BAB IV

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap aplikasi. Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman yang akan digunakan yaitu bahasa pemrograman *java*. Setelah implementasi, maka dilakukan pengujian aplikasi dimana akan dilihat implementasi *class – class* yang dikembangkan dan hasil masing – masing pengujian yang dilakukan

IV.1. Implementasi Sistem

Dalam lingkungan pengembangan, dijelaskan tentang perangkat – perangkat yang digunakan dalam pengembangan aplikasi *text mining* dengan metode *clustering*. Berikut spesifikasi lingkungan pengembangan yang digunakan.

IV.1.1. Perangkat Keras

Spesifikasi komputer yang digunakan dalam pengembangan aplikasi *text mining* dengan metode *clustering* ini adalah sebagai berikut :

1. Processor Intel (R) Celeron (R) CPU N2840 @ 2166GHz 2.16GHz
2. Ram 2 GB
3. VGA OnBoard
4. HDD 500Gb
5. Monitor 14,1”

IV.1.2. Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan aplikasi *text mining* dengan metode *clustering* ini adalah sebagai berikut :

1. Sistem operasi Windows 8
2. *Java* SDK 1.8 (*Java Standard Development Kit*)

   Java SDK 1.8 digunakan untuk membangun aplikasi dengan menggunakan bahasa pemrograman *java*, yang didalmnya terdapat beberapa perangkat pembantu untuk memudahkan *developer* untuk membangun suatu aplikasi bebrbasis *java* diantaranya adalah :
3. *Java Compiler*
4. *Java Virtual Machine/ Java Runtime Environtment* (JRE) 6.
5. *Java Class Libraries*
6. *Java Debugger*
7. Netbeans IDE 8.0
8. Interpreter *Java*

IV.2. Implementasi Modul Program

Seteleh menganalisis dan merancang sistem yang akan dibanguan secara rinci, maka selanjutnya menuju tahap implementasi. Implementasi dilakukan dalam 2 tahap, yaitu implementasi *Graphic User Interface* (GUI) dan implementasi modul program.

IV.2.1. Implementasi Graphic User Interface

Implementasi *Graphic User Interface* (GUI) pada aplikasi ini menggunakan komponen java swing, Tampilan Perancangan Antarmuka terdapat empat komponen dari Swing yang digunakan yaitu :

1. *Top-level Container*, merupakan *container* dasar di mana komponen lainnya diletakan. *Top-level Container* yang digunakan pada aplikasi ini yaitu *Frame* yang diimplementasikan pada kelas MainFrame.
2. *Intermediate Container*, merupakan komponen perantara di mana komponen lainnya akan diletakan. *Intermediate Container* yang digunakan pada aplikasi ini adalah *Jpanel*.
3. *Atomic Component*, merupakan komponen yang memiliki fungsi spesifik dimana *user* akan berinteraksi langsung dengan komponen ini. *Atomic Component* yang digunkan pada aplikasi ini adalah JButton, JLabel, JTextField, JcomboBox dan JTextArea.
4. *Event Handling*, untuk menangani *event* yang dilakukan oleh user, pada aplikasi ini diimplementasikan pada saat *user* menekan tombol, memilih dokumen, klik kanan pada mouse dan pada saat mouse memasuki JtextField.

IV.2.2. Implementasi Modul Program

Modul program yang diimplementasi mencakup program utama dan sub program. Program utama adalah modul yang berfungsi sebagai driver yang akan memanggil sub program. Sub program adalah modul yang merepresentasikan tugas tertentu dari proses *text mining* menggunakan metode *clustering*.

Sesuai dengan *class diagram* yang dijelaskan, pencarian dokumen diimplementasikan melalui kelas MainFrame yang akan melakukan proses transformasi dan clustering dengan memanggil beberapa kelas, yaitu kelas Stopword, kelas Clustering, kelas KNN, kelas KMeans, kelas TfIdf, Kelas GetAbstract. Kegunaan dan fungsi kelas akan dijelaskan pada bab ini.

IV.2.3. Implementasi Kelas MainFrame

Kelas MainFrame adalah kelas utama, dimana sistem akan memanggil terlebih dahulu kelas ini untuk menampilkan *graphic user interface* (GUI) sebagai tampilan aplikasi *Text Mining* menggunakan metode *Clustering*. Kelas ini memiliki beberapa *method* yaitu *method* showClusterPanel() dan *method* showPreparePanel() dan showOutlierPanel(). Fungsi dan kegunaan *method* diuraikan pada Tabel Lampiran 1 Lampiran B Function Class.

IV.2.4. Implementasi Kelas Transformasi

Kelas Transformasi adalah kelas yang berfungsi untuk memasukkan dan mendapatkan teks yang berada pada dokumen yang memiliki format *.pdf. Untuk dapat membaca dan menyimpan teks dokumen diimplementasikan *java class* library yaitu library apache PDFBox.

Kelas ini memiliki beberapa *method* yaitu *method* readFiles(String fileName), *method* readDoc(String fileName), *method* readPdf(String fileName), *method* getFile(String path), *method* checkRead(String path, String namaFile) dan *method* checkReadOnlyPath(String path). Fungsi dan kegunaan *method* diuraikan pada Tabel Lampiran 2 Lampiran B Function Class.

IV.2.5. Implementasi Kelas Clustering

Kelas Clustering adalah kelas yang berfungsi untuk mengelompokkan document – document yang sudah di transformasi ke format *.Txt. kelas ini memiliki beberapa *method* yaitu *method* mainController, *method* getkMeans(), *method* getAbstractSkripsi() dan *method* showFileList(). Fungsi dan kegunaan *method* diuraikan pada Tabel Lampiran 4 Lampiran B Function Class.

IV.2.6. Implementasi Kelas AnalisisOutlier

Kelas AnalisisOutlier adalah kelas yang berfungsi untuk menampilkan hasil clustering dan menganalisis document yang dihitung sebagai outlier. Kelas ini memiliki beberapa *method* yaitu *method* initComponents(), *method*

setClusterTableValue(), *method* initTableListener(). Fungsi dan kegunaan *method* diuraikan pada Tabel Lampiran 5 Lampiran B Function Class.

IV.3. Pengujian Sistem

Pada pengujian sistem ini bertujuan untuk memeriksa aplikasi dalam menjalankan fungsionalitas yang telah dirancang dan dibangun. Tujuan dari proses pengujian ini dilakukan untuk memeriksa apakah aplikasi Text Mining dengan metode Clustering ini telah melakukan fungsionalitas sesuai dengan awal perancangan pembuatan. Adapun tahap pengujian sistem terdiri dari rencana pengujian, skenario pengujian, hasil pengujian dan evaluasi.

IV.3.1. Rencana Pengujian

Pengujian terhadap fungsionalitas perangkat lunak ini menggunakan metode *Black Box.* Pengujian ini merupakan pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Tujuan dari metode *Black Box* ini adalah untuk menemukan kesalahan fungsi pada perangkat lunak. Pengujian ini dilakukan dengan mencoba semua kemungkinan yang terjadi dan dilakukan berurang-ulang.

IV.3.2. Skenario Pengujian

Skenario pengujian memaparkan urutan dan hal yang diuji pada aplikasi *Text Mining* ini. Adapun skenario pengujian yang dilakukan adalah sebagai berikut :

Tabel.IV.1. Skenario Pengujian

Item Uji	Butir Uji	Jenis Pengujian
Input Dokumen	Memilih folder kumpulan dokumen yang akan di ekstrak ke txt	*Black Box*
Output Dokumen	Memilih folder dokumen tempat file ekstraksi pdf ke txt	*Black Box*
Read Document	Melihat isi dokumen	*Black Box*
Clustering	Melihat proses tokenizing Memasukkan jumlah kluster	*Black Box*

Outlier	Melihat proses outlier	*Black Box*
Output	Melihat hasil clustering dan outlier	*Black Box*

IV.3.3. Hasil Pengujian

Untuk menguji ketepatan dan keakuratan aplikasi ini, dilakukan pengujian dengan menghitung nilai precision dan nilai recall pada setiap clustering, dimulai dari dokumen yang berjumlah 5 hingga dokumen yang berjumlah 30.

Precision adalah rasio jumlah dokumen relevan yang ditemukan dengan total jumlah dokumen yang ditemukan oleh sistem, dengan rumus precision adalah sebagai berikut**[II.2.7]**.

$$precision = \frac{\text{benar}\, jumlah\ dokumen\ yang\ dicluster\ dengan\ benar}{jumlah\ seluruh\ dokumen} \text{.....(8)}$$

Diketahui :

a : jumlah dokumen relevan

b : jumlah dokumen yang diuji

Precision mengindikasikan kualitas himpunan jawaban, tetapi tidak memandang total jumlah dokumen yang relevan dalam kumpulan dokumen. Sedangkan Recall adalah rasio jumlah dokumen relevan yang ditemukan kembali dengan total jumlah dokumen dalam kumpulan dokumen yang dianggap relevan, dengan rumus recall adalah sebagai berikut **[II.2.7]**:

$$recall = \frac{jumlah\ dokumen\ yang\ di\ cluster\ dengna\ benar}{jumlah\ dokumen\ yang\ benar} \text{.....(7)}$$

Diketahui :

a : jumlah dokumen relevan

b : jumlah dokumen yang diuji

Dalam percobaan pertama menjalankan aplikasi dengan input dokumen *.pdf 5 dokumen.

Jumlah dokumen yang relevan terambil dari 5 dokumen = 3

Jumlah seluruh dokumen tercluster = 5

Jumlah seluruh dokumen relevan dalam koreksi = 4

Hasil perhitungan recall dan precision :

Precision = 3/5 = 60%

Recall = 3/4 = 75%

Hasil pengujian nilai precision dan recall yang di-*cluster* oleh aplikasi yang telah dilakukan terhadap 5 sampai 30 dokumen dapat dilihat pada tabel berikut.

Tabel.IV.2. Precision dan Recall

No	**Jumlah Dokumen**	**Dokumen Cluster**	**P**	**R**	**Sesuai Cluster**
1	5	5	3/5	¾	3
2	10	10	6/10	6/8	6
3	15	15	9/25	9/12	9
4	20	20	12/20	12/16	12
5	25	25	15250	15/20	15
6	30	30	18/30	18/24	18

IV.3.4. Evaluasi pengujian

Berdasarkan hasil pengujian dengan kasus uji diatas dapat disimpulkan bahwa jika data yang dimasukan benar maka sistem secara fungsional akan mengeluarkan keluaran sesuai dengan harapan. Seperti, inputan dalam tahap transformasi berformat *.Pdf dengan kata lain bahwa pembangunan sistem ini secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.