

BAB III

ANALISIS DAN PERANCANGAN

3.1 Variabel Penelitian

Penelitian yang berjudul “Implementasi Sistem Temu Kembali Informasi Pada E-Library Universitas Udayana Menggunakan Metode Vector Space Model Dan Query Expansion” mempunyai beberapa variabel penelitian. Variabel yang digunakan adalah tingkat keberhasilan temu informasi yang dapat diukur dengan nilai *precision*, *recall*, dan *f-measure*. Semakin relevan dokumen yang dihasilkan dari kata kunci (*query*) yang diinputkan, maka tingkat keberhasilan suatu sistem temu kembali informasi dapat dikatakan akan semakin tinggi. Dalam penelitian ini akan dibandingkan metode vector space model dengan metode vector space model dengan menambahkan penerapan query expansion untuk mengetahui tingkat keberhasilan dari kedua metode yang diterapkan. Hal yang diperhatikan apakah dengan penambahan query expansion dapat meningkatkan nilai *precision*, *recall*, dan *f-measure* dari suatu sistem temu kembali informasi.

Data yang digunakan dalam penelitian ini adalah data skripsi mahasiswa Universitas Udayana yang telah diupload dalam waktu enam bulan terakhir. Skripsi mahasiswa ini berasal dari Fakultas Ekonomi, Pertanian, dan Matematika dan Ilmu Pengetahuan Alam. Sedangkan untuk query expansion digunakan kamus kata tesaurus Bahasa Indonesia. Kamus tesaurus tersebut diterbitkan oleh Pusat Bahasa Departemen Pendidikan Nasional.

3.2 Analisis Kebutuhan

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kebutuhan Fungsional Sistem

No	Kebutuhan Fungsional	Target Pengguna
1.	Sistem menyediakan fitur <i>login</i> dan <i>logout</i> untuk pengguna	Admin, User
2.	Sistem dapat melakukan manajemen data user, dengan rincian sebagai berikut. 1. Menambah user baru 2. Mengedit data user 3. Menghapus data user 4. Melihat semua data user	Admin
3.	Sistem dapat melakukan manajemen dokumen user, dengan rincian sebagai berikut. 1. Melihat semua dokumen yang diupload user 2. Menghapus dokumen user	Admin
4.	Sistem dapat melakukan proses preprocessing dan indexing dokumen	Admin
5.	Sistem menyediakan fitur upload dokumen PDF	User
6.	Sistem dapat melakukan proses pencarian dokumen atau temu kembali informasi	User

3.2.2. Kebutuhan Non Fungsional

No	Kebutuhan Fungsional	Deskripsi Kebutuhan
1.	Reability	Sistem mampu menyediakan dan memberikan informasi yang sesuai, akurat dan terpercaya
2.	User-friendly	Sistem didukung dengan tampilan yang menarik dan mudah digunakan

3.3 Analisis Sistem

3.3.1 Penentuan Input-Input Sistem

Input-input sistem merupakan memasukkan apa saja yang nantinya akan diproses oleh sistem. Pada sistem ini, input yang digunakan adalah berupa file dokumen skripsi dari mahasiswa Universitas Udayana. Dokumen tersebut merupakan dokumen yang telah diupload pada sistem yang asli dalam waktu enam bulan terakhir. Dokumen skripsi yang digunakan berasal dari Fakultas Ekonomi, Fakultas Pertanian, dan Fakultas Matematika dan Ilmu Pengetahuan Alam. Semua dokumen tersebut masih dalam format PDF.

3.3.2 Penentuan Proses-Proses dalam Operasi Sistem

Dilihat dari sisi pegguaan, sebuah sistem temu kembali informasi biasanya melibatkan dua proses. Proses pertama memasukkan koleksi dokumen ke dalam sistem yang biasanya dilakukan dari sisi administrator sedangkan yang kedua adalah dari sisi pengguna yaitu memasukkan *query* yang digunakan untuk mendapatkan dokumen yang diinginkan.

Dalam sistem temu kembali informasi yang dibuat, proses-proses yang dilakukan adalah, sebagai berikut.

- a. Proses upload dokumen dengan format PDF ke dalam sistem
- b. Proses konversi dokumen PDF menjadi string.
- c. Proses *tokenizing* yaitu menghilangkan angka, tanda baca, dan karakter asing lainnya sehingga hanya menghasilkan string dengan karakter alfabet.
- d. Proses *filtering* atau *stopword removal* yaitu menghilangkan kata-kata yang dianggap tidak terlalu merepresentasikan isi dokumen. Pada proses *filtering*, *stopword* yang dihilangkan adalah *stopword* dalam Bahasa Indonesia dan Bahasa Inggris.
- e. Proses *indexing*, yaitu membangun database indeks dokumen dengan data yang telah dilakukan *preprocessing*. Pada proses *indexing*, kata pada dokumen sebelumnya akan dipecah per kata, kemudian akan dimasukkan ke dalam database. Hal yang membedakan satu kata dengan yang lainnya ditandai dengan jarak satu spasi.

- f. Proses *term weighting*, yaitu pemberian bobot pada kata dalam setiap dokumen. Metode pembobotan yang digunakan adalah *term frequency – inverse document frequency*. Proses *term weighting* akan menghasilkan matriks ruang vektor (*vector space model*) sebagai kata-dokumen matriks (Mandala & Setiawan, 2002).
- g. *Query formulation*, yaitu pemberian bobot pada query yang dimasukkan oleh pengguna.
- h. *Query expansion*, yaitu pemberian bobot pada query hasil perluasan (*expansion*) pada query awal. Pada proses ini setiap kata query awal akan dicari sinonimnya pada database Kamus Tesaaurus Bahasa Indonesia. Query hasil perluasan query awal mempunyai bobot setengah dari query awal.
- i. Proses menghitung kesamaan antara query dengan koleksi dokumen. Pada proses ini teknik yang digunakan adalah dengan *cosine similarity*. Cosine similarity akan menghasilkan nilai yang merepresentasikan tingkat kesamaan antara query dengan suatu dokumen.
- j. Proses perankingan hasil dokumen. Setelah menghasilkan nilai kesamaan, maka dilakukan perankingan dokumen. Dokumen dengan nilai kesamaan tertinggi akan ditampilkan pada posisi teratas.

3.3.3 Penentuan Output dari Proses Input Proses

Output yang dihasilkan dari sistem adalah berupa kumpulan dokumen relevan sesuai dengan query atau kata kunci yang dimasukkan oleh pengguna. Selain dokumen akan ditampilkan pula tingkat kemiripan atau *similarity* antara query dengan masing-masing dokumen.

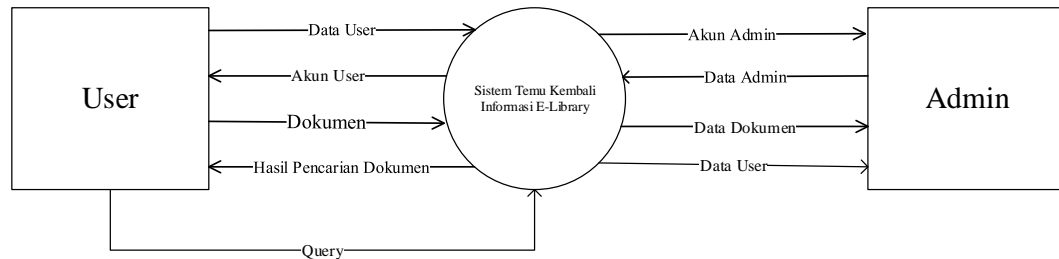
3.4. Perancangan Sistem

Perancangan sistem meliputi perancangan data flow diagram, entity relationship diagram, flowchart, dan rancangan antarmuka.

3.4.1 Data Flow Diagram

a. Context Diagram

Context diagram merupakan diagram yang terdiri dari suatu proses yang menggambarkan sistem secara keseluruhan.



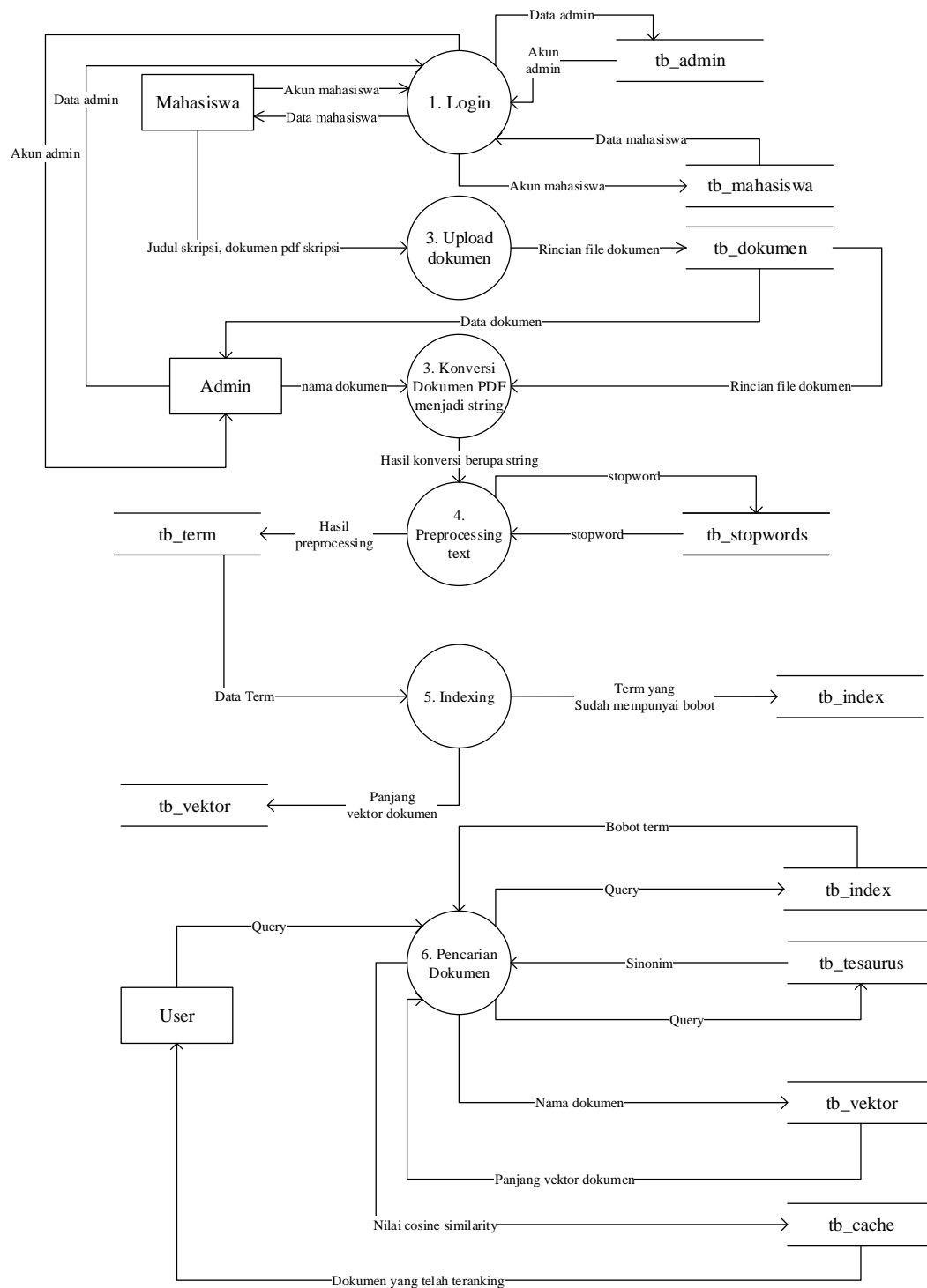
Gambar 3.1 Context Diagram

Gambar 3.1 merupakan context diagram dari sistem temu kembali informasi e-library. Context diagram menggambarkan keseluruhan input dan output dari sebuah sistem dari semua entitas yang berperan. Dalam diagram tersebut terdapat dua entitas yang akan menggunakan sistem yaitu user dan admin. User menginputkan data user, dokumen serta query untuk pencarian dokumen. Sedangkan yang dihasilkan pada output adalah akun user dan hasil pencarian dokumen. Pada sisi admin, yang diinputkan adalah data admin sehingga akan menghasilkan akun admin. Selain itu, dari sisi output, admin juga dapat melihat semua dokumen yang diupload oleh user.

b. DFD Level 0 Sistem Temu Kembali Informasi

Gambar 3.2 adalah DFD Level 0 dari Sistem Temu Kembali Informasi E-Library. Diagram ini menggambarkan keseluruhan proses yang dilakukan oleh sistem.

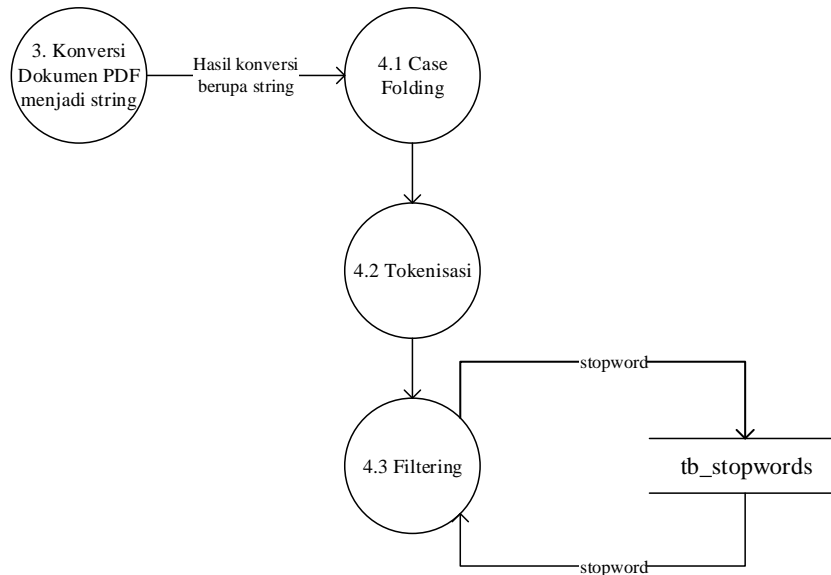
Secara umum terdapat 7 buah proses utama dari sistem yang dibuat yaitu: 1) proses login untuk admin dan user; 2) Proses upload dokumen; 3) Proses konversi dokumen pdf menjadi text biasa; 4) *Preprocessing* teks; 5) Proses *indexing*; 6) Proses pencarian dokumen. Beberapa proses penting akan dijelaskan secara lebih rinci pada DFD Level 1.



Gambar 3.2 Diagram Level 0 Sistem Temu Kembali Informasi

c. DFD Level 1 Preprocessing Text

Dari DFD level 0 pada proses 4 yaitu *preprocessing text*, dapat dirinci seperti dilihat pada Gambar 3.3.

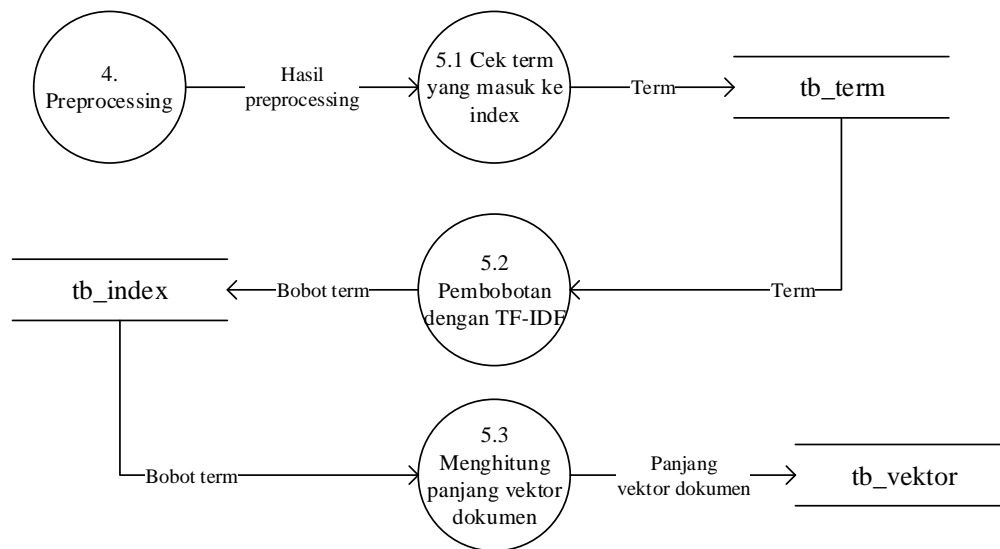


Gambar 3.3 Diagram Level 1 Preprocessing Text

Pada *preprocessing* teks, terdapat 3 proses yaitu *case folding*, tokenisasi, dan *filtering*. Proses diawali dari mengambil hasil konversi dokumen pdf menjadi teks atau *string*. Selanjutnya akan dilakukan *case folding* untuk menghapus karakter asing yang tidak diperlukan agar suatu teks hanya menyisakan huruf latin saja. Kemudian dilanjutkan dengan proses tokenisasi yang bertujuan untuk memecah-mecah *string* menjadi kata atau term tunggal. Proses diakhiri dengan *filtering* untuk menyaring *term* dari *stopword* atau kata yang tidak mempunyai pengaruh signifikan pada pencarian dokumen. Pada *filtering* daftar *stopword* diambil dari tabel *tb_stopwords*.

d. DFD Level 1 Indexing

Setelah preprocessing, dilanjutkan dengan proses indexing seperti yang ditunjukkan pada Gambar 3.4.



Gambar 3.4 Diagram Level 1 Indexing

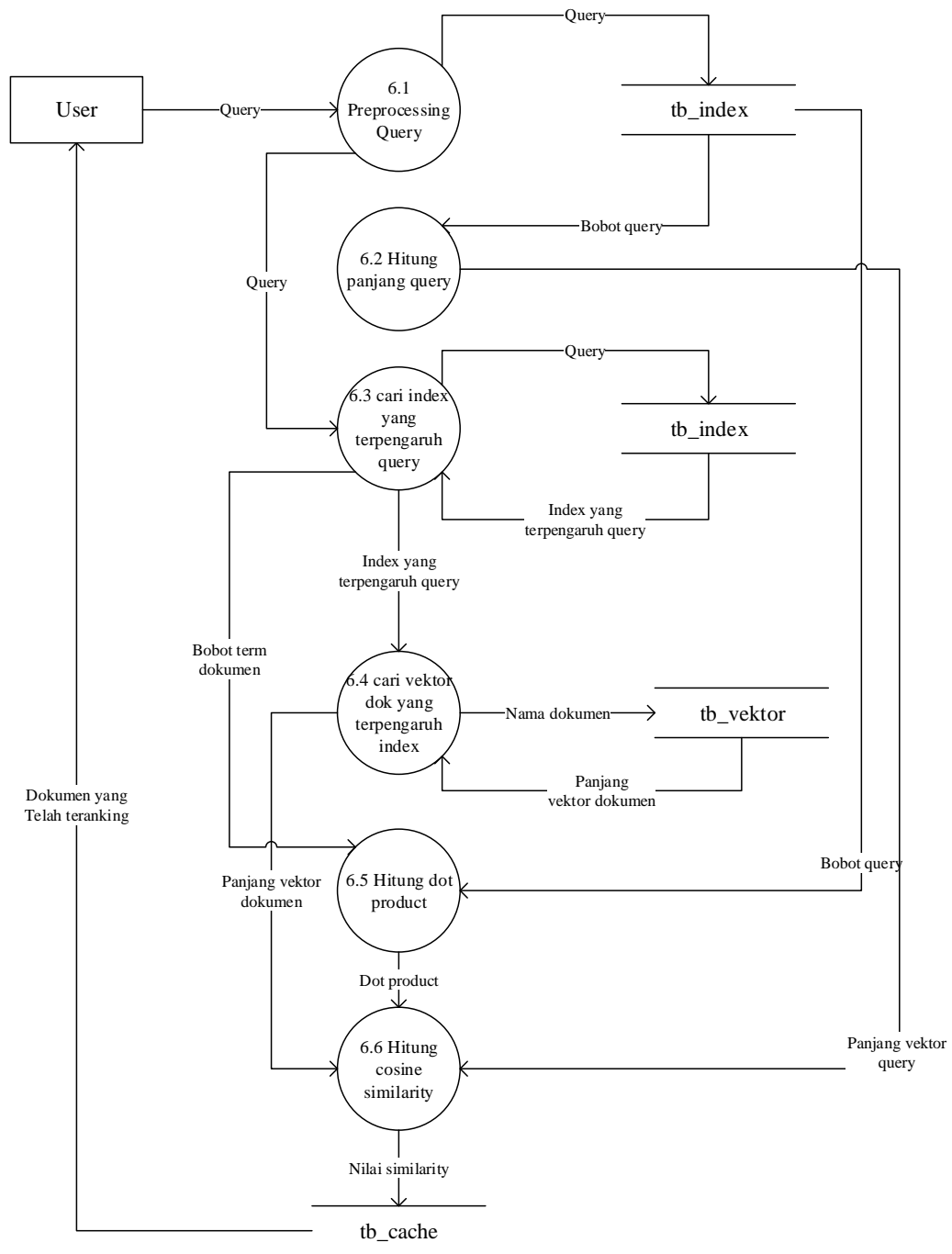
Pada proses *indexing* terdapat 3 proses yaitu proses mengecek term yang dapat masuk ke dalam tabel `tb_tabel`, proses pembobotan, proses menghitung panjang vektor dokumen. Proses pengecekan term berfungsi untuk mengecek term-term yang akan masuk ke `tb_term`. Hal ini untuk memastikan array kosong dan term yang kurang atau mempunyai dua huruf tidak dimasukkan ke dalam tabel `tb_term`. Pembobotan term dengan metode TF-IDF adalah proses selanjutnya. Setelah semua term diberi bobot, maka term-term akan dimasukkan ke dalam tabel `tb_index`.

Setiap dokumen mempunyai term yang telah diberikan bobot. Langkah selanjutnya adalah menghitung panjang vektor masing-masing dokumen dengan menggunakan bobot dari masing term. Hasil penghitungan panjang vektor kemudian dimasukkan ke dalam tabel `tb_vektor`. Jika ketiga proses di atas sudah dilakukan maka dokumen telah terindeks dan pencarian dokumen dengan *query* telah siap dilakukan.

e. DFD Level 1 Pencarian Dokumen tanpa Query Expansion

Proses 6 pada DFD level 0 dapat dibedakan menjadi dua yaitu, pencarian dokumen tanpa query expansion dan pencarian dokumen dengan query expansion.

Di sini akan dijelaskan pencarian dokumen tanpa query expansion terlebih dahulu. Setiap rincian proses dapat dilihat pada Gambar 3.5.



Gambar 3.5 Diagram Level 1 Pencarian Dokumen tanpa Query Expansion

Proses dimulai saat user memasukkan kata kunci atau *query* pencarian. Query akan melalui proses *preprocessing* terlebih dahulu. Query yang telah dipreprocessing akan digunakan sebagai kata kunci pada tabel *tb_index* untuk mencari term yang sama dengan query yang dimasukkan. Dari sana kemudian diperoleh bobot query. Bobot query selanjutnya digunakan untuk mencari panjang query.

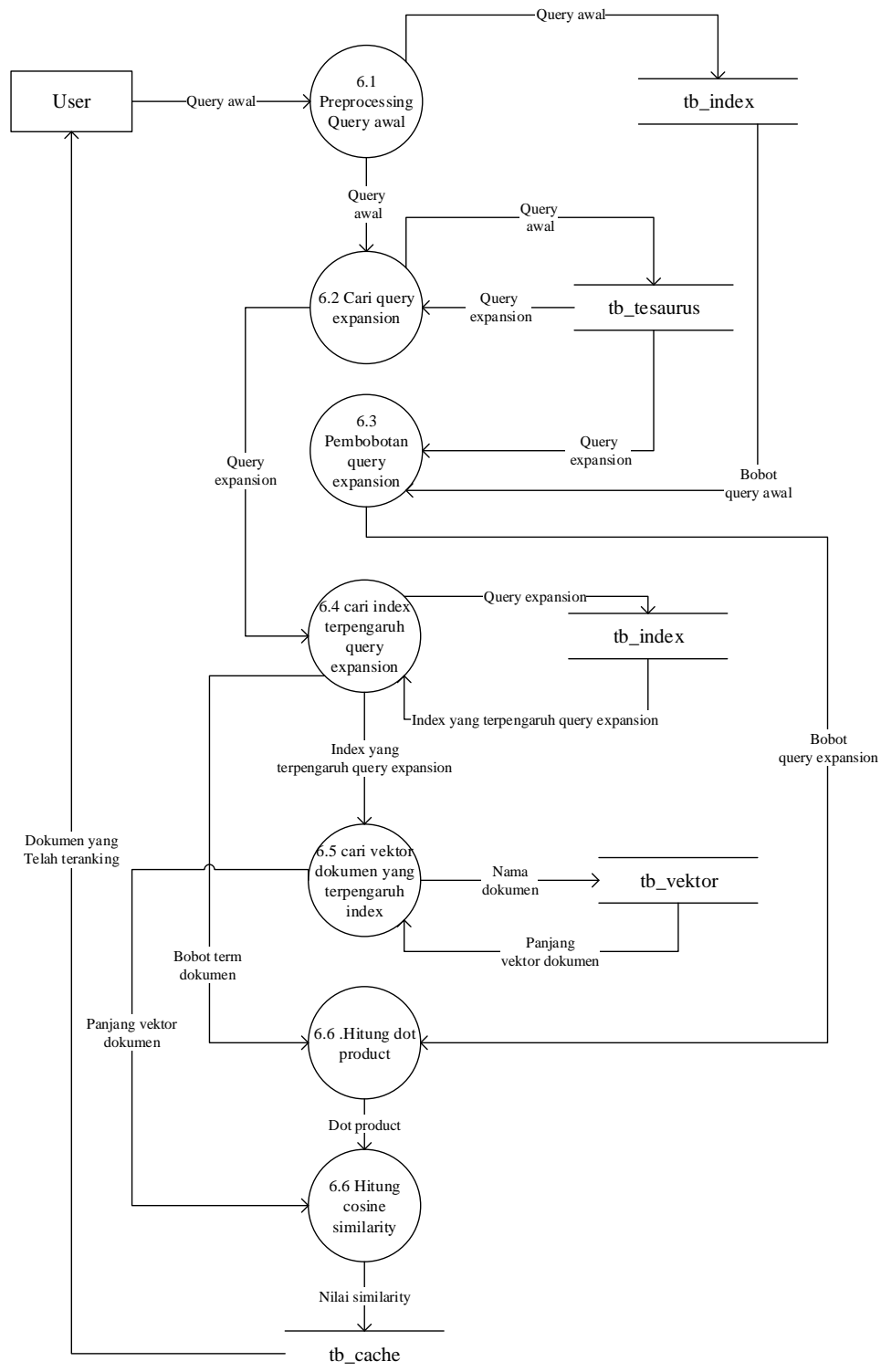
Query yang telah di-*preprocessing* juga digunakan untuk mencari term pada dokumen yang terpengaruh pada tabel *tb_index*. Dari proses tersebut akan didapat daftar term, bobot, dan nama dokumen yang hanya berisi query yang dimasukkan.

Setelah mendapat daftar index yang terpengaruh query, kemudian dilanjutkan dengan menghitung dot product masing-masing dokumen dengan query. Hasil dari dot product kemudian dimanfaatkan untuk mencari nilai similarity antara query dengan masing-masing dokumen. Sebelumnya akan diambil nilai panjang query dan nilai masing-masing panjang vektor dokumen pada tabel *tb_vektor*.

Nilai similarity yang dihasilkan kemudian akan dimasukkan ke tabel *tb_cache*. Nilai similarity berisi tingkat kesamaan dokumen dengan query, dengan nilai maksimal 1. Nama dokumen dan nilai similarity kemudian dimasukkan ke dalam *tb_cache*. Sebelum ditampilkan kepada user, nilai similarity akan diurutkan dari yang tertinggi. Dari user, akan dilihat daftar dokumen hasil pencarian yang telah diurutkan sesuai dengan nilai similarity-nya.

e. DFD Level 1 Pencarian Dokumen dengan Query Expansion

Pada DFD ini akan dijelaskan secara rinci pencarian dokumen dengan menggunakan query expansion. Pencarian dokumen dengan menggunakan query expansion menggunakan bantuan kamus tesaurus Bahasa Indonesia. Gambar 3.6 merupakan DFG yang menggambarkan proses pencarian dokumen dengan query expansion.



Gambar 3.6 Diagram Level 1 Pencarian Dokumen dengan Query Expansion

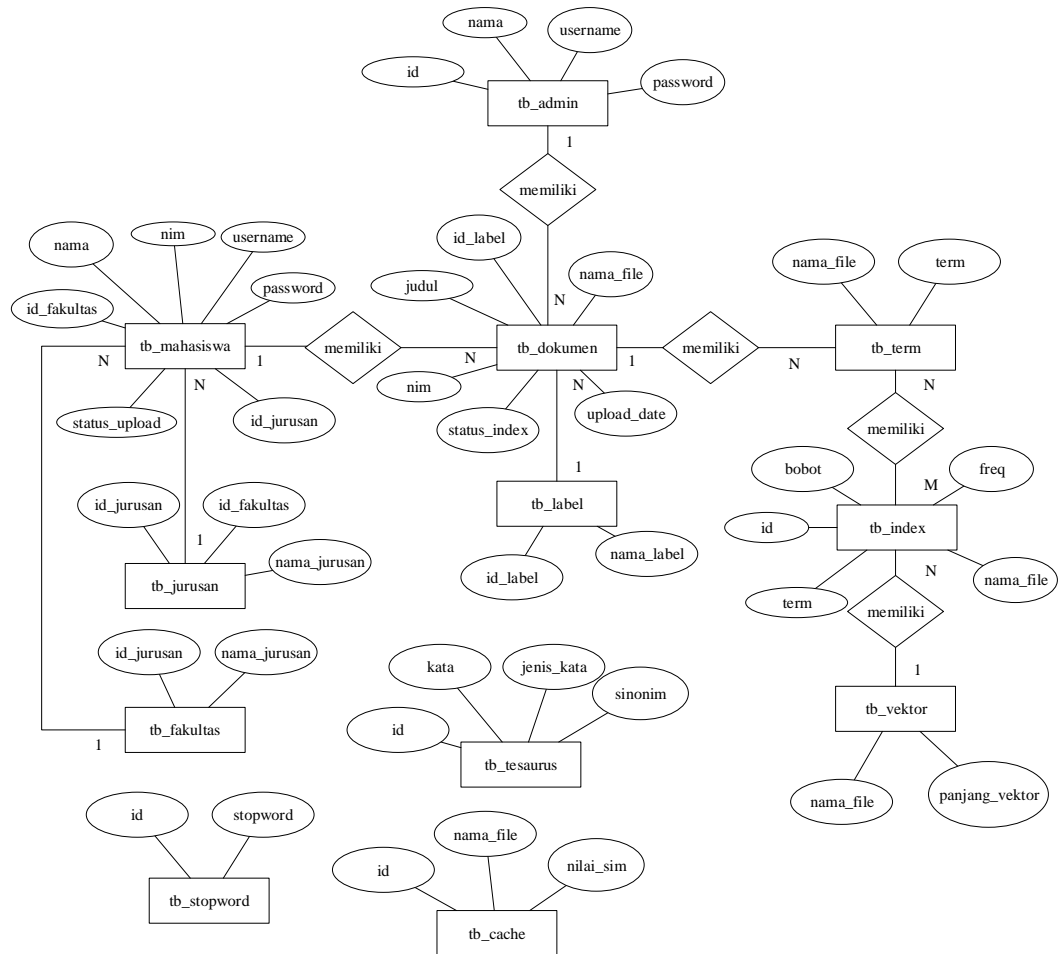
Proses dimulai ketika user memasukkan sebuah query. Kemudian akan dilakukan proses *preprocessing* pada query. Misalkan query yang pertama kali diinputkan pengguna ini disebut dengan query awal. Proses dilanjutkan dengan mencari bobot term query awal pada tabel *tb_index*. Dari sana akan didapatkan bobot query awal.

Kemudian masing-masing kata pada query awal akan digunakan sebagai kata kunci untuk mencari sinonim kata pada kamus tesaurus. Hal ini bertujuan untuk memperluas query awal dengan menggunakan sinonim yang didapatkan. Sinonim kata sebagai perluasan query ini selanjutnya disebut dengan query expansion. Query expansion kemudian diberikan bobot setengah dari bobot query awal. Query expansion yang telah diberikan bobot kemudian dihitung panjang vektornya.

Selanjutnya dilakukan proses mencari term dan nama dokumen yang terpengaruh oleh query expansion pada tabel *tb_index*. Term-term ini yang nantinya akan digunakan untuk menghitung dot product. Nama dokumen yang didapatkan akan digunakan untuk sebagai parameter untuk mencari panjang dokumen yang sesuai pada tabel *tb_vektor*.

Proses dilanjutkan dengan menghitung dot product masing-masing dokumen dengan query expansion. Setelah nilai dot product didapatkan, maka akan dihitung nilai similarity masing-masing dokumen. Nama dokumen dan nilai similarity yang dihasilkan kemudian dimasukkan ke dalam tabel *tb_cache*. Dari sisi user, akan dilihat nilai similarity dokumen yang telah terurut dari yang tertinggi.

3.4.2 Entity Relationship Diagram (ERD)



Gambar 3.7 Entity Relationship Diagram

Sistem ini memiliki 12 entitas, yaitu tb_mahasiswa, tb_admin, tb_dokumen, tb_term, tb_index, tb_vektor, tb_jurusan, tb_fakultas, tb_label, tb_tesaurus, tb_cache, tb_jurusan, dan tb_fakultas. Relasi yang terdapat dalam entitas tersebut adalah sebagai berikut.

- Relasi 1:N antara entitas tb_admin dengan entitas tb_dokumen, di mana seorang admin dapat mempunyai lebih dari satu dokumen.
- Relasi 1:N antara entitas tb_mahasiswa dengan entitas tb_dokumen di mana seorang mahasiswa dapat memiliki lebih dari satu dokumen.

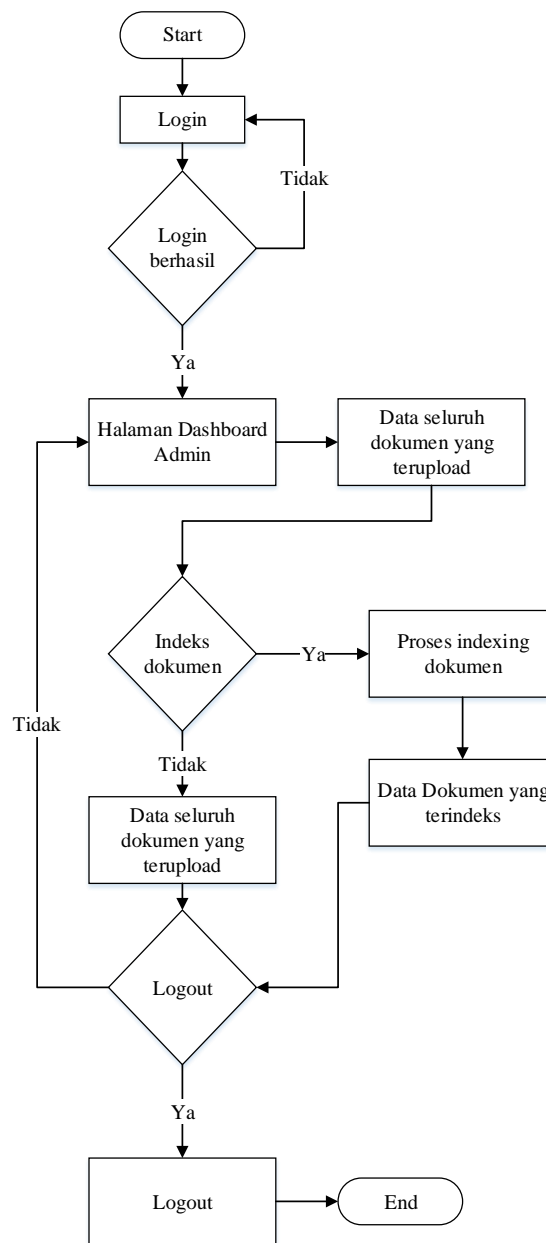
- c. Relasi 1:N antara entitas tb_dokumen dengan tb_term di mana satu dokumen dapat mempunyai banyak term.
- d. Relasi N:M antara tb_term dengan tb_index di mana banyak term dapat ada di lebih dari satu index.
- e. Relasi N:M antara antara tb_index dengan tb_vektor di mana banyak tb_index hanya memiliki satu panjang vektor.
- f. Relasi 1:N antara tb_label dengan tb_dokumen di mana satu label dapat dimiliki banyak dokumen.
- g. Relasi 1:N antara tb_jurusan dengan tb_mahasiswa di mana satu jurusan dapat memiliki banyak mahasiswa.
- h. Relasi 1:N antara tb_fakultas dengan tb_mahasiswa di mana satu fakultas dapat memiliki banyak mahasiswa.
- i. Entitas tb_stopword, tb_tesaurus, dan tb_cache tidak memiliki relasi dengan entitas karena hanya sebagai penyimpan data.

3.4.3 Flowchart Sistem

3.4.3.1 Flowchart Admin

- a. Flowchart Admin saat Mengindeks dokumen

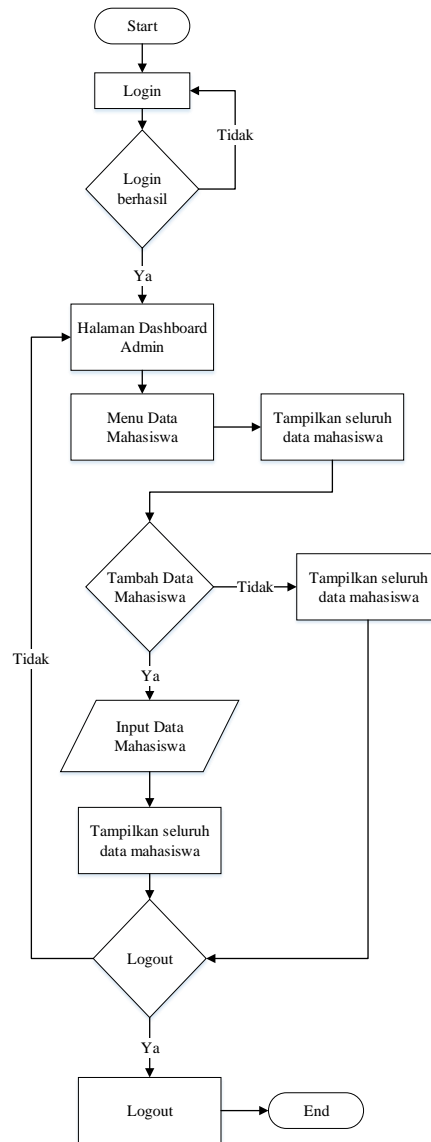
Flowchart pada Gambar 3.8 dapat dilihat proses pada saat admin ingin mengindeks dokumen. Saat admin berhasil login, maka akan dihadapkan pada halaman dashboard. Pada menu data dokumen ditampilkan semua data dokumen yang telah terupload. Dapat harus memilih button indeks dokumen untuk mulai melakukan indexing. Ketika indexing telah selesai maka akan ditampilkan



Gambar 3.8 Flowchart Admin saat Mengindeks Dokumen

dokumen yang telah terindeks. Ketika admin tidak ingin mengindeks dokumen, maka hanya akan ditampilkan dokumen yang telah terupload.

b. Flowchart Admin saat Menambah Data Mahasiswa

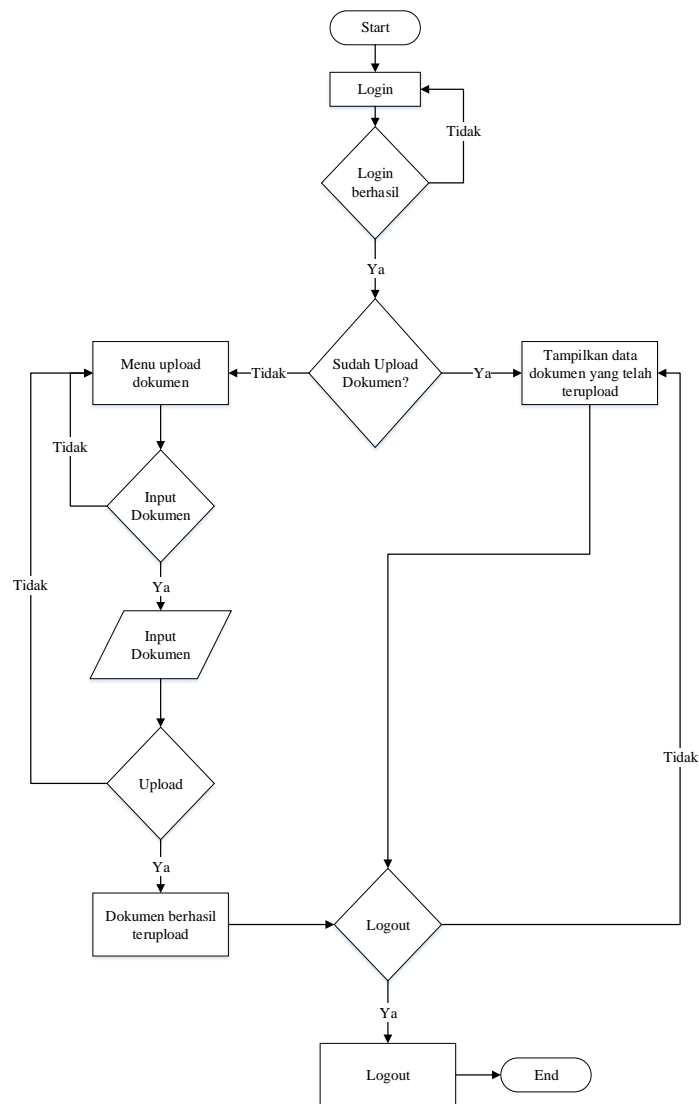


Gambar 3.9 Flowchart Admin saat Menambah Data Mahasiswa

Untuk menambah data mahasiswa seorang admin melewati proses seperti pada flowchart pada Gambar 3.9. Setelah berhasil login, maka admin masuk ke halaman dashboard. Dari halaman dashboard admin dapat memilih menu Data Mahasiswa. Dari menu ini admin dapat menambah data mahasiswa baru. Ketika data berhasil diinput maka akan ditampilkan kembali semua data mahasiswa.

3.4.3.2 Flowchart User

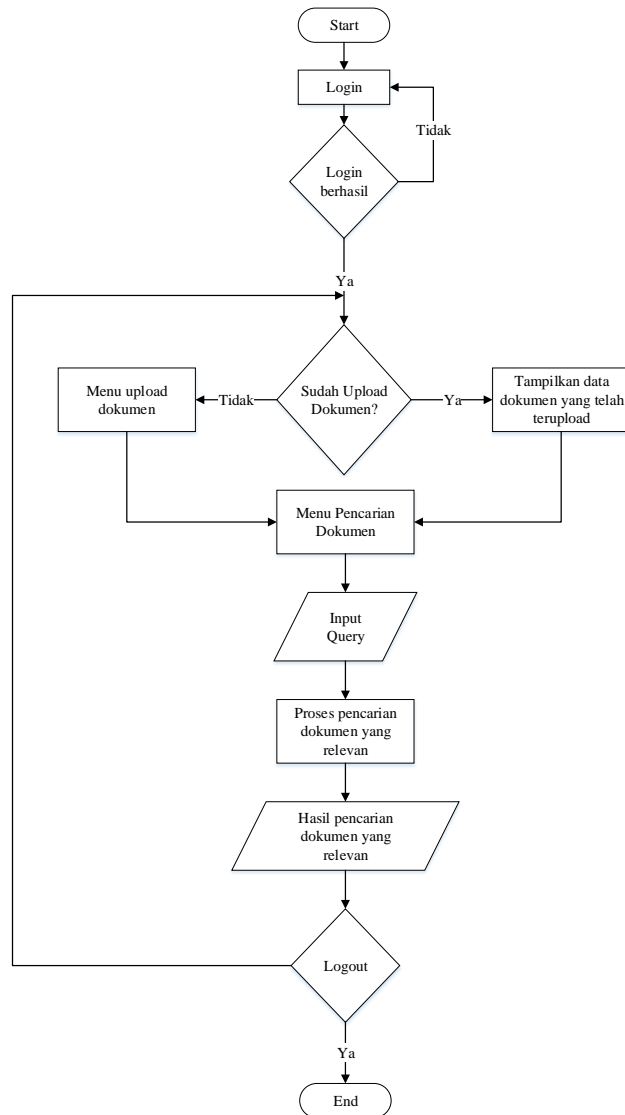
a. Flowchart User Saat Mengupload Dokumen



Gambar 3.10 Flowchart User Mengupload Dokumen

Proses dimulai ketika user login ke dalam sistem. Apabila user sudah pernah mengupload dokumen sebelumnya, maka akan dialihkan ke halaman yang menampilkan dokumen yang sudah pernah diupload. Apabila belum, maka akan dialihkan ke halaman upload dokumen. Dari halaman tersebut user dapat melakukan proses upload dokumen.

b. Flowchart User Saat Melakukan Temu Kembali Informasi Dokumen



Gambar 3.11 Flowchart User Saat Melakukan Temu Kembali Informasi

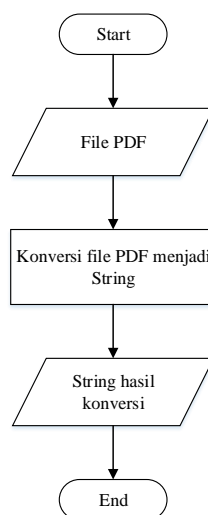
Proses dimulai ketika user melakukan login ke dalam sistem. Apabila user telah mengupload dokumen sebelumnya, maka akan dialihkan ke halaman dokumen yang sudah pernah diupload, sedangkan jika belum maka akan dialihkan ke halaman upload dokumen. Dari kedua halaman tersebut user dapat masuk ke halaman pencarian dokumen. Pada halaman tersebut disediakan form untuk menginputkan query atau kata kunci dari dokumen yang ingin dicari. Proses

pencarian dokumen kemudian dilakukan dengan metode yang digunakan. Hasil dari proses ini adalah daftar dokumen yang telah diranking menurut tingkat kemiripannya dengan query. User akan disajikan tabel yang memuat dokumen yang dicari.

3.4.3.3 Flowchart *Preprocessing*

a. Flowchart Parsing dokumen

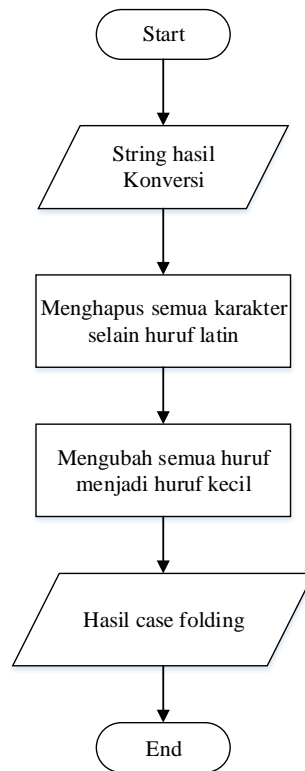
Parsing dokumen merupakan tahap pertama yang harus dilakukan dalam pengolahan data teks.



Gambar 3.12 Flowchart *Parsing*

File PDF yang diupload user pertama-tama diubah menjadi data teks atau string. Pengubahan file PDF menjadi string bertujuan untuk memudahkan dalam pengolahan selanjutnya.

b. Flowchart *Case Folding*

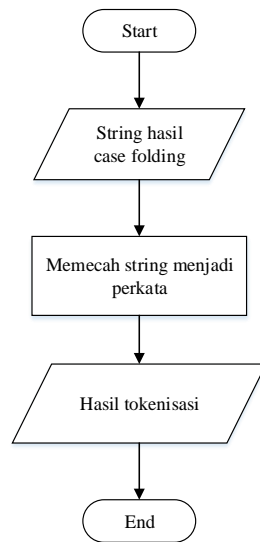


Gambar 3.12 Flowchart Case Folding

Setelah file PDF diubah menjadi string, proses selanjutnya adalah case folding. Proses pertama dari case folding adalah menghapus semua karakter kecuali karakter alfabet (huruf a sampai z). Kemudian semua karakter tersebut akan diubah menjadi huruf kecil (*lower case*).

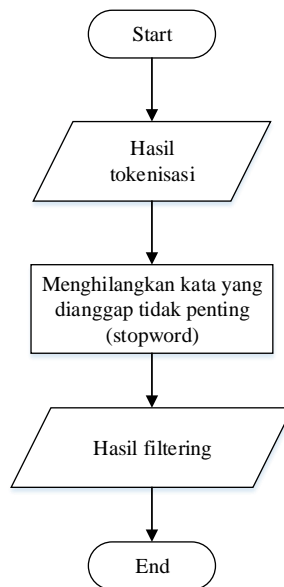
c. Flowchart *Tokenisasi*

Proses ketiga adalah tokenisasi. Tokenisasi bertujuan memisahkan kata-kata di dalam teks menjadi kata tunggal. Biasanya setiap kata dipisahkan oleh satu spasi. Dengan demikian proses tokenisasi menggunakan spasi sebagai parameter untuk memisahkan kata.



Gambar 3.13 Flowchart Tokenisasi

d. Flowchart Filtering

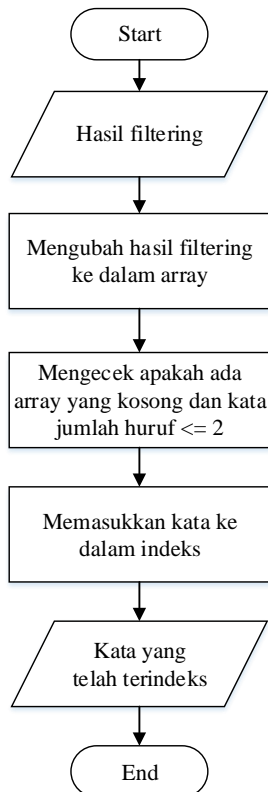


Gambar 3.14 Flowchart *Filtering*

Tahap *filtering* bertujuan untuk menghilangkan kata-kata yang kurang berpengaruh signifikan pada pencarian dokumen. Kata-kata ini sering disebut dengan stopwords. Stopwords yang dihilangkan pada proses ini adalah stopwords

Bahasa Indonesia dan Bahasa Inggris. Gambar 3.14 merupakan alur yang digunakan untuk menghilangkan stopwords.

e. Flowchart Indexing

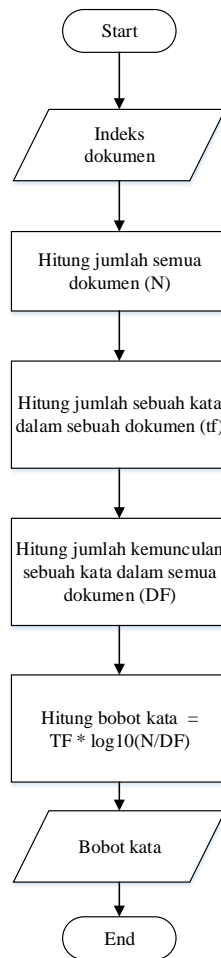


Gambar 3.14 Flowchart *Indexing*

Indexing merupakan proses memasukkan koleksi dokumen ke dalam suatu basis data. Hasil filtering sebelumnya pertama-tama diubah ke dalam bentuk array. Proses selanjutnya setiap array yang sudah berisi term, akan dilakukan pengecekan kembali. Array yang kosong atau berisi term dengan jumlah huruf kurang dari atau sama dengan dua tidak akan dimasukkan ke dalam tabel term. Alur proses indexing dapat dilihat pada Gambar 3.14.

3.4.3.4 Flowchart Pembobotan Indeks dengan TF – IDF

Setelah dihasilkan tabel yang berisi term beserta nama dokumennya, langkah selanjutnya adalah pembobotan term. Metode pembobotan yang digunakan adalah metode TF – IDF (Term Frquency Inverse Document Frequency).

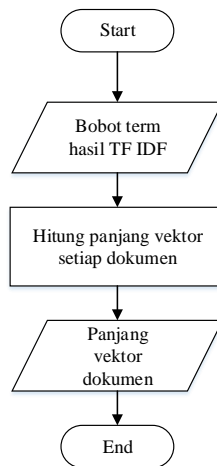


Gambar 3.15 Flowchart Pembobotan

Pertama-tama akan dihitung jumlah semua dokumen (N) yang ada pada tabel term. Selanjutnya akan dihitung jumlah suatu term yang ada pada satu dokumen (tf). Kemudian akan dicari jumlah term tersebut pada semua dokumen (DF). Setelah semua nilai tersebut didapatkan, maka akan dilakukan pembobotan term dengan rumus TF IDF (Bobot term = $TF * \log_{10}(N/DF)$). Begitu seterusnya hingga semua term mempunyai bobot. Selanjutnya term yang sudah mempunyai bobot dimasukkan ke dalam tabel index.

3.4.3.5 Flowchart Menghitung Panjang Vektor

Bobot term-term pada sebuah dokumen berguna untuk menghitung panjang vektor dokumen. Flowchart untuk menghitung panjang dokumen dapat dilihat pada Gambar 3.16



Gambar 3.16 Flowchart Menghitung Panjang Vektor

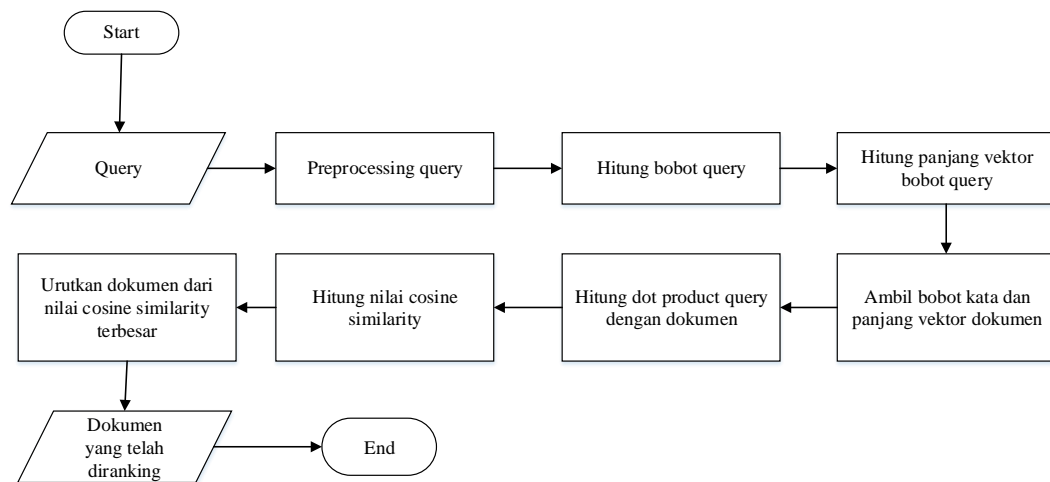
Pertama-tama akan diambil bobot term pada suatu dokumen pada tabel index. Kemudian dilakukan penghitungan panjang vektor dokumen dengan rumus

$$|D| = \sqrt{\sum_i W_{Q,j}^2}$$

Hasil panjang vektor lalu dimasukkan ke dalam tabel vektor.

3.4.3.6 Flowchart Temu Kembali Informasi tanpa Query Expansion

Proses temu kembali tanpa query expansion terdiri dari beberapa tahap yaitu, 1) preprocessing query, 2) menghitung bobot query, 3) menghitung panjang vektor bobot query, 4) Mengambil bobot term dokumen dan panjang vektor dokumen. 5) Menghitung dot product dokumen dengan query, 6) menghitung nilai similarity dokumen dengan query, 7) mengurutkan dokumen menurut nilai similarity.



Gambar 3.17 Flowchart Temu Kembali Informasi tanpa Query Expansion

Pada proses preprocessing query, query yang diinputkan pengguna akan dilakukan case folding, filtering, dan tokenisasi. Saat selesainya proses ini, kata-kata dalam query akan terbagi menjadi term-term yang berdiri sendiri. Term-term ini akan digunakan sebagai kata kunci untuk bobot term tersebut di dalam tabel index. Setelah bobot didapatkan bobot masing-masing term query, maka dilakukan proses menghitung panjang vektor bobot query.

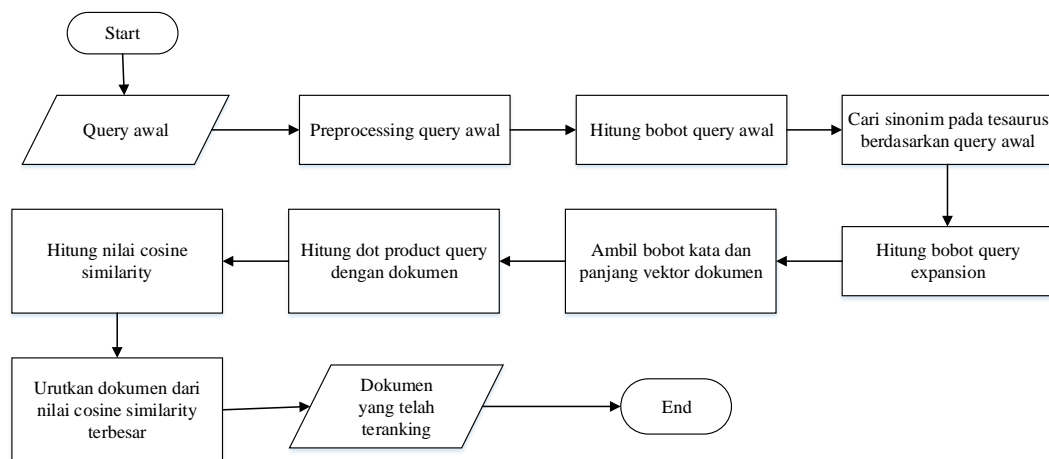
Setelah itu akan diambil bobot dari term pada tabel index yang mengandung query yang dimasukkan pengguna. Kemudian akan diambil panjang vektor dokumen mengandung query pada tabel vektor. Semua nilai yang diambil tersebut digunakan untuk menghitung dot product antara query dengan masing-masing dokumen.

Langkah selanjutnya adalah menghitung nilai cosine similarity. Dot product dari masing-masing dokumen akan dibagi dengan panjang vektor query yang dikalikan dengan panjang vektor dokumen. Hasilnya nilai similarity antara query dengan dokumen. Semakin besar nilai similarity maka makin tinggi tingkat kemiripan antar query dengan dokumen. Nilai similarity akan diurutkan dari besar ke kecil.

Proses terakhir adalah menampilkan nama dokumen beserta nilai similarity-nya kepada user dalam bentuk tabel.

3.4.3.7 Flowchart Temu Kembali Informasi dengan Query Expansion

Jika pada temu kembali informasi pada query expansion menggunakan query awal saja sebagai kata kunci dalam pencarian dokumen, maka pada jika menggunakan query expansion, query pengguna akan diperluas atau ditambahkan dengan term-term lainnya. Pada kasus ini query awal akan ditambahkan kata sinonim dari query tersebut.



Gambar 3.18 Flowchart Temu Kembali Informasi dengan Query Expansion

Proses STKI dengan menggunakan query expansion ada 8 tahap yaitu, 1) preprocessing query awal; 2) Menghitung bobot query awal; 3) Mencari sinonim pada tesaurus berdasarkan query awal; 4) Menghitung bobot query expansion; 5) Mengambil bobot kata dan panjang vektor dokumen; 6) Menghitung dot product antara query dengan dokumen; 7) Menghitung nilai cosine similarity antara query dengan dokumen; 8) Mengurutkan dokumen berdasarkan nilai similarity tertinggi.

Tahap pertama adalah tahap preprocessing query awal yang dimasukkan oleh user. Pada proses preprocessing query, query awal yang diinputkan pengguna akan dilakukan proses case folding, filtering, dan tokenisasi. Saat selesainya proses ini, kata-kata dalam query awal akan terbagi menjadi term-term yang berdiri

sendiri. Term-term ini akan dijadikan parameter untuk mencari bobot term pada tabel index. Bobot term-term ini akan disimpan ke dalam sebuah array.

Kemudian akan dicari sinonim kata pada kamus tersaurus berdasarkan term awal yang dimasukkan pengguna. Setelah sinonim didapatkan maka akan dilakukan pembobotan sinonim berdasarkan bobot query awal. Bobot sinonim adalah setengah dari bobot query awal. Kata sinonim yang telah diberi bobot dan dimasukkan ke dalam sebuah array selanjutnya disebut dengan query expansion. Dengan menggunakan bobot query expansion, maka dapat dihitung panjang vektor dari query expansion.

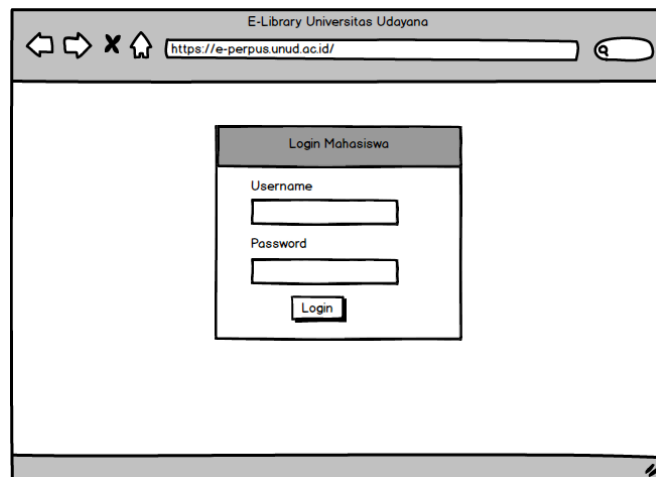
Langkah selanjutnya adalah mencari bobot term dan panjang dokumen yang mengandung query expansion. Bobot term dan panjang dokumen digunakan untuk menghitung nilai dot product. Kini query yang digunakan untuk mencari dot product adalah query expansion.

Setelah dicari dot product antar query expansion dengan dokumen, maka dilanjutkan dengan mencari nilai similarity. Untuk menghitung nilai similarity dibutuhkan panjang vektor query expansion dan panjang vektor dokumen. Nilai similarity adalah nilai dot product dibagi dengan panjang vektor query dikalikan panjang vektor dokumen. Nilai similarity yang didapatkan kemudian diurutkan dari yang terbesar.

Proses terakhir adalah menampilkan dokumen yang relevan kepada user berdasarkan nilai similarity yang diranking sebelumnya.

3.4.4 Perancangan Antarmuka Sistem

Bagian ini membahas rancangan antarmuka Sistem Temu Kembali Informasi E-Library.



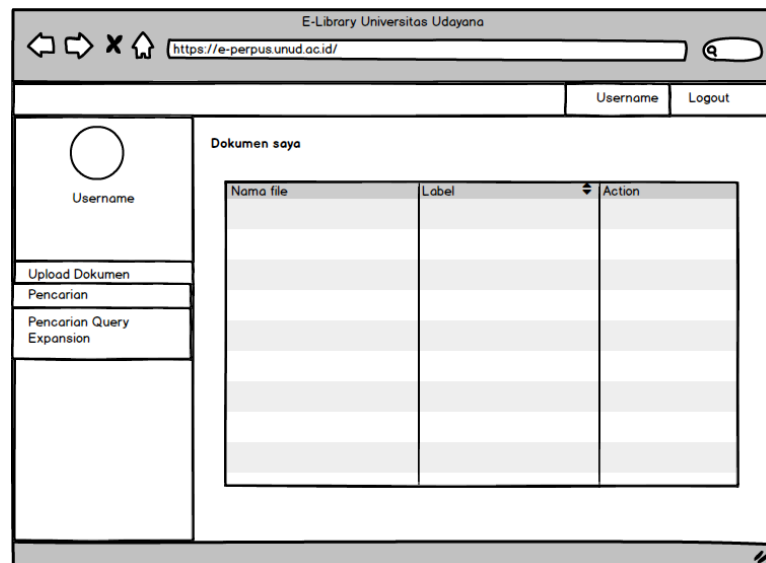
Gambar 3.19 Halaman Login Mahasiswa

Gambar 3.19 merupakan tampilan login untuk mahasiswa. Mahasiswa harus mempunyai username dan password untuk bisa login.

Gambar 3.20 Halaman Upload Dokumen

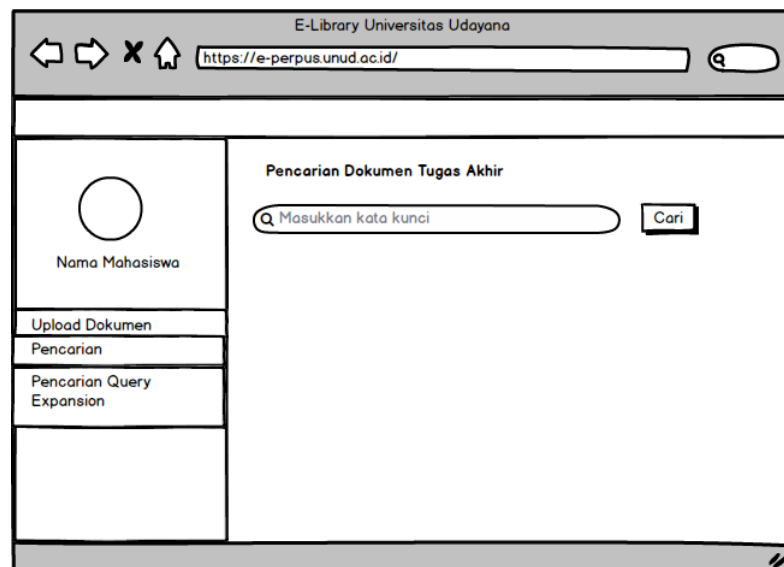
Setelah berhasil login, mahasiswa akan dibawa menuju halaman upload dokumen. Di sana terdapat form untuk menginput judul skripsi dan mengupload file skripsi mulai dari cover sampai halaman belakang.

Jika mahasiswa sudah pernah mengupload dokumen sebelumnya, maka akan dialihkan ke halaman seperti pada Gambar 3.21



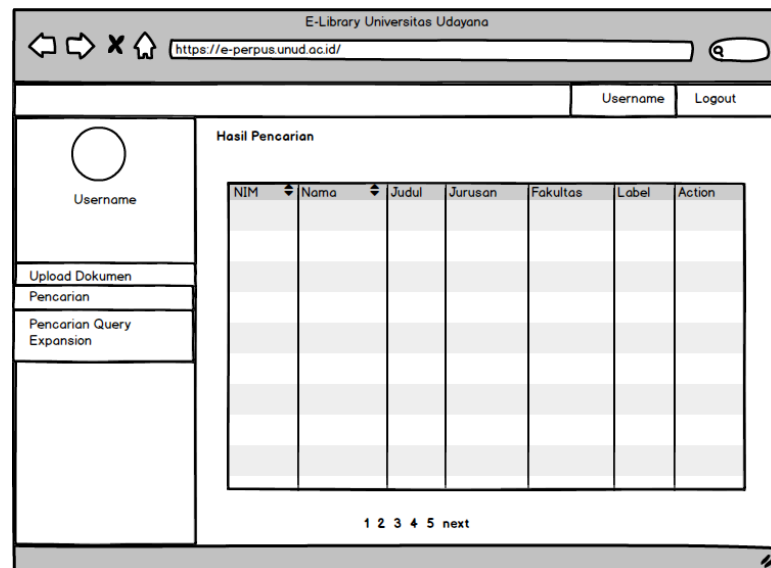
Gambar 3.21 Halaman Dokumen saya

Pada halaman tersebut akan ditampilkan dokumen yang pernah diupload oleh mahasiswa. Kemudian pada rancangan halaman pencarian dokumen dapat dilihat pada Gambar 3.22



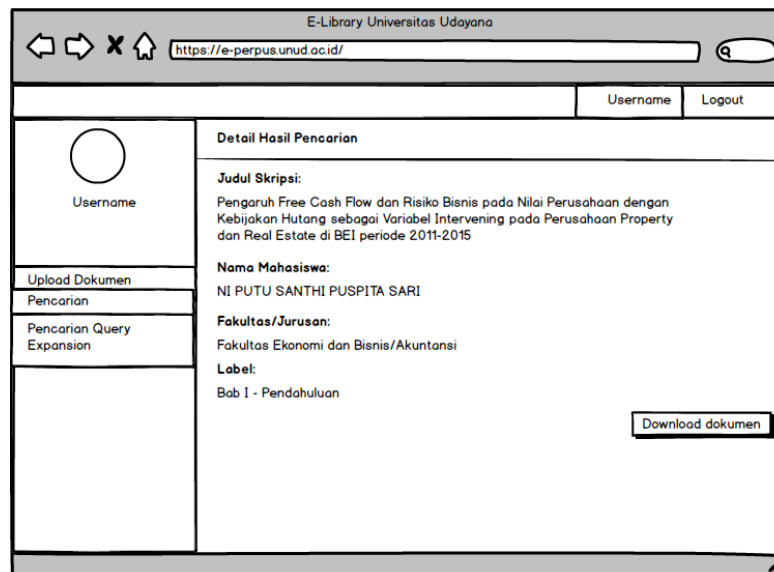
Gambar 3.22 Halaman Pencarian Dokumen

Ketika pengguna memasukkan query dan memilih button “Cari” maka akan ditampilkan halaman seperti pada Gambar 3.23



Gambar 3.23 Halaman Hasil Pencarian Dokumen

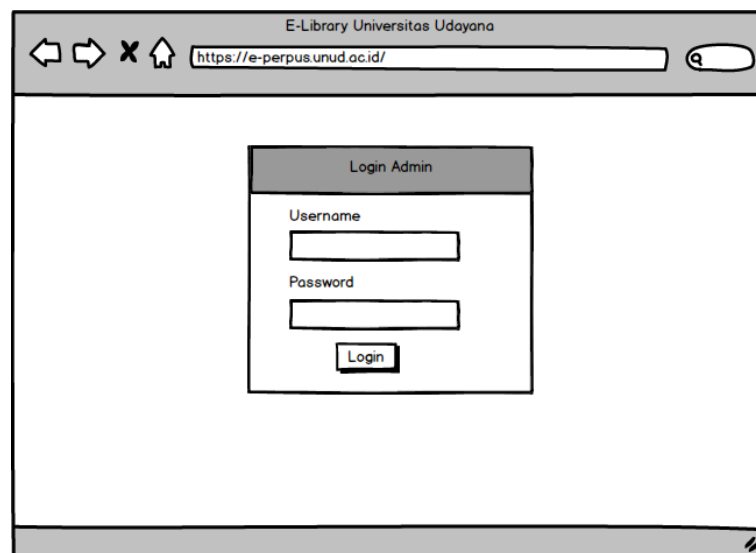
Saat pengguna memilih salah satu dokumen yang diinginkan, maka akan muncul halaman seperti Gambar 3.24.



Gambar 3.24 Halaman Data Hasil Pencarian

Pada halaman di Gambar 3.24 disajikan detail dari dokumen yang dipilih, berisikan judul skripsi, nama mahasiswa, fakultas/jurusa, dan label dokumen. Pengguna juga dapat mendownload dokumen melalui button “Download”.

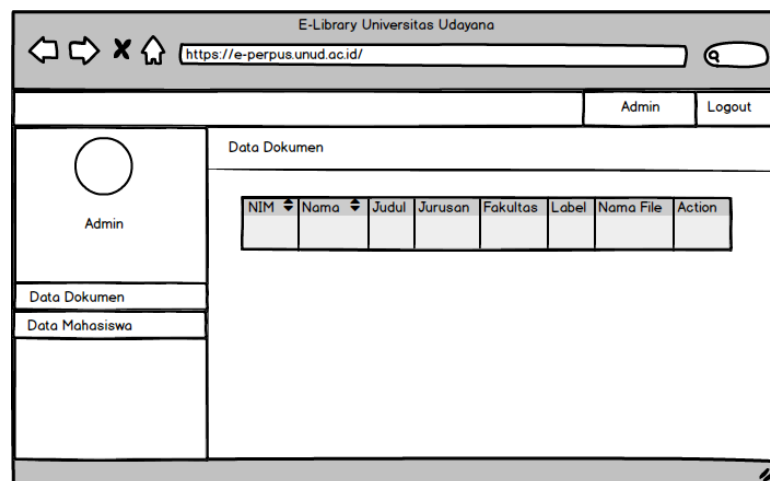
Dari sisi admin, rancangan antarmuka yang digunakan adalah sebagai berikut.



The screenshot shows a web browser window with the address bar containing 'https://e-perpus.unud.ac.id/'. The page title is 'E-Library Universitas Udayana'. The main content area is titled 'Login Admin' and contains a form with 'Username' and 'Password' input fields, and a 'Login' button.

Gambar 3.25 Halaman Login Admin

Gambar 3.25 adalah halaman login untuk admin. Seorang admin harus mempunyai username dan password agar dapat login. Setelah berhasil login, admin akan dialihkan ke halaman seperti pada Gambar 3.26.



The screenshot shows a web browser window with the address bar containing 'https://e-perpus.unud.ac.id/'. The page title is 'E-Library Universitas Udayana'. The dashboard has a sidebar with 'Admin' and 'Logout' buttons. The main content area is titled 'Data Dokumen' and contains a table with columns: NIM, Nama, Judul, Jurusan, Fakultas, Label, Nama File, and Action.

NIM	Nama	Judul	Jurusan	Fakultas	Label	Nama File	Action

Gambar 3.26 Halaman Data Dokumen

Gambar 3.26 merupakan halaman data dokumen. Pada halaman ini admin dapat melihat semua dokumen yang diupload oleh mahasiswa. Admin juga dapat mengindeks dokumen dari halaman ini. Jika admin memilih button “Data Mahasiswa” maka akan ditampilkan halaman seperti pada Gambar 3.27.

The screenshot shows a web browser window with the address bar displaying 'https://e-perpus.unud.ac.id/'. The page title is 'E-Library Universitas Udayana'. The main content area is titled 'Data Mahasiswa' and contains a table with the following columns: NIM, Nama, Jurusan, Fakultas, and Action. Below the table is a button labeled 'Tambah User'. The left sidebar contains a navigation menu with a circular icon and the text 'Admin', and two buttons labeled 'Data Dokumen' and 'Data Mahasiswa'.

Gambar 3.27 Halaman Data Mahasiswa

Admin dapat menambah data mahasiswa baru dengan memilih button “Tambah Mahasiswa”. Setelah ini akan ditampilkan form untuk menambah data mahasiswa seperti yang terlihat pada Gambar 3.28.

The screenshot shows a web browser window with the address bar displaying 'https://e-perpus.unud.ac.id/'. The page title is 'E-Library Universitas Udayana'. The main content area is titled 'Tambah Mahasiswa' and contains a form with the following fields: Nama, NIM, Fakultas, and Jurusan. Below the form is a button labeled 'Tambah'. The left sidebar contains a navigation menu with a circular icon and the text 'Admin', and two buttons labeled 'Data Dokumen' and 'Data Mahasiswa'.

Gambar 3.27 Halaman Tambah Data Mahasiswa