BAB 3

ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

Bab analisis dan perancangan perangkat lunak berisi analisis aplikasi yang sudah ada serta analisis dan perancangan dari perangkat lunak yang akan dibangun. Pada bab ini akan ditentukan pula kebutuhan dan pemodelan dari perangkat lunak yang akan dibangun, dimana pemodelan menggunakan *Unified Modeling Language* (UML).

3.1 Analisis Aplikasi eBdesk Collaboration

Pada sub bab ini akan dijelaskan mengenai perangkat lunak yang sudah ada sebelumnya, dalam hal ini adalah aplikasi eBdesk Collaboration, dimana aplikasi pencarian dokumen akan diletakkan pada eBdesk Collaboration.

3.1.1 Deskripsi Sistem

Aplikasi eBdesk Collaboration adalah sebuah aplikasi yang ditujukan sebagai sarana untuk berkolaborasi antar pegawai dalam sebuah perusahaan. Beberapa fitur utama dalam aplikasi eBdesk Collaboration adalah *sharing* dokumen, galeri, blog, kegiatan dan forum. Berikut ini akan dijelaskan secara singkat fungsi dari masing-masing fitur yang ada :

1. Pengaturan Profile

   Pada modul aplikasi Profile, pengguna dapat melakukan beberapa fungsionalitas seperti mengganti password, mengganti atau mengupload

foto, mengedit personal informasi pengguna baik itu berupa nama depan, nama lengkap, alamat rumah, alamat email dan sebagainya.

2. Pengelolaan Dokumen

   Pada modul dokumen, pengguna dapat mengelola dokumen seperti mengupload dokumen, mengisi informasi terhadap dokumen yang diupload, mendownload *file* yang ada dalam dokumen.

3. Pengelolaan Galeri

   Pada modul galeri , pengguna dapat mengelola galeri seperti membuat galeri foto baru, menambahkan foto ke galeri yang sudah ada, men*download* foto yang ada di dalam suatu galeri, memberikan komentar terhadap foto yang sudah di upload.

4. Pengelolaan Kegiatan

   Pada modul kegiatan, pengguna dapat melihat kegiatan yang ada, membuat kegiatan baru, menambah informasi pada kegiatan dan memberikan suatu komentar terhadap event.

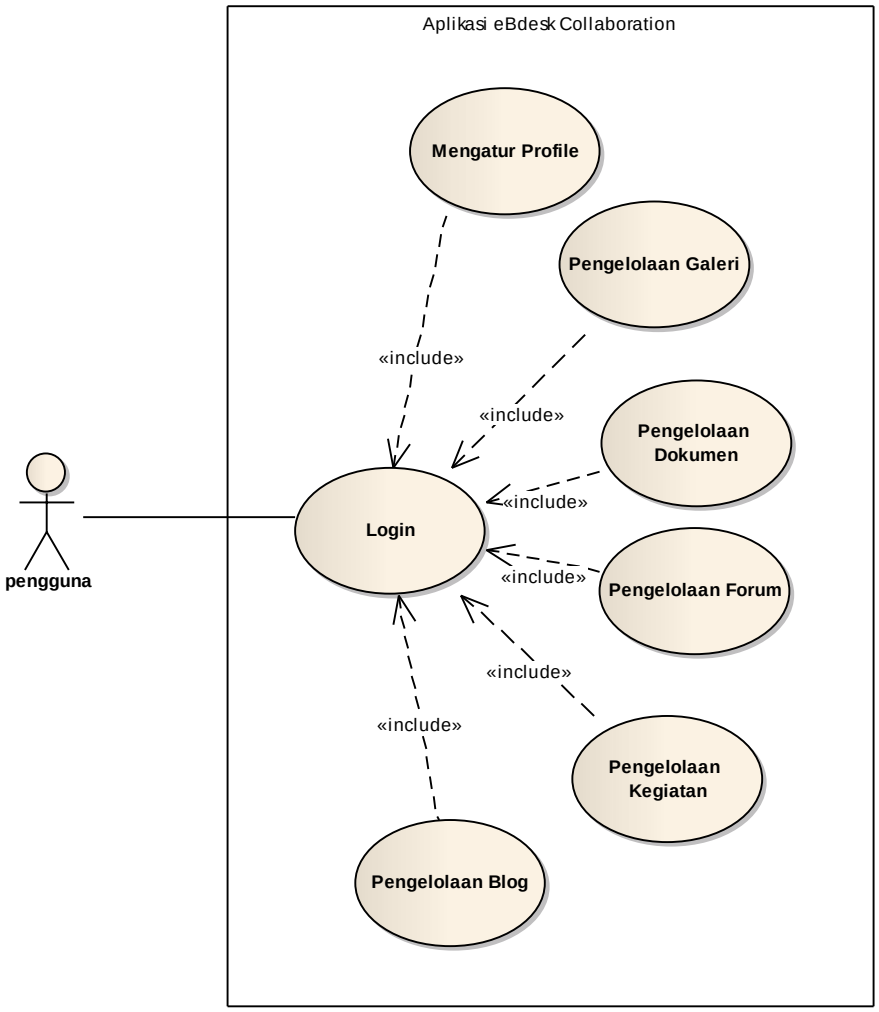
5. Pengelolaan Blog

   Pada modul blog, pengguna dapat mengelola blog seperti membuat blog baru, mengedit dan menghapus blog, dan memberikan komentar terhadap blog yang telah dibuat.

6. Pengelolaan Forum

   Pada modul forum, pengguna bisa saling bertukar informasi, berdiskusi tentang suatu hal dan memberikan komentar terhadap suatu topik yang ada.

3.1.2 *Use Case* Aplikasi eBdesk Collaboration

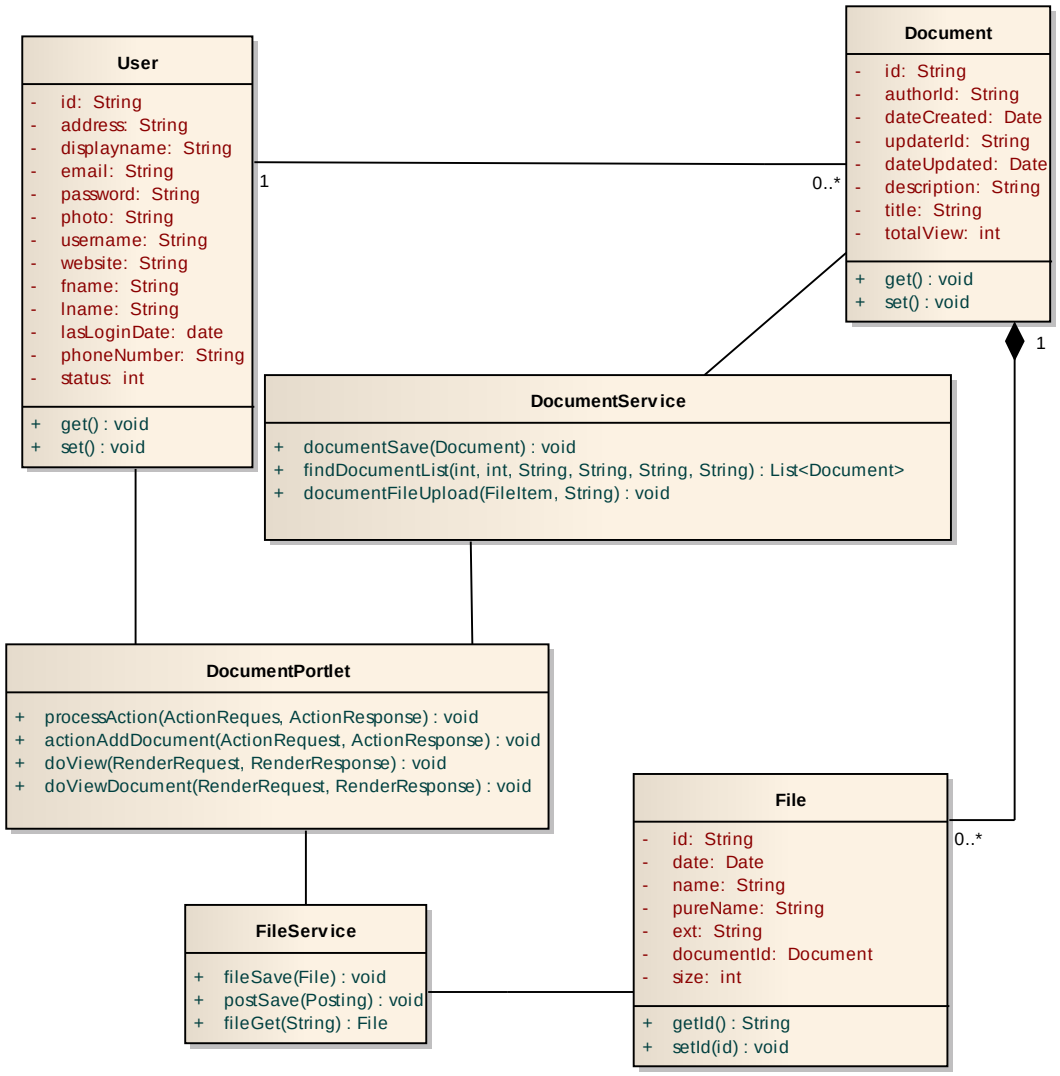


Gambar 3.1 Use Case Aplikasi eBdesk Collaboration

Gambar 3.1 menjelaskan diagram *use case* dari aplikasi eBdesk Collaboration. Dalam diagram *use case* ini hanya terdapat satu jenis aktor, sehingga bisa disebut pengguna. Dalam diagram ini, terdapat tujuh proses yang bisa dilakukan pengguna yaitu : Login, Mengatur Profile, Pengelolaan Galeri,

Pengelolaan Dokumen, Pengelolaan Forum, Pengelolaan Kegiatan, dan Pengelolaan Blog.

3.1.3 Class Diagram Pada Pengelolaan Dokumen



Gambar 3.2 *Class diagram* pada pengelolaan dokumen

Diagram kelas dalam Gambar 3.2 diatas ini memberikan gambaran keterkaitan antara kelas-kelas pada aplikasi pengelolaan dokumen. Seperti terlihat

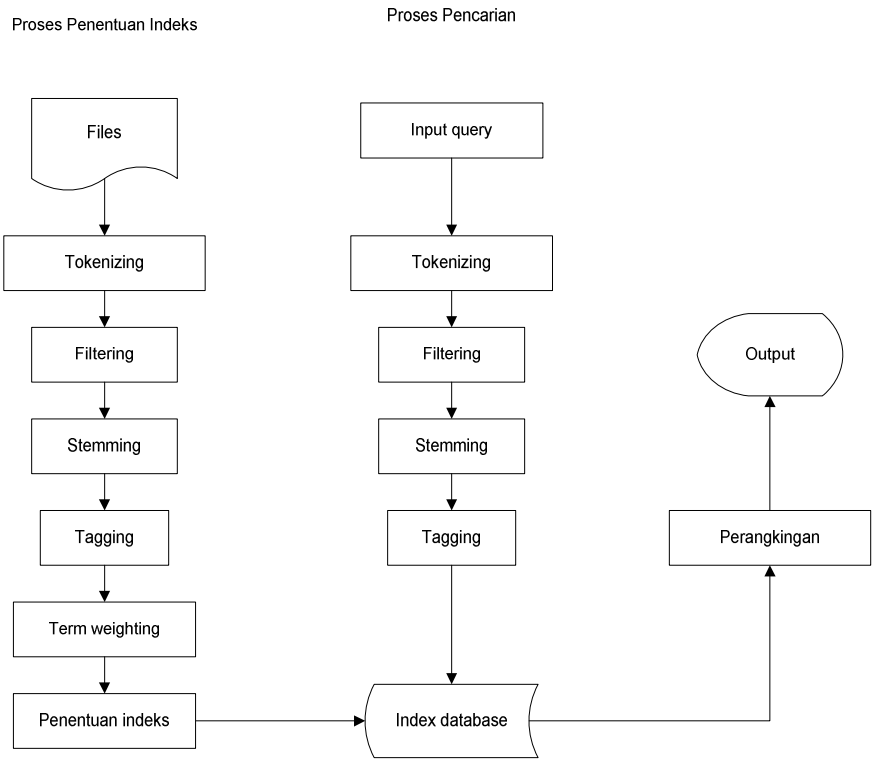
pada Gambar 3.2 bahwa adanya keterhubungan kelas-kelas antara lain : DocumentPortlet, DocumentService, FileService, User, Document, dan File.

3.2 Analisis Perangkat Lunak Yang Akan Dibangun

Pada analisis perangkat lunak ini akan dijelaskan mengenai deskripsi umum dari perangkat lunak yang akan dibangun yaitu aplikasi pencarian dokumen, *use case* model berupa *use case* diagram serta *use case* skenario, realisasi *use case* dan *activity diagram*.

3.2.2 Deskripsi Umum Perangkat Lunak

Perangkat lunak yang akan dibangun adalah perangkat lunak yang mampu melakukan proses penentuan indeks dan proses pencarian. *Input* dari perangkat lunak ini berupa *file text* yang disimpan dalam jenis *file* *.txt dan *.doc. Dan *output* dari perangkat lunak ini adalah *file* yang sesuai dengan *query* yang dimasukkan. Jika dibuat penggambarannya dapat dilihat sebagai berikut :



Gambar 3.3 Perangkat Lunak Aplikasi Pencarian Dokumen

Gambar 3.3 memperlihatkan gambaran perangkat lunak yang akan dibangun, untuk proses penentuan indeks dimulai dengan mengubah isi setiap *file* yang terdapat di dalam *file* koleksi menjadi *term*, proses ini disebut *tokenizing*. Setiap *term* yang dihasilkan dari setiap *file* akan dilakukan proses *filtering*, *stemming* dan *tagging*, lalu dari setiap *term* tersebut akan diberikan bobot (*term weighting*). *Term* yang telah mengalami proses *tokenizing*, *filtering*, *stemming*, *tagging* dan *term weighting* merupakan *term* yang akan menjadi indeks dari *file* koleksi. Pada proses pencarian pengguna memasukkan *query*, lalu *query* tersebut dipecah (*parse query*), hasil dari *parse query* tersebut akan dilakukan proses *stoplist*, *stemming* dan *tagging*, *query* yang telah mengalami proses *parse query*, *filtering*, *stemming* dan *tagging* akan dicari di dalam indeks *database* dan

selanjutnya akan dilakukan perangkingan. Perangkingan disusun berdasarkan bobot masing-masing *file* terhadap *query* pencarian.

3.3 Analisis Kebutuhan Sistem

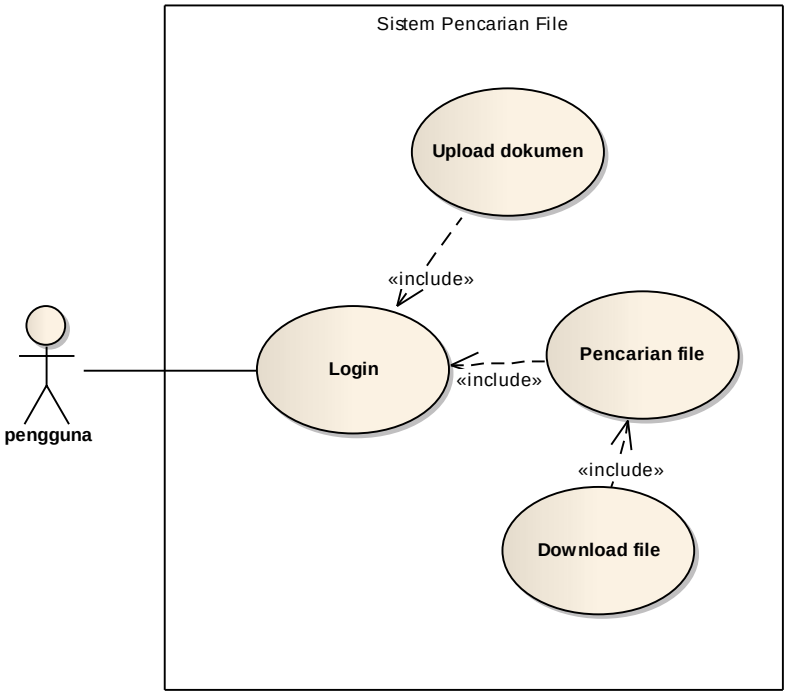
3.3.2 Kebutuhan Fungsional

Kebutuhan fungsional pada sistem ini adalah :

1. *Upload file* dari pengguna
2. Proses penentuan index
3. Input kata kunci dari pengguna
4. Load data dari *database*
5. Proses pembobotan menggunakan metode tf-idf
6. Tampilkan hasil ke pengguna.
7. *Download file* yang dipilih pengguna

3.3.3 Use Case Diagram

Diagram *use case* menggambarkan proses-proses apa saja yang dapat ditangani oleh aplikasi peran *actor-actor* yang terlibat di dalamnya.



Gambar 3.4 Use case diagram

Gambar 3.4 menjelaskan diagram *use case* dari sistem pencarian *file*. Dalam diagram *use case* ini hanya terdapat satu jenis aktor, sehingga bisa disebut pengguna. Dalam diagram ini, terdapat empat proses yang bisa dilakukan pengguna yaitu : Login, *Upload* dokumen, Pencarian *file*, dan *Download file*. Proses login yaitu melakukan otentikasi pengguna yang akan masuk ke aplikasi. Proses upload dokumen yaitu melakukan *upload file*, dan mengindeks *file* ke dalam basis data. Proses pencarian *file* yaitu melakukan pencarian terhadap *file* dan melakukan perangkingan terhadap dokumen hasil pencarian. Proses download file yaitu melakukan proses *download* terhadap *file* yang dipilih. Untuk mengetahui penjelasan lebih detail dari *use case* pada Gambar 3.4 bisa dilihat melalui skenario *use case* pada tabel 3.1 – 3.4 berikut ini :

Tabel 3.1 Skenario *use case* login

Skenario	
Nama Use case : Login	
Nama *Actor* : Pengguna	
Deskripsi : Melakukan pengecekan data pengguna.	
Skenario Awal	
Kondisi Awal **:** Terdapat pengguna yang akan menggunakan aplikasi	
Aksi Aktor	**Respon Sistem**
1. Pengguna memasukkan *username* dan *password*	
	2. Sistem mencari data pengguna dalam basis data berdasarkan *username* dan *password* yang dimasukkan
	3. Jika data tidak ditemukan, maka sistem menampilkan pesan kesalahan. Jika ditemukan maka pengguna masuk ke dalam aplikasi eBdesk Collaboration
4. Tampil halaman utama eBdesk Collaboration atau tampil pesan kesalahan	

Tabel 3.2 Skenario *use case* upload dokumen

Skenario	
Nama Use case : *Upload* Dokumen	
Nama *Actor* : Pengguna	
Deskripsi : Melakukan *upload file*, dan mengindeks *file* ke dalam basis data.	
Skenario Awal	
Kondisi Awal : Terdapat *file* yang belum diindeks Terdapat form untuk meng-*upload* dokumen	
Aksi Aktor	**Respon Sistem**
1. Pengguna memasukkan judul dan deskripsi singkat dokumen	
2. Pengguna mengunggah *file* untuk diindeks ke dalam sistem	
	3. Sistem menyimpan judul dan deskripsi ke dalam basis data
	4. Sistem membaca isi *file* kemudian melakukan proses *tokenizing, filtering, stemming, tagging,* dan *term weighting*
	5. Sistem menyimpan data indeks ke basis data.
6. Pengguna mendapatkan pesan berhasil tidaknya proses pembuatan dokumen.	

Tabel 3.3 Skenario *use case* pencarian *file*

Skenario	
Nama Use case : Pencarian *File*	
Nama *Actor* : Pengguna	
Deskripsi : Melakukan pencarian terhadap *file* dan melakukan perangkingan terhadap dokumen hasil pencarian.	
Skenario Awal	
Kondisi Awal **:** Tersedia form pencarian untuk memasukkan kata kunci pencarian yang diinginkan oleh pengguna.	
Aksi Aktor	**Respon Sistem**
1. Pengguna memasukkan kata kunci pencarian terhadap *file*.	
	2. Sistem kata kunci kemudian melakukan proses *tokenizing, filtering, stemming, tagging* terhadap kata kunci
	3. Server akan mengambil *file* dari *database* yang isinya mengandung kata kunci pencarian
	4. Setelah *query* berhasil dieksekusi dan *file* telah difilter lalu dokumen diproses menggunakan metode tf-idf supaya

	dihasilkan dokumen yang relevan terhadap *query*
	5. Sistem menampilkan *file* sesuai dengan kata kunci pencarian.
6. Pengguna mendapatkan *file* sesuai dengan kata kunci pencarian.	

Tabel 3.4 Skenario *use case download file*

Skenario	
Nama Use case : *Download File*	
Nama *Actor* : Pengguna	
Deskripsi : Melakukan proses *download* terhadap *file* yang dipilih.	
Skenario Awal	
Kondisi Awal **:** Tersedia tampilan daftar dokumen hasil pencarian.	
Aksi Aktor	**Respon Sistem**
1. Pengguna memilih *file* yang akan di *download*	
	2. Sistem mencari *file* fisik yang akan di-*download*
	3. Setelah *file* ditemukan sistem menampilkan popup *download*
4. Pengguna mendapatkan *file* yang diinginkan.	

3.3.4 Analisis Perangkat Lunak

Pembangunan sistem menggunakan beberapa perangkat lunak sebagai berikut:

1. Sistem Operasi : Windows 7 Ultimate
2. Bahasa Pemrograman : Java
3. Web Server: Apache Tomcat 6.0.26
4. Netbeans IDE 6.8 sebagai *editor*
5. Database : MySQL
6. Library
    a. Spring 2.6
    b. Hibernate 3.1
    c. MySQL Connector
    d. POI 3.0

3.3.5 Analisis Perangkat Keras

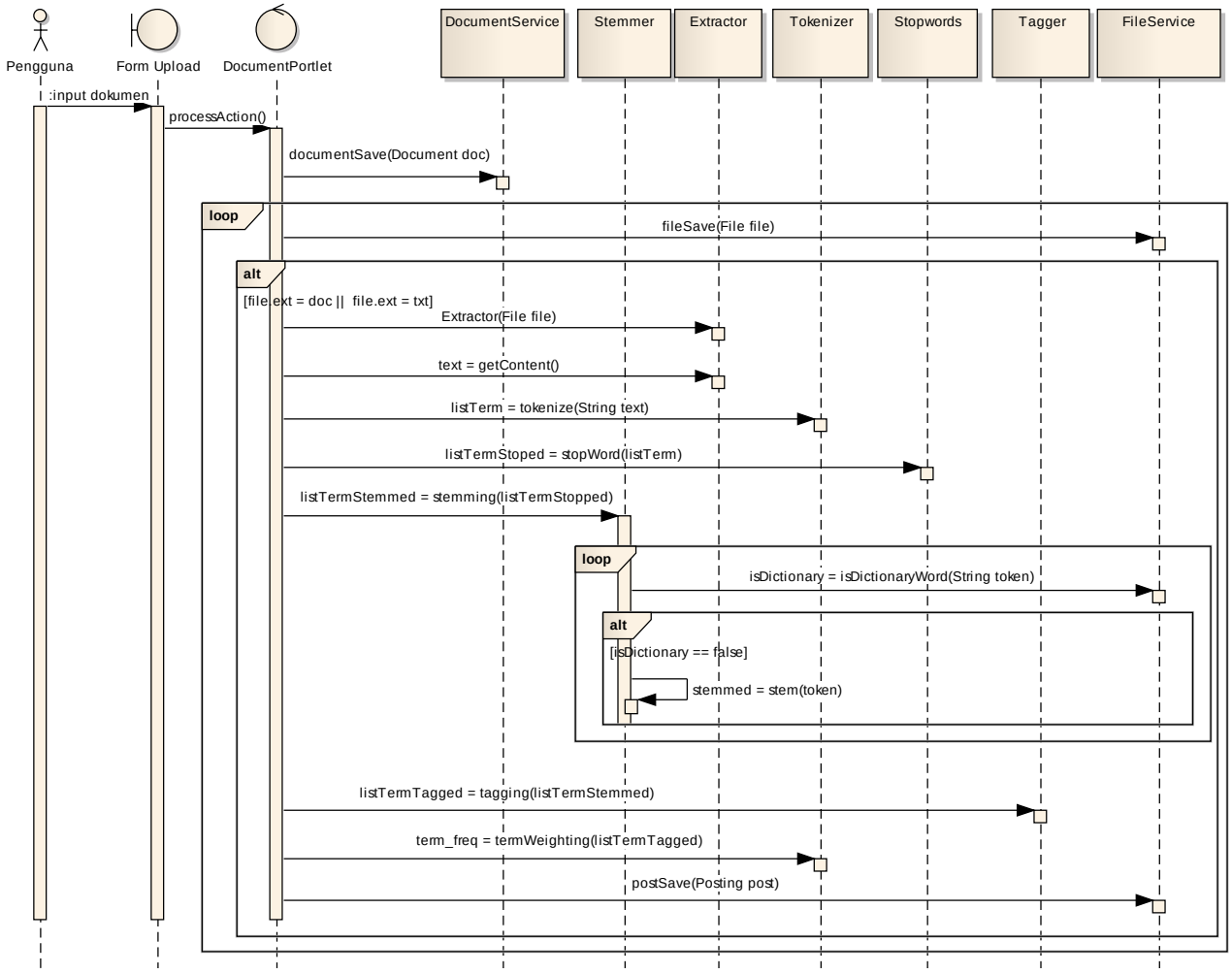
Adapun daftar spesifikasi perangkat keras yang digunakan dalam pembangunan sistem adalah sebagai berikut.

1. Processor Intel(R) Core(TM) 2 Duo processor 2.5GHz
2. Memori 3072MB
3. Hardisk 296GB
4. Monitor 14`
5. *Keyboard* dan *Mouse*

3.4 Perancangan Applikasi

3.4.1 Sequence Diagram

3.4.1.1 Sequence diagram upload dokumen



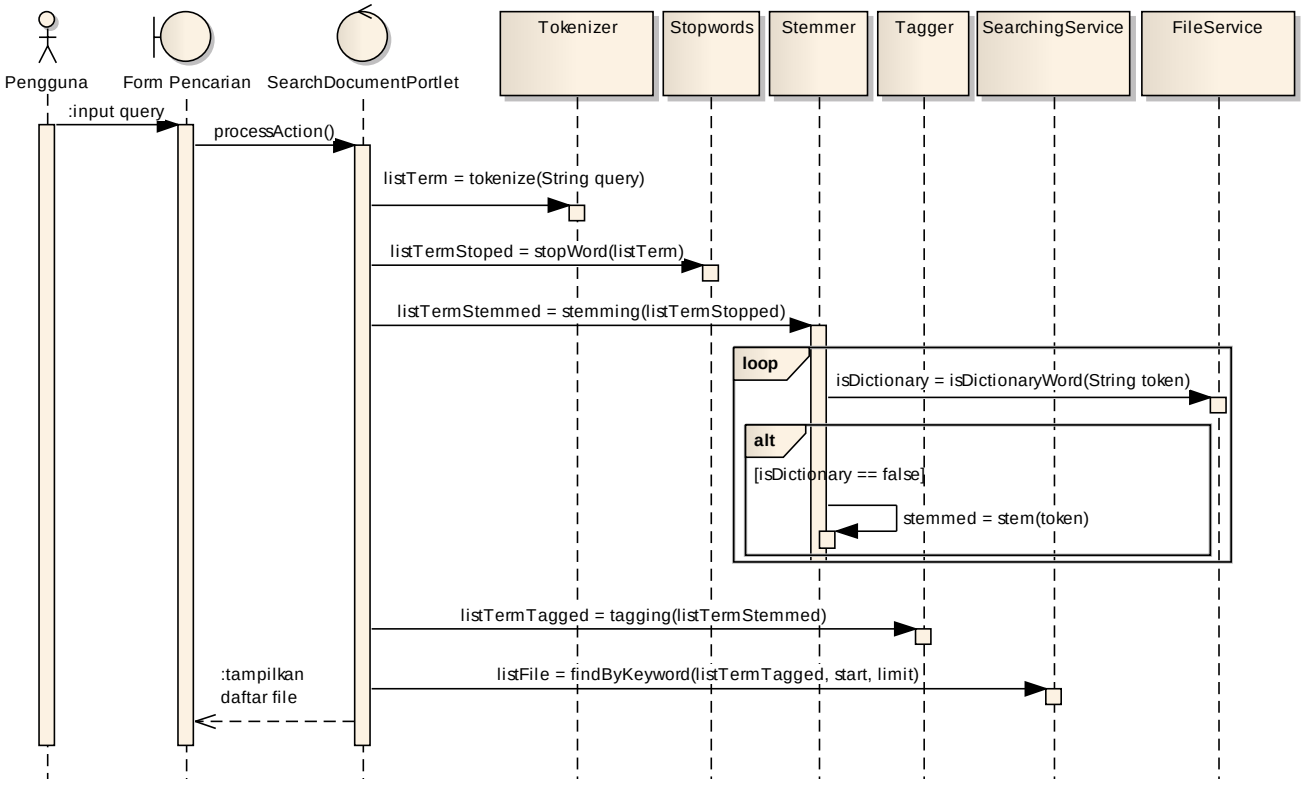
Gambar 3.5 *Sequence diagram upload* dokumen

Sequence diagram dalam Gambar 3.5 memberikan gambaran pada saat terjadinya *use case upload* dokumen. Pada awalnya pengguna memasukkan judul, deskripsi singkat, dan *file-file* untuk disimpan ke dalam sistem. Kemudian kelas DocumentPortlet bertugas untuk mengatur urutan proses-proses yang harus dilakukan mulai dari menyimpan data dokumen ke dalam *database*, memanggil kelas-kelas Extractor, Tokenizer, Stopwords, Stemmer, Tagger untuk mengolah teks, dan yang terakhir menyimpan data posting ke dalam *database*.

3.4.1.2 Sequence diagram pencarian *file*

Sequence diagram dalam Gambar 3.6 dibawah ini memberikan gambaran pada saat terjadinya *use case* pencarian *file*. Pada awalnya pengguna memasukkan kata kunci pencarian di form pencarian. Lalu sistem akan mengolah *query* sehingga dihasilkan kata dasar dari masing-masing term. Mengirim parameter kata kunci pencarian dan akan memproses *query* ke *database*.

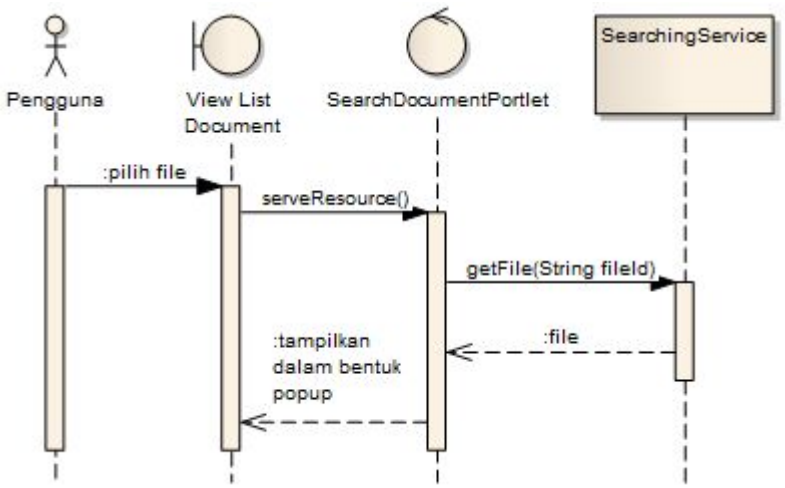
Setelah *database* menghasilkan *query*, lalu proses selanjutnya adalah dengan memproses hasil query tersebut melalui metode tf-idf. Hasil dari algoritma ini berupa *file* yang telah di-*ranking*.



Gambar 3.6 *Sequence diagram* pencarian *file*

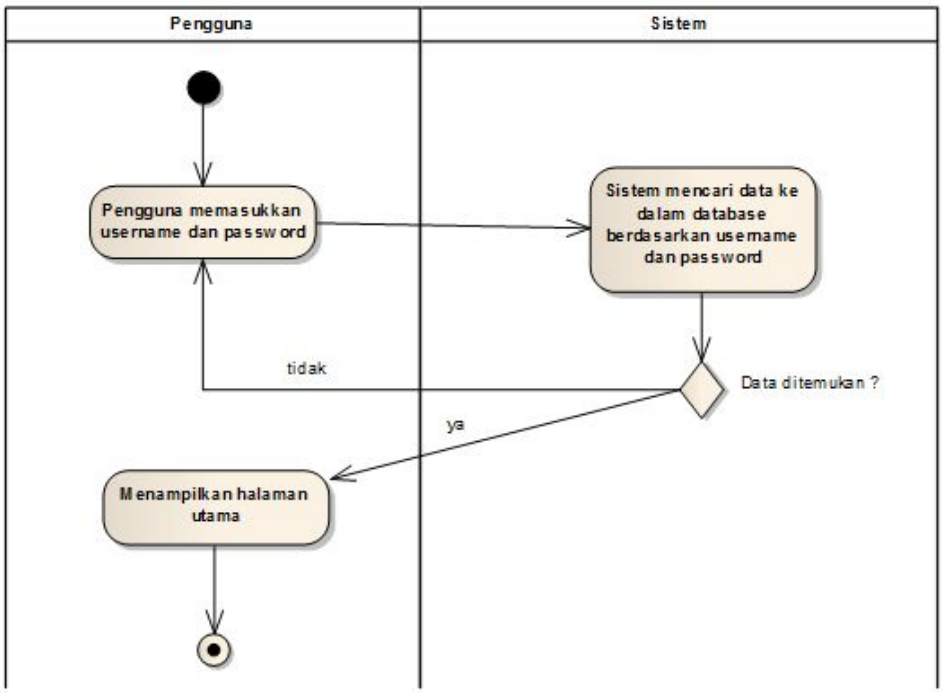
3.4.1.3 Sequence diagram download file

Sequence diagram dalam Gambar 3.7 dibawah ini memberikan gambaran pada saat terjadinya *use case download file*. Pada awalnya terdapat daftar *file* hasil dari proses pencarian. Kemudian pengguna memilih *file* mana yang akan di *download*. Sistem akan menampilkan popup untuk proses *download file*.



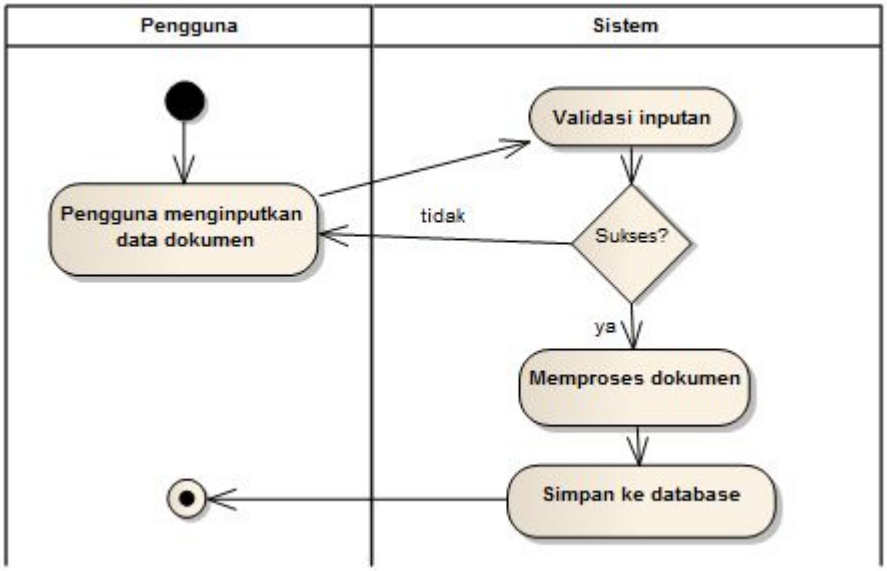
Gambar 3.7 ***Sequence diagram download file***

3.4.2 Diagram Aktivitas



Gambar 3.8 Diagram aktivitas login

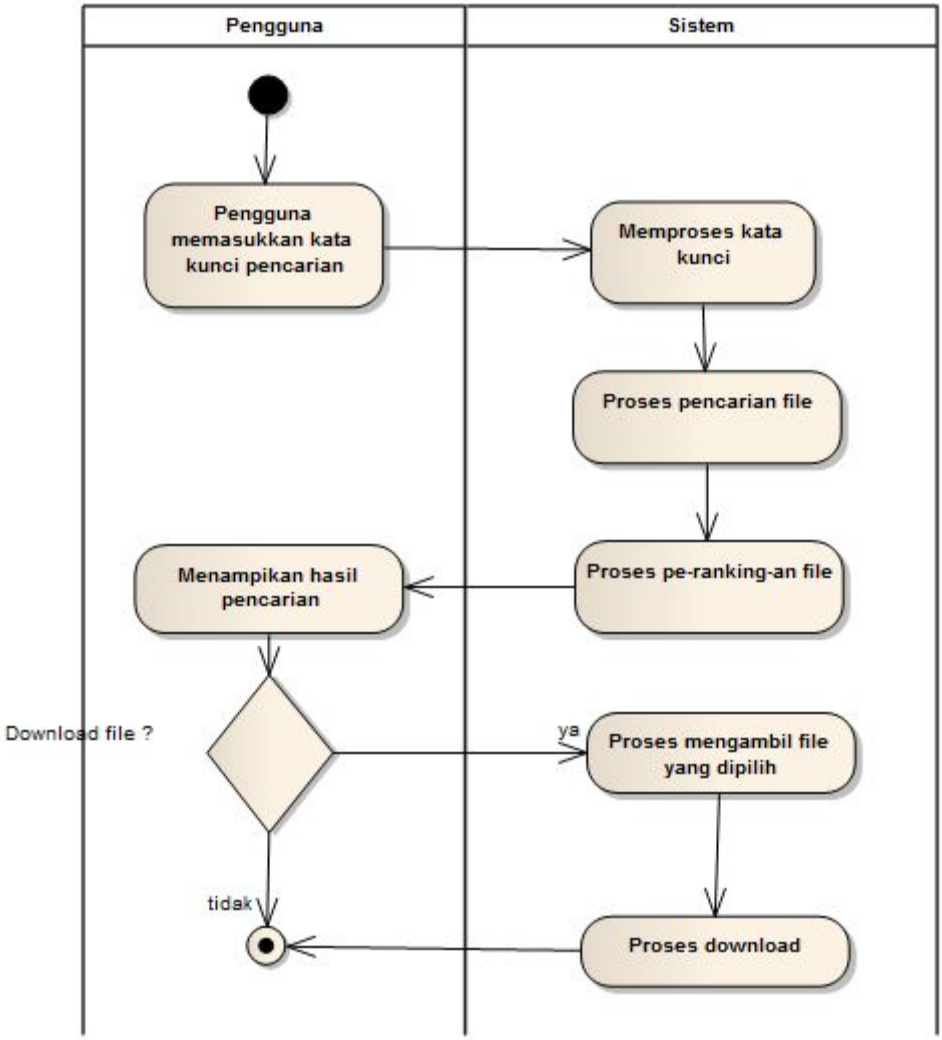
Gambar 3.8 menjelaskan mengenai diagram aktivitas login, dimana aktivitas awalnya adalah pengguna memasukkan *username* dan *password*. Kemudian sistem memproses dengan melakukan pencarian data user ke database berdasarkan *username* dan *password.* Jika data ditemukan maka pengguna diarahkan ke halaman utama. Jika tidak maka kembali ke halaman login dan menampilkan pesan.



Gambar 3.9 Diagram aktivitas *upload file*

Gambar 3.9 menjelaskan mengenai diagram aktivitas *upload* dokumen, dimana aktivitas awalnya adalah pengguna memasukkan data dokumen beserta *file-file* yang akan diunggah. Kemudian sistem memproses data tersebut dengan melakukan pengecekan validasi, jika validasi tidak lolos maka pengguna wajib melengkapi data yang kurang. Jika validasi lolos, maka sistem memproses dokumen tersebut dengan menyimpan data dokumen dan *file* ke basis data. Jika file berekstensi *.txt atau *.doc maka dilakukan proses selanjutnya yaitu

penentuan indeks, kemudian hasil dari penentuan indeks tersebut disimpan ke dalam basis data.



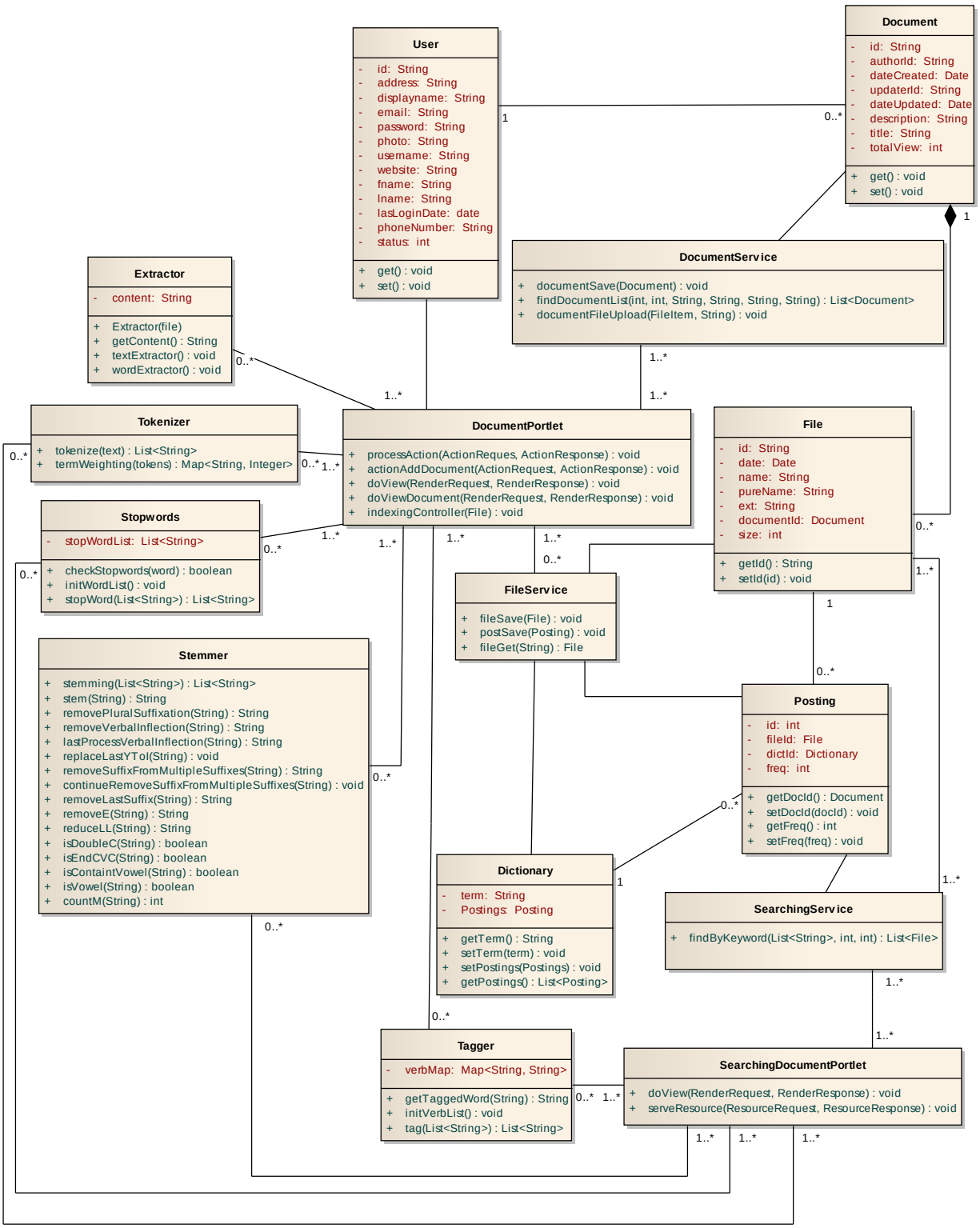
Gambar 3.10 Diagram aktivitas pencarian

Gambar 3.10 menjelaskan mengenai diagram aktivitas pencarian *file*, dimana aktivitas awalnya adalah pengguna memasukkan kata kunci (*query*) pencarian. Kemudian sistem memproses kata kunci tersebut untuk dilakukan tokenisasi, *stoplist, stemmer*. Setelah dibentuk kata-kata dasar dari kata kunci, maka dilakukan proses pencarian *file* ke dalam basis data. Jika ditemukan hasil

dari pencarian tersebut, maka sistem melakukan proses perangkingan terhadap *file-file* hasil pencarian. Selanjutnya *file* hasil yang telah di-*ranking* ditampilkan kepada pengguna. Jika pengguna ingin mengunduh *file*, maka sistem mengambil *file* fisik yang tersimpan untuk diberikan kepada pengguna.

3.4.3 Diagram Kelas

Diagram Kelas *(Class diagram)* menggambarkan kelas-kelas yang menjadi inti dari program yang akan dibuat. Kelas adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

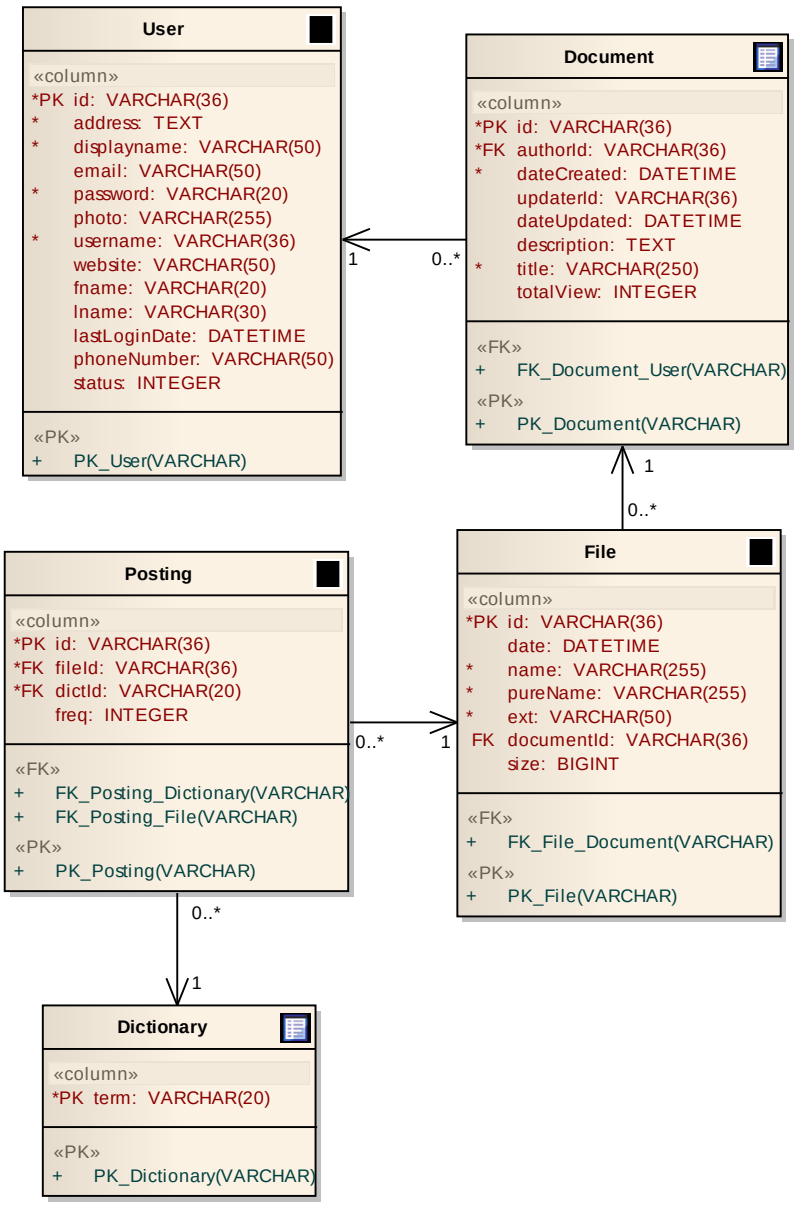


Gambar 3.11 Class diagram aplikasi pencarian dokumen

Diagram kelas dalam Gambar 3.11 diatas ini memberikan gambaran keterkaitan antara kelas-kelas pada aplikasi pencarian dokumen. Seperti terlihat pada bagian 3.4.1.1 dan 3.4.1.2, bahwa adanya keterhubungan kelas-kelas antara lain : DocumentPortlet, DocumentService, FileService, Extractor, Tokenizer, Stemmer, Tagger, Stopwords, Dictionary, Posting, Document, File, SearchingPortlet, dan SearchingService

3.4.4 Skema Relasi

Dari diagram kelas pada Gambar 3.11 bisa ditentukan skema relasi database yang akan dibuat. Bisa terlihat pada Gambar 3.12 bahwa ada 5 tabel yang nantinya akan digunakan pada aplikasi pencarian dokumen ini, antara lain : user, document, file, posting, dan dictionary.



Gambar 3.12 Skema relasi subsistem pencarian dokumen

3.4.5 Struktur File

Sistem aplikasi membutuhkan spesifikasi file yang dimaksudkan untuk memudahkan sistem kerja komputer dalam melakukan pengaturan dan pencarian data. Struktur file digunakan dalam perancangan sistem untuk menentukan struktur fisik *database* dengan menjelaskan rincian dari setiap file (nama file,

kunci utama, jumlah atribut, nama atribut, ukuran atribut, dan keterangan atribut). Adapun rincian struktur file yang digunakan sistem pencarian dokumen dapat dilihat dalam tabel 3.5 - 3.9 dibawah ini :

Tabel 3.5 Struktur File Document

Nama File : document Kunci Utama : document_id Jumlah Atribut : 8				
No	**Nama Atribut**	**Tipe Data**	**Ukuran**	**Keterangan**
1	document_id	VARCHAR	36	-
2	document_author_id	VARCHAR	36	Id pembuat dokumen
3	document_last_updater_id	VARCHAR	36	Isi pengguna yang mengubah dokumen.
4	document_datecreated	TIMESTAMP		Tanggal dokumen dibuat
5	document_total_view	INT		Jumlah dokumen dilihat
6	document_title	VARCHAR	250	Judul dokumen
7	document_description	VARCHAR	1000	Deskripsi dokumen
8	document_dateupdated	TIMESTAMP		Tanggal dokumen diubah

Tabel 3.5 merupakan struktur file untuk menyimpan data dokumen. Tabel dokumen ini dipetakan dalam kelas Document pada diagram kelas Gambar 3.11.

Tabel 3.6 Struktur File *File*

Nama File Kunci Utama Jumlah Atribut	: file : file_id : 7			
No	**Nama Atribut**	**Tipe Data**	**Ukuran**	**Keterangan**
1	file_id	VARCHAR	36	-
2	file_name	VARCHAR	500	Nama *file* fisik
3	pure_name	VARCHAR	500	Nama asli file yang di-*upload*.
4	file_date	TIMESTAMP		Tanggal file di-*upload*
5	file_ext	VARCHAR	5	Extention dari *file*
6	file_size	LONG		Ukuran *file*
7	document_id	VARCHAR	36	

Tabel 3.6 merupakan struktur file untuk menyimpan data *file*. Tabel file ini dipetakan dalam kelas File pada diagram kelas Gambar 3.11.

Tabel 3.7 Struktur File Dictionary

Nama File : dictionary Kunci Utama : dictId Jumlah Atribut : 1				
No	**Nama Atribut**	**Tipe Data**	**Ukuran**	**Keterangan**
1	dictId	VARCHAR	20	-

Tabel 3.7 merupakan struktur file untuk menyimpan data *term* yang terdapat pada koleksi *file* . Tabel dictionary ini dipetakan dalam kelas Dictionary pada diagram kelas Gambar 3.11.

Tabel 3.8 Struktur File Posting

Nama File : posting Kunci Utama : posting_id Jumlah Atribut : 4				
No	**Nama Atribut**	**Tipe Data**	**Ukuran**	**Keterangan**
1	posting_id	VARCHAR	36	-
2	file_id	VARCHAR	36	Id file
3	dict_id	VARCHAR	20	Id dictionary
4	frequency	INTEGER		Frekuensi

Tabel 3.8 merupakan struktur file untuk menyimpan data *inverted index* dari setiap *file* yang terdapat didalam koleksi *file*, dimana *inverted index* yang direkam belum terurut secara alphabetis. Tabel posting ini dipetakan dalam kelas Posting pada diagram kelas Gambar 3.11.

Tabel 3.9 Struktur File User

Nama File : user Kunci Utama : user_id Jumlah Atribut : 13				
No	**Nama Atribut**	**Tipe Data**	**Ukuran**	**Keterangan**
1	user_id	VARCHAR	36	-
2	user_address	VARCHAR	2000	Alamat user
3	user_displayname	VARCHAR	50	Nama yang ditampilkan
4	user_email	VARCHAR	50	Alamat email user
5	user_fname	VARCHAR	20	Nama depan
6	user_lastlogindate	TIMESTAMP		Tanggal terakhir login
7	user_lname	VARCHAR	30	Nama belakang
8	user_password	VARCHAR	20	Password user
9	user_phonenumber	VARCHAR	50	Nomor telepon
10	user_photo	VARCHAR	255	Path foto
11	user_status	INTEGER		Status user
12	user_username	VARCHAR	36	Username user
13	user_website	VARCHAR	50	Alamat website user

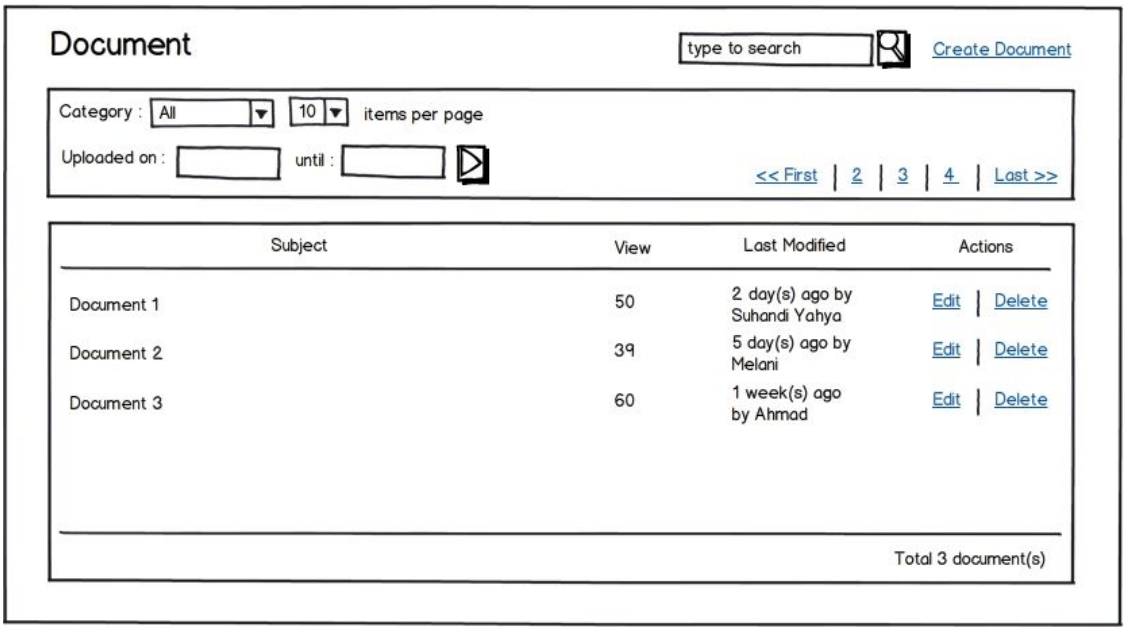
Tabel 3.9 merupakan struktur file untuk menyimpan data user. Tabel user ini dipetakan dalam kelas User pada diagram kelas Gambar 3.11.

3.5 Perancangan Antarmuka

Perancangan antarmuka untuk perangkat lunak yang akan dibangun dijelaskan pada tabel 3.10 – 3.13. Pada tabel 3.10 menjelaskan komponen-komponen yang ada pada halaman daftar dokumen. Dimana pada halaman ini, pengguna bisa melihat daftar dari dokumen yang ada dan melakukan pencarian terhadap dokumen berdasarkan judul, kategori, dan tanggal *upload* dokumen. Pada tabel 3.11 menjelaskan komponen-komponen yang ada pada halaman *upload* dokumen. Dimana pada halaman ini, pengguna dapat mengisi *form* untuk menambah dokumen. Pada tabel 3.12 menjelaskan komponen-komponen yang ada pada halaman detail dokumen. Dimana pada halaman ini, pengguna dapat melihat data detail dari dokumen, yaitu : judul, deskripsi, daftar *file* yang dimiliki. Pada halaman ini juga pengguna bisa mengunduh file. Pada tabel 3.13 menjelaskan komponen-komponen yang ada pada halaman pencarian file. Dimana pada halaman ini, pengguna dapat memasukkan kata kunci pencarian dan memproses pencarian supaya mendapatkan daftar *file* hasil pencarian yang telah di-*ranking*.

Tabel 3.10 Perancangan antarmuka halaman daftar dokumen

TAMPILAN ID	T1
NAMA TAMPILAN	Halaman Daftar Dokumen
KETERANGAN	Untuk menampilkan daftar dokumen

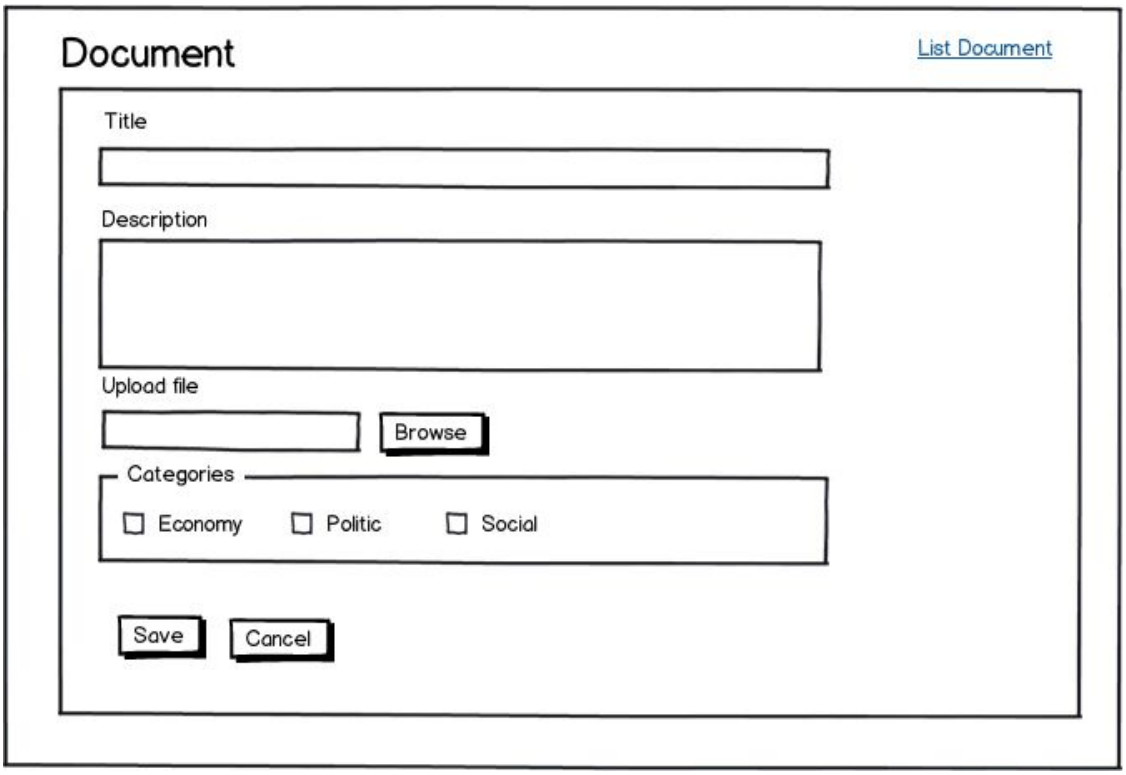


DETAIL

Label	Tipe	Keterangan
Items per page	COMBO BOX	Komponen form untuk memilih jumlah dokumen per halaman
Category	COMBO BOX	Komponen form untuk memilih kategori
Uploaded on	TEXTBOX	Komponen form untuk memasukkan tanggal awal
Until	TEXTBOX	Komponen form untuk memasukkan tanggal akhir
Type to search	TEXTBOX	Komponen form untuk memasukkan kata kunci pencarian
Create Document	LINK	Berfungsi untuk menampilkan halaman *upload* dokumen
Edit	LINK	Berfungsi untuk menampilkan halaman pengubahan dokumen
Delete	LINK	Berfungsi untuk menampilkan menghapus dokumen
<<First	LINK	Berfungsi untuk menampilkan dokumen pada level pertama
Last>>	LINK	Berfungsi untuk menampilkan dokumen pada level terakhir

Tabel 3.11 Perancangan antarmuka *upload* dokumen

TAMPILAN ID	T2
NAMA TAMPILAN	Halaman *Upload* Dokumen
KETERANGAN	Untuk menambah dokumen

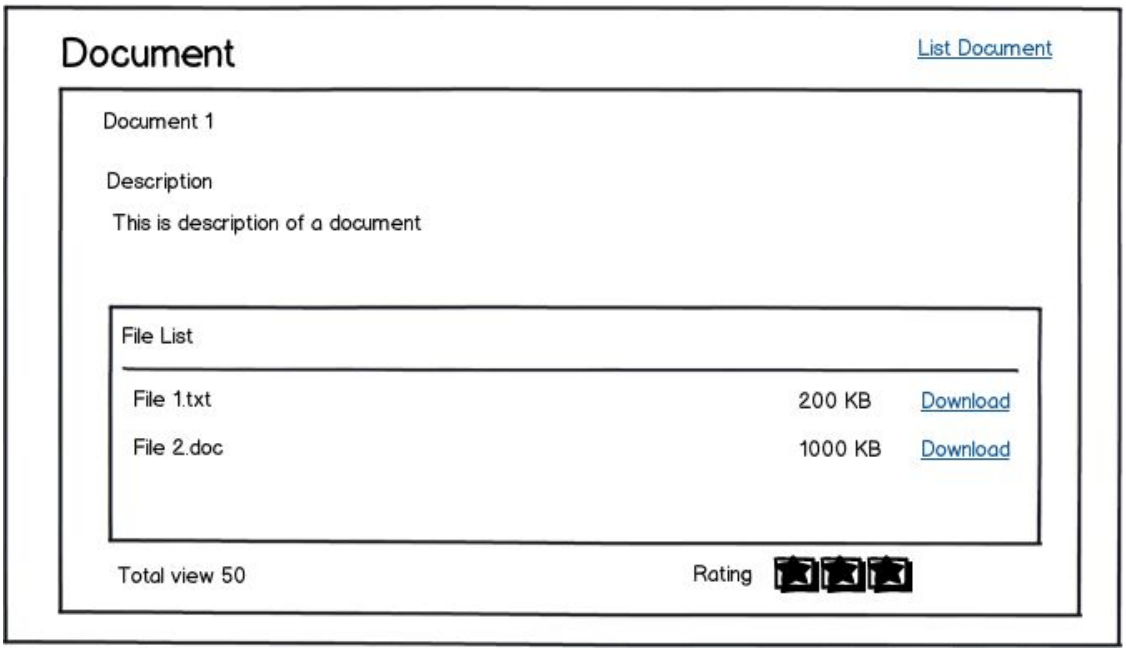


DETAIL

Label	Tipe	Keterangan
Title	TEXTBOX	Komponen form untuk memasukkan judul
Description	TEXTAREA	Komponen form untuk memasukkan keterangan
Upload file	TEXTBOX	Komponen form untuk memasukkan *file*
Categories	CHECKBOX	Komponen form untuk memilih kategori
List Document	LINK	Berfungsi untuk menampilkan halaman daftar dokumen
Save	BUTTON	Berfungsi untuk menyimpan dokumen
Cancel	BUTTON	Berfungsi untuk membatalkan proses dan menampilkan halaman daftar dokumen

Tabel 3.12 Perancangan antarmuka halaman detail dokumen

TAMPILAN ID	T3
NAMA TAMPILAN	Halaman Detail Dokumen
KETERANGAN	Untuk menampilkan detail dokumen

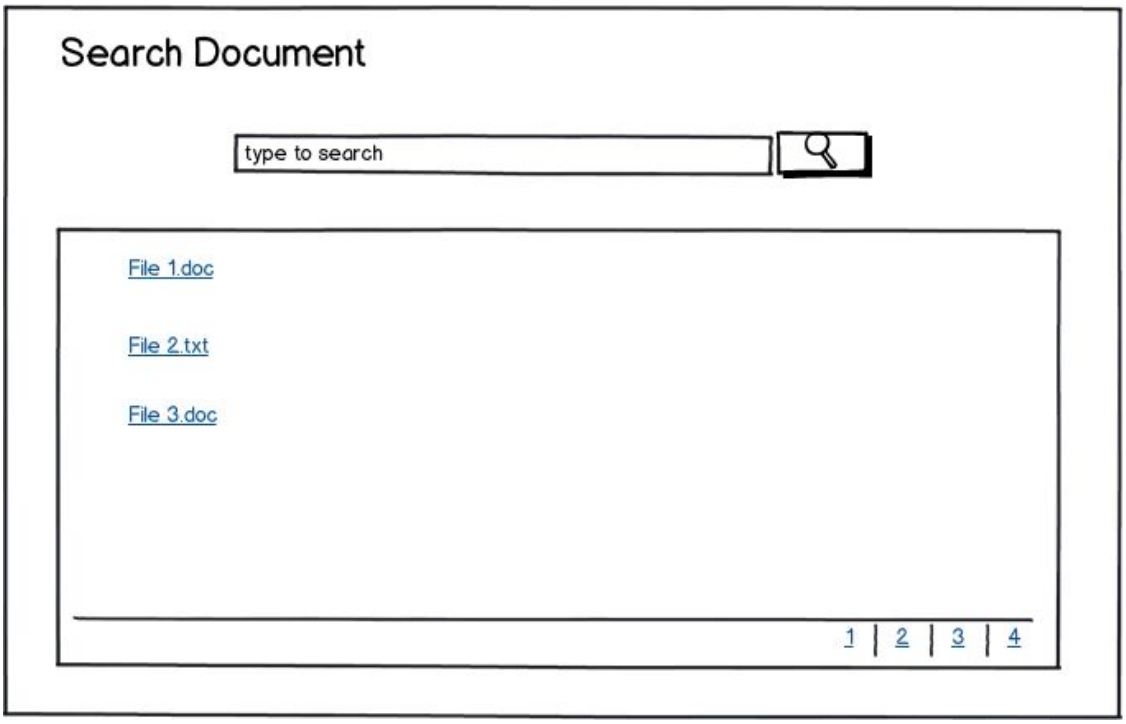


DETAIL

Label	Tipe	Keterangan
List Document	LINK	Berfungsi untuk menampilkan halaman daftar dokumen
Download	LINK	Berfungsi untuk mengunduh file
Rating	LINK	Berfungsi untuk memberikan penilaian terhadap dokumen

Tabel 3.13 Perancangan antarmuka halaman pencarian dokumen

TAMPILAN ID	T4
NAMA TAMPILAN	Halaman Pencarian Dokumen
KETERANGAN	Untuk melakukan pencarian dokumen



DETAIL

Label	Tipe	Keterangan
Type to search	TEXTBOX	Komponen form untuk memasukkan kata kunci pencarian
	BUTTON	Berfungsi untuk memproses pencarian
File 1.doc	LINK	Berfungsi untuk mengunduh file