SKRIPSI

APLIKASI PEMERIKSA KESALAHAN DOKUMEN SKRIPSI INFORMATIKA UNPAR

Marcell Trixie Alexander

NPM: 2014730003

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

UNDERGRADUATE THESIS

GENERAL ERROR CHECKER APPLICATION FOR UNPAR INFORMATICS THESIS DOCUMENT

Marcell Trixie Alexander

NPM: 2014730003

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

ABSTRAK

Skripsi merupakan karangan ilmiah yang wajib ditulis oleh mahasiswa sebagai bagian dari persyaratan akhir pendidikan akademiknya di Perguruan Tinggi. Namun dalam penulisannya, mahasiswa sering melakukan kesalahan kecil yang tidak dapat diabaikan. Mahasiswa seharusnya dapat memeriksa dan meminimalisir kesalahan sendiri, sehingga waktu bimbingan dapat dimanfaatkan untuk membahas konten skripsi. Pada skripsi ini 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 sejumlah kasus uji yang dibuat menggunakan template skripsi yang digunakan Fakultas Teknologi Informasi dan Sains. 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 essay that must be written by students as part of the final requirement of academic education in University Degree. But in writing, students often make small mistakes that cannot be ignored. 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. This thesis will be developed a software that be able to check thesis document errors.

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 some of test cases, that created using a thesis template of Faculty of Information Technology and Science. Testing by test cases as input and comparing the error report issued with the expected results.

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

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

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

1.1 Latar Belakang

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

Dari masalah tersebut dapat dibuat sebuah aplikasi untuk melakukan pemeriksaan pada dokumen skripsi. Kesalahan yang akan diperiksa berasal dari survei yang dilakukan kepada dosen-dosen Informatika Unpar. Hasil dari survei tersebut akan diseleksi untuk diimplementasikan ke dalam aplikasi. Aplikasi sederhana ini dapat dimanfaatkan oleh mahasiswa Informatika Unpar secara mandiri. Aplikasi akan memeriksa dokumen *PDF* skripsi dan 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. Dapat memeriksa kesalahan yang ada pada dokumen skripsi.
2. Dapat membangun perangkat lunak untuk memeriksa kesalahan yang ada pada dokumen skripsi.

1.4 Batasan Masalah

Batasan masalah skripsi ini adalah sebagai berikut:

1. Jenis dokumen yang dapat diperiksa oleh perangkat lunak yang dibuat adalah dokumen dengan ekstensi *PDF*.
2. Pemeriksaan menggunakan *pattern matching* tanpa analisis gramatikal.
3. Pemeriksaan kosakata tanpa imbuhan.

1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan pada skripsi ini adalah sebagai berikut:

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

1.6 Sistematika Pembahasan

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

1. Bab 1 Pendahuluan
   Bab 1 akan membahas latar belakang dibuatnya perangkat lunak untuk memeriksa kesalahan dokumen skripsi. Pada bab ini dibahas juga rumusan masalah, tujuan skripsi, batasan masalah dan metodologi penelitian yang digunakan pada skripsi.
2. Bab 2 Landasan Teori
   Bab 2 yang merupakan landasan teori akan berisi teori-teori yang menjadi dasar-dasar dalam penulisan skripsi ini. Teori yang akan dibahas pada bab 2, yaitu *Regular Expression*, *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 survei secara personal kepada dosen-dosen Informatika

Unpar. Hasil dari setiap survei akan dipilih mana saja yang dapat diimplementasikan dalam perangkat lunak.

4. Bab 4 Perancangan
   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 pemrogramam *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. PCRE memiliki beberapa fungsi yang dapat digunakan untuk mencocokan pola *regex*. Berikut ini adalah fungsi-fungsi yang terdapat pada PCRE:

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. Pada karakter meta *outside square brackets* terdapat 14 simbol, sedangkan karakter meta *inside square brackets* terdapat 3 simbol. 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
.	Dot
[ ]	Square Bracket
|	Vertical Bar
( )	Parenthesis
{ }	Curly Bracket
?	Question Mark
*	Asterisk
+	Plus

Pada tabel 2.1, terdapat 11 simbol yang termasuk karakter meta *outside square brackets*. Setiap simbol 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 mencocokan 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, keculai karakter *line break*. Contohnya untuk pola "/HelloW.rld/", pola tersebut dapat cocok dengan HelloWarld, 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 mencocokan 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 mengelompokan 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 diletakan 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 2.2.

Tabel 2.2: Tabel metakarakter *inside square brackets*

Simbol	Nama
\	Backslash
^	Circumflex
-	Hyphen

Pada tabel 2.2 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 [ˆ0] 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 dikelompokan 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:

Tabel 2.3: 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.
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.

Tabel 2.4: Tabel kelas karakter

Kelas	Deskripsi	Keterangan
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 hexadesimal, 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 yang belum bisa ditangani oleh *library* ini. *PdfParser* belum dapat mengekstrak dokumen yang diamankan. Selain itu, *library* ini tidak dapat mendeteksi jenis teks yang dicetak miring, tebal dan bergaris bawah. Hingga saat ini, pengembangan *library PdfParser* masih terus berjalan. *PdfParser* memiliki beberapa kelas yang menjalankan fungsional dari *library ini*, salah satunya kelas *Parser*. Kelas *Parser* berfungsi untuk mengatur ekstraksi file PDF. *Method-method* yang terdapat pada kelas ini adalah sebagai berikut:

1. $object
   Atribut ini adalah sebuah array, yang akan menyimpan objek-objek dari file PDF yang akan diekstrak.

2. public function _construct()
   Berfungsi untuk konstruktor kelas Parser.

3. public function parseFile($filename)
   Method ini berfungsi untuk mengekstrak isi file dokumen PDF.
   Parameter: nama file yang akan diekstrak.
   Kembalian: konten dari file yang akan diekstrak.

4. public function parseContent($content)
   Method ini berfungsi untuk mengekstrak konten pada dokumen.
   Parameter: konten dari file PDF.
   Kembalian: dokumen yang akan diekstrak.

5. protected function parseTrailer($structure, $document)
   Method ini berfungsi untuk
   Parameter: struktur dan dokumen.
   Kembalian: header dengan parameter trailer dan dokumen.

6. protected function parseObject($id, $structure, $document)
   Method ini berfungsi untuk.
   Parameter: id, struktur dan dokumen

7. protected function parseHeader($structure, $document)
   Method ini berfungsi untuk mengekstrak header pada dokumen.
   Parameter: struktur dan dokumen
   Kembalian: header dengan parameter trailer dan dokumen.

8. protected function parseHeaderElement($type, $value, $document)
   Method ini berfungsi untuk mengekstrak elemen-elemen yang ada pada header.
   Parameter: tipe, value dan dokumen

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 untuk kamus Indonesia. 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.oxt*. 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 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-31128 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 awalan dan akhiran untuk kamus Indonesia. Awalan akan disimbolkan dengan huruf kapital, sedangkan akhiran akan disimbolkan dengan huruf kecil. Selain ke-2 hal tesebut, ada juga yang merupakan gabungan dari awalan dan akhiran. 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 menunjukan awalan yang dimulai dengan huruf B, dan angka 0 menunjukan 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 menunjukan akhiran yang dimulai dengan huruf k, dan angka 0 menunjukan 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 imbuhan tunggal yang terbentuk dari perpaduan awalan dan akhiran.

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 menunjukan 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 menunjukan berapa banyak jumlah konfiks yang dapat digunakan. Pada baris 1 terdapat 2 jenis, sedangkan pada baris 5 terdapat 1 jenis saja.
- Kolom ke-5 menunjukan 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 Moertini, MT.

Tabel 3.2: Tabel informasi sidang skripsi yang diamati

Tanggal	Mahasiswa	Judul Skripsi	Penguji
17-05-2019	Billy Adiwijaya	Pembangkit Timelapse Pengembangan Proyek Perangkat Lunak	-Kristopher David Harjono M.T. -Elisati Hulu M.T.
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	Open Source Snake 360	-Chandra Wijaya S.T., M.T. -Raymond Chandra Putra, S.T., M.T.

Dari ke-7 pengamatan tersebut, terdapat beberapa kesalahan-kesalahan yang terjadi dalam penulisan dokumen 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 *typo*.
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 dimasukan 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*user* dan pengguna. Mahasiswa harus memilih antara memakai *user* 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 Keputusan Implementasi Hasil Survei

Pada bagian ini akan dijelaskan tentang keputusan implementasi dari hasil survei. 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 *regex*. Metode yang digunakan untuk mendeteksi kesalahan yaitu dengan *pattern matching*, sehingga hal-hal yang bersifat kontekstual tidak dapat dicek dengan *regex*. 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	Diimplementasi	Dapat diselesaikan menggunakan *regular expression*
PS-02	Penggunaan imbuhan di- dan kata depan di-	Tidak diimplementasi	Tidak dapat diselesaikan menggunakan *regular expression*, karena pada kamus Indonesia *LibreOffice* 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	Diimplementasi	Dapat diselesaikan menggunakan *regular expression*
PS-04	Terdapat ruang kosong yang besar	Tidak diimplementasi	Tidak dapat diselesaikan menggunakan *regular expression*, karena hasil ekstrak dari PDF menggunakan *PDF Parser* tidak mendeteksi adanya baris kosong.
PS-05	Awal kalimat tidak menggunakan huruf kapital	Diimplementasi	Dapat diselesaikan menggunakan *regular expression*
PS-06	Tidak ada spasi antar kata	Tidak diimplementasi	Dapat diselesaikan menggunakan *regular expression*. Namun survei ini tidak diimplementasi, karena penyelesaiannya sama dengan survei PS-01, sehingga akan disatukan implementasinya.
PS-07	Gambar tidak sesuai tempatnya	Tidak diimplementasi	Tidak dapat diselesaikan menggunakan *regular expression*, karena hasil ekstrak PDF dari *PDF Parser* tidak dapat mendeteksi gambar.
PS-08	Tidak ada keterangan untuk gambar dan tabel	Tidak diimplementasi	Tidak dapat diselesaikan menggunakan *regular expression*, karena hasil ekstrak PDF dari *PDF Parser* tidak dapat mendeteksi gambar dan tabel.
PS-09	Jumlah sub bab, sub sub bab tidak boleh hanya 1	Diimplementasi	Dapat diselesaikan menggunakan *regular expression*
KAL-01	Cetak miring untuk bahasa asing	Tidak diimplementasi	Tidak dapat diselesaikan menggunakan *regular expression*, karena membutuhkan kamus bahasa Inggris. Selain itu *PDF Parser* tidak dapat mencocokan teks yang cetak miring.
KAL-02	Kalimat pengantar untuk setiap subbab	Diimplementasi	Dapat diselesaikan menggunakan *regular expression*.
KAL-03	Kelengkapan data skripsi	Diimplementasi	Dapat diselesaikan menggunakan *regular expression*

Tabel 3.10: Tabel keputusan implementasi

Kode	Survei	Keputusan	Alasan
CHW-01	Letak keterangan untuk gambar dan tabel	Tidak diimplementasi	Tidak dapat diselesaikan menggunakan *regular expression*, karena hasil ekstrak PDF dari *PDF Parser* tidak dapat mendeteksi gambar dan tabel.
CHW-02	Penggunaan bahasa yang benar	Tidak diimplementasi	Dapat diselesaikan menggunakan *regular expression*. Namun survei ini tidak diimplementasi, karena penyelesaiannya sama dengan survei PS-01, sehingga akan disatukan implementasinya.
HUH-01	Rujukan untuk gambar dan tabel	Tidak diimplementasi	Tidak dapat diselesaikan menggunakan *regular expression*, karena hasil ekstrak PDF dari *PDF Parser* tidak dapat mendeteksi gambar dan tabel.
HUH-02	Penulisan pseudocode	Tidak diimplementasi	Tidak dapat diselesaikan menggunakan *regular expression*, karena hasil ekstrak PDF dari *PDF Parser* tidak dapat mendeteksi pseudocode.
HUH-03	Penulisan kata hubung	Tidak diimplementasi	Tidak dapat diselesaikan menggunakan *regular expression*, karena *regex* 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 diimplementasi	Dapat diselesaikan menggunakan *regular expression*. Namun survei ini tidak diimplementasi, karena penyelesaiannya sama dengan survei KAL-03, sehingga akan disatukan implementasinya.
VAN-02	Konsistensi penggunaan kata	Tidak diimplementasi	Tidak dapat diselesaikan menggunakan *regular expression*, karena tidak dapat membuat padanan kata untuk memeriksa kosistensi penggunaan kata.

Tabel 3.11: Tabel keputusan implementasi

Kode	Survei	Keputusan	Alasan
VAN-03	Penggunaan kata ganti orang	Diimplementasi	Dapat diselesaikan menggunakan *regular expression*
NAT-01	Penulisan daftar referensi	Diimplementasi	Dapat diselesaikan menggunakan *regular expression*

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.

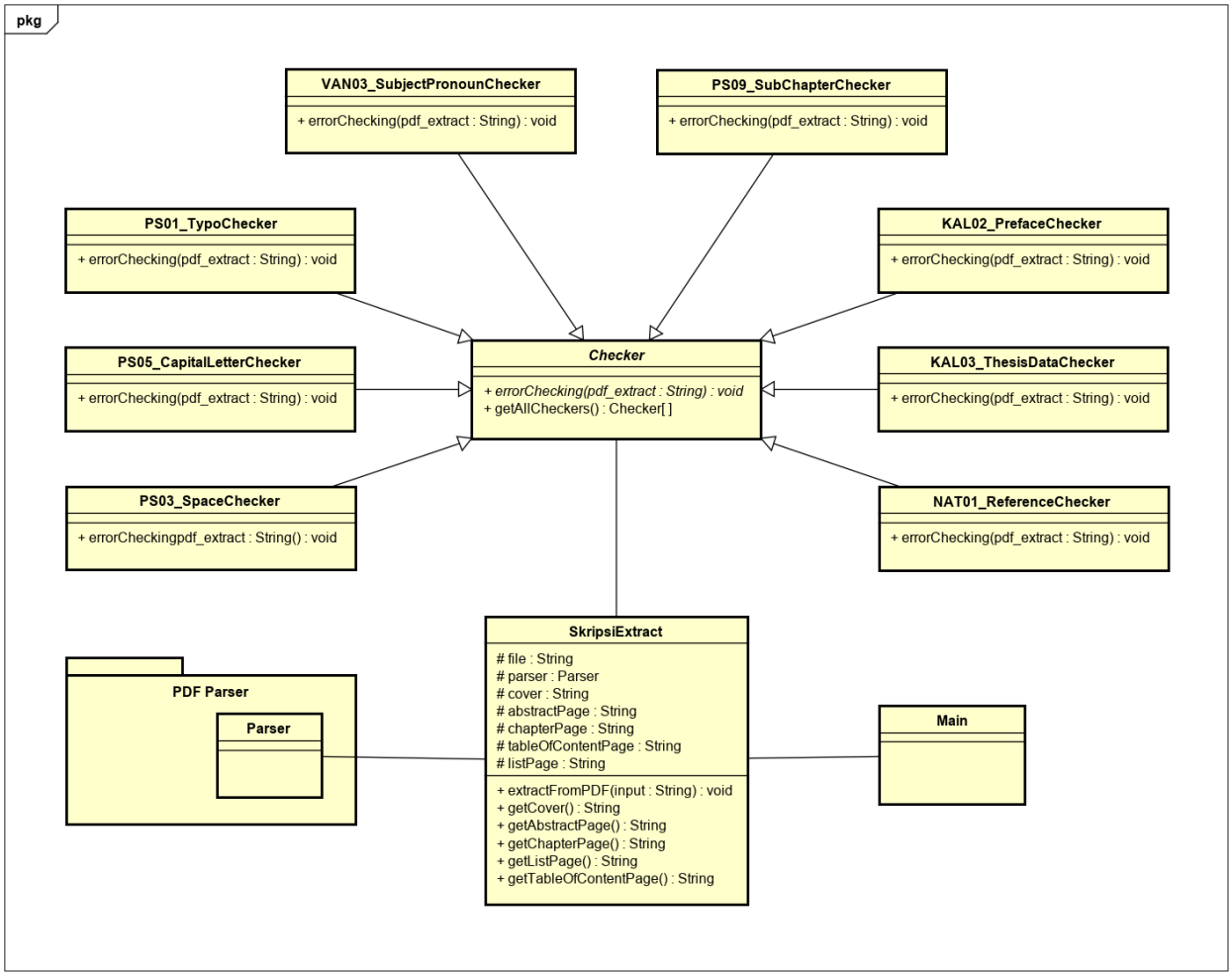
BAB 4

PERANCANGAN

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

4.1 Perancangan Kelas

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



Gambar 4.1: Diagram kelas Aplikasi Pemeriksa Kesalahan Dokumen Skripsi

Gambar 4.1 merupakan diagram kelas untuk perangkat lunak yang akan dibuat. Pada diagram kelas tersebut, ditunjukan bahwa perangkat lunak memiliki sebelas kelas dan sebuah package *library PDF Parser*. Perangkat lunak akan menggunakan paradigma *Object Oriented Programming* dalam pengembangannya. Rincian dari setiap kelas tersebut akan dijelaskan sebagai berikut.

1. 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*. 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.
   - getAllChecker()
     Method ini berfungsi untuk melakukan instansiasi seluruh anak kelas *Checker*. Method ini mengembalikan hasil instansiasi dari anak kelas *Checker*.

2. 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). 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
   Kelas ini bertanggungjawab untuk memeriksa kelengkapan data skripsi yang ditulis dalam bahasa Indonesia maupun bahasa Inggris. elas 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
   Kelas ini bertanggungjawab untuk memeriksa referensi yang akan dirujuk dalam dokumen. elas 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 bab 1 hingga 6.

5. Kelas PS01_TypoChecker
   Kelas ini bertanggungjawab untuk memeriksa kesalahan penulisan kata. 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 bab 1 hingga 6.

6. Kelas PS03_SpaceChecker
   Kelas ini bertanggungjawab untuk memeriksa penggunaan spasi sebelum dan setelah tanda baca. 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 bab 1 hingga 6.

7. Kelas PS05_CapitalLetterChecker
   Kelas ini bertanggungjawab untuk memeriksa penggunaan huruf kapital pada awal kalimat. 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 bab 1 hingga 6.

8. 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). *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
   Kelas ini bertanggungjawab untuk memeriksa penggunaan kata ganti orang pada dokumen skripsi. 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 bab 1 hingga 6.

10. Kelas SkripsiExtract
    Kelas ini berfungsi untuk melakukan proses ekstak 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.

4.2 Perancangan Perangkat Lunak

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

4.2.1 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 4.1: 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 4.2: 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 4.1 dan listing 4.2 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 4.1 menggunakan *method getText()* dan hasil dari ekstraknya disimpan menjadi sebuah kesatuan dari halaman awal hingga halaman akhir. Listing 4.2 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.

4.2.2 Pola Pengecekan Kesalahan

Hasil survei kesalahan-kesalahan dalam penulisan dokumen skripsi sudah disaring dan akan di-implementasikan 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:  function ErrorChecking($pdf_exctract)
2:      Input: An object from the SkripsiExtract class to call the getter method
3:      Output: An array containing error reports
4:      temp ← Explode dictionary based on newline
5:      dictionary ← Array for store dictionary
6:      for i = 0 to size of temp do
7:          Remove affix code from each index dictionary array
8:      word ← Call getter from class SkripsiExtract and split with non word character
9:      typos ← Array for store typos
10:     for each word in array do
11:         if value not contain in dictionary and value not contain in typos then
12:             Fill typos with value
13:     result ← Array for store all errors report
14:     if size of typos greater than 0 then
15:         result ← Add all errors report into result
16:     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 mencocokan kata yang akan diperiksa dengan kamus bahasa Indonesia *LibreOffice*. Berikut adalah contoh dari kesalahan dan laporan yang dikeluarkan:

- Contoh kesalahan
  Istilah *regex* berasal dari toeri 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:  function ErrorChecking($pdf_exctract)
2:      Input: An object from the SkripsiExtract class to call the getter method
3:      Output: An array containing error reports
4:      result ← Array for store all errors report
5:      sentence ← Call getter from class SkripsiExtract
6:      for each sentence in sentence do
7:          pattern ← Define regex /([A-Za-z]2,[,.!?][A-Za-z]+)/
8:          if pattern contain in array then
9:              result ← Fill with errors report
10:     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 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

```
function ErrorChecking($pdf_exctract)
    Input: An object from the SkripsiExtract class to call the getter method
    Output: An array containing error reports
    result ← Array for store all errors report
    sentence ← Call getter from class SkripsiExtract
    for each sentence in sentence do
        pattern ← Define regex /^[a-z]+ | [a-z]+[a-z] /
        if pattern contain in array then
            if first character in a sentence is a non word character then
                Continue
        else
            result ← Fill with errors report
    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

```
function ErrorChecking($pdf_exctract)
    Input: An object from the SkripsiExtract class to call the getter method
    Output: An array containing error reports
    result ← Array for store all errors report
    sentence ← Call getter from class SkripsiExtract
    for each sentence in sentence do
        pattern ← Define regex /^/
        if pattern contain in array then
            fill result with errors report
    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:  function ErrorChecking($pdf_exctract)
2:      Input: An object from the SkripsiExtract class to call the getter method
3:      Output: An array containing error reports
4:      result ← Array for store all errors report
5:      sentence ← Call getter from class SkripsiExtract
6:      for each sentence in sentence do
7:          pattern ← Define regex /^[A-Z0-9][A-Za-z] /
8:          if pattern contain in array then
9:              fill result with errors report
10:     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

```
function ErrorChecking($pdf_exctract)
    Input: An object from the SkripsiExtract class to call the getter method
    Output: An array containing error reports
    result ← Array for store all errors report
    sentence ← Call getter from class SkripsiExtract
    for each sentence in sentence do
        pattern ← Define regex /JUDUL BAHASA INDONESIA|JUDUL BAHASA INGGRIS|Nama Lengkap|10 digit NPM UNPAR|tahun/
        if pattern contain in array then
            fill result with errors report
    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

```
function ErrorChecking($pdf_exctract)
    Input: An object from the SkripsiExtract class to call the getter method
    Output: An array containing error reports
    result ← Array for store all errors report
    sentence ← Split PDF parsing results based on dots and space
    for each sentence in sentence do
        pattern ← Define regex /saya|kamu|dia/i
        if pattern contain in array then
            fill result with errors report
    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:  function ErrorChecking($pdf_exctract)
2:      Input: An object from the SkripsiExtract class to call the getter method
3:      Output: An array containing error reports
4:      result ← Array for store all errors report
5:      sentence ← Call getter from class SkripsiExtract
6:      for each sentence in sentence do
7:          pattern ← Define regex /[?]/
8:          if pattern contain in array then
9:              fill result with errors report
10:     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 5.14 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
       kecil yang tidak dapat diabaikan.kesalahan sering terjadi dalam penggunaan
       imbuhan, kata keterangan, penulisan kata dan sebagainya
6
7  ERROR 2
8  ==========
9  Error Code: PS-03
10 Note: Perhatikan spasi setelah tanda baca.
11 Excerpt: 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
12
13 ERROR 3
14 ==========
15 Error Code: PS-05
16 Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
17 Excerpt: Namun dalam penulisannya, peserta skripsi sering melakukan kesalahan
       kecil yang tidak dapat diabaikan.kesalahan sering terjadi dalam penggunaan
       imbuhan, kata keterangan, penulisan kata dan sebagainya
```

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 dimasukan 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. Pada bab 1, beberapa karakter pertama dalam awal kalimat menggunakan huruf kecil. Hal ini dilakukan untuk menguji fitur pemeriksa huruf kapital (PS-05).

3. 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).

4. 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).

5. Pada bab 4, 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. 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.

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: Perintah yang digunakan untuk menjalankan perangkat lunak

```
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, batasan,
      penelitian, merupakan, karangan, ditulis, sebagai, bagian, persya, ratan,
      pendidikan, akademiknya, penulisannya, peserta, melakukan, kesalahan,
      diabaikan, Kesalahan, terjadi, penggunaan, imbuhan, keterangan, sebagainya,
      seharusnya, diperiksa, diminimalisir, bimbingan, pembimbing, dimanfaatkan,
      membahas, konten, dibanding, memeriksa, tersebut, dibuat, sebuah,
      pemeriksaan, berasal, dilakukan, Unpar, diseleksi, diimplementasikan, secara
      , dijalankan, melalui, command, Windows, menerima, masukan, berupa, le, PDF,
       menampilkan, laporan, berisi, ditemukan, Rumusan, Berdasarkan, dirumuskan,
      berikut, membuat, Tujuan, adalah, membangun, LANDASAN, TEORIPada, dibahas,
      landasan, expression, library, LibreO, Expression, regex, tertentu,
      digunakan, pemrograman, Regex, biasanya, memveri, kecocokan, input,
      menemukan, mengganti, menyusun, membagi, menjadi, subteks, pencocokan,
      misalnya, validasi, string, username, password, mail, IP, Pemanfaatan,
      menyederhanakan, pemrosesan, kehidupan, sehari, mencerminkan, disebut,
      keteraturan, menggunakan, Deterministic, Finite, DFA, nite, state,
      machineyang, backtracking, berbagai, satunya, Perl, Compa, tible, PCRE,
      serangkaian, menerapkan, sintaks, perbedaan, Perlversi, Metakarakter,
      dibedakan, berdasarkan, posisinya, metakarakter, outside, square, brackets,
      fungsinya, berbeda, terdapat, sedangkan, inside, bracketsterdapat, Karakter,
       karakter, memiliki, spesi, k, dikelompokan, setiap, didalam, bracket,
      mendukung, POSIX, penggunaannya, diantara, Sebagai, alnum, keputusan,
      UmumPada, mengumpulkan, dibutuhkan, pengembangan, dicari, dipilih,
      pelaksanaannya, dibagi, pengamatan, Pengamatan, berlangsung, Mei, diamati,
      melainkan, diambil, pertimbangan, diuji, menghadiri, disajikan, Penguji,
      Osfaldo, Mickael, Oktavianus, Naibaho, Penjualan, Vania, Natali, S, M, T,
      Elisati, Ricky, Wahyudi, Menggunakan, Fitur, Surf, Dr, rer, nat, Cecilia,
      Esti, Nugraheni, ST, MT, Ir, Veronica, Moer, tini, selanjutnya, diminta,
      PERANCANGANPada, perancangan, dibangun, meliputi, algoritma, pengecekan,
      Perancangan, rancangan, Rancangan, ditunjukan, Pemeriksa, PENGUJIANPada,
      pengujian, terhadap, Lingkungan, Pengujian, lingkungan, beserta,
      implementasinya, Berikut, pembangunan, Spesi, Processor, AMD, Bristol, Ridge
      , Quad, Core, FX, P, GHz, GB, GPU, Radeon, RX, Storage, Harddisk, TB, PHP,
      cli, Composer, Sublime, Text, PdfParser, Graphical, User, Interface,
      kegiatan, disimpan, disediakan, mengeluarkan, Listing, menjalankan, phpmain,
       php, r, s, n, m, f, i, l, p, d, KESIMPULAN, SARANPada, kesimpulan,
      Kesimpulan
```

2. Kasus uji TC_PF_02.pdf

Listing 5.4: Perintah yang digunakan untuk menjalankan perangkat lunak

```
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, batasan,
      penelitian, merupakan, karangan, ditulis, sebagai, bagian, persya, ratan,
      pendidikan, akademiknya, penulisannya, peserta, melakukan, kesalahan,
      diabaikan, terjadi, penggunaan, imbuhan, keterangan, sebagainya, seharusnya,
       diperiksa, diminimalisir, bimbingan, pembimbing, dimanfaatkan, membahas,
      konten, dibanding, memeriksa, tersebut, dibuat, sebuah, pemeriksaan, berasal
      , dilakukan, Unpar, diseleksi, diimplementasikan, secara, dijalankan,
      melalui, command, Windows, menerima, masukan, berupa, le, PDF, menampilkan,
      laporan, berisi, ditemukan, Rumusan, Berdasarkan, dirumuskan, berikut,
      membuat, Tujuan, adalah, membangun, LANDASAN, TEORIPada, dibahas, landasan,
      expression, library, LibreO, Expression, regex, tertentu, digunakan,
      pemrograman, Regex, biasanya, memveri, kecocokan, input, menemukan,
      mengganti, menyusun, membagi, menjadi, subteks, pencocokan, misalnya,
      validasi, string, username, password, mail, IP, Pemanfaatan, menyederhanakan
      , pemrosesan, kehidupan, sehari, mencerminkan, disebut, keteraturan,
      menggunakan, Deterministic, Finite, DFA, nite, state, machineyang,
      backtracking, berbagai, satunya, Perl, Compa, tible, PCRE, serangkaian,
      menerapkan, sintaks, perbedaan, Perlversi, Metakarakter, dibedakan,
      berdasarkan, posisinya, metakarakter, outside, square, brackets, fungsinya,
      berbeda, terdapat, sedangkan, inside, bracketsterdapat, Kesalahan, UmumPda,
      baggian, mengumpulkan, dibutuhkan, pengembangan, Infrmasi, yng, dicari,
      ksalahan, umumm, yaang, serring, terjaadi, paada, metod, ang, dipilih,
      pelaksanaannya, dibagi, menjhadi, pengamatan, Pengamatan, berlangsung, Mei,
      diamati, melainkan, diambil, pertimbangan, diuji, menghadiri, disajikan,
      Penguji, Osfaldo, Mickael, Oktavianus, Naibaho, Penjualan, Vania, Natali, S,
       M, T, Elisati, Ricky, Wahyudi, Menggunakan, Fitur, Surf, Dr, rer, nat,
      Cecilia, Esti, Nugraheni, ST, MT, Ir, Veronica, Moer, tini, PERANCANGANPada,
       perancangan, dibangun, meliputi, algoritma, pengecekan, Perancangan,
      rancangan, Rancangan, ditunjukan, Pemeriksa, PENGUJIANPada, pengujian,
      terhadap, Lingkungan, Pengujian, lingkungan, beserta, implementasinya,
      Berikut, spesi, pembangunan, Spesi, Processor, AMD, Bristol, Ridge, Quad,
      Core, FX, P, GHz, GB, GPU, Radeon, RX, Storage, Harddisk, TB, PHP, cli,
      Composer, Sublime, Text, KESIMPULAN, SARANPada, kesimpulan, Kesimpulan,
      berikan
38
```

Laporan yang ditunjukan pada listing sudah sesuai dengan yang diharapkan. Fitur ini melakukan pengecekan pada setiap kata yang ada pada dokumen. Namun fitur ini belum dapat membedakan kata yang termasuk nama orang, nama tempat, nama barang, kata yang berimbuhan dan kata yang menggunakan bahasa asing.

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: Perintah yang digunakan untuk menjalankan perangkat lunak

```
ERROR 1
==========
Error Code: PS-03
Note: Perhatikan spasi setelah tanda baca.
Excerpt: Listing 5.1: Perintah yang digunakan untuk menjalankan perangkat lunak
    1phpmain.php
```

2. Kasus uji TC_PF_02.pdf

Listing 5.6: Perintah yang digunakan untuk menjalankan perangkat lunak

```
ERROR 1
==========
Error Code: PS-03
Note: Perhatikan spasi setelah tanda baca.
Excerpt: Namun dalam penulisannya, peserta skripsi sering melakukan kesalahan
    kecil yang tidak dapat diabaikan.kesalahan sering terjadi dalam penggunaan
    imbuhan, kata keterangan, penulisan kata dan sebagainya

ERROR 2
==========
Error Code: PS-03
Note: Perhatikan spasi setelah tanda baca.
Excerpt: 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
```

Laporan yang ditunjukan pada gambar **??** sudah sesuai dengan yang diharapkan. Fitur ini masih belum bisa membedakan penggunaan tanda titik pada gelar pendidikan, perangkat lunak mendeteksi hal tersebut menjadi sebuah kesalahan dalam fitur ini.

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: Perintah yang digunakan untuk menjalankan perangkat lunak

```
ERROR 1
==========
Error Code: PS-05
Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
```

```
Excerpt: Namun dalam penulisannya, peserta skripsi sering melakukan kesalahan kecil yang tidak dapat diabaikan.kesalahan sering terjadi dalam penggunaan imbuhan, kata keterangan, penulisan kata dan sebagainya

ERROR 2
==========
Error Code: PS-05
Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
Excerpt: pada saat bimbingan, waktu dosen pembimbing lebih baik dimanfaatkan untuk membahas konten dibanding memeriksa kesalahan-kesalahan tersebut

ERROR 3
==========
Error Code: PS-05
Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
Excerpt: dari masalah tersebut dapat dibuat sebuah aplikasi untuk melakukan pemeriksaan pada dokumen skripsi

ERROR 4
==========
Error Code: PS-05
Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
Excerpt: kesalahan yang akan diperiksa berasal dari survei yang dilakukan kepada dosen-dosen Informatika Unpar

ERROR 5
==========
Error Code: PS-05
Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
Excerpt: hasil dari survei tersebut akan diseleksi untuk diimplementasikan ke dalam aplikasi

ERROR 6
==========
Error Code: PS-05
Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
Excerpt: aplikasi sederhana ini dapat dimanfaatkan oleh mahasiswa Informatika Unpar secara mandiri

ERROR 7
==========
Error Code: PS-05
Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
Excerpt: aplikasi ini dijalankan melalui melalui terminal command Windows

ERROR 8
==========
Error Code: PS-05
Note: Huruf pertama pada kalimat ini tidak menggunakan huruf kapital
Excerpt: aplikasi menerima masukan berupa file PDF skripsi dan menampilkan laporan yang berisi kesalahan-kesalahan yang ditemukan pada dokumen skripsi
```

Laporan yang ditunjukan pada gambar **??** sudah sesuai dengan yang diharapkan.

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: Perintah yang digunakan untuk menjalankan perangkat lunak

```
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 ditunjukan pada gambar **??** sudah sesuai dengan yang diharapkan.

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: Perintah yang digunakan untuk menjalankan perangkat lunak

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

Laporan yang ditunjukan pada gambar **??** sudah sesuai dengan yang diharapkan.

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: Perintah yang digunakan untuk menjalankan perangkat lunak

```
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
      NPM UNPAR PROGRAM STUDI MATEMATIKA/FISIKA/TEKNIK INFORMATIKA FAKULTAS
      TEKNOLOGI INFORMASI DAN SAINS UNIVERSITAS KATOLIK PARAHYANGAN tahun FINAL
      PROJECT/UNDERGRADUATE THESIS JUDUL BAHASA INGGRIS Nama Lengkap NPM: 10 digit
       NPM UNPAR DEPARTMENT OF MATHEMATICS/PHYSICS/INFORMATICS FACULTY OF
      INFORMATION TECHNOLOGY AND SCIENCES PARAHYANGAN CATHOLIC UNIVERSITY tahun
```

Laporan yang ditunjukan pada gambar **??** sudah sesuai dengan yang diharapkan.

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: Perintah yang digunakan untuk menjalankan perangkat lunak

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

Laporan yang ditunjukan pada gambar **??** sudah sesuai dengan yang diharapkan.

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 *test case* ini, perangkat lunak tidak menemukan kesalahan dalam dokumen skripsi.

2. Kasus uji TC_PF_02.pdf

Listing 5.12: Perintah yang digunakan untuk menjalankan perangkat lunak

```
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
       lunak ini adalah sebagai berikut
6
7  ERROR 2
8  ==========
9  Error Code: VAN-03
10 Note: Kalimat ini mengandung kata ganti orang
11 Excerpt: 6.2Saran Saran yang saya berikan untuk pengembangan perangkat lunak
       ini adalah sebagai berikut
```

Laporan yang ditunjukan pada gambar **??** sudah sesuai dengan yang diharapkan.

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. TC_PE_01.pdf [5]
2. TC_PE_02.pdf [6]
3. TC_PE_03.pdf [7]
4. TC_PE_04.pdf [8]
5. 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: Perintah yang digunakan untuk menjalankan perangkat lunak

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

Pada kasus uji ini, perangkat lunak mendeteksi bahwa file ini merupakan file yang *secured*. Hal ini sudah dijelaskan pada sub bab 2.2, bahwa *library* ini tidak dapat melakukan ekstrak dokumen yang *secured*.

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.

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.

[1] https://github.com/ftisunpar/Skripsi

4. TC_PE_04.pdf

Listing 5.14: Perintah yang digunakan untuk menjalankan perangkat lunak

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

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.

5.3.1 Pengujian terhadap PDF yang tidak mengeluarkan laporan

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 5.15: 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 5.15, 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 5.16: Potongan kode pada *method parseContent*

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

Pada listing 5.16, 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 5.17: 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 5.17, 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 5.18: 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 5.18, 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 5.19: 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 5.19, sumber masalah berada pada baris ke-3 (baris ke-781 pada *source code*). Baris tersebut merujuk pada kelas TCPDF_Filters.

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

Listing 5.20: Potongan kode pada *method decodeFilter()*

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

Pada listing 5.20, sumber masalah ada pada baris ke-2 (baris ke-94 pada *source code*). Baris tersebut merujuk pada method *decodeFilterFlateCode()*.

Listing 5.21: 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 5.21, 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 5.22: 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 menggunakan kode pada baris ke-2 di listing5.23, perangkat lunak berhasil mengeluarkan laporan kesalahan.

Listing 5.23: Pesan kesalahan memory limit

```
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 5.23, maka ada beberapa hal yang perlu ditambahkan pada perintah 5.14 untuk menjalankan perangkat lunak.

Listing 5.24: Perintah untuk mengatasi masalah *memory limit*

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

Dengan menggunakan perintah pada 5.24, 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 Pengujian terhadap Secured PDF

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

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.

BAB 6

KESIMPULAN DAN SARAN

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

6.1 Kesimpulan

Dari hasil penelitian yang dilakukan, didapatkan kesimpulan-kesimpulan sebagai berikut:

1. Kesalahan yang ada pada dokumen skripsi diperiksa menggunakan metode pencocokan pola *regular expression*. Pola akan memeriksa kesalahan yang bersifat tekstual pada dokumen skripsi.

2. Perangkat lunak pemeriksa kesalahaan dokumen skripsi 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.

6.2 Saran

Dari hasil penelitian yang dilakukan, berikut adalah beberapa saran yang dapat digunakan sebagai pengembangan perangkat lunak:

1. Mengembangkan fitur pemeriksa kesalahan kata, agar dapat memeriksa kata-kata yang menggunakan imbuhan.

2. Mencari alternatif *library PdfParser*, karena *library* ini masih ada beberapa kekurangan.

3. 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

```php
<?php

    /*
     * Kelas untuk menjalankan checker
     * Author Marcell Trixie
     */
    class SkripsiExtract{

        protected $file;
        protected $parser;
        protected $cover;
        protected $abstractPage;
        protected $otherPage;
        protected $tableOfContentPage;
        protected $listPage;
        protected $contentPage;

        /**
         * Method untuk mengekstrak file PDF skripsi
         * @param input file pdf yang akan diesktrak
         */
        public function extractFromPDF($input){
            $directory = getcwd();
            $this->parser = new \Smalot\PdfParser\Parser();
            $this->file = $directory . '/' . $input;
            $pdf = $this->parser->parseFile($this->file);
            $pages  = $pdf->getPages();

            foreach ($pages as $page => $value) {
                if($page == 0 || $page == 2){
                    $this->cover .= preg_replace('/\s+/', '␣', $value->getText());
                }
                else if (preg_match("/\bLEMBAR␣PENGESAHAN\b|\bPERNYATAAN\b|\bKATA␣PENGANTAR\b/", $value->getText())) {
                    $this->otherPage .= preg_replace('/\s+/', '␣', $value->getText());
                }
                else if (preg_match("/\bABSTRAK\b|\bABSTRACT\b/", $value->getText())) {
                    $this->abstractPage .= preg_replace('/\s+/', '␣', $value->getText());
                }
                else if (preg_match("/\bDAFTAR␣ISI\b/", $value->getText())) {
                    $this->tableOfContentPage .= preg_replace('/\s+/', '␣', $value->getText());
                }
                else if (preg_match("/\bDAFTAR␣GAMBAR\b|\bDAFTAR␣TABEL\b|\bDAFTAR␣REFERENSI\b|\bLAMPIRAN\b/", $value->getText())) {
                    $this->listPage .= preg_replace('/\s+/', '␣', $value->getText());
                }
                else if (preg_match("/(BAB␣)[0-9]{1}/", $value->getText())) {
                    $temp= trim(preg_replace('/\s+/', '␣', $value->getText()));
                    $this->contentPage .= preg_replace("/[0-9]*$/", "", $temp);
                }
                else{
                    continue;
                }
            }
        }

        /**
         * Method untuk mendapatkan daftar isi
         * @return nomor bab dan subbab yang ada pada halaman daftar isi
         */
        public function getTableOfContentPage(){
            $temp = preg_replace('/[A-Za-z]/', '', $this->tableOfContentPage);
            $array = explode("␣", $temp);
            for ($i=0 ; $i < sizeof($array) ; $i++ ) {
                if(preg_match('/^[0-9]{1,}$/', $array[$i])){
                    $array[$i] = "";
                }
                if(preg_match('/^.$/', $array[$i])){
                    $array[$i] = "";
                }
            }

            foreach ($array as $key => $value) {
                if(empty($array[$key])){
                    unset($array[$key]);
                }
```

```
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
```

```
    foreach ($checkers as $checker) {
        $all_errors = array_merge($all_errors, $checker->errorChecking($skripsi_extract));
    }

    $counter = 1;
    foreach ($all_errors as $error) {
        echo "\nERROR $counter\n";
        echo "==================\n";
        $counter++;
        foreach ($error as $key => $value) {
            echo "$key: $value\n";
        }
        echo "\n";
    }

?>
```

Listing A.4: KAL02_PrefaceChecker.php

```
<?php

    include_once "Checker.php";

    /*
     * Sub class dari kelas Checker
     * Memeriksa kata pengantar sebelum subbab
     * @author Marcell Trixie
     */
    class KAL02_PrefaceChecker extends Checker{

        /**
         * Method untuk memeriksa kesalahan dalam dokumen skripsi
         * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
         * @return result[] laporan kesalahan yang ditemukan
         */
        public function errorChecking($pdf_extract){
            $result = [];
            $sentence = $pdf_extract->splitContentPage();
            foreach ($sentence as $index => $value) {
                $pattern = "/^(BAB )[0-9]{1}[\sA-Z]{1,}[0-9]+/";
                if (preg_match($pattern, $value)) {
                    $result[] = [
                        "Error Code" => "KAL-02",
                        "Note" => "Berilah kata pengantar untuk setiap bab",
                        "Excerpt" => $value
                    ];
                }
            }
            return $result;
        }

    }

?>
```

Listing A.5: KAL03_ThesisDataChecker.php

```
<?php

    /*
     * Sub class dari kelas Checker
     * Memeriksa kelengkapan data skripsi
     * @author Marcell Trixie
     */
    class KAL03_ThesisDataChecker extends Checker{

        /**
         * Method untuk memeriksa kesalahan dalam dokumen skripsi
         * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
         * @return result[] laporan kesalahan yang ditemukan
         */
        public function errorChecking($pdf_extract){
            $result = [];
            $cover = $pdf_extract->getCoverPage();
            $pattern = "/JUDUL BAHASA INDONESIA|JUDUL BAHASA INGGRIS|Nama Lengkap|10 digit NPM UNPAR|tahun/";
            if (preg_match($pattern, $cover)) {
                $result[] = [
                    "Error Code" => "KAL-03",
                    "Note" => "Ada data skripsi yang belum dilengkapi pada halaman cover",
                    "Excerpt" => $cover
                ];
            }
            return $result;
        }

    }

?>
```

Listing A.6: NAT01_ReferenceChecker.php

```
<?php

    /*
     * Sub class dari kelas Checker
```

```php
     * Memeriksa subbab dalam sebuah bab
     * @author Marcell Trixie
     */
    class NAT01_ReferenceChecker extends Checker{

        /**
         * Method untuk memeriksa kesalahan dalam dokumen skripsi
         * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
         * @return result[] laporan kesalahan yang ditemukan
         */
        public function errorChecking($pdf_extract){
            $result = [];
            $sentence = $pdf_extract->splitContentPage();
            foreach ($sentence as $index => $value) {
                $pattern = "/\B\[ \?]\B/";
                if (preg_match($pattern, $value)) {
                    $result[] = [
                        "Error Code" => "NAT-01",
                        "Note" => "Referensi tidak dirujuk dengan baik, lakukan perintah PDFLatex->BibTex->PDFLatex->PDFLatex untuk memperbaikinya",
                        "Excerpt" => $value
                    ];
                }
            }
            return $result;
        }


    }
?>
```

Listing A.7: PS01_TypoChecker.php

```php
<?php

    /*
     * Sub class dari kelas Checker
     * Memeriksa kesalahan penulisan kata
     * @author Marcell Trixie
     */
    class PS01_TypoChecker extends Checker{

        /**
         * Method untuk memeriksa kesalahan dalam dokumen skripsi
         * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
         * @return result[] laporan kesalahan yang ditemukan
         */
        public function errorChecking($pdf_extract){
            $filename = __DIR__ . "/../../dictionary/id_ID/id_ID.dic";
            $file = fopen($filename, "r");
            $temp = explode("\n", fread($file,filesize($filename)));

            $dictionary = [];
            for ($i = 0; $i < sizeof($temp); $i++) {
                $dictionary[$i] = preg_replace('/\/.+$/', '', $temp[$i]);
            }

            $word = preg_split("/[^A-Za-z]/", $pdf_extract->getContentPage());
            $typos = [];
            foreach ($word as $index => $value) {
                if (!in_array(strtolower($value), $dictionary) && !in_array($value, $typos)) {
                    $typos[] = $value;
                }
            }

            $result = [];
            if (sizeof($typos) > 0) {
                $result[] = [
                    "Error Code" => "PS-01",
                    "Note" => "Ditemukan penulisan kata yang tidak sesuai dengan kamus",
                    "Excerpt" => "\n\n" . implode(", ", $typos)
                ];
            }
            return $result;
        }

    }

?>
```

Listing A.8: PS03_SpaceChecker.php

```php
<?php

    /*
     * Sub class dari kelas Checker
     * Memeriksa penggunaan spasi setelah tanda baca
     * @author Marcell Trixie
     */
    class PS03_SpaceChecker extends Checker{

        /**
         * Method untuk memeriksa kesalahan dalam dokumen skripsi
         * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
         * @return result[] laporan kesalahan yang ditemukan
         */
```

```
        public function errorChecking($pdf_extract){
            $result = [];
            $sentence = $pdf_extract->splitContentPage();
            foreach ($sentence as $index => $value) {
                $pattern = "/[A-Za-z]{4,}[,.!?][A-Za-z]+/";
                if (preg_match($pattern, $value)) {
                    $result[] = [
                        "Error_Code" => "PS-03",
                        "Note" => "Perhatikan spasi setelah tanda baca.",
                        "Excerpt" => $value
                    ];
                }
            }
            return $result;
        }

    }
?>
```

Listing A.9: PS05_CapitalLetterChecker.php

```
<?php

    /*
     * Sub class dari kelas Checker
     * Memeriksa penggunaan huruf kapital pada karakter awal kalimat
     * @author Marcell Trixie
     */
    class PS05_CapitalLetterChecker extends Checker{

        /**
         * Method untuk memeriksa kesalahan dalam dokumen skripsi
         * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
         * @return result[] laporan kesalahan yang ditemukan
         */
        public function errorChecking($pdf_extract){
            $result = [];
            $sentence = $pdf_extract->splitContentPage();
            foreach ($sentence as $index => $value) {
                $pattern = "/^[a-z]+ |_[a-z]+\.[a-z]/";
                if (preg_match($pattern, $value)) {
                    if(preg_match("/^\W/", $value)){
                        continue;
                    }
                    else{
                        $result[] = [
                            "Error_Code" => "PS-05",
                            "Note" => "Huruf pertama pada kalimat ini tidak menggunakan huruf kapital",
                            "Excerpt" => $value
                        ];
                    }
                }
            }
            return $result;
        }

    }
?>
```

Listing A.10: PS09_SubChapterChecker.php

```
<?php

    /*
     * Sub class dari kelas Checker
     * Memeriksa jumlah subbab dalam sebuah bab
     * @author Marcell Trixie
     */
    class PS09_SubChapterChecker extends Checker{

        /**
         * Method untuk memeriksa kesalahan dalam dokumen skripsi
         * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
         * @return result[] laporan kesalahan yang ditemukan
         */
        public function errorChecking($pdf_extract){
            $result = [];
            $temp = $pdf_extract->getTableOfContentPage();
            $array = array_merge($temp);
            for ($i=0;  $i<sizeof($array); $i++) {
                if(preg_match('/\.1$/', $array[$i])){
                    $substring = substr($array[$i+1], 0, strlen($array[$i])-1);
                    if(!array_search($substring.'2', $array)){
                        $result[] = [
                            "Error_Code" => "PS-09",
                            "Note" => "Bab/Subbab " . $array[$i] . " ini hanya terdapat 1 sub_bab/sub sub_bab",

                        ];
                    }
                }
            }
            return $result;
        }

    }
```

```
34
35 ?>
```

Listing A.11: VAN03_SubjectPronounChecker.php

```php
<?php

    /*
     * Sub class dari kelas Checker
     * Memeriksa penggunaan kata ganti orang
     * @author Marcell Trixie
     */
    class VAN03_SubjectPronounChecker extends Checker{

        /**
         * Method untuk memeriksa kesalahan dalam dokumen skripsi
         * @param pdf_extract untuk memanggil method getter kelas Checker sesuai kebutuhan
         * @return result[] laporan kesalahan yang ditemukan
         */
        public function errorChecking($pdf_extract){
            $result = [];
            $sentence = $pdf_extract->splitContentPage();
            foreach ($sentence as $index => $value) {
                $pattern = "/\bsaya\b|\bkamu\b|\bdia\b/i";
                if (preg_match($pattern, $value)) {
                    $result[] = [
                        "Error Code" => "VAN-03",
                        "Note" => "Kalimat ini mengandung kata ganti orang",
                        "Excerpt" => $value
                    ];
                }
            }
            return $result;
        }

    }

?>
```

LAMPIRAN B

TEST CASE PENGUJIAN FUNGSIONAL

SKRIPSI

APLIKASI PEMERIKSA KESALAHAN DOKUMEN SKRIPSI INFORMATIKA UNPAR

Marcell Trixie Alexander

NPM: 2014730003

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

UNDERGRADUATE THESIS

GENERAL ERROR CHECKER APPLICATION FOR INFORMATICS ENGINEERING UNPAR THESIS DOCUMENT

Marcell Trixie Alexander

NPM: 2014730003

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

LEMBAR PENGESAHAN

APLIKASI PEMERIKSA KESALAHAN DOKUMEN SKRIPSI INFORMATIKA UNPAR

Marcell Trixie Alexander

NPM: 2014730003

Bandung, «tanggal» «bulan» 2019

Menyetujui,

Pembimbing Utama	Pembimbing Pendamping
«pembimbing utama/1»	«pembimbing pendamping/2»

Ketua Tim Penguji	Anggota Tim Penguji
«penguji 1»	«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

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

APLIKASI PEMERIKSA KESALAHAN DOKUMEN SKRIPSI INFORMATIKA UNPAR

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

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

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

Meterai
Rp. 6000

Marcell Trixie Alexander
NPM: 2014730003

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

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

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

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

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

KATA PENGANTAR

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

Bandung, «bulan» 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

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

1.1 Latar Belakang

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

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

1.2 Rumusan Masalah

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

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

1.3 Tujuan

Tujuan dari skripsi ini adalah sebagai berikut:

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

BAB 2

LANDASAN TEORI

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

2.1 *Regular Expression*

Regular expression (*regex*) [1] adalah jenis pola teks tertentu yang dapat digunakan pada banyak aplikasi modern dan bahasa pemrograman. *Regex* biasanya dimanfaatkan untuk memverifikasi kecocokan antara input dengan pola teks, untuk menemukan teks yang cocok dengan pola dalam teks yang lebih besar, untuk mengganti teks yang cocok dengan pola dengan teks lain atau menyusun ulang bit dari teks yang cocok dan untuk membagi sebuah blok teks menjadi beberapa subteks.

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

Regex dapat digunakan dalam berbagai bahasa pemrograman, salah satunya yaitu, *Perl Compatible Regular Expression* (PCRE). PCRE [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

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

2.1.2 Kelas Karakter

Kelas karakter adalah karakter yang memiliki atribut yang spesifik yang dikelompokan 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:].

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 yang belum bisa ditangani oleh *library* ini. *PdfParser* belum dapat mengekstrak dokumen yang diamankan. Selain itu, *library* ini tidak dapat mendeteksi jenis teks yang dicetak miring, tebal dan bergaris bawah. Hingga saat ini, pengembangan *library PdfParser* masih terus berjalan. *PdfParser* memiliki beberapa kelas yang menjalankan fungsional dari *library ini*. Pada subbab berikut akan dijelaskan beberapa kelas dari *library PdfParser*.

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 **??**:

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 Moertini, MT.

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 3.2 hingga tabel **??**:

Tabel 3.2: 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.

3.2 Keputusan Implementasi Hasil Survei

Pada bagian ini akan dijelaskan tentang keputusan implementasi dari hasil survei. 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 *regex*. Metode yang digunakan untuk mendeteksi kesalahan yaitu dengan *pattern matching*, sehingga hal-hal yang bersifat kontekstual tidak dapat dicek dengan *regex*. Berikut ini adalah hasil keputusan yang telah diambil pada setiap hasil survei di atas:

Tabel 3.3: Tabel keputusan implementasi

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

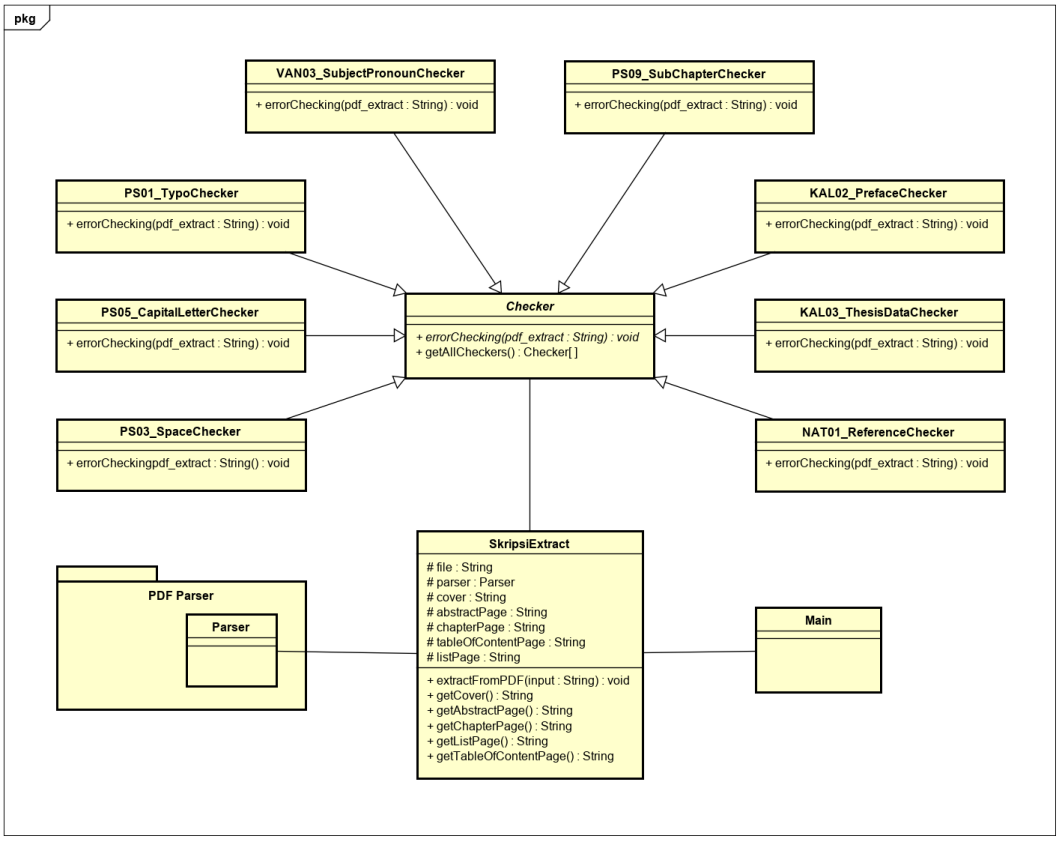
BAB 4

PERANCANGAN

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

4.1 Perancangan Kelas

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



Gambar 4.1: Diagram kelas Aplikasi Pemeriksa Kesalahan Dokumen Skripsi

4.2 Perancangan Perangkat Lunak

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

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.

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 *test case* dokumen skripsi Informatika Unpar dengan mode sidang akhir. Kedua *test case* yang digunakan yaitu, "TC_PF_01.pdf" dan "TC_PF_02.pdf". Isi dari ke-2 file tersebut sama, 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 dimasukan ke dalam *test case* tersebut.

BAB 6

KESIMPULAN DAN SARAN

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

6.1 Kesimpulan

6.2 Saran

DAFTAR REFERENSI

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

[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.

«SKRIPSI/TUGAS AKHIR»

«JUDUL BAHASA INDONESIA»

«Nama Lengkap»

NPM: «10 digit NPM UNPAR»

PROGRAM STUDI «MATEMATIKA/FISIKA/TEKNIK INFORMATIKA»
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
«tahun»

«FINAL PROJECT/UNDERGRADUATE THESIS»

«JUDUL BAHASA INGGRIS»

«Nama Lengkap»

NPM: «10 digit NPM UNPAR»

DEPARTMENT OF «MATHEMATICS/PHYSICS/INFORMATICS»
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
«tahun»

LEMBAR PENGESAHAN

«JUDUL BAHASA INDONESIA»

«Nama Lengkap»

NPM: «10 digit NPM UNPAR»

Bandung, «tanggal» «bulan» «tahun»

Menyetujui,

Pembimbing Utama	Pembimbing Pendamping
«pembimbing utama/1»	«pembimbing pendamping/2»

Ketua Tim Penguji	Anggota Tim Penguji
«penguji 1»	«penguji 2»

Mengetahui,

Ketua Program Studi

«Ketua Program Studi»

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa «skripsi/tugas akhir» dengan judul:

«JUDUL BAHASA INDONESIA»

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

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

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

Meterai
Rp. 6000

«Nama Lengkap»
NPM: «10 digit NPM UNPAR»

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

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

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

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

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

KATA PENGANTAR

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

Bandung, «bulan» «tahun»

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

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

1.1 Latar Belakang

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

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

1.2 Rumusan Masalah

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

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

1.3 Tujuan

Tujuan dari skripsi ini adalah sebagai berikut:

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

BAB 2

LANDASAN TEORI

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

2.1 *Regular Expression*

Regular expression (*regex*) [**?**] adalah jenis pola teks tertentu yang dapat digunakan pada banyak aplikasi modern dan bahasa pemrograman. *Regex* biasanya dimanfaatkan untuk memverifikasi kecocokan antara input dengan pola teks,untuk menemukan teks yang cocok dengan pola dalam teks yang lebih besar, untuk mengganti teks yang cocok dengan pola dengan teks lain atau menyusun ulang bit dari teks yang cocok dan untuk membagi sebuah blok teks menjadi beberapa subteks.

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

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

2.1.1 Metakarakter

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

BAB 3

ANALISIS MASALAH

3.1 Survei Kesalahan Umum

Pda baggian ini akan dijelaskan tentang survei yang dilakukan untuk mengumpulkan informasi yang dibutuhkan dalam pengembangan perangkat lunak. Infrmasi yng dicari adalah tentang ksalahan-kesalahan umumm yaang serring terjaadi paada penulisan dokumen skripsi. Untuk mengumpulkan informasi tersebut, metod ang dipilih adalah melakukan survei. Dalam pelaksanaannya, survei dibagi menjhadi 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 **??**:

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 Moertini, MT.

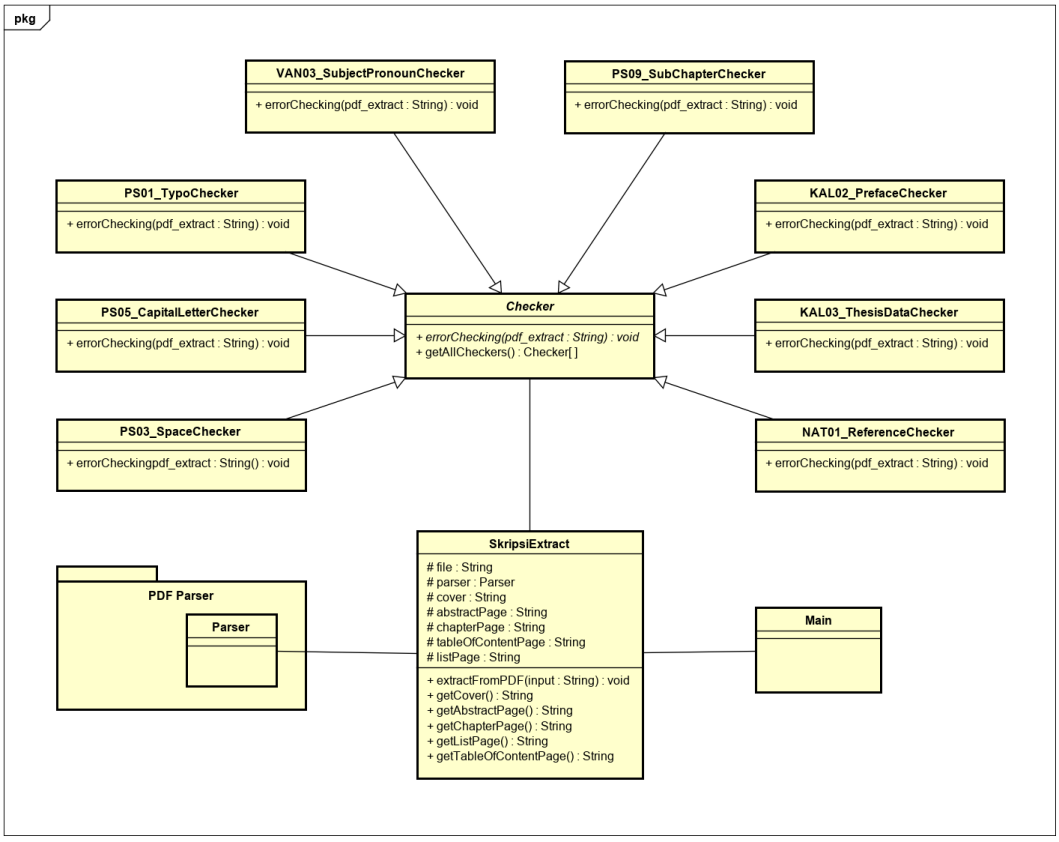
BAB 4

PERANCANGAN

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

4.1 Perancangan Kelas

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



Gambar 4.1: Diagram kelas Aplikasi Pemeriksa Kesalahan Dokumen Skripsi

BAB 5

IMPLEMENTASI DAN PENGUJIAN

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

5.1 Implementasi

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

5.1.1 Lingkungan Implementasi

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

1. Spesifikasi Perangkat Keras
    - Perangkat: Laptop
    - Processor: AMD Bristol Ridge Quad Core FX-9830P 3GHz
    - RAM: 8GB
    - GPU: Radeon RX 460
    - Storage: Harddisk 1TB
2. Spesifikasi Perangkat Lunak
    - Sistem Operasi Windows 10 64-bit
    - PHP 7.3.5 (cli)
    - Composer versi 1.8.5
    - Sublime Text versi 3.2.1

BAB 6

KESIMPULAN DAN SARAN

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

6.1 Kesimpulan

Kesimpulan yang saya dapat dari pengembangan perangkat lunak ini adalah sebagai berikut.

6.2 Saran

Saran yang saya berikan untuk pengembangan perangkat lunak ini adalah sebagai berikut.