.2 PdfParser	10
2.3 Kamus Indonesia <i>LibreOffice</i>	10
2.3.1 File <i>id_ID.dic</i>	11
2.3.2 File <i>id_ID.aff</i>	11
3 ANALISIS MASALAH	15
3.1 Survei Kesalahan Umum	15
3.1.1 Pengamatan Sidang	15
3.1.2 Wawancara Personal	18
3.2 Masalah yang Ditangani Pada Perangkat Lunak	20
3.3 Analisis Cara Memeriksa Kesalahan Pada Dokumen Skripsi	23
4 ANALISIS DAN PERANCANGAN PERANGKAT LUNAK	25
4.1 Perancangan Metode Pemeriksaan	25
4.2 Perancangan Perangkat Lunak	26
4.3 Perancangan Algoritma	32
4.3.1 Algoritma untuk Mengekstrak Dokumen	33
4.3.2 Algoritma Pemeriksa Kesalahan	33
5 IMPLEMENTASI DAN PENGUJIAN	39
5.1 Implementasi	39
5.1.1 Lingkungan Implementasi	39
5.1.2 Hasil Implementasi	39
5.2 Pengujian Fungsional	40

5.2.1	Menguji fitur PS-01	41
5.2.2	Menguji fitur PS-03	43
5.2.3	Menguji fitur PS-05	44
5.2.4	Menguji fitur PS-09	45
5.2.5	Menguji fitur KAL-02	46
5.2.6	Menguji fitur KAL-03	46
5.2.7	Menguji fitur NAT-01	47
5.2.8	Menguji fitur VAN-03	47
5.2.9	Kesimpulan Pengujian Fungsional	48
5.3	Pengujian Eksperimental	48
5.3.1	Perangkat Lunak Tidak Mengeluarkan Laporan Kesalahan	61
5.3.2	Perangkat Lunak Tidak Dapat Memeriksa Dokumen Skripsi	63
5.3.3	Kesimpulan Pengujian Eksperimental	63
6	KESIMPULAN DAN SARAN	65
6.1	Kesimpulan	65
6.2	Saran	65
	DAFTAR REFERENSI	67
	A KODE PROGRAM	69

DAFTAR GAMBAR

3.1	Flow Chart Aplikasi Pemeriksa Kesalahan Dokumen Skripsi	23
4.1	Diagram kelas yang disederhanakan	27
4.2	Kelas Checker	27
4.3	Kelas KAL02_PrefaceChecker	28
4.4	Kelas KAL03_ThesisDataChecker	28
4.5	Kelas NAT01_ReferenceChecker	29
4.6	Kelas PS01_TypoChecker	29
4.7	Kelas PS03_SpaceChecker	29
4.8	Kelas PS05_CapitalLetterChecker	30
4.9	Kelas PS09_SubChapterChecker	30
4.10	Kelas VAN03_SubjectProunounChecker	30
4.11	Kelas SkripsiExtract	31

DAFTAR TABEL

2.1	Tabel metakarakter <i>outside square brackets</i>	5
2.2	Tabel metakarakter <i>outside square brackets</i>	6
2.3	Tabel metakarakter <i>inside square brackets</i>	7
2.4	Tabel kelas karakter	8
2.5	Tabel kelas karakter	9
3.1	Tabel informasi sidang skripsi yang diamati	15
3.2	Tabel informasi sidang skripsi yang diamati	16
3.3	Tabel hasil pengamatan sidang skripsi	16
3.4	Tabel hasil pengamatan sidang skripsi	17
3.5	Tabel hasil wawancara dosen	18
3.6	Tabel hasil wawancara dosen	19
3.7	Tabel hasil wawancara dosen	20
3.8	Tabel keputusan implementasi	20
3.9	Tabel keputusan implementasi	21
3.10	Tabel keputusan implementasi	22
3.11	Tabel keputusan implementasi	23

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.

1.1 Latar Belakang

Skripsi merupakan karangan ilmiah yang wajib ditulis oleh mahasiswa sebagai bagian dari persyaratan akhir pendidikan akademiknya di perguruan tinggi. Namun, mahasiswa sering melakukan kesalahan-kesalahan dalam penulisan dokumen, misalnya dalam penggunaan imbuhan, kata keterangan, penulisan kata dan sebagainya. Hal-hal seperti itu seharusnya dapat diperiksa dan diminimalisir oleh diri sendiri. Pada saat bimbingan, waktu dosen pembimbing lebih baik dimanfaatkan untuk membahas konten dalam skripsi. Dosen pembimbing tidak memiliki waktu untuk memeriksa kesalahan-kesalahan dalam penulisan dokumen.

Pada saat memeriksa dokumen secara manual, seringkali ada beberapa kesalahan yang terlewat atau tidak terlihat oleh mahasiswa. Namun, dengan menggunakan komputer, tingkat keakuratan pemeriksaan akan lebih baik daripada manual. 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. Kesalahan-kesalahan tersebut akan diperiksa dengan menggunakan teknik *pattern matching* menggunakan *regular expression*.

Aplikasi sederhana ini dapat dimanfaatkan oleh mahasiswa Informatika UNPAR secara mandiri. Aplikasi akan memeriksa dokumen *PDF* skripsi Informatika UNPAR yang menggunakan mode *final*. Selanjutnya, aplikasi akan menampilkan laporan yang berisi kesalahan-kesalahan yang ditemukan pada dokumen tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang 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. Merumuskan cara untuk memeriksa kesalahan yang ada pada dokumen skripsi dengan menggunakan *regular expression*.
2. Membangun perangkat lunak untuk memeriksa kesalahan yang ada pada dokumen skripsi.

1.4 Batasan Masalah

Batasan masalah skripsi ini adalah sebagai berikut:

1. Jenis dokumen yang dapat diperiksa oleh perangkat lunak yang dibuat adalah dokumen skripsi Informatika UNPAR dengan mode *final*.
2. Pemeriksaan menggunakan *pattern matching* tanpa analisis gramatikal dengan *regular expression*
3. Pemeriksaan penulisan kata dasar bahasa Indonesia tanpa memperhitungkan imbuhan

1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan pada skripsi ini adalah sebagai berikut:

1. Melakukan survei kepada dosen-dosen Informatika UNPAR 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 dan eksperimen terhadap 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*, *library PDF Parser* dan kamus bahasa Indonesia *LibreOffice*.
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 wawancara secara personal kepada dosen-dosen Informatika UNPAR. Hasil dari setiap survei akan dipilih mana saja yang dapat diimplementasikan dalam perangkat lunak.
4. Bab 4 Analisis dan Perancangan Perangkat Lunak
Bab 4 berisi rancangan perangkat lunak yang akan dibuat, seperti perancangan kelas dan algoritma yang digunakan dalam pembangunan perangkat lunak. Perangkat lunak akan dibuat dengan 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 ini akan dibahas mengenai landasan teori yang membahas *regular expression*, *library PdfParser* 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 [2] 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

Karakter meta pada *regex* dibedakan menjadi 2 jenis berdasarkan dari posisinya, yaitu karakter meta *inside square brackets* dan karakter meta *outside square brackets*. Meskipun ada beberapa simbol karakter meta yang sama, namun fungsinya agak berbeda. Karakter meta *outside square brackets* memiliki 14 jenis karakter, sedangkan karakter meta *inside square brackets* hanya memiliki 3 jenis karakter. Rincian dari ke-2 karakter meta tersebut akan dijelaskan sebagai berikut:

Tabel 2.1: Tabel metakarakter *outside square brackets*

Simbol	Nama
\	Backslash
^	Circumflex Anchor
\$	Dollar Anchor

Tabel 2.2: Tabel metakarakter *outside square brackets*

Simbol	Nama
.	Dot
[Left Square Bracket
]	Right Square Bracket
	Vertical Bar
(Left Parenthesis
)	Right Parenthesis
{	Left Curly Bracket
}	Right Curly Bracket
?	Question Mark
*	Asterisk
+	Plus

Pada Tabel 2.1 dan 2.2, terdapat 14 karakter yang termasuk karakter meta *outside square brackets*. Setiap karakter yang ada pada tabel tersebut memiliki fungsi yang berbeda-beda. Berikut ini adalah penjelasan fungsi dari setiap karakter meta.

1. Backslash

Karakter meta *backslash* yang berada di luar kelas karakter memiliki beberapa fungsi, yaitu:

- Membuat karakter lepas
Apabila diikuti oleh karakter non-alfanumerik, karakter meta ini dapat menghilangkan makna khusus yang dimiliki oleh karakter tersebut. Misalnya pada saat ingin mencocokkan karakter `"*"`, dengan menggunakan karakter meta *backslash* dapat dituliskan dengan `"\"`. Jadi karakter *asterisk* akan terbaca sebagai karakter biasa bukan sebagai karakter meta.
- Menggunakan karakter yang tidak dapat ditulis dalam pola, seperti `\n` (*newline*), `\R` (*line break*), `\t` (*tab*) dan lain-lain.
- Menentukan jenis karakter dalam pola, seperti `\d` (angka desimal), `\D` (non-angka desimal), `\w` (karakter kata), `\W` (non-karakter kata) dan lain-lain.
- Menggunakan *assertions*, seperti `\b` (*word boundary*), `\B` (*non-word boundary*) dan lain-lain.

2. Anchor

Anchor merupakan karakter meta yang terdiri dari simbol *circumflex* (`^`) dan *dollar* (`$`). Simbol *Circumflex* digunakan untuk menandai awal dari pola pencarian, sedangkan simbol *dollar* digunakan untuk menandai akhir dari pola pencarian. Contohnya pola `"/^ [0-9].*[A-Za-z]$)/"`, yang artinya pola tersebut dimulai dengan karakter numerik dan diakhiri dengan karakter huruf.

3. Dot

Dot akan cocok dengan karakter apapun, kecuali karakter *line break*. Contohnya untuk pola

"/HelloW.rld/", pola tersebut dapat cocok dengan HelloWorld, HelloWirld, HelloWorld, dan seterusnya.

4. Square brackets

Square brackets terdiri dari pasangan "[" dan "]", memiliki fungsi untuk mendefinisikan kelas karakter. Kelas karakter akan berada di antara 2 karakter tersebut. Contohnya kelas karakter numerik "[0-9]" sama dengan "[0123456789]".

5. Vertical bar

Vertical bar atau garis tegak lurus, berfungsi untuk memisahkan beberapa kondisi pola alternatif yang berbeda. Sebagai contohnya untuk pola "/(A | B)/", maka hasilnya akan mencocokkan karakter "A" atau karakter "B".

6. Parenthesis

Parenthesis atau kurung biasa, merupakan karakter meta yang terdiri dari pasangan "(" dan ")". Karakter meta ini memiliki fungsi untuk mengelompokkan suatu pola dalam *regex*. Pola yang berada di dalam kurung paling dalam akan diolah terlebih dahulu, setelah itu baru mengolah tanda kurung yang ada di luarnya.

7. Curly Brackets

Curly Brackets atau kurung kurawal, merupakan karakter meta terdiri dari pasangan "{" dan "}". Karakter meta ini memiliki fungsi untuk memberi informasi jumlah karakter atau pola di dalam kurung siku yang harus ada. Kurung kurawal biasanya diletakkan setelah kurung siku. Misalnya pola "[a]{1,5}", akan cocok dengan "a", "aa", "aaa", "aaaa" atau "aaaaa".

8. Quantifiers

Quantifiers berfungsi untuk menunjukkan berapa banyak instance karakter, set karakter, atau kelas karakter yang harus dicocokkan. Pada karakter meta ini terdapat 3 jenis, yaitu:

- *Question mark* (?)

Pengulangan yang dapat dilakukan oleh kuantifier ini yaitu { 0,1 } (0 hingga 1). Contohnya pola "/(Hello)?World/" akan cocok dengan "World" atau "HelloWorld".

- *Asterisk* (*)

Pengulangan yang dapat dilakukan oleh kuantifier ini yaitu { 0, } (0 atau lebih). Contohnya pola "/(He)*lloWorld/" akan cocok dengan "lloWorld", "HelloWorld" atau "HeHelloWorld".

- *Plus* (+)

Pengulangan yang dapat dilakukan oleh kuantifier ini yaitu { 1, } (1 atau lebih). Contohnya pola "/(HelloWorld)+/" akan cocok dengan "HelloWorld", "HelloWorldHelloWorld" dan seterusnya.

Karakter meta *inside square brackets* merupakan bagian dari kelas karakter. Pada bagian kelas karakter ini hanya terdapat 3 macam karakter meta saja. Jenis-jenis karakter meta tersebut akan dijelaskan pada Tabel 2.3.

Tabel 2.3: Tabel metakarakter *inside square brackets*

Simbol	Nama
\	Backslash
^	Circumflex
-	Hyphen

Pada Tabel 2.3 telah disebutkan macam-macam karakter meta *inside square brackets*. Dari ke-3 metakarakter tersebut ada beberapa yang memiliki simbol yang sama dengan karakter meta *outside square brackets*, namun fungsinya berbeda. Fungsi dari setiap karakter meta tersebut akan dijelaskan sebagai berikut.

1. *Backslash*

Karakter meta ini fungsinya sama dengan *backslash* yang ada pada *outside square brackets*. Namun dari ke-4 fungsi tersebut hanya 3 fungsi saja yang digunakan, yaitu membuat karakter lepas, Menggunakan karakter yang tidak dapat ditulis dalam pola dan menentukan jenis karakter dalam pola.

2. *Circumflex*

Karakter meta ini memiliki fungsi yang berbeda dengan yang digunakan pada *outside square brackets*. Fungsinya untuk membuat negasi, namun hanya berlaku untuk karakter pertamanya saja. Contohnya `[^0]` akan cocok dengan semua karakter kecuali karakter 0.

3. *Hyphen*

Karakter meta ini berfungsi untuk menentukan jangkauan dari sebuah karakter dalam kelas karakter, seperti `[0-9]` yang menandakan jangkauan karakter dari angka 0 hingga 9.

2.1.2 Kelas Karakter

Kelas karakter adalah karakter yang memiliki atribut yang spesifik yang dikelompokkan dalam sebuah kelas. Karakter tersebut dapat berbeda di setiap negara. Kelas karakter hanya valid digunakan pada *regex* didalam tanda kurung siku pada *bracket expression*. Perl mendukung notasi POSIX yang digunakan untuk kelas karakter. Dalam penggunaannya, kelas-kelas tersebut ditulis diantara "[" dan "]". PCRE juga mendukung penggunaan notasi ini. Sebagai contoh untuk kelas alfanumerik, penulisannya yaitu `[:alnum:]`. Berikut ini akan dijelaskan macam-macam kelas karakter yang digunakan pada Tabel 2.4 dan 2.5.

Tabel 2.4: Tabel kelas karakter

Kelas	Deskripsi	Keterangan
alnum	Alfanumerik	Kelas yang berisi dengan karakter alfanumerik, meliputi angka dan huruf. Karakter-karakter yang termasuk yaitu, huruf a-Z, A-Z dan angka 0-9. Simbol atau karakter khusus tidak termasuk dalam kelas ini.
alpha	Huruf	Kelas yang berisi dengan karakter huruf. Karakter-karakter yang termasuk yaitu, huruf a-Z atau A-Z. Simbol atau karakter khusus tidak termasuk dalam kelas ini.
ascii	Kode karakter	Kelas yang merepresentasikan kode karakter dari 0-127.
blank	Spasi dan Tab	Karakter-karakter yang termasuk yaitu, TAB dan spasi.

Tabel 2.5: Tabel kelas karakter

Kelas	Deskripsi	Keterangan
blank	Spasi dan Tab	Karakter-karakter yang termasuk yaitu, TAB dan spasi.
cntrl	Karakter kontrol	Karakter kontrol adalah karakter yang tidak merepresentasikan simbol tetapi merepresentasikan character encoding.
digit	Angka desimal	Kelas yang berisi dengan karakter numerik. Karakter-karakter yang termasuk yaitu, angka 0-9.
graph	Karakter cetak (kecuali spasi)	Kelas yang berisi karakter yang dapat dicetak dan tampak. Contoh karakter yang dapat dicetak namun tidak tampak adalah karakter spasi dan TAB. Jadi pada kelas ini karakter spasi dan tab tidak termasuk.
lower	Huruf kecil	Kelas yang berisi karakter dengan huruf kecil. Karakter yang termasuk kelas ini adalah huruf a-z.
print	Karakter cetak (termasuk spasi)	Kelas yang berisi karakter yang dapat dicetak. Karakter yang termasuk kelas ini adalah spasi.
punct	Karakter cetak (kecuali alfanumerik)	Kelas yang berisi karakter yang dapat dicetak, namun tidak termasuk alfanumerik.
space	Ruang putih	Karakter yang termasuk kelas ini adalah spasi dan TAB.
upper	Huruf kapital	Kelas yang berisi karakter dengan huruf kapital.
word	Karakter "word"	Kelas yang berisi karakter kata.
xdigit	Heksadesimal	Kelas yang merepresentasikan bilangan heksadesimal, a-f atau A-F dan 0-9.

2.2 PdfParser

PdfParser [3] adalah sebuah *library* yang digunakan untuk mengekstrak data yang ada dalam sebuah file PDF. *Library* ini digunakan untuk kebutuhan pengguna yang menggunakan bahasa pemrograman PHP. *PDF Parser* dapat digunakan pada *PHP* dengan versi 5.3 ke atas. Terdapat beberapa fitur yang dimiliki oleh *library* ini. Berikut adalah fitur yang sudah dapat digunakan:

- Memuat / mengurai objek dan header.
- Menampilkan data meta, seperti nama penulis, deskripsi dan sebagainya.
- Menampilkan isi dari teks PDF.
- Dapat digunakan untuk kompresi PDF.
- Mendukung *MAC OS Roman charset encoding*.
- Menangani *encoding* heksa dan oktal pada teks.

Dari fitur yang sudah ada, masih ada beberapa hal yang masih belum bisa ditangani oleh *library* ini. Berikut adalah beberapa kekurangan yang dimiliki oleh *PdfParser* berdasarkan pengamatan:

- Tidak dapat mengekstrak PDF yang termasuk dokumen *secured*.
- Tidak dapat mendeteksi jenis teks (cetak miring, tebal dan garis bawah)
- Tidak dapat mengesktrak gambar
- Hasil ekstrak tabel hanya berupa string saja (tidak dalam bentuk tabel)

2.3 Kamus Indonesia *LibreOffice*

LibreOffice [4] adalah sebuah paket aplikasi perkantoran berfitur lengkap yang tersedia secara gratis. LibreOffice menggunakan *Open Document Format* (ODF) sebagai format aslinya untuk menyimpan dokumen. ODF menjadi format standar terbuka yang sedang diadopsi sebagai format file yang dibutuhkan untuk penerbitan dan penerimaan dokumen. LibreOffice juga dapat membuka dan menyimpan dokumen dalam format lainnya. Salah satunya format yang digunakan oleh beberapa versi dari Microsoft Office.

LibreOffice memiliki ekstensi kamus Indonesia yang digunakan pada LibreOffice Writer, untuk melakukan pemeriksaan kata yang dituliskan oleh pengguna. Ekstensi ini sudah mengalami beberapa perkembangan, mulai dari versi 1.0 yang rilis pada tanggal 19 Mei 2012. Versi 1.0 merupakan hasil unggahan kembali dari ekstensi *OpenOffice* yang terakhir diperbaharui pada tahun 2009. Pada tanggal 17 Mei 2014 versi 1.1 dirilis, pada versi ini LibreOffice versi 4.0 dapat menggunakan ekstensi ini. Selanjutnya, pada tanggal 15 Juli 2014 versi 2.0 dirilis. Pada versi yang paling baru ini digunakan metode baru dengan sirkumfiks yang kemudian mengubah daftar kata, sehingga memuat semua lema dari Kamus Besar Indonesia.

Ekstensi kamus Indonesia ini dapat diunduh secara gratis oleh para pengguna melalui halaman resmi dari *LibreOffice*. Ekstensi yang dapat diunduh tersebut memiliki nama *id_id.ort*. Dalam ekstensi tersebut, terdapat beberapa file yang berisi informasi tentang kamus Indonesia, yaitu *id_ID.aff* dan *id_ID.dic*.

2.3.1 File *id_ID.dic*

File ini berisi kata-kata yang akan digunakan dalam kamus bahasa Indonesia. File ini dapat dibuka dengan menggunakan aplikasi *notepad* atau teks editor lainnya. Setelah dibuka menggunakan teks editor, akan terlihat 31129 baris yang berisi kata-kata dasar yang ada dalam kamus bahasa Indonesia *LibreOffice*. Untuk menjelaskan hal tersebut, akan diambil beberapa potong baris dari file tersebut.

Listing 2.1: Potongan kode untuk file *id_ID.dic*

```

1 | 31128
2 | a
3 | ab
4 | aba
5 | aba-aba
6 | abad/i0
7 | abadi/DkMkO0k0nl
8 | abadiah
9 | abah/DkMk
10 | abah-abah

```

Pada Listing 2.1, baris pertama pada file ini menyatakan banyaknya kata yang ada pada kamus. Kata yang ada dalam kamus tersebut berjumlah 31128 buah. Baris ke-2 hingga ke-31129 merupakan kata-kata yang ada pada kamus. Dalam kamus ini belum dapat pengelompokan kata berdasarkan jenisnya, misalkan kata benda, kata sifat, dan sebagainya. Namun ada beberapa kata tersebut ada yang terlihat berbeda, seperti pada baris 6, 7 dan 9. Di sebelah kiri dari kata tersebut terdapat garis miring yang disertai sebuah kode. Kode tersebut berarti bahwa kata tersebut dapat ditambahkan oleh imbuhan tertentu. Penjelasan tentang kode tersebut akan diuraikan lebih lanjut pada pembahasan file *id_ID.aff*.

2.3.2 File *id_ID.aff*

File ini berisi daftar prefiks, sufiks dan konfiks, untuk kamus Indonesia. Prefiks akan disimbolkan dengan huruf kapital, sedangkan sufiks akan disimbolkan dengan huruf kecil. Selain ke-2 hal tersebut, ada juga yang merupakan gabungan dari prefiks dan sufiks, yaitu konfiks. Setiap imbuhan memiliki kodenya masing-masing, dan sebuah kata dapat memiliki lebih dari 1 kombinasi imbuhan. Berikut adalah penjelasan dari imbuhan yang telah disebutkan:

1. *Prefiks*

Prefiks atau awalan adalah imbuhan yang ditambahkan pada bagian awal sebuah kata dasar atau bentuk dasar, misalnya ber-, pe-, me- dan lain-lain. Penulisan kode awalan pada file ini akan disimbolkan dengan huruf kapital.

Listing 2.2: Potongan kode untuk prefiks

```

1 | PFX B0 Y 2 # ber-
2 | PFX B0 0 ber  [ ^ r ]
3 | PFX B0 0 be   r

```

Listing 2.2 merupakan salah satu contoh dari awalan yang terdapat pada kamus bahasa Indonesia. Berikut ini adalah penjelasan dari baris kode tersebut:

- PFX
Menandakan bahwa kode tersebut merupakan awalan
- B0
Huruf B menunjukkan awalan yang dimulai dengan huruf B, dan angka 0 menunjukkan bentuk awalan asli

- 2
Menandakan bahwa ada 2 jenis awalan yang dapat digunakan, yaitu ber- dan be-
- -ber
Menandakan bahwa kode tersebut merupakan awalan ber-

2. *Sufiks*

Sufiks atau akhiran adalah imbuhan yang ditambahkan pada bagian belakang kata dasar, misalnya -an, -kan, -i dan lain-lain. Penulisan kode akhiran pada file ini akan disimbolkan dengan huruf kecil.

Listing 2.3: Potongan kode untuk sufiks

```
1 | SFX k0 Y 3 # -kan
2 | SFX k0 0 kan .
3 | SFX k0 0 kanlah
4 | SFX k0 0 kankah
```

Listing 2.3 merupakan salah satu contoh dari akhiran yang terdapat pada kamus bahasa Indonesia. Berikut ini adalah penjelasan dari baris kode tersebut:

- SFX
Menandakan bahwa kode tersebut merupakan akhiran
- k0
Huruf k menunjukkan akhiran yang dimulai dengan huruf k, dan angka 0 menunjukkan bentuk akhiran asli
- 3
Menandakan bahwa ada 3 jenis akhiran yang dapat digunakan, yaitu -kan, -kanlah dan -kankah
- -kan
Menandakan bahwa kode tersebut merupakan akhiran -kan

3. *Konfiks*

Konfiks adalah afiks tunggal yang terjadi dari dua unsur yang terpisah, misalnya ke-...-an dalam kemerdekaan.

Listing 2.4: Potongan kode untuk konfiks

```
1 | PFX KT Y 2 # keter- (keter-an)
2 | PFX KT 0 keter/A1 [^r]
3 | PFX KT 0 kete/A1 r
4 |
5 | SFX Sa Y 1 # -an (se-an) + o0
6 | SFX Sa 0 an/S1o0A1 .
```

Listing 2.4 merupakan salah satu contoh dari konfiks yang terdapat pada kamus bahasa Indonesia. Berikut ini adalah penjelasan dari baris kode tersebut:

- Kolom pertama biasanya digunakan sebagai tanda untuk mengetahui apakah kode tersebut termasuk awalan atau akhiran. Namun pada konfiks tidak ada simbol khusus yang menandakan bahwa kode tersebut adalah sebuah konfiks. Pada baris 1 tertulis "PFX", yang berarti konfiks ini berakar pada awalan tertentu. Pada baris 5 tertulis "SFX", yang berarti konfiks ini berakar pada akhiran tertentu.
- Kolom ke-2 menunjukkan simbol dari awalan atau akhiran yang digunakan. Pada baris 1, konfiks itu berasal dari awalan keter-, dapat dilihat bahwa huruf K berarti "ke" dan huruf T adalah "ter". Pada baris 5, huruf S berarti awalan yang berasal dari huruf S dan huruf a berarti akhiran yang berasal dari huruf a.

-
- Kolom ke-4 menunjukkan berapa banyak jumlah konfiks yang dapat digunakan. Pada baris 1 terdapat 2 jenis, sedangkan pada baris 5 terdapat 1 jenis saja.
 - Kolom ke-5 menunjukkan jenis konfiks yang digunakan. Baris 1-3 merupakan contoh dari konfiks keter-an, sedangkan baris 5-6 merupakan contoh dari konfiks se-an.

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 3.1 dan 3.2:

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.
17-05-2019	Billy Adiwijaya	Pembangkit Timelapse Pe- ngembangan Proyek Per- angkat Lunak	-Kristopher David Harjono M.T. -Elisati Hulu M.T.

Tabel 3.2: Tabel informasi sidang skripsi yang diamati

Tanggal	Mahasiswa	Judul Skripsi	Penguji
20-05-2019	Ihsan Fajari	Sistem Informasi Rekomendasi Pariwisata di Tasikmalaya	-Dra. Rosa de Lima Endang Padmowati, MT -Dr. Ir. Veronica Sri Moertini, MT.
22-05-2019	Muhammad Adrian Putra Zubir	Sistem Informasi Penyediaan Barang Pada Apotek	-Dra. Rosa de Lima Endang Padmowati, MT -Pascal Alfadian Nugroho, S.Kom, M.Comp.
23-05-2019	Ellena Angelica	Kolektor Pengumuman Informatika	-Natalia S.Si, M.Si -Dr. Ir. Veronica Sri Moertini, MT.
24-05-2019	Evelyn Wijaya	Pembangunan Gim Snake 360 Berbasis Web dengan Kode Terbuka	-Chandra Wijaya S.T., M.T. -Raymond Chandra Putra, S.T., M.T.

Pada Tabel 3.1 dan 3.2, telah dijelaskan rincian dari pengamatan sidang skripsi yang telah dilakukan. Dari ke-7 pengamatan tersebut, ditemukan beberapa kesalahan-kesalahan yang terjadi dalam penulisan dokumen skripsi. Kesalahan-kesalahan tersebut didapat pada saat dosen penguji membahas dokumen skripsi milik mahasiswa yang menjalani sidang skripsi. Hasil dari pengamatan tersebut akan dijelaskan pada Tabel 3.3 dan 3.4:

Tabel 3.3: Tabel hasil pengamatan sidang skripsi

Kode	Jenis kesalahan	Keterangan
PS-01	Penulisan kata	Kesalahan dalam penulisan kata merupakan salah satu kesalahan yang sering terjadi. Pada umumnya lebih dikenal dengan istilah <i>typo</i> .
PS-02	Penggunaan imbuhan di- dan kata depan di	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.

Tabel 3.4: Tabel hasil pengamatan sidang skripsi

Kode	Jenis kesalahan	Keterangan
PS-03	Pemberian spasi setelah tanda baca	Salah satu hal kecil yang sering mengganggu adalah penggunaan spasi setelah tanda baca. Tanda baca yang paling sering dipakai, seperti titik, koma, tanya, dan seru harus diberi spasi setelahnya. Spasi juga digunakan sebelum menggunakan tanda kurung buka. Ada beberapa kesalahan yang masih ditemukan seperti, memberi spasi sebelum tanda tanya ataupun memberi spasi sebelum dan setelah garis miring.
PS-04	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.
PS-05	Awal kalimat tidak menggunakan huruf kapital	Setiap huruf pertama pada kata pertama dalam sebuah kalimat harus ditulis dengan huruf kapital.
PS-06	Tidak ada spasi antar kata	Setiap kata dalam sebuah kalimat dipisahkan dengan jarak 1 spasi agar kalimat dapat dibaca dan dimengerti dengan baik.
PS-07	Gambar tidak sesuai dengan 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.
PS-08	Tidak ada keterangan untuk gambar dan tabel	Dalam penulisan dokumen skripsi, setiap gambar dan tabel perlu diberikan keterangan.
PS-09	Jumlah sub bab, sub sub bab tidak boleh hanya 1	Dalam sebuah bab, biasanya jumlah sub bab lebih dari 1. Kesalahan yang sering dilakukan oleh mahasiswa yaitu, hanya terdapat 1 sub bab saja pada 1 bab. Apabila dalam bab tersebut hanya terdapat 1 sub bab, lebih baik tidak perlu dibuat sub bab.

3.1.2 Wawancara Personal

Survei tahap selanjutnya yaitu melakukan dengan melakukan wawancara secara personal. Narasumber dari wawancara ini adalah dosen-dosen Informatika Unpar. Namun tidak semua dosen Informatika diminta untuk menjadi narasumber. Hasil dari wawancara tersebut akan dijelaskan pada Tabel 3.5 hingga tabel 3.7:

Tabel 3.5: Tabel hasil wawancara dosen

Tanggal	Narasumber	Hasil Wawancara	Penjelasan
9-07-2019	Keenan Adiwijaya Leeman S.T.	KAL-01 Cetak miring untuk bahasa asing	Penggunaan kata dalam bahasa asing harus ditulis menggunakan cetak miring. Mahasiswa sering lupa untuk menulis cetak miring bahasa asing.
		KAL-02 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.
		KAL-03 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.
9-07-2019	Chandra Wijaya S.T., M.T.	CHW-01 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.

Tabel 3.6: Tabel hasil wawancara dosen

Tanggal	Narasumber	Hasil Wawancara	Penjelasan
		CHW-02 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.
15-07-2019	Husnul Hakim, S.Kom., M.T.	HUH-01 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.
		HUH-02 Penulisan pseudocode	Dalam penulisan pseudocode hal-hal yang perlu diperhatikan antara lain nama method, masukan serta keluaran pada method dan no baris pada pseudocode.
		HUH-03 Penulisan kata hubung	Kesalahan penggunaan konjungsi akan berakibat tidak jelasnya makna kalimat karena hubungan antar frasa dan antar klausa tidak jelas.
16-07-2019	Vania Natali, S.Kom, M.T.	VAN-01 Tahun skripsi pada cover skripsi	Penulisan tahun skripsi harus sama dengan tahun dimana mahasiswa mengambil skripsi tersebut. Kesalahan yang pernah terjadi, yaitu mahasiswa salah menuliskan tahun skripsi. Meskipun terlihat sepele, namun hal ini perlu diperhatikan.
		VAN-02 Konsistensi penggunaan kata	Mahasiswa harus konsisten dalam penulisan kata, misalnya kata <i>user</i> dan pengguna. Mahasiswa harus memilih antara memakai <i>user</i> atau pengguna.

Tabel 3.7: Tabel hasil wawancara dosen

Tanggal	Narasumber	Hasil Wawancara	Penjelasan
		VAN-02 Penggunaan kata ganti orang	Dalam penulisan dokumen skripsi, tidak boleh ada kata ganti orang. Jika karya non-ilmiah lebih santai karena memakai gaya bahasa non-formal, maka berbeda dengan karya ilmiah. Karya ilmiah memiliki aturan baku dan menggunakan bahasa formal.
16-07-2019	Natalia S.Si, M.Si	NAT-01 Penulisan daftar referensi	Kesalahan yang sering terjadi, yaitu tidak ditemukannya referensi yang akan digunakan. Pada teks yang akan dirujuk, akan terdapat tanda [?], seharusnya tanda tanya tersebut diisi oleh nomor dari referensi.

3.2 Masalah yang Ditangani Pada Perangkat Lunak

Pada bagian ini akan dijelaskan masalah yang ditangani pada perangkat lunak. Setiap hasil survei yang didapatkan melalui pengamatan sidang skripsi dan wawancara dosen, telah diberikan sebuah kode untuk digunakan dalam proses implementasi. Namun, tidak semua dari hasil survei tersebut dapat diimplementasikan menggunakan *regular expression*. Metode yang digunakan untuk mendeteksi kesalahan yaitu dengan *pattern matching*, sehingga hal-hal yang bersifat kontekstual tidak dapat diperiksa dengan menggunakan *regular expression*. Berikut ini adalah hasil keputusan yang telah diambil pada setiap hasil survei di atas:

Tabel 3.8: Tabel keputusan implementasi

Kode	Survei	Keputusan	Alasan
PS-01	Penulisan Kata	Ditangani	Dapat diselesaikan menggunakan <i>regular expression</i>
PS-02	Penggunaan imbuhan di- dan kata depan di-	Tidak ditangani	Tidak dapat diselesaikan menggunakan <i>regular expression</i> , karena pada kamus Indonesia <i>LibreOffice</i> tidak ada fitur untuk membedakan kata sebagai keterangan atau bukan.

Tabel 3.9: Tabel keputusan implementasi

Kode	Survei	Keputusan	Alasan
PS-03	Pemberian spasi setelah tanda baca	Ditangani	Dapat diselesaikan menggunakan <i>regular expression</i>
PS-04	Terdapat ruang kosong yang besar	Tidak ditangani	Tidak dapat diselesaikan menggunakan <i>regular expression</i> , karena hasil ekstrak dari PDF menggunakan <i>PDF Parser</i> tidak mendeteksi adanya baris kosong.
PS-05	Awal kalimat tidak menggunakan huruf kapital	Ditangani	Dapat diselesaikan menggunakan <i>regular expression</i>
PS-06	Tidak ada spasi antar kata	Tidak ditangani	Dapat diselesaikan menggunakan <i>regular expression</i> . Namun survei ini tidak diimplementasi, karena penyelesaiannya sama dengan survei PS-01, sehingga akan disatukan implementasinya.
PS-07	Gambar tidak sesuai tempatnya	Tidak ditangani	Tidak dapat diselesaikan menggunakan <i>regular expression</i> , karena hasil ekstrak PDF dari <i>PDF Parser</i> tidak dapat mendeteksi gambar.
PS-08	Tidak ada keterangan untuk gambar dan tabel	Tidak ditangani	Tidak dapat diselesaikan menggunakan <i>regular expression</i> , karena hasil ekstrak PDF dari <i>PDF Parser</i> tidak dapat mendeteksi gambar dan tabel.
PS-09	Jumlah sub bab, sub sub bab tidak boleh hanya 1	Ditangani	Dapat diselesaikan menggunakan <i>regular expression</i>
KAL-01	Cetak miring untuk bahasa asing	Tidak ditangani	Tidak dapat diselesaikan menggunakan <i>regular expression</i> , karena membutuhkan kamus bahasa Inggris. Selain itu <i>PDF Parser</i> tidak dapat mencocokkan teks yang cetak miring.
KAL-02	Kalimat pengantar untuk setiap subbab	Ditangani	Dapat diselesaikan menggunakan <i>regular expression</i> .
KAL-03	Kelengkapan data skripsi	Ditangani	Dapat diselesaikan menggunakan <i>regular expression</i>

Tabel 3.10: Tabel keputusan implementasi

Kode	Survei	Keputusan	Alasan
CHW-01	Letak keterangan untuk gambar dan tabel	Tidak ditangani	Tidak dapat diselesaikan menggunakan <i>regular expression</i> , karena hasil ekstrak PDF dari <i>PDF Parser</i> tidak dapat mendeteksi gambar dan tabel.
CHW-02	Penggunaan bahasa yang benar	Tidak ditangani	Dapat diselesaikan menggunakan <i>regular expression</i> . Namun survei ini tidak diimplementasi, karena penyelesaiannya sama dengan survei PS-01, sehingga akan disatukan implementasinya.
HUH-01	Rujukan untuk gambar dan tabel	Tidak ditangani	Tidak dapat diselesaikan menggunakan <i>regular expression</i> , karena hasil ekstrak PDF dari <i>PDF Parser</i> tidak dapat mendeteksi gambar dan tabel.
HUH-02	Penulisan pseudocode	Tidak ditangani	Tidak dapat diselesaikan menggunakan <i>regular expression</i> , karena hasil ekstrak PDF dari <i>PDF Parser</i> tidak dapat mendeteksi pseudocode.
HUH-03	Penulisan kata hubung	Tidak ditangani	Tidak dapat diselesaikan menggunakan <i>regular expression</i> , karena <i>regex</i> tidak dapat memeriksa kata hubung yang digunakan sudah tepat atau belum berdasarkan fungsi dari kata hubung tersebut.
VAN-01	Tahun skripsi pada cover skripsi	Tidak ditangani	Dapat diselesaikan menggunakan <i>regular expression</i> . Namun survei ini tidak diimplementasi, karena penyelesaiannya sama dengan survei KAL-03, sehingga akan disatukan implementasinya.
VAN-02	Konsistensi penggunaan kata	Tidak ditangani	Tidak dapat diselesaikan menggunakan <i>regular expression</i> , karena tidak dapat membuat padanan kata untuk memeriksa konsistensi penggunaan kata.

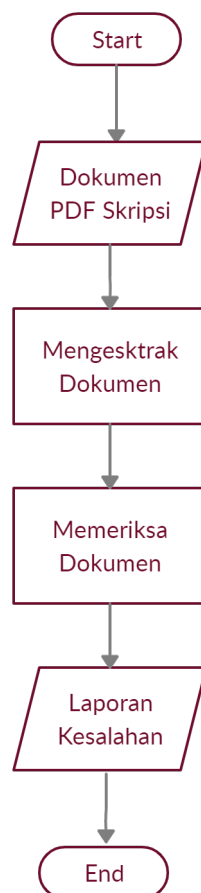
Tabel 3.11: Tabel keputusan implementasi

Kode	Survei	Keputusan	Alasan
VAN-03	Penggunaan kata ganti orang	Ditangani	Dapat diselesaikan menggunakan <i>regular expression</i>
NAT-01	Penulisan daftar referensi	Ditangani	Dapat diselesaikan menggunakan <i>regular expression</i>

Seperti yang sudah dijabarkan pada Tabel 3.8 hingga Tabel 3.11, 8 dari 21 hasil survei akan diimplementasikan menjadi fitur dalam perangkat lunak. Keputusan tersebut diambil berdasarkan dapat / tidaknya kesalahan tersebut diperiksa menggunakan *pattern matching regex* dan tingkat kesulitan untuk memeriksa kesalahan tersebut.

3.3 Analisis Cara Memeriksa Kesalahan Pada Dokumen Skripsi

Pada bagian ini akan dijelaskan proses yang diperlukan untuk dapat memeriksa kesalahan dalam dokumen skripsi hingga menghasilkan laporan kesalahan yang ditemukan. Hal tersebut akan dijelaskan pada *flow chart* berikut ini:



Gambar 3.1: Flow Chart Aplikasi Pemeriksa Kesalahan Dokumen Skripsi

Gambar 3.1 merupakan proses yang dibutuhkan untuk dapat memeriksa dan menghasilkan laporan kesalahan dokumen. Penjelasan dari *flow chart* di atas adalah sebagai berikut:

1. Aplikasi akan menerima masukan berupa dokumen PDF skripsi Informatika UNPAR dengan mode *final*. Aplikasi akan mengambil nama dari dokumen beserta lokasi dari dokumen tersebut.
2. Tahap kedua adalah mengekstrak dokumen PDF skripsi. Pada tahap ini, aplikasi akan menggunakan *library Pdf Parser* untuk mengesktrak dokumen PDF skripsi. Dokumen skripsi akan diekstrak menjadi beberapa bagian, seperti halaman cover, daftar isi, bab, daftar pustaka dan lain-lain. Hasil ekstrak dari setiap bagian-bagian tersebut akan disimpan dalam sebuah string. Jumlah string yang tersedia bergantung dengan jumlah konten yang akan digunakan. Untuk mempermudah proses pemeriksaan, setiap string akan dipecah menjadi kalimat-kalimat yang disimpan dalam struktur data *Array* 1 dimensi.
3. Tahap ketiga adalah memeriksa kesalahan pada dokumen skripsi. Pada skripsi ini, fitur kesalahan yang telah diseleksi berjumlah 8 buah. Setiap fitur akan memiliki tugasnya masing-masing dalam mencari kesalahan. Setiap fitur pemeriksa akan memeriksa kesalahan menggunakan pencocokan pola dengan *regular expression*. Pemeriksa akan mencocokkan pola dengan kalimat-kalimat yang sudah disimpan *array*. Misalnya, fitur pemeriksa kelengkapan data skripsi akan memeriksa array yang berisi halaman cover saja.
4. Tahap terakhir adalah mengeluarkan laporan kesalahan pada dokumen skripsi. Pada tahap ini, laporan kesalahan akan ditampilkan secara terurut dan sekaligus setelah aplikasi dijalankan. Format laporan yang dikeluarkan meliputi kode kesalahan, jenis kesalahan dan bagian yang terdapat kesalahan. Laporan kesalahan akan dikeluarkan pada terminal.

BAB 4

ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

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

4.1 Perancangan Metode Pemeriksaan

Pada bagian ini akan dijelaskan rancangan metode yang akan digunakan oleh setiap fitur pemeriksaan. Pemeriksaan kesalahan dokumen skripsi akan menggunakan metode pencocokan pola. Pola tersebut akan dibuat dengan menggunakan *regular expression*. Setiap fitur memiliki pola yang berbeda, sesuai dengan fungsi yang dimiliki oleh masing-masing fitur. Berikut ini akan dijelaskan pola-pola yang akan digunakan:

- Fitur PS-01
Pada fitur ini, pemeriksaan dilakukan dengan cara membandingkan kata-kata yang ada pada dokumen dengan kata-kata yang ada pada kamus. Kode imbuhan yang ada pada kamus dihilangkan dengan menggunakan *regular expression*. Hal ini dilakukan karena pemeriksaan kata mengabaikan imbuhan yang ada pada kamus.
- Fitur PS-03
Pada fitur ini, pemeriksaan dilakukan dengan cara mencari pola kalimat yang penulisan tanda bacanya berhimpitan antara 2 kata. Selain itu tanda baca titik pada akhir kalimat yang berhimpitan dengan kata pertama pada kalimat baru. Pola *regular expression* yang digunakan akan diuraikan pada Listing 4.1.

Listing 4.1: Pola *regular expression* fitur PS-03

1 | / ([A-Za-z]{2},\$){1}[A-Za-z]+) /

- Fitur PS-05
Pada fitur ini, pemeriksaan dilakukan dengan cara mencari pola kalimat yang karakter pertamanya menggunakan huruf kecil. Pola *regular expression* yang digunakan akan diuraikan pada Listing 4.2.

Listing 4.2: Pola *regular expression* fitur PS-05

1 | / \hat{ }\$ [a-z]+ | [a-z]+\backslash\$.[a-z] /

- Fitur PS-09
Pada fitur ini, pemeriksaan dilakukan dengan cara melihat halaman daftar isi. Halaman ini harus dibersihkan dulu dari karakter yang menggunakan huruf, untuk mendapatkan nomor bab, sub bab maupun sub sub bab yang ada pada dokumen skripsi. Nomor tersebut diurutkan dari bab paling awal hingga terakhir. Kesalahan akan ditemukan apabila jumlah sub bab dalam bab atau jumlah sub sub bab dalam sub bab hanya ada satu.

- Fitur KAL-02

Pada fitur ini, pemeriksaan dilakukan dengan cara mencari pola bab beserta judulnya, misal BAB 1 PENDAHULUAN. Pola selanjutnya adalah mencocokkan karakter setelah judul dari bab tersebut. Apabila setelah judul bab, karakter yang ditemukan adalah karakter angka, berarti pada bab tersebut tidak diberikan kata pengantar. Pola *regular expression* yang digunakan akan diuraikan pada Listing 4.3.

Listing 4.3: Pola *regular expression* fitur KAL-02

```
1 | / $ \hat{ } $ (BAB ) [0-9] $ \lbrace $ 1 $ \rbrace $ [ $ \backslash $ s A - Z ] $ \lbrace $ $ \lbrace $ 1 , [0-9] + /
```

- Fitur KAL-03

Pada fitur ini, pemeriksaan dilakukan dengan cara mencari pola template yang diberikan pada halaman cover bahasa Indonesia dan bahasa Inggris. Pola dibuat dengan memasukan kata-kata template yang ada pada file data.tex. Pola *regular expression* yang digunakan akan diuraikan pada Listing 4.4.

Listing 4.4: Pola *regular expression* fitur KAL-03

```
1 | / $ \backslash $ b JUDUL BAHASA INDONESIA $ \backslash $ b | $ \backslash $ b JUDUL BAHASA INGGRIS $ \backslash $ b | $ \backslash $ b Nama Lengkap $ \backslash $ b | $ \backslash $ b 10 digit NPM UNPAR $ \backslash $ b | $ \backslash $ b tahun $ \backslash $ b /
```

- Fitur NAT-01

Pada fitur ini, pemeriksaan dilakukan dengan mencari pola "[?]" pada dokumen skripsi. Pola tersebut menandakan bahwa referensi yang dimasukan pada referensi.bib belum di-compile dengan benar. Pola *regular expression* yang digunakan akan diuraikan pada Listing 4.5.

Listing 4.5: Pola *regular expression* fitur NAT-01

```
1 | / $ \backslash $ B $ \backslash $ $ [ $ \backslash $ $ ? $ \backslash $ $ ] $ \backslash $ B /
```

- Fitur VAN-03

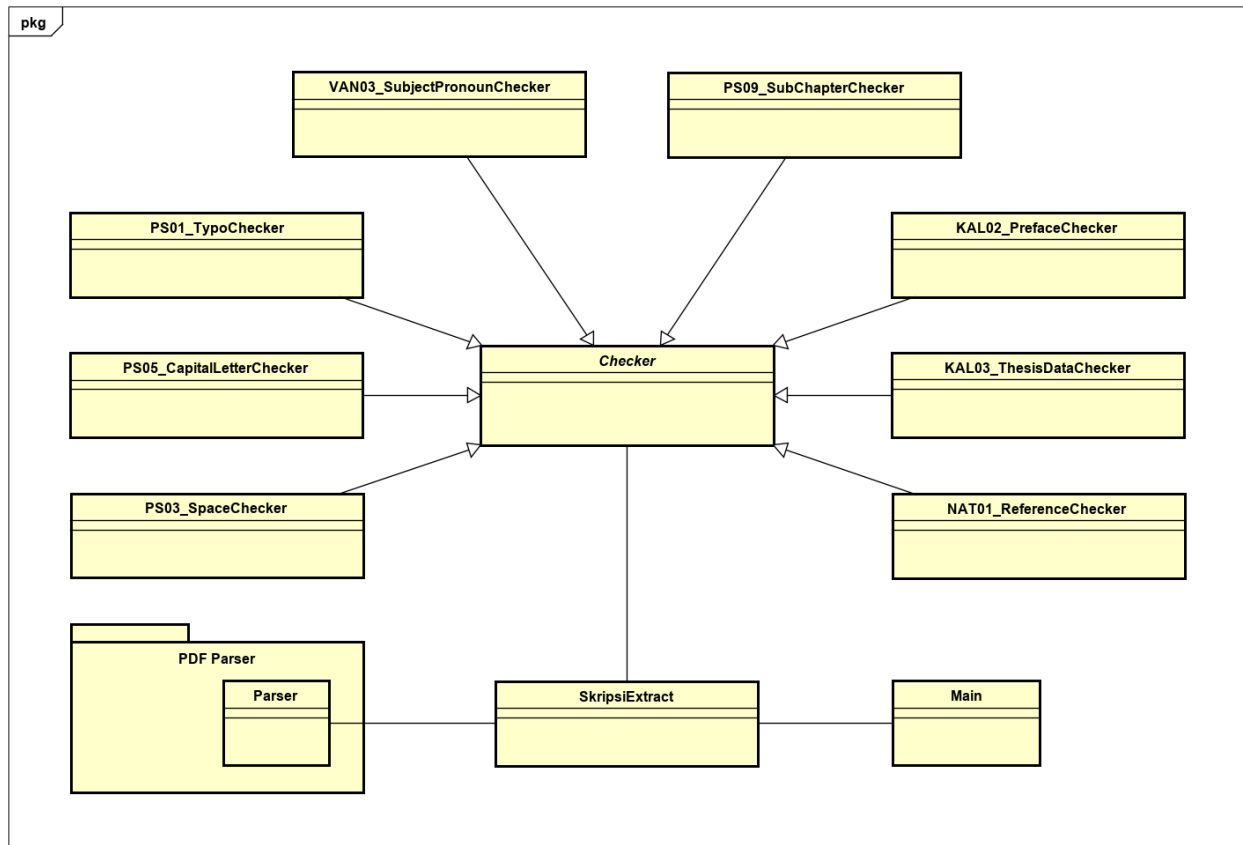
Pada fitur ini, pemeriksaan dilakukan dengan mencari pola kata ganti orang yang ada pada dokumen skripsi. Pola *regular expression* yang digunakan akan diuraikan pada Listing 4.6.

Listing 4.6: Pola *regular expression* fitur VAN-03

```
1 | / $ \backslash $ bsaya $ \backslash $ b | $ \backslash $ bkamu $ \backslash $ b | $ \backslash $ bdia $ \backslash $ b / i /
```

4.2 Perancangan Perangkat Lunak

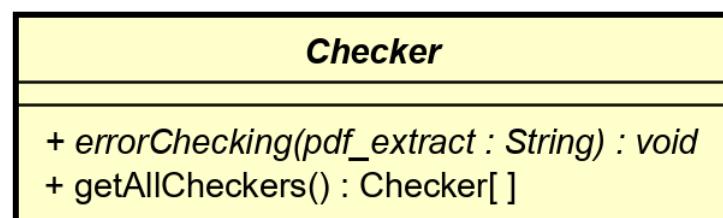
Sebelum implementasi pada perangkat lunak dilakukan, langkah yang dilakukan adalah merancang diagram kelas. Perancangan diagram kelas bertujuan untuk menentukan objek-objek yang terlibat dalam proses memeriksa kesalahan dokumen skripsi. Hubungan antar objek juga didefinisikan dalam diagram kelas. Perangkat lunak akan menggunakan paradigma *Object Oriented Programming* dalam pengembangannya. Rancangan kelas tersebut akan ditunjukkan oleh diagram kelas di bawah ini:



Gambar 4.1: Diagram kelas yang disederhanakan

Gambar 4.1 merupakan diagram kelas aplikasi pemeriksa kesalahan dokumen skripsi Informatika UNPAR yang telah disederhanakan. Pada diagram kelas tersebut, ditunjukkan bahwa perangkat lunak memiliki sebelas kelas dan sebuah package *library PDF Parser*. Terdapat sebuah kelas abstrak yang menjadi kelas *parent* dari ke-8 fitur yang akan diimplementasikan dalam perangkat lunak. Delapan fitur yang ada akan memiliki method yang sama, namun berbeda proses pemeriksaanya. Rincian dari setiap kelas tersebut akan dijelaskan sebagai berikut.

1. Kelas Checker

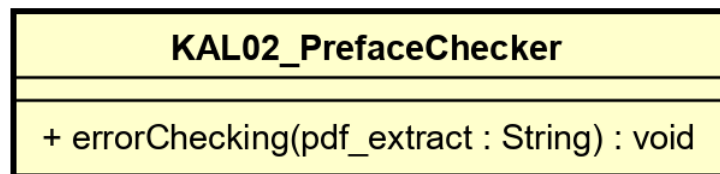


Gambar 4.2: Kelas Checker

Kelas ini merupakan kelas *Parent* dari semua *checker* yang akan diimplementasi pada perangkat lunak. Kelas ini memiliki sebuah *method* abstrak dan sebuah *method* *getter*, seperti yang ditunjukkan pada Gambar 4.2. Semua anak kelas *Checker* akan mengadopsi method yang ada pada kelas ini. Berikut ini adalah *method* yang terdapat pada kelas *Checker*.

- `errorChecking($pdf_extract)`
Method ini merupakan method abstrak, yang akan diturunkan kepada seluruh anak kelasnya. Method ini berfungsi untuk memeriksa kesalahan pada dokumen skripsi sesuai dengan peran yang diberikan pada kelas tersebut. Method ini menerima masukan `pdf_extract` dengan tipe data kelas *SkripsiExtract*. Parameter tersebut dapat digunakan oleh masing-masing kelas *Checker* untuk memanggil *method getter* yang diperlukan. Tidak semua pemeriksa memerlukan seluruh isi halaman dari dokumen skripsi. Sehingga proses pemeriksaan menjadi lebih efektif.
- `getAllChecker()`
Method ini berfungsi untuk melakukan instansiasi seluruh anak kelas *Checker*. Method ini mengembalikan hasil instansiasi dari anak kelas *Checker*. Hal ini akan mempermudah saat akan memanggil *method errorChecking* pada setiap *Checker* yang diimplementasi.

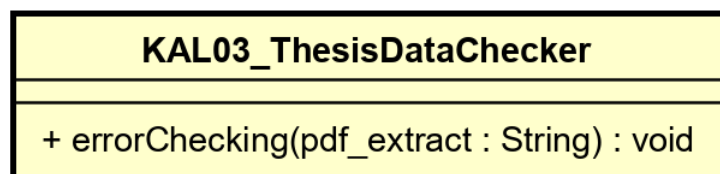
2. Kelas KAL02_PrefaceChecker



Gambar 4.3: Kelas KAL02_PrefaceChecker

Kelas ini bertanggungjawab untuk memeriksa ada atau tidaknya kata pengantar sebelum memulai bab. Kelas ini memiliki sebuah method, yaitu `errorChecking($pdf_extract)`, seperti yang ditunjukkan pada Gambar 4.3. Method ini berfungsi untuk memeriksa kesalahan pada dokumen skripsi. Method ini menerima masukan `pdf_extract` dengan tipe data kelas *SkripsiExtract*. Bagian yang diperiksa pada *checker* ini hanya bab 1 hingga 6.

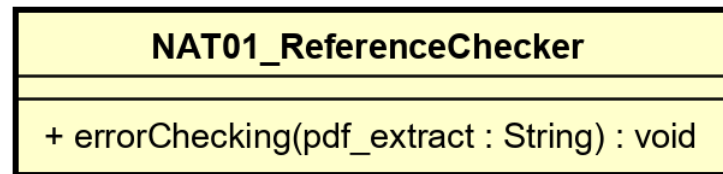
3. Kelas KAL03_ThesisDataChecker



Gambar 4.4: Kelas KAL03_ThesisDataChecker

Kelas ini bertanggungjawab untuk memeriksa kelengkapan data skripsi yang ditulis dalam bahasa Indonesia maupun bahasa Inggris. Dokumen skripsi akan menampilkan tulisan dari *template* dokumen skripsi, apabila data skripsi belum diisi. Kelas ini memiliki sebuah *method*, yaitu `errorChecking($pdf_extract)`. *Method* ini berfungsi untuk memeriksa kesalahan pada dokumen skripsi. Method ini menerima masukan `pdf_extract` dengan tipe data kelas *SkripsiExtract*. Bagian yang diperiksa pada *checker* ini adalah halaman cover bahasa Indonesia dan bahasa Inggris.

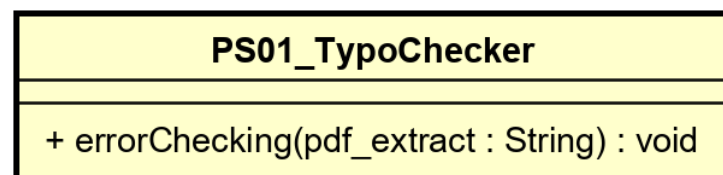
4. Kelas NAT01_ReferenceChecker



Gambar 4.5: Kelas NAT01_ReferenceChecker

Kelas ini bertanggungjawab untuk memeriksa referensi yang akan dirujuk dalam dokumen. kelas ini memiliki sebuah *method*, yaitu `errorChecking($pdf_extract)`, seperti yang ditunjukkan pada Gambar 4.5. *Method* ini berfungsi untuk memeriksa kesalahan pada dokumen skripsi. Method ini menerima masukan `pdf_extract` dengan tipe data kelas *SkripsiExtract*. Bagian yang diperiksa pada *checker* ini adalah bab 1 hingga 6.

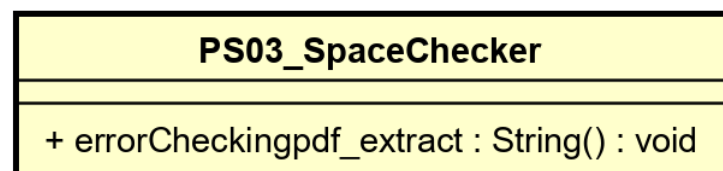
5. Kelas PS01_TypoChecker



Gambar 4.6: Kelas PS01_TypoChecker

Kelas ini bertanggungjawab untuk memeriksa kesalahan penulisan kata. Kelas ini memiliki sebuah *method*, yaitu `errorChecking($pdf_extract)`, seperti yang ditunjukkan pada Gambar 4.6. *Method* ini berfungsi untuk memeriksa kesalahan pada dokumen skripsi. Method ini menerima masukan `pdf_extract` dengan tipe data kelas *SkripsiExtract*. Bagian yang diperiksa pada *checker* ini adalah bab 1 hingga 6.

6. Kelas PS03_SpaceChecker

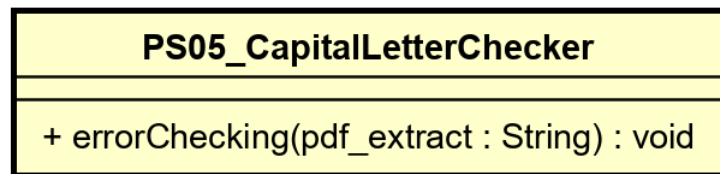


Gambar 4.7: Kelas PS03_SpaceChecker

Kelas ini bertanggungjawab untuk memeriksa penggunaan spasi sebelum dan setelah tanda baca. Kelas ini memiliki sebuah *method*, yaitu `errorChecking($pdf_extract)`, seperti yang ditunjukkan pada Gambar 4.7. *Method* ini berfungsi untuk memeriksa kesalahan pada dokumen

skripsi. Method ini menerima masukan pdf_extract dengan tipe data kelas *SkripsiExtract*. Bagian yang diperiksa pada *checker* ini adalah bab 1 hingga 6.

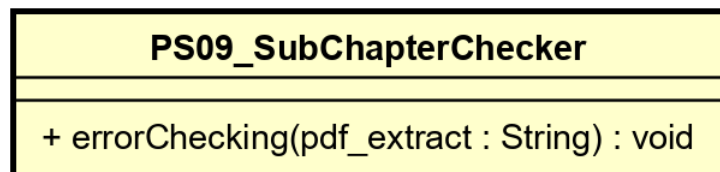
7. Kelas PS05_CapitalLetterChecker



Gambar 4.8: Kelas PS05_CapitalLetterChecker

Kelas ini bertanggungjawab untuk memeriksa penggunaan huruf kapital pada awal kalimat. Kelas ini memiliki sebuah method, yaitu errorChecking(\$pdf_extract), seperti yang ditunjukkan pada Gambar 4.8. Method ini berfungsi untuk memeriksa kesalahan pada dokumen skripsi. Method ini menerima masukan pdf_extract dengan tipe data kelas *SkripsiExtract*. Bagian yang diperiksa pada *checker* ini adalah bab 1 hingga 6.

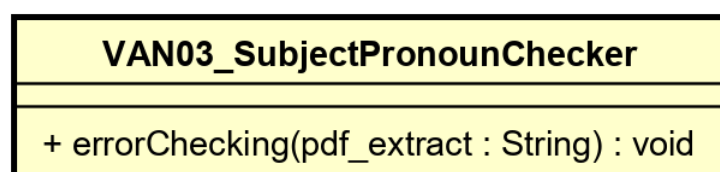
8. Kelas PS09_SubChapterChecker



Gambar 4.9: Kelas PS09_SubChapterChecker

Kelas ini bertanggungjawab untuk memeriksa jumlah sub bab atau sub sub bab yang ada dalam sebuah bab atau sub bab. Kelas ini memiliki sebuah method, yaitu errorChecking(\$pdf_extract), seperti yang ditunjukkan pada Gambar 4.9. Method ini berfungsi untuk memeriksa kesalahan pada dokumen skripsi. Method ini menerima masukan pdf_extract dengan tipe data kelas *SkripsiExtract*. Bagian yang diperiksa pada *checker* ini adalah halaman daftar isi.

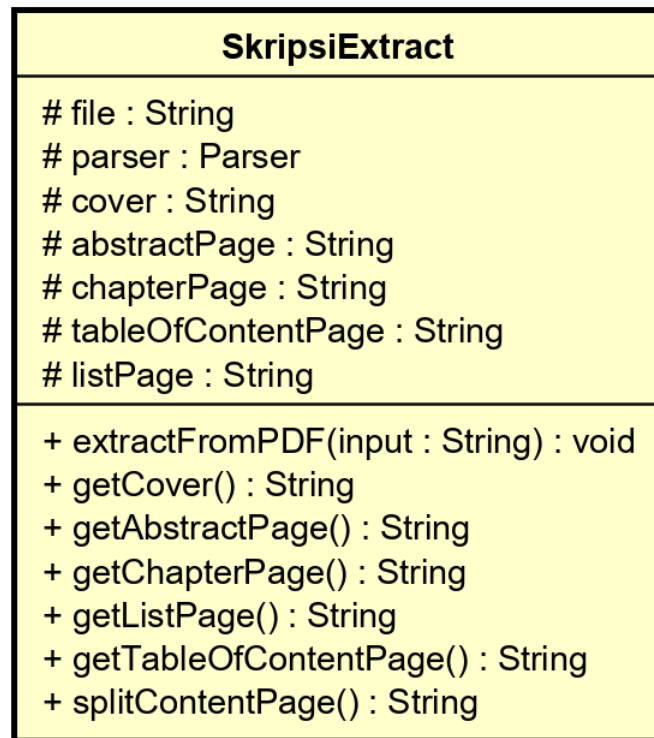
9. Kelas VAN03_SubjectProunounChecker



Gambar 4.10: Kelas VAN03_SubjectProunounChecker

Kelas ini bertanggungjawab untuk memeriksa penggunaan kata ganti orang pada dokumen skripsi. Kelas ini memiliki sebuah method, yaitu `errorChecking($pdf_extract)`, seperti yang ditunjukkan pada Gambar 4.10. *Method* ini berfungsi untuk memeriksa kesalahan pada dokumen skripsi. Method ini menerima masukan `pdf_extract` dengan tipe data kelas *SkripsiExtract*. Bagian yang diperiksa pada *checker* ini adalah bab 1 hingga 6.

10. Kelas SkripsiExtract



Gambar 4.11: Kelas SkripsiExtract

Kelas ini berfungsi untuk melakukan proses ekstrak dokumen PDF skripsi, dengan menggunakan *library PdfParser*. Hasil dari ekstrak tersebut akan disimpan menjadi beberapa bagian, berdasarkan konten-konten yang ada pada dokumen skripsi. Kelas ini memiliki beberapa atribut yang akan dijelaskan sebagai berikut:

- file
Atribut ini berfungsi untuk menyimpan lokasi dokumen skripsi beserta nama dari dokumen tersebut.
- parser
Atribut ini berfungsi sebagai instansiasi kelas Parser dari *Library PDF Parser*, untuk memanggil method yang dibutuhkan dalam proses ekstrak dokumen.
- cover
Atribut ini berfungsi untuk menyimpan hasil ekstrak dari halaman cover bahasa Indonesia dan bahasa Inggris.
- abstractPage
Atribut ini berfungsi untuk menyimpan hasil ekstrak dari halaman abstrak bahasa Indonesia dan bahasa Inggris.

- `otherPage`
Atribut ini berfungsi untuk menyimpan hasil ekstrak dari halaman lembar pengesahan, pernyataan dan kata pengantar.
- `tableOfContentPage`
Atribut ini berfungsi untuk menyimpan hasil ekstrak dari halaman daftar isi.
- `contentPage`
Atribut ini berfungsi untuk menyimpan hasil ekstrak mulai dari bab 1 hingga bab 6.
- `listPage`
Atribut ini berfungsi untuk menyimpan hasil ekstrak pada halaman daftar tabel, daftar gambar, daftar referensi dan lampiran.

Kelas ini juga memiliki *method* yang berkaitan dengan proses ekstrak dokumen skripsi dan penyimpanan hasil ekstraknya. Berikut adalah *method* yang terdapat pada kelas ini:

- `extractFromPDF($input)`
Method ini berfungsi untuk melakukan proses ekstrak file PDF skripsi. Hasil dari ekstrak tersebut akan disimpan ke dalam beberapa bagian, seperti cover, halaman abstrak, dan sebagainya.
- `getCoverPage()`
Method ini berfungsi untuk mendapatkan halaman cover skripsi dalam bahasa Indonesia dan bahasa Inggris.
- `getAbstractPage()`
Method ini berfungsi untuk mendapatkan halaman abstrak dalam bahasa Indonesia dan bahasa Inggris.
- `getOtherPage()`
Method ini berfungsi untuk mendapatkan halaman lembar pengesahan, pernyataan dan kata pengantar.
- `getTableOfContentPage()`
Method ini berfungsi untuk mendapatkan halaman daftar isi. Method ini akan menyaring sub bab dan sub sub bab yang terdapat pada daftar isi.
- `getListPage()`
Method ini berfungsi untuk mendapatkan halaman daftar gambar, daftar tabel dan daftar referensi.
- `getContentPage()`
Method ini berfungsi untuk mendapatkan halaman konten skripsi dari bab 1 sampai 6.
- `splitContentPage()`
Method ini memiliki fungsi yang hampir sama dengan method *getContentPage*, yaitu untuk mendapatkan halaman konten skripsi. Namun, pada method ini ada beberapa proses pemisahan isi konten (per kalimat) ke dalam beberapa array. Fitur-fitur dalam perangkat lunak yang memerlukan pemeriksaan berdasarkan kalimat dapat langsung memanggil method ini.

11. Kelas Main

Kelas ini digunakan untuk menjalankan perangkat lunak. Kelas ini tidak memiliki atribut dan *method*, hanya terdiri dari beberapa baris kode yang memanggil *method* untuk mengekstrak dan memeriksa file PDF skripsi.

4.3 Perancangan Algoritma

Pada bagian ini akan dijelaskan perancangan algoritma untuk mengekstrak dokumen skripsi dan pola pengecekan kesalahan yang akan digunakan pada perangkat lunak.

4.3.1 Algoritma untuk Mengekstrak Dokumen

Dokumen skripsi yang dapat diperiksa dalam perangkat lunak merupakan dokumen skripsi Teknik Informatika Unpar, yang memiliki ekstensi *PDF*. Sebelum dilakukan pemeriksaan kesalahan, dokumen tersebut perlu diekstrak terlebih dulu. Pada skripsi ini, aplikasi akan menggunakan *library PDFParser* untuk mengekstrak dokumen skripsi. Pada *library* tersebut, terdapat 2 metode yang dapat digunakan untuk mengekstrak dokumen. Penjelasan dari ke-2 metode tersebut akan diuraikan pada sebagai berikut.

Listing 4.7: Potongan kode untuk mengesktrak seluruh halaman dokumen

```
1 | include 'vendor/autoload.php';
2 |
3 | $parser = new \Smalot\PdfParser\Parser();
4 | $pdf    = $parser->parseFile('document.pdf');
5 | $text   = $pdf->getText();
```

Listing 4.8: Potongan kode untuk mengesktrak halaman dokumen secara spesifik

```
1 | include 'vendor/autoload.php';
2 |
3 | $parser = new \Smalot\PdfParser\Parser();
4 | $pdf    = $parser->parseFile('document.pdf');
5 | $pages  = $pdf->getPages();
```

Listing 4.7 dan Listing 4.8 merupakan potongan kode yang terdapat pada dokumentasi *library PDF Parser*, yang digunakan untuk mengekstrak isi dokumen. Kode dari kedua listing tersebut hampir mirip, yang menjadi pembeda adalah pada baris ke-5. Listing 4.7 menggunakan *method getText()* dan hasil dari ekstraknya disimpan menjadi sebuah kesatuan dari halaman awal hingga halaman akhir. Listing 4.8 menggunakan *method getPages()* dan hasil dari ekstraknya disimpan dalam sebuah array String sejumlah halaman yang ada pada dokumen. Pengguna dapat menentukan secara spesifik halaman apa saja yang ingin diekstrak.

Pada skripsi ini akan digunakan metode yang mengesktrak halaman-halaman secara spesifik. Hasil ekstrak akan disimpan dalam berdasarkan konten-konten yang ada, seperti halaman cover, abstrak, daftar isi, daftar tabel, daftar gambar dan seterusnya. Ada beberapa alasan yang mendorong untuk melakukan pemisahan konten-konten tersebut, salah satunya yaitu agar pemeriksa dapat memeriksa bagian-bagian spesifik yang dibutuhkan. Contohnya, untuk memeriksa kelengkapan data skripsi tidak perlu memeriksa seluruh isi dari dokumen, halaman yang dibutuhkan hanya cover saja. Dengan demikian, proses pemeriksaan menjadi lebih efektif karena tidak perlu memeriksa seluruh halaman yang ada pada dokumen.

4.3.2 Algoritma Pemeriksa Kesalahan

Hasil survei kesalahan-kesalahan dalam penulisan dokumen skripsi sudah disaring dan akan diimplementasikan dalam perangkat lunak. Pengecekan kesalahan tersebut akan dilakukan dengan menggunakan teknik *pattern matching*. Hasil ekstrak dari PDF skripsi akan dipotong per kalimat dengan *delimiter* karakter titik dan spasi ke dalam sebuah array. Berikut adalah rincian dari hasil survei yang dipilih beserta penyelesaiannya:

1. Penulisan kata (PS-01)

Untuk mendeteksi kesalahan penulisan kata, akan digunakan ekstensi kamus bahasa Indonesia *LibreOffice*. Dengan digunakannya kamus tersebut, kesalahan penulisan suatu kata dapat diminimalisir. Pada skripsi ini, pemeriksaan kata-kata yang menggunakan imbuhan tidak ditangani pada masalah ini, karena keterbatasan waktu yang dimiliki. Terdapat kode yang harus diterjemahkan terlebih dahulu untuk dapat mengetahui imbuhan yang dapat digunakan dalam sebuah kata, seperti yang sudah dijelaskan pada sub bab 2.3. Pola pengecekan yang digunakan akan dijelaskan pada Pseudocode 1.

Algorithm 1 Typo checker function

```

1: Input: An object from the SkripsiExtract class to call the getter method
2: Output: An array containing error reports
3:
4: function ERRORCHECKING($pdf_extract)
5:   temp  $\leftarrow$  Split dictionary based on newline
6:   dictionary  $\leftarrow$  Array for store dictionary
7:   for i = 0 to size of temp do
8:     Remove affix code from each index dictionary array
9:   word  $\leftarrow$  Call getter from class SkripsiExtract and split with non word character
10:  typos  $\leftarrow$  Array for store typos
11:  for each word in array do
12:    if value not contain in dictionary and value not contain in typos then
13:      Fill typos with value
14:  result  $\leftarrow$  Array for store all errors report
15:  if size of typos greater than 0 then
16:    result  $\leftarrow$  Add all errors report into result
17:  return result

```

Pada Pseudocode 1, pola akan memeriksa setiap kata yang ada pada indeks array. Namun untuk pemeriksaan kata ini hanya untuk kata yang menggunakan bahasa Indonesia saja. Pola akan mencocokkan kata yang akan diperiksa dengan kamus bahasa Indonesia *LibreOffice*. Berikut adalah contoh dari kesalahan dan laporan yang dikeluarkan:

- Contoh kesalahan
Istilah *regex* berasal dari teori matematika dan komputer sains, yang mencerminkan sifat ekspresi dalam matematika yang disebut keteraturan.
- Laporan kesalahan
Pada fitur ini, setiap menemukan kesalahan penulisan kata akan ditampung terlebih dahulu sampai seluruh kata selesai diperiksa. Pada akhirnya seluruh kesalahan yang ditemukan akan dikeluarkan.

2. Pemberian spasi sesudah tanda baca (PS-03)

Kesalahan ini akan terdeteksi apabila tidak ada karakter spasi sesudah tanda baca. Pola pengecekan yang digunakan akan dijelaskan pada *pseudocode 2*.

Algorithm 2 Space checker function

```

1: Input: An object from the SkripsiExtract class to call the getter method
2: Output: An array containing error reports
3:
4: function ERRORCHECKING($pdf_extract)
5:   result  $\leftarrow$  Array for store all errors report
6:   sentence  $\leftarrow$  Call getter from class SkripsiExtract
7:   for each sentence in sentence do
8:     pattern  $\leftarrow$  Define regex /([A-Za-z]{2,}[.,!][A-Za-z]+)/
9:     if pattern contain in array then
10:      result  $\leftarrow$  Fill with errors report
11:  return result

```

Pada Pseudocode 2, pola akan memeriksa setiap kalimat yang ada pada indeks array. Pola *regex* yang telah didefinisikan akan mencari pola yang apabila setelah tanda baca titik, koma, seru atau tanya tidak terdapat spasi yang memisahkan tanda baca dengan kata setelah tanda baca. Berikut adalah contoh dari kesalahan dan laporan yang dikeluarkan:

- Contoh kesalahan
Aplikasi ini dijalankan melalui terminal command Windows. laporan dari hasil kesalahan tersebut akan ditampilkan melalui terminal command Windows.
- Laporan kesalahan
Perhatikan spasi setelah tanda baca.

3. Awal kalimat tidak menggunakan huruf kapital (PS-05)

Kesalahan ini akan terdeteksi apabila setelah tanda baca pada akhir kalimat, karakter pertama setelah spasi menggunakan huruf kecil. Pola pengecekan yang digunakan akan dijelaskan pada *pseudocode* 3.

Algorithm 3 Capital letter checker function

```

1: Input: An object from the SkripsiExtract class to call the getter method
2: Output: An array containing error reports
3: function ERRORCHECKING($pdf_extract)
4:   result ← Array for store all errors report
5:   sentence ← Call getter from class SkripsiExtract
6:
7:   for each sentence in sentence do
8:     pattern ← Define regex /^[a-z]+ | [a-z]+\.[a-z] /
9:     if pattern contain in array then
10:      if first character in a sentence is a non word character then
11:        Continue
12:      else
13:        result ← Fill with errors report
14:   return result

```

Pada Pseudocode 3, pola akan memeriksa setiap kalimat yang ada pada indeks array. Pola *regex* yang telah didefinisikan akan mencari pola dimana karakter pertama dalam sebuah kalimat menggunakan huruf kecil. Namun perlu dipastikan bahwa karakter pertama tersebut adalah karakter *word*. Berikut adalah contoh dari kesalahan dan laporan yang dikeluarkan:

- Contoh kesalahan
aplikasi sederhana ini, dapat dimanfaatkan oleh mahasiswa Informatika Unpar secara mandiri.
- Laporan kesalahan
Huruf pertama pada kalimat ini tidak menggunakan huruf kapital

4. Jumlah subbab dalam 1 bab tidak boleh hanya 1 (PS-09)

Kesalahan ini dapat diselesaikan dengan mencari jumlah subbab yang ada dalam sebuah bab. Pola pengecekan yang digunakan akan dijelaskan pada *pseudocode* 4.

Algorithm 4 Subchapter checker function

```

1: Input: An object from the SkripsiExtract class to call the getter method
2: Output: An array containing error reports
3:
4: function ERRORCHECKING($pdf_extract)
5:   result  $\leftarrow$  Array for store all errors report
6:   sentence  $\leftarrow$  Call getter from class SkripsiExtract
7:   array  $\leftarrow$  Merge an empty array with temp
8:   for i  $\leftarrow$  0 to size of array do if(array[i] contains ".1")
9:     substring  $\leftarrow$  get substring element from index i + 1 (substring from 0 to length of
       array[i] - 1)
10:    if array not contain substring append with 2 then
11:      fill result with errors report
12:  return result

```

Pada Pseudocode 4, pola akan memeriksa setiap kalimat yang ada pada indeks array. Kesalahan akan terdeteksi apabila karakter pertama dalam sebuah kalimat menggunakan huruf kecil. Berikut adalah contoh dari kesalahan dan laporan yang dikeluarkan:

- Contoh kesalahan
 Bab 4
 PERANCANGAN

 4.1 Perancangan Kelas
- Laporan kesalahan
 Bab/Sub bab ini hanya terdapat 1 sub bab/sub sub bab.

5. Kalimat pengantar untuk setiap subbab (KAL-02)

Kesalahan ini dapat dideteksi dengan melihat ada atau tidaknya kalimat setelah subbab dibuat. Pola pengecekan yang digunakan akan dijelaskan pada *pseudocode 5*.

Algorithm 5 Preface checker function

```

1: Input: An object from the SkripsiExtract class to call the getter method
2: Output: An array containing error reports
3:
4: function ERRORCHECKING($pdf_extract)
5:   result  $\leftarrow$  Array for store all errors report
6:   sentence  $\leftarrow$  Call getter from class SkripsiExtract
7:   for each sentence in sentence do
8:     pattern  $\leftarrow$  Define regex / ^ (BAB ) [0-9] {1} [ \s A-Z ] { } 1, [0-9] + /
9:     if pattern contain in array then
10:      fill result with errors report
11:  return result

```

Pada pseudocode 5, pola akan memeriksa setiap kalimat yang ada pada indeks array. Kesalahan akan terdeteksi apabila tidak terdapat kalimat yang mengawali bab tersebut. Berikut adalah contoh dari kesalahan dan laporan yang dikeluarkan:

- Contoh kesalahan
 Bab 4

PERANCANGAN

4.1 Perancangan Kelas

4.2 Perancangan Algoritma

- Laporan kesalahan
Berilah kata pengantar untuk setiap bab.

6. Kelengkapan data skripsi (KAL-03)

Kesalahan ini dapat terlihat pada halaman cover skripsi. Pada halaman cover bahasa Inggris dan bahasa Indonesia akan ditemukan tulisan *template*, apabila mahasiswa belum mengisi data skripsi. Data tersebut diisi pada file "Data.tex". Pola pengecekan yang digunakan akan dijelaskan pada *pseudocode* 6.

Algorithm 6 Thesis data checker function

```

1: Input: An object from the SkripsiExtract class to call the getter method
2: Output: An array containing error reports
3:
4: function ERRORCHECKING($pdf_extract)
5:   result ← Array for store all errors report
6:   sentence ← Call getter from class SkripsiExtract
7:   for each sentence in sentence do
8:     pattern ← Define regex /\bJUDUL BAHASA INDONESIA\b|\bJUDUL BAHASA
      INGGRIS\b|\bNama Lengkap\b|\b10 digit NPM UNPAR\b|\btahun\b/
9:     if pattern contain in array then
10:       fill result with errors report
11:   return result

```

Pada Pseudocode 6, pola akan memeriksa halaman cover bahasa Indonesia dan bahasa Inggris. Kesalahan akan terdeteksi pada halaman cover masih terdapat kalimat yang terdapat dalam template. Berikut adalah contoh dari kesalahan dan laporan yang dikeluarkan:

- Contoh kesalahan
«SKRIPSI/TUGAS AKHIR»
«Judul Bahasa Indonesia»
Marcell Trixie Alexander
«10 digit NPM UNPAR»
- Laporan kesalahan
Ada data skripsi yang belum dilengkapi pada halaman cover.

7. Penggunaan kata ganti orang (VAN-03)

Kesalahan ini dapat diatasi dengan memasukan kata-kata yang termasuk dalam kata ganti orang menjadi kata-kata yang tidak dapat digunakan. Pola pengecekan yang digunakan akan dijelaskan pada *pseudocode* 7.

Algorithm 7 Subject Pronoun checker function

```

1: Input: An object from the SkripsiExtract class to call the getter method
2: Output: An array containing error reports
3:
4: function ERRORCHECKING($pdf_extract)
5:   result  $\leftarrow$  Array for store all errors report
6:   sentence  $\leftarrow$  Split PDF parsing results based on dots and space
7:   for each sentence in sentence do
8:     pattern  $\leftarrow$  Define regex /\bsaya\b | \bkamu\b | \bdia\b/i/
9:     if pattern contain in array then
10:       fill result with errors report
11:   return result

```

Pada Pseudocode 7, pola akan memeriksa setiap kalimat yang ada pada indeks array. Kesalahan akan terdeteksi pada kalimat terdapat kata ganti orang. Berikut adalah contoh dari kesalahan dan laporan yang dikeluarkan:

- Contoh kesalahan
Saya akan membuat aplikasi ini menggunakan bahasa pemrograman PHP.
- Laporan kesalahan
Kalimat ini mengandung kata ganti orang.

8. Penulisan daftar referensi (NAT-01)

Kesalahan dalam penulisan daftar referensi dapat dilihat dengan munculnya tanda "[?]", yang menandakan bahwa referensi tersebut tidak dirujuk dengan baik. Pola pengecekan yang digunakan akan dijelaskan pada *pseudocode* 8.

Algorithm 8 Reference checker function

```

1: Input: An object from the SkripsiExtract class to call the getter method
2: Output: An array containing error reports
3:
4: function ERRORCHECKING($pdf_extract)
5:   result  $\leftarrow$  Array for store all errors report
6:   sentence  $\leftarrow$  Call getter from class SkripsiExtract
7:   for each sentence in sentence do
8:     pattern  $\leftarrow$  Define regex /\B\[ \?\]\B/
9:     if pattern contain in array then
10:       fill result with errors report
11:   return result

```

Pada Pseudocode 8, pola akan memeriksa setiap kalimat yang ada pada indeks array. Kesalahan akan terdeteksi kalimat terdapat simbol "[?]", yang menandakan referensi tidak dirujuk dengan baik. Berikut adalah contoh dari kesalahan dan laporan yang dikeluarkan:

- Contoh kesalahan
Regular expression [?] adalah jenis pola teks tertentu yang dapat digunakan pada banyak aplikasi modern dan bahasa pemrograman.
- Laporan kesalahan
Referensi tidak dirujuk dengan baik, lakukan perintah PDFLatex->BibTex->PDFLatex->PDFLatex untuk memperbaikinya.

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

5.1.2 Hasil Implementasi

Perangkat lunak dibangun menggunakan bahasa pemrograman *PHP* dan *library PdfParser*. Perangkat lunak tidak memiliki *Graphical User Interface*, sehingga seluruh kegiatan dilakukan melalui terminal. Perangkat lunak akan menerima input berupa file PDF skripsi yang disimpan pada folder yang telah disediakan, dan mengeluarkan laporan kesalahan pada terminal.

Listing 5.1: Perintah yang digunakan untuk menjalankan perangkat lunak

```
1 | php main.php ../res/nama_file.pdf
```

Listing 5.1 merupakan perintah yang perlu dituliskan pada terminal, untuk menjalankan perangkat lunak. Kelas Main menjadi kelas yang digunakan untuk menjalankan seluruh proses yang berjalan dalam perangkat lunak. File PDF skripsi yang akan diperiksa harus berada di folder yang telah disediakan, yaitu pada folder Skripsi\src\res. Nama file yang digunakan pada umumnya sesuai dengan *template* skripsi yang diberikan, yaitu "skripsi.pdf". Namun pengguna juga dapat menggunakan nama yang berbeda, yang paling utama file tersebut memiliki ekstensi PDF.

Listing 5.2: Perintah yang digunakan untuk menjalankan perangkat lunak

```

1 | ERROR 1
2 | =====
3 | Error Code: PS-03
4 | Note: Perhatikan spasi setelah tanda baca.
5 | Excerpt: Namun dalam penulisannya, peserta skripsi sering melakukan kesalahan
6 | kecil yang tidak dapat diabaikan. kesalahan sering terjadi dalam penggunaan
7 | imbuhan, kata keterangan, penulisan kata dan sebagainya
8 |
9 | ERROR 2
10 | =====
11 | Error Code: PS-03
12 | Note: Perhatikan spasi setelah tanda baca.
13 | Excerpt: Regex biasanya dimanfaatkan untuk memverifikasi kecocokan antara input
14 | dengan pola teks, untuk menemukan teks yang cocok dengan pola dalam teks yang
15 | lebih besar, untuk mengganti teks yang cocok dengan pola dengan teks lain atau
16 | menyusun ulang bit dari teks yang cocok dan untuk membagi sebuah blok teks
17 | menjadi beberapa subteks
18 |
19 | ERROR 3
20 | =====
21 | Error Code: PS-05
22 | Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
23 | Excerpt: Namun dalam penulisannya, peserta skripsi sering melakukan kesalahan
24 | kecil yang tidak dapat diabaikan. kesalahan sering terjadi dalam penggunaan
25 | imbuhan, kata keterangan, penulisan kata dan sebagainya

```

Listing 5.2 merupakan hasil laporan yang dikeluarkan oleh perangkat lunak melalui *terminal windows*. Informasi yang diberikan oleh laporan tersebut yaitu, kode kesalahan, jenis kesalahan yang ditemukan dan kesalahan yang ditemukan. Laporan kesalahan yang dikeluarkan sudah diurutkan dari fitur pertama hingga terakhir, yaitu dari fitur PS-01 hingga NAT-03.

5.2 Pengujian Fungsional

Pengujian fungsional bertujuan untuk menguji fungsionalitas perangkat lunak. Perangkat lunak memiliki 8 fitur yang telah diimplementasikan. Fitur-fitur tersebut akan diuji untuk melihat kebenaran dan kesesuaian fitur tersebut dengan yang diharapkan. Untuk melakukan pengujian ini, perangkat lunak akan dijalankan sebanyak jumlah fitur yang ada. Setiap pengujian yang dilakukan, fitur yang diaktifkan hanya 1 saja secara bergantian. Hal ini dilakukan hingga seluruh fitur telah diuji.

Pada pengujian ini, perangkat lunak akan diuji dengan 2 buah kasus uji. Kedua kasus uji tersebut menggunakan *template* dokumen skripsi Informatika Unpar. Mode dokumen yang digunakan adalah mode *final*. Kedua *test case* tersebut diberi nama "TC_PF_01.pdf" dan "TC_PF_02.pdf". Isi dari kedua file tersebut serupa, namun pada file "TC_PF_02.pdf" sudah disisipkan kesalahan-kesalahan yang dapat dideteksi oleh setiap fitur yang ada. Berikut ini adalah rincian dari kesalahan-kesalahan yang dimasukkan ke dalam kasus uji tersebut.

1. Pada halaman cover bahasa Indonesia dan bahasa Inggris, data yang meliputi judul, nama mahasiswa, NPM mahasiswa, dan yang lainnya tidak diisi. Hal ini dilakukan untuk menguji fitur pemeriksa kelengkapan data skripsi (KAL-03).

2. Untuk menguji fitur pemeriksa jumlah sub bab atau sub sub bab (PS-09), pada beberapa bab hanya memiliki sebuah sub bab atau sebuah sub sub bab.
3. Pada bab 1, beberapa karakter pertama dalam awal kalimat menggunakan huruf kecil. Hal ini dilakukan untuk menguji fitur pemeriksa huruf kapital (PS-05).
4. Pada bab 2, terdapat teori yang referensinya tidak dirujuk dengan baik. File referensi.bib tidak diikutsertakan pada saat file latex dieksekusi. Hal ini dilakukan untuk menguji fitur pemeriksa referensi (NAT-01).
5. Pada bab 1 dan 2, terdapat beberapa kata yang tidak diberikan karakter spasi sebelum ataupun setelah tanda baca. Hal ini dilakukan untuk menguji fitur pemeriksa karakter spasi sebelumm atau setelah tanda baca (PS-03).
6. Pada bab 3, terdapat beberapa kata yang diketik tidak sesuai dengan kamus bahasa Indonesia. Hal ini dilakukan untuk menguji fitur pemeriksa kata (PS-01). Selain itu pada bab ini tidak diberikan kata pengantar sebelum memulai subbab, untuk menguji fitur pemeriksa kata pengantar pada bab (KAL-02).
7. Pada bab 6, terdapat kalimat yang disisipkan kata ganti orang , untuk menguji fitur pemeriksa kata ganti orang (VAN-03).

5.2.1 Menguji fitur PS-01

Hasil yang diharapkan dari pengujian fitur ini adalah perangkat lunak dapat menemukan kata-kata yang tidak terdapat pada kamus bahasa Indonesia *LibreOffice*. Berikut ini adalah hasil dari pengujian yang telah dilakukan:

1. Kasus uji TC_PF_01.pdf

Listing 5.3: Laporan Kesalahan Kasus Uji TC_PF_01

```

1 | ERROR 1
2 | =====
3 | Error Code: PS-01
4 | Note: Ditemukan penulisan kata yang tidak sesuai dengan kamus
5 | Excerpt:
6 |
7 | PENDAHULUAN Pada, dijelaskan, mengenai, penulisan, rumusan, tujuan, batasan,
8 | penelitian, merupakan, karangan, ditulis, sebagai, bagian, persya, ratan,
9 | pendidikan, akademiknya, penulisan, peserta, melakukan, kesalahan,
10 | diabaikan, Kesalahan, terjadi, penggunaan, imbuhan, keterangan,
11 | sebagainya, seharusnya, diperiksa, diminimalisir, bimbingan, pembimbing,
12 | dimanfaatkan, membahas, konten, dibanding, memeriksa, tersebut, dibuat,
13 | sebuah, pemeriksaan, berasal, dilakukan, Unpar, diseleksi,
14 | diimplementasikan, secara, dijalankan, melalui, command, Windows,
15 | menerima, masukan, berupa, le, PDF, menampilkan, laporan, berisi,
16 | ditemukan, Rumusan, Berdasarkan, dirumuskan, berikut, membuat, Tujuan,
17 | adalah, membangun, LANDASAN, TEORI Pada, dibahas, landasan, expression,
18 | library, LibreO, Expression, regex, tertentu, digunakan, pemrograman,
19 | Regex, biasanya, memveri, kecocokan, input, menemukan, mengganti,
20 | menyusun, membagi, menjadi, subteks, pencocokan, misalnya, validasi,
21 | string, username, password, mail, IP, Pemanfaatan, menyederhanakan,
22 | pemrosesan, kehidupan, sehari, mencerminkan, disebut, keteraturan,
23 | menggunakan, Deterministic, Finite, DFA, nite, state, machine yang,
24 | backtracking, berbagai, satunya, Perl, Compa, tible, PCRE, serangkaian,
25 | menerapkan, sintaks, perbedaan, Perl versi, Metakarakter, dibedakan,
26 | berdasarkan, posisinya, metakarakter, outside, square, brackets,
27 | fungsinya, berbeda, terdapat, sedangkan, inside, bracketsterdapat,
28 | Karakter, karakter, memiliki, spesi, k, dikelompokkan, setiap, didalam,
29 | bracket, mendukung, POSIX, penggunaannya, diantara, Sebagai, alnum,

```

```

30 keputusan, UmumPada, mengumpulkan, dibutuhkan, pengembangan, dicari,
31 dipilih, pelaksanaannya, dibagi, pengamatan, Pengamatan, berlangsung, Mei,
32 diamati, melainkan, diambil, pertimbangan, diuji, menghadiri, disajikan,
33 Penguji, Osfaldo, Mickael, Oktavianus, Naibaho, Penjualan, Vania, Natali,
34 S, M, T, Elisati, Ricky, Wahyudi, Menggunakan, Fitur, Surf, Dr, rer, nat,
35 Cecilia, Esti, Nugraheni, ST, MT, Ir, Veronica, Moer, tini, selanjutnya,
36 diminta, PERANCANGANPada, perancangan, dibangun, meliputi, algoritma,
37 pengecekan, Perancangan, rancangan, Rancangan, ditunjukan, Pemeriksa,
38 PENGUJIANPada, pengujian, terhadap, Lingkungan, Pengujian, lingkungan,
39 beserta, implementasinya, Berikut, pembangunan, Spesi, Processor, AMD,
40 Bristol, Ridge, Quad, Core, FX, P, GHz, GB, GPU, Radeon, RX, Storage,
41 Harddisk, TB, PHP, cli, Composer, Sublime, Text, PdfParser, Graphical,
42 User, Interface, kegiatan, disimpan, disediakan, mengeluarkan, Listing,
43 menjalankan, phpmain, php, r, s, n, m, f, i, l, p, d, KESIMPULAN,
44 SARANPada, kesimpulan, Kesimpulan

```

2. Kasus uji TC_PF_02.pdf

Listing 5.4: Laporan Kesalahan Kasus Uji TC_PF_02

```

1 ERROR 1
2
3 Error Code: PS-01
4 Note: Ditemukan penulisan kata yang tidak sesuai dengan kamus
5 Excerpt:
6
7 PENDAHULUANpada, dijelaskan, mengenai, penulisan, rumusan, tujuan,
8 batasan, penelitian, merupakan, karangan, ditulis, sebagai, bagian,
9 persya, ratan, pendidikan, akademiknya, penulisannya, peserta, melakukan,
10 kesalahan, diabaikan, terjadi, penggunaan, imbuhan, keterangan,
11 sebagainya, seharusnya, diperiksa, diminimalisir, bimbingan, pembimbing,
12 dimanfaatkan, membahas, konten, dibanding, memeriksa, tersebut, dibuat,
13 sebuah, pemeriksaan, berasal, dilakukan, Unpar, diseleksi,
14 diimplementasikan, secara, dijalankan, melalui, command, Windows,
15 menerima, masukan, berupa, le, PDF, menampilkan, laporan, berisi,
16 ditemukan, Rumusan, Berdasarkan, dirumuskan, berikut, membuat, Tujuan,
17 adalah, membangun, LANDASAN, TEORIPada, dibahas, landasan, expression,
18 library, LibreO, Expression, regex, tertentu, digunakan, pemrograman,
19 Regex, biasanya, memveri, kecocokan, input, menemukan, mengganti,
20 menyusun, membagi, menjadi, subteks, pencocokan, misalnya, validasi,
21 string, username, password, mail, IP, Pemanfaatan, menyederhanakan,
22 pemrosesan, kehidupan, sehari, mencerminkan, disebut, keterangan,
23 menggunakan, Deterministic, Finite, DFA, nite, state, machineyang,
24 backtracking, berbagai, satunya, Perl, Compa, tible, PCRE, serangkaian,
25 menerapkan, sintaks, perbedaan, Perlversi, Metakarakter, dibedakan,
26 berdasarkan, posisinya, metakarakter, outside, square, brackets,
27 fungsinya, berbeda, terdapat, sedangkan, inside, bracketsterdapat,
28 Kesalahan, UmumPda, baggian, mengumpulkan, dibutuhkan, pengembangan,
29 Infrmasi, yng, dicari, ksalahan, unummm, yaang, serring, terjaadi, paada,
30 metod, ang, dipilih, pelaksanaannya, dibagi, menjhadi, pengamatan,
31 Pengamatan, berlangsung, Mei, diamati, melainkan, diambil, pertimbangan,
32 diuji, menghadiri, disajikan, Penguji, Osfaldo, Mickael, Oktavianus,
33 Naibaho, Penjualan, Vania, Natali, S, M, T, Elisati, Ricky, Wahyudi,
34 Menggunakan, Fitur, Surf, Dr, rer, nat, Cecilia, Esti, Nugraheni, ST, MT,
35 Ir, Veronica, Moer, tini, PERANCANGANPada, perancangan, dibangun,
36 meliputi, algoritma, pengecekan, Perancangan, rancangan, Rancangan,
37 ditunjukan, Pemeriksa, PENGUJIANPada, pengujian, terhadap, Lingkungan,
38 Pengujian, lingkungan, beserta, implementasinya, Berikut, spesi,
39 pembangunan, Spesi, Processor, AMD, Bristol, Ridge, Quad, Core, FX, P,
40 GHz, GB, GPU, Radeon, RX, Storage, Harddisk, TB, PHP, cli, Composer,
41 Sublime, Text, KESIMPULAN, SARANPada, kesimpulan, Kesimpulan, berikan

```

Pada kasus uji TC_PF_02.pdf disisipkan 10 buah kesalahan penulisan kata. Kata-kata yang disisipkan dalam kasus uji tersebut, yaitu: "Pda", "baggian", "Infrmasi", "yng", "ksalahan",

"umumm", "yaang", "serring", "terjaadi", "paada", "metod", "ang" dan "menjadhi". Seluruh kesalahan yang disisipkan berhasil diperiksa dan dikeluarkan pada laporan yang ditunjukkan oleh Listing 5.4. Tingkat akurasi untuk fitur ini adalah 100%. Namun, ada beberapa hal yang belum dapat dilakukan fitur ini, berikut adalah contohnya:

- Mendeteksi nama orang.
Pada listing 5.4 terdapat beberapa nama orang yang termasuk ke dalam kesalahan, contohnya Osfaldo, Mickael, Ricky dan yang lainnya.
- Mendeteksi kata yang menggunakan bahasa asing
Pada listing 5.4 terdapat beberapa bahasa asing yang termasuk ke dalam kesalahan, contohnya *outside*, *square*, *brackets* dan yang lainnya.
- Mendeteksi kata yang memiliki imbuhan
Pada listing 5.4 terdapat beberapa kata berimbuhan yang termasuk ke dalam kesalahan, contohnya menggunakan, pengecekan, perancangan dan yang lainnya.
- Mendeteksi kata yang saling berdempetan
Pada listing 5.4 terdapat beberapa kata berdempetan yang termasuk ke dalam kesalahan, contohnya PERANCANGANPada, UmumPda dan yang lainnya.
- Mendeteksi kata yang tidak utuh
Pada listing 5.4 terdapat beberapa kata yang tidak utuh (terpotong oleh "-") yang termasuk ke dalam kesalahan, contohnya persya-ratan menjadi persya dan ratan.

5.2.2 Menguji fitur PS-03

Hasil yang diharapkan dari pengujian fitur ini adalah perangkat lunak dapat menemukan kata-kata yang tidak diberi spasi setelah tanda baca. Berikut ini adalah hasil dari pengujian yang telah dilakukan:

1. Kasus uji TC_PF_01.pdf

Listing 5.5: PLaporan Kesalahan Kasus Uji TC_PF_01

```

1 | ERROR 1
2 | =====
3 | Error Code: PS-03
4 | Note: Perhatikan spasi setelah tanda baca.
5 | Excerpt: Listing 5.1: Perintah yang digunakan untuk menjalankan perangkat lunak
6 | lphpmain.php

```

2. Kasus uji TC_PF_02.pdf

Listing 5.6: Laporan Kesalahan Kasus Uji TC_PF_02

```

1 | ERROR 1
2 | =====
3 | Error Code: PS-03
4 | Note: Perhatikan spasi setelah tanda baca.
5 | Excerpt: Namun dalam penulisannya, peserta skripsi sering melakukan kesalahan
6 | kecil yang tidak dapat diabaikan. kesalahan sering terjadi dalam penggunaan
7 | imbuhan, kata keterangan, penulisan kata dan sebagainya
8 |
9 | ERROR 2
10 | =====
11 | Error Code: PS-03
12 | Note: Perhatikan spasi setelah tanda baca.
13 | Excerpt: Regex biasanya dimanfaatkan untuk memverifikasi kecocokan antara input
14 | dengan pola teks, untuk menemukan teks yang cocok dengan pola dalam teks yang

```

```

15 | lebih besar , untuk mengganti teks yang cocok dengan pola dengan teks lain atau
16 | menyusun ulang bit dari teks yang cocok dan untuk membagi sebuah blok teks
17 | menjadi beberapa subteks

```

Pada kasus uji TC_PF_02.pdf disisipkan 2 buah kesalahan dalam penggunaan spasi setelah tanda baca. Kesalahan tersebut terdapat pada sub bab 1.1 baris ke-3 dan sub bab 2.1 baris ke-3. Kedua kesalahan tersebut berhasil diperiksa dan dikeluarkan pada laporan yang ditunjukkan oleh Listing 5.6. Tingkat akurasi untuk fitur ini adalah 100%. Namun, ada beberapa hal yang belum dapat dilakukan fitur ini, berikut adalah contohnya:

- Fitur ini masih belum bisa membedakan penggunaan tanda titik pada gelar pendidikan, perangkat lunak mendeteksi hal tersebut menjadi sebuah kesalahan.
- Pada listing 5.5, fitur ini juga belum dapat membedakan penggunaan tanda titik pada nama file. Hal ini juga berlaku untuk kata yang merujuk tabel, gambar dan lain-lain. Misalnya Tabel 4.1, Gambar 2.5 dan lain-lain.

5.2.3 Menguji fitur PS-05

Hasil yang diharapkan dari pengujian fitur ini adalah perangkat lunak dapat menemukan karakter pertama yang tidak menggunakan huruf kapital pada sebuah kalimat. Berikut ini adalah hasil dari pengujian yang telah dilakukan:

1. Kasus uji TC_PF_01.pdf

Pada kasus uji ini, perangkat lunak tidak menemukan kesalahan dalam dokumen skripsi.

2. Kasus uji TC_PF_02.pdf

Listing 5.7: Laporan Kesalahan Kasus Uji TC_PF_02

```

1 | ERROR 1
2 | =====
3 | Error Code: PS-05
4 | Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
5 | Excerpt: Namun dalam penulisannya , peserta skripsi sering melakukan kesalahan
6 | kecil yang tidak dapat diabaikan. kesalahan sering terjadi dalam penggunaan
7 | imbuhan, kata keterangan , penulisan kata dan sebagainya
8 |
9 | ERROR 2
10 | =====
11 | Error Code: PS-05
12 | Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
13 | Excerpt: pada saat bimbingan , waktu dosen pembimbing lebih baik dimanfaatkan
14 | untuk membahas konten dibanding memeriksa kesalahan-kesalahan tersebut
15 |
16 | ERROR 3
17 | =====
18 | Error Code: PS-05
19 | Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
20 | Excerpt: dari masalah tersebut dapat dibuat sebuah aplikasi untuk melakukan
21 | pemeriksaan pada dokumen skripsi
22 |
23 | ERROR 4
24 | =====
25 | Error Code: PS-05
26 | Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
27 | Excerpt: kesalahan yang akan diperiksa berasal dari survei yang dilakukan
28 | kepada dosen-dosen Informatika Unpar
29 |
30 | ERROR 5

```

```

31 |=====
32 | Error Code: PS-05
33 | Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
34 | Excerpt: hasil dari survei tersebut akan diseleksi untuk diimplementasikan ke
35 | dalam aplikasi
36 |
37 | ERROR 6
38 |=====
39 | Error Code: PS-05
40 | Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
41 | Excerpt: aplikasi sederhana ini dapat dimanfaatkan oleh mahasiswa Informatika
42 | Unpar secara mandiri
43 |
44 | ERROR 7
45 |=====
46 | Error Code: PS-05
47 | Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
48 | Excerpt: aplikasi ini dijalankan melalui melalui terminal command Windows
49 |
50 | ERROR 8
51 |=====
52 | Error Code: PS-05
53 | Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
54 | Excerpt: aplikasi menerima masukan berupa file PDF skripsi dan menampilkan
55 | laporan yang berisi kesalahan-kesalahan yang ditemukan pada dokumen skripsi

```

Pada kasus uji TC_PF_02.pdf disisipkan 9 buah kesalahan dalam penggunaan huruf kapital pada awal kalimat. Kesalahan tersebut disisipkan pada 9 kalimat yang ada pada bab 1. Listing 5.7 hanya menemukan 8 buah kesalahan. Satu kesalahan yang tidak ditemukan berada pada kata pengantar setelah penulisan bab dan judulnya. Kesalahan ini tidak terdeteksi karena hasil ekstrak judul bab dan kata pertama pada kalimat pengantar tergabung menjadi satu. Tingkat akurasi untuk fitur ini adalah 88,89%.

5.2.4 Menguji fitur PS-09

Hasil yang diharapkan dari pengujian fitur ini adalah perangkat lunak dapat menemukan bab atau sub bab yang hanya memiliki satu sub bab atau sub sub bab. Berikut ini adalah hasil dari pengujian yang telah dilakukan:

1. Kasus uji TC_PF_01.pdf

Pada kasus uji ini, perangkat lunak tidak menemukan kesalahan dalam dokumen skripsi.

2. Kasus uji TC_PF_02.pdf

Listing 5.8: Laporan Kesalahan Kasus Uji TC_PF_02

```

1 | ERROR 1
2 |=====
3 | Error Code: PS-09
4 | Note: Bab/Subbab 2.1 ini hanya terdapat 1 sub bab/sub sub bab
5 |
6 | ERROR 2
7 |=====
8 | Error Code: PS-09
9 | Note: Bab/Subbab 2.1.1 ini hanya terdapat 1 sub bab/sub sub bab
10 |
11 | ERROR 3
12 |=====
13 | Error Code: PS-09
14 | Note: Bab/Subbab 3.1 ini hanya terdapat 1 sub bab/sub sub bab
15 |

```

```

16 | ERROR 4
17 |=====
18 | Error Code: PS-09
19 | Note: Bab/Subbab 3.1.1 ini hanya terdapat 1 sub bab/sub sub bab
20 |
21 | ERROR 5
22 |=====
23 | Error Code: PS-09
24 | Note: Bab/Subbab 4.1 ini hanya terdapat 1 sub bab/sub sub bab
25 |
26 | ERROR 6
27 |=====
28 | Error Code: PS-09
29 | Note: Bab/Subbab 5.1 ini hanya terdapat 1 sub bab/sub sub bab
30 |
31 | ERROR 7
32 |=====
33 | Error Code: PS-09
34 | Note: Bab/Subbab 5.1.1 ini hanya terdapat 1 sub bab/sub sub bab

```

Laporan yang ditunjukkan pada Listing 5.8 sudah sesuai dengan yang diharapkan.

Pada kasus uji TC_PF_02.pdf disisipkan 7 buah kesalahan. Kesalahan tersebut berkaitan dengan jumlah sub bab atau sub sub bab yang ada. Ada 4 bab yang hanya memiliki 1 sub bab, yaitu bab 2, 3, 4 dan 5. Selain itu ada 3 sub bab yang hanya memiliki 1 sub sub bab, yaitu sub bab 2.1, 3.1 dan 5.1. Pada Listing 5.7 seluruh kesalahan berhasil ditemukan. Tingkat akurasi untuk fitur ini adalah 100%.

5.2.5 Menguji fitur KAL-02

Hasil yang diharapkan dari pengujian fitur ini adalah perangkat lunak dapat menemukan bab yang tidak diberikan kata pengantar. Berikut ini adalah hasil dari pengujian yang telah dilakukan:

1. Kasus uji TC_PF_01.pdf
Pada kasus uji ini, perangkat lunak tidak menemukan kesalahan dalam dokumen skripsi.
2. Kasus uji TC_PF_02.pdf

Listing 5.9: Laporan Kesalahan Kasus Uji TC_PF_02

```

1 | ERROR 1
2 |=====
3 | Error Code: KAL-02
4 | Note: Berilah kata pengantar untuk setiap bab
5 | Excerpt: BAB 3 ANALISIS MASALAH 3.1 Survei Kesalahan Umum Pada bagian ini akan
6 | dijelaskan tentang survei yang dilakukan untuk mengumpulkan informasi yang
7 | dibutuhkan dalam pengembangan perangkat lunak

```

Pada kasus uji TC_PF_02.pdf disisipkan sebuah kesalahan. Kesalahan tersebut disisipkan pada bab 3. Pada bab 3 setelah penulisan bab dan judul bab, tidak diberikan kata pengantar. Pada Listing 5.9 kesalahan tersebut berhasil ditemukan. Tingkat akurasi untuk fitur ini adalah 100%.

5.2.6 Menguji fitur KAL-03

Hasil yang diharapkan dari pengujian fitur ini adalah perangkat lunak dapat menemukan data skripsi yang belum diisi. Berikut ini adalah hasil dari pengujian yang telah dilakukan:

1. Kasus uji TC_PF_01.pdf
Pada kasus uji ini, perangkat lunak tidak menemukan kesalahan dalam dokumen skripsi.

2. Kasus uji TC_PF_02.pdf

Listing 5.10: Laporan Kesalahan Kasus Uji TC_PF_02

```

1 | ERROR 1
2 | =====
3 | Error Code: KAL-03
4 | Note: Ada data skripsi yang belum dilengkapi pada halaman cover
5 | Excerpt: SKRIPSI/TUGAS AKHIR JUDUL BAHASA INDONESIA Nama Lengkap NPM: 10 digit
6 | NPM UNPAR PROGRAM STUDI MATEMATIKA/FISIKA/TEKNIK INFORMATIKA FAKULTAS
7 | TEKNOLOGI INFORMASI DAN SAINS UNIVERSITAS KATOLIK PARAHYANGAN tahun FINAL
8 | PROJECT/UNDERGRADUATE THESIS JUDUL BAHASA INGGRIS Nama Lengkap NPM: 10 digit
9 | NPM UNPAR DEPARTMENT OF MATHEMATICS/PHYSICS/INFORMATICS FACULTY OF INFORMATION
10 | TECHNOLOGY AND SCIENCES PARAHYANGAN CATHOLIC UNIVERSITY tahun

```

Pada kasus uji TC_PF_02.pdf disisipkan dua buah kesalahan. Kesalahan tersebut disisipkan pada halaman cover Bahasa Indonesia dan Bahasa Inggris. Isi dari file data.tex tidak diisi satupun, sehingga masih berisi kode *template* yang diberikan. Pada Listing 5.10 kesalahan tersebut berhasil ditemukan. Tingkat akurasi untuk fitur ini adalah 100%.

5.2.7 Menguji fitur NAT-01

Hasil yang diharapkan dari pengujian fitur ini adalah perangkat lunak dapat menemukan referensi yang tidak dirujuk dengan baik. Berikut ini adalah hasil dari pengujian yang telah dilakukan:

1. Kasus uji TC_PF_01.pdf

Pada kasus uji ini, perangkat lunak tidak menemukan kesalahan dalam dokumen skripsi.

2. Kasus uji TC_PF_02.pdf

Listing 5.11: Laporan Kesalahan Kasus Uji TC_PF_02

```

1 | ERROR 1
2 | =====
3 | Error Code: NAT-01
4 | Note: Referensi tidak dirujuk dengan baik, lakukan perintah PDFLatex->BibTex->
5 | PDFLatex->PDFLatex untuk memperbaikinya
6 | Excerpt: 2.1 Regular Expression Regular expression (regex) [?] adalah jenis
7 | pola teks tertentu yang dapat digunakan pada banyak aplikasi modern dan bahasa
8 | pemrograman
9 |
10 |
11 | ERROR 2
12 | =====
13 | Error Code: NAT-01
14 | Note: Referensi tidak dirujuk dengan baik, lakukan perintah PDFLatex->BibTex->
15 | PDFLatex->PDFLatex untuk memperbaikinya
16 | Excerpt: PCRE [?] adalah serangkaian fungsi yang menerapkan pencocokan pola
17 | regex dengan menggunakan sintaks dan semantik yang sama dengan bahasa
18 | pemrograman Perl 5, meskipun ada beberapa sedikit perbedaan

```

Pada kasus uji TC_PF_02.pdf disisipkan dua buah kesalahan. Kesalahan tersebut disisipkan pada bab 2. Pada bab 2, terdapat 2 buah landasan teori yang referensinya tidak muncul. Pada Listing 5.11 kedua kesalahan tersebut berhasil ditemukan. Tingkat akurasi untuk fitur ini adalah 100%.

5.2.8 Menguji fitur VAN-03

Hasil yang diharapkan dari pengujian fitur ini adalah perangkat lunak dapat menemukan kata ganti orang pada kalimat. Berikut ini adalah hasil dari pengujian yang telah dilakukan:

1. Kasus uji TC_PF_01.pdf
Pada kasus uji ini, perangkat lunak tidak menemukan kesalahan dalam dokumen skripsi.
2. Kasus uji TC_PF_02.pdf

Listing 5.12: Laporan Kesalahan Kasus Uji TC_PF_02

```

1 | ERROR 1
2 | =====
3 | Error Code: VAN-03
4 | Note: Kalimat ini mengandung kata ganti orang
5 | Excerpt: 6.1Kesimpulan Kesimpulan yang saya dapat dari pengembangan perangkat
6 | lunak ini adalah sebagai berikut
7 |
8 | ERROR 2
9 | =====
10 | Error Code: VAN-03
11 | Note: Kalimat ini mengandung kata ganti orang
12 | Excerpt: 6.2Saran Saran yang saya berikan untuk pengembangan perangkat lunak
13 | ini adalah sebagai berikut

```

Pada kasus uji TC_PF_02.pdf disisipkan dua buah kesalahan. Kesalahan tersebut disisipkan pada bab 6. Pada bab 6, terdapat 2 sub bab untuk kesimpulan dan saran. Masing-masing sub bab disisipkan kata ganti orang sebanyak 1 kata. Pada Listing 5.12 kedua kesalahan tersebut berhasil ditemukan. Tingkat akurasi untuk fitur ini adalah 100%.

5.2.9 Kesimpulan Pengujian Fungsional

Pengujian fungsional telah dilakukan dengan menggunakan 2 kasus uji, berupa dokumen skripsi yang dibuat menggunakan *template* skripsi FTIS UNPAR. Pada pengujian ini dapat ditarik beberapa kesimpulan, yaitu sebagai berikut:

- Fitur-fitur yang ada pada perangkat lunak dapat mengeluarkan laporan kesalahan sesuai dengan hasil yang diharapkan. Hal tersebut dapat dilihat dari presentase keakuratan dari setiap fitur yang sangat tinggi.
- Dari 8 fitur yang diimplementasi, masih ada 2 fitur yang perlu disempurnakan lagi. Masih ada yang belum bisa ditangani seperti yang sudah dijelaskan pada sub bab 5.2.

5.3 Pengujian Eksperimental

Pada pengujian eksperimental, perangkat lunak akan diuji dengan 5 buah kasus uji dokumen skripsi yang diambil dari *Github* FTIS Unpar ¹. Dokumen yang digunakan sebagai pengujian, yaitu:

1. Skripsi yang ditulis oleh Cornelius David Herianto, dengan judul Pengelompokan Dokumen Berbasis Algoritma Genetika. Dokumen ini diberi nama TC_PE_01.pdf [5]
2. Skripsi yang ditulis oleh Stillmen Vallian, dengan judul Kustomisasi Sharif Judge untuk Kebutuhan Program Studi Teknik Informatika. Dokumen ini diberi nama TC_PE_02.pdf [6]
3. Skripsi yang ditulis oleh Nancy Valentina, dengan judul Aplikasi Pratinjau 3 Dimensi Berbasis Web. Dokumen ini diberi nama TC_PE_03.pdf [7]
4. Skripsi yang ditulis oleh Evelyn Wijaya, dengan judul Pembangunan Gim Snake 360 Berbasis Web dengan Kode Terbuka. Dokumen ini diberi nama TC_PE_04.pdf [8]

¹<https://github.com/ftisunpar/Skripsi>

5. Skripsi yang ditulis oleh Ellena Angelica, dengan judul Kolektor Pengumuman Informatika. Dokumen ini diberi nama TC_PE_05.pdf [9]

Pada pengujian ini, ke-5 kasus uji tersebut akan dijalankan pada perangkat lunak. Berikut adalah hasil yang dikeluarkan oleh perangkat lunak dari setiap kasus uji yang digunakan:

1. TC_PE_01.pdf

Listing 5.13: Pesan Kesalahan Secured PDF

```

1 Fatal error: Uncaught Exception: Object list not found. Possible secured file .
2 in D:\Skripsi\src\vendor\smalot\pdfparser\src\Smalot\PdfParser\Parser.php:105
3 Stack trace:
4 #0 D:\Skripsi\src\vendor\smalot\pdfparser\src\Smalot\PdfParser\Parser.php(81):
5 Smalot\PdfParser\Parser->parseContent('%PDF-1.5\n%\xD0\xD4\xC5\xD8\n...')
6 #1 D:\Skripsi\src\pdf_checker\SkripsiExtract.php(26): Smalot\PdfParser\Parser->
7 parseFile('D:\Skripsi\src\...')
8 #2 D:\Skripsi\src\pdf_checker\Main.php(9): SkripsiExtract->extractFromPDF('../
9 res/TC_PE_01...')
10 #3 {main}
11 thrown in D:\Skripsi\src\vendor\smalot\pdfparser\src\Smalot\PdfParser\Parser .
12 php on line 105

```

Pada kasus uji ini, perangkat lunak mendeteksi bahwa file ini merupakan file yang *secured*. Sebagaimana telah dijelaskan pada sub bab 2.2, bahwa *library* ini tidak dapat melakukan ekstrak dokumen yang *secured*. Namun masalah ini dapat diatasi dan akan dibahas lebih lanjut pada sub sub bab 5.3.2. Berikut ini adalah laporan kesalahan yang ditemukan:

Listing 5.14: Laporan Kesalahan Kasus Uji TC_PE_01

```

1
2 ERROR 1
3
4 Error Code: PS-01
5 Note: Ditemukan penulisan kata yang tidak sesuai dengan kamus
6 Excerpt:
7
8 PENDAHULUAN, BelakangPengelompokan, clustering, merupakan, mencari,
9 kumpulan, melibatkan, pemilihan, cluster, dibandingkan, berada,
10 Clustering, berguna, mereduksi, karakteristik, tertentu, mengembangkan,
11 klasi, dikenal, sebagai, memberikan, masukkan, dukungan, terhadap,
12 mengenai, pembelajaran, terarah, unsupervised, learning, Pembagian,
13 berdasarkan, diketahui, sebelumnya, melainkan, kesamaan, menurut, ukuran,
14 Document, pengelompokan, lakukan, Pengelompokan, diterapkan, penambangan,
15 web, pencari, search, engine, information, retrieval, dilakukan, adalah,
16 mengukur, kemiripan, similarity, mengelompokan, serupa, terdiri, tulisan,
17 algoritma, digunakan, K, means, membagi, tersebut, dibentuk, meminimalkan,
18 centroid, setiap, dicari, menggunakan, mean, dimodelkan, n, banyaknya,
19 terbukti, melakukan, apapun, memiliki, kekurangan, terjebak, local,
20 optima, tergantung, centroidawal, ditangani, Genetic, Algorithm, GA,
21 menyelesaikan, pencarian, optimasi, heuristik, menirukan, secara, terjadi,
22 survival, of, the, ttest, News, disediakan, menjadi, Karakteristik,
23 datasetini, Terdiri, berasal, website, Newsdari, ditulis, Terbagi,
24 business, entertainment, politics, dantech, sportterdapat, techterdapat,
25 plain, textyang, le, TXT, diproses, terlebih, membersihkan, mengambil,
26 berupa, karakter, kapitalisasi, diabaikan, diubah, tech, dilampirkan,
27 Lampiran, menemukan, direpresentasikan, diambil, informasinya, diolah,
28 merepresentasikan, PERANCANGANPada, perancangan, rancangan, pengguna,
29 Kebutuhan, kebutuhan, masukan, diakomodasi, Graphical, User, Interface,
30 Rincian, melalui, berformat, CSV, dibuka, pengolah, spreadsheet,
31 Microsoft, Excel, File, laporan, dibutuhkan, keluaran, leCSV, dilihat,
32 penamaan, YYYY, MM, DD, hh, mm, ss, csv, Januari, KMeans, PENGUJIAN,
33 Pengujian, spesi, berikut, processor, R, Core, TM, i, HQ, CPU, GHz, MB,

```

Windows, Build, diteliti, antaranya, terikat, tujuan, mengetahui, hubungan, Banyaknya, individu, pembobotan, penghitungan, TF, IDF, terjadinya, pembentukan, keturunan, Individu, elitisme, tnessterbaik, selanjutnya, selama, terakhir, nya, KESIMPULAN, Kesimpulan, Berdasarkan, penge, lumpokan, diperlukan, sebelum, mengelompokkan, tasi, Merepresentasikan, tersusun, Kcentroid, tnessyang, intracluster, berhasil, dibuat, purity, sebesar, mengatasi, membutuhkan, disebabkan, komputasi, alasan, berjalan, Mulai, kedua, menyusun, bersifat, sparse, bernilai, Centroid, memperlambat, perhitungan, cosine, seharusnya, mengabaikan, berbobot, merepre, sentasikan, berbanding, kemungkinannya, semakin, sejauh, Kemungkinan, jaraknya

ERROR 2

Error Code: PS-03
 Note: Perhatikan spasi setelah tanda baca.
 Excerpt: Format penamaan dari **file** ini adalah "algoritma-YYYY.MM.DD hh_mm_ss.csv"

ERROR 3

Error Code: KAL-02
 Note: Berilah kata pengantar untuk setiap bab
 Excerpt: BAB 1 PENDAHULUAN 1.1 Latar Belakang Pengelompokan (clustering) merupakan prosedur untuk mencari struktur alami dari suatu kumpulan data

ERROR 4

Error Code: KAL-02
 Note: Berilah kata pengantar untuk setiap bab
 Excerpt: BAB 5 PENGUJIAN DAN EKSPERIMEN 5.1 Skenario Pengujian
 Eksperimental Eksperimen dilakukan dengan menggunakan spesifikasi komputer sebagai berikut: 1. Tipe processor : Intel(R) Core(TM) i7-4720HQ CPU @2.60GHz 2. Memori: 12288MB RAM 3. Sistem operasi: Windows 10 Pro 64-bit (10.0, Build 17763) Pada suatu penelitian, ada beberapa jenis variabel yang diteliti di antaranya variabel bebas, variabel terikat, dan variabel kontrol

ERROR 5

Error Code: KAL-02
 Note: Berilah kata pengantar untuk setiap bab
 Excerpt: BAB 6 KESIMPULAN DAN SARAN 6.1 Kesimpulan Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut: 1. Berdasarkan dataset yang telah digunakan, algoritma genetika dapat digunakan dalam penge- lumpokan dokumen

ERROR 6

Error Code: NAT-01
 Note: Referensi tidak dirujuk dengan baik, lakukan perintah PDFLatex->BibTex->PDFLatex->PDFLatex untuk memperbaikinya
 Excerpt: Pembagian kelompok dalam clustering tidak berdasarkan sesuatu yang telah diketahui sebelumnya, melainkan berdasarkan kesamaan tertentu menurut suatu ukuran tertentu [?]

ERROR 7

Error Code: NAT-01
 Note: Referensi tidak dirujuk dengan baik, lakukan perintah PDFLatex->BibTex->PDFLatex->PDFLatex untuk memperbaikinya
 Excerpt: Pengelompokan dokumen diterapkan dalam beberapa bidang seperti penambangan web, mesin pencari (search engine), dan temu kembali informasi (information retrieval) [?]


```

97 |
98 | ERROR 8
99 | =====
100 | Error Code: NAT-01
101 | Note: Referensi tidak dirujuk dengan baik, lakukan perintah PDFLatex->BibTex->
102 | PDFLatex->PDFLatex untuk memperbaikinya
103 | Excerpt: GA merupakan teknik pencarian heuristik tingkat tinggi yang menirukan
104 | proses evolusi yang secara alami terjadi [ ?] berdasarkan prinsip survival of
105 | the fittest
106 |
107 | ERROR 9
108 | =====
109 | Error Code: NAT-01
110 | Note: Referensi tidak dirujuk dengan baik, lakukan perintah PDFLatex->BibTex->
111 | PDFLatex->PDFLatex untuk memperbaikinya
112 | Excerpt: Algoritma ini dinamakan demikian karena menggunakan konsep-konsep
113 | dalam genetika sebagai model pemecahan masalahnya [ ?]
114 |
115 | ERROR 10
116 | =====
117 | Error Code: NAT-01
118 | Note: Referensi tidak dirujuk dengan baik, lakukan perintah PDFLatex->BibTex->
119 | PDFLatex->PDFLatex untuk memperbaikinya
120 | Excerpt: 2.1 Pengelompokan 2.1.1 Definisi Pengelompokan Pengelompokan (
121 | clustering ) merupakan sebuah metode untuk menggabungkan himpunan objek ke
122 | dalam kelompok-kelompok sedemikian rupa sehingga objek dalam satu kelompok
123 | ( cluster ) lebih mirip (karena suatu hal) satu sama lain daripada objek di
124 | kelompok lain [ ?]

```

2. TC_PE_02.pdf

Perangkat lunak tidak mengeluarkan laporan kesalahan yang ditemukan pada dokumen. Pada terminal juga tidak diberikan pesan error yang menandakan bahwa terdapat kesalahan pada perangkat lunak. Namun masalah ini dapat diatasi dan akan dibahas lebih lanjut pada sub bab 5.3.1. Berikut ini adalah laporan kesalahan yang ditemukan:

Listing 5.15: Laporan Kesalahan Kasus Uji TC_PE_02

```

1 | ERROR 1
2 | =====
3 | Error Code: PS-01
4 | Note: Ditemukan penulisan kata yang tidak sesuai dengan kamus
5 | Excerpt:
6 |
7 | PENDAHULUAN, BelakangPengumuman, jurusan, UNPAR, umumnya, dilakukan,
8 | Pengumuman, pengumuman, dijamin, dituju, setelah, dikirim, layanan,
9 | memiliki, terorganisir, Berbagai, tercampur, menyulitkan, pemilik,
10 | mencari, mengakibatkan, terbaca, secara, Penelitian, membuat, kekurangan,
11 | tersebut, BlueTape, adalah, berfungsi, membantu, urusan, paper, based,
12 | FTIS, menjadi, paperless, Fitur, mengumpulkan, berisi, menampilkannya,
13 | penerima, memanfaatkan, LINE, Line, Corporation, memungkinkan, disebut,
14 | mengirim, pengikut, bersamaan, Penerima, diminta, mengikuti, menerima,
15 | noti, diakses, melalui, internet, Heroku, cloud, ketiga, penggunanya,
16 | membangun, menjalankan, mengoperasikan, Rumusan, memodi, berjalan,
17 | mengimplementasikan, menguji, Tujuan, Mempelajari, Memodi, Menguji,
18 | LANDASAN, landasan, dipakai, berjalan, mendukung, pemrograman, meliputi,
19 | Ruby, Node, js, Java, Python, Clojure, Scala, Go, PHP, menyebarkan,
20 | mengelola, ditulis, didukung, mende, nisikan, sebagai, gabungan, source,
21 | code, dependencyyang, leProc, le, Dependency, dibangun, dijalankan,
22 | membutuhkan, dependency, Aturan, peletakkan, berbeda, menuliskan,
23 | lepackage, json, Dyno, berbasis, Unix, web, dyno, disebar, dikelompokkan,
24 | Web, process, type, satunya, HTTP, routerHeroku, Worker, PERANCANGANBab,
25 | membahas, perancangan, Pembahasan, dibagi, bagian, Perancangan, Class,
26 | merupakan, pembangunan, Penjelasan, diberikan, control, view, PENGUJIAN,

```

pengujian, ImplementasiBagian, terlebih, deploy, dalamnya, diberi,
 Shadowtape, shadowtape, herokuapp, Setelah, dimodi, peneliti, Gmail,
 shadowbluetape, gmail, bernama, Lingkungan, Pengembangan, Berikut, spesi,
 Spesi, Processor, R, Celeron, CPU, U, GHz, x, Graphics, Ivybridge, Mobile,
 GB, Harddisk, SATA, Ubuntu, LTS, Code, version, apache, v, cli, Composer,
 pgAdmin, Application, Desktop, psql, PostgreSQL, heroku, linux, node,
 KESIMPULAN, Kesimpulan, kesimpulan, diambil, penelitian, comsetiap,
 diidenti, ditampilkan, memunculkan, teruji, Mengadaptasi, kon, gurasi, O,
 cial, Account, Meneliti, loginyang, ditemukan, Meningkatkan, usabilitas,
 menggunakannya

ERROR 2

Error Code: PS-03
 Note: Perhatikan spasi setelah tanda baca.
 Excerpt: Heroku mendukung beberapa bahasa pemrograman, meliputi: Ruby, Node.js,
 Java, Python, Clojure, Scala, Go, dan PHP

ERROR

Error Code: PS-03
 Note: Perhatikan spasi setelah tanda baca.
 Excerpt: Contoh: Aplikasi dengan bahasa Node.js menuliskan deskripsi dependency
 padafilepackage.json

ERROR 4

Error Code: PS-03
 Note: Perhatikan spasi setelah tanda baca.
 Excerpt: azurewebsites.net/ (Gambar3.1)

ERROR 5

Error Code: PS-03
 Note: Perhatikan spasi setelah tanda baca.
 Excerpt: Kode program ini dapat diakses di <https://github.com/ftisunpar/BlueTape>

ERROR 6

Error Code: PS-03
 Note: Perhatikan spasi setelah tanda baca.
 Excerpt: MVC (Model-View-Control ler) adalah sebuah metode untuk membuat
 perangkat lunak menjadi tiga1 <https://github.com/ftisunpar/BlueTape>

PERANCANGANBab ini membahas perancangan fitur kolektor pengumuman

ERROR 7

Error Code: PS-03
 Note: Perhatikan spasi setelah tanda baca.
 Excerpt: Aplikasi web tersebut diberi nama Shadowtape dan dapat diakses di
<https://shadowtape.herokuapp.com/>

ERROR 8

Error Code: PS-03
 Note: Perhatikan spasi setelah tanda baca.
 Excerpt: Akun email baru menggunakan layanan email Gmail dengan alamat
 shadowbluetape@gmail.com

ERROR 9

Error Code: PS-03

```

90 | Note: Perhatikan spasi setelah tanda baca.
91 | Excerpt: BlueTape versi skripsi ini diberi nama shadowtape dan dapat diakses di
92 | https://shadowtape.herokuapp.com/
93 |
94 | ERROR 10
95 | =====
96 | Error Code: PS-03
97 | Note: Perhatikan spasi setelah tanda baca.
98 | Excerpt: Fitur ini dapat melakukan sinkronisasi kotak masuk untuk alamat
99 | email shadowbluetape@gmail.comsetiap jam
100 |
101 | ERROR 11
102 | =====
103 | Error Code: PS-03
104 | Note: Perhatikan spasi setelah tanda baca.
105 | Excerpt: Email yang diidentifikasi sebagai pengumuman dapat ditampilkan di
106 | https://shadowtape.herokuapp
107 |
108 | ERROR 12
109 | =====
110 | Error Code: PS-09
111 | Note: Bab/Subbab 2.1 ini hanya terdapat 1 sub bab/sub sub bab
112 |
113 | ERROR 13
114 | =====
115 | Error Code: PS-09
116 | Note: Bab/Subbab 3.1 ini hanya terdapat 1 sub bab/sub sub bab
117 |
118 | ERROR 14
119 | =====
120 | Error Code: PS-09
121 | Note: Bab/Subbab 3.1.1 ini hanya terdapat 1 sub bab/sub sub bab
122 |
123 | ERROR 15
124 | =====
125 | Error Code: KAL-02
126 | Note: Berilah kata pengantar untuk setiap bab
127 | Excerpt: BAB 1 PENDAHULUAN 1.1Latar BelakangPengumuman di jurusan Teknik
128 | Informatika UNPAR pada umumnya dilakukan lewat email

```

3. TC_PE_03.pdf

Perangkat lunak tidak mengeluarkan laporan kesalahan yang ditemukan pada dokumen. Pada terminal juga tidak diberikan pesan error yang menandakan bahwa terdapat kesalahan pada perangkat lunak. Namun masalah ini dapat diatasi dan akan dibahas lebih lanjut pada sub bab 5.3.1. Berikut ini adalah laporan kesalahan yang ditemukan:

Listing 5.16: Laporan Kesalahan Kasus Uji TC_PE_03

```

1 | ERROR 1
2 | =====
3 | Error Code: PS-01
4 | Note: Ditemukan penulisan kata yang tidak sesuai dengan kamus
5 | Excerpt:
6 |
7 | PENDAHULUAN, BelakangSnake, merupakan, sebuah, permainan, dibuat,
8 | Peter, Trefonas, Snake, berasal, Blockade, Awalnya, dimainkan, Nokia,
9 | bermain, adalah, pemain, mengerjakan, tersebut, mendapatkan, makanan,
10 | sebanyak, banyaknya, menabrak, Setiap, memakan, memanjang, semakin, meng,
11 | gerakan, menutupi, HTML, Hyper, Text, Markup, Language, digunakan,
12 | membuat, web, penerus, XHTML, memiliki, satunya, Canvas, menggambar,
13 | pixel, ditulis, gunakan, pemrograman, JavaScript, Javascript, menjadi,
14 | jQuery, library, traversal, penanganan, event, Ajax, de, ngan,
15 | Application, Programming, Interface, GitHub, layanan, hosting, bersama,

```

pengembangan, menggunakan, version, control, Git, adanya, programmer,
 etahui, perubahan, termakan, ditempatkan, secara, Permainan, berakhir,
 tubuhnya, jumlahnya, warnanya, bermacam, menambah, Silther,
 PERANCANGANPada, mengenai, perancangan, dibangun, Perancangan, dilakukan,
 meliputi, sequence, state, tampilan, Rancangan, Sequence, ditunjukan,
 dijelaskan, Source, sequenceyang, memuat, sequenceuntuk, memilih, Berikut,
 penjelasan, menerima, input, mencari, le, Filelabirin, ditemukan,
 memanggil, method, Maze, lelabirin, Setelah, DrawingObject, digambar,
 PENGUJIANPada, pengujian, linkini, generaldevilx, github, lingkungan,
 Lingkungan, pembangunan, Processor, Core, i, U, GHz, GB, Card, GeForce,
 MX, Storage, TB, mobile, berbasis, Android, desktop, SM, J, G, Exynos,
 Octa, MHz, Cortex, KESIMPULAN, SARANPada, berisi, kesimpulan, Kesimpulan,
 diambil, diantaranya, Pembangunan, berhasil, berbelok, smartphone,
 diharapkan, menekan, Play, Menambahkan, berdasarkan, penguji, buatan,
 Menyimpan, disimpan, dimuat, Berdasarkan, dipaparkan, mengurangi,
 kesalahan, pembuatan, terutama, memposisikan, Seharusnya, melainkan,
 dituliskan, Sebelum, sebaiknya, preview, terlebih, melihat, dipilih
 33
 34 ERROR 2
 35
 36 Error Code: PS-03
 37 Note: Perhatikan spasi setelah tanda baca.
 38 Excerpt: 1.2Rumusan Masalah Rumusan dari masalah yang akan dibahas pada skripsi
 39 ini adalah sebagai berikut: Bagaimana membangun permainan Snakemenggunakan
 40 HTML5 Canvas? Bagaimana cara menyimpan labirin pada fileeksternal? 1
 41 [https://en.wikipedia.org/wiki/Snake_\(video_game_genre\)](https://en.wikipedia.org/wiki/Snake_(video_game_genre)) BAB 2 LANDASAN
 42 TEORI
 43 2.1 Snake GameSnake Game merupakan permainan mengendalikan ular untuk
 44 mendapatkan makanan yang terdapat pada labirin
 45
 46 ERROR 3
 47
 48 Error Code: PS-03
 49 Note: Perhatikan spasi setelah tanda baca.
 50 Excerpt: Contoh singleplayer game Snake adalah Snake pada telepon genggam Nokia
 51 yang dapat dilihat pada Gambar2.1 1dan contoh multiplayer game Snake adalah
 52 Slither.io yang dapat dilihat Gambar2.2 2
 53
 54 ERROR 4
 55
 56 Error Code: PS-03
 57 Note: Perhatikan spasi setelah tanda baca.
 58 Excerpt: Gambar 2.1: Permainan Snake pada telepon genggam NokiaGambar 2.2:
 59 Permainan Slither.iopadaAndroid 1 [https://en.wikipedia.org/wiki/](https://en.wikipedia.org/wiki/Snake_(video_game_genre))
 60 [Snake_\(video_game_genre\)](https://play.google.com/store/apps/details?id=com.hypah.io.slither) 2 [https://play.google.com/store/apps/details?](https://play.google.com/store/apps/details?id=com.hypah.io.slither)
 61 [id=com.hypah.io.slither](https://play.google.com/store/apps/details?id=com.hypah.io.slither) BAB 3 ANALISIS 3.1 Analisis Permainan Snakeyang
 62 Sudah AdaPermainan Snake yang akan dianalisis adalah Slither.io dan Snake
 63 pada telepon genggam Nokia
 64
 65 ERROR 5
 66
 67 Error Code: PS-03
 68 Note: Perhatikan spasi setelah tanda baca.
 69 Excerpt: Slither.io adalah permainan web yang dapat dimainkan oleh lebih dari 1
 70 pemain(multiplayer)
 71
 72 ERROR 6
 73
 74 Error Code: PS-03
 75 Note: Perhatikan spasi setelah tanda baca.
 76 Excerpt: 3.1.1Ular dan Makanan Ular pada Slither.io dibentuk dengan menggunakan
 77 sekumpulan lingkaran yang saling berdempetan satu sama lain seperti pada
 78 Gambar3.1

```

79 |
80 | ERROR 7
81 | =====
82 | Error Code: PS-03
83 | Note: Perhatikan spasi setelah tanda baca.
84 | Excerpt: Makanan pada Slither.io berbentuk lingkaran
85 |
86 | ERROR 8
87 | =====
88 | Error Code: PS-03
89 | Note: Perhatikan spasi setelah tanda baca.
90 | Excerpt: Gambar 3.1: Ular pada Silther.io BAB 4 PERANCANGAN Pada bab ini akan
91 | dibahas mengenai perancangan permainan yang dibangun
92 |
93 | ERROR 9
94 | =====
95 | Error Code: PS-03
96 | Note: Perhatikan spasi setelah tanda baca.
97 | Excerpt: Permainan ini dapat dimainkan pada linkini : https://generaldevilx.
98 | github.io/Snake360/ 5.1 Implementasi Pada bagian ini akan dijelaskan mengenai
99 | lingkungan yang digunakan untuk membangun dan implementasi antarmuka dari Open
100 | Source Snake 360
101 |
102 | ERROR 10
103 | =====
104 | Error Code: KAL-02
105 | Note: Berilah kata pengantar untuk setiap bab
106 | Excerpt: BAB 1 PENDAHULUAN 1.1 Latar Belakang Snake merupakan sebuah permainan
107 | yang pertama kali dibuat oleh Peter Trefonas pada tahun 1978

```

4. TC_PE_04.pdf

Listing 5.17: Pesan Kesalahan Secured PDF

```

1 | Fatal error: Uncaught Exception: Object list not found. Possible secured file.
2 | in D:\Skripsi\src\vendor\smalot\pdfparser\src\Smalot\PdfParser\Parser.php:105
3 | Stack trace:
4 | #0 D:\Skripsi\src\vendor\smalot\pdfparser\src\Smalot\PdfParser\Parser.php(81):
5 | Smalot\PdfParser\Parser->parseContent('%PDF-1.5\n%\xD0\xD4\xC5\xD8\n...')
6 | #1 D:\Skripsi\src\pdf_checker\SkripsiExtract.php(26): Smalot\PdfParser\Parser->
7 | parseFile('D:\Skripsi\src\...')
8 | #2 D:\Skripsi\src\pdf_checker\Main.php(9): SkripsiExtract->extractFromPDF('../
9 | res/TC_PE_04...')
10 | #3 {main}
11 | thrown in D:\Skripsi\src\vendor\smalot\pdfparser\src\Smalot\PdfParser\Parser.
12 | php on line 105

```

Pada kasus uji ini, perangkat lunak mendeteksi bahwa file ini merupakan file yang *secured*. Hal ini sama dengan yang terjadi pada kasus uji TC_PE_01.pdf. Namun masalah ini dapat diatasi dan akan dibahas lebih lanjut pada sub bab 5.3.2. Berikut ini adalah laporan kesalahan yang ditemukan:

Listing 5.18: Laporan Kesalahan Kasus Uji TC_PE_04

```

1 | ERROR 1
2 | =====
3 | Error Code: PS-01
4 | Note: Ditemukan penulisan kata yang tidak sesuai dengan kamus
5 | Excerpt:
6 |
7 | PENDAHULUAN, Belakang Aplikasi, merupakan, sebuah, membantu, penggunaanya,
8 | meninjau, dihasilkan, secara, sebelum, pengguna, tersebut, melakukan,
9 | pembuatan, Kelebihan, adalah, peninjauan, berbagai, memaksimalkan,
10 | memungkinkan, merubah, bertujuan, memutuskan, dasarnya, terhindar,

```

ekspektasi, Penggunaan, web, memudahkan, mela, kukan, menggunakan, berbasis, lingkungan, Windows, Linux, Mac, OS, membatasi, cakupan, dibuat, kustomisasi, mengajar, perkuliahan, Melalui, diharapkan, memiliki, gambaran, mengenai, ruangan, komposisi, memanfaatkan, WebGL, Three, js, Application, Programming, Interface, gra, s, berdasar, OpenGL, terbuka, ECMAScript, melalui, canvas, HTML, membuat, digunakan, sebagai, Ruangan, dilengkapi, peralatan, menunjang, pengajaran, internet, learning, menjamin, kenyamanan, selama, pendingin, Rumusan, Berikut, dibahas, pendukung, lainnya, direpresentasikan, tampilan, OrbitControls, berkaitan, index, html, HyperText, Markup, Language, mem, json, JSON, constant, diinisialisasi, imported, representasi, ujian, berlangsung, img, jpg, pilihan, textureatop, PENGUJIANBab, membahas, pengujian, rancangan, Lingkungan, Pengujian, Sedangkan, spesi, Processor, GHz, Core, Graphics, Processing, GPU, HD, MB, Random, Access, Memory, GB, MHz, DDR, Storage, SSD, macOS, Sierra, Pemrograman, CSS, Text, Code, MAMP, version, Google, Chrome, Ver, sion, O, cial, Build, sebelumnya, Pengimplementasian, pemro, gram, dijabarkan, KESIMPULAN, SARANBab, kesimpulan, pengembangan, Kesimpulan, Berdasarkan, Prantijau, diperoleh, tujuan, merepresentasikan, Parahyangan, Fitur, dikembangkan, Mengganti, Unggah, Object, Notation, mengubah, Menghasilkan, cetakan, dikarenakan, keterbatasan, ngembangan, Mengimplementasi, penyembunyian, menutupi, penyediaan, kemudahan, mengunggah

ERROR 2

Error Code: PS-03

Note: Perhatikan spasi setelah tanda baca.

Excerpt: Perangkat lunak akan dibuat dengan memanfaatkan WebGL dan pustaka Three.js

ERROR 3

Error Code: PS-03

Note: Perhatikan spasi setelah tanda baca.

Excerpt: Sementara itu pustaka Three.js bertujuan membuat pustaka 3 dimensi yang mudah dan ringan untuk digunakan

ERROR 4

Error Code: PS-03

Note: Perhatikan spasi setelah tanda baca.

Excerpt: 1.2Rumusan Masalah Berikut ini masalah-masalah yang dibahas dalam skripsi ini: Bagaimana ruangan kelas dan objek pendukung lainnya dapat direpresentasikan dalam WebGL? Bagaimana membuat tampilan responsif pada aplikasi agar terlihat bagus saat dicetak? BAB 2 LANDASAN TEORIBab ini berisi penjelasan mengenai teori-teori yang menjadi dasar penelitian ini, seperti WebGL dan Three.js library

ERROR 5

Error Code: PS-03

Note: Perhatikan spasi setelah tanda baca.

Excerpt: BAB 3 ANALISISBab ini berisi analisis pemodelan properti kelas, analisis pemanfaatan pustaka Three.js, dan analisis penggunaan WebGL

ERROR 6

Error Code: PS-03

Note: Perhatikan spasi setelah tanda baca.

Excerpt: 3.2 Analisis pemanfaatan pustaka Three.js Pada pengimplementasian Aplikasi Pratinjau 3 Dimensi Berbasis Web digunakan banyak fitur yang telah disediakan oleh pustaka Three.js

74 ERROR 7

75

76 Error Code: PS-03

77 Note: Perhatikan spasi setelah tanda baca.

78 Excerpt: Berikut ini merupakan daftar fitur pustaka Three.js yang digunakan
79 pada implementasi Aplikasi Pratinjau 3 Dimensi Berbasis Web: 3.2.1 Panggung

80 Panggung (scene) merupakan sebuah wadah untuk menempatkan sesuatu pada
81 pustaka Three.js (subbab2.2.1)

82

83 ERROR 8

84

85 Error Code: PS-03

86 Note: Perhatikan spasi setelah tanda baca.

87 Excerpt: 1 <https://www.blender.org/> BAB 4 PERANCANGAN Pada bab ini dibahas
88 mengenai perancangan perangkat lunak yang meliputi: perancangan struk- tur
89 web, perancangan antarmuka, dan perancangan fitur yang diimplementasikan
90 pada Aplikasi Pratinjau 3 Dimensi Berbasis

91

92 ERROR 9

93

94 Error Code: PS-03

95 Note: Perhatikan spasi setelah tanda baca.

96 Excerpt: Berkas yang ada pada folder ini hanya satu yaitu custom.css

97

98 ERROR 10

99

100 Error Code: PS-03

101 Note: Perhatikan spasi setelah tanda baca.

102 Excerpt: Terdapat berbagai file JavaScript di dalam folder ini, daftar
103 berkasnya adalah sebagai berikut: Main.js, berkas JavaScript ini berisi
104 berbagai fungsi utama yang digunakan untuk membangun Aplikasi Pratinjau 3
105 Dimensi Berbasis Web

106

107 ERROR 11

108

109 Error Code: PS-03

110 Note: Perhatikan spasi setelah tanda baca.

111 Excerpt: three.js, berkas JavaScript ini berisi berbagai fungsi yang
112 disediakan oleh pustaka Three.js

113

114 ERROR 12

115

116 Error Code: PS-03

117 Note: Perhatikan spasi setelah tanda baca.

118 Excerpt: Nantinya fungsi yang terdapat di dalam berkas ini akan dipanggil oleh
119 Main.js OrbitControls.js, berkas JavaScript ini berisi berbagai fungsi yang
120 juga disediakan oleh pustaka Three.js

121

122 ERROR 13

123

124 Error Code: PS-03

125 Note: Perhatikan spasi setelah tanda baca.

126 Excerpt: berkas index.html, merupakan berkas HyperText Markup Language (HTML)
127 yang membentuk web untuk aplikasi pratinjau ini

128

129 ERROR 14

130

131 Error Code: PS-03

132 Note: Perhatikan spasi setelah tanda baca.

133 Excerpt: Terdapat 2 berkas json di dalam folder ini yaitu sebagai berikut:
134 constant.json, berkas ini berisi berbagai informasi awal untuk
135 diinisialisasi ke aplikasi sehingga dapat menampilkan gambaran awal dari
136 ruangan kelas Fakultas Teknologi Informasi dan Sains

ERROR 15

Error Code: PS-03

Note: Perhatikan spasi setelah tanda baca.

Excerpt: imported.json , berkas ini berisi informasi untuk representasi ruangan kelas saat ujian sedang berlangsung di Fakultas Teknologi Informasi dan Sains

ERROR 16

Error Code: PS-03

Note: Perhatikan spasi setelah tanda baca.

Excerpt: Berikut ini merupakan daftar berkas yang ada pada folder ini: textureatap.jpg , merupakan tekstur untuk bagian atap ruangan kelas

ERROR 17

Error Code: PS-05

Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital

Excerpt: typedef unsigned long GLenum ; typedef boolean ; typedef unsigned long GLinti field ; typedef byte GLbyte ; typedef short GLshort ; typedef long GLint ; typedef long GLsizei ; typedef long GLintptr ; typedef long GLsizeiptr ; typedef octet GLubyte ; typedef unsigned short GLushort ; typedef unsigned long GLuint ; typedef unsigned restricted float GLfloat ; Listing 2.1: Alias untuk tipe pada WebGL

ERROR 18

Error Code: PS-05

Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital

Excerpt: Berkas yang ada pada folder ini hanya satu yaitu custom.css

ERROR 19

Error Code: PS-09

Note: Bab/Subbab 4.2.1 ini hanya terdapat 1 sub bab/sub sub bab

ERROR 20

Error Code: KAL-02

Note: Berilah kata pengantar untuk setiap bab

Excerpt: BAB 1 PENDAHULUAN 1.1 Latar Belakang Aplikasi pratinjau 3 dimensi merupakan sebuah perangkat lunak yang membantu pengguna untuk meninjau kembali desain dari produk yang ingin dihasilkan secara 3 dimensi, sebelum pengguna tersebut melakukan implementasi pembuatan produk

ERROR 21

Error Code: KAL-03

Note: Ada data skripsi yang belum dilengkapi pada halaman cover

Excerpt: SKRIPSI APLIKASI PRATINJAU 3 DIMENSI BERBASIS WEB Nancy Valentina NPM: 2014730049 PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI DAN SAINS UNIVERSITAS KATOLIK PARAHYANGAN tahun UNDERGRADUATE THESIS JUDUL BAHASA INGGRIS Nancy Valentina NPM: 2014730049 DEPARTMENT OF INFORMATICS FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES PARAHYANGAN CATHOLIC UNIVERSITY tahun

ERROR 22

```

200 | Error Code: NAT-01
201 | Note: Referensi tidak dirujuk dengan baik, lakukan perintah PDFLatex->BibTex->
202 | PDFLatex->PDFLatex untuk memperbaikinya
203 | Excerpt: Canvas pada HTML menyediakan suatu destinasi untuk pembangunan objek
204 | secara programatik pada halaman web dan memungkinkan menampilkan objek yang
205 | sedang dibangun menggunakan API pembangun objek yang berbeda [ ?]

```

5. TC_PE_05.pdf

Perangkat lunak tidak mengeluarkan laporan kesalahan yang ditemukan pada dokumen. Pada terminal juga tidak diberikan pesan error yang menandakan bahwa terdapat kesalahan pada perangkat lunak. Namun masalah ini dapat diatasi dan akan dibahas lebih lanjut pada sub bab 5.3.1. Berikut ini adalah laporan kesalahan yang ditemukan:

Listing 5.19: Laporan Kesalahan Kasus Uji TC_PE_05

```

1 | ERROR 1
2 | =====
3 | Error Code: PS-01
4 | Note: Ditemukan penulisan kata yang tidak sesuai dengan kamus
5 | Excerpt:
6 |
7 | PENDAHULUAN, BelakangOnline, judge, merupakan, sebuah, online, dirancang,
8 | mengevaluasi, algoritma, dikumpulkan, pengguna, mengcompile, mengeksekusi,
9 | mengujinya, lingkungan, menggunakan, disediakan, dijalankan, batasan,
10 | time, memory, keamanan, sebagainya, Penggunaan, ditemukan, pemrograman,
11 | Sharif, Judge, adalah, C, Java, Python, diciptakan, Mohammad, Javad,
12 | Naderi, bersifat, source, ditulis, PHP, framework, CodeIgniter,
13 | danbackend, BASH, menilai, jawaban, peserta, secara, penilaian, dimulai,
14 | membuat, assignment, dibutuhkan, mulai, berhenti, tambahan, mengunduh,
15 | mengerjakan, tersebut, Peserta, mengumpulkan, menjalankan, membandingkan,
16 | Setelah, menyesuaikan, keluaran, digunakan, Parahyangan, Algoritma,
17 | membantu, memanfaatkan, memberikan, ujian, melihat, setelah, memperbaiki,
18 | dikerjakan, melewati, pengumpulan, prakteknya, terkini, pengembangan,
19 | Pengembangan, memiliki, kebutuhan, spesi, k, Kebutuhan, login,
20 | terintegrasi, server, membatasi, pengaksesan, lainnya, terakhir, dicommit,
21 | GitHub, Juli, bug, diperbaiki, menyebabkan, memenuhi, dikembnangkan,
22 | sebutkan, disebutkan, Flow, Chart, menunjukan, mengalir, ANALISISBab,
23 | ditawarkan, didapat, issue, repository, dicatat, Google, Sheets,
24 | didiskusikan, pembimbing, dilakukan, menganalisis, setiap, terbuka,
25 | terdapat, diikuti, urutan, dibuat, analisa, didapatkan, pertanyaan,
26 | usulan, Bebera, dijadikan, pertimbangan, melalui, diwawancarai, Husnul,
27 | Claudio, Franciscus, Vania, Natali, Luciana, Abednego, Joanna, Helga,
28 | PERANCANGANBab, perancangan, diimplementasi, Mengganti, Method, shel, l,
29 | exec, rm, Menjadiunlink, menghapus, le, control, Assignment, diubah,
30 | method, unlink, Listing, perubahan, syntax, Perubahan,
31 | classAssignmentextendsCI, Controller, Upload, Tests, zip, file, shell, f,
32 | assignments, root, config, array, upload, path, allowed, types, else,
33 | foreach, old, pdf, files, name, this, messages, type, success, text, PDF,
34 | uploaded, successfully, Chrome, Version, O, cial, Build, Firefox, Quantum,
35 | Microsoft, Excel, Kedua, KESIMPULAN, SARANBab, kesimpulan, selanjutnya,
36 | Kesimpulan, Berdasarkan, diperoleh, sebagai, berikut, dikembangkan,
37 | mengimplementasi, dilihat, padaTabel, Membatasi, Mensupport, ledengan,
38 | TXT, Display, Name, pendaftaran, Lock, Student, Archived, Logs, Fame,
39 | loginkeserver, Branding, Mengubah, PHPExcel, menjadilibrary,
40 | PHPSpreadsheet, menjadiunlink, methodrekoneksi, berhasil, ditentukan,
41 | Fitur, JAR, dikarenakan, keterbatasan, pembaruan, Mengimplementasi, multi,
42 | class, masukan, padasub, Round, Robin, menggunakanCactiuntuk
43 |
44 | ERROR 2
45 | =====
46 | Error Code: PS-03
47 | Note: Perhatikan spasi setelah tanda baca.
48 | Excerpt: Contoh perbandingan URL biasa dengan Clean URL s: URL biasa: \example.

```

```

49 | com\index.php?page=news ,Clean URLs :\ example.com ews
50 |
51 | ERROR 3
52 | =====
53 | Error Code: PS-03
54 | Note: Perhatikan spasi setelah tanda baca.
55 | Excerpt: 4.1Mengganti Method shell_exec("rm ...")Menjadiunlink() Method shell
56 | l_exec("rm ...") yang memiliki fungsi untuk menghapus sebuah file terdapat
57 | pada kelas control ler Assignment.php
58 |
59 | ERROR 4
60 | =====
61 | Error Code: PS-03
62 | Note: Perhatikan spasi setelah tanda baca.
63 | Excerpt: Listing 4.1: Perubahan kode program pada Assignment.php@@-433,8+433,7
64 | @@classAssignmentsextendsCI_Controller // Upload Tests (zip file) -
65 | shell_exec('rm-f' . $assignments_root . '/*.zip');
66 | +unlink($assignments_root . '/*.zip'); $config = array( 'upload_path' =>
67 | $assignments_root , 'allowed_types' => 'zip',
68 | @@-482,7+481,7@@classAssignmentsextendsCI_Controller else {
69 | foreach($old_pdf_files as $old_name) -shell_exec("rm-f$old_name");
70 | +unlink($old_name); $this->messages[] = array( 'type' => 'success', 'text'
71 | => 'PDF file uploaded successfully.' 4.2Menambahkan Method Rekoneksi ke
72 | Database Method rekoneksi ke database ditambahkan pada kelas control ler
73 | Queueprocess.php .Method yang digunakan yaitu $this->db->reconnect()
74 |
75 | ERROR 5
76 | =====
77 | Error Code: PS-03
78 | Note: Perhatikan spasi setelah tanda baca.
79 | Excerpt: Listing4.2menunjukan perubahan kode program yang dilakukan
80 | diQueueprocess.php Listing 4.2: Perubahan kode program pada Queueprocess.
81 | php1 GNU Operating System, "Comparing and Merging Files" terakhir diubah 6
82 | Mei 2017
83 |
84 | ERROR 6
85 | =====
86 | Error Code: PS-03
87 | Note: Perhatikan spasi setelah tanda baca.
88 | Excerpt: https://www.gnu.org/software/diutils/manual/diutils.html#Detailed-
89 | Unified BAB 5 IMPLEMENTASI DAN PENGUJIANBab ini membahas tentang
90 | implementasi dan pengujian perangkat lunak berdasarkan rancangan yang sudah
91 | dibuat
92 |
93 | ERROR 7
94 | =====
95 | Error Code: PS-05
96 | Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
97 | Excerpt: Contoh perbandingan URL biasa dengan Clean URL s: URL biasa: \example.
98 | com\index.php?page=news ,Clean URLs :\ example.com ews
99 |
100 | ERROR 8
101 | =====
102 | Error Code: PS-09
103 | Note: Bab/Subbab 3.1 ini hanya terdapat 1 sub bab/sub sub bab
104 |
105 | ERROR 9
106 | =====
107 | Error Code: KAL-02
108 | Note: Berilah kata pengantar untuk setiap bab
109 | Excerpt: BAB 1 PENDAHULUAN 1.1Latar BelakangOnline judge merupakan sebuah
110 | sistem online yang dirancang untuk mengevaluasi kode algoritma yang
111 | dikumpulkan oleh pengguna

```

5.3.1 Perangkat Lunak Tidak Mengeluarkan Laporan Kesalahan

Hipotesis yang didapatkan dari laporan yang dikeluarkan saat perangkat lunak menggunakan kasus uji TC_PF_02.pdf, TC_PF_03.pdf dan TC_PF_05.pdf adalah ada kesalahan yang perlu ditelusuri pada *library PdfParser*.

1. Pertama melakukan pengecekan terhadap method `extractFromPDF($input)` yang ada pada kelas `SkripsiExtract`. Langkah yang dilakukan untuk mencari kesalahan tersebut adalah dengan menambahkan kode print start dan stop di dalam method tersebut pada awal dan akhir sebelum *return value*. Kode print stop akan bergeser ke atas hingga bertemu dengan sumber masalah.

Listing 5.20: Potongan kode pada *method extractFromPDF*

```

1 | $directory = getcwd();
2 | $this->parser = new \Smalot\PdfParser\Parser();
3 | $this->file = $directory . '/' . $input;
4 | echo "start";
5 | $pdf = $this->parser->parseFile($this->file);
6 | echo "stop";
7 | $pages = $pdf->getPages();

```

Pada saat perangkat lunak dijalankan, hasil yang dikeluarkan hanya tulisan *start* saja. Pada listing 5.20, sumber masalah ditemukan pada baris ke-5 (baris ke-26 pada *source code*). Setelah sumber ditemukan, masalah yang ditemui berasal dari *library PdfParser*.

2. Langkah ke-2 yaitu menelusuri kelas `Parser` dari *library PdfParser*. Langkah yang dilakukan sama dengan langkah yang pertama. Pada kelas tersebut sumber masalah berada pada *method parseContent()*.

Listing 5.21: Potongan kode pada *method parseContent*

```

1 | ob_start();
2 | echo "start";
3 | @$parser = new \TCPDF_PARSER(trim($content));
4 | echo "stop";
5 | list($xref, $data) = $parser->getParsedData();
6 | unset($parser);
7 | ob_end_clean();

```

Pada listing 5.21, sumber masalah berada pada baris ke-3 (baris ke-93 pada *source code*). Baris tersebut merujuk pada kelas `TCPDF_Parser`.

3. Langkah ke-3 yaitu menelusuri kelas `TCPDF_Parser`. Pada kelas ini terdapat beberapa *method*, sehingga *method* yang pertama kali dicek adalah konstruktornya.

Listing 5.22: Potongan kode pada konstruktor kelas `TCPDF_Parser`

```

1 | foreach ($this->xref['xref'] as $obj => $offset) {
2 |     if (!isset($this->objects[$obj]) AND ($offset > 0)) {
3 |         // decode objects with positive offset
4 |         echo "start";
5 |         $this->objects[$obj] = $this->getIndirectObject($obj, $offset, true);
6 |         echo "stop";
7 |     }
8 | }

```

Pada listing 5.22, sumber masalah berada pada baris ke-5 (baris ke-123 pada *source code*). Baris tersebut merujuk pada *method getIndirectObject()* yang ada pada kelas yang sama.

Listing 5.23: Potongan kode pada *method getIndirectObject()*

```

1 | if ($decoding AND ($element[0] == 'stream') AND (isset($objdata[( $i - 1)][0]))
   |   AND ($objdata[( $i - 1)][0] == '<<')) {
2 |     echo "start";
3 |     $element[3] = $this->decodeStream($objdata[( $i - 1)][1], $element[1]);
4 |     echo "stop";
5 | }

```

Pada listing 5.23, sumber masalah berada pada baris ke-4 (baris ke-702 pada *source code*). Baris tersebut merujuk pada *method decodeStream()* yang ada pada kelas yang sama.

Listing 5.24: Potongan kode pada *method decodeStream()*

```

1 | try {
2 |     echo "start";
3 |     $stream = TCPDF_FILTERS::decodeFilter($filter, $stream);
4 |     echo "stop";
5 | } catch (Exception $e) {
6 |     $emsg = $e->getMessage();
7 |     if ((( $emsg[0] == '~') AND !$this->cfg['ignore_missing_filter_decoders']) OR
   |       (( $emsg[0] != '~') AND !$this->cfg['ignore_filter_decoding_errors'])) {
8 |         $this->Error($e->getMessage());
9 |     }
10 | }

```

Pada listing 5.24, sumber masalah berada pada baris ke-3 (baris ke-781 pada *source code*). Baris tersebut merujuk pada kelas `TCPDF_Filters`.

- Langkah terakhir yaitu melakukan pengecekan pada kelas `TCPDF_Filters`. Pada kelas ini, *method* yang diperiksa adalah *decodeFilter()*.

Listing 5.25: Potongan kode pada *method decodeFilter()*

```

1 | case 'FlateDecode': {
2 |     return self::decodeFilterFlateDecode($data);
3 |     break;
4 | }

```

Pada listing 5.25, sumber masalah ada pada baris ke-2 (baris ke-94 pada *source code*). Baris tersebut merujuk pada *method decodeFilterFlateCode()*.

Listing 5.26: Potongan kode pada *method decodeFilterFlateCode()*

```

1 | // initialize string to return
2 | echo "start";
3 | $decoded = @gzuncompress($data);
4 | echo "stop";
5 | if ($decoded == false) {
6 |     self::Error('decodeFilterFlateDecode: invalid code');
7 | }
8 | return $decoded;

```

Pada listing 5.26, sumber dari masalah utamanya berada pada baris ke-2 (baris ke-357 pada *source code*). Dari hasil pengujian yang dilakukan, alasan bahwa perangkat lunak tidak mengeluarkan pesan kesalahan karena pada baris ke-2, terdapat karakter "@" sebelum *method gzuncompress()*. Fungsi karakter tersebut adalah untuk membungkam seluruh pesan kesalahan, sehingga pengguna tidak mengetahui kesalahan yang ada. Untuk melihat pesan kesalahan, karakter "@" dihapuskan dan menambahkan sebuah kode untuk melaporkan kesalahan.

Listing 5.27: Potongan kode pada *method decodeFilterFlateCode()*

```

1 | // initialize string to return
2 | error_reporting(E_ALL);

```

```

3 | $decoded = gzuncompress($data);
4 | if ($decoded == false) {
5 |     self::Error('decodeFilterFlateDecode: invalid code');
6 | }
7 | return $decoded;

```

Dengan memakai kode pada baris ke-2 di listing 5.28, perangkat lunak berhasil mengeluarkan laporan kesalahan.

Listing 5.28: Pesan kesalahan memory limit

```

1 | Fatal error: Allowed memory size of 134217728 bytes exhausted (tried to
   | allocate .....(size tergantung file) bytes) in D:\Skripsi\src\vendor\
   | tecnickcom\tcpdf\include\tcpdf_filters.php on line 357

```

5. Untuk mengatasi masalah yang ada pada listing 5.28, maka ada beberapa hal yang perlu ditambahkan pada perintah 5.1 untuk menjalankan perangkat lunak.

Listing 5.29: Perintah untuk mengatasi masalah *memory limit*

```

1 | php -d memory_limit=(ukuran sesuai kebutuhan)M main.php ../res/
   |     nama_file.pdf

```

Dengan menggunakan perintah pada 5.29, perangkat lunak dapat mengeluarkan pesan kesalahan dalam dokumen skripsi. Proses untuk menyelesaikan masalah, bahwa perangkat lunak tidak mengeluarkan pesan pun telah berhasil diselesaikan.

5.3.2 Perangkat Lunak Tidak Dapat Memeriksa Dokumen Skripsi

Pada beberapa kasus uji, perangkat lunak mengeluarkan kesalahan yang mengatakan bahwa file yang digunakan termasuk *secured document*. File yang terkait yaitu "TC_PE_01.pdf" dan "TC_PE_04.pdf". Hipotesis yang didapatkan dari laporan adalah kemungkinan bahwa dokumen yang diunggah ke *Github* akan menjadi *secured document*.

Berdasarkan hipotesis yang telah disampaikan, ada sebuah percobaan yang dilakukan pada kasus uji yang terkait. Hal yang dilakukan adalah mengunduh seluruh data skripsi dari file yang terkait. Setelah itu dilakukan proses *compile* ulang, untuk mendapatkan PDF dokumen skripsi yang baru. Dokumen hasil *compile* ulang diberi nama TC_PE_01_v2.pdf dan TC_PE_04_v2.pdf.

Pengujian file yang sebelumnya bermasalah dilakukan dengan menggunakan file hasil *compile* ulang. Perangkat lunak berhasil mengeluarkan laporan kesalahan yang ada pada dokumen skripsi. Permasalahan yang berkaitan dengan *secured PDF* berhasil diatasi.

5.3.3 Kesimpulan Pengujian Eksperimental

Pengujian eksperimental telah dilakukan dengan menggunakan 5 kasus uji, berupa dokumen skripsi milik mahasiswa Informatika UNPAR. Pada pengujian ini dapat ditarik beberapa kesimpulan, yaitu sebagai berikut:

- Dokumen PDF skripsi yang ada pada *Github* FTIS UNPAR, tidak dapat digunakan sebagai kasus uji. Ada faktor bahwa file dokumen PDF yang telah diunggah pada *Github* akan menjadi *secured document*. *Library PdfParser* tidak dapat mengeskrak dokumen tersebut. Untuk dapat mengatasi hal ini, pengguna perangkat lunak memerlukan file latex dari dokumen tersebut, lalu dicompile ulang. Hal ini sudah dibuktikan pada pengujian eksperimental, dan perangkat lunak dapat mengeluarkan laporan kesalahan dari dokumen tersebut.
- Kasus dimana perangkat lunak tidak mengeluarkan laporan kesalahan, terjadi karena adanya *memory limit*. *Memory limit* yang diatur pada konfigurasi *php.ini* adalah 128Mb. Masalah ini dapat diselesaikan dengan menambahkan memori seperti yang sudah dilakukan pada pengujian eksperimental.

- Dari pengujian ini, dapat dilihat bahwa ada sebuah *library Pdf Parser* yang tidak ditangani dan kesalahan tersebut tidak dimunculkan pada *exception*. Pada pengujian ini telah dijelaskan bahwa pesan kesalahan dari sistem disembunyikan dengan menggunakan tanda "@".

BAB 6

KESIMPULAN DAN SARAN

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

6.1 Kesimpulan

Dari pengujian yang telah dilakukan, didapatkan kesimpulan-kesimpulan sebagai berikut:

1. Kesalahan dalam dokumen skripsi diperiksa menggunakan pencocokan pola, dengan menggunakan *regular expression*. Kesalahan yang dapat diperiksa pada perangkat lunak adalah kesalahan yang bersifat tekstual. Pola *regular expression* tidak dapat memeriksa kesalahan yang bersifat kontekstual.
2. Perangkat lunak pemeriksa kesalahan dokumen skripsi berhasil dikembangkan dengan menggunakan bahasa pemrograman *PHP*. Perangkat lunak menerapkan prinsip paradigma *Object Oriented Programming*. Dalam pengembangannya, dibutuhkan sebuah *library PdFParser* yang digunakan untuk melakukan ekstrak dokumen skripsi. Perangkat lunak akan menerima input nama file dokumen skripsi yang akan diperiksa, dan memberikan output berupa laporan kesalahan yang ditemukan.
3. Hasil pengujian yang diperoleh, secara umum sudah memberikan hasil yang sesuai dengan yang diharapkan. Setiap fitur yang ada pada perangkat lunak dapat mencari dan melaporkan kesalahan yang ditemukan.

6.2 Saran

Dari pengujian yang telah dilakukan, terdapat beberapa saran yang dapat digunakan sebagai pengembangan perangkat lunak:

1. Menambahkan nomor halaman yang menjadi tempat ditemukannya kesalahan pada setiap fitur yang ada pada perangkat lunak.
2. Mengembangkan fitur pemeriksa kesalahan kata, agar dapat memeriksa kata-kata yang menggunakan imbuhan.
3. Membuat kamus yang dinamis untuk fitur pemeriksa kesalahan kata, agar dapat ditambahkan sesuai dengan kebutuhan.
4. Mencari alternatif *library PdFParser*, karena *library* ini masih ada beberapa kekurangan.
5. Menambahkan fitur-fitur yang belum ada pada perangkat lunak ini.

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.
- [2] PHP Perl compatible regular expression. <https://www.php.net/manual/en/book.pcre.php>. 24 Juli 2019.
- [3] Malot, S. Pdf parser. <https://www.pdfparser.org/>. 7 Mei 2019.
- [4] LibreOffice (2018) Getting started with libreoffice 6.0. <https://documentation.libreoffice.org/en/english-documentation/writer/>. 24 Juli 2019.
- [5] Herianto, C. D. (2019) Pengelompokan dokumen berbasis algoritma genetika. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [6] Vallian, S. (2018) Kustomisasi sharif judge untuk kebutuhan program studi teknik informatika. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [7] Valentina, N. (2018) Aplikasi pratinjau 3 dimensi berbasis web. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [8] Wijaya, E. (2019) Pembangunan gim snake 360 berbasis web dengan kode terbuka. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [9] Angelica, E. (2019) Kolektor pengumuman informatika. Skripsi. Universitas Katolik Parahyangan, Indonesia.

LAMPIRAN A

KODE PROGRAM

Listing A.1: SkripsiExtract.php

```
1 <?php
2
3 /*
4  * Kelas untuk menjalankan checker
5  * Author Marcell Trixie
6  */
7 class SkripsiExtract{
8
9     protected $file;
10    protected $parser;
11    protected $cover;
12    protected $abstractPage;
13    protected $otherPage;
14    protected $tableOfContentPage;
15    protected $listPage;
16    protected $contentPage;
17
18    /**
19     * Method untuk mengekstrak file PDF skripsi
20     * @param input file pdf yang akan diekstrak
21     */
22    public function extractFromPDF($input){
23        $directory = getcwd();
24        $this->parser = new \Smalot\PdfParser\Parser();
25        $this->file = $directory . '/' . $input;
26        $pdf = $this->parser->parseFile($this->file);
27        $pages = $pdf->getPages();
28
29        foreach ($pages as $page => $value) {
30            if($page == 0 || $page == 2){
31                $this->cover .= preg_replace('/\s+/', ' ', $value->getText());
32            }
33            else if (preg_match("/\bLEMBAR_PENGESAHAN\b|\bPERNYATAAN\b|\bKATA_PENGANTAR\b/", $value->getText())) {
34                $this->otherPage .= preg_replace('/\s+/', ' ', $value->getText());
35            }
36            else if (preg_match("/\bABSTRAK\b|\bABSTRACT\b/", $value->getText())) {
37                $this->abstractPage .= preg_replace('/\s+/', ' ', $value->getText());
38            }
39            else if (preg_match("/\bDAFTAR_ISI\b/", $value->getText())) {
40                $this->tableOfContentPage .= preg_replace('/\s+/', ' ', $value->getText());
41            }
42            else if (preg_match("/\bDAFTAR_GAMBAR\b|\bDAFTAR_TABEL\b|\bDAFTAR_REFERENSI\b|\bLAMPIRAN\b/", $value->getText()))
43            {
44                $this->listPage .= preg_replace('/\s+/', ' ', $value->getText());
45            }
46            else if (preg_match("/(BAB_)[0-9]{1}/", $value->getText())) {
47                $temp= trim(preg_replace('/\s+/', ' ', $value->getText()));
48                $this->contentPage .= preg_replace("/[0-9]*$/", "", $temp);
49            }
50            else{
51                continue;
52            }
53        }
54
55        /**
56         * Method untuk mendapatkan daftar isi
57         * @return nomor bab dan subbab yang ada pada halaman daftar isi
58         */
59        public function getTableOfContentPage(){
60            $temp = preg_replace('/[A-Za-z]/', '', $this->tableOfContentPage);
61            $array = explode("\n", $temp);
62            for ($i=0 ; $i < sizeof($array) ; $i++ ) {
63                if(preg_match('/^[0-9]{1,}$/', $array[$i])){
64                    $array[$i] = "";
65                }
66                if(preg_match('/^.\$/ ', $array[$i])){
67                    $array[$i] = "";
68                }
69            }
70
71            foreach ($array as $key => $value) {
72                if(empty($array[$key])){
73                    unset($array[$key]);
74                }
75            }
76        }
77    }
78 }
```

```

75         }
76         return $array;
77     }
78
79     /**
80      * Method untuk mendapatkan halaman konten dari bab 1 sampai bab 6
81      * @return array yang sudah diexplode
82      */
83     public function splitContentPage() {
84         $temp = explode(".", $this->contentPage);
85         foreach ($temp as $index => $value) {
86             preg_replace('/[0-9]$/', "", $value);
87         }
88         return $temp;
89     }
90
91     public function getCoverPage() {
92         return $this->cover;
93     }
94
95     public function getAbstractPage() {
96         return $this->abstractPage;
97     }
98
99     public function getListPage() {
100         return $this->listPage;
101     }
102
103     public function getContentPage() {
104         return $this->contentPage;
105     }
106 }
107
108
109 ?>

```

Listing A.2: Checker.php

```

1 <?php
2
3 include "checkers/PS01_TypoChecker.php";
4 include "checkers/PS03_SpaceChecker.php";
5 include "checkers/PS05_CapitalLetterChecker.php";
6 include "checkers/PS09_SubChapterChecker.php";
7 include "checkers/KAL02_PrefaceChecker.php";
8 include "checkers/KAL03_ThesisDataChecker.php";
9 include "checkers/NAT01_ReferenceChecker.php";
10 include "checkers/VAN03_SubjectPronounChecker.php";
11
12 /**
13  * Kelas Parent untuk seluruh checker
14  * @author Marcell Trixie
15  */
16 abstract class Checker{
17
18     /**
19      * Method untuk memeriksa kesalahan dalam dokumen skripsi
20      * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
21      */
22     public abstract function errorChecking($pdf_extract);
23
24     /**
25      * Mendapatkan seluruh checkers yang tersedia
26      * @return Checkers[] seluruh checker yang bisa dipakai
27      */
28     public static function getAllCheckers() {
29         return [
30             new PS01_TypoChecker(),
31             new PS03_SpaceChecker(),
32             new PS05_CapitalLetterChecker(),
33             new PS09_SubChapterChecker(),
34             new KAL02_PrefaceChecker(),
35             new KAL03_ThesisDataChecker(),
36             new NAT01_ReferenceChecker(),
37             new VAN03_SubjectPronounChecker()
38         ];
39     }
40 }
41
42
43 ?>

```

Listing A.3: Main.php

```

1 <?php
2
3 include "../vendor/autoload.php";
4 require "Checker.php";
5 include "SkripsiExtract.php";
6
7 $file = $argv[1];
8 $skripsi_extract = new SkripsiExtract();
9 $skripsi_extract->extractFromPDF($file);
10
11 $checkers = Checker::getAllCheckers();
12 $all_errors = [];
13

```

```

14 |         foreach ($checkers as $checker) {
15 |             $all_errors = array_merge($all_errors, $checker->errorChecking($skripsi->extract));
16 |         }
17 |
18 |         $counter = 1;
19 |         foreach ($all_errors as $error) {
20 |             echo "\nERROR_$counter\n";
21 |             echo "=====\n";
22 |             $counter++;
23 |             foreach ($error as $key => $value) {
24 |                 echo "$key: $value\n";
25 |             }
26 |             echo "\n";
27 |         }
28 |
29 | ?>

```

Listing A.4: KAL02_PrefaceChecker.php

```

1 | <?php
2 |
3 | include_once "Checker.php";
4 |
5 | /**
6 |  * Sub class dari kelas Checker
7 |  * Memeriksa kata pengantar sebelum subbab
8 |  * @author Marcell Trixie
9 |  */
10 | class KAL02_PrefaceChecker extends Checker{
11 |
12 |     /**
13 |      * Method untuk memeriksa kesalahan dalam dokumen skripsi
14 |      * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
15 |      * @return result[] laporan kesalahan yang ditemukan
16 |      */
17 |     public function errorChecking($pdf_extract){
18 |         $result = [];
19 |         $sentence = $pdf_extract->splitContentPage();
20 |         foreach ($sentence as $index => $value) {
21 |             $pattern = "/^(BAB_)[0-9]{1}[\sA-Z]{1,}[0-9]+/";
22 |             if (preg_match($pattern, $value)) {
23 |                 $result[] = [
24 |                     "Error_Code" => "KAL-02",
25 |                     "Note" => "Berilah kata pengantar untuk setiap bab",
26 |                     "Excerpt" => $value
27 |                 ];
28 |             }
29 |         }
30 |         return $result;
31 |     }
32 | }
33 |
34 |
35 | ?>

```

Listing A.5: KAL03_ThesisDataChecker.php

```

1 | <?php
2 |
3 | /**
4 |  * Sub class dari kelas Checker
5 |  * Memeriksa kelengkapan data skripsi
6 |  * @author Marcell Trixie
7 |  */
8 | class KAL03_ThesisDataChecker extends Checker{
9 |
10 |     /**
11 |      * Method untuk memeriksa kesalahan dalam dokumen skripsi
12 |      * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
13 |      * @return result[] laporan kesalahan yang ditemukan
14 |      */
15 |     public function errorChecking($pdf_extract){
16 |         $result = [];
17 |         $cover = $pdf_extract->getCoverPage();
18 |         $pattern = "/\bJUDUL_BAHASA_INDONESIA\b\bJUDUL_BAHASA_INGGRIS\b\bNama_Lengkap\b\b10_digit_NPM_UNPAR\b\btahun\b/";
19 |         if (preg_match($pattern, $cover)) {
20 |             $result[] = [
21 |                 "Error_Code" => "KAL-03",
22 |                 "Note" => "Ada data skripsi yang belum dilengkapi pada halaman cover",
23 |                 "Excerpt" => $cover
24 |             ];
25 |         }
26 |         return $result;
27 |     }
28 | }
29 |
30 |
31 | ?>

```

Listing A.6: NAT01_ReferenceChecker.php

```

1 | <?php
2 |
3 | /**
4 |  * Sub class dari kelas Checker

```

```

5      * Memeriksa subbab dalam sebuah bab
6      * @author Marcell Trixie
7      */
8      class NAT01_ReferenceChecker extends Checker{
9
10         /**
11          * Method untuk memeriksa kesalahan dalam dokumen skripsi
12          * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
13          * @return result[] laporan kesalahan yang ditemukan
14          */
15         public function errorChecking($pdf_extract){
16             $result = [];
17             $sentence = $pdf_extract->splitContentPage();
18             foreach ($sentence as $index => $value) {
19                 $pattern = "/\B\[_\?\]\B/";
20                 if (preg_match($pattern, $value)) {
21                     $result[] = [
22                         "Error_Code" => "NAT-01",
23                         "Note" => "Referensi_tidak_dirujuk_dengan_baik,_lakukan_perintah_PDFLatex->BibTex->PDFLatex->PDFLatex_
                                untuk_memperbaikinya",
24                         "Excerpt" => $value
25                     ];
26                 }
27             }
28             return $result;
29         }
30     }
31 }
32 }
33 }
34 ?>

```

Listing A.7: PS01_TypoChecker.php

```

1 <?php
2
3 /*
4  * Sub class dari kelas Checker
5  * Memeriksa kesalahan penulisan kata
6  * @author Marcell Trixie
7  */
8  class PS01_TypoChecker extends Checker{
9
10     /**
11      * Method untuk memeriksa kesalahan dalam dokumen skripsi
12      * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
13      * @return result[] laporan kesalahan yang ditemukan
14      */
15     public function errorChecking($pdf_extract){
16         $filename = __DIR__ . "/../dictionary/id_id/id_ID.dic";
17         $file = fopen($filename, "r");
18         $temp = explode("\n", fread($file, filesize($filename)));
19
20         $dictionary = [];
21         for ($i = 0; $i < sizeof($temp); $i++) {
22             $dictionary[$i] = preg_replace('/\./+$/',' ', $temp[$i]);
23         }
24
25         $word = preg_split("/[^\A-Za-z]/", $pdf_extract->getContentPage());
26         $typos = [];
27         foreach ($word as $index => $value) {
28             if (!in_array(strtolower($value), $dictionary) && !in_array($value, $typos)) {
29                 $typos[] = $value;
30             }
31         }
32
33         $result = [];
34         if (sizeof($typos) > 0) {
35             $result[] = [
36                 "Error_Code" => "PS-01",
37                 "Note" => "Ditemukan_penulisan_kata_yang_tidak_sesuai_dengan_kamus",
38                 "Excerpt" => "\n\n" . implode(", ", $typos)
39             ];
40         }
41         return $result;
42     }
43 }
44 }
45 }
46 ?>

```

Listing A.8: PS03_SpaceChecker.php

```

1 <?php
2
3 /*
4  * Sub class dari kelas Checker
5  * Memeriksa penggunaan spasi setelah tanda baca
6  * @author Marcell Trixie
7  */
8  class PS03_SpaceChecker extends Checker{
9
10     /**
11      * Method untuk memeriksa kesalahan dalam dokumen skripsi
12      * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
13      * @return result[] laporan kesalahan yang ditemukan
14      */

```

```

15     public function errorChecking($pdf_extract){
16         $result = [];
17         $sentence = $pdf_extract->splitContentPage();
18         foreach ($sentence as $index => $value) {
19             $pattern = "[A-Za-z]{4,}[.,!?!?][A-Za-z]+/";
20             if (preg_match($pattern, $value)) {
21                 $result[] = [
22                     "Error_Code" => "PS-03",
23                     "Note" => "Perhatikan_spasi_setelah_tanda_baca.",
24                     "Excerpt" => $value
25                 ];
26             }
27         }
28         return $result;
29     }
30 }
31 }
32 }
33 ?>

```

Listing A.9: PS05_CapitalLetterChecker.php

```

1 <?php
2
3 /*
4  * Sub class dari kelas Checker
5  * Memeriksa penggunaan huruf kapital pada karakter awal kalimat
6  * @author Marcell Trixie
7  */
8 class PS05_CapitalLetterChecker extends Checker{
9
10     /**
11      * Method untuk memeriksa kesalahan dalam dokumen skripsi
12      * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
13      * @return result[] laporan kesalahan yang ditemukan
14      */
15     public function errorChecking($pdf_extract){
16         $result = [];
17         $sentence = $pdf_extract->splitContentPage();
18         foreach ($sentence as $index => $value) {
19             $pattern = "/^[a-z]+_[a-z]+\.[a-z]/";
20             if (preg_match($pattern, $value)) {
21                 if(preg_match("/^W/", $value)){
22                     continue;
23                 }
24                 else{
25                     $result[] = [
26                         "Error_Code" => "PS-05",
27                         "Note" => "Huruf_pertama_pada_kalimat_in_i_tidak_menggunakan_huruf_kapital",
28                         "Excerpt" => $value
29                     ];
30                 }
31             }
32         }
33         return $result;
34     }
35 }
36 }
37 }
38 ?>

```

Listing A.10: PS09_SubChapterChecker.php

```

1 <?php
2
3 /*
4  * Sub class dari kelas Checker
5  * Memeriksa jumlah subbab dalam sebuah bab
6  * @author Marcell Trixie
7  */
8 class PS09_SubChapterChecker extends Checker{
9
10     /**
11      * Method untuk memeriksa kesalahan dalam dokumen skripsi
12      * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
13      * @return result[] laporan kesalahan yang ditemukan
14      */
15     public function errorChecking($pdf_extract){
16         $result = [];
17         $temp = $pdf_extract->getTableOfContentPage();
18         $array = array_merge($temp);
19         for ($i=0; $i<sizeof($array); $i++) {
20             if(preg_match('/\.$/', $array[$i])){
21                 $substring = substr($array[$i+1], 0, strlen($array[$i])-1);
22                 if(!array_search($substring.'2', $array)){
23                     $result[] = [
24                         "Error_Code" => "PS-09",
25                         "Note" => "Bab/Subbab_" . $array[$i] . "_ini_hanya_terdapat_1_sub_bab/sub_sub_bab",
26                     ];
27                 }
28             }
29         }
30         return $result;
31     }
32 }
33 }

```

```
34 |
35 | ?>
```

Listing A.11: VAN03_SubjectPronounChecker.php

```
1 <?php
2
3 /*
4  * Sub class dari kelas Checker
5  * Memeriksa penggunaan kata ganti orang
6  * @author Marcell Trixie
7  */
8 class VAN03_SubjectPronounChecker extends Checker{
9
10     /**
11      * Method untuk memeriksa kesalahan dalam dokumen skripsi
12      * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
13      * @return result[] laporan kesalahan yang ditemukan
14      */
15     public function errorChecking($pdf_extract){
16         $result = [];
17         $sentence = $pdf_extract->splitContentPage();
18         foreach ($sentence as $index => $value) {
19             $pattern = "/\bsaya\b|_\bkamu\b|_\bdia\b/i";
20             if (preg_match($pattern, $value)) {
21                 $result[] = [
22                     "Error_Code" => "VAN-03",
23                     "Note" => "Kalimat_ini_mengandung_kata_ganti_orang",
24                     "Excerpt" => $value
25                 ];
26             }
27         }
28         return $result;
29     }
30 }
31
32
33 ?>
```