



TUGAS AKHIR - KI091391

RANCANG BANGUN MODUL PENGELOMPOKAN DOKUMEN PADA SISTEM MANAJEMEN DOKUMEN KEPEGAWAIAN.

**Anita Susanti
NRP 5110100145**

**Dosen Pembimbing
Umi Laili Yuhana, S.Kom., M.Sc.
Nurul Fajrin Ariyani, S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI091391

DESIGN OF DOCUMENT GROUPING MODULE IN ADMINISTRATION INFORMATION SYSTEM MANAGEMENT

**Anita Susanti
NRP 5110100145**

**Advisor
Umi Laili Yuhana, S.Kom., M.Sc.
Nurul fajrin Ariyani, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2014**

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN MODUL PENGELOMPOKAN DOKUMEN PADA SISTEM MANAJEMEN DOKUMEN KEPEGAWAIAN

Nama Mahasiswa : Anita Susanti
NRP : 5110100145
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Umi Laili Yuhana, S.Kom, M.Sc.
Dosen Pembimbing 2 : Nurul Fajrin Ariyani, S.Kom, M.Sc.

ABSTRAK

Saat ini sudah banyak sekali dokumen-dokumen yang dipublikasikan di internet maupun media lainnya. Dokumen merupakan salah satu produk yang dihasilkan dalam sebuah tatanan organisasi, mulai dari perencanaan, pelaksanaan, laporan dan evaluasi hampir seluruhnya menghasilkan dokumen. Dengan banyaknya dokumen yang dihasilkan, maka manajemen dokumen menjadi peranan yang sangat penting.

Untuk mengatasi masalah tersebut, maka akan digunakan metode klasifikasi untuk melakukan pengelompokan dokumen kepegawaian ke dalam sejumlah kategori. Metode pengelompokan yang digunakan adalah metode klasifikasi Naive Bayes yang memungkinkan sebuah dokumen memiliki beberapa kategori. Dimana dokumen diolah dengan pra proses dan pembobotan TF sehingga dapat melakukan proses pelatihan dokumen yang diperlukan untuk mendapatkan model yang nantinya berperan pada proses testing untuk mengelompokkan dokumen. Aplikasi telah diuji dan menghasilkan nilai presisi 0,89 dan nilai recall 0,83. Hasil tersebut menunjukkan bahwa aplikasi dapat digunakan untuk mengelompokkan dokumen berdasarkan kategori.

Kata kunci: Manajemen dokumen, naïve bayes, klasifikasi, dokumen.

[Halaman ini sengaja dikosongkan]

Design Of Document Grouping Module In Administration Information System Management

Student Name : Anita Susanti
Student ID : 5110100145
Major : Teknik Informatika FTIf-ITS
Advisor 1 : Umi Laili Yuhana, S.Kom., M.Sc.
Advisor 2 : Nurul Fajrin Ariyani, S.Kom, M.Sc.

ABSTRACT

Nowadays, there are many documents was published on internet or other media. Document is one of product which produced by organization, start from planning, implementation, report and evaluation, almost entirely producing document. Because of many produced documents, so that document management becomes very important.

To resolve this problem, so it will be used classification method for classify administration document into a number of categories. Classification method that used in this application is Naive Bayes classification method which allows a document have multiple categories where used in testing process of document classification. Application have been tested and results precision values 0.89 and recall value 0.83. The result shows that application can be used to classify document based on category.

Keyword: document management, naive bayes, classification, document

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**Rancang Bangun Modul Pengelompokan Dokumen Pada
Sistem Manajemen Dokumen Kepegawaian**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :
ANITA SUSANTI
NRP : 5110 100 145

Disetujui oleh Dosen Pembimbing tugas akhir :

UMI LAILI YUHANA, S.Kom., M.Sc.
NIP: 197906262005012002

(pembimbing 1)

**NURUL FAJRIN ARIYANI, S.Kom.,
M.Sc.**
NIP: 051100124

(pembimbing 2)

**SURABAYA
JULI 2014**

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul :

“Rancang Bangun Modul Pengelompokan Dokumen pada Sistem Manajemen Dokumen Kepegawaian ”

Harapan dari penulis semoga apa yang tertulis di dalam buku tugas akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan tugas akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak, Ibu, Adik dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan tugas akhir ini.
2. Ibu Umi Laili Yuhana dan Ibu Nurul Fajrin Ariyani selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
3. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Pak Yudi, Pak Sugeng serta seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Arifin yang selalu ada untuk membantu, memberikan semangat, dan hiburan ketika penulis merasa jenuh.
6. Teman-teman penghuni Lab Rekayasa Sistem Berbasis Pengetahuan, Aida, Nurul, Mira, Ika, Linda, Jarwe, Nanta, Orysa dan Tommy yang telah memberikan banyak dukungan dan semangat kepada penulis.

7. Keluarga Lab *Intelligent System* yang telah menemani penulis selama berkuliah di Teknik Informatika ITS.
8. Munif, Fajri, Arya, Monika yang telah membantu penulis selama menyelesaikan tugas akhir.
9. Teman-teman seperjuangan GEMASTIK dan PKM Smartgarmen Dany, Hardika, Sasa dan Helmy atas kebersamaanya.
10. Teman-teman angkatan 2010 jurusan Teknik Informatika ITS yang telah menemani perjuangan selama 4 tahun ini atas saran, masukan, dan dukungan terhadap pengerjaan tugas akhir ini.
11. Serta pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu penulis dalam penyusunan tugas akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juli 2014

Anita Susanti

DAFTAR ISI

LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER.....	xxi
DAFTAR PERSAMAAN	xxiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Tujuan.....	2
1.3. Rumusan Permasalahan.....	2
1.4. Batasan Permasalahan	2
1.5. Metodologi	2
1.6. Sistematika Penulisan	4
BAB II DASAR TEORI.....	7
2.1. Sistem Manajemen Dokumen.....	7
2.2. Klasifikasi Dokumen	7
2.3. Dokumen Kepegawaian.....	8
2.4. Klasifikasi Naive Bayes	8
2.5. WEKA	10
2.6. Pembobotan TF	10
2.7. Crawl Portal.....	10
2.8. Text Mining.....	11
2.9. Presisi dan Recall	12
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	15
3.1. Analisis	15
3.1.1. Analisis Permasalahan.....	15
3.1.2. Deskripsi Umum Sistem.....	16
3.1.3. Spesifikasi Kebutuhan Perangkat Lunak	16
3.1.4. Aktor.....	16

3.1.5.	Kasus Penggunaan.....	17
3.2.	Perancangan Perangkat Lunak.....	29
3.2.1.	Perancangan Basis Data.....	29
3.2.2.	Perancangan Diagram Kelas.....	31
3.2.3.	Perancangan Arsitektur Sistem.....	32
3.2.4.	Perancangan Proses	34
3.2.5.	Perancangan Antarmuka.....	37
BAB IV	IMPLEMENTASI	41
4.1.	Implementasi Lapisan Antarmuka.....	41
4.2.	Implementasi Lapisan Kontrol	41
4.2.1.	Kelas Pencarian	41
4.2.2.	Kelas Main.....	42
4.2.3.	Kelas Admin	42
4.2.4.	Kelas TextClassifier	43
4.3.	Implementasi Lapisan Data	43
4.3.1.	Kelas Dokumen	43
4.3.2.	Kelas mdlAdmin.....	44
4.4.	Implementasi Crawl.....	44
4.5.	Implementasi Antarmuka Pengguna.....	44
4.5.1.	Tampilan Halaman Beranda	44
4.5.2.	Tampilan Halaman Pencarian Dokumen	46
4.5.3.	Tampilan Halaman Beranda Administrator Utama	46
4.5.4.	Tampilan Halaman Beranda Administrator	
Tambahan		47
BAB V	PENGUJIAN DAN EVALUASI	49
5.1.	Lingkungan Pengujian.....	49
5.2.	Skenario Pengujian	49
5.2.1.	Pengujian Fungsionalitas	50
5.2.2.	Pengujian Kegunaan Sistem	56
5.3.	Skenario Pengujian	59
5.3.1.	Evaluasi Pengujian Fungsionalitas	59
5.3.2.	Evaluasi Hasil Pengelompokan	59
5.3.3.	Evaluasi Pengujian Kegunaan Sistem	61
BAB VI	KESIMPULAN DAN SARAN.....	63

6.1. Kesimpulan.....	63
6.2. Saran.....	63
DAFTAR PUSTAKA.....	65
Lampiran A. Diagram Alir	67
Lampiran B.Kode Sumber.....	70
Lampiran C. Hasil pengujian.....	82
BIODATA PENULIS.....	95

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 3.1 Diagram Kasus Penggunaan	18
Gambar 3.2 Diagram Aktivitas Melakukan Pencarian Dokumen Kepegawaian	20
Gambar 3.3. Diagram Sekuens Melakukan Pencarian Dokumen Kepegawaian	21
Gambar 3.4. Diagram Aktivitas Mengunggah Dokumen Kepegawaian	23
Gambar 3.5. Diagram Sekuens Mengunggah Dokumen Kepegawaian	24
Gambar 3.6 Diagram Aktivitas Menambah Admin.....	26
Gambar 3.7 Diagram Sekuens Menambah Admin	26
Gambar 3.8 Diagram Aktivitas Mengganti <i>Password</i>	28
Gambar 3.9 Diagram Sekuens Mengganti <i>Password</i>	28
Gambar 3.10 Conceptual Data Model (CDM) Sistem Manajemen Dokumen Kepegawaian.....	29
Gambar 3.11 Physical Data Model (CDM) Sistem Manajemen Dokumen Kepegawaian.....	30
Gambar 3.12 Diagram Kelas Sistem Manajemen Dokumen Kepegawaian	32
Gambar 3.13 Arsitektur Sistem Manajemen Dokumen Kepegawaian	34
Gambar 3.14 Rancangan Antarmuka Halaman Utama.....	38
Gambar 3.15 Rancangan Antarmuka Halaman Pencarian Dokumen	39
Gambar 3.16 Rancangan Antarmuka Halaman Beranda Administrator Utama	40
Gambar 3.17 Rancangan Antarmuka Halaman Beranda Administrator Tambahan.....	40
Gambar 4.1 Tampilan Halaman Beranda	45
Gambar 4.2 Tampilan Tombol Pencarian.....	45
Gambar 4.3 Tampilan Halaman Pencarian Dokumen	46
Gambar 4.4 Tampilan Halaman Beranda Administrator Utama ..	47

Gambar 4.5 Tampilan Halaman Beranda Administrator Tambahan	47
Gambar 5.1. Hasil Uji Melakukan Pencarian Dokumen Skenario 1	51
Gambar 5.2 Hasil Uji Fitur Mengunggah Dokumen	53
Gambar 5.3 Hasil Uji Fitur Menambah Administrator	54
Gambar 5.4 Hasil Uji Fitur Mengganti <i>Password</i>	56
Gambar 5.5 Grafik Hasil Survei Tingkat Kegunaan Aplikasi dalam Membantu Pencarian Dokumen	57
Gambar 5.6 Grafik Hasil Survei Tingkat Akurasi Pengelompokan Dokumen	57
Gambar 5.7 Grafik Hasil Survei Tingkat Kemudahan Akses.....	58
Gambar 5.8 Grafik Hasil Survei Penilaian Fitur Aplikasi	58
Gambar 9.1. Hasil Uji Melakukan Pencarian Dokumen Skenario 2	83
Gambar 9.2 Hasil Uji Melakukan Pencarian Dokumen Skenario 3	84
Gambar 9.3 Hasil Uji Mengunggah Dokumen Skenario 2.....	86
Gambar 9.4 Hasil Uji Mengunggah Dokumen Skenario 3.....	87
Gambar 9.5 Hasil Uji Menambah Administrator Skenario 2	89

DAFTAR KODE SUMBER

Kode Sumber 8.1 Fungsi <i>Index</i> pada Kelas Pencarian	70
Kode Sumber 8.2 Fungsi <i>Download</i> pada Kelas Pencarian	71
Kode Sumber 8.3 Fungsi Kategori pada Kelas Pencarian	71
Kode Sumber 8.4 Fungsi <i>Index</i> pada Kelas Main	72
Kode Sumber 8.5 Fungsi Hasil pada Kelas Main	72
Kode Sumber 8.6 Fungsi <i>do_upload</i> pada Kelas Admin	72
Kode Sumber 8.7 Fungsi <i>gantiPassword</i> pada Kelas Admin	73
Kode Sumber 8.8 Fungsi <i>tambahAdmin</i> pada Kelas Admin	74
Kode Sumber 8.9 Fungsi <i>Pelatihan</i> pada Kelas <i>TextClassifier</i>	75
Kode Sumber 8.10 Fungsi <i>Testing</i> pada Kelas <i>TextClassifier</i>	76
Kode Sumber 8.11 Kelas <i>Crawl</i>	78

[Halaman ini sengaja dikosongkan]

DAFTAR PERSAMAAN

Persamaan	
(2.1) Peluang Dokumen.....	9
(2.2) Peluang <i>Term</i>	9
(2.3) Perhitungan Klasifikasi Naïve Bayes.....	9
(2.4) Pembobotan TF.....	10
(2.5) Presisi.....	12
(2.6) <i>Recall</i>	12

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Tabel Kode Kasus Penggunaan	17
Tabel 3.2. Spesifikasi Kasus Penggunaan Melakukan Pencarian Dokumen Kepegawaian.....	19
Tabel 3.3 Spesifikasi Kasus Penggunaan Mengunggah Dokumen Kepegawaian	22
Tabel 3.4 Spesifikasi Kasus Penggunaan Menambah Admin	25
Tabel 3.5 Spesifikasi Kasus Penggunaan Mengganti <i>Password</i>	27
Tabel 3.6 Penjelasan Tabel Kategori.....	30
Tabel 3.7 Penjelasan Tabel Admin.....	31
Tabel 3.8 Penjelasan Tabel Jenis Admin.....	31
Tabel 5.1 Pengujian Fitur Melakukan Pencarian Dokumen Skenario 1	50
Tabel 5.2 Pengujian Fitur Mengunggah Dokumen Skenario 1	52
Tabel 5.3 Pengujian Fitur Menambah Administrator	53
Tabel 5.4 Pengujian Fitur Mengganti <i>Password</i>	55
Tabel 5.5 Rangkuman Hasil Pengujian	59
Tabel 5.6 Hasil Perhitungan Presisi dan Recall Pengelompokan Dokumen	60
Tabel 9.1. Pengujian Fitur Melakukan Pencarian Dokumen Skenario 2.....	82
Tabel 9.2 Pengujian Fitur Melakukan Pencarian Dokumen Skenario 3.....	83
Tabel 9.3 Pengujian Fitur Mengunggah Dokumen Skenario 3	84
Tabel 9.4. Pengujian Fitur Mengunggah Dokumen Skenario 3	86
Tabel 9.5 Pengujian Menambah Administrator Skenario 2.....	88
Tabel 9.6 Profil Responden Pengguna Aplikasi	89

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Saat ini sudah banyak sekali dokumen-dokumen yang dipublikasikan di internet maupun media lainnya. Orang atau institusi seperti perpustakaan yang membutuhkan dapat mencari dokumen dari mana saja. Setelah mencari dokumen, kemudian perlu dipilah-pilah agar penyimpanan dokumen lebih terorganisir.

Dokumen adalah surat berharga yang tertulis atau tercetak yang dapat dipakai sebagai bukti atau keterangan. Dokumen merupakan salah satu produk yang dihasilkan dalam sebuah tatanan organisasi, mulai dari perencanaan, pelaksanaan, laporan dan evaluasi hampir seluruhnya menghasilkan dokumen. Selain itu rapat, presentasi, kebijakan, kajian maupun penelitian juga menghasilkan dokumen. Dengan banyaknya dokumen yang dihasilkan tersebut, maka manajemen dokumen memiliki peranan yang sangat penting. [1]

Dokumen kepegawaian sangat erat hubungannya dengan pegawai institusi, baik itu struktural maupun fungsional. Dokumen informasi yang dibutuhkan antara pegawai struktural dan fungsional pastilah berbeda. Akan tetapi, pegawai kesulitan untuk mendapatkan dokumen terkait. Disebabkan oleh beberapa hal diantaranya kurangnya informasi ataupun perubahan peraturan yang mendadak.

Untuk mengatasi masalah tersebut, maka akan digunakan metode klasifikasi untuk melakukan pengelompokan dokumen kepegawaian ke dalam sejumlah kategori. Klasifikasi dokumen merupakan masalah yang mendasar namun sangat penting karena manfaatnya cukup besar mengingat jumlah

dokumen yang ada setiap hari semakin bertambah. Sebuah dokumen dapat dikelompokkan ke dalam kategori tertentu berdasarkan kata-kata dan kalimat yang terdapat di dalam dokumen tersebut. Tujuan dari tugas akhir ini adalah untuk membuat sistem pengelompokan dokumen kepegawaian dengan beberapa kategori.

1.2. Tujuan

Tujuan dari tugas akhir ini adalah membuat aplikasi yang dapat melakukan pengelompokan dokumen secara otomatis ke dalam kategori yang telah ditentukan.

1.3. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini adalah bagaimana membuat aplikasi yang dapat melakukan pengelompokan dokumen secara otomatis ke dalam kategori yang telah ditentukan.

1.4. Batasan Permasalahan

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Data masukan berupa *file* pdf akan tetapi tidak termasuk *file* pdf hasil pemindaian;
2. Data yang digunakan untuk pembelajaran yaitu dokumen kepegawaian dan dokumen yang didapatkan dari website yang memuat dokumen kepegawaian;
3. Kategori dokumen sudah ditentukan antara lain: dosen, tenaga kependidikan dan jabatan.

1.5. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini yaitu:

1. Studi literatur

Pada tahap ini dilakukan pengumpulan informasi mengenai dokumen kepegawaian. Mengumpulkan dan menggali

informasi dan literatur yang diperlukan dalam proses perancangan dan implementasi sistem yang dibangun. Literatur yang digunakan adalah sebagai berikut:

- a. konsep *crawl* portal;
- b. teori pembobotan TF;
- c. teori klasifikasi naive bayes.

2. Analisis dan Perancangan Sistem

Pada tahap ini dilakukan analisis dan pendefinisian kebutuhan sistem untuk masalah yang sedang dihadapi. Selanjutnya, dilakukan perancangan sistem dengan beberapa tahap sebagai berikut:

- a. perancangan proses aplikasi;
- b. perancangan antar muka sistem;
- c. perancangan diagram kelas sistem;
- d. perancangan basis data sistem; dan
- e. perancangan arsitektur sistem.

3. Implementasi

Pada tahap ini dilakukan pembuatan elemen perangkat lunak. Sistem yang dibuat berpedoman pada rancangan yang telah dibuat pada proses perancangan dan analisis sistem. Dalam pembangunan perangkat lunak ini, penulis akan menggunakan bahasa pemrograman php dengan *framework* CodeIgniter dan *database* MySQL.

4. Pengujian dan evaluasi

Pada tahap ini dilakukan pengujian terhadap perangkat lunak berdasarkan skenario yang telah ditentukan. Pengujian ini bertujuan untuk menguji kebutuhan fungsional yang dibutuhkan, masalah yang timbul, kekurangan program kegunaan dan kemudahan pengguna dalam menggunakan aplikasi.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang

telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.6. Sistematika Penulisan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka pada kanvas.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan perangkat lunak.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat hasil keluaran yang dihasilkan oleh aplikasi, dan evaluasi untuk mengetahui kemampuan aplikasi.

Bab VI Kesimpulan

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan tugas akhir. Teori-teori tersebut meliputi pembobotan TF, klasifikasi naive bayes, Weka, *crawl* portal dan *text mining*.

2.1. Sistem Manajemen Dokumen

Sistem manajemen dokumen adalah suatu sistem pengelolaan dokumen yang berfungsi untuk mengatur dan mengelola surat-surat ataupun dokumen-dokumen penting agar dokumen tersebut mudah untuk dicari dan ditemukan kembali. Sistem manajemen dokumen bermanfaat dalam mengefektifkan dan mengefisienkan proses bisnis. Manfaat yang utama adalah pengguna dapat menemukan informasi yang dibutuhkan dengan cepat, sehingga dapat membantu proses menjadi lebih cepat, baik dan mudah [2].

Secara lebih spesifik lagi Sistem Manajemen Dokumen dapat memberikan manfaat sebagai berikut:

- a. Meningkatkan produktivitas
- b. Meningkatkan waktu respon
- c. Mengurangi biaya dokumen dan meningkatkan efisiensi ruang penyimpanan
- d. Mengurangi resiko kehilangan ataupun kerusakan dokumen
- e. Keamanan dokumen yang handal

2.2. Klasifikasi Dokumen

Klasifikasi dokumen adalah proses pengelompokan dokumen sesuai dengan kategori yang dimilikinya. Klasifikasi dokumen merupakan masalah yang mendasar namun sangat penting karena manfaatnya cukup besar mengingat jumlah dokumen yang ada setiap hari semakin bertambah. Sebuah dokumen dapat dikelompokkan ke dalam kategori tertentu

berdasarkan kata-kata dan kalimat yang terdapat di dalam dokumen tersebut. Kata atau kalimat yang terdapat di dalam sebuah dokumen memiliki makna tertentu dan dapat digunakan sebagai dasar untuk menentukan kategori dari dokumen itu sendiri [3].

Manfaat dari klasifikasi dokumen untuk pengorganisasian dokumen. Dengan jumlah dokumen yang besar, untuk mencari sebuah dokumen akan lebih mudah apabila kumpulan dokumen yang dimiliki terorganisir dan telah dikelompokkan sesuai kategorinya masing-masing.

2.3. Dokumen Kepegawaian

Dokumen kepegawaian adalah dokumen dari Pegawai Negeri Sipil berupa surat keputusan di bidang kepegawaian yang dikeluarkan oleh Pejabat yang berwenang. Dokumen kepegawaian yang dipakai dalam tugas akhir ini adalah dokumen kepegawaian perguruan tinggi.

Jenis dokumen tersebut meliputi surat keputusan, peraturan, surat rektor, surat edaran menteri maupun informasi dan buku panduan. Jenis dokumen yang dibahas meliputi pengangkatan pegawai, mutasi pegawai, administrasi pegawai, kesejahteraan pegawai, proses pemberhentian pegawai dan dokumentasi pegawai. Dokumen kepegawaian yang dipakai di perguruan tinggi membawahi subjek-subjeknya yaitu pegawai fungsional dan pegawai struktural. Pegawai fungsional meliputi tenaga pendidik atau dosen dan tenaga kependidikan. Tenaga kependidikan terdiri dari berbagai posisi meliputi staf akademik, staf administrasi, pustakawan, laboran, pranata komputer dan lain sebagainya. Selain pegawai fungsional adapula pegawai struktural yang menempati jabatan pada struktur organisasi perguruan tinggi.

2.4. Klasifikasi *Naive Bayes*

Algoritma pembelajaran bayes nilai k awal menghitung probabilitas eksplisit untuk menggambarkan hipotesa yang dicari. Suatu data pada klasifikasi naive bayes direpresentasikan dengan konjungsi dari nilai-nilai atribut dan sebuah fungsi

target $f(x)$ yang dapat memiliki nilai apapun dari himpunan set domain V [4].

Sistem dilatih menggunakan data latih lengkap berupa pasangan nilai-nilai atribut dan nilai target kemudian sistem akan diberikan sebuah data baru dalam bentuk $\langle a_1, a_2, a_3, \dots, a_n \rangle$ dan sistem diberi tugas untuk menebak nilai fungsi target dari data tersebut [5].

Dalam proses klasifikasi metode naive bayes yang digunakan dalam penelitian ini, ditentukan terlebih dahulu nilai peluang dokumen terhadap setiap kategori, dan nilai peluang tiap term terhadap frekuensi kemunculannya pada tiap kategori menggunakan Persamaan (2.1) dan (2.2).

$$P(V_j) = \frac{D_j}{D} \quad (2.1)$$

$$P(a_k | v_j) = \frac{n_f + 1}{n + n_j} \quad (2.2)$$

Dimana,

$P(V_j)$:Peluang Dokumen berkategori j
$P(a_k v_j)$:Peluang <i>term</i> k pada kategori j
D_j	:Jumlah dokumen yang memiliki kategori j
D	:Jumlah keseluruhan dokumen
n_f	:Nilai frekuensi <i>term</i>
n	:Jumlah keseluruhan <i>term</i>
n_j	:Jumlah keseluruhan <i>term</i> dalam kategori j .

Setelah didapatkan nilai dan $P(V_j)$ dan $P(a_k | v_j)$, maka dapat dilanjutkan dengan proses perhitungan klasifikasi dengan metode naive bayes. Perhitungan klasifikasi naive bayes ditunjukkan pada persamaan (2.3).

$$V_{NB} = \operatorname{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad (2.3)$$

2.5. WEKA

Weka adalah perangkat mesin praktis. Weka merupakan singkatan dari *Waikato Environment for knowledge analysis*, yang dibuat di Universitas Waikato, New Zealand. Weka mampu menyelesaikan masalah-masalah data *mining* di dunia nyata. Perangkat lunak ini ditulis dalam hirarki *class* Java dengan metode berorientasi objek dan dapat berjalan hampir disemua *platform*.

Weka mudah digunakan dan diterapkan pada beberapa tingkatan yang berbeda. Weka terdiri dari aplikasi untuk *pre-processing* data, klasifikasi, regresi, *clustering*, aturan asosiasi dan visualisasi. Perangkat yang digunakan *pre-processing* dataset membuat pengguna berfokus pada algoritma yang digunakan tanpa terlalu memperhatikan detail seperti pembacaan data dari *file-file*, implementasi algoritma filtering dan penyediaan kode untuk evaluasi hasil.

2.6. Pembobotan TF

Pembobotan TF adalah pembobotan yang mengitung frekuensi kemunculan term (t) pada dokumen (d). Cara perhitungan TF adalah sebagai berikut:

Misalkan n adalah jumlah dokumen yang terkumpul dan kata kunci k_i muncul pada sejumlah n_i dokumen. Kemudian misalkan $f_{i,j}$ adalah banyaknya kata kunci k_i yang ditemukan pada dokumen d_j . Maka $TF_{i,j}$, yaitu *term frequency* dari kata kunci k_i pada dokumen d_j dapat didefinisikan pada Persamaan (2.4).

$$TF = \frac{f_{i,j}}{\text{length}(d_j)} \quad (2.4)$$

2.7. Crawl Portal

Crawl Portal adalah proses mengambil kumpulan halaman dari sebuah web untuk dilakukan pengindeksan sehingga mendukung kinerja mesin pencari. Tujuan dari proses *crawl* adalah

mempercepat dan mengefisienkan kumpulan halaman *web* dengan cepat, mengumpulkan struktur tautan yang menghubungkan kumpulan halaman web tersebut.

Crawl portal juga sering dikenal sebagai *web spider* atau *web robot* adalah salah satu komponen penting dalam sebuah mesin pencari modern. Fungsi utama *web crawler* adalah untuk melakukan penjelajahan dan pengambilan halaman-halaman *web* yang ada di Internet. Hasil pengumpulan situs *web* selanjutnya akan di indeks oleh mesin pencari sehingga mempermudah pencarian informasi di internet.

Mendesain sebuah *crawler* yang baik saat ini menemui banyak tantangan [6]. Secara eksternal, *crawler* harus mengatasi besarnya situs *web* dan *link* jaringan. Secara internal, *crawler* harus mengatasi besarnya volume data. Sehubungan dengan terbatasnya sumber daya komputasi dan keterbatasan waktu, maka harus hati-hati memutuskan *url* apa yang harus di scan dan bagaimana urutannya. *crawler* tidak dapat mengunduh semua halaman *web*.

Kebanyakan *crawler* tidak dapat mengunjungi setiap halaman *web* yang mungkin, dengan dua alasan utama :

- *Client* mempunyai kapasitas penyimpanan yang terbatas, dan tidak dapat untuk mengindeks atau menganalisa semua halaman.
- Proses *crawl* memakan waktu, pada suatu waktu *crawler* mungkin perlu mulai mengunjungi halaman yang telah di scan sebelumnya, untuk menguji perubahan.

Berikut ini proses yang dilakukan *web crawler* pada saat bekerja :

- a. Mengunduh halaman *web*.
- b. Memparsing halaman yang diunduh dan mengambil semua *link*.
- c. Untuk setiap *link* yang diambil, ulangi proses.

2.8. *Text Mining*

Text mining memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapat dari dokumen dan tujuannya adalah mencari kata-kata yang dapat mewakili

isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen [7].

Text mining memproses informasi yang tidak terstruktur berupa data teks, kemudian mengambil informasi dari teks itu untuk dirubah ke dalam data numerik yang lebih berarti. Informasi yang ada dalam teks itu dapat digunakan untuk meramalkan pola dan kecenderungan yang dimiliki berdasarkan pada kata yang dikandungnya. Data numerik yang dihasilkan tersebut berasal dari kata-kata yang dikandung oleh suatu dokumen. Sehingga dengan menganalisa kata yang sudah diubah ke dalam data numerik ini maka pembelajaran pola statistik dapat dilakukan misalnya untuk dokumen ataupun mencari kemiripan antar dokumen [8].

2.9. Presisi dan Recall

Untuk menghitung tingkat performansi suatu sistem yang memiliki kemampuan untuk menghasilkan informasi-informasi tertentu dapat digunakan perhitungan presisi dan *recall*. Presisi adalah probabilitas informasi yang benar dari semua informasi yang dihasilkan sistem. Secara matematis, rumus untuk menghitung presisi dapat dilihat pada Persamaan (2.5).

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \quad (2.5)$$

Recall adalah probabilitas informasi benar yang dihasilkan dibandingkan jumlah informasi-informasi yang relevan. Secara matematis rumus untuk menghitung recall dapat dilihat pada Persamaan (2.6).

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \quad (2.6)$$

Sebagai contoh dokumen akan diklasifikasikan ke dalam tiga kategori yaitu dosen, jabatan dan tenaga kependidikan.

Dimana,

True positive = Dokumen hasil klasifikasi yang benar.

False positive = Dokumen yang seharusnya dikategorikan salah tetapi sistem mengkategorikan dokumen tersebut sesuai kategori yang diminta.

False negative = Dokumen yang seharusnya dikategorikan sebagai kategori dokumen yang diminta tetapi sistem mengkategorikan sebagai kategori lain.

True negative = Dokumen yang seharusnya tidak dikategorikan sebagai kategori yang diminta, dan sistem tidak mengkategorikannya sebagai kategori yang diminta.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis kebutuhan bisnis dari sistem yang akan dibangun dan perancangannya. Tahapan analisis ini membahas mengenai analisis kebutuhan yang menjadi dasar pada tahap perancangan yang akan dipaparkan pada subbab selanjutnya.

3.1. Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan perangkat lunak, aktor dan kasus penggunaan. Berikut penjabaran bagian-bagian tahap analisis.

3.1.1. Analisis Permasalahan

Dokumen merupakan salah satu produk yang dihasilkan dalam sebuah tatanan organisasi, mulai dari perencanaan, pelaksanaan, laporan dan evaluasi hampir seluruhnya menghasilkan dokumen. Selain itu rapat, presentasi, kebijakan, kajian maupun penelitian juga menghasilkan dokumen. Dengan banyaknya dokumen yang dihasilkan tersebut, maka manajemen dokumen menjadi peranan yang sangat penting.

Terkadang pegawai kesulitan untuk mendapatkan dokumen yang dibutuhkan. Hal itu disebabkan oleh beberapa alasan diantaranya kurangnya informasi ataupun perubahan peraturan yang mendadak. Untuk mengatasi masalah tersebut, maka akan digunakan metode klasifikasi untuk melakukan pengelompokan dokumen kepegawaian tersebut ke dalam sejumlah kategori. Dengan adanya sistem pengelompokan ini, pegawai dapat terbantu dalam pencarian dokumen yang lebih cepat dibandingkan pencarian konvensional pada lemari arsip dokumen.

Oleh karena itu dibutuhkan sebuah sistem yang membantu pengelompokan dokumen secara otomatis. Sehingga memudahkan pegawai dalam melakukan pencarian dokumen yang dibutuhkan.

3.1.2. Deskripsi Umum Sistem

Sistem yang akan dibuat yaitu berupa aplikasi *web* yang dapat melakukan pengelompokan dokumen secara otomatis ke dalam kategori yang telah ditentukan sebelumnya. Dokumen masukan dari aplikasi *web* berasal dari hasil upload dan *crawl* portal. Dokumen masukan berupa *pdf* akan tetapi tidak termasuk *pdf* hasil pemindaian. Dokumen yang masuk akan melalui tahap konversi ke teks, pra proses, pembobotan TF dan klasifikasi.

Diharapkan dengan adanya aplikasi *web* ini, pengguna dapat lebih mudah melakukan pencarian dokumen kepegawaian dengan adanya pengelompokan dokumen.

3.1.3. Spesifikasi Kebutuhan Perangkat Lunak

Sistem yang akan dibuat membutuhkan beberapa fungsi yang dapat membantu proses bisnis dalam sistem. Fungsi-fungsi tersebut antara lain:

1. Menampilkan hasil pengelompokan dokumen
Fungsi ini berguna untuk melihat hasil dari pengelompokan dokumen yang dimasukkan. Sistem akan menampilkan kategori dokumen beserta jumlah dokumen yang ada dalam kategori tersebut.
2. Menampilkan hasil pencarian
Fungsi ini berguna untuk melihat hasil pencarian dokumen setelah pengguna memasukkan kata kunci. Sistem menampilkan dokumen dan pengguna dapat langsung mengunduh dokumen yang tersedia.
3. Menampilkan dokumen terbaru
Fungsi ini berguna untuk menampilkan dokumen terbaru yang masuk ke *database*.

3.1.4. Aktor

Aktor mendefinisikan entitas-entitas yang terlibat dan berinteraksi langsung dengan sistem. Entitas ini bisa berupa manusia maupun sistem atau perangkat lunak yang lain. Aktor

yang terdapat pada sistem ini adalah pegawai ITS, admin utama dan admin tambahan. Berikut penjelasan mengenai aktor tersebut:

3.1.4.1. Pengguna

Pengguna adalah pegawai yang bekerja di lingkungan ITS. Pegawai ITS terdiri dari dua yaitu dosen dan tenaga kependidikan. Dalam sistem ini, pegawai ITS dapat melakukan pencarian dokumen, melihat maupun mengunduh dokumen yang ada.

3.1.4.2. Admin Utama

Admin utama adalah pegawai di bidang kepegawaian institut. Dalam sistem ini, admin utama dapat melakukan unggah dokumen, menambah admin dan mengganti *password*.

3.1.4.3. Admin Tambahan

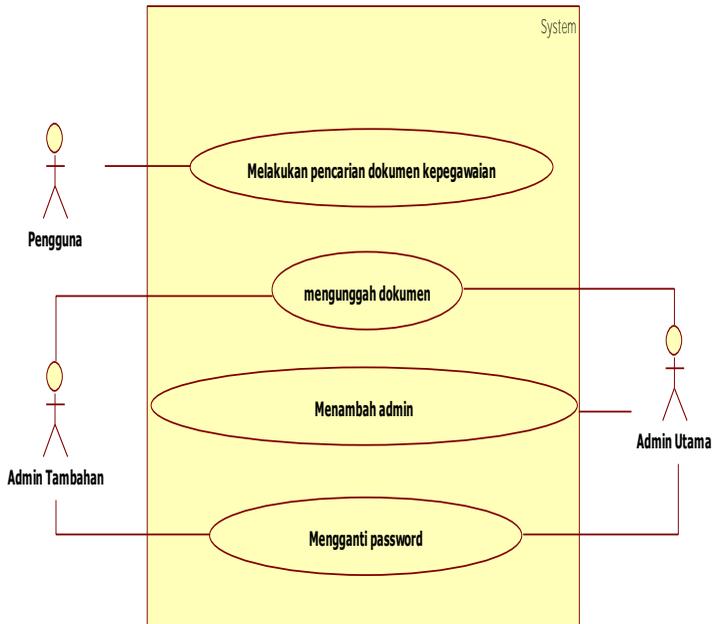
Admin tambahan adalah pegawai di bidang kepegawaian jurusan. Dalam sistem ini, admin tambahan dapat melakukan unggah dokumen dan mengganti *password*.

3.1.5. Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem dibuat kasus penggunaan sistem. Kasus-kasus penggunaan dalam sistem ini akan dijelaskan secara rinci pada subbab ini. Kasus penggunaan digambarkan dalam sebuah diagram kasus penggunaan yang dapat diketahui terdapat empat kasus penggunaan. Diagram kasus penggunaan dapat dilihat pada Gambar 3.1 dan berisi penjelasan dari setiap kasus penggunaan.

Tabel 3.1 Tabel Kode Kasus Penggunaan

Kode Kasus Penggunaan	Nama
UC-0001	Melakukan pencarian dokumen kepegawaian
UC-0002	Mengunggah dokumen kepegawaian
UC-0003	Menambah admin
UC-0004	Mengganti <i>password</i>



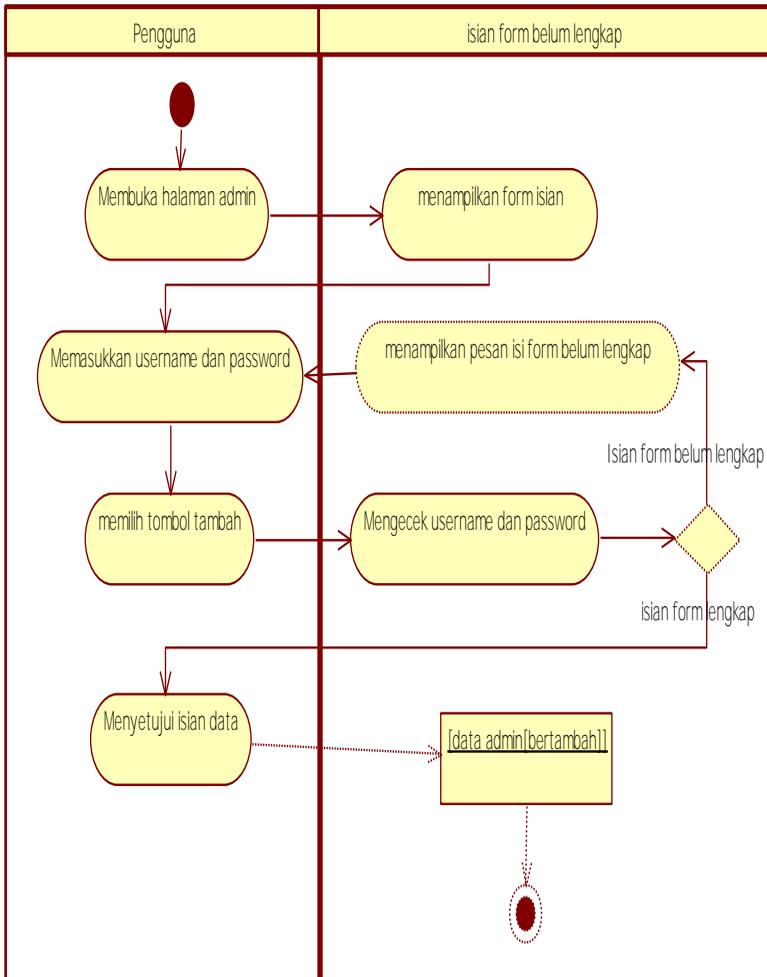
Gambar 3.1 Diagram Kasus Penggunaan

3.1.5.1. Melakukan Pencarian Dokumen Kepegawaian

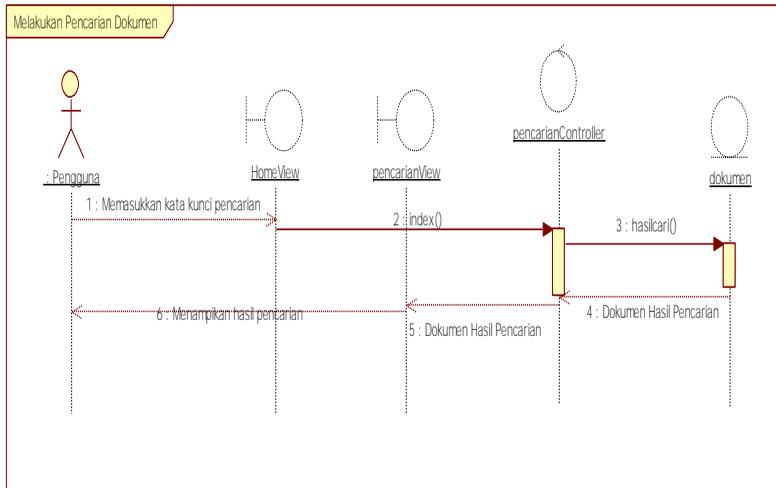
Pada kasus penggunaan ini, aktor yang berperan adalah pengguna. Aktor pengguna memiliki hak akses dalam melakukan pencarian dokumen kepegawaian yang nantinya sistem menerima *input* berupa kata kunci pencarian. Setelah itu, sistem akan mengolah kata kunci dan akan menampilkan daftar dokumen yang berhubungan dengan kata kunci yang dimasukkan. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.2. Diagram aktivitas dan diagram sekuens dari kasus penggunaan ini bisa dilihat pada Gambar 3.2 dan Gambar 3.3.

Tabel 3.2. Spesifikasi Kasus Penggunaan Melakukan Pencarian Dokumen Kepegawaian

Nama	Melakukan pencarian dokumen kepegawaian
Kode	UC-0001
Deskripsi	Menampilkan hasil pencarian dokumen kepegawaian sesuai dengan kata kunci yang dimasukkan.
Tipe	Fungsional
Pemicu	Pengguna menekan tombol perintah cari untuk memulai proses pencarian.
Aktor	Pegawai ITS
Kondisi Awal	Pengguna berada dalam halaman beranda
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna memasukkan kata kunci pencarian dokumen kepegawaian. A.1 . Sistem memberi pesan kata kunci pencarian tidak ditemukan 2. Pengguna menekan tombol cari untuk memulai pencarian. 3. Sistem mengolah kata kunci yang dimasukkan. 4. Sistem menampilkan pencarian dokumen sesuai kata kunci. 5. Pengguna melihat dokumen hasil pencarian
- Kejadian Alternatif	<p>A1. Sistem memberi pesan kata kunci pencarian tidak ditemukan</p> <p>A1.1 Menuju alur nomor 1</p>
Kondisi Akhir	Daftar dokumen kepegawaian kepegawaian sesuai kata kunci



Gambar 3.2 Diagram Aktivitas Melakukan Pencarian Dokumen Kepegawaian



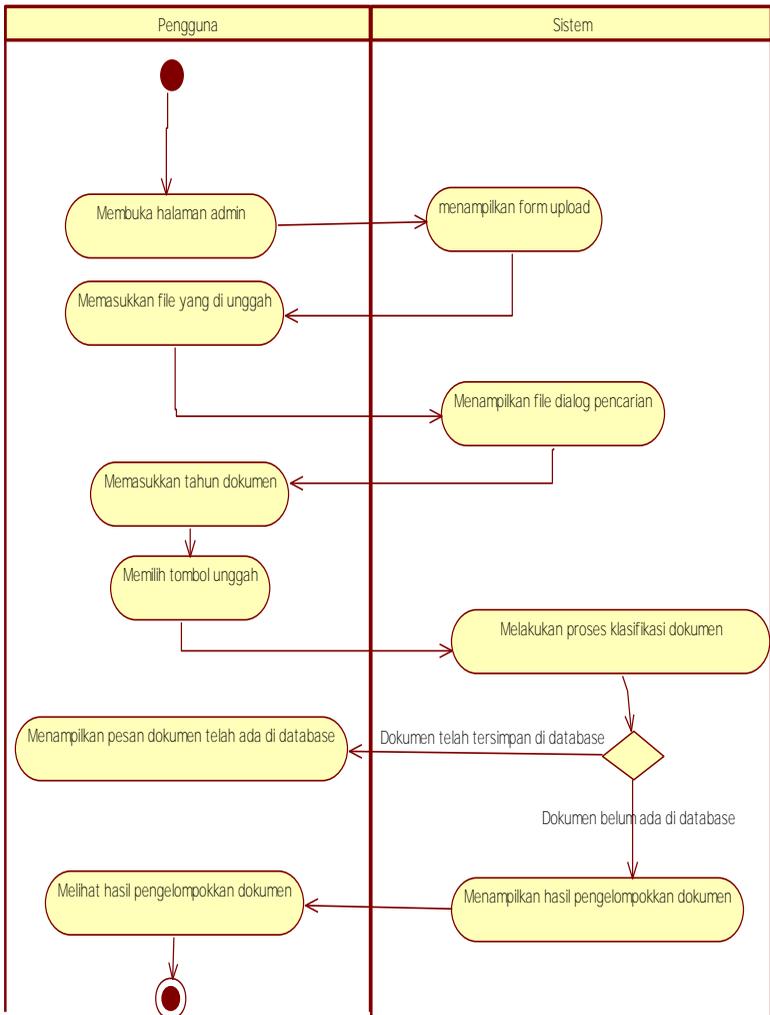
Gambar 3.3. Diagram Sekuens Melakukan Pencarian Dokumen Kepegawaian

3.1.5.2. Mengunggah Dokumen Kepegawaian

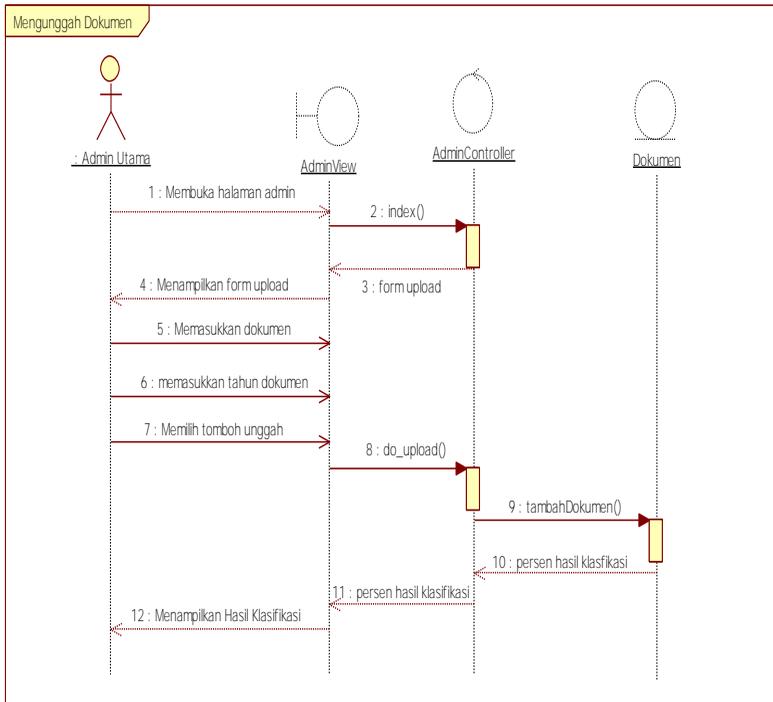
Pada kasus penggunaan ini, aktor yang berperan adalah admin utama dan admin tambahan. Aktor admin memiliki hak akses mengunggah dokumen kepegawaian. Sistem akan mengolah dokumen yang masuk apabila pengguna menekan tombol unggah dan system akan menampilkan hasil persentase pengelompokan ke dalam kategori yang telah ditentukan. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.3. Diagram aktivitas dan diagram sekuens dari kasus penggunaan ini bisa dilihat pada Gambar 3.4 dan Gambar 3.5.

Tabel 3.3 Spesifikasi Kasus Penggunaan Mengunggah Dokumen Kepegawaian

Nama	Mengunggah Dokumen Kepegawaian
Kode	UC-0002
Deskripsi	Admin dapat mengunggah Dokumen Kepegawaian
Tipe	Fungsional
Pemicu	Admin memilih menekan tombol unggah
Aktor	Admin Utama
Kondisi Awal	Admin berada dalam halaman admin
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Admin membuka halaman admin 2. Sistem menampilkan <i>form</i> upload 3. Admin memasukkan dokumen yang akan diunggah <ul style="list-style-type: none"> A1. Sistem menampilkan pesan dokumen telah ada di <i>database</i> 4. Sistem menampilkan file dialog pencarian 5. Admin memasukkan tahun dokumen yang diunggah 6. Admin memilih tombol unggah 7. Sistem melakukan proses klasifikasi 8. Sistem menampilkan hasil pengelompokan sesuai kategori 9. Admin melihat hasil pengelompokan
- Kejadian Alternatif	<p>A1. Sistem menampilkan pesan dokumen telah ada di <i>database</i></p> <p>A1.1 Menuju alur nomor 3</p>
Kondisi Akhir	Dokumen terunggah dan sistem menampilkan hasil persentase pengelompokan dokumen sesuai kategori yang telah ditentukan,
Kebutuhan Khusus	Tidak ada



Gambar 3.4. Diagram Aktivitas Mengunggah Dokumen Kepegawaian



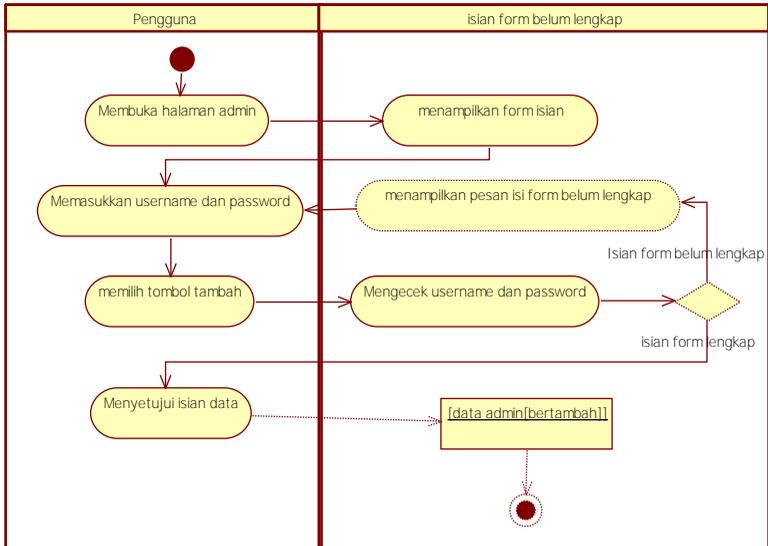
Gambar 3.5. Diagram Sekuens Mengunggah Dokumen Kepegawaian

3.1.5.3. Menambah Admin

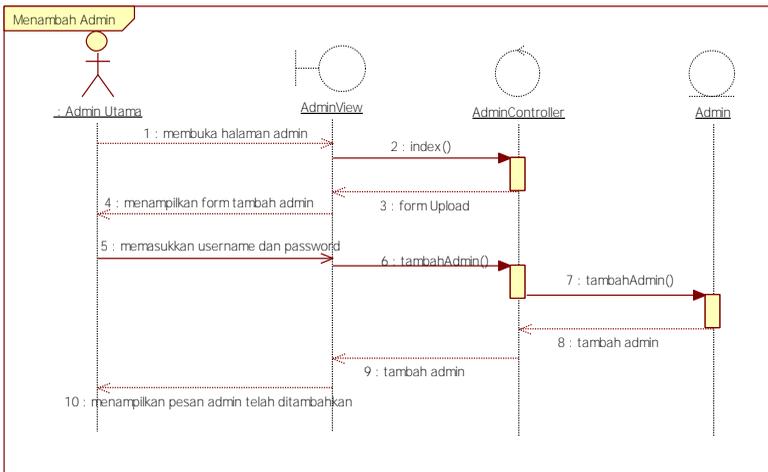
Pada kasus penggunaan ini, aktor yang berperan adalah admin utama. Aktor admin memiliki hak akses menambah admin baru. Sistem menampilkan *form* isian tambah admin baru. Admin memasukkan *username* dan *password* untuk admin baru. Sistem akan melakukan validitas *username* dan *password*. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.4. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 3.6 dan Gambar 3.7.

Tabel 3.4 Spesifikasi Kasus Penggunaan Menambah Admin

Nama	Menambah admin
Kode	UC-0003
Deskripsi	Admin utama dapat menambah admin baru yaitu pegawai bisang kepegawaian di tingkat jurusan
Tipe	Fungsional
Pemicu	Admin memilih menekan tombol tambah
Aktor	Admin Utama
Kondisi Awal	Admin berada dalam halaman admin
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Admin membuka halaman admin <ol style="list-style-type: none"> A.1 Sistem menampilkan pesan <i>form</i> isian belum lengkap 2. Sistem menampilkan form isian tambah admin 3. Admin memasukkan <i>username</i> dan <i>password</i> 4. Admin memilih tombol tambah 5. Sistem memverifikasi isian <i>form</i> tambah admin 6. Admin menyetujui isian data
- Kejadian Alternatif	<ol style="list-style-type: none"> A1. Sistem menampilkan pesan <i>form</i> isian belum lengkap <ol style="list-style-type: none"> A1.1 Menuju alur nomor 3
Kondisi Akhir	Data admin baru berhasil ditambahkan
Kebutuhan Khusus	Tidak ada



Gambar 3.6 Diagram Aktivitas Menambah Admin



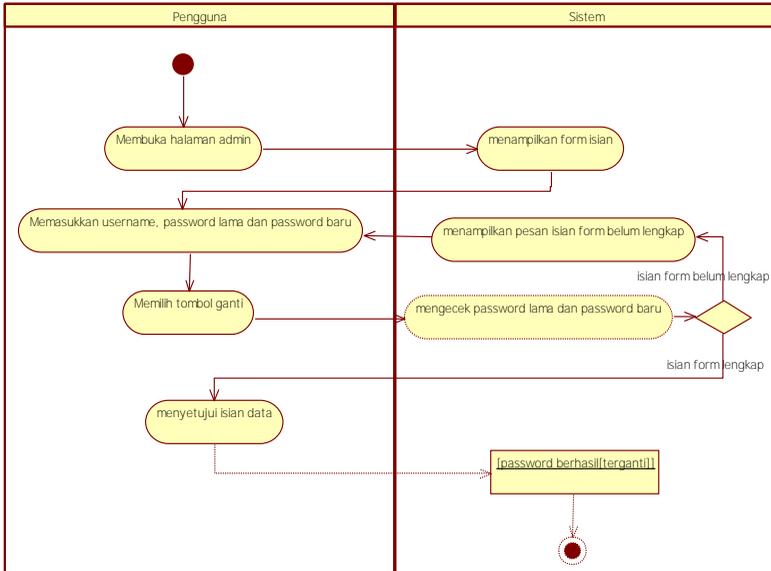
Gambar 3.7 Diagram Sekuens Menambah Admin

3.1.5.4. Mengganti *Password Admin*

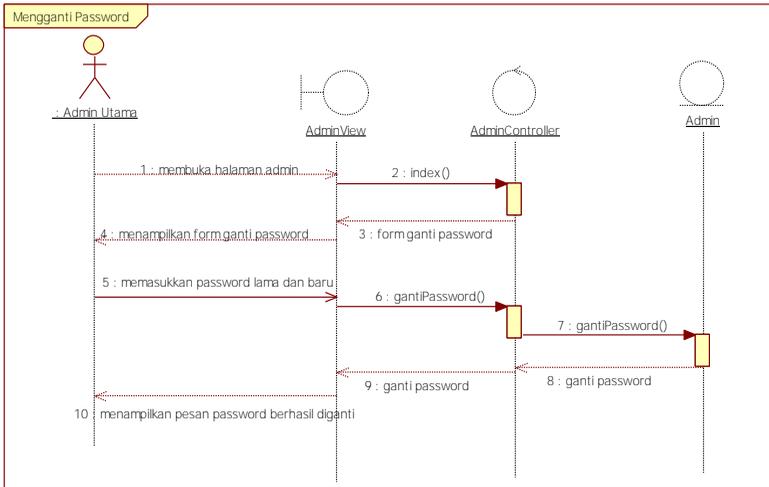
Pada kasus penggunaan ini, aktor yang berperan adalah admin utama dan admin tambahan. Aktor admin memiliki hak akses mengganti *password*. Sistem menampilkan *form* isian. Admin memasukkan *username*, *password* lama dan *password* baru. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.5. Diagram aktivitas dan diagram urutan dari kasus penggunaan ini bisa dilihat pada Gambar 3.8 dan Gambar 3.9.

Tabel 3.5 Spesifikasi Kasus Penggunaan Mengganti *Password*

Nama	Mengganti <i>password</i>
Kode	UC-0004
Deskripsi	Admin dapat mengganti <i>password login</i>
Tipe	Fungsional
Pemicu	Admin memilih menekan tombol ganti
Aktor	Admin utama dan admin tambahan
Kondisi Awal	Admin berada dalam halaman admin
Aliran:	
- Kejadian Normal	<ol style="list-style-type: none"> 1. Admin membuka halaman admin A.1 Sistem menampilkan pesan <i>form</i> isian belum lengkap 2. Sistem menampilkan form isian 3. Admin memasukkan <i>username</i>, <i>password</i> lama dan <i>password</i> baru 4. Sistem memverifikasi isian <i>form</i> 5. Admin menyetujui isian data
- Kejadian Alternatif	A1. Sistem menampilkan pesan <i>form</i> isian belum lengkap A1.1 Menuju alur nomor 3
Kondisi Akhir	<i>Password</i> lama berhasil diganti dengan <i>password</i> baru
Kebutuhan Khusus	Tidak ada



Gambar 3.8 Diagram Aktivitas Mengganti Password



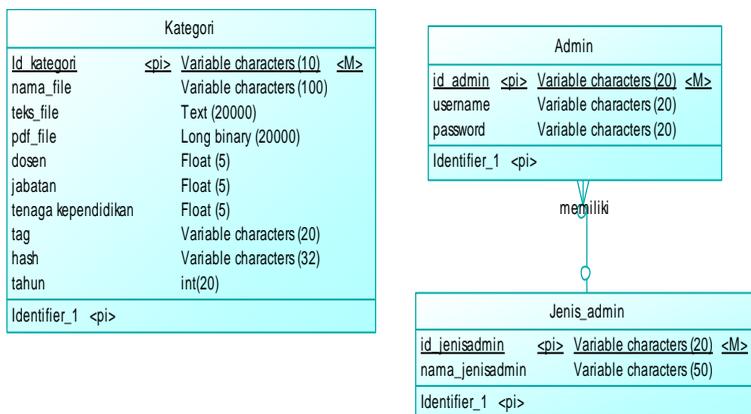
Gambar 3.9 Diagram Sekuens Mengganti Password

3.2. Perancangan Perangkat Lunak

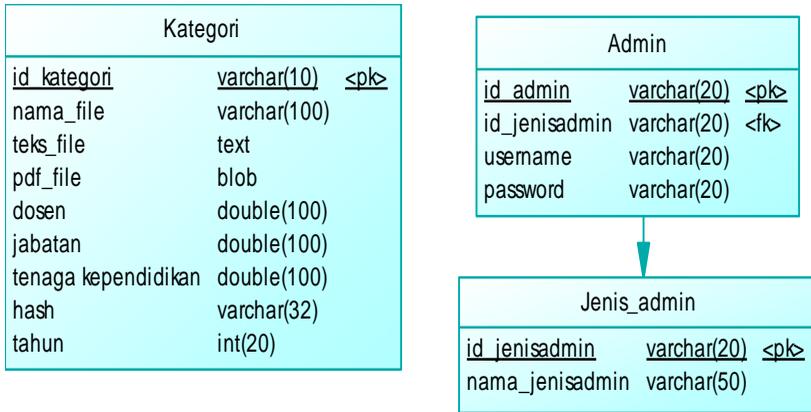
Penjelasan tahap perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu perancangan basis data, perancangan diagram kelas, perancangan arsitektur sistem, perancangan proses dan perancangan antarmuka. Berikut penjelasan masing-masing bagian perancangan.

3.2.1. Perancangan Basis Data

Pada subbab ini dijelaskan tentang bagaimana perancangan basis data yang akan digunakan pada sistem manajemen dokumen kepegawaian. Basis data pada sistem yang akan dibangun pada tugas akhir ini menggunakan sistem manajemen basis data relasional MySQL. *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM) dari desain basis data sistem ini dijelaskan pada Gambar 3.10 dan Gambar 3.11. Basis data aplikasi ini terdiri dari tiga tabel yaitu kategori, admin dan jenis admin. Tabel kategori mengakomodir atribut dari dokumen yang ada.



Gambar 3.10 Conceptual Data Model (CDM) Sistem Manajemen Dokumen Kepegawaian



Gambar 3.11 Physical Data Model (CDM) Sistem Manajemen Dokumen Kependidikan

3.2.1.1. Rancangan Tabel Kategori

Tabel ini digunakan untuk menyimpan data dokumen kepegawaian. Penjelasan tabel kategori dapat dilihat pada Tabel 3.6.

Tabel 3.6 Penjelasan Tabel Kategori

Nama Kolom	Keterangan
Id_kategori	Merupakan <i>primary key</i>
Nama_file	Nama file
Teks_file	<i>File teks</i>
Pdf_file	<i>File pdf</i>
Dosen	Persentase dokumen pada kategori dosen
Jabatan	Persentase dokumen pada kategori jabatan
Tenaga kependidikan	Persentase dokumen pada kategori tenaga kependidikan
hash	Menyimpan <i>file hash</i>
Tahun	Tahun dokumen

3.2.1.2. Rancangan Tabel Admin

Tabel ini digunakan untuk menyimpan data administrator yang diperlukan untuk keperluan *login*. Penjelasan tabel kategori dapat dilihat pada Tabel 3.7.

Tabel 3.7 Penjelasan Tabel Admin

Nama Kolom	Keterangan
Id_admin	Id dari administrator yang merupakan <i>primary key</i>
username	Nama administrator
Password	<i>Password</i> admin yang dipergunakan untuk <i>login</i> .

3.2.1.3. Rancangan Jenis Admin

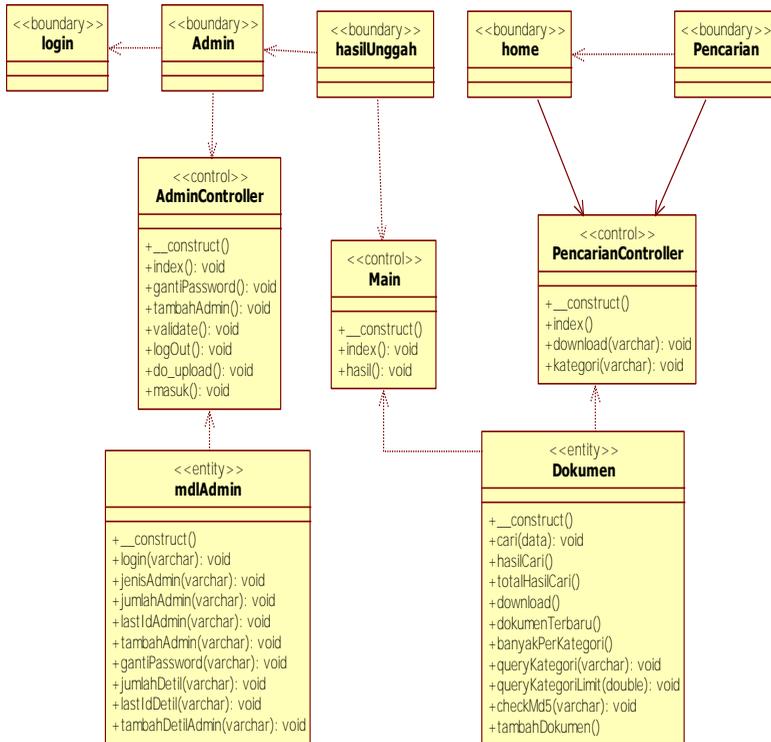
Tabel ini digunakan untuk menyimpan informasi jenis admin yang ada pada sistem. Penjelasan mengenai tabel jenis admin dapat dilihat pada Tabel 3.8.

Tabel 3.8 Penjelasan Tabel Jenis Admin

Nama Kolom	Keterangan
Id_jenisadmin	Merupakan <i>primary key</i>
nama_jenisadmin	Nama jenis admin

3.2.2. Perancangan Diagram Kelas

Perancangan diagram kelas berisi rancangan dari kelas-kelas yang digunakan untuk membangun sistem. Tiga lapisan pada arsitektur ini terdiri dari lapisan antarmuka, kontrol, dan data. Lapisan kontrol merupakan penghubung antara lapisan antarmuka dengan lapisan data. Lapisan antarmuka yang menampilkan tampilan pada *website*. Dan lapisan data yang menyimpan data-data yang diolah oleh sistem. Diagram kelas dari sistem ini bisa dilihat pada Gambar 3.12.



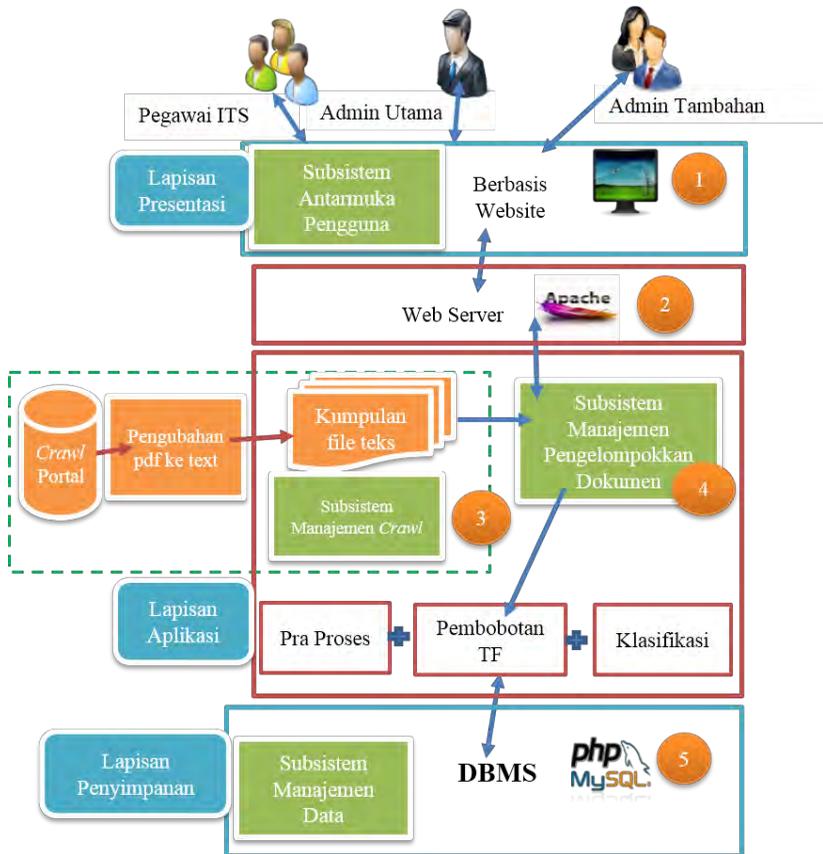
Gambar 3.12 Diagram Kelas Sistem Manajemen Dokumen Kepegawaian

3.2.3. Perancangan Arsitektur Sistem

Arsitektur dari perangkat lunak ini dapat dilihat pada Gambar 3.13 sistem akan memiliki tiga aktor yang memiliki peran masing-masing yaitu pengguna, admin utama dan admin tambahan. Administrator disini adalah pegawai di bidang kepegawaian. Admin utama yaitu pegawai bidang kepegawaian institut dan admin tambahan adalah pegawai bidang kepegawaian di jurusan. Kedua memiliki hak akses yang sama yaitu mengunggah dokumen dan mengganti *password*. Hanya saja yang dapat melakukan

tambah admin hanya admin utama saja. Pengguna disini adalah pegawai ITS yang terdiri dari dosen dan tenaga kependidikan. Masukan dari sistem adalah dokumen kepegawaian yang didapatkan dari hasil unggah administrator dan *crawl* portal yang memuat dokumen kepegawaian. Pengguna dapat melakukan pencarian dokumen pada sistem. Keluaran yang dihasilkan adalah daftar dokumen kepegawaian yang sudah terkelompokan menjadi beberapa kategori. Kategori tersebut adalah dosen, jabatan dan tenaga kependidikan. Sistem ini berjalan dengan urutan seperti bawah ini:

1. Pada lapisan presentasi terdapat subsistem manajemen antarmuka pengguna yang berbasis *website*.
2. Pada lapisan aplikasi menggunakan *webservice* Apache karena website dibuat dengan bahasa pemrograman php dan framework Codeigniter.
3. Di luar sistem terdapat aplikasi yang melakukan pengambilan dokumen dari portal yang terdapat pada subsistem manajemen *crawl*. Dokumen yang diambil berekstensi *pdf* yang nantinya akan diubah terlebih dahulu menjadi teks. Kemudian *file* teks tersebut akan melalui proses pengelompokan dokumen pada subsistem manajemen pengelompokan dokumen.
4. Pada subsistem manajemen pengelompokan dokumen terdapat proses pengelompokan dokumen ke dalam beberapa kategori yang akan melewati beberapa tahap yaitu pra proses, pembobotan TF dan klasifikasi dengan metode naïve bayes.
5. Pada lapisan penyimpanan yang merupakan subsistem manajemen data menggunakan DBMS MySQL. Pada basis data ini menyimpan data dokumen kepegawaian dan admin.



Gambar 3.13 Arsitektur Sistem Manajemen Dokumen Kepegawaian

3.2.4. Perancangan Proses

Pada subbab ini akan dijelaskan mengenai rancangan proses aplikasi algoritma yang digunakan untuk pencapaian suatu fungsi pada program. Sehingga antar fungsi nantinya bisa dijalankan dengan baik.

3.2.4.1. Proses Pengelompokan Dokumen pada Data Pelatihan

Proses pengelompokan dokumen pada data pelatihan dilakukan dengan langkah manual yaitu memasukkan dokumen yang relevan ke dalam kategori dosen, jabatan dan tenaga kependidikan. Satu dokumen dapat memiliki lebih dari satu kategori. Adapun parameter dokumen dapat dimasukkan ke dalam kategori dosen, jabatan dan tenaga kependidikan adalah sebagai berikut:

a. Kategori Dosen

Kategori dosen merepresentasikan dokumen yang berkaitan dengan kegiatan yang dilakukan oleh dosen meliputi peraturan yang membahas tentang dosen, pension, tunjangan, angka kredit, kenaikan pangkat, beban kerja dosen, pengangkatan dan pemberhentian dosen, pedoman nidn dosen, beasiswa, tugas belajar, panduan pelaksanaan tridharma perguruan tinggi dan dokumen lainnya yang berhubungan dengan dosen.

b. Kategori Jabatan

Kategori jabatan merepresentasikan dokumen yang secara umum berkaitan dengan seluruh jabatan yang ada di perguruan tinggi yaitu jabatan fungsional maupun jabatan struktural yang telah dijelaskan pada 2.3. Dokumen tersebut antara lain pemberhentian PNS, tunjangan pegawai kemdikbud, prestasi kerja PNS, PNS jabatan rangkap, batas usia pension PNS, Usia pension jabatan struktural dan dokumen lainnya yang berhubungan dengan seluruh jabatan pegawai perguruan tinggi.

c. Kategori Tenaga Kependidikan

Kategori tenaga kependidikan merepresentasikan dokumen yang mewakili segala kegiatan yang berhubungan dengan tenaga kependidikan. Dokumen tersebut antara lain Kenaikan pangkat, tunjangan, kode etik, tugas belajar, beasiswa dan dokumen lainnya yang berhubungan dengan tenaga kependidikan.

Setelah dokumen yang relevan terkumpul sesuai kategorinya, maka kumpulan dokumen selanjutnya akan dijadikan data pelatihan untuk mendapatkan model pada sub proses pengelompokan dokumen.

3.2.4.2. Proses Pengelompokan Dokumen Kepegawaian Menggunakan Klasifikasi *Naïve Bayes*

Proses pengelompokan dokumen kepegawaian menggunakan klasifikasi *Naïve Bayes* terdapat dua proses yaitu proses pelatihan dan testing. Proses pelatihan pengelompokan dokumen dapat dilihat pada diagram alir proses pengambilan data pada lampiran A Diagram Alir 7.2.

Proses pelatihan pengelompokan dokumen dengan klasifikasi *Naïve Bayes* dimulai dengan menginisialisasi variabel klasifikasi kemudian memasukkan dokumen pelatihan yang akan dikelompokkan. Setelah itu dilakukan tokenisasi dokumen dan pembobotan TF untuk menghitung (*term*) jumlah kata dalam dokumen. Melakukan pelatihan dokumen dan menyimpan model hasil *pelatihan* yang nantinya akan digunakan pada proses testing.

Setelah model klasifikasi didapatkan dari proses pelatihan pengelompokan dokumen. Proses testing pengelompokan dokumen sudah dapat dilakukan. Dalam sistem ini digunakan pada saat proses mengunggah dokumen dan proses *crawl* dokumen. Proses testing pengelompokan dokumen kepegawaian menggunakan klasifikasi *Naïve Bayes* dapat dilihat pada diagram alir testing proses pengambilan data pada lampiran A Diagram Alir 7.3.

Proses testing pengelompokan dokumen diawali dengan membuka model klasifikasi yang telah didapatkan dari hasil pelatihan dokumen. Setelah itu dokumen kepegawaian dimasukkan untuk di proses tokenisasi, pembobotan TF dan dilakukan klasifikasi. Sehingga dihasilkan persentase per kategori dimana dokumen akan masuk ke beberapa kategori ataupun satu kategori. Persentase kategori akan disimpan ke dalam *database*.

3.2.4.3. Proses Pengambilan Dokumen dari Portal yang Memuat Dokumen Kepegawaian

Proses pengambilan dokumen dari portal yang memuat dokumen kepegawaian dapat dilihat pada diagram alir proses pengambilan data pada lampiran A Diagram Alir 7.1. Proses pengambilan data menggunakan *crawl*.

Tahapan dari proses *crawl* itu sendiri yaitu pertama harus diketahui dulu *link* dari *website* yang memuat dokumen kepegawaian, setelah itu membuka teks filter yang dibuat tersendiri yang digunakan untuk memfilter dokumen yang tidak diinginkan. Kemudian mengecek file yang akan diunduh tersebut *file pdf*. Mengecek hash file yang digunakan untuk melihat dokumen tersebut sudah ada di database atau belum. Apabila dokumen tersebut sudah tersedia maka dokumen tersebut akan diabaikan.

Ketika dokumen berekstensi pdf sudah didapatkan maka dokumen tersebut akan di konversi ke teks. Setelah itu akan dilakukan pengecekan kata-kata di dalam teks apakah memuat kata-kata yang ada di dalam teks filter. Dokumen yang lolos filter nantinya akan diklasifikasi dan menghasilkan dokumen dengan kategori tertentu. Proses akan menghasilkan persentase hasil klasifikasi yang akan disimpan di dalam *database*.

3.2.5. Perancangan Antarmuka

Pada subbab ini akan dijelaskan mengenai rancangan antarmuka grafis yang akan memberi fasilitas pengguna untuk berinteraksi mudah dengan sistem. Rancangan yang dibahas meliputi kontrol pada antarmuka dan ketentuan input. Rancangan antarmuka tersebut antara lain : halaman utama, halaman pencarian dokumen, halaman beranda administrator utama dan halaman administrator tambahan.

3.2.5.1. Antarmuka Halaman Utama

Halaman ini merupakan tampilan utama yang muncul ketika sistem pertama kali dijalankan. Terdapat informasi tentang aplikasi dan petunjuk penggunaan aplikasi. Pada halaman utama

sistem manajemen dokumen kepegawaian terdapat kolom pencarian untuk memudahkan pegawai dalam mencari dokumen. Pada halaman utama juga terdapat menu kategori dokumen, dokumen terbaru, *link website* terkait dan kontak person. Rancangan tampilan dapat dilihat pada Gambar 3.14.



Gambar 3.14 Rancangan Antarmuka Halaman Utama

3.2.5.2. Antarmuka Halaman Pencarian Dokumen

Aplikasi dapat menampilkan dokumen kepegawaian sesuai kata kunci yang dimasukkan serta mengunduh langsung dokumen tersebut. Dokumen tersebut didapat dari hasil unggahan administrator dan diambil langsung dari *website* kopertis 12 dan kopertis 3. Rancangan antarmuka pencarian dokumen dapat dilihat pada Gambar 3.15.



Gambar 3.15 Rancangan Antarmuka Halaman Pencarian Dokumen

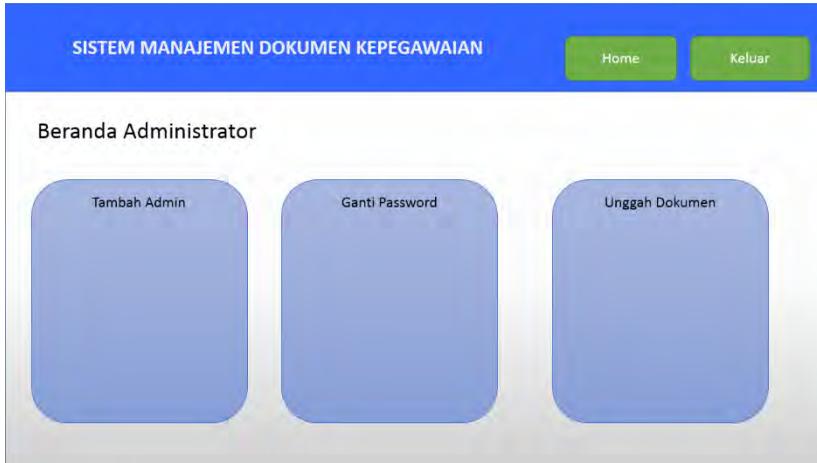
3.2.5.3. Antarmuka Halaman Beranda Administrator Utama

Administrator utama memiliki hak akses fitur aplikasi untuk menambah admin, mengganti *password* dan mengunggah dokumen.

Data masukan untuk tambah admin adalah *username* dan *password*. Data masukan untuk ganti password adalah password lama dan password baru. Sedangkan untuk unggah dokumen, data masukan berupa dokumen kepegawaian berekstensi *pdf*. Rancangan antarmuka halaman beranda administrator utama diperlihatkan pada Gambar 3.16.

3.2.5.4. Antarmuka Halaman Beranda Administrator Tambahan

Administrator tambahan memiliki hak akses fitur aplikasi untuk mengganti *password* dan mengunggah dokumen. Data masukan untuk ganti password adalah password lama dan password baru. Sedangkan untuk unggah dokumen, data masukan berupa dokumen kepegawaian berekstensi *pdf*. Rancangan antarmuka halaman beranda administrator tambahan diperlihatkan pada Gambar 3.17.



Gambar 3.16 Rancangan Antarmuka Halaman Beranda Administrator Utama



Gambar 3.17 Rancangan Antarmuka Halaman Beranda Administrator Tambahan

BAB IV IMPLEMENTASI

Bab ini membahas tentang implementasi dari perancangan sistem. Bab ini berisi proses implementasi dari setiap kelas pada semua modul. Bahasa pemrograman yang digunakan adalah bahasa pemrograman Java dan PHP dengan *framework* CodeIgniter.

4.1. Implementasi Lapisan Antarmuka

Lapisan antarmuka merupakan lapisan yang bertugas mengatur tampilan sistem. Untuk menyusun tampilan dari aplikasi ini digunakan bahasa markah HTML (*HyperText Markup Language*) yang dikombinasikan dengan CSS (*Cascading Style Sheet*) dan *JavaScript* untuk meningkatkan kesan interaktif dari aplikasi.

4.2. Implementasi Lapisan Kontrol

Lapisan kontrol merupakan lapisan yang bertanggung jawab dengan tingkah laku sistem. Lapisan ini bertugas menghubungkan lapisan data dengan lapisan antarmuka.

4.2.1. Kelas Pencarian

Kelas Pencarian merupakan kelas *controller* yang menangani proses pencarian dokumen berdasarkan kata kunci dari pengguna yang merupakan salah satu fitur dari aplikasi. Pada kelas ini terdapat fungsi *index* yang melakukan *load* model dan melakukan *query* data untuk mencari dokumen berdasarkan parameter berupa kata kunci dari pengguna dan akan mengembalikan data dokumen yang bersesuaian untuk dikirim ke *view* untuk ditampilkan. Kode sumber untuk fungsi ini dapat dilihat pada Kode Sumber 8.1.

Selain fungsi *index* untuk pencarian dokumen, pada kelas ini juga terdapat fungsi *download* yang memudahkan pengguna untuk mengunduh dokumen setelah melihat hasil pencarian dokumen yang dibutuhkan. Kode sumber fungsi ini dapat dilihat

pada Kode Sumber 8.2. Untuk memudahkan pengguna dalam hal pencarian, pada aplikasi ini terdapat fitur *tag* di fungsi kategori. Setiap dokumen yang ada akan mempunyai *tag* yang mewakili kategori dari dokumen tersebut. Satu dokumen dapat memiliki satu *tag* atau lebih. Kode sumber fungsi ini dapat dilihat pada Kode Sumber 8.3.

4.2.2. Kelas Main

Kelas Main merupakan kelas *controller* yang menangani fitur di halaman beranda yaitu menampilkan jumlah dokumen dan dokumen terbaru yang ada di *database*. Fitur tersebut terdapat pada fungsi *index* yang melakukan *load* model dan melakukan *query* data untuk menghitung jumlah dokumen yang ada di *database* dan dokumen terbaru di *database* dan akan mengembalikan data jumlah dokumen dan dokumen terbaru dikirim ke *view* untuk ditampilkan. Kode sumber untuk fungsi ini dapat dilihat pada Kode Sumber 8.4. Selain itu juga terdapat fungsi hasil untuk melihat hasil unggah dokumen yang dilakukan oleh administrator. Kode sumber untuk fungsi ini dapat dilihat pada Kode Sumber 8.5.

4.2.3. Kelas Admin

Kelas ini menangani proses yang hanya bisa dilakukan oleh pengguna yang memiliki hak akses sebagai *admin*. *Admin* dapat mengunggah dokumen. *File* dokumen yang dapat diunggah hanya dokumen berekstensi *pdf*. Setelah memasukkan dokumen yang akan diunggah, *admin* harus memasukkan tahun dokumen. Fungsi ini akan dipanggil apabila *admin* menekan tombol ‘unggah’. Rincian kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 8.6.

Selain itu *admin* dapat melakukan ganti password. *Admin* harus memasukkan *password* lama dan *password* baru. Fungsi ini akan dipanggil apabila *admin* menekan tombol ‘ganti’. Rincian kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 8.7. *Admin* juga dapat menambahkan *admin*. Fungsi ini hanya dapat

dilakukan oleh admin utama saja. *Admin* utama memasukkan *username* dan *password* untuk *admin* tambahan. Fungsi ini akan dipanggil apabila *admin* menekan tombol ‘tambah’. *Admin* yang ditambahkan oleh admin utama hanya dapat melakukan fungsi unggah dan ganti *password*. Rincian kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 8.8.

4.2.4. Kelas TextClassifier

Kelas ini menangani proses klasifikasi dokumen yang terdiri dari proses pelatihan dan testing dokumen. Proses pelatihan data dilakukan pada setiap kategori secara bergantian. Kemudian dilakukan *tokenisasi* dan pembobotan TF seperti yang dijelaskan pada subbab 2.5. Dokumen di pelatihan dan akan menghasilkan model yang digunakan untuk pengujian nantinya. Rincian kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 8.9.

Proses testing dokumen dimulai dengan membuka model yang telah dihasilkan pada proses pelatihan dokumen. Kemudian dokumen akan di-*tokenisasi* dan dilakukan pembobotan TF. Setelah itu dokumen diklasifikasi sesuai kategori yang sudah disiapkan. Hasil dari klasifikasi adalah persentase dari setiap kategori yang akan disimpan ke *database*. Rincian kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 8.10. Kelas *TextClassifier* nantinya akan dijadikan dalam bentuk *jar* untuk diakses menjadi *library* di *website*. *Jar* ini yang akan mengolah pengelompokan dokumen baik dari hasil unggah dokumen maupun *crawl* portal.

4.3. Implementasi Lapisan Data

Lapisan ini hanya berhubungan dengan kelas-kelas pada lapisan kontrol. Lapisan data menyimpan atribut yang diperlukan oleh kelas-kelas pada lapisan *control* dan berisi atribut yang bersifat *private*.

4.3.1. Kelas Dokumen

Kelas ini berisi atribut-atribut yang diperlukan untuk merepresentasikan kumpulan dokumen. Atribut tersebut antara lain

merepresentasikan nama dokumen, teks dari dokumen, *file pdf* dari dokumen, persentase tiap kategori dari dokumen, *file hash* dari dokumen, *tag* dari dokumen dan tahun dokumen. Kelas ini juga dilengkapi dengan fungsi-fungsi untuk mengakses atribut-atribut tersebut.

4.3.2. Kelas mdlAdmin

Kelas ini merepresentasikan atribut –atribut yang diperlukan untuk merepresentasikan kerja dari administrator yaitu mengunggah dokumen, mengganti *password* dan menambah admin baru. Kelas ini juga dilengkapi dengan fungsi-fungsi untuk mengakses atribut-atribut tersebut.

4.4. Implementasi Crawl

Kelas ini merupakan kelas yang bertugas menangani proses pengambilan dokumen dari portal. Proses *crawl* dimulai dengan memasukkan *link* portal yang akan diambil dokumennya. Hanya file pdf saja yang akan diambil. Proses *crawl* juga terhubung dengan proses konversi *pdf* ke teks serta klasifikasi dokumen. Proses *crawl* dijalankan pada aplikasi bawaan *windows* dengan nama *task scheduler*. Proses *crawl* dapat berjalan dengan memasukkan *script* program ke dalam *task scheduler* serta diatur tiap berapa waktu proses *crawl* akan dijalankan. Rincian kode sumber untuk kelas ini dapat dilihat pada Kode Sumber 8.11.

4.5. Implementasi Antarmuka Pengguna

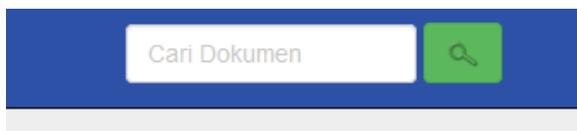
Implementasi tampilan antarmuka pengguna dibuat dengan *HyperText Markup Language* (HTML) yang disertai dengan *Cascading Style Sheet* (CSS) dan *JavaScript* untuk menambah kesan interaktif pada website. Berikut ini akan dijelaskan mengenai implementasi tampilan antarmuka pengguna aplikasi.

4.5.1. Tampilan Halaman Beranda

Halaman Beranda merupakan halaman pertama yang dilihat oleh pengguna ketika membuka aplikasi. Pada halaman ini

terdapat penjelasan tentang aplikasi, petunjuk pengguna, kategori dokumen dengan menampilkan jumlah dokumen yang ada, dokumen terbaru yang ada di *database*, *link website* terkait dan kontak person. Tampilan halaman beranda dapat dilihat pada Gambar 4.1. Halaman Beranda juga menyediakan tombol *direct* pencarian. Tampilan halaman beranda dapat dilihat pada Gambar 4.2.

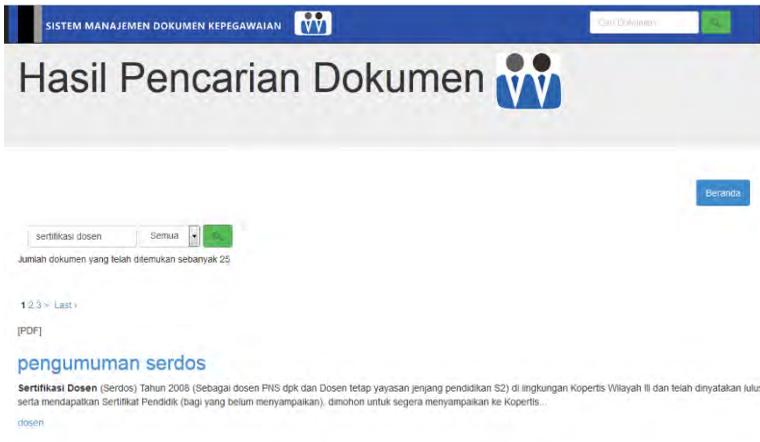
Gambar 4.1 Tampilan Halaman Beranda



Gambar 4.2 Tampilan Tombol Pencarian

4.5.2. Tampilan Halaman Pencarian Dokumen

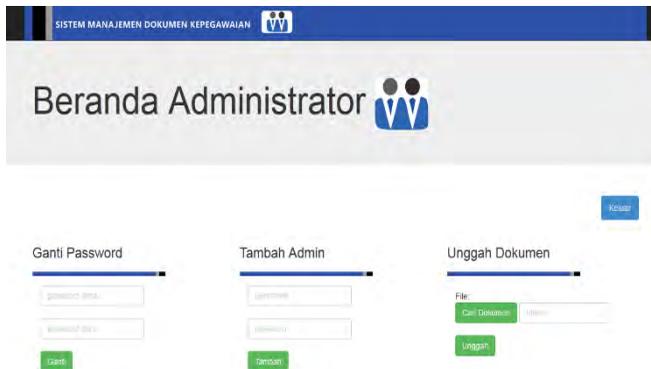
Halaman Pencarian Dokumen digunakan untuk melakukan pencarian terhadap dokumen tertentu berdasarkan kata kunci dari pengguna. Pada halaman ini terdapat form pencarian yang terdiri atas sebuah textbox untuk memasukkan kata kunci dan sebuah tombol untuk mengirim perintah pencarian ke controller. Hasil pencarian akan ditampilkan pada panel di bawah form pencarian. Tampilan halaman pencarian barang dapat dilihat pada Gambar 4.3.



Gambar 4.3 Tampilan Halaman Pencarian Dokumen

4.5.3. Tampilan Halaman Beranda Administrator Utama

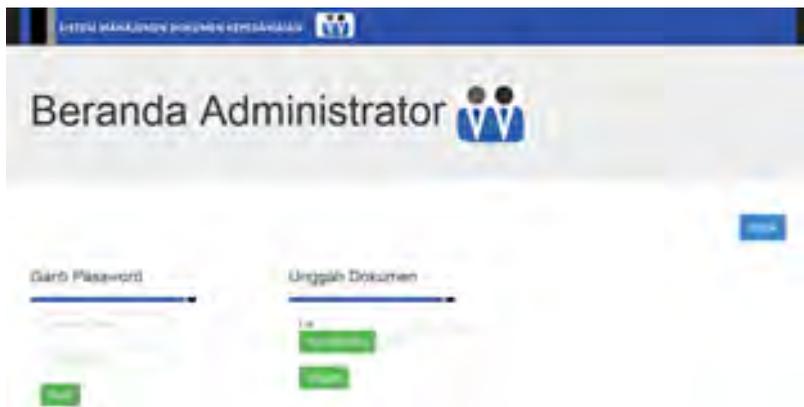
Halaman Beranda Administrator utama menyediakan 3 hak akses yang hanya bisa dilakukan oleh *admin* aplikasi. Tampilan halaman beranda administrator utama ditunjukkan pada Gambar 4.4. Pada halaman tersebut *admin* dapat melakukan 3 fitur yaitu mengunggah dokumen, mengganti password dan menambah admin baru.



Gambar 4.4 Tampilan Halaman Beranda Administrator Utama

4.5.4. Tampilan Halaman Beranda Administrator Tambahan

Halaman Beranda Administrator utama menyediakan 3 hak akses yang hanya bisa dilakukan oleh *admin* aplikasi.. Pada halaman tersebut *admin* dapat melakukan 2 fitur yaitu mengunggah dokumen dan mengganti password. Tampilan halaman beranda administrator utama ditunjukkan pada Gambar 4.5.



Gambar 4.5 Tampilan Halaman Beranda Administrator Tambahan

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada sistem yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem dan kegunaan sistem. Pengujian fungsionalitas mengacu pada kasus penggunaan pada bab tiga. Pengujian kegunaan program dilakukan dengan mengetahui tanggapan dari pengguna terhadap sistem. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan tugas akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Prosesor	: Intel Core i3-3240 CPU @ 3.40GHz
Memori	: 4.00 GB
Jenis Device	: Personal Computer
Sistem Operasi	: Microsoft Windows 7 Ultimate 64 bit
Browser	: Mozilla Firefox 30.0

5.2. Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian yang dilakukan adalah pengujian kebutuhan fungsionalitas, pengujian kegunaan sistem dan pengujian hasil pengelompokan . Pengujian fungsionalitas menggunakan metode kotak hitam (*black box*). Metode ini menekankan pada kesesuaian hasil keluaran sistem.

Pengujian terhadap hasil pengelompokan dilakukan dengan membandingkan hasil pelabelan kelas oleh sistem dengan hasil analisis manual yang dilakukan oleh pengembang aplikasi. Hasil perbandingan tersebut digunakan untuk mendapatkan nilai presisi dan *recall*. Pengujian kegunaan

sistem dilakukan dengan melakukan survei pada pengguna aplikasi.

5.2.1. Pengujian Fungsionalitas

Pengujian fungsionalitas sistem dilakukan dengan menyiapkan sejumlah skenario sebagai tolok ukur keberhasilan pengujian. Pengujian fungsionalitas dilakukan dengan mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 3.1.5. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

5.2.1.1. Pengujian Fitur Melakukan Pencarian Dokumen

Pengujian fitur melakukan pencarian dokumen tertentu merupakan pengujian terhadap kemampuan aplikasi dalam mencari dokumen yang bersesuaian dengan kata kunci yang diberikan oleh pengguna. Rincian skenario pengujian kasus penggunaan ini dapat dilihat pada Tabel 5.1.

Tabel 5.1 Pengujian Fitur Melakukan Pencarian Dokumen Skenario 1

ID	UJ.UC-0001
Referensi Kasus Penggunaan	UC-0001
Nama	Pengujian fitur melakukan pencarian dokumen tertentu pada aplikasi
Tujuan Pengujian	Menguji fitur untuk mencari dokumen berdasarkan kata kunci pengguna
Skenario 1	Dokumen yang bersesuaian dengan kata kunci dari pengguna akan ditampilkan pada panel hasil pencarian.
Kondisi Awal	Halaman beranda dari aplikasi telah dibuka pada <i>browser</i> .
Data Uji	Data uji berupa kata kunci pencarian yaitu ‘sertifikasi dosen’
Langkah Pengujian	Pengguna memasukkan kata kunci pencarian dan memilih tombol ‘cari’

Hasil Yang Diharapkan	Aplikasi menampilkan dokumen yang bersesuaian dengan kata kunci pengguna di panel hasil pencarian.
Hasil Yang Didapat	Aplikasi menampilkan dokumen yang bersesuaian dengan kata kunci pengguna di panel hasil pencarian.
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan hasil deteksi dapat dilihat pada Gambar 5.1

The screenshot shows the interface of the 'SISTEM MANAJEMEN DOKUMEN KEPEGAWAIAN' (Employee Document Management System). At the top, there is a search bar with the text 'Cari Dokumen' and a 'Go' button. Below the search bar, there is a 'Beranda' (Home) button. The search results section shows a search for 'sertifikasi dosen' in the year 2013, with 3 documents found. The first result is a PDF document titled 'Edaran 3693 Info Awal Perubahan Serdos 2013' from the Indonesian Ministry of Education and Culture (KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN). The document is dated December 10, 2012, and is related to 'Sertifikasi Dosen' (Teacher Certification).

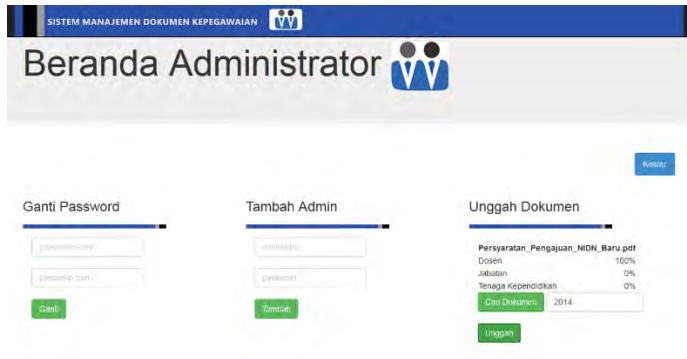
Gambar 5.1. Hasil Uji Melakukan Pencarian Dokumen Skenario 1

5.2.1.2. Pengujian Fitur Mengunggah Dokumen

Pengujian fitur mengunggah dokumen merupakan pengujian yang dilakukan untuk mengetahui kemampuan mengunggah dokumen melalui aplikasi. *File* yang akan diunggah dalam bentuk *pdf*, dimana dokumen tersebut bukan hasil pemindaian. Rincian skenario pengujian pada kasus penggunaan mengunggah dokumen dapat dilihat pada Tabel 5.2.

Tabel 5.2 Pengujian Fitur Mengunggah Dokumen Skenario 1

ID	UJ.UC-0002
Referensi Kasus Penggunaan	UC-0002
Nama	Pengujian fitur mengunggah dokumen
Tujuan Pengujian	Menguji fitur untuk mengunggah dokumen melalui aplikasi
Skenario	Mengunggah dokumen kepegawaian dengan ekstensi file <i>pdf</i>
Kondisi Awal	Pengguna merupakan <i>admin</i> dan telah masuk dalam halaman administrator. Halaman administrator telah dibuka pada <i>browser</i> .
Data Uji	Data uji berupa dokumen kepegawaian ‘persyaratan nidn baru’
Langkah Pengujian	Pengguna memasukkan dokumen ‘persyaratan nidn baru’ lalu memilih tombol ‘unggah’
Hasil Yang Diharapkan	Unggahan dokumen berhasil dan sistem menampilkan hasil persentase pengelompokan dokumen. Dokumen yang diunggah bisa ditemukan kembali di halaman pencarian.
Hasil Yang Didapat	Unggahan dokumen berhasil dan sistem menampilkan hasil persentase pengelompokan dokumen. Dokumen yang diunggah bisa ditemukan kembali di halaman pencarian.
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan unggahan berhasil serta hasil persentase pengelompokan ditunjukkan pada Gambar 5.2



Gambar 5.2 Hasil Uji Fitur Mengunggah Dokumen

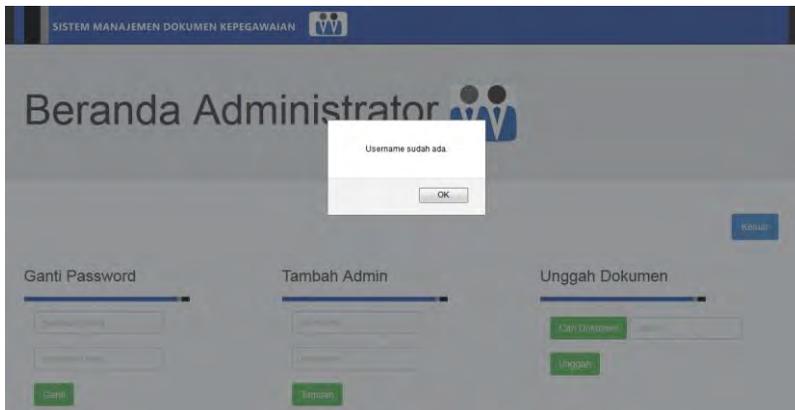
5.2.1.3. Pengujian Fitur Menambah Administrator

Pengujian fitur menambah administrator pada aplikasi adalah pengujian terhadap kemampuan aplikasi dalam menambah administrator baru melalui aplikasi. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.3.

Tabel 5.3 Pengujian Fitur Menambah Administrator

ID	UJ.UC-0003
Referensi Kasus Penggunaan	UC-0003
Nama	Pengujian fitur menambah administrator
Tujuan Pengujian	Menguji fitur untuk menambah administrator baru melalui aplikasi
Skenario	Memasukkan <i>username admin</i> yang telah tersimpan di <i>database</i>
Kondisi Awal	Pengguna merupakan <i>admin</i> dan telah masuk dalam halaman administrator. Halaman administrator telah dibuka pada <i>browser</i> .
Data Uji	Data uji berupa <i>username 'admin355'</i> dan <i>password</i>
Langkah Pengujian	Pengguna memasukkan <i>username</i> dan <i>password</i> lalu memilih tombol 'tambah'

Hasil Diharapkan	Yang	Aplikasi menampilkan username telah ada di <i>database</i>
Hasil Didapat	Yang	Aplikasi menampilkan username telah ada di <i>database</i>
Hasil Pengujian		Berhasil
Kondisi Akhir		Tampilan menambahkan administrator baru ditunjukkan pada Gambar 5.3.



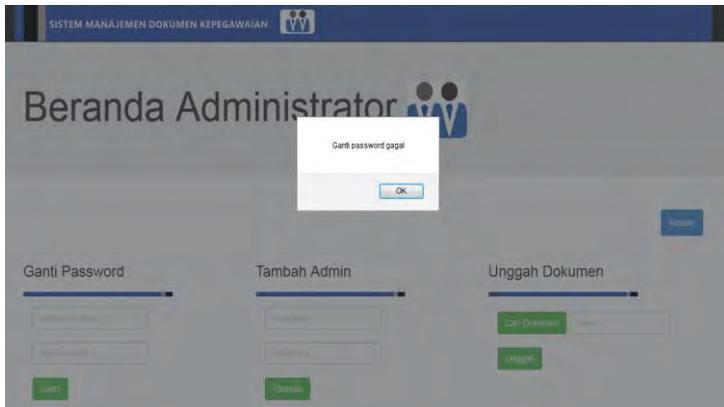
Gambar 5.3 Hasil Uji Fitur Menambah Administrator

5.2.1.4. Pengujian Fitur Mengganti *Password*

Pengujian fitur mengganti *password* pada aplikasi adalah pengujian terhadap kemampuan aplikasi dalam mengganti *password* baru melalui aplikasi. Rincian skenario pengujian pada kasus penggunaan mengganti *password* dapat dilihat pada Tabel 5.4.

Tabel 5.4 Pengujian Fitur Mengganti *Password*

ID	UJ.UC-0004
Referensi Kasus Penggunaan	UC-0004
Nama	Pengujian fitur mengganti <i>password</i>
Tujuan Pengujian	Menguji fitur untuk mengganti <i>password</i> baru melalui aplikasi
Skenario	Salah memasukkan <i>password</i> lama
Kondisi Awal	Pengguna merupakan <i>admin</i> dan telah masuk dalam halaman administrator. Halaman administrator telah dibuka pada <i>browser</i> .
Data Uji	Data uji berupa <i>password</i> lama yang salah dan <i>password</i> baru
Langkah Pengujian	Pengguna memasukkan <i>password</i> lama yang salah dan <i>password</i> baru lalu memilih tombol ‘ganti’
Hasil Yang Diharapkan	Aplikasi menampilkan pesan ‘ganti <i>password</i> gagal’
Hasil Yang Didapat	Aplikasi menampilkan pesan ‘ganti <i>password</i> gagal’
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan pesan ‘ganti <i>password</i> gagal’ ditunjukkan pada Gambar 5.4.

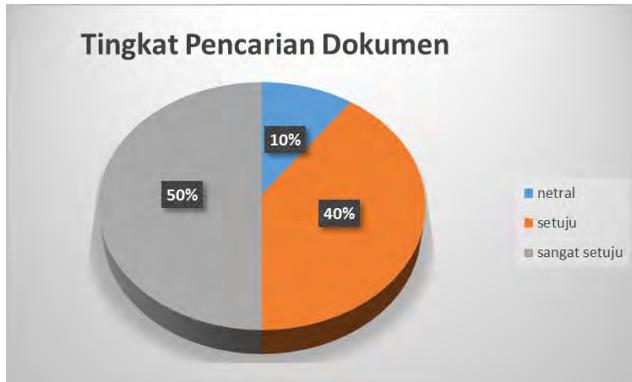


Gambar 5.4 Hasil Uji Fitur Mengganti *Password*

5.2.2. Pengujian Kegunaan Sistem

Pengujian kegunaan sistem dilakukan dengan cara melakukan survei pada pengguna dari aplikasi. Pengguna diminta untuk menguji fitur-fitur yang terdapat pada aplikasi kemudian diminta untuk mengisi kuisioner pendapat pengguna tentang aplikasi. Dalam pengujian kegunaan ini terdapat 10 orang responden. Pengujian ditekankan pada pengukuran tingkat kegunaan aplikasi dalam membantu pencarian dokumen, tingkat akurasi pengelompokan, tingkat kemudahan akses dan penilaian fitur aplikasi.

Detail mengenai kuisioner pengguna dan profil responden dapat dilihat pada Lampiran C. Hasil Pengujian. Hasil survei tingkat kegunaan aplikasi dalam pencarian dokumen dibagi menjadi tiga parameter yaitu informasi akurat, detail dan relevan serta pencarian sesuai kebutuhan. Hasil survei tingkat kegunaan aplikasi dalam membantu pencarian dokumen ditunjukkan pada Gambar 5.5.



Gambar 5.5 Grafik Hasil Survei Tingkat Kegunaan Aplikasi dalam Membantu Pencarian Dokumen

Hasil survei tingkat akurasi dalam pengelompokan dokumen dibagi menjadi dua parameter yaitu pengelompokan sesuai kebutuhan dan pengelompokan membantu proses pencarian. Hasil survei tingkat akurasi pengelompokan dokumen ditunjukkan pada Gambar 5.6.



Gambar 5.6 Grafik Hasil Survei Tingkat Akurasi Pengelompokan Dokumen

Hasil survei tingkat kemudahan akses dibagi menjadi dua parameter yaitu mudah dipahami, mudah digunakan, input mudah dan dokumen mudah diunduh. Hasil survei tingkat kemudahan akses ditunjukkan pada Gambar 5.7.



Gambar 5.7 Grafik Hasil Survei Tingkat Kemudahan Akses

Hasil survei penilaian fitur aplikasi dibagi menjadi tiga fitur yaitu unggah, pencarian dan pengelompokan.. Hasil survei penilaian fitur aplikasi ditunjukkan pada Gambar 5.8.



Gambar 5.8 Grafik Hasil Survei Penilaian Fitur Aplikasi

5.3. Skenario Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional, evaluasi hasil pengelompokan dan evaluasi kegunaan sistem.

5.3.1. Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.5. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik disimpulkan bahwa fungsionalitas dari program telah bisa bekerja sesuai dengan yang diharapkan.

Tabel 5.5 Rangkuman Hasil Pengujian

ID	Nama	Skenario	Hasil
UJ.UC-0001	Pengujian fitur melakukan pencarian dokumen	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ.UC-0002	Pengujian fitur untuk mengunggah dokumen	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ.UC-0003	Pengujian fitur untuk menambah administrator	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ.UC-0004	Pengujian fitur untuk mengganti administrator	Skenario 1	Berhasil

5.3.2. Evaluasi Hasil Pengelompokan

Perhitungan presisi dan *recall* dilakukan untuk mengukur performa dari sistem. Presisi dihitung dari jumlah dokumen relevan yang berhasil dideteksi dibandingkan dengan seluruh hasil deteksi. Sedangkan *recall* didapatkan dari perbandingan jumlah dokumen relevan yang berhasil dideteksi dengan jumlah dokumen relevan yang sebenarnya yang didapat dari analisis pengembang aplikasi.

Perhitungan dan kriteria presisi dan recall telah dijelaskan pada subbab 2.9. Pengujian presisi dan *recall* menggunakan data uji sebanyak 50 dokumen yang diambil secara acak dari *database*.

Tabel 5.6 Hasil Perhitungan Presisi dan Recall Pengelompokan Dokumen

Kategori	TP	FP	FN	TN	Presisi	Recall
Dosen	35	2	1	9	0,94	0,97
Jabatan	9	1	5	32	0,9	0,64
Tenaga Kependidikan	21	4	3	21	0,84	0,87
Rata-rata		7	9	69	0,89	0,83

Dari Tabel 5.6 Hasil Perhitungan Presisi dan Recall Pengelompokan Dokumen didapatkan nilai presisi dan *recall* dari tiap kategori dengan perhitungan sebagai berikut:

a) Kategori dosen

Didapatkan hasil presisi 0.94 dan *recall* 0.97 menunjukkan bahwa dari 50 dokumen yang di uji secara acak, ditemukan dokumen berlabel dosen sebanyak 34 yang dikategorikan benar oleh sistem. Hasil tersebut dinamakan *true positive*. Sedangkan, nilai *false positive* diwakili oleh 2 dokumen yang salah dikategorikan sebagai benar, dan nilai *false negative* diwakili oleh 1 dokumen yang seharusnya benar namun dikategorikan sebagai bukan kategori dosen.

b) Kategori jabatan

Didapatkan hasil presisi 0.9 dan *recall* 0.64 menunjukkan bahwa dari 50 dokumen yang di uji secara acak, ditemukan dokumen berlabel jabatan sebanyak 9 yang dikategorikan benar oleh sistem. Hasil tersebut dinamakan *true positive*. Sedangkan, nilai *false positive* diwakili oleh 1 dokumen yang salah dikategorikan sebagai benar, dan nilai *false negative* diwakili oleh 5 dokumen yang seharusnya benar namun dikategorikan sebagai bukan kategori jabatan.

- c) Kategori tenaga kependidikan
Didapatkan hasil presisi 0.84 dan *recall* 0.87 menunjukkan bahwa dari 50 dokumen yang di uji secara acak, ditemukan dokumen berlabel tenaga kependidikan sebanyak 21 yang dikategorikan benar oleh sistem. Hasil tersebut dinamakan *true positive*. Sedangkan, nilai *false positive* diwakili oleh 4 dokumen yang salah dikategorikan sebagai benar, dan nilai *false negative* diwakili oleh 3 dokumen yang seharusnya benar namun dikategorikan sebagai bukan kategori tenaga kependidikan.

Dari ketiga kategori tersebut didapatkan rata-rata nilai presisi dan dokumen kepegawaian yaitu 0,89 dan nilai *recall* 0,83 menunjukkan bahwa dari 50 dokumen yang di uji secara acak, ditemukan sebanyak 35 dokumen berlabel dosen, 9 dokumen berlabel jabatan dan 21 berlabel tenaga kependidikan yang dikategorikan benar oleh sistem. Hasil tersebut dinamakan *true positive*. Sedangkan, nilai *false positive* diwakili oleh 7 dokumen yang salah dikategorikan sebagai benar, dan nilai *false negative* diwakili oleh 9 kategori dokumen yang seharusnya benar namun dikategorikan sebagai bukan kategori yang benar. Dapat disimpulkan performa dari sistem untuk melakukan pengelompokan dokumen dengan baik dan benar.

5.3.3. Evaluasi Pengujian Kegunaan Sistem

Berdasarkan survei yang dilakukan terhadap 10 responden, didapatkan sekitar 95% responden berpendapat bahwa aplikasi ini dapat membantu mempermudah pengguna dalam mencari dokumen tertentu dengan sekitar 90% responden berpendapat bahwa tingkat kemudahan akses aplikasi baik. Akan tetapi pada penilaian tingkat akurasi pengelompokan hanya 86% responden yang berpendapat bahwa tingkat akurasi sistem baik. Tingkat Penilaian fitur aplikasi menghasilkan 93%. Oleh karena itu dapat

disimpulkan bahwa tujuan dari aplikasi ini yaitu untuk membantu mempermudah pengguna dalam mencari dokumen telah tercapai.

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil selama proses perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut:

1. Aplikasi dapat melakukan pengelompokan dokumen kepegawaian dengan metode klasifikasi naïve bayes dan pembobotan TF dengan nilai presisi 0,89 dan nilai *recall* 0,83.
2. Aplikasi dapat mempermudah pengguna dalam mencari dokumen kepegawaian yang dibutuhkan. Hasil survei terhadap 10 orang pengguna aplikasi dimana 95% pengguna menyatakan aplikasi dapat membantu pengguna dalam melakukan pencarian dokumen.
3. Metode Naïve Bayes merupakan metode klasifikasi yang mudah diimplementasikan karena sederhana dan cocok untuk menyelesaikan masalah klasifikasi.

6.2. Saran

Berikut saran-saran untuk pengembangan dan perbaikan sistem di masa yang akan datang. Diantaranya adalah sebagai berikut:

1. Memperindah tampilan *website* yang ada.
2. Meningkatkan tingkat ketelitian dari sistem dengan menambah jumlah dokumen pada kategori sehingga model yang dihasilkan memiliki kualitas yang lebih baik.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] E. Setiawan, "Kamus Besar Bahasa Indonesia (KBBI)," Badan Pengembangan dan Pembinaan Bahasa, Kemdikbud, 2012-2014. [Online]. Available: <http://www.kbbi.web.id/kelompok>. [Accessed 1 Juli 2014].
- [2] T. Suryana, "Sistem Manajemen Dokumen Komunikasi Internal," Bandung, 2013.
- [3] B. D. Trisedya, "Klasifikasi Dokumen Dengan Algoritma Naïve Bayes dengan Penambahan Parameter Probabilitas Parent," Ahmad Rizal Afani, 2009.
- [4] P. Chairina, "Pengenalan Individu berdasarkan GAIT," Bandung, Departemen Teknik Informatika Institut Teknologi Telkom, 2008.
- [5] J. P. S Dumais and M. S. D Heckerman, "Inductive Learning Algorithms and Representations," in *Workshop on Text Categorization*, 1998, p. 98.
- [6] H. G.-M. L. P. Junghoo Cho, "Efficient crawling through URL ordering," in *InProceedings ofthe Seventh International World WideWeb Conference*, April 1998, pp. 162-172.
- [7] P. Nugroho, "Klasifikasi Email Spam dengan Metode Naive Bayes Classifier menggunakan Java Programming," in *Tugas Akhir Politeknik Elektronika Negri Surabaya*, Surabaya, Institut Teknologi Sepuluh Nopember, 2010.
- [8] R. Andriana, "Pengklasifikasian Spam Email menggunakan Algoritma Improved K-Nearest Neighbors," in *Tugas Akhir Jurusan Ilmu Komputer*, Malang, Universitas Brawijaya, 2012.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, Anita Susanti, lahir di kota Sukoharjo pada tanggal 20 Februari 1992. Penulis adalah anak pertama dari dua bersaudara dan dibesarkan di kota Sukoharjo, Jawa Tengah.

Penulis menempuh pendidikan formal di SD Muhammadiyah Wonorejo (1998-2004), SMPN 1 Mojolaban (2004-2007), SMAN 1 Sukoharjo (2007-2010). Pada tahun 2010, penulis memulai pendidikan S1 jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur.

Di jurusan Teknik Informatika, penulis mengambil bidang minat Rekayasa Perangkat Lunak. Penulis juga aktif dalam organisasi kemahasiswaan seperti Himpunan Mahasiswa Teknik Computer (HMTC) baik sebagai staf maupun kepala departemen kesejahteraan mahasiswa dan Keluarga Muslim Informatika. Dan penulis juga beberapa kali menjadi asisten dosen baik di Teknik Informatika maupun PIKTI-ITS, diantaranya Asisten Dosen mata kuliah Riset Operasional, Asisten Dosen mata kuliah Aljabar Linier, Asisten Dosen mata kuliah Sosio Etika, dan Asisten PIKTI Teknologi Pembuatan Game. Penulis dapat dihubungi melalui alamat email 145anita@gmail.com.

LAMPIRAN A. DIAGRAM ALIR

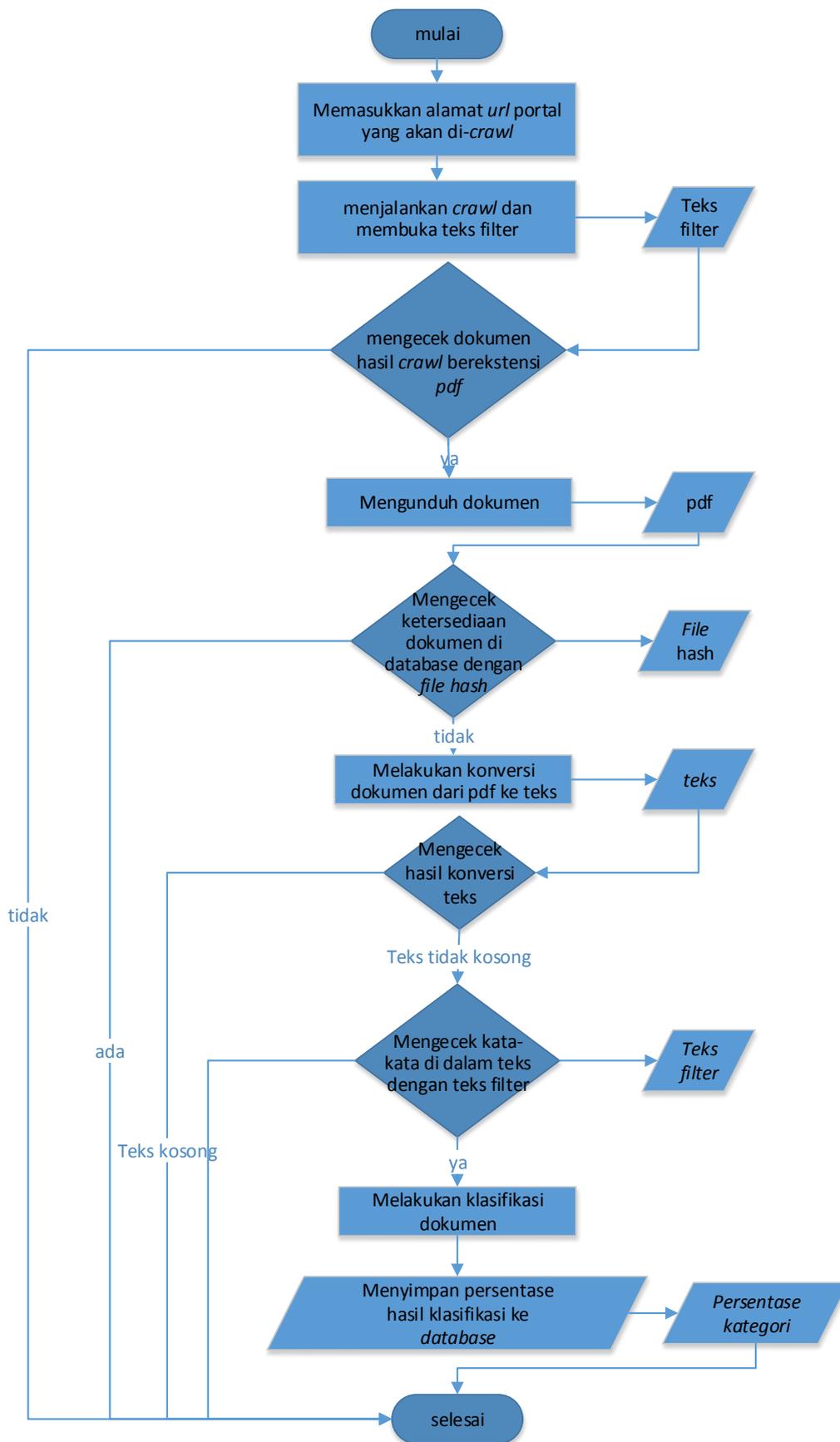


Diagram Alir 7.1 Proses Pengambilan Dokumen dari Portal yang Memuat Dokumen Kepegawaian

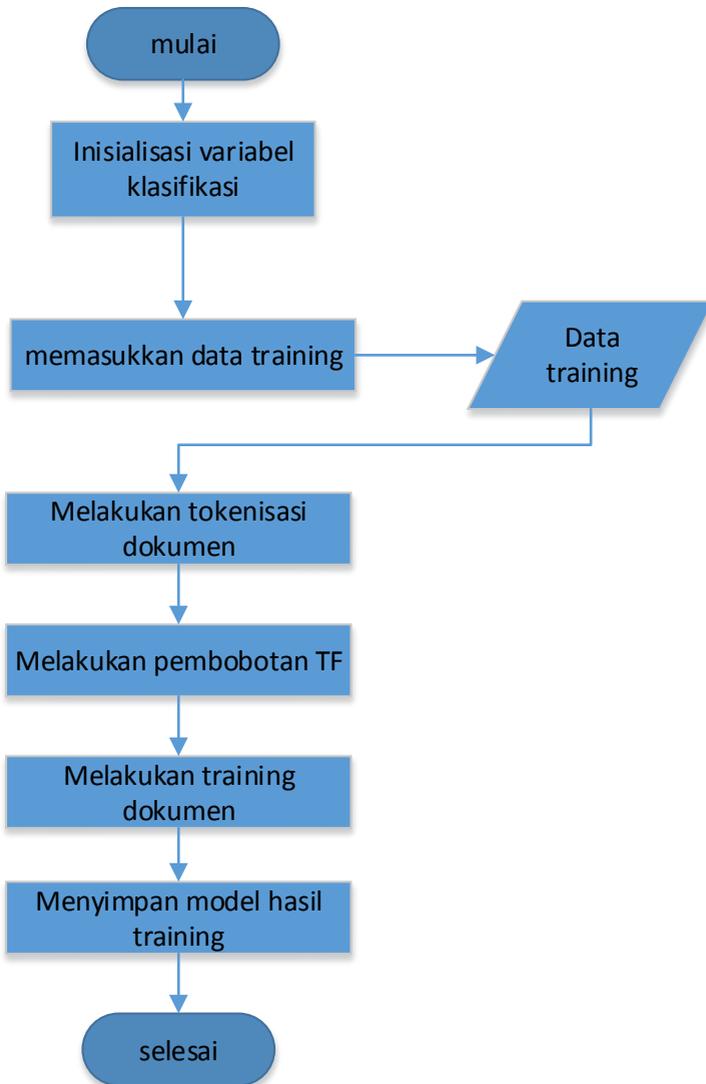
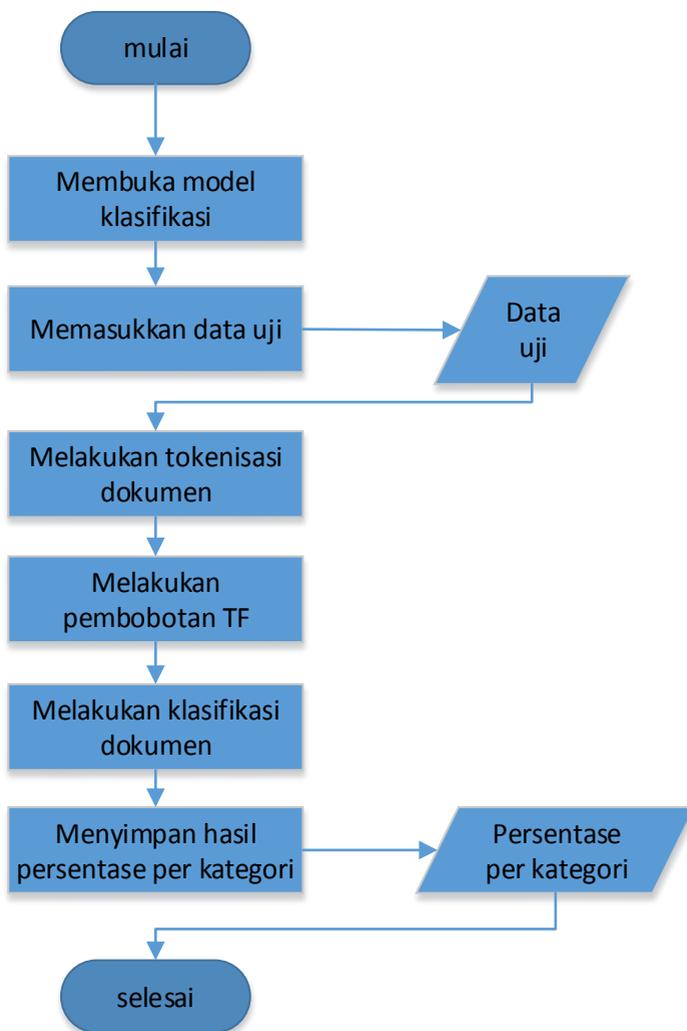


Diagram Alir 7.2 Pelatihan Proses Pengelompokan Dokumen Kepegawaian Menggunakan Klasifikasi Naïve Bayes



**Diagram Alir 7.3 Testing Proses Pengelompokan Dokumen
Kepewajiban Menggunakan Klasifikasi Naïve Bayes**

LAMPIRAN B.KODE SUMBER

Kode Sumber 8.1 Fungsi *Index* pada Kelas Pencarian

```

public function index($searchTerm = null)
{
    if ($searchTerm === null || trim($searchTerm) === " ) {
        $searchTerm = $this->input->post('searchTerm');
    }
    $tahun = $this->input->post('tahun');
    $searchTerm = rawurldecode($searchTerm);
    $data['searchTerm'] = $searchTerm;
    $data['tahunSearchTerm'] = $tahun;

    $totalUri    = $this->uri->total_segments();
    $totalHasilCari = $this->dokumen-
>totalHasilCari($searchTerm, $tahun);
    $config['base_url'] =
'http://simadok.if.its.ac.id/pencarian/index/'.$searchTerm;
    $config['total_rows'] = $totalHasilCari;
    $config['per_page'] = 5;
    $config['uri_segment'] = $totalUri;

    $this->pagination->initialize($config);
    $start = ($this->uri->segment($totalUri) &&
is_numeric($this->uri->segment($totalUri))) ? $this->uri-
>segment($totalUri) : '0';

    $data['searchResult'] = $this->dokumen->hasilCari($start,
$config['per_page'], $searchTerm, $tahun);
    $data['totalhasil'] = $totalHasilCari;
    $data['links']
= $this->pagination-
>create_links();
    $data['daftarTahun'] = $this->dokumen->daftarTahun();
    $this->load->view('pencarian', $data);
}

```

Kode Sumber 8.2 Fungsi *Download* pada Kelas Pencarian

```

public function download($id) {
    $pdf = $this->dokumen->download($id);
    header('Content-type: application/pdf');
    echo $pdf->row()->pdf_file;
}

```

Kode Sumber 8.3 Fungsi *Kategori* pada Kelas Pencarian

```

public function kategori($tag) {
    $tag = str_replace("%20", "_", $tag);
    $totalUri = $this->uri->total_segments();
    $banyakPerKategori = $this->dokumen->banyakPerKategori();
    $config["base_url"] =
'http://simadok.if.its.ac.id/pencarian/kategori/'. $tag;
    $config['total_rows'] = $banyakPerKategori[$tag];
    $config['per_page'] = 5;
    $config['uri_segment'] = $totalUri;

    $data['totalhasil'] = $banyakPerKategori[$tag];
    $this->pagination->initialize($config);
    $start = ($this->uri->segment($totalUri) && is_numeric($this-
>uri->segment($totalUri))) ? $this->uri->segment($totalUri) : '0';

    $data['searchResult'] = $this->dokumen-
>queryKategoriLimit($start, $config['per_page'], $tag);
    $data['daftarTahun'] = $this->dokumen->daftarTahun();
    $data['links'] = $this->pagination->create_links();
    $data['searchTerm'] = "";
    $data['tahunSearchTerm'] = "";
    $this->load->view('pencarian', $data);
}

```

Kode Sumber 8.4 Fungsi Index pada Kelas Main

```

public function index() {
    $data['banyak'] = $this->dokumen-
->banyakPerKategori();
    $data['dokumenTerbaru'] = $this->dokumen-
->dokumenTerbaru();

    $this->load->view('home', $data);
}

```

Kode Sumber 8.5 Fungsi Hasil pada Kelas Main

```

public function hasil(){
    $this->load->view('hasilUnggah');
}

```

Kode Sumber 8.6 Fungsi do_upload pada Kelas Admin

```

public function do_upload() {
    header("Access-Control-Allow-Origin: *");
    $config['upload_path'] = 'upload/';
    $config['allowed_types'] = 'pdf';
    $config['max_size'] = '0';

    $this->load->library('upload', $config);

    if (! $this->upload->do_upload('file')) {
        // echo "<script>alert('tes -1');</script>";
        $data['error'] = $this->upload->display_errors();
        // echo $data['error'];
    }
    else {
        //echo "<script>alert('tes 0');</script>";
        $tahun = $this->input->post('tahun');
        $this->load->model('dokumen');
        $pdfData = $this->upload->data();
        $pdfFile = file_get_contents('upload/'.$pdfData['file_name']);
    }
}

```

```

                                $md5File = md5($pdfFile);
if ($this->dokumen->checkMd5($md5File)) {
    $parser = new \Smalot\PdfParser\Parser();
    $pdf = $parser->parseContent($pdfFile);
    $text = $pdf->getText();
    $percentage = $this->classifier->classify($text);

    $data['nama_file'] = $pdfData['file_name'];
    $data['teks_file'] = $text;
    $data['pdf_file'] = $pdfFile;
    $data['dosen'] = $percentage[0];
    $data['jabatan'] = $percentage[1];
    $data['tenaga_kependidikan'] = $percentage[2];
    $data['hash'] = md5($data['pdf_file']);
    $data['tahun'] = $tahun;
        if (trim($text) !== "")
            $this->dokumen->tambahDokumen($data);
            else
                $data['error'] = 'Dokumen kosong.';
            } else {
                $data['error'] = 'Dokumen telah ada dalam
database.';
            }
        }

    unlink('upload/'.$pdfData['file_name']);
    }
    $result = $this->load->view('hasilUnggah', $data, true);
    $json = array('result' => $result, 'id' => 'id');
    echo json_encode($json);
}

```

Kode Sumber 8.7 Fungsi gantiPassword pada Kelas Admin

```

public function gantiPassword() {
    $this->load->model('mdlAdmin');
    $username = $this->session->userdata('user');
    $return = $this->mdlAdmin->gantiPassword($username,
    md5($this->input->post('pass_lama')),md5($this-
    >input->post('pass_baru')));
    if ($return) {
        $data['error'] = 'Ganti password berhasil!';
    } else {
        $data['error'] = 'Ganti password gagal!';
    }
    $this->session->set_flashdata('gantiPassData', $data);
    redirect('/admin');
}

```

Kode Sumber 8.8 Fungsi tambahAdmin pada Kelas Admin

```

public function tambahAdmin() {
    $username = $this->input->post('username');
    $password = $this->input->post('pass');
    if (trim($username) !== "" && trim($password) !== "") {
        $hashPassword = md5($password);
        $this->load->model('mdlAdmin');

        $jumlah = $this->mdlAdmin->jumlahAdmin();
        $sidAdmin=$this->mdlAdmin->lastIdAdmin($jumlah);
        $stemp = explode("A",$sidAdmin);
        $stemp = "A".((int)$stemp[1]+1);
        $data = array(
            'id_admin' => $stemp,
            'username' => $username,
            'password' => $hashPassword
        );
        $berhasil = $this->mdlAdmin->tambahAdmin($data);

        if ($berhasil) {

```

```

        $data['message'] = "Tambah admin berhasil.";
        } else {
        $data['message'] = "Username sudah ada.";
        }
    } else {
        $data['message'] = "Tambah admin gagal.";
    }
}
$this->session->set_flashdata('tambahAdminData', $data);
redirect(base_url().'admin');
}

```

Kode Sumber 8.9 Fungsi *Pelatihan* pada Kelas *TextClassifier*

```

public void pelatihan() throws IOException {
    System.out.println("Mulai membuat classifier...");
    System.out.println("Membaca folder dosen");
    String trainDosen[] = datasetFromDir(Constants.dokumen +
"PELATIHAN\\TEXT\\DOSEN\\");
    System.out.println("Membaca folder jabatan");
    String trainJabatan[] = datasetFromDir(Constants.dokumen +
"PELATIHAN\\TEXT\\JABATAN\\");
    System.out.println("Membaca folder tenaga kepegawaian");
    String trainTenKep[] = datasetFromDir(Constants.dokumen +
"PELATIHAN\\TEXT\\TENAGA KEPENDIDIKAN\\");

    /*String          trainDosen[]           =
(String[])ArrayUtils.subarray(dataDosen , 0, 24);
    String          trainJabatan[]         =
(String[])ArrayUtils.subarray(dataJabatan, 0, 24);
    String          trainTenKep[]          =
(String[])ArrayUtils.subarray(dataTenKep , 0, 24);*/

    clDosen=makeClassifier(trainDosen,
(String[])ArrayUtils.addAll(trainJabatan, trainTenKep));
    System.out.println("Classifier dosen selesai");
}

```

```

        clJabatan          =          makeClassifier(trainJabatan,
(String[])ArrayUtils.addAll(trainDosen, trainTenKep));
        System.out.println("Classifier jabatan selesai");

        clTenKep          =          makeClassifier(trainTenKep
(String[])ArrayUtils.addAll(trainJabatan, trainDosen));
        System.out.println("Classifier tenaga kepegawaian selesai");
        try {
            clDosen.buildIfNeeded();
            clJabatan.buildIfNeeded();
            clTenKep.buildIfNeeded();
        } catch (Exception ex) {

Logger.getLogger(TextClassifier.class.getName()).log(Level.SEVERE, null, ex);
        }
        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(Constants.root + "br.model"))) {
            oos.writeObject(this);
            oos.flush();
        }
    }
}

```

Kode Sumber 8.10 Fungsi Testing pada Kelas *TextClassifier*

```

public void testing() throws Exception {
    File dirDosen          =          new File(Constants.dokumen +
"TESTING\\TEXT\\DOSEN\\");
    File dirJabatan       =          new File(Constants.dokumen +
"TESTING\\TEXT\\JABATAN\\");
    File dirTenKep        =          new File(Constants.dokumen +
"TESTING\\TEXT\\TENAGA KEPENDIDIKAN\\");

    int nDosen           =          dirDosen.listFiles().length;
    int nJabatan         =          dirJabatan.listFiles().length;
    int nTenKep          =          dirTenKep.listFiles().length;

```

```

System.out.println("Dosen");
for (int i = 0; i < nDosen; i++) {
    File file = dirDosen.listFiles()[i];
    String str = FileToString(file);
    String dosenPercent = String.format("%.2f",
cIDosen.classifyMessage(str) * 100);
    String jabatanPercent = String.format("%.2f",
cIJabatan.classifyMessage(str) * 100);
    String tenKepPercent = String.format("%.2f",
cITenKep.classifyMessage(str) * 100);

    System.out.println(dosenPercent + "\t" + jabatanPercent +
"\t" + tenKepPercent + "\t" + file.getName());
}
System.out.println("Jabatan");
for (int i = 0; i < nJabatan; i++) {
    File file = dirJabatan.listFiles()[i];
    String str = FileToString(file);
    String dosenPercent = String.format("%.2f",
cIDosen.classifyMessage(str) * 100);
    String jabatanPercent = String.format("%.2f",
cIJabatan.classifyMessage(str) * 100);
    String tenKepPercent = String.format("%.2f",
cITenKep.classifyMessage(str) * 100);

    System.out.println(dosenPercent + "\t" + jabatanPercent +
"\t" + tenKepPercent + "\t" + file.getName());
}
System.out.println("Tenaga kepegawaian");
for (int i = 0; i < nTenKep; i++) {
    File file = dirTenKep.listFiles()[i];
    String str = FileToString(file);
    String dosenPercent = String.format("%.2f",
cIDosen.classifyMessage(str) * 100);

```

```

        String jabatanPercent = String.format("%.2f",
clJabatan.classifyMessage(str) * 100);
        String tenKepPercent = String.format("%.2f",
clTenKep.classifyMessage(str) * 100);

        System.out.println(dosenPercent + "\t" + jabatanPercent +
"\t" + tenKepPercent + "\t" + file.getName());
    }
}

```

Kode Sumber 8.11 Kelas Crawl

```

class MyCrawler extends PHPCrawler
{
    function handleDocumentInfo($DocInfo)
    {
        // Just detect linebreak for output ("\n" in CLI-mode, otherwise
"<br>").
        if (PHP_SAPI == "cli") $lb = "\n";
        else $lb = "<br />";

        // Print the URL and the HTTP-status-Code
        echo "Page requested: ".$DocInfo->url." (". $DocInfo->
http_status_code.").$lb;

        // Open filter file
        $filters = file('C:\xampp\htdocs\PHPCrawl_082\filter.txt');

        // Print the referring URL
        echo "Referer-page: ".$DocInfo->referer_url.$lb;

        // Print if the content of the document was be recieved or not
        if ($DocInfo->received == true)
            echo "Content received: ".$DocInfo->bytes_received." bytes".$lb;
        else
            echo "Content not received".$lb;

        // Check whether pdf file or not

```

```

    $ext = pathinfo($DocInfo->url, PATHINFO_EXTENSION);
    if ($ext === 'pdf') {
        echo ($DocInfo->url).$lb;
        // Connect to database
        $mysqli = new mysqli("localhost", "root", "",
"kepegawaian");
        if ($mysqli->connect_errno) {
            echo "Failed to connect to MySQL: " .
mysqli_connect_error().$lb;
        }

        // Download file
        //$content = file_get_contents($DocInfo->url);
        $content = $DocInfo->source;

        // Check file hash
        $result = $mysqli->query("select count(*) as total from
kategori where hash='".md5($content)."'");
        $data = $result->fetch_assoc();
        $result->close();
        try {
            if ($data['total'] == 0) {
                // Convert pdf to text
                $path = explode('/', $DocInfo->url);
                $year = $path[5];
                $name = basename($DocInfo->url);
                $parser = new
\Smalot\PdfParser\Parser();
                $pdf = $parser-
>parseContent($content);
                $text = $pdf->getText();
                file_put_contents('D:\temp', $text);

                // Filter
                $passFilter = true;
                if (trim($text) === "") {
                    $passFilter = false;
                }
                foreach($filters as $line) {

```

```

                                echo $line;
                                if
(strpos(strtolower($text),strtolower($line)) !== false) {
                                    $passFilter = false;
                                }
                            }
                                if (!$passFilter) {
                                echo "File contains unrelated
words.".$lb;
                                } else {
                                    // Classification
                                    exec("c:\Program
Files\Java\jre7\bin\java" -jar
C:\xampp\htdocs\PHPCrawl_082\classification\KlasifikasiDokumenWek
a.jar D:\temp', $output);
                                $percentage = explode("\t",
$Soutput[0]);
                                $name = str_replace("%20", "
",$name);
                                // Insert to database
                                $string = "insert into
kategori (nama_file, teks_file, pdf_file, dosen, jabatan,
tenaga_kependidikan, hash, tahun) values('$name', '".mysqli-
>escape_string($text)."', '".mysqli->escape_string($content)."',
$percentage[0], $percentage[1], $percentage[2], '".md5($content)."',
"$year."");
                                $result = $mysqli-
>query($string);
                                if (!$result) {
                                    echo $mysqli-
>error.$lb;
                                }
                            }
                        } catch(Exception $e) {
                            echo "-- Error detected on ".$DocInfo-
>url.$lb;
                            echo "-- Message: ".$e->getMessage().$lb;

```

```
    }  
    // Close connection  
    $mysqli->close();  
    echo "Done.".$lb;  
  }  
  echo "Memory usage: ".memory_get_usage().$lb;  
  
  echo $lb;  
  
  flush();  
}  
}
```

LAMPIRAN C. HASIL PENGUJIAN

**Tabel 9.1. Pengujian Fitur Melalui Pencarian Dokumen
Skenario 2**

ID	UJ.UC-0001
Referensi Kasus Penggunaan	UC-0001
Nama	Pengujian fitur melakukan pencarian dokumen pada aplikasi
Tujuan Pengujian	Menguji fitur untuk menampilkan pemberitahuan dokumen tidak ditemukan apabila pengguna tidak memasukkan kata kunci
Skenario 2	Mengosongkan kata kunci pada kolom pencarian
Kondisi Awal	Halaman beranda dari aplikasi telah dibuka pada <i>browser</i> .
Data Uji	Data uji berupa mengosongkan kata kunci
Langkah Pengujian	Pengguna mengosongkan kata kunci pencarian dan memilih tombol 'cari'
Hasil Yang Diharapkan	menampilkan pemberitahuan dokumen tidak ditemukan
Hasil Yang Didapat	menampilkan pemberitahuan dokumen tidak ditemukan
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan hasil deteksi dapat dilihat pada Gambar 9.1



Gambar 9.1. Hasil Uji Melakukan Pencarian Dokumen Skenario 2

Tabel 9.2 Pengujian Fitur Melakukan Pencarian Dokumen Skenario 3

ID	UJ.UC-0001
Referensi Kasus Penggunaan	UC-0001
Nama	Pengujian fitur melakukan pencarian dokumen tertentu pada aplikasi
Tujuan Pengujian	Menguji fitur untuk mencari dokumen berdasarkan kata kunci pengguna
Skenario 1	Dokumen yang bersesuaian dengan kata kunci dari pengguna akan ditampilkan pada panel hasil pencarian.
Kondisi Awal	Halaman beranda dari aplikasi telah dibuka pada <i>browser</i> .
Data Uji	Data uji berupa kata kunci pencarian yaitu ‘tunjangan profesor’

Langkah Pengujian	Pengguna memasukkan kata kunci pencarian dan memilih tombol 'cari'
Hasil Yang Diharapkan	Aplikasi menampilkan dokumen yang bersesuaian dengan kata kunci pengguna di panel hasil pencarian.
Hasil Yang Didapat	Aplikasi menampilkan dokumen yang bersesuaian dengan kata kunci pengguna di panel hasil pencarian.
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan hasil deteksi dapat dilihat pada



Gambar 9.2 Hasil Uji Melakukan Pencarian Dokumen Skenario 3

Tabel 9.3 Pengujian Fitur Mengunggah Dokumen Skenario 3

ID	UJ.UC-0002
Referensi Kasus Penggunaan	UC-0002
Nama	Pengujian fitur mengunggah dokumen

Tujuan Pengujian	Menguji fitur untuk mengunggah dokumen melalui aplikasi
Skenario	Mengunggah dokumen kepegawaian dengan ekstensi selain <i>pdf</i>
Kondisi Awal	Pengguna merupakan <i>admin</i> dan telah masuk dalam halaman administrator. Halaman administrator telah dibuka pada <i>browser</i> .
Data Uji	Data uji berupa dokumen berupa <i>.txt</i>
Langkah Pengujian	Pengguna memasukkan dokumen lalu memilih tombol ‘unggah’
Hasil Yang Diharapkan	Menampilkan pemberitahuan bahwa ‘ <i>file extension error</i> ’ <i>file</i> dokumen tidak sesuai
Hasil Yang Didapat	Menampilkan pemberitahuan bahwa ‘ <i>file extension error</i> ’ <i>file</i> dokumen tidak sesuai
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan pemberitahuan ekstensi <i>file</i> eror ditunjukkan pada Gambar 9.3.

Unggah Dokumen

Error: File extension error.

Cari Dokumen

tahun

Unggah

Gambar 9.3 Hasil Uji Mengunggah Dokumen Skenario 2

Tabel 9.4. Pengujian Fitur Mengunggah Dokumen Skenario 3

ID	UJ.UC-0002
Referensi Kasus Penggunaan	UC-0002
Nama	Pengujian fitur mengunggah dokumen
Tujuan Pengujian	Menguji fitur untuk mengunggah dokumen melalui aplikasi
Skenario	Mengunggah dokumen kepegawaian yang sudah tersedia di <i>database</i>
Kondisi Awal	Pengguna merupakan <i>admin</i> dan telah masuk dalam halaman administrator. Halaman administrator telah dibuka pada <i>browser</i> .
Data Uji	Data uji berupa dokumen kepegawaian yang sudah pernah diunggah

Langkah Pengujian	Pengguna memasukkan dokumen lalu memilih tombol ‘unggah’
Hasil Yang Diharapkan	Menampilkan pemberitahuan bahwa ‘ dokumen telah ada dalam <i>database</i> ”
Hasil Yang Didapat	Menampilkan pemberitahuan bahwa ‘ dokumen telah ada dalam <i>database</i> ”
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan pemberitahuan ekstensi <i>file</i> eror ditunjukkan pada

Unggah Dokumen



File: permendikbud_nomor 78_tahun
2013-Tunjangan Profesor.pdf (71 kb)

Cari Dokumen

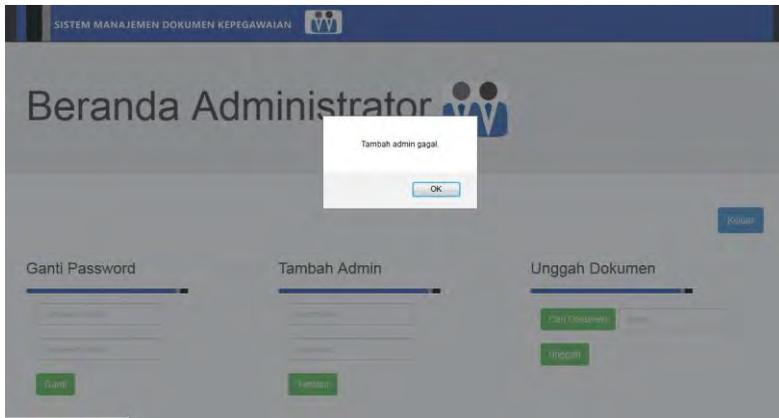
2013

Unggah

Gambar 9.4 Hasil Uji Mengunggah Dokumen Skenario 3

Tabel 9.5 Pengujian Menambah Administrator Skenario 2

ID	UJ.UC-0003
Referensi Kasus Penggunaan	UC-0003
Nama	Pengujian fitur menambah administrator
Tujuan Pengujian	Menguji fitur untuk menambah administrator baru melalui aplikasi
Skenario	Memasukkan <i>username</i> baru dan tidak memasukkan <i>password</i>
Kondisi Awal	Pengguna merupakan <i>admin</i> dan telah masuk dalam halaman administrator. Halaman administrator telah dibuka pada <i>browser</i> .
Data Uji	Data uji berupa mengisi <i>username</i> dan mengosongkan isian <i>password</i>
Langkah Pengujian	Pengguna memasukkan <i>username</i> lalu memilih tombol 'tambah'
Hasil Yang Diharapkan	Aplikasi menampilkan menampilkan pesan 'tambah admin gagal'
Hasil Yang Didapat	Aplikasi menampilkan menampilkan pesan 'tambah admin gagal'
Hasil Pengujian	Berhasil
Kondisi Akhir	Tampilan menambahkan administrator baru ditunjukkan pada Gambar 9.5.



Gambar 9.5 Hasil Uji Menambah Administrator Skenario 2

Tabel 9.6 Profil Responden Pengguna Aplikasi

Responden	Jenis Kelamin	Jabatan
1	Perempuan	Pustakawan
2	Perempuan	RBTC
3	Laki-laki	Dosen
4	Laki-laki	Umum& Perlengkapan
5	Perempuan	Dosen
6	Perempuan	Staf Kemahasiswaan
7	Laki-laki	Staf Akademik
8	Laki-laki	Dosen
9	Perempuan	Dosen
10	Laki-laki	Staf Akademik

Kuisisioner Pengguna

KUESIONER ANALISA TINGKAT KEGUNAAN TUGAS AKHIR SISTEM INFORMASI MANAJEMEN DOKUMEN KEPEGAWAIAN

Dengan hormat,

Bersama ini saya mohon kesediaan anda untuk membantu mengisi kuisisioner yang telah disiapkan dengan maksud :

1. untuk mengumpulkan data yang akan menjadi bagian dari tugas akhir dengan judul “*Rancang Bangun Modul Pengelompokan Dokumen pada Sistem Manajemen Dokumen Kepegawaian*”
2. untuk menguji tingkat kegunaan sistem informasi manajemen kepegawaian dari pengguna di lingkungan ITS.

Tidak ada jawaban yang benar maupun salah, anda diharapkan untuk menjawab pertanyaan-pertanyaan yang diberikan seakurat mungkin berdasarkan pengalaman anda. Kerahasiaan anda akan dijamin sepenuhnya.

Terima kasih atas bantuan dan dukungannya.

Hormat Saya,

(Anita Susanti)

A. KARAKTERISTIK RESPONDEN

Nama :
Jabatan :

B. SISTEM MANAJEMEN DOKUMEN KEPEGAWAIAN

Berikut ini pertanyaan-pertanyaan untuk mengetahui tingkat kegunaan dari sistem informasi manajemen kepegawaian terhadap pengguna. Anda cukup menilai apakah anda setuju atau tidak terhadap pertanyaan yang diajukan dengan ketentuan sebagai berikut:

1 = Sangat tidak setuju

2 = Tidak setuju

3 = Netral

4 = Setuju

5 = Sangat setuju

1. Pencarian Dokumen

No	KETERANGAN	1	2	3	4	5
1	Sistem menghasilkan informasi yang akurat					
2	Informasi yang disajikan sesuai dengan kebutuhan anda					
3	Sistem menyediakan informasi yang detil dan relevan					

2. Kategori Dokumen

No	KETERANGAN	1	2	3	4	5
1	Sistem menghasilkan kategori yang sesuai dengan kebutuhan anda					
2	Informasi yang disajikan jelas untuk dimengerti					

3	kategori memudahkan dalam pencarian dokumen					
---	---	--	--	--	--	--

3. Fitur penunjang

No	KETERANGAN	1	2	3	4	5
1	Kemudahan untuk melihat dokumen yang ada					
2	Dokumen yang disajikan sistem mudah untuk diunduh					

4. Tampilan Sistem

No	KETERANGAN	1	2	3	4	5
1	Sistem yang ada mudah diakses					
2	Sistem yang ada mudah dipahami					
3	Sistem yang ada mudah digunakan					
4	Tata letak (display) mudah dibaca					
5	Sistem memiliki proses input yang mudah					

5. Unggah Dokumen

No	KETERANGAN	1	2	3	4	5
1	Dokumen yang terunggah sudah sesuai kategori					

2	Informasi yang disajikan dapat dipercaya					
3	Sistem yang ada mudah di implementasikan					

6. SARAN

[Halaman ini sengaja dikosongkan]