BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

3.1.1 Analisis Masalah

Dilihat dari sistem yang sedang berjalan saat ini sistem pencarian yang berlaku masih manual, dimana pengunjung perpustakaan harus mencari buku satu per satu dengan atau tanpa bertanya ke pustakawan untuk mendapatkan informasi buku yang dibutuhkan. Pada saat mencari buku, sedikit pengunjung sudah mengetahui judul dan pengarang dari buku yang akan dicari, akan tetapi kebanyakan pengunjung perpustakaan justru tidak tahu identitas buku yang akan dicari seperti judul, pengarang, penerbit, dan lain-lain yang hendak dicari.

Permasalahan pada penelitian ini adalah sulitnya mencari buku yang sesuai kebutuhan jika tidak mengetahui informasi sedikitpun mengenai identitas buku yang dibutuhkan sehingga menyebabkan lamanya proses pencarian buku jika harus menganalisis satu per satu buku di perpustakaan.

Oleh karena itu dibutuhkan sistem yang dapat melakukan proses pencarian berdasarkan representasi kebutuhan dari pencari buku kedalam *query*. Dengan kata lain, sistem yang dibangun dapat menghitung ukuran kemiripan antara dokumen dengan *query*. Akan tetapi, ukuran kemiripan suatu dokumen yang dinilai oleh sistem belum tentu sama dengan yang dinilai oleh pengguna[11]. Oleh karena itu, sistem perlu mengklasifikasi ulang dokumen-dokumen yang akan ditampilkan ke pengguna.

3.1.2 Analisis Sistem yang sedang berjalan

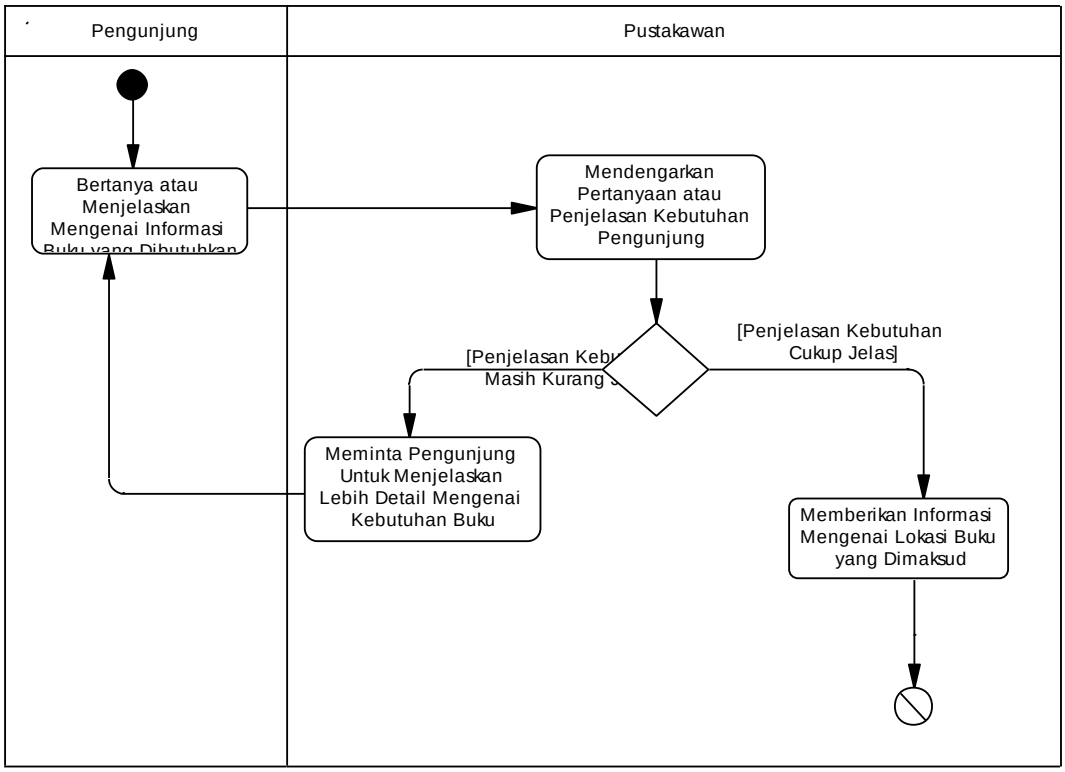
Proses pencarian buku diperpustakaan bisa dilakukan dengan cara pengunjung bertanya atau menjelaskan langsung mengenai buku yang dibutuhkan kepada pustakawan perpustakaan, jika dirasa pertanyaan atau penjelasan masih kurang jelas maka pustakawan meminta pengunjung untuk menjelaskan gambaran

kebutuhan pengunjung terhadap buku lebih jelas, jika sudah cukup jelas maka pustakawan merekomendasikan kepada pengunjung untuk mencari buku dilokasi yang disebutkan oleh pustakawan atau pustakawan turut mencari buku yang dibutuhkan pengunjung.

Proses pencarian yang sudah dijelaskan sebelumnya, memiliki banyak kekurangan, seperti :

1. Keterbatasan pengetahuan pustakawan mengenai informasi buku yang terdaftar di perpustakaan,
2. Proses pencarian membutuhkan waktu yang sangat lama.

Namun proses pencarian ini juga memiliki kelebihan yaitu tingkat keakuratan yang cukup tinggi, karena baik pengunjung atau pustakawan yang melakukan pencarian, sama-sama menganalisis buku satu per satu dari mulai judul hingga isi buku.



Gambar 3.1 Aktivitas Pencarian Buku di Perpustakaan

3.1.3 Analisis Kebutuhan Data

Sistem yang akan dibangun merupakan sistem pencarian buku untuk perpusatakaan daerah. Sebuah sistem pencarian hanya dapat melakukan proses pencarian jika tempat untuk pencarian ada, dan tempat pencarian tersebut adalah identitas buku, dalam penelitian ini data yang dibutuhkan berupa judul, pengarang dan deskripsi. Oleh karena itu, untuk menunjang pembangungan sistem ini, dibutuhkan identitas (judul dan pengarang) dan deskripsi buku yang sudah terdaftar di perpustakaan.

Berdasarkan hasil analisis dari data buku sebanyak 60 data buku dari berbagai golongan, maka diperoleh karakteristik teks dari data buku yang perlu diperhatikan adalah sebagai berikut :

a. Bahasa

   Terdapat beragam bahasa yang digunakan dalam buku yang ada di perpustakaan. Pada penelitian ini data buku yang digunakan merupakan data buku dengan teks berbahasa indonesia dan bahasa inggris.

b. Deskripsi Buku

   Untuk mendapatkan deskripsi buku, dapat dilihat dari *cover* belakang buku. Alternatif lain yang bisa dijadikan deskripsi buku yaitu kata pengantar atau daftar isi dari data buku.

c. Jenis Kata

   Terdapat beragam jenis kata yang ada data buku, yaitu :

   1. kata sesuai EYD,
   2. kata yang biasa digunakan sehari-hari,
   3. kata berimbuhan, dan
   4. kata dasar.

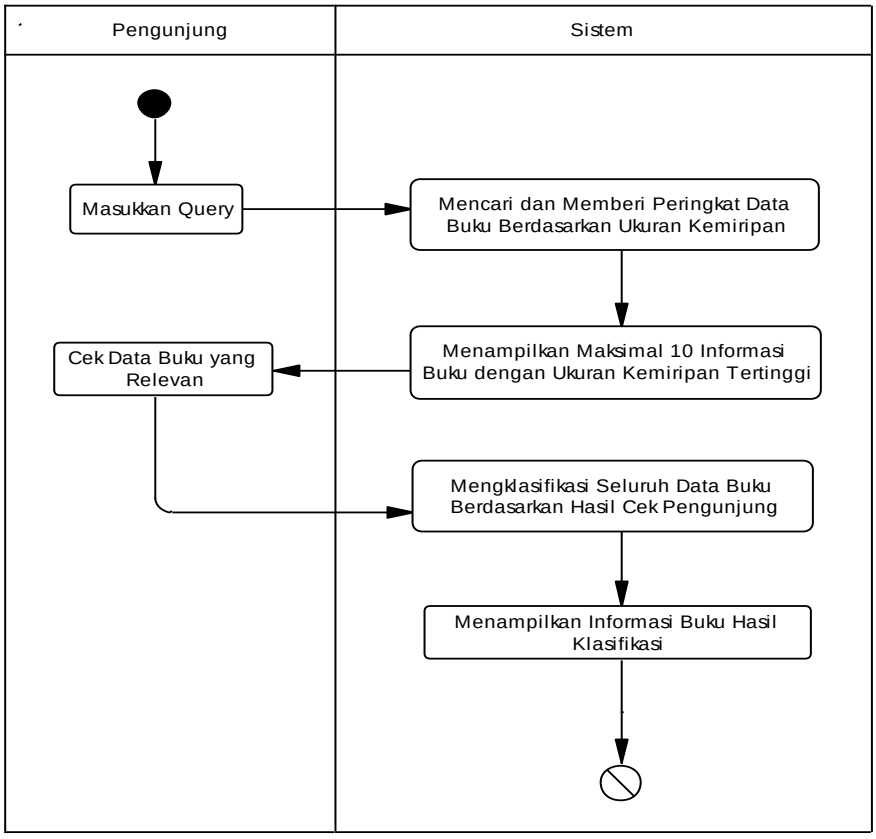
3.1.4 Analisis Metode

Sistem dibagi menjadi dua proses penting, yaitu proses pencarian dan proses pengolahan data.

Proses pencarian dapat digunakan oleh pengunjung untuk melakukan proses pencarian informasi buku dan lihat deskripsi dari buku itu sendiri. Proses pencarian dalam penelitian ini adalah proses pencarian dinamis, dimana pengguna juga dapat menentukan sesuai atau tidak nya *query* dengan informasi buku yang ditampilkan dari hasil pencarian, setelah itu sistem akan mengklasifikasikan seluruh data buku yang terdapat dalam *database* kedalam kategori relevan dan tidak relevan. Data buku yang ditampilkan adalah data buku yang masuk dalam kategori relevan.

Proses pengolahan data, khusus digunakan oleh operator untuk melakukan proses pengolahan (tambah, ubah, lihat detail, cari, dan hapus) data buku, pengolahan (tambah dan ubah) data golongan, dan melakukan optimisasi pada sistem pencarian.

Berikut dibawah ini adalah gambar dari aktivitas sistem pencarian informasi buku :



Gambar 3.2 Aktivitas Pencarian Informasi Buku

Terdapat pula sub proses untuk menunjang proses-proses yang sudah diajabarkan sebelumnya, yaitu:

3.1.4.1 Analisis *Vector Space Model* (VSM)

A. Analisis *Tokenizing*

Tokenizing adalah proses pemotongan string input berdasarkan tiap kata yang menyusunya serta membedakan karakter-karakter tertentu yang dapat diperlakukan sebagai pemisah kata atau bukan. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua kata ke bentuk huruf kecil (*lower case*).

Karakter-karakter yang akan dihapus atau dianggap sebagai pemisah kata, dapat dilihat di tabel 3.1.

Tabel 3.1 Karakter yang akan Dihapus

Karakter			
!	~	+	\
@	&	=	/
#	*	{	“
$	(	}	‘
%	)	[	:
^	-	]	;
`	-	|	.
,	<	>	?
white space (tab, spasi, enter)			

Tabel 3.2 Ilustrasi *Tokenizing*

No.	Isi	Hasil *Tokenizing*	
D1	Bersama Buku Pintar TOEFL ini, Anda bisa benar-benar pintar menguasai TOEFL dalam waktu singkat!	bersama buku pintar toefl ini anda bisa	benar pintar menguasai dalam waktu singkat
D2	Haji adalah sebuah ibadah yang menuntut pengorbanan total para pelakunya.	haji adalah sebuah ibadah yang	menuntut pengorbanan total para pelakunya
D3	Informasi yang terdapat dalam Buku Pintar Seri Junior mencakup banyak hal penting.	informasi yang terdapat dalam buku pintar	seri junior mencakup banyak hal penting
Q	Pintar menguasai TOEFL	pintar menguasai	toefl

B. Analisis *Filtering*

Filterting adalah Proses mengambil kata-kata penting dari hasil *tokenizng*. Untuk dapat melakukan proses *filtering* ini dapat menggunakan algoritma *stop-list* (membuang kata yang kurang penting / *stop word*) atau *word-list* (menyimpan kata penting).

Pada penelitian ini, untuk proses *filtering* digunakan algoritma *stop-list*

Berikut dibawah ini adalah daftar sebagian kata-kata *stop word* dari 569 kata yang terdapat dalam *database*. Jika dalam suatu data buku terdapat kata kata *stop-word*, maka sistem secara otomatis akan menghapus kata-kata tersebut.

Tabel 3.3 Daftar Sebagian *Stop-word*

Daftar *Stop Word*	
yang	sudah
mampu	tetapi
tentang	bisa
di	melakukannya

setelah	lakukan
semua	memang
hampir	baik
juga	lain
am	pernah
antara	setiap
dan	untuk
ada	dari
seperti	mendapatkan
jadi	punya
karena	telah
of	mr
mrs	…, dan lain-lain

Tabel 3.4 Ilustrasi *Filtering*

No.	**Hasil *Tokenizing***		**Hasil *Filtering***
D1	bersama buku pintar toefl ini anda bisa	benar pintar menguasai dalam waktu singkat	Pintar Toefl Menguasai Waktu
D2	haji adalah sebuah ibadah yang	menuntut pengorbanan total para pelakunya	haji ibadah menuntut pengorbanan total pelakunya
D3	informasi yang terdapat dalam buku pintar	seri junior mencakup banyak hal penting	informasi terdapat pintar seri junior mencakup
Q	pintar menguasai	toefl	pintar menguasai toefl

C. Analisis *Stemming*

Stemming adalah proses mencari kata dasar dari tiap kata hasil *filtering*. Pada penelitian ini digunakan algoritma Nazief & Adriani[23] untuk melakukan proses *stemming*.

Pada dasarnya, algoritma ini mengelompokkan imbuhan ke dalam beberapa kategori sebagai berikut:

1. *Inflection Suffixes* yakni kelompok-kelompok akhiran yang tidak mengubah bentuk kata dasar. Kelompok ini dapat dibagi menjadi dua:
   a. *Particle (P)* atau partikel, termasuk di dalamnya adalah partikel "-lah", "-kah", "-tah", dan "-pun".
   b. *Possessive Pronoun (PP)* atau kata ganti kepunyaan, termasuk di dalamnya adalah "-ku" , "-mu", dan "-nya".
2. *Derivation Suffixes (DS)* yakni kumpulan akhiran yang secara langsung dapat ditambahkan pada kata dasar. Termasuk di dalam tipe ini adalah akhiran "-i", "-kan", dan "-an".
3. *Derivation Prefixes (DP)* yakni kumpulan awalan yang dapat langsung diberikan pada kata dasar murni, atau pada kata dasar yang sudah mendapatkan penambahan sampai dengan 2 awalan. Termasuk di dalamnya adalah awalan yang dapat bermorfologi ("me-", "be-", "pe-", dan "te-") dan awalan yang tidak bermorfologi ("di-", "ke-" dan "se-").

Pada tabel 3.5, merupakan aturan pemenggalan jika kata berhimbuhan tidak termasuk ke dalam kategori manapun. Simbol C merupakan konsonan, V menandakan vokal, A merupakan vokal atau konsonan, dan P merupakan partikel.

Tabel 3.5 Aturan Pemenggalan Awalan

Aturan	Format Kata	Pemenggalan
1	berV…	ber-V... | be-rV...
2	berCAP…	ber-CAP... dimana C!=’r’ & P!=’er’
3	berCAerV...	ber-CaerV... dimana C!=’r’
4	belajar	bel-ajar
5	beC1erC2...	be-C1erC2... dimana C1!={’r’|’l’}
6	terV...	ter-V... | te-rV...
7	terCerV...	ter-CerV... dimana C!=’r’
8	terCP...	ter-CP... dimana C!=’r’ dan P!=’er’
9	teC1erC2...	te-C1erC2... dimana C1!=’r’
10	me{l|r|w|y}V...	me-{l|r|w|y}V...
11	mem{b|f|v}...	mem-{b|f|v}...
12	mempe...	mem-pe...
13	mem{rV|V}...	me-m{rV|V}... |me-p{rV|V}…
14	men{c|d|j|z}...	men-{c|d|j|z}...
15	menV...	me-nV... | me-tV
16	meng{g|h|q|k}...	meng-{g|h|q|k}...
17	mengV...	meng-V... | meng-kV...
18	menyV...	meny-sV...
19	mempV...	mem-pV... dengan V!=’e’
20	pe{w|y}V...	pe-{w|y}V...
21	perV...	per-V... | pe-rV...
22	perCAP	per-CAP... dimana C!=’r’ dan P!=’er’
23	perCAerV...	per-CAerV... dimana C!=’r’
24	pem{b|f|V}...	pem-{b|f|V}...
25	pem{rV|V}...	pe-m{rV|V}... | pe-p{rV|V}…
26	pen{c|d|j|z}...	pen-{c|d|j|z}...
27	penV...	pe-nV... | pe-tV...
28	peng{g|h|q}...	peng-{g|h|q}...
29	pengV...	peng-V... | peng-kV...
30	penyV...	peny-sV...
31	pelV...	pe-lV... kecuali “pelajar” yang menghasilkan “ajar”
32	peCerV...	per-erV... dimana C!={r|w|y|l|m|n}
33	peCP...	pe-CP... dimana C!={r|w|y|l|m|n} dan P!=’er’
34	terC1erC2...	ter-C1erC2... dimana C1!=’r’
35	peC1erC2...	pe-C1erC2... dimana C1!={r|w|y|l|m|n}

Pada hasil *filtering* ada beberapa kata yang masih memiliki imbuhan, maka kata-kata tersebut harus diproses lebih lanjut yaitu dengan menggunakan algoritma stemming Nazief dan Adriani. Pada contoh kasus hanya mengambil kata “pengorbanan”. Tahapan yang dilakukan untuk kata “pengorbanan”, dikerjakan dengan langkah-langkah berikut:

Berikut ini adalah contoh-contoh per tahapan dari proses penghilangan imbuhan :

1. Pada awal proses dan setiap langkah menghilangkan imbuhan akan mengecek ke kamus kata dasar, bila kata ditemukan maka proses berhenti.
2. Pada kata “pengorbanan” akan dicek apakah memiliki *inflection suffixes*, yaitu akhiran -lah, -kah, -ku, -mu, -nya, -pun, -tah. Ternyata tidak terdapat *inflection suffixes* pada kata “pengorbanan”, maka proses selanjutnya dijalankan.
3. Kemudian kata “pengorbanan” dicek apakah memiliki *derivation suffixes*, yaitu akhiran -i, -an, -kan. Ternyata ditemukan akhiran -an. Maka hapus akhiran -an, sehingga kata menjadi “pengenal”, kemudian sistem mengerjakan pencarian ke *database*, jika kata “pengenal” ditemukan maka proses berhenti jika tidak maka lanjut pada proses berikutnya.
4. Kata “pengorban” akan dicek, apakah memiliki *derivation prefiixes*, ternyata kata “pengorban” mengandung *derivation prefiixes* –pe, sehingga kata menjadi “ngorban”, kemudian sistem mencari kata “ngorban” ke *database* jika ditemukan maka kata “ngorban adalah kata dasar dan proses berhenti, jika tidak maka kata “ngorban” kembali menjadi kata “pengorban” dan melanjutkan proses berikutnya.
5. Setelah selesai menjalankan langkah 1 sampai 4, tetapi masih belum menemukan kata dasar maka tahapan selanjutnya adalah *recoding* menggunakan aturan pada tabel 3.4. Kata “pengorban” merupakan kata yang memiliki imbuhan peng- dan diikuti huruf vokal, dengan kondisi tersebut maka aturan yang dipakai adalah aturan nomor 29. Pertama sistem akan memotong kata dengan peng- maka hasilnya menjadi kata “orban”

selanjutnya sistem akan mencari kata “orban” ke *database* jika ditemukan maka kata “orban” adalah kata dasar jika tidak maka sistem akan memenggal kata “pengenal” dengan peng- dan menambah huruf “k” pada hasil pemenggalan sehingga kata “pengorban” menjadi “korban”, selanjutnya sistem mencari kata “korban” ke *database* dan ternyata kata “korban” terdapat dalam *database*, maka proses berhenti.

Tabel 3.6 Ilustrasi *Stemming* Nazief & Adriani

No.	Hasil *Filtering*	Hasil Stemming
D1	Pintar Toefl Menguasai Waktu	pintar toefl kuasa waktu
D2	haji ibadah menuntut pengorbanan total pelakunya	haji ibadah tuntut korban total laku
D3	informasi terdapat pintar seri junior mencakup	informasi dapat pintar seri junior cakup
Q	pintar menguasai toefl	pintar kuasa toefl

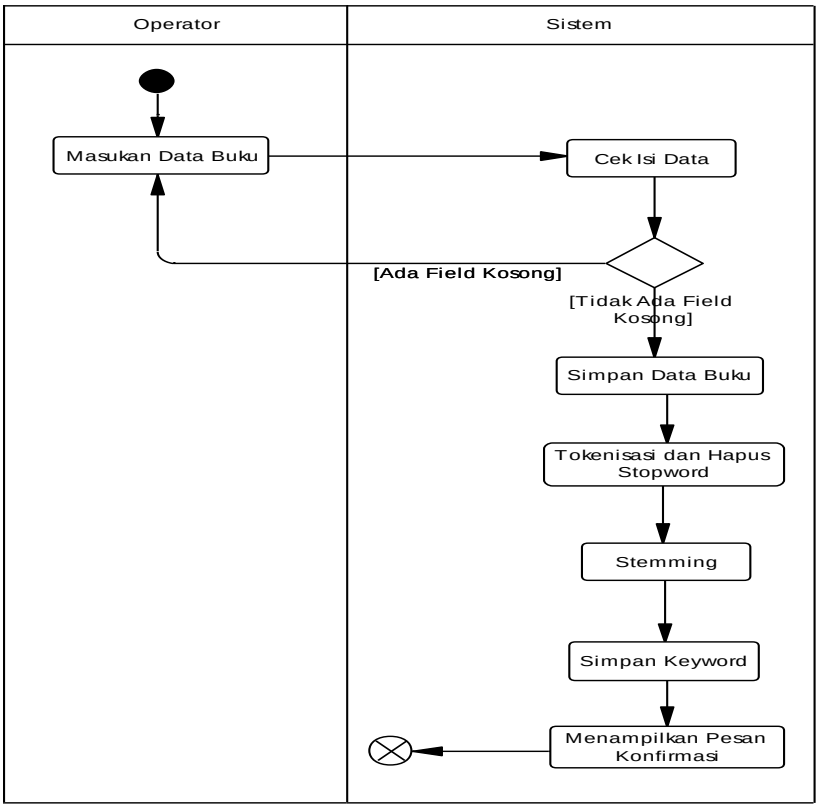
D. Analisis Pembuatan *Keyword*

Keyword yang dimaksud dalam penelitian ini adalah data buku yang merupakan gabungan dari judul, pengarang dan deskripsi buku yang sudah mengalami proses *tokenizing*, *filtering* dan *stemming*. Setiap data buku mempunyai *keyword*-nya masing-masing yang akan digunakan sebagai poros data untuk proses pencarian informasi buku.

Agar sistem dapat membuat *keyword* yang optimal maka disarankan untuk mengikuti aturan-aturan pengarangan data buku di bawah ini :

1. Data buku langsung diketik di form Penambahan atau Pengubahan Data Buku tidak disarankan untuk menyalin melalui media lain, seperti Microsoft Office.
2. Jika ada lebih dari satu pengarang maka gunakan tanda ” ; ” atau “ : ”sebagai pemisah antara pengarang yang satu dengan yang lainnya.
3. Jika ada poin-poin dalam deskripsi gunakan tanda “ - “ sebagai tanda poin dalam kalimat.

Berikut dibawah ini adalah gambar yang menjelaskan alur proses pembuatan *keyword* :



Gambar 3.3 *Activity Diagram* Pembuatan *Keyword*

E. Analisis Pembobotan

Pembobotan adalah proses merubah *keyword* dan *query* menjadi bentuk vektor. Ada beberapa tahapan yang harus dilalui agar tujuan tersebut bisa dicapai seperti menghitung *term frequency* (tf), *document frequency* (df),

inverse document frequency (idf) dan terakhir, mengalikan tf dengan idf sebagai bobot dari kata dalam suatu *keyword*.

tf : banyaknya kata yang dicari pada sebuah dokumen

df : banyak dokumen yang mangandung query

d : jumlah dokumen

$$idf = \log(d / df) \quad (1)$$

$$w = tf \times idf \quad (2)$$

Tabel 3.7 Pembobotan Kata

Kata	tf				df	d/df	idf	w			
	Q	D1	D2	D3				Q	D1	D2	D 3
pintar	1	1	0	1	2	3/2	0,176	0,176	0,176	0	0,176
toefl	1	1	0	0	1	3/1	0,477	0,477	0,477	0	0
kuasa	1	1	0	0	1	3/1	0,477	0,477	0,477	0	0
waktu	0	1	0	0	1	3/1	0,477	0	0,477	0	0
haji	0	0	1	0	1	3/1	0,477	0	0	0,477	0
ibadah	0	0	1	0	1	3/1	0,477	0	0	0,477	0
tuntut	0	0	1	0	1	3/1	0,477	0	0	0,477	0
korban	0	0	1	0	1	3/1	0,477	0	0	0,477	0
total	0	0	1	0	1	3/1	0,477	0	0	0,477	0
laku	0	0	1	0	1	3/1	0,477	0	0	0,477	0
informasi	0	0	0	1	1	3/1	0,477	0	0	0	0,477
dapat	0	0	0	1	1	3/1	0,477	0	0	0	0,477
seri	0	0	0	1	1	3/1	0,477	0	0	0	0,477
junior	0	0	0	1	1	3/1	0,477	0	0	0	0,477
cakup	0	0	0	1	1	3/1	0,477	0	0	0	0,477

F. Analisis Normalisasi

Setelah berhasil melakukan pembobotan, tahap selanjutnya yaitu melakukan normalisasi pada tiap dokumen, menggunakan rumus :

$$w(word_i) = \frac{w\,(word_i)}{\sqrt{w^2(word_1) + w^2(word_2) + \cdots + w^2(word_n)}} \quad (3)$$

Contoh bobot normalisasi dari D1 dengan kata “pintar” :

$$w(pintar) = \frac{0{,}176}{\sqrt{0{,}176^2 + 0{,}477^2 + 0{,}477^2 + 0{,}477^2}}$$

$$w(pintar) = \frac{0{,}176}{0{,}846} = 0{,}208$$

Maka didapatlah bobot yang baru tiap dokumen menjadi :

Tabel 3.8 Normalisasi

Kata	tf				df	d/df	idf	w			
	Q	D1	D2	D3				Q	D1	D2	D 3
pintar	1	1	0	1	2	3/2	0,176	0,252	0,208	0	0,163
toefl	1	1	0	0	1	3/1	0,477	0,683	0,564	0	0
kuasa	1	1	0	0	1	3/1	0,477	0,683	0,564	0	0
waktu	0	1	0	0	1	3/1	0,477	0	0,564	0	0
haji	0	0	1	0	1	3/1	0,477	0	0	0,408	0
ibadah	0	0	1	0	1	3/1	0,477	0	0	0,408	0
tuntut	0	0	1	0	1	3/1	0,477	0	0	0,408	0
korban	0	0	1	0	1	3/1	0,477	0	0	0,408	0
total	0	0	1	0	1	3/1	0,477	0	0	0,408	0
laku	0	0	1	0	1	3/1	0,477	0	0	0,408	0
informasi	0	0	0	1	1	3/1	0,477	0	0	0	0,441
dapat	0	0	0	1	1	3/1	0,477	0	0	0	0,441
seri	0	0	0	1	1	3/1	0,477	0	0	0	0,441
junior	0	0	0	1	1	3/1	0,477	0	0	0	0,441
cakup	0	0	0	1	1	3/1	0,477	0	0	0	0,441

G. Analisis Ukuran Kemiripan

Setelah melakukan perhitungan normalisasi, dilakukan suatu perhitungan untuk mendapatkan ukuran kemiripan antara dokumen d dengan *query* q. Ukuran ini memungkinkan perankingan dokumen sesuai dengan kemiripan relevansinya terhadap *query*. Setelah dokumen diberi peringkat selanjutnya sejumlah maksimal 10 dokumen, ditampilkan kepada pengunjung[11].

Untuk mendapatkan ukuran kemiripan, maka digunakan rumus :

$$sim(D_i, Q) = \frac{\sum D_i * Q}{\sqrt{\sum D_i^2 * \sum Q^2}} \quad (4)$$

Berikut dibawah ini adalah contoh perhitungan ukuran kemiripan dokumen :

$$sim(D1, Q) = \frac{(0{,}208 * 0{,}252) + (0{,}564 * 0{,}683) + (0{,}564 * 0{,}683)}{\sqrt{\left(0{,}208^2 + \left((0{,}564^2) * 3\right)\right) * (0{,}252^2 + 0{,}683^2 + 0{,}683^2)}}$$

$$sim(D1, Q) = \frac{0{,}822}{\sqrt{0{,}996 * 0{,}997}} = \frac{0{,}822}{0{,}996} = 0{,}825$$

Dengan cara yang sama, maka :

$$sim(D2, Q) = \frac{0}{\sqrt{0{,}996 * 0{,}996}} = \frac{0}{0{,}996} = 0$$

$$sim(D3, Q) = \frac{0{,}252 * 0{,}163}{\sqrt{1{,}171 * 0{,}996}} = \frac{0{,}041}{1{,}08} = 0{,}038$$

Dari hasil perhitungan di atas, dapat disimpulkan bahwa D1 memliki tingkat kemiripan paling tinggi yang berarti D1 merupakan informasi buku yang paling sesuai dengan *query* diikuti oleh D3 dan D2.

3.1.4.2 Analisis Support Vector Machines (SVMs)

A. Analisis Representasi Data

Dalam penelitian ini untuk format representasi data menggunakan format SVM light, hasil penelitian dari Thorsten Joachims. Format data input untuk klasifikasi SVM dalam penelitian ini adalah : +1 1:0.049 45:0.0294. Dengan masukan yang pertama +1 atau -1 menyatakan dua kelas Angka kedua menyatakan dimensi (row id) dan angka ketiga (setelah karakter ":")

menyatakan bobot dari *term* tersebut, tiap *term* dalam sebuah dokumen dipisahkan dengan spasi.

Data latih dan data uji secara bersamaan akan diubah menjadi data vektor. Data latih yang dimaksud adalah repersentasi data dari informasi yang ditampilkan pada proses pencarian menggunakan *Vector Space Model*. Sedangkan data tes adalah representasi seluruh data buku yang ada di *database*.

Data latih digunakan untuk melakukan proses pembelajaran terhadap sistem. Proses pembelajaran ini akan menghasilkan sebuah model data baru, atau dalam penelitian ini disebut *Model-File*. Setelah melakukan proses pembelajaran, sistem akan menguji kebenaran data tes terhadap *model-file*.

Sebagai contoh dalam pengubahan data teks menjadi data vektor digunakan sebuah *keyword* seperti di bawah ini :

"7700 peribahasa indonesia drs nur arifin chaniago bagas pratama spd tata bahasa bahasa indonesia kenal peribahasa peribahasa kandung makna dalam hidup itu sebab orang peribahasa sampai maksud ungkap judul 7700 peribahasa indonesia saji lengkap bahasa ringkas padat"

Format data vektornya adalah sebagai berikut :

```
1 1:0.239011 2:0.348346 3:0.0256246 4:0.119506 5:0.157205
6:0.157205 7:0.157205 8:0.157205 9:0.157205 10:0.157205
11:0.157205 12:0.113099 15:0.0818057 18:0.157205 19:0.157205
20:0.0974526 21:0.0696691 22:0.119506 23:0.157205 24:0.0974526
26:0.157205 27:0.119506 28:0.157205 29:0.157205 33:0.0597528
34:0.0376998 36:0.157205 37:0.157205
```

Angka 1 pada karakter pertama menyatakan data tersebut masuk dalam kelas dokumen relevan. Kata "7700" pada *keyword* di atas digantikan dengan 1:0.239011 pada data vektor yang berarti 1 sebagai indeks kata untuk "7700" dan 0.239011 sebagai bobot normalisasi untuk kata "7700" dalam file. Begitu seterusnya sehingga semua kata dalam *keyword* terwakili oleh data vektor. Untuk kata yang sama muncul lebih dari sekali dalam sebuah *keyword* akan

diwakili oleh sebuah data vektor saja dengan bobot normalisasi yang bersesuaian. Data vektor untuk sebuah kalimat diurutkan berdasarkan indeks kata dari terkecil hingga terbesar.

B. Analisis Pelatihan

Pada proses pelatihan SVM bertujuan untuk menemukan vektor α dan konstanta b. Dalam proses pelatihan dibutuhkan satu set *input-output* data atau dalam kasus ini dibutuhkan dokumen relevan dan dokumen tidak relevan dimana penilaian relevan atau tidaknya suatu dokumen ditentutkan oleh pengguna. Dokumen-dokumen relevan diberi label 1 dan dokumen-dokumen tidak relevan deberi label -1.

Untuk contoh kasus, digunakan dokumen-dokumen yang terdapat dalam contoh kasus VSM dan diasumsikan D1 dan D2 sebagai dokumen relevan menurut pengguna dan D3 sebagai dokumen tidak relavan. Maka, satu set *input-output* data dalam pelatihan bisa dilihat di tabel 3.9.

Tabel 3.9 Data Latih

	D_1	D_2	D_3
x1	0,208	0	0,163
x2	0,564	0	0
x3	0,564	0	0
x4	0,564	0	0
x5	0	0,408	0
x6	0	0,408	0
x7	0	0,408	0
x8	0	0,408	0
x9	0	0,408	0
x10	0	0,408	0
x11	0	0	0,441
x12	0	0	0,441
x13	0	0	0,441
x14	0	0	0,441
x15	0	0	0,441
y	1	1	-1

Tahap selanjutnya yaitu melakukan kernelisasi menggunakan fungsi polynomial kernel pangkat 2 yang didefinisikan sebagai $K(x_i, x_i) = (x_i + x_i' +$

1)^2. Matriks kernel K dihitung dengan dimensi lxl, dimana l adalah banyaknya data.

x_1 = [0,208 0,564 0,564 0,564 0 0 0 0 0 0 0 0 0 0]

x_2 = [0 0 0 0 0,408 0,408 0,408 0,408 0,408 0,408 0 0 0 0 0]

x_3 = [0,163 0 0 0 0 0 0 0 0 0 0,411 0,411 0,411 0,411 0,411]

x_1x_1' = [0,208 0,564 0,564 0,564 0 0 0 0 0 0 0 0 0 0]

[0,208 0,564 0,564 0,564 0 0 0 0 0 0 0 0 0 0]'

= 1

x_1x_2' = [0,208 0,564 0,564 0,564 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0,408 0,408 0,408 0,408 0,408 0,408 0 0 0 0 0]'

= 0

x_1x_3' = [0,208 0,564 0,564 0,564 0 0 0 0 0 0 0 0 0 0]

[0,163 0 0 0 0 0 0 0 0 0 0,411 0,411 0,411 0,411 0,411]'

= 0,034

x_2x_1' = [0 0 0 0 0,408 0,408 0,408 0,408 0,408 0,408 0 0 0 0 0]

[0,208 0,564 0,564 0,564 0 0 0 0 0 0 0 0 0 0 0]'

= 0

x_2x_2' = [0 0 0 0 0,408 0,408 0,408 0,408 0,408 0,408 0 0 0 0 0]

[0 0 0 0 0,408 0,408 0,408 0,408 0,408 0,408 0 0 0 0 0]'

= 1

x_2x_3' = [0 0 0 0 0,408 0,408 0,408 0,408 0,408 0,408 0 0 0 0 0]

[0,163 0 0 0 0 0 0 0 0 0 0,411 0,411 0,411 0,411 0,411]'

= 0

x_3x_1' = [0,163 0 0 0 0 0 0 0 0 0 0,411 0,411 0,411 0,411 0,411]

[0,208 0,564 0,564 0,564 0 0 0 0 0 0 0 0 0 0 0]'

= 0,034

x_3x_2' = [0,163 0 0 0 0 0 0 0 0 0 0,411 0,411 0,411 0,411 0,411]

[0 0 0 0 0,408 0,408 0,408 0,408 0,408 0,408 0 0 0 0 0]’

= 0

x_3x_3’ = [0,163 0 0 0 0 0 0 0 0 0 0,411 0,411 0,411 0,411 0,411]

[0,163 0 0 0 0 0 0 0 0 0 0,411 0,411 0,411 0,411 0,411]’

= 1

Maka matriks K adalah sebagai berikut :

$$\begin{bmatrix} 1 & 0 & 0{,}034 \\ 0 & 1 & 0 \\ 0{,}034 & 0 & 1 \end{bmatrix}$$

Selanjutnya hasil dari kernelisasi disubtitusikan ke persamaaan,

$$\max \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (4)$$

Maka didapatkan formulasi sebagai berikut :

$$\max \alpha_1\alpha_2\alpha_3 - \frac{1}{2} * \alpha_1^2 - (2 * 0{,}034\alpha_1\alpha_3) + \alpha_2^2 - \alpha_3^2$$

Untuk setiap $\alpha_1 + \alpha_2 - \alpha_3$ dimana $\alpha_1 \geq 0$.

Jika masalah pada formulasi diselesaikan, maka akan menghasilkan nilai

α_1 = 0,588	α_2 = 1,352	α_3 = 1,94	*b* = -0,409

Hasil tersebut diperoleh dengan menggunakan *liblary* SVM versi Ian Barber.

C. Analisis Klasifikasi Dokumen

Takashi Onoda, dkk[9], mengusulkan penggunaan SVM sebagai pengklasifikasian dokumnen dengan tahap-tahap berikut ini :

1. Inisialisasi Pencarian

Hasil pencarian ini diperoleh dari metode VSM dengan cara menampilkan *N* peringkat teratas dokumen yang memiliki kemiripan terhadap *query*.

2. Penilaian Dokumen oleh Pengguna

Pengguna mengevaluasi *N* peringkat teratas dokumen dan mengklasifikasi dokumen tersebut ke dalam dokumen relevan atau tidak relevan. Setelah pengguna selesai mengevaluasi, relevan dokumen dan dokumen yang tidak relevan diberi label +1 dan -1.

3. **Penentuan Optimal *Hyper-plane***

   Optimal *hyper-plane* untuk mengklasifikasikan relevan dan tidak relevan suatu dokumen dihasilkan oleh metode SVM.

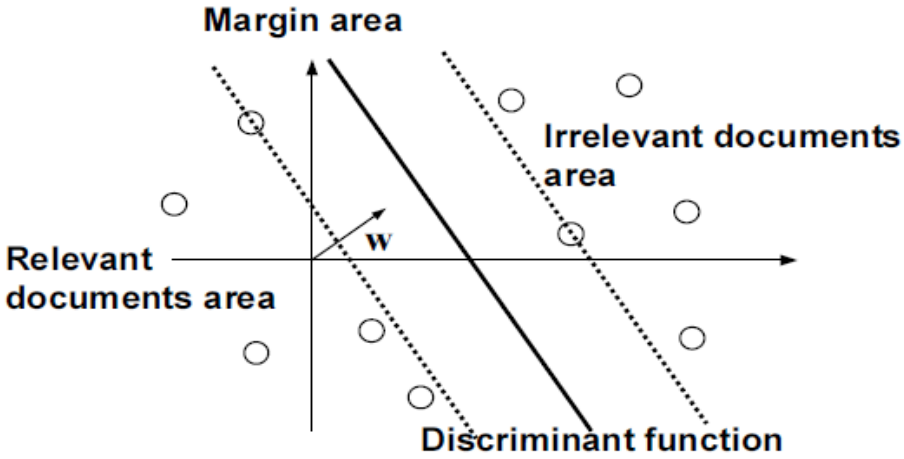
4. **Disikriminasi Dokumen dan Pencarian Informasi**

   Dokumen yang diperoleh dari langkah1 dipetakan ke dalam ruang fitur, SVM mengklasifikasikan dokumen-dokumen tersebut sebagai dokumen yang relevan dan tidak relevan. Kemudian sistem akan memilih dokumen-dokumen berdasarkan jarak optimal *hyper-plane* dan *margin area*.

5. **Menampilkan Hasil Akhir dari Pencarian**

   Dokumen yang didapat, di beri peringkat sesuai dengan jarak antara dokumen dengan fungsi pemisah *hyper-plane*, dimana fungsi pemisah ini ditetapkan oleh SVM. Dokumen yang didapat di tampilkan berdasarkan peringkat.

Pada inisiasi pencarian sistem akan menampilkan maksimal 10 dokumen yang memiliki ukuran kemiripan tertinggi[11]. Kemudian pengunjung menandai/*check list* beberapa dokumen yang relevan menurut penilaian pengunjung. Dokumen yang ditandai termasuk kedalam kategori relevan (yang tidak di tandai termasuk dokumen tidak relevan). Dokumen-dokumen tersebut dipetakan dalam ruang fitur (dengan cara kernelisasi) sebagai data latih, kemudian sistem melakukan pembelajaran atau dengan kata lain menghitung fungsi pemisah optimal yang dapat memisahkan dokumen relevan dengan tidak relevan secara optimal. Sehingga di ruang fitur terdapat dua area yaitu area dokumen relevan dan tidak relevan.



Gambar 3.4 Fungsi Pemisah antara Dokumen Relevan dan Tidak Relevan

Setelah selesai melakukan proses pembelajaran maka untuk menguji relevan atau tidaknya suatu dokumen dapat ditentukan dengan menggunakan fungsi pemisah sebagai berikut:

$$f(x) = \sum y_i \alpha_i K(x_{test}, x_{train}) + b \quad (5)$$

Dimana vektor α dan konstanta b diperoleh pada proses pembelajaran.

Setelah mendapatkan fungsi pemisah untuk dokumen relevan dan tidak relevan, sistem mengklasifikasikan seluruh dokumen yang terdapat di *database*, dengan cara membiarkan seluruh dokumen masuk kedalam area dokumen relevan (y =1), jika benar dokumen-dokumen tersebut masuk dalam area dokumen relevan maka sistem akan menampilkan dokumen-dokumen tersebut dan di urut berdasarkan jarak antara fungsi pemisah dengan dokumen. Jika salah dokumen-dokumen tersebut masuk dalam area relevan dokumen relevan maka sistem tidak akan menampilkan dokumen-dokumen tersebut.

3.1.5 Analisis Kebutuhan Non-Fungsional

Kebutuhan non fungsional yang dibutuhkan untuk membangun sistem ini mencakup dua hal yaitu; kebutuhan perangkat keras dan perangkat lunak.

3.1.5.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras minimum yang dibutuhkan untuk membangun sistem ini adalah :

1. Processor 2.0Ghz.
2. RAM 2 GB.
3. Hard Disk 40Gb.
4. Mouse dan Keyboard.
5. Jaringan internet dengan kecepatan 153 Kbps atau lebih.

Perangkat keras minimum yang dibutuhkan untuk dapat menjalankan sistem adalah :

1. Processor 2.0Ghz.
2. RAM 512 MB.
3. Mouse dan Keyboard.
4. Jaringan internet dengan kecepatan 153 Kbps.

3.1.5.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk membangun sistem ini adalah :

1. Adobe Dreamweaver CS5 atau Macromedia Dreamweaver, sebagai perangkat lunak yang digunakan untuk membangun sistem.
2. XAMPP 1.6.1 digunakan sebagai *webserver* dan PHP *Intepreter* di komputer lokal.
3. Sistem operasi seperti Windows, Ubuntu, dan lain-lain.
4. Aplikasi Browser seperti Maxthon, Chrome, Comodo dan lain-lain.

Adapun perangkat lunak yang dibutuhkan untuk dapat menjalankan sistem ini adalah :

1. Sistem operasi seperti Windows, Ubuntu, MacOS, dan lain-lain.
2. Aplikasi Browser seperti Maxthon, Chrome, Comodo dan lain-lain.

3.1.5.3 Analisis Kebutuhan Pengguna

Dalam penelitian ini ada dua jenis pengguna yang dapat menjalankan aplikasi dengan dua jenis hak akses yang berbeda yaitu :

1. Pengguna sebagai pengunjung : jenis pengguna yang memiliki hak akses untuk melakukan proses pencarian informasi buku dan pencarian informasi buku lebih lanjut serta melihat deskripsi dari hasil pencarian informasi buku agar pengunjung dapat menentukan relevan atau tidaknya informasi buku yang dihasilkan.
2. Pengguna sebagai operator : jenis pengguna yang memiliki hak akses untuk melakukan proses pengolahan (tambah, ubah, lihat detail, cari, dan hapus) data buku, pengolahan (tambah dan ubah) data golongan, dan melakukan optimisasi pada sistem pencarian.

Untuk dapat menjalankan aplikasi ini dibutuhkan spesifikasi pengguna baik sebagai pengunjung maupun sebagai operator.

Spesifikasi pengunjung diperlukan untuk dapat menjalankan aplikasi pada saat proses pencarian, sedangkan spesifikasi operator diperlukan untuk dapat menjalankan aplikasi pada saat proses pengolahan (tambah, ubah, lihat detail, cari, dan hapus) data buku, pengelolaan (tambah dan ubah) data golongan, dan melakukan optimalisasi pada sistem pencarian.

Operator akan diperankan oleh staff IT perpustakaan sedangkan pengunjung diperankan oleh siapapun yang hendak mencari buku diperpustakaan.

Tabel 3.10 Spesifikasi Pengguna

Pengguna	Tanggung Jawab	Hak Akses	Tingkat Keterampilan
Pengunjung	Mencari informasi buku.	Melakukan proses pencarian informasi buku dan lihat deskripsi buku.	Mampu mengoprasikan komputer, dapat melakukan *browsing* di *internet*, dan dapat mengikuti petunjuk penggunaan sistem

			pencarian informasi buku.
Operator (Staff IT)	Melaksanakan pengelolaan pangkalan data perpustakaan, pembinaan dan pengembangan otomasi perpustakaan, dan pengelolaan internet.	Melakukan pengelolaan data buku, pengelolaan data golongan, dan optimisasi sistem pencarian.	Mampu mengoprasikan komputer, dapat melakukan *browsing* di *internet*, dan mengerti teknis pengelolaan data.

Dilihat dari tabel spesifikasi pengguna diatas, tidak ada syarat yang terlalu rumit supaya bisa menjalankan aplikasi dari sistem pencarian informasi buku. Oleh karena itu tidak diperlukan adanya pelatihan kepada pengguna tetapi cukup dengan mengenalkan aplikasi dari sistem pencarian informasi buku.

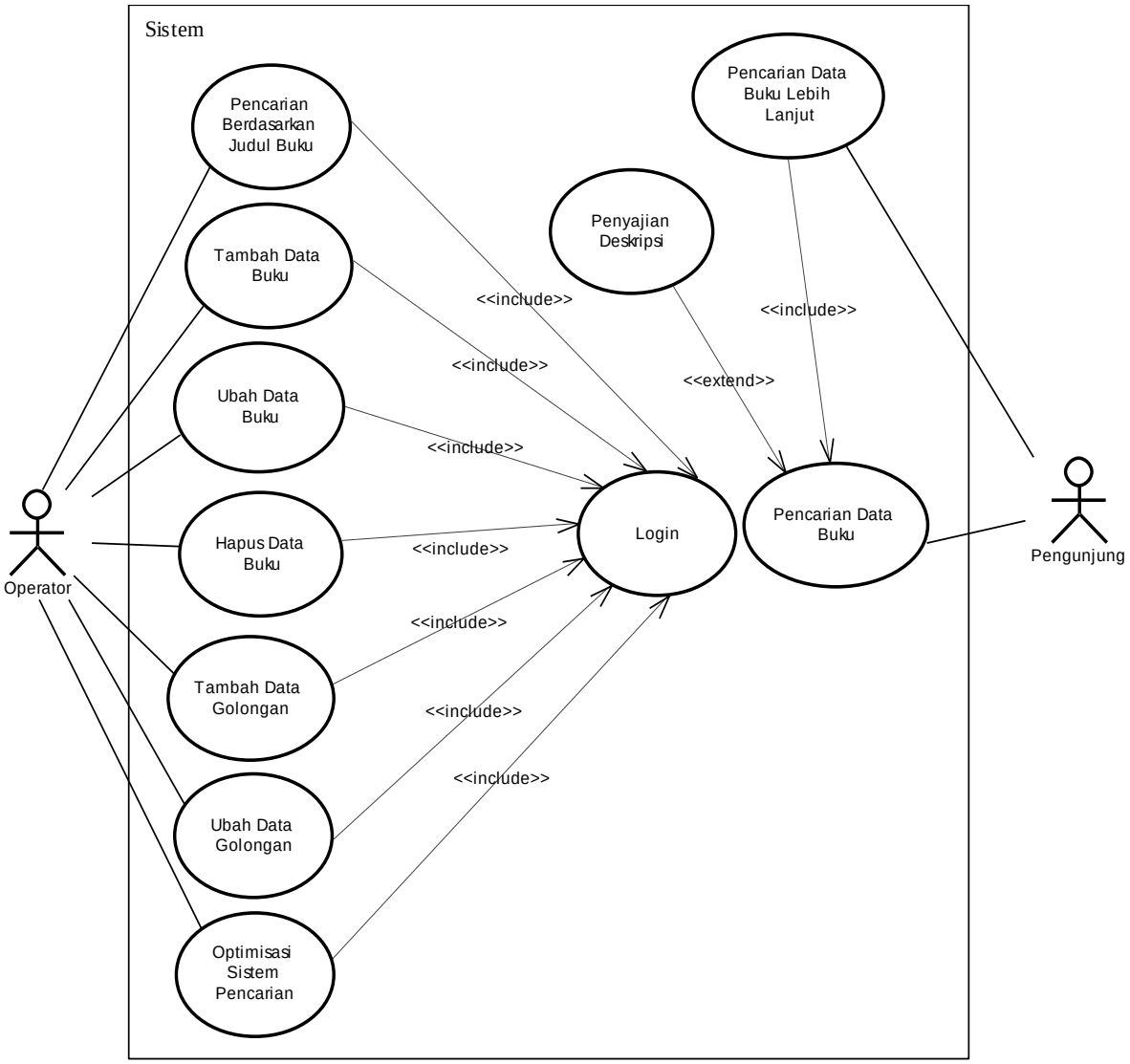
3.1.6 Analisis Kebutuhan Fungsional

Mendefinisikan analisis kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan aplikasi. Pada aplikasi ini yang akan bertindak sebagai aktor yaitu pengunjung dan operator. Operator berperan mengelola seluruh konten perangkat lunak. Pengunjung berperan dalam melakukan proses pencarian.

Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain identifikasi aktor, *usecase diagram*, skenario *usecase*, *activity diagram, sequence diagram* dan *class diagram*.

3.1.6.1 *Use Case Diagram*

Diagram use case memperlihatkan hubungan-hubungan yang terjadi antara aktor-aktor dengan use case dalam sistem. Salah satu manfaat dari diagram use case adalah untuk komunikasi. Calon pengguna sistem/perangkat lunak dapat mengamati diagram use case untuk mendapatkan pemahaman yang utuh tentang sistem yang akan dikembangkan.



Gambar 3.5 *Use Case Diagram* Sistem Pencarian Informasi Buku

3.1.6.2 Definisi Aktor

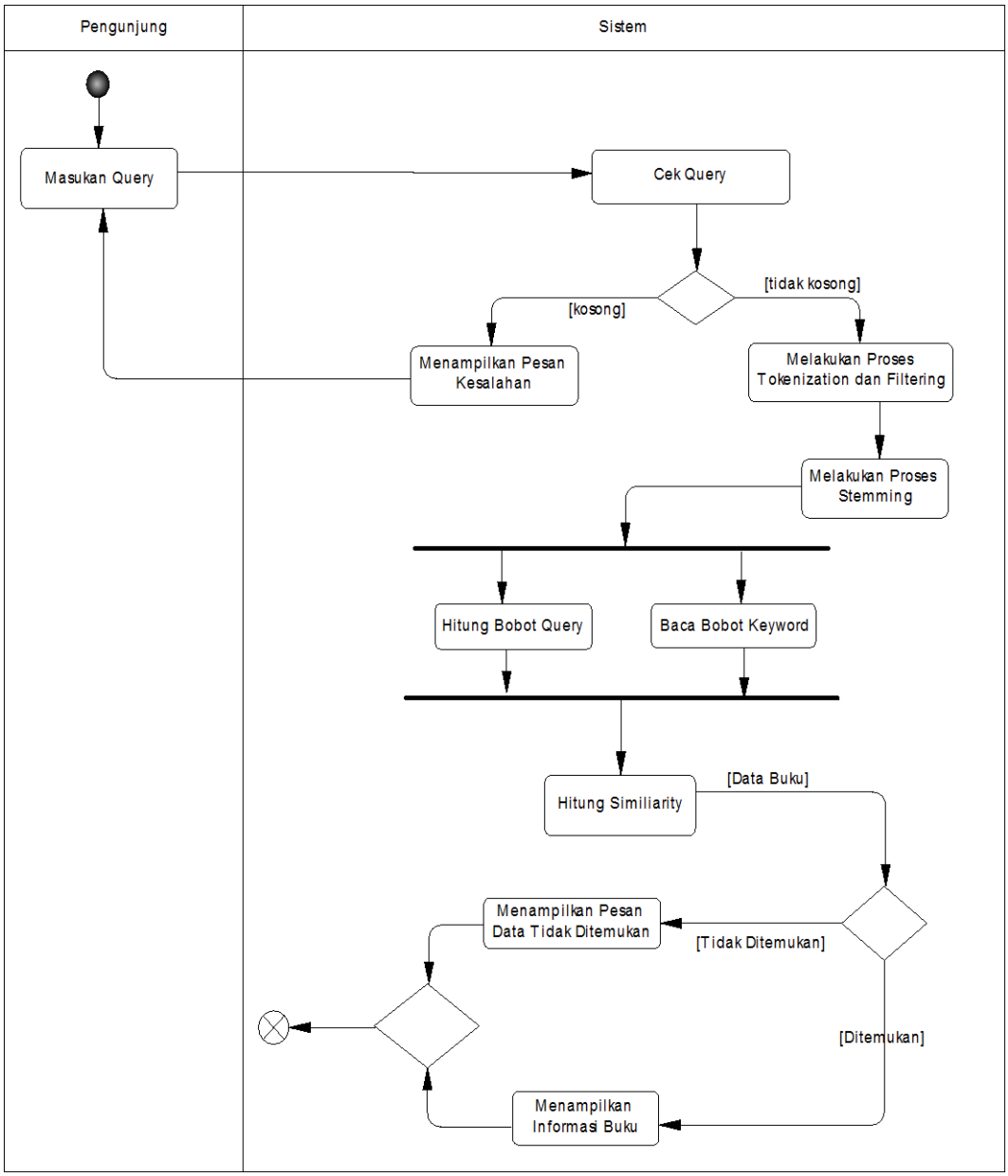
Tabel 3.11 Definisi Aktor

No	Aktor	Deskripsi
1	Pengunjung	Aktor ini mempunyai wewenang untuk mencari informasi buku berdasarkan representasi kebutuhan terhadap informasi buku yang dibutuhkan dan melihat deskripsi buku dari hasil proses pencarian.
2	Operator	Aktor ini mempunyai wewenang untuk melakukan proses pengolahan (tambah, ubah, lihat detail, dan hapus) data buku, pengolahan (tambah dan ubah) data golongan dan optimisasi sistem pencarian.

3.1.6.3 *Activity Diagram*

A. Pencarian Data Buku

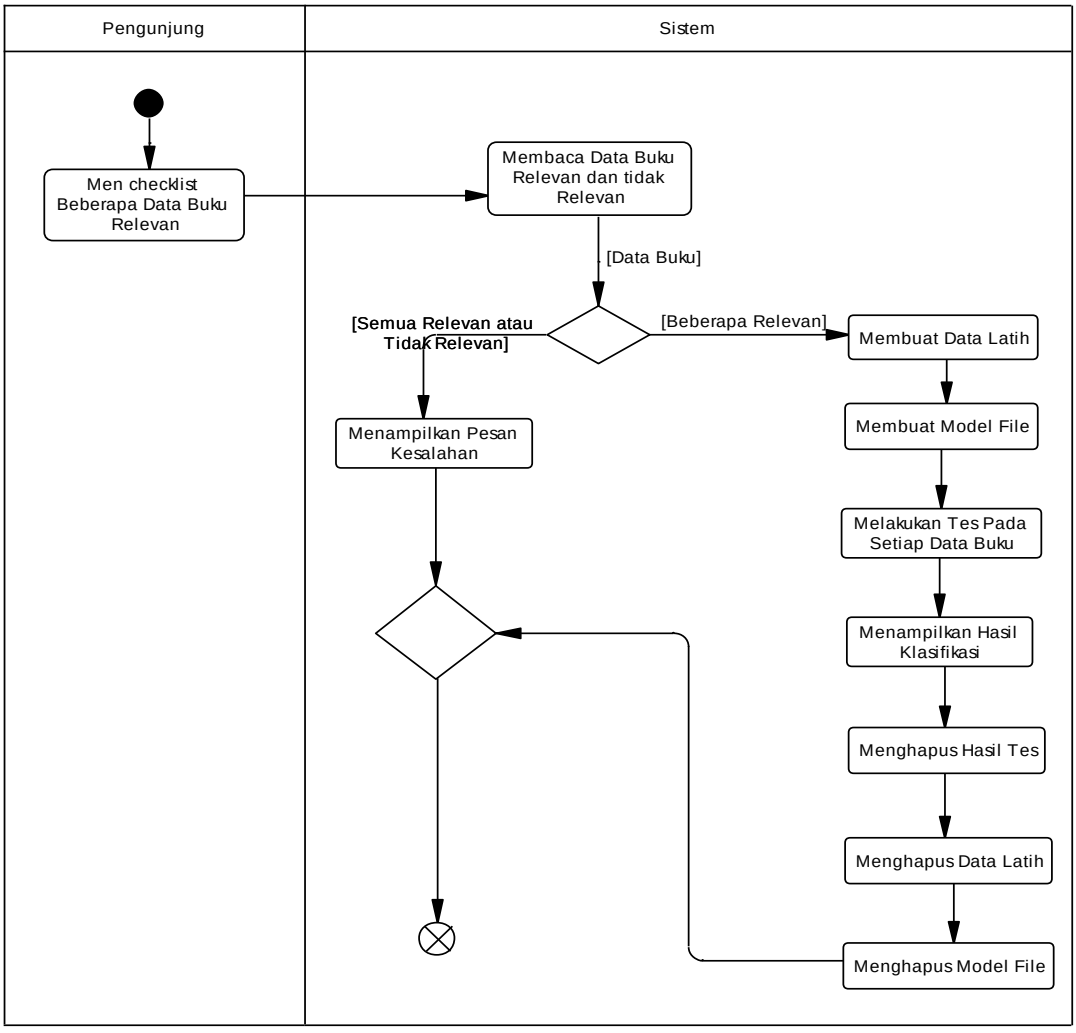
Aktifivas pencarian data buku merupakan aktivitas pengunjung untuk mencari informasi berdasarkan representasi kebutuhan pengguna terhadap buku yang dibutuhkan. Tahapan-tahapan aktivitas yang dilakukan pada saat pencarian data buku adalah :



Gambar 3.6 *Activity Diagram* Pencarian Data Buku

B. Pencarian Data Buku Lebih Lanjut

Aktivitas pencarian data buku lebih lanjut merupakan aktivitas pengunjung untuk menemukan informasi buku yang lebih relevan dengan cara memilih beberapa informasi buku hasil dari proses pencarian data buku yang relevan berdasarkan penilaian pengunjung. Tahapan-tahapan aktivitas yang dilakukan pada saat pencarian data buku lebih lanjut adalah :

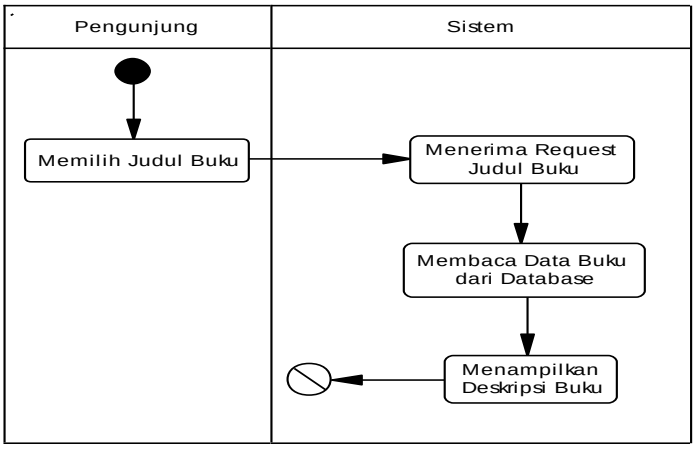


Gambar 3.7 *Activity Diagram* Pengklasifikasian Data Buku

C. Penyajian Deskripisi

aktivitas Penyajian deskripsi merupakan aktivitas pengunjung untuk melihat sebuah deskripsi buku aktivitas ini dapat dilakukan jika pengunjung sudah melakukan aktivitas pencarian data buku atau pencarian data buku lebih

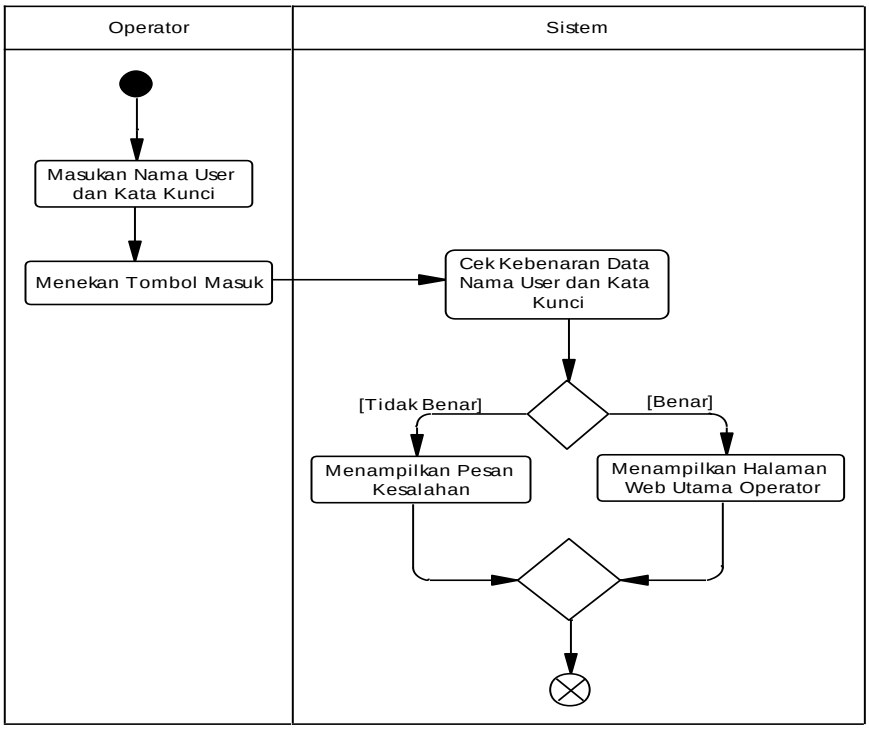
lanjut. Tahapan-tahapan aktivitas yang dilakukan pada saat lihat deskripsi buku adalah :



Gambar 3.8 *Activity Diagram* Lihat Deskripsi

D. Login

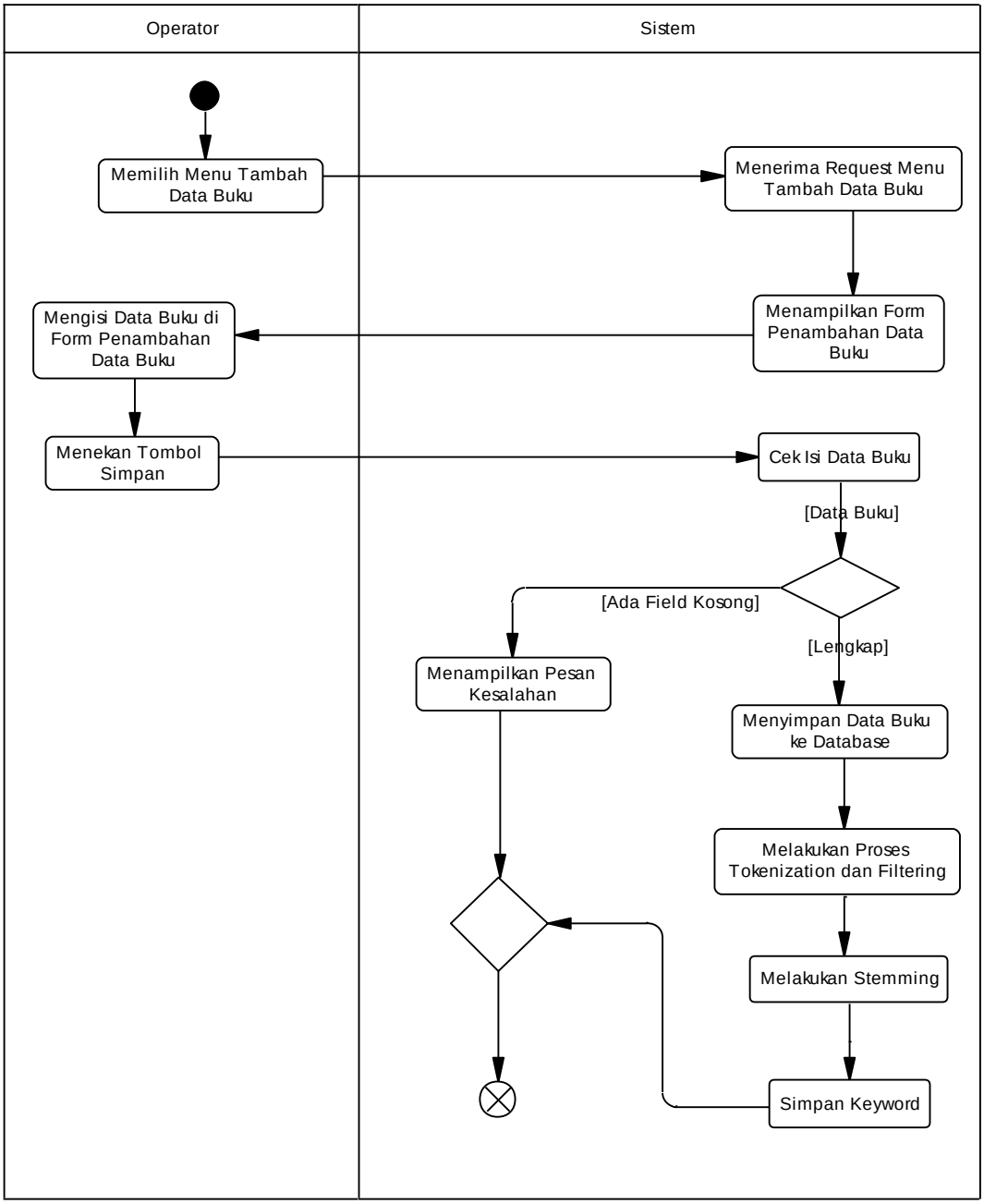
Aktifivas login merupakan aktivitas operator untuk mendapatkan hak akses pengolahan data buku dan golongan serta optimisasi sistem pencarian. Tahapan-tahapan aktivitas yang dilakukan pada saat pencarian data buku adalah :



Gambar 3.9 *Activity Diagram* Login

E. Tambah Data Buku

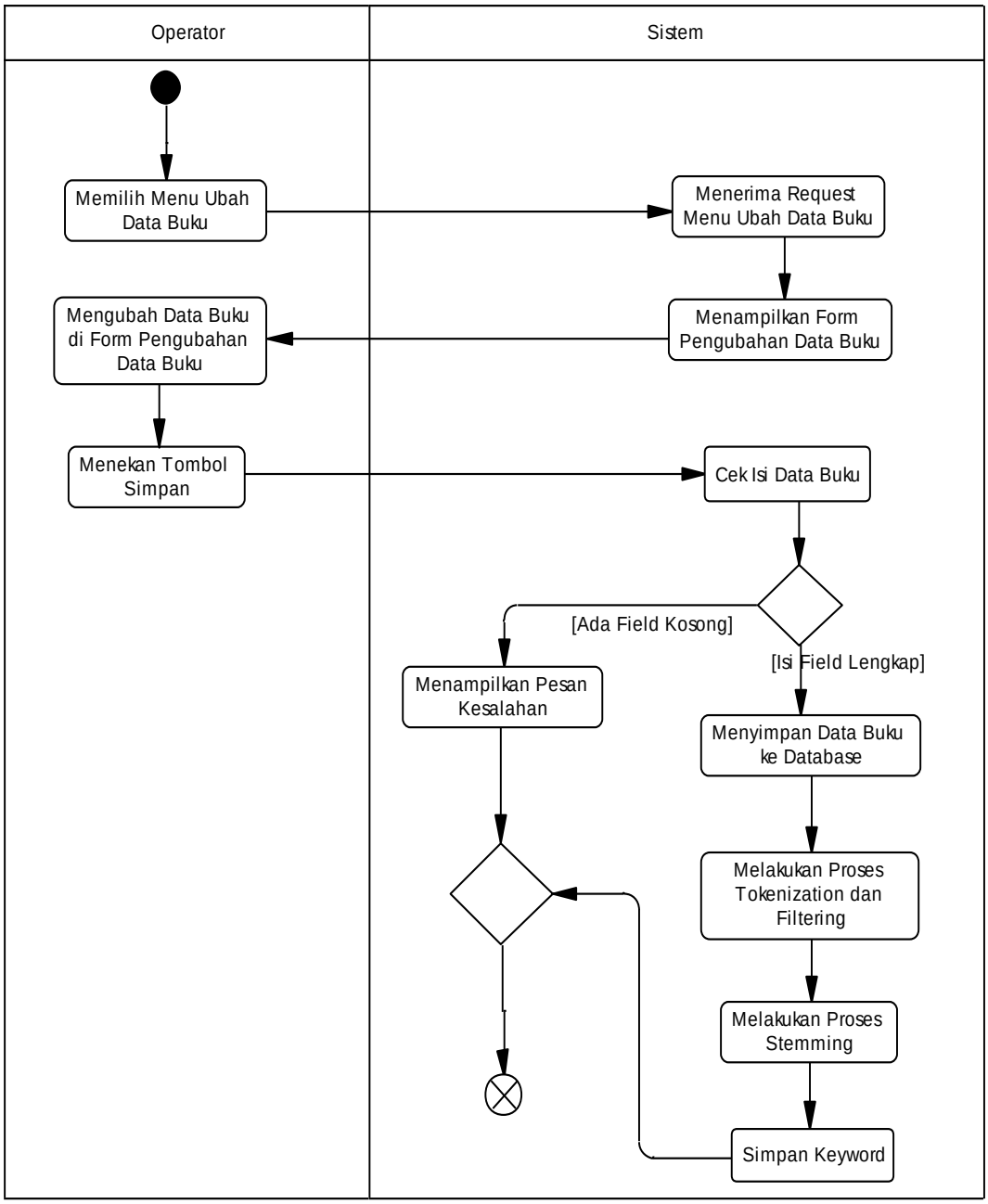
Aktivitas tambah data buku merupakan aktivitas operator untuk menambah data buku, aktivitas ini dapat dilakukan jika operator memilih menu tambah buku terlebih dahulu. Tahapan-tahapan aktivitas yang dilakukan pada saat tambah data buku adalah :



Gambar 3.10 *Activity Diagram* Tambah Data Buku

F. Ubah data buku

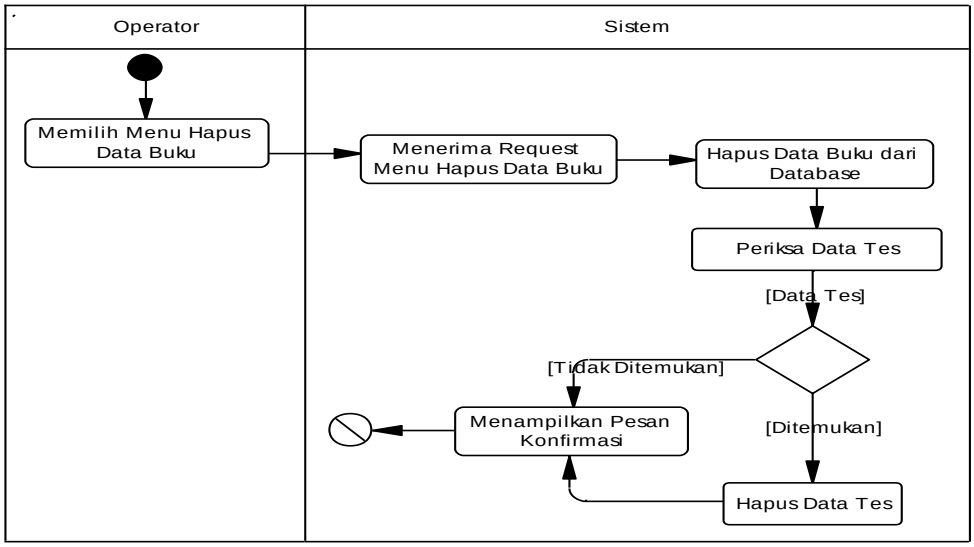
Aktivitas ubah data buku merupakan aktivitas operator untuk mengubah data buku, aktivitas ini dapat dilakukan jika operator memilih menu ubah buku terlebih dahulu. Tahapan-tahapan aktivitas yang dilakukan pada saat ubah data buku adalah :



Gambar 3.11 Activity Diagram Ubah Data Buku

G. Hapus Data Buku

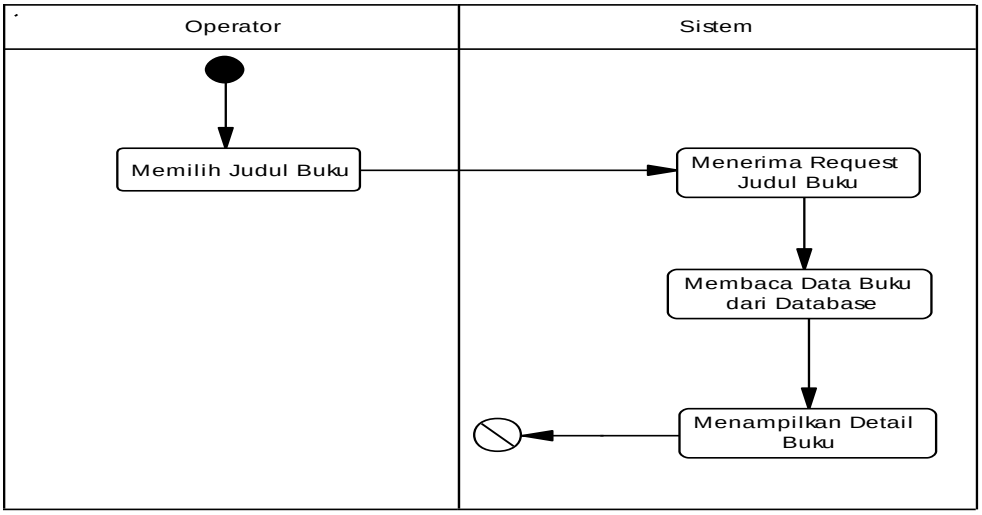
Aktivitas hapus data buku merupakan aktivitas operator untuk menghapus data buku, aktivitas ini dapat dilakukan jika operator memilih menu hapus buku terlebih dahulu. Tahapan-tahapan aktivitas yang dilakukan pada saat ubah data buku adalah :



Gambar 3.12 *Activity Diagram* Hapus Data Buku

H. Penyajian Detail Data Buku

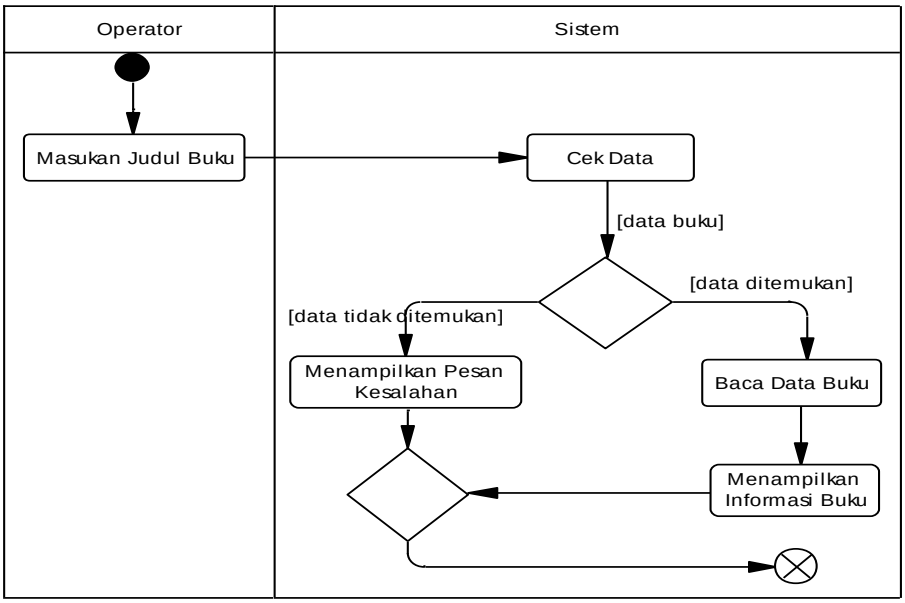
Aktivitas pencarian berdasarkan judul buku merupakan aktivitas yang dilakukan operator untuk mencari data buku. Tahapan-tahapan aktivitas yang dilakukan pada saat ubah data buku adalah :



Gambar 3.13 *Activity Diagram* Penyajian Detail Data Buku

I. Pencarian Data Buku Berdasarkan Judul

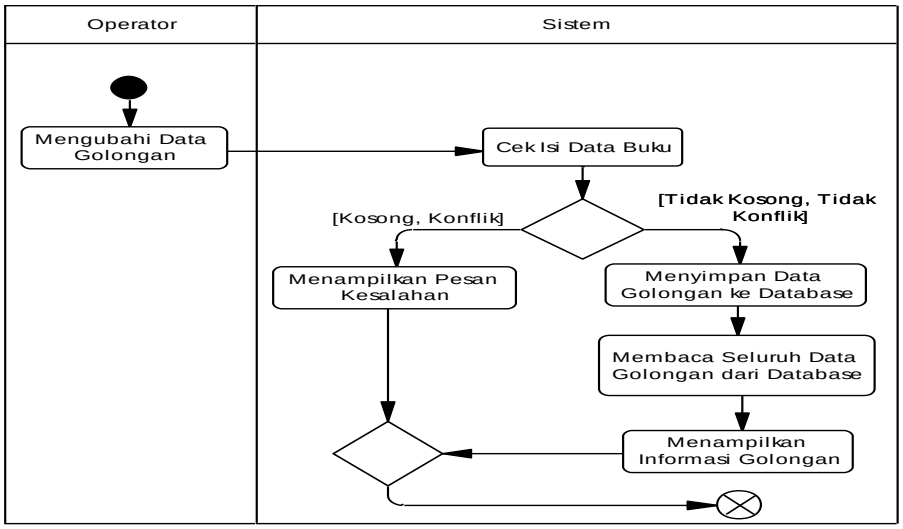
Aktivitas pencarian berdasarkan judul buku merupakan aktivitas yang dilakukan operator untuk mencari data buku. Tahapan-tahapan aktivitas yang dilakukan pada saat ubah data buku adalah :



Gambar 3.14 *Activity Diagram* Pencarian Berdasarkan Judul Buku

J. Tambah Data Golongan

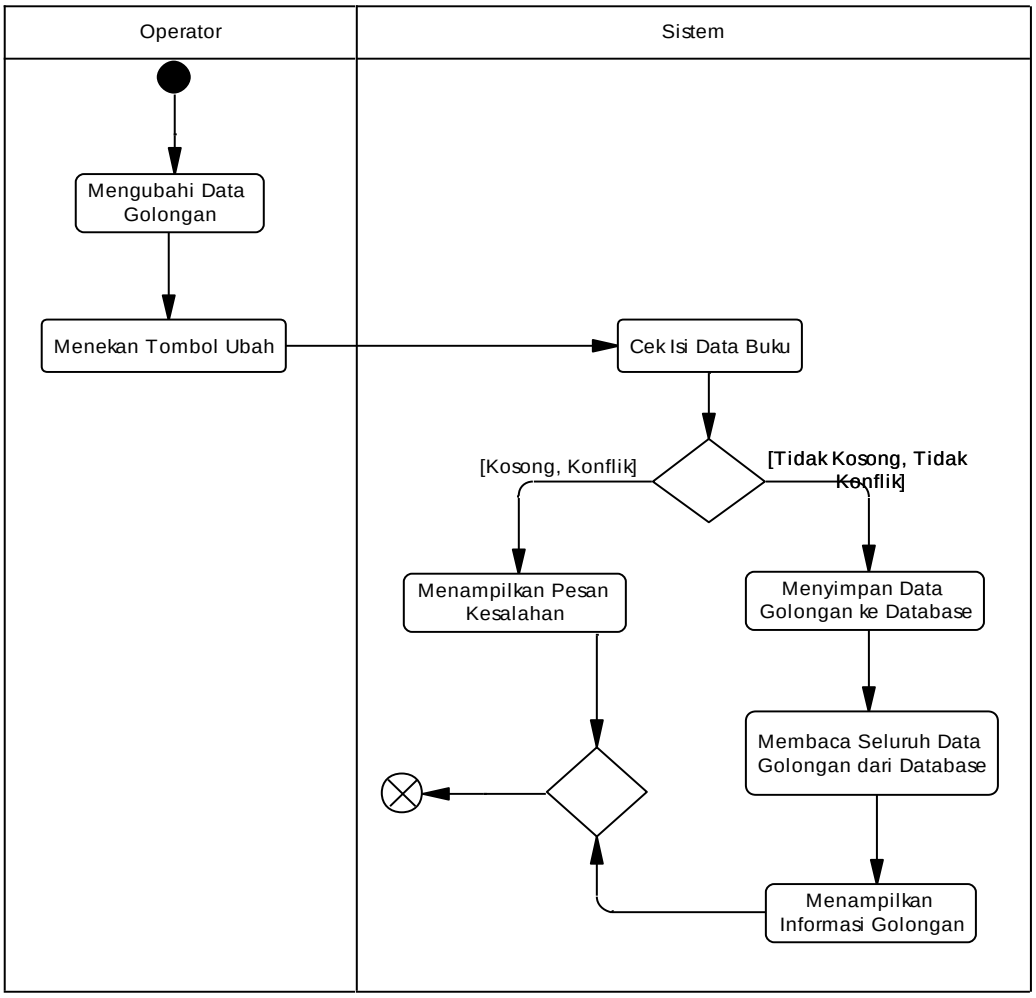
Aktivitas tambah data gologan merupakan aktivitas operator untuk menambah data golongan Tahapan-tahapan aktivitas yang dilakukan pada saat tambah data golongan adalah :



Gambar 3.15 *Activity Diagram* Tambah Data Golongan

K. Ubah Data Golongan

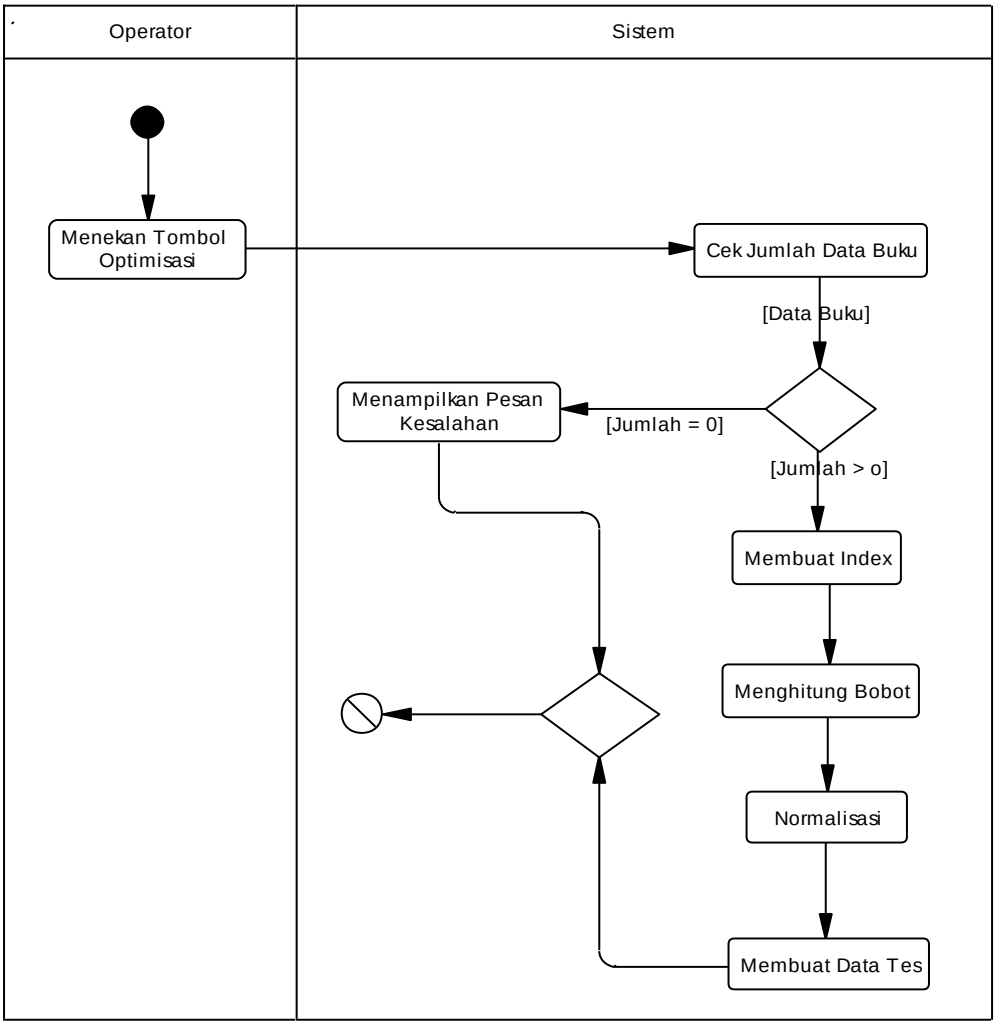
Aktivitas ubah data golongan merupakan aktivitas operator untuk mengubah data golongan. Tahapan-tahapan aktivitas yang dilakukan pada saat ubah data golongan adalah :



Gambar 3.16 Activity Diagram Ubah Data Golongan

L. Optimisasi Sistem Pencarian

Aktivitas Optimisasi merupakan aktivitas operator untuk menghitung bobot dan normalisasi data buku serta membuat data tes untuk setiap data buku. Tahapan-tahapan aktivitas yang dilakukan pada saat optimisasi adalah :



Gambar 3.17 *Activity Diagram* Optimisasi Sistem Pencarian

3.1.6.4 *Seqeunce Diagram*

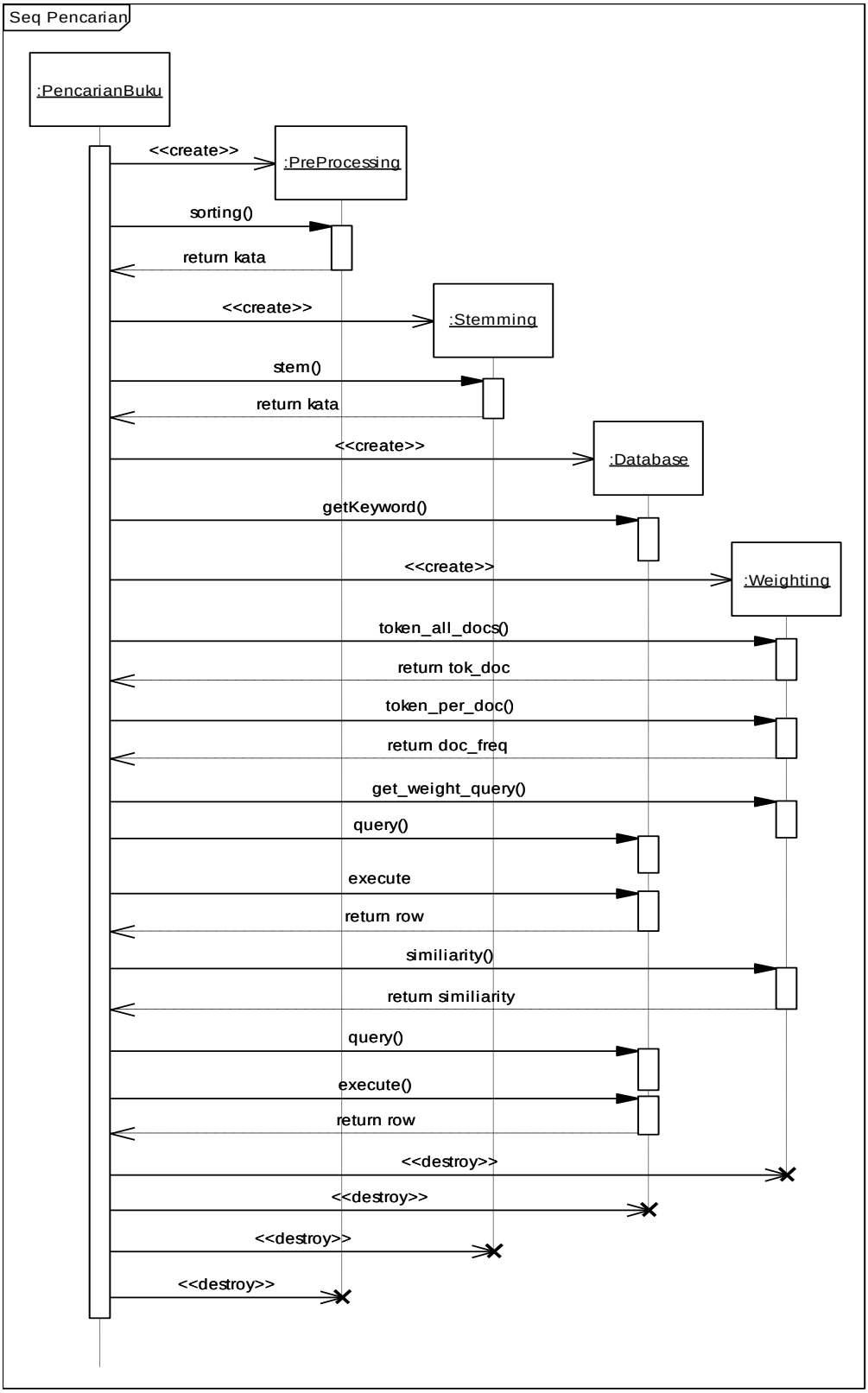
A. Pencarian Data Buku

Tabel 3.12 Skenario Pencarian Data Buku

Nama Use Case	Pencarian Data Buku
Aktor	Pengunjung
Deskripsi	Proses pencarian data buku berdasarkan *query* (representasi kebutuhan pengguna terhadap informasi buku yang dibutuhkan yang sesuai dengan kebutuhan).
Langkah-langkah	**Informasi**
1. Pengunjung memasukkan *query* dan menekan tombol cari atau tekan *enter*.	Form Halaman Web Pencarian
2. Sistem membaca *query* dan melakukan proses *tokenizing*,	Halaman Web Pencarian, *Query*

filtering dan *stemming* pada *query*.	
3. Sistem membaca *keyword*.	Halaman Web Pencarian, *Keyword*
4. Sistem menghitung *similiarity* per dokumen	Halaman Web Pencarian, Bobot *Keyword*, Bobot *Query*
5. Menampilkan maksimal 10 informasi buku dengan nilai *similiarity* terbesar.	Halaman Web Pencarian, Data Buku
Kondisi Awal	Informasi buku tidak ditampilkan.
Kondisi Akhir	Menampilkan informasi buku yang relevan terhadap kebutuhan pengunjung.
Asumsi	Terdapat informasi buku yang sesuai dengan *query* yang dimasukkan oleh pengunjung.

Pencarian data buku merupakan proses yang dilakukan sistem agar pengunjung mendapatkan informasi buku yang dibutuhkan. Interaksi antar kelas yang terjadi saat proses pencarian data buku adalah sebagai berikut :



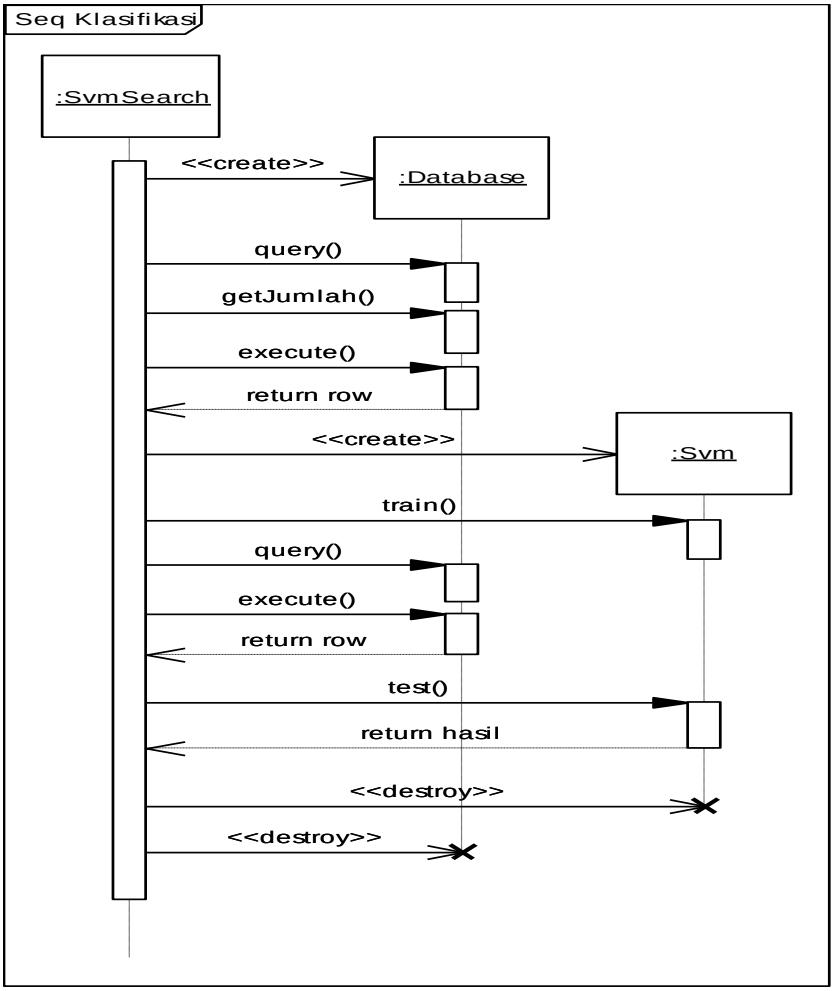
Gambar 3.18 *Sequence Diagram* Pencarian Data Buku

B. Pencarian Data Buku Lebih Lanjut

Tabel 3.13 Pencarian Data Buku Lebih Lanjut

Nama Use Case	Pencarian Data Buku Lebih Lanjut	
Aktor	Pengunjung	
Deskripsi	Proses mengklasifikasi ulang informasi buku yang diperoleh pada proses pencarian data buku untuk mendapatkan informasi buku yang lebih relevan.	
Langkah-langkah		**Informasi**
1. Memilih beberapa dokumen yang relevan menurut pengunjung		Halaman Web Pencarian, Data Duku
2. Sistem melakukan cek terhadap hasil pilihan pengunjung. Jika pengunjung menilai seluruh dokumen sebagai dokumen relevan atau tidak relevan maka sistem akan menampilkan pesan kesalahan.		Halaman Web Pencarian, Data Buku, Pesan Kesalahan
3. Sistem membuat data latih.		Halaman Klasifikasi, Data Latih
4. Sistem melakukan proses pembelajaran.		Halaman Klasifikasi, Data Latih, Model File
5. Sistem akan menguji seluruh data tes.		Halaman Klasifikasi, Data Tes, Model File
6. Menampilkan dokumen yang memiliki hasil uji positif.		Halaman Klasifikasi, Hasil Tes, Data Buku
Kondisi Awal	Data buku hasil klasifikasi belum ditampilkan.	
Kondisi Akhir	Menampilkan seluruh data buku yang memiliki hasil uji positif.	
Asumsi	Terdapat beberapa data buku yang memiliki hasil uji positif.	

Klasifikasi merupakan proses yang dilakukan sistem agar pengunjung mendapatkan hasil pencarian yang lebih akurat menurut pengunjung. Interaksi antar kelas yang terjadi saat proses klasifikasi adalah sebagai berikut :



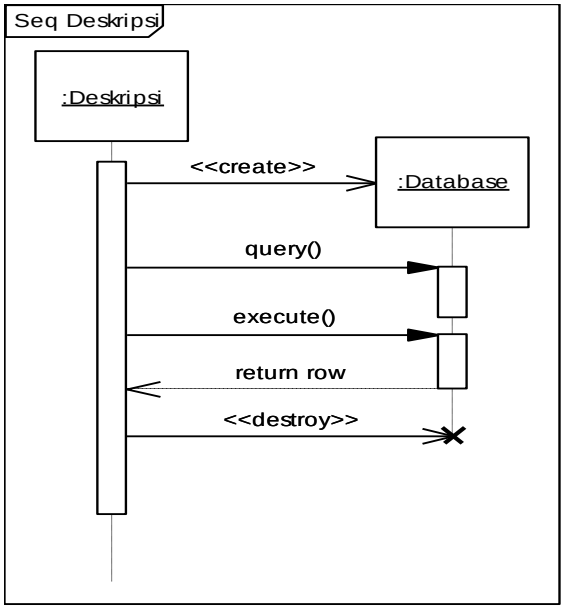
Gambar 3.19 *Sequence Diagram* Klasifikasi

C. Penyajian Deskripsi

Tabel 3.14 Skenario Penyajian Deksripsi

Nama Use Case	Penyajian Deskripsi	
Aktor	Pengunjung	
Deskripsi	Proses untuk melihat deskripsi dari informasi buku.	
Langkah-langkah		**Informasi**
1. Pengunjung memilih atau meng-*click* salah satu judul buku.		Halaman Web Pencarian
2. Sistem membaca data buku		Halaman Web Pencarian , Data Buku
3. Sistem menampilkan deksripsi buku.		Halaman Deskripsi, Data Buku
Kondisi Awal	Deskripsi buku, kosong.	
Kondisi Akhir	Sistem menampilkan deskripsi buku dari salah satu judul buku yang dipilih pengunjung.	
Asumsi	Pengunjung : Sudah melakukan proses pencarian.	

Menampilkan deskripsi buku merupakan proses yang dilakukan sistem agar pengunjung dapat melihat deksripsi dari informasi buku hasil proses pencarian data buku atau pencarian data buku lebih lanjut. Interaksi antar kelas yang terjadi saat proses lihat deksripsi buku adalah sebagai berikut :



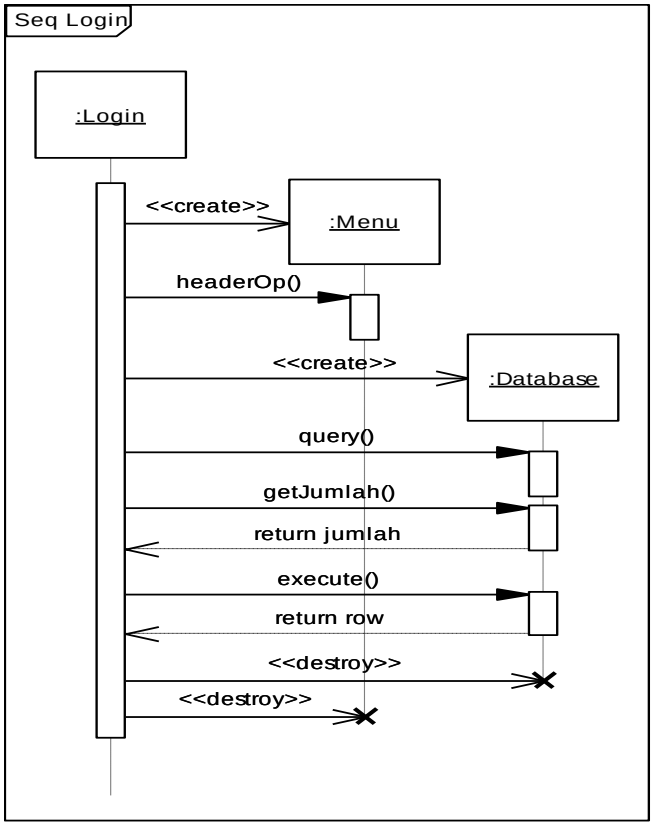
Gambar 3.20 *Sequence Diagram* Penyajian Deskripsi

D. Login

Tabel 3.15 Skenario Login

Nama Use Case	Login	
Aktor	Operator	
Deskripsi	Proses untuk mendapatkan hak akses untuk melakukan pengolahan data.	
Langkah-langkah		**Informasi**
1. Operator mengisi nama user dan kata kunci serta menekan tombol masuk.		Halaman Login
2. Sistem menampilkan halaman web utama operator.		Halaman Web Utama Operator, Data Nama User, Data Kata Kunci
Kondisi Awal	Data buku belum ditampilkan.	
Kondisi Akhir	Menampilkan seluruh data buku.	
Asumsi	Data nama user dan kata kunci terdaftar dalam *database*.	

Login merupakan proses verifikasi sistem terhadap akun operator agar operator mendapatkan hak akses melakukan pengolahan data. Interaksi antar kelas yang terjadi saat proses login adalah sebagai berikut :



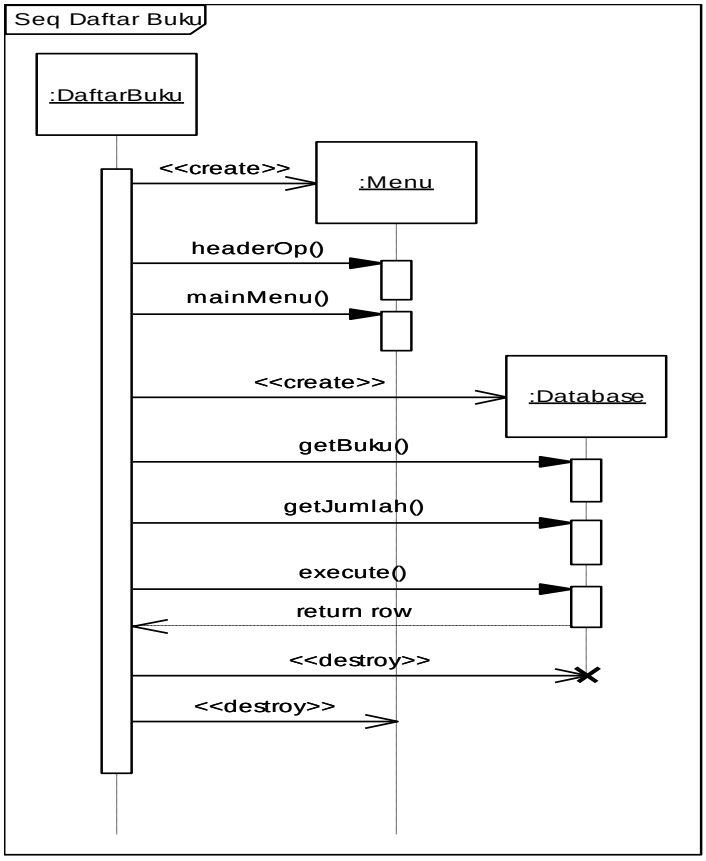
Gambar 3.21 *Sequence Diagram* Login

E. Menampilkan Daftar Buku

Tabel 3.16 Skenario Lihat Data Buku

Nama Use Case	Lihat Info Buku
Aktor	Operator
Deskripsi	Proses untuk melihat seluruh informasi buku yang terdaftar di perpustakaan.
Langkah-langkah	**Informasi**
1. Operator memilih menu buku.	Halaman Web Utama Operator
2. Sistem menampilkan seluruh informasi buku.	Halaman Web Buku, Data Buku
Kondisi Awal	Data buku belum ditampilkan.
Kondisi Akhir	Menampilkan seluruh data buku.
Asumsi	Sudah ada data buku dalam *database*.

Menampilkan daftar buku merupakan proses yang diperankan oleh operator sebelum dapat melakukan proses tambah atau ubah data buku. Interaksi antar kelas yang terjadi saat proses menampilkan daftar buku adalah sebagai berikut :



Gambar 3.22 *Sequence Diagram* Menampilkan Daftar Buku

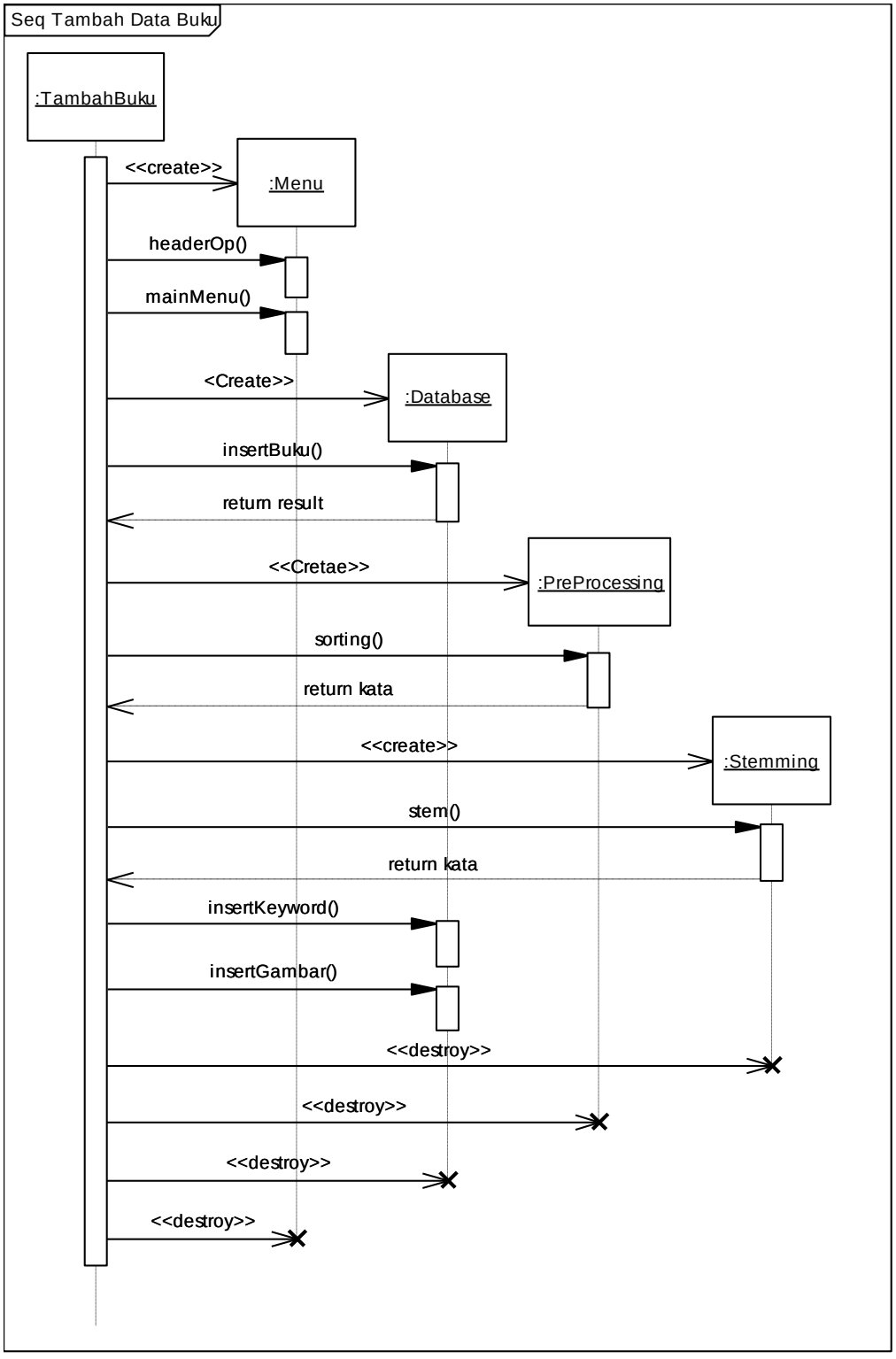
F. Tambah Data Buku

Tabel 3.17 Skenario Tambah Data Buku

Nama Use Case	Tambah Data Buku
Aktor	Operator
Deskripsi	Proses untuk menambah data buku dan membuat *keyword*.
Langkah-langkah	**Informasi**
1. Operator mengisi data buku dan menekan tombol simpan.	Form Penambahan Data Buku
2. Sistem mem-validasi data buku	Form Penambahan Data Buku, Data

	yang telah dimasukkan.	Buku
3.	Sistem menyimpan data buku ke *database*.	Form Penambahan Data Buku, Data Buku
4.	Sistem membuat *keyword* dari data buku tersebut.	Form Penambahan Data Buku, Data Buku
5.	Halaman konfirmasi dikirim ke operator.	Halaman Konfirmasi
Kondisi Awal	Data buku belum disimpan ke *database*.	
Kondisi Akhir	Sistem menyimpan data buku ke *database* dan membuat *keyword* dari data buku yang baru disimpan.	
Asumsi	1. Sudah melakukan proses lihat data buku. 2. Berhasil menyimpan data buku ke *database.*	

Tambah data buku merupakan proses yang diperankan operator, bertujuan untuk menambah data buku serta membuat *keyword* dari data buku yang berhasil dimasukkan ke *database*. Interaksi antar kelas yang terjadi saat proses tambah data buku adalah sebagai berikut :



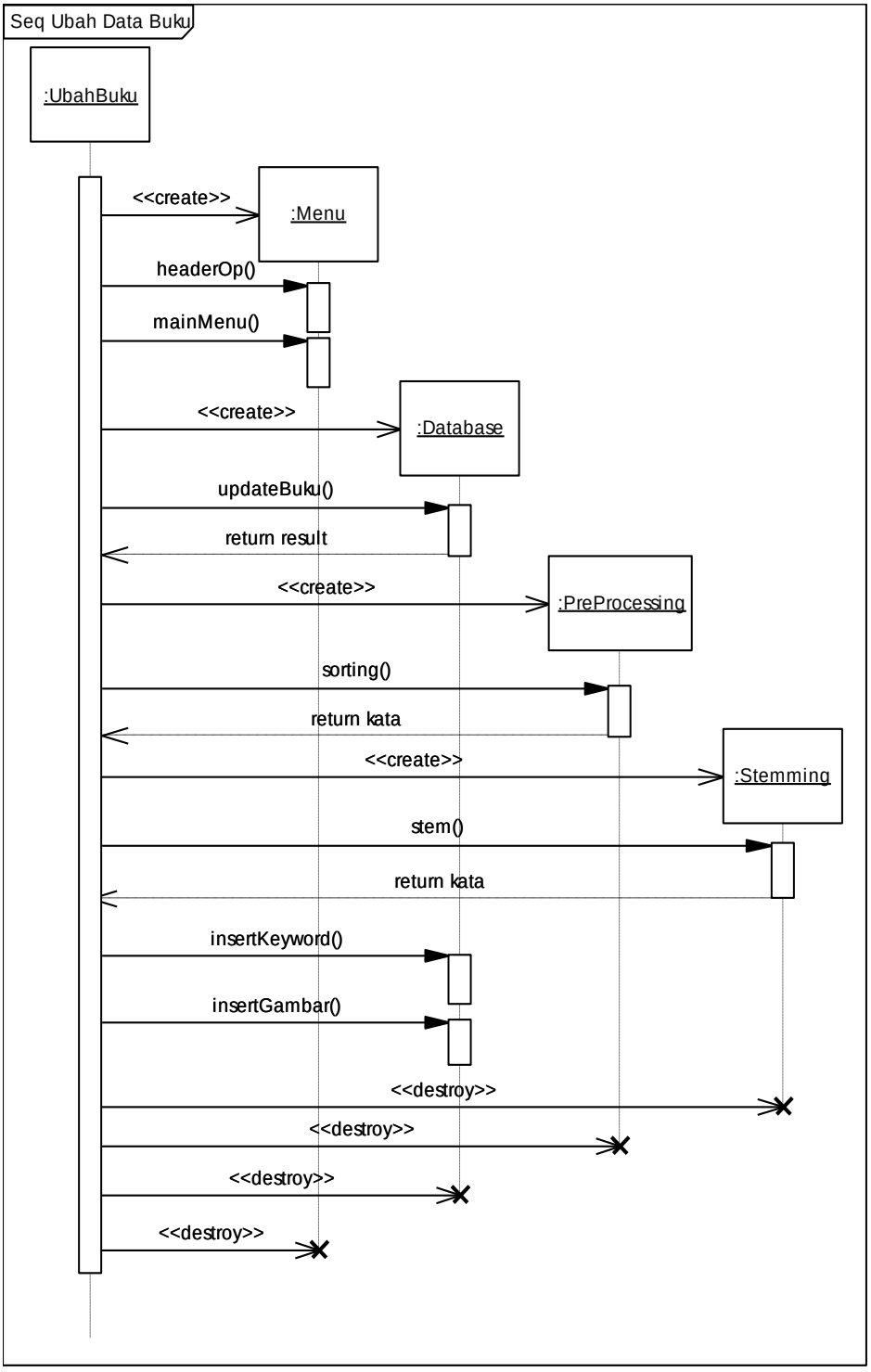
Gambar 3.23 *Sequence Diagram* Tambah Data Buku

G. Ubah Data Buku

Tabel 3.18 Skenario Ubah Data Buku

Nama Use Case	Ubah Data Buku	
Aktor	Operator	
Deskripsi	Proses untuk mengubah data buku dan mengubah *keyword*.	
Langkah-langkah		**Informasi**
1. Operator mengubah data buku dan menekan tombol simpan.		Form Pengubahan Data Buku
2. Sistem mem-validasi data buku yang telah dimasukkan.		Form Pengubahan Data Buku, Data Buku
3. Sistem menyimpan data buku ke *database*.		Form Pengubahan Data Buku, Data Buku
4. Sistem mengubah *keyword* dari data buku tersebut.		Form Pengubahan Data Buku, Data Buku
5. Halaman konfirmasi dikirim ke operator.		Halaman Konfirmasi
Kondisi Awal	Menampilkan data buku sebelum diubah.	
Kondisi Akhir	Sistem menyimpan data buku.	
Asumsi	1. Sudah melakukan proses lihat data buku. 2. Berhasil menyimpan data buku ke *database.*	

Ubah data buku merupakan proses yang diperankan oleh operator. Bertujuan untuk mengubah data buku serta mengubah *keyword* dari data buku yang ada di *database*. Interaksi antar kelas yang terjadi saat proses tambah buku adalah sebagai berikut :



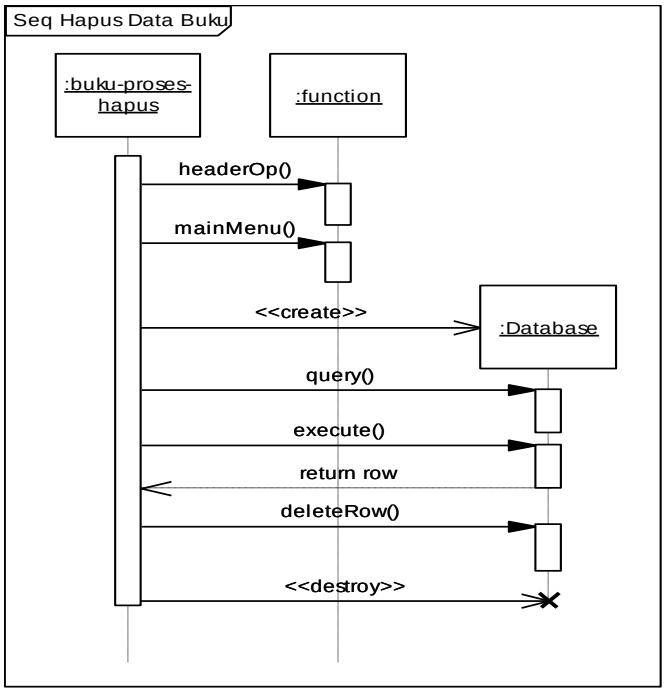
Gambar 3.24 *Sequence Diagram* Ubah Data Buku

H. Hapus Data Buku

Tabel 3.19 Skenario Hapus Data Buku

Nama Use Case	Hapus Data Buku	
Aktor	Operator	
Deskripsi	Proses untuk menghapus data buku yang ada di *database* secara permanen.	
Langkah-langkah		**Informasi**
1. Operator memilih data buku yang akan dihapus.		Halaman Web Data Buku
2. Sistem menghapus data buku dati *database*.		Halaman Web Data Buku, Data Buku
3. Sistem menghapus data tes dari data buku tersebut.		Halaman Web Data Buku, Data Tes
Kondisi Awal	Data buku belum dihapus.	
Kondisi Akhir	Data buku dihapus dari *database*.	
Asumsi	1. Sudah melakukan proses lihat data buku. 2. Berhasil menghapus data buku ke *database.*	

Hapus data buku merupakan proses yang diperankan oleh operator bertujuan untuk menghapus data buku dari database serta *keyword* dan data tes dari data buku tersebut. Interaksi antar kelas yang terjadi saat proses hapus data buku adalah sebagai berikut :



Gambar 3.25 *Sequence Diagram* Hapus Data Buku

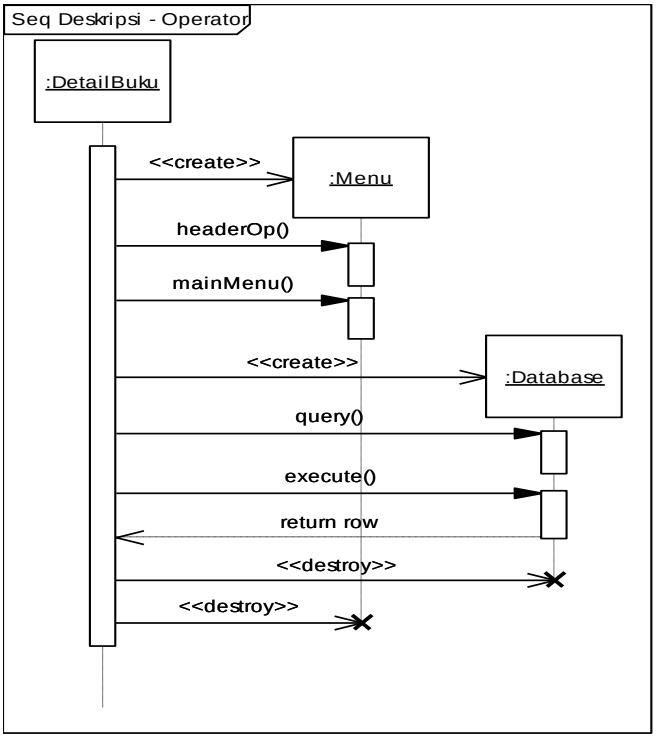
I. Penyajian Detail Data Buku

Tabel 3.20 Skenario Penyajian Detail Data Buku

Nama Use Case	Lihat Detail Data Buku	
Aktor	Operator	
Deskripsi	Proses untuk melihat detail dari data buku.	
Langkah-langkah		**Informasi**
1. Pilih atau klik judul buku.		Halaman Web Buku
2. Sistem menampilkan detail dari data buku.		Halaman Detail, Detail Data Buku
Kondisi Awal	Deskripsi buku, kosong.	
Kondisi Akhir	Sistem menampilkan deskripsi buku dari salah satu judul buku.	
Asumsi	Sudah melakukan proses lihat info buku	

Menampilkan deskripsi buku merupakan proses yang dilakukan sistem agar operator dapat melihat deksripsi dari buku. deskripsi yang dimaksud berupa judul, pengarang, deskripsi, *keywrod* dan nama file data tes. Interaksi antar kelas yang terjadi saat proses menampilkan deskripsi buku adalah sebagai berikut :

1. Metode headerOp( ) dan mainMenu dari kelas Menu digunakan sebagai template antarmuka.
2. Melalui metode query( ) dan execute( ) dari kelas Database, sistem membaca *field- field* buku yang dibutuhkan untuk ditampilkan ke operator.



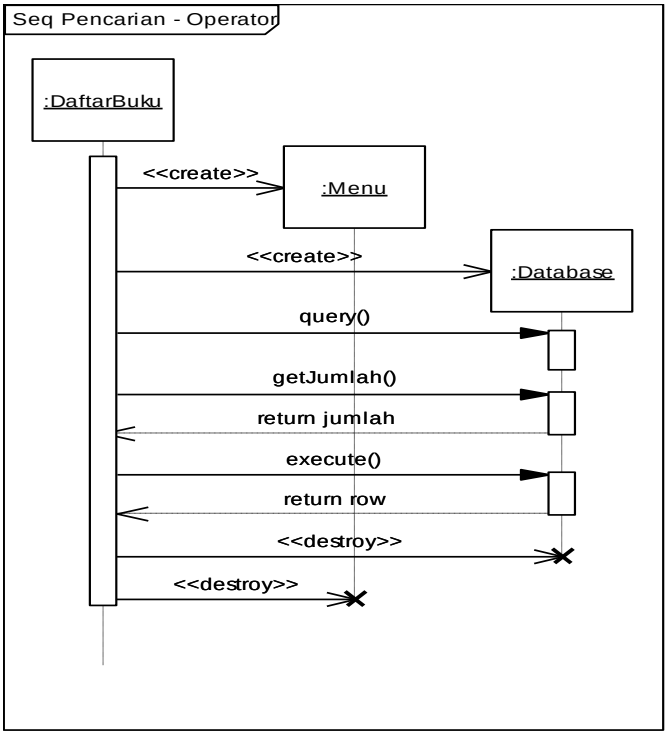
Gambar 3.26 *Sequence Diagram* Penyajian Deskripsi

J. Pencarian Data Buku Berdasarkan Judul

Tabel 3.21 Pencarian Data Buku Berdasarkan Judul

Nama Use Case	Pencarian Berdasarkan Judul Buku	
Aktor	Operator	
Deskripsi	Proses untuk mencari informasi buku dengan memasukkan judul buku.	
Langkah-langkah		**Informasi**
1. Operator memasukkan judul buku dan menekan tombol cari.		Halaman Web Utama Operator
2. Sistem menampilkan informasi buku.		Halaman Web Buku, Data Buku
Kondisi Awal	Informasi buku ditampilkan lengkap.	
Kondisi Akhir	Menampilkan Informasi buku yang sesuai dengan judul.	
Asumsi	Informasi buku yang dicari terdapat di *database*.	

Pencarian data buku merupakan proses yang dilakukan sistem agar operator mendapatkan informasi buku yang dibutuhkan. Pencarian ini hanya pencarian berdasarkan judul biku. Interaksi antar kelas yang terjadi saat proses pencarian data buku adalah sebagai berikut :



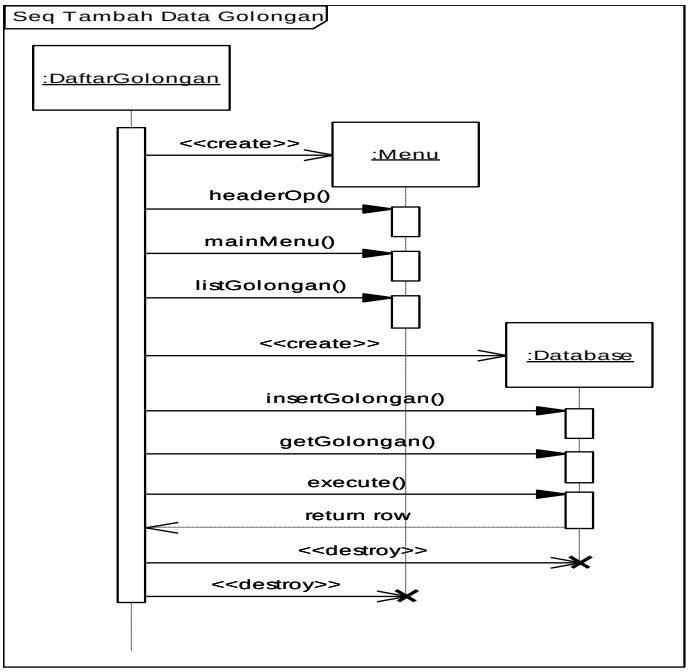
Gambar 3.27 *Sequence Diagram Pencarian* Berdasarkan Judul Buku

K. Tambah Data Golongan

Tabel 3.22 *Sequence Diagram* Tambah Data Golongan

Nama Use Case	Tambah Data Golongan	
Aktor	Operator	
Deskripsi	Proses untuk menambah data golongan.	
Langkah-langkah		**Informasi**
1. Operator mengisi data golongan dan menekan tombol tambah.		Halaman Web Golongan,
2. Sistem mem-validasi data golongan yang telah dimasukkan.		Halaman Web Golongan, Data Golongan
3. Sistem menyimpan data golongan ke *database*.		Halaman Web Golongan, Data Golongan
4. Sistem menampilkan seluruh informasi golongan.		Halaman Web Golongan, Data Golongan
Kondisi Awal	Data golongan belum disimpan ke *database*.	
Kondisi Akhir	Sistem menyimpan data golongan ke *database*.	
Asumsi	Sudah melakukan proses lihat info golongan dan berhasil menyimpan data golongan ke *database.*	

Tambah data golongan merupakan proses yang dilakukan oleh operator. Interaksi antar kelas yang terjadi saat proses tambah data golongan adalah sebagai berikut :



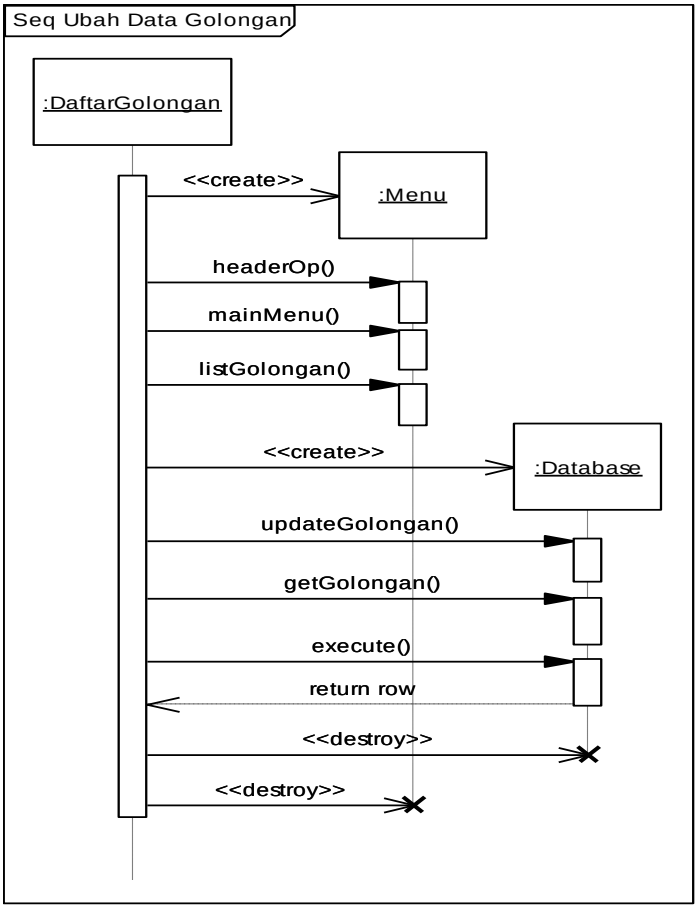
Gambar 3.28 *Sequence Diagram* Tambah Data Golongan

L. Ubah Data Golongan

Tabel 3.23 Skenario Ubah Data Golongan

Nama Use Case	Tambah Data Golongan
Aktor	Operator
Deskripsi	Proses untuk menambah data golongan.
Langkah-langkah	**Informasi**
5. Operator mengubah data golongan dan menekan tombol ubah.	Halaman Web Golongan,
6. Sistem mem-validasi data golongan yang telah dimasukkan.	Halaman Web Golongan, Data Golongan
7. Sistem menyimpan data golongan ke *database*.	Halaman Web Golongan, Data Golongan
8. Sistem menampilkan seluruh informasi golongan.	Halaman Web Golongan, Data Golongan
Kondisi Awal	Data golongan belum dubah dan disimpan ke *database*.
Kondisi Akhir	Sistem menyimpan data golongan ke *database*.
Asumsi	Sudah melakukan proses lihat info golongan dan berhasil menyimpan data golongan ke *database.*

Ubah data golongan merupakan proses yang dilakukan oleh operator untuk mengubah data golongan. Interaksi antar kelas yang terjadi saat proses ubah data golongan adalah sebagai berikut :



Gambar 3. 29 *Sequence Diagram* Ubah Data Golongan

M. Optimisasi Sistem Pencarian

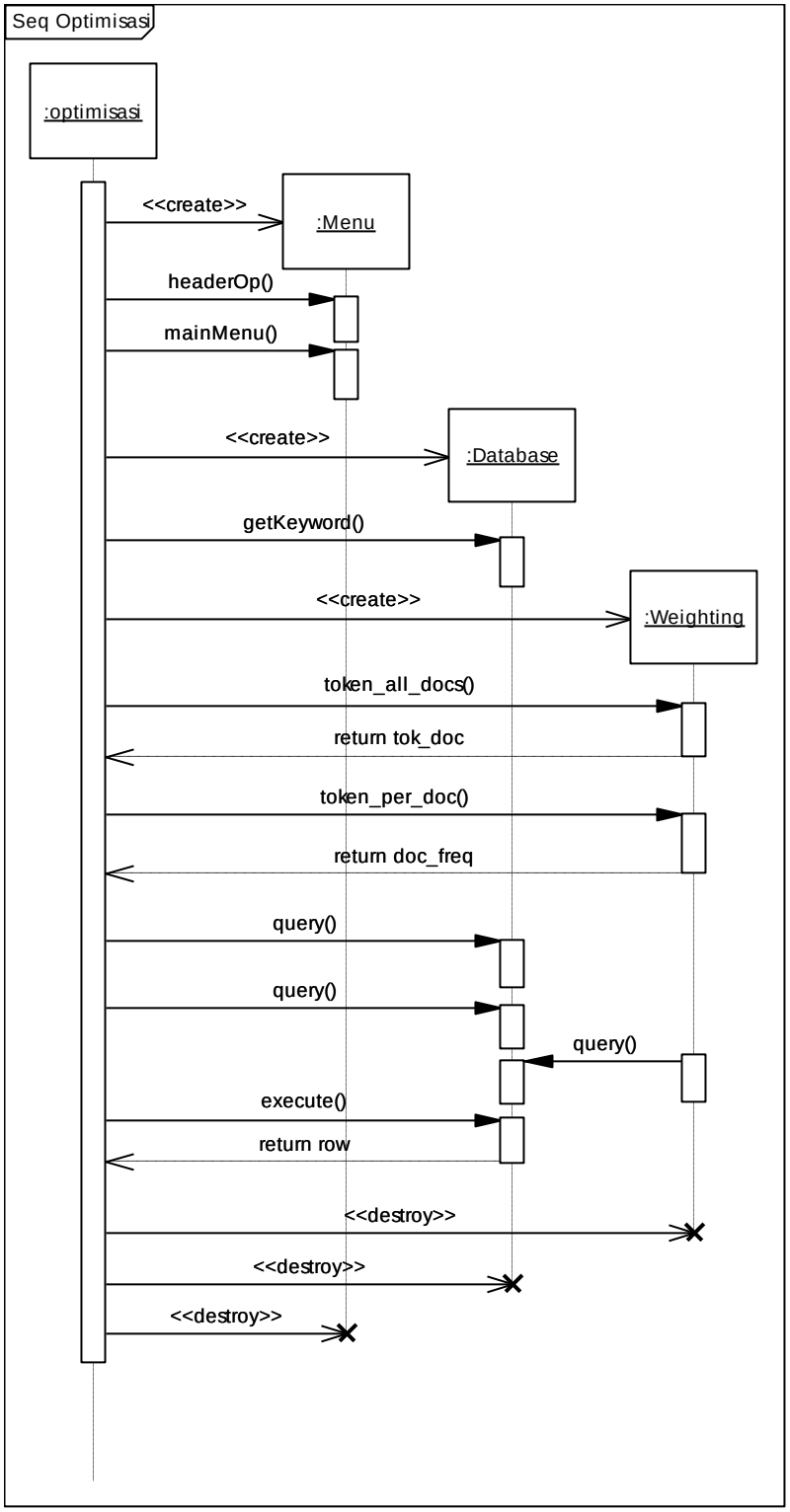
Tabel 3.24 Skenario Optimisasi Sistem Pencarian

Nama Use Case	Optimisasi Sistem Pencarian
Aktor	Operator
Deskripsi	Proses perhitungan bobot dengan skema *tf-idf*, setelah itu membangun *vector space model* kemudian membuat data tes untuk klasifikasi *support vector machines*.

Langkah-langkah	**Informasi**
1. Operator menekan tombol optimisasi.	Halaman Web Utama Operator
2. Sistem menghitung bobot setiap dokumen dengan skema *tf-idf*.	Halaman Web Utama Operator, *Keyword*

3. Sistem melakukan proses normalisasi.	Halaman Web Utama Operator, Bobot *Keyword*
4. Sistem menyimpan hasil perhitungan normalisasi ke *database*.	Halaman Web Utama Operator, Bobot *Keyword*
5. Sistem membaca hasil perhitungan normalisasi yang tidak sama dengan 0 dari *database*.	Halaman Web Utama Operator, Bobot *Keyword*
6. Sistem membuat data tes untuk setiap data buku	Halaman Web Utama Operator, Bobot *Keyword*
7. Sistem menampilkan pesan hasil optimisasi.	Halaman Web Utama Operator
Kondisi Awal	Belum terdapat hasil normalisasi dan data tes di *database*.
Kondisi Akhir	Hasil perhitungan normalisasi disimpan di *database* dan data tes telah berhasil dibuat.
Asumsi	Sudah ada data buku di *database.*

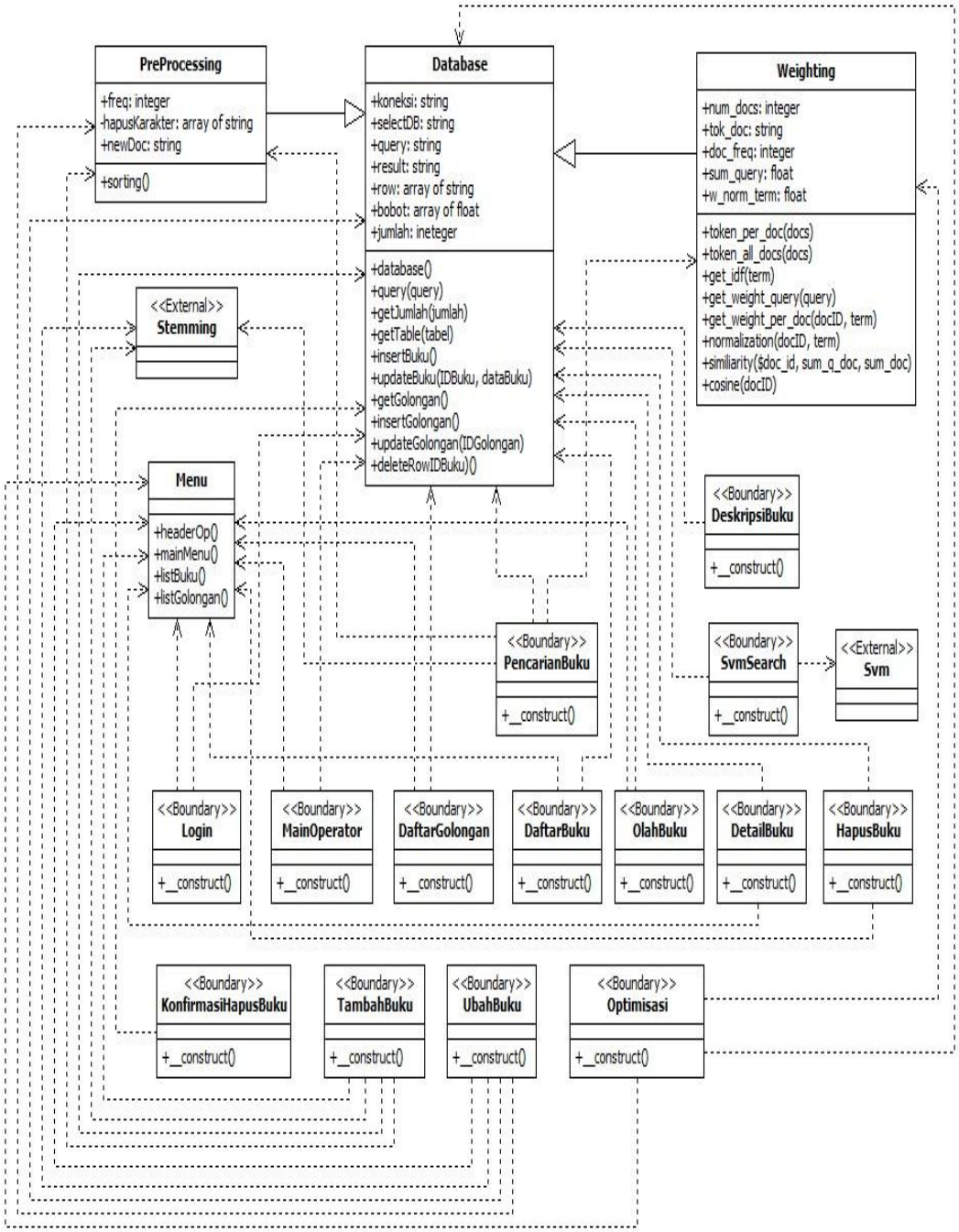
Optimisasi merupakan proses yang diperankan oleh operator dengan tujuan untuk menghitung bobot dari setiap *term keyword* serta menormalisasikannya. Hasil normalisasi yang bernilai tidak sama dengan 0, ditulis ke dalam data tes. Interaksi antar kelas yang terjadi saat proses optimisasi adalah sebagai berikut :



Gambar 3.30 *Sequence Diagram* Optimisasi

3.1.6.5 Class Diagram

Diagram ini ditujukan untuk mendefinisikan proses yang berjalan pada aplikasi klasifikasi ini.

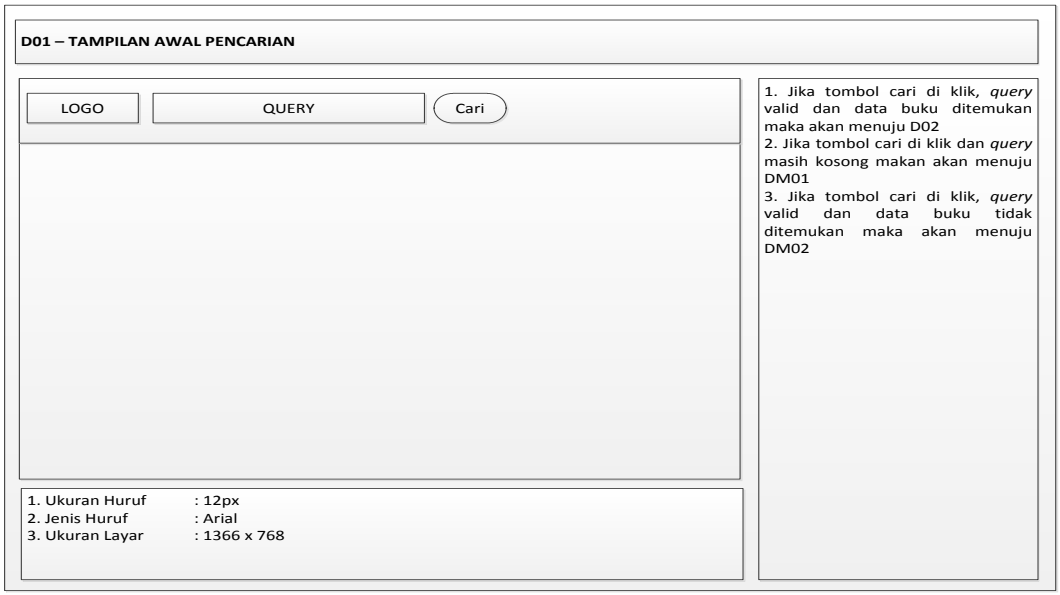


Gambar 3.31 *Class Diagram* Sistem Pencarian Informasi Buku

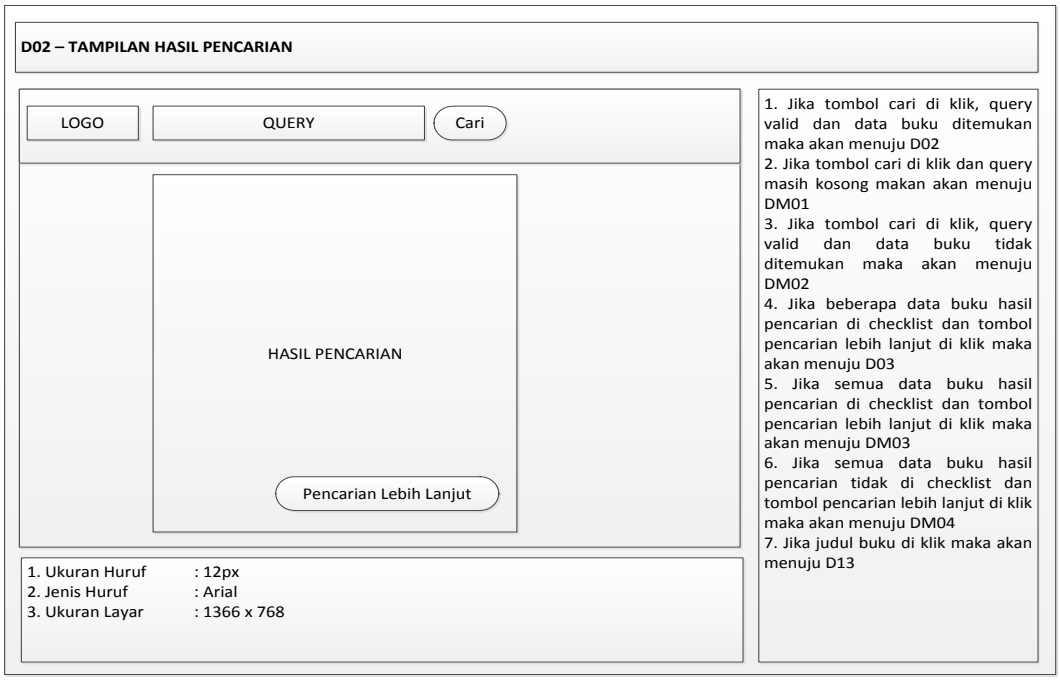
3.2 Perancangan Sistem

3.2.1 Perancangan Antarmuka

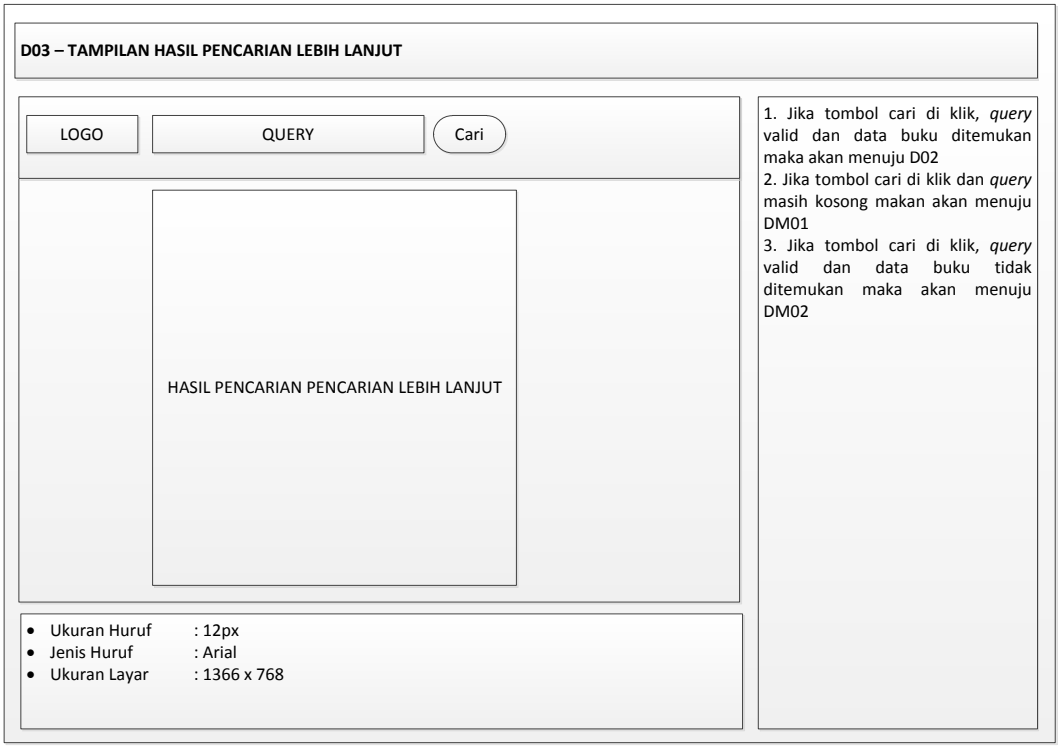
3.2.1.1 Perancangan Antarmuka Pengunjung



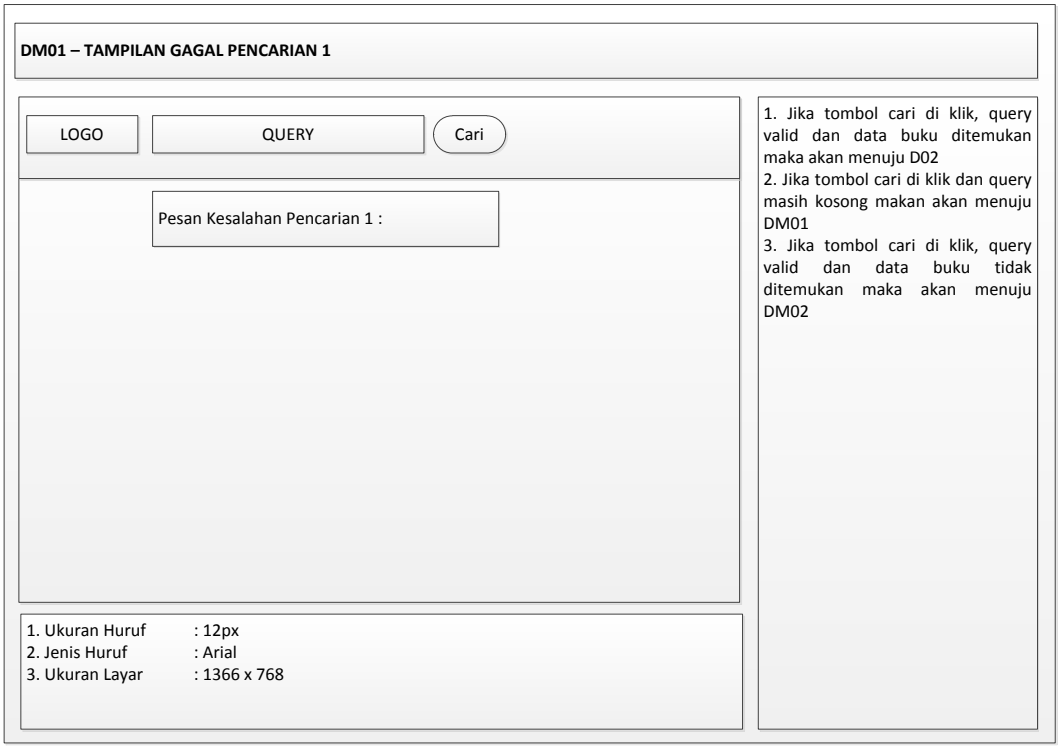
Gambar 3.32 Tampilan Awal Pencarian



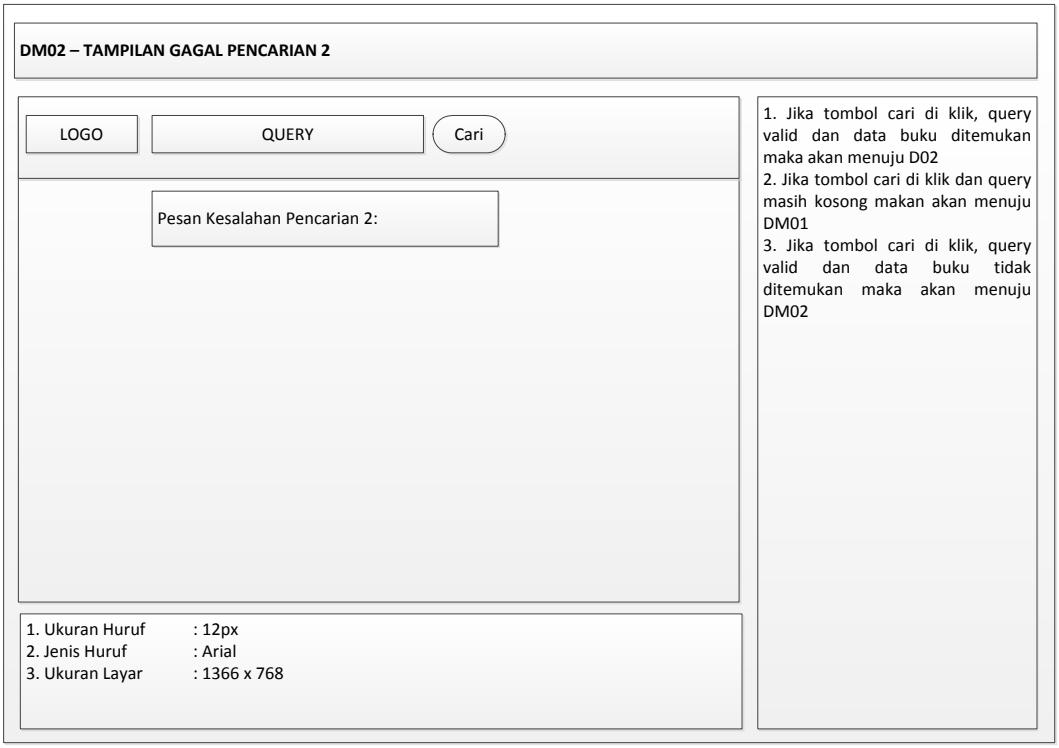
Gambar 3.33 Tampilan Hasil Pencarian



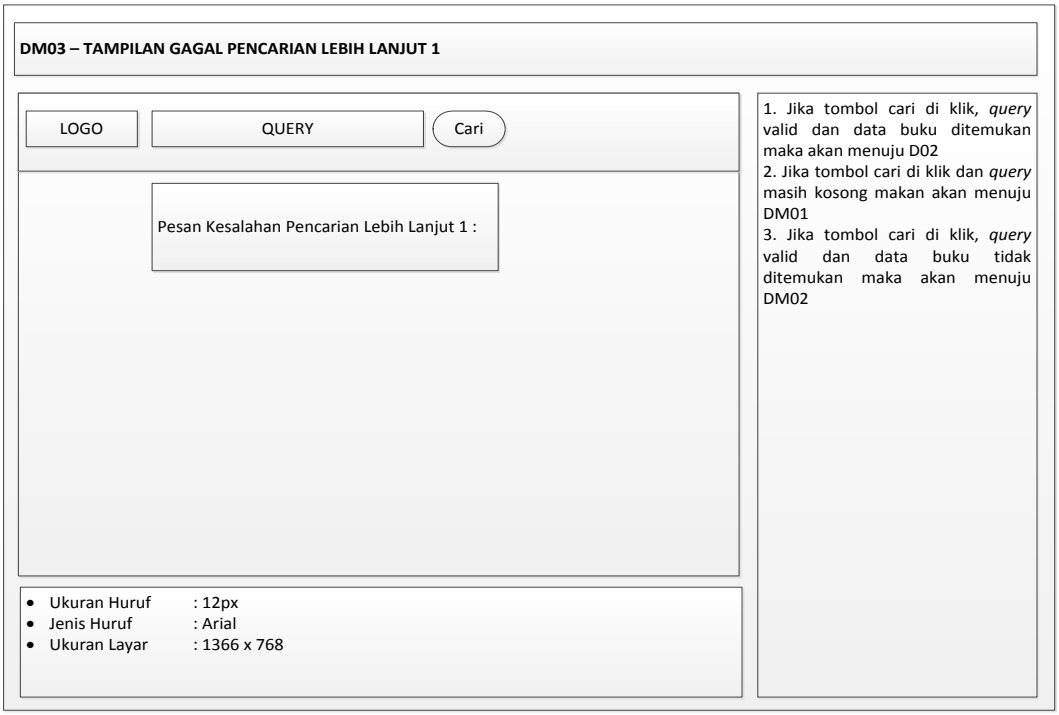
Gambar 3.34 Tampilan Hasil Pencarian Lebih Lanjut



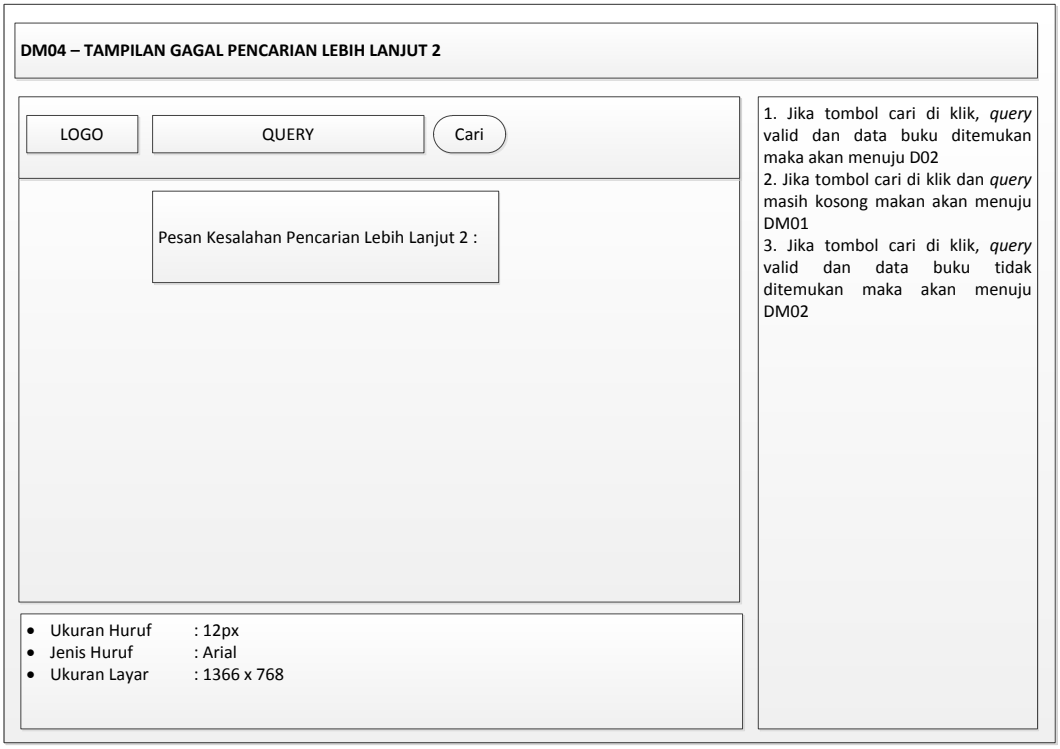
Gambar 3.35 Tampilan Gagal Pencarian 1



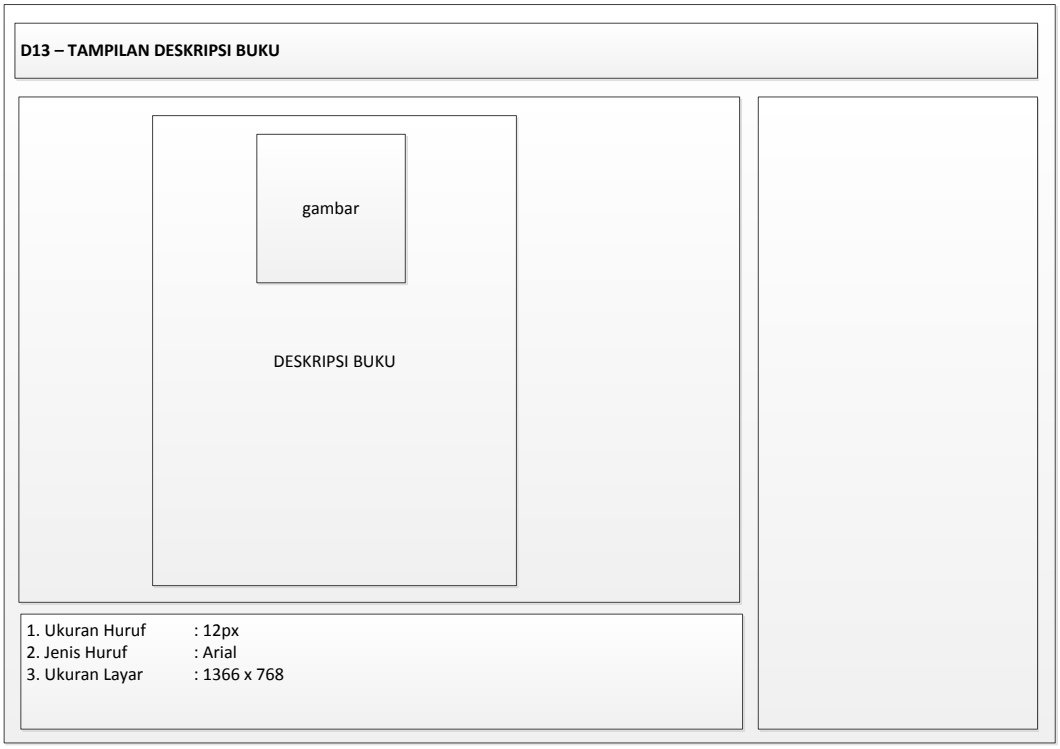
Gambar 3.36 Tampilan Gagal Pencarian 2



Gambar 3.37 Tampilan Gagal Lebih Lanjut 1

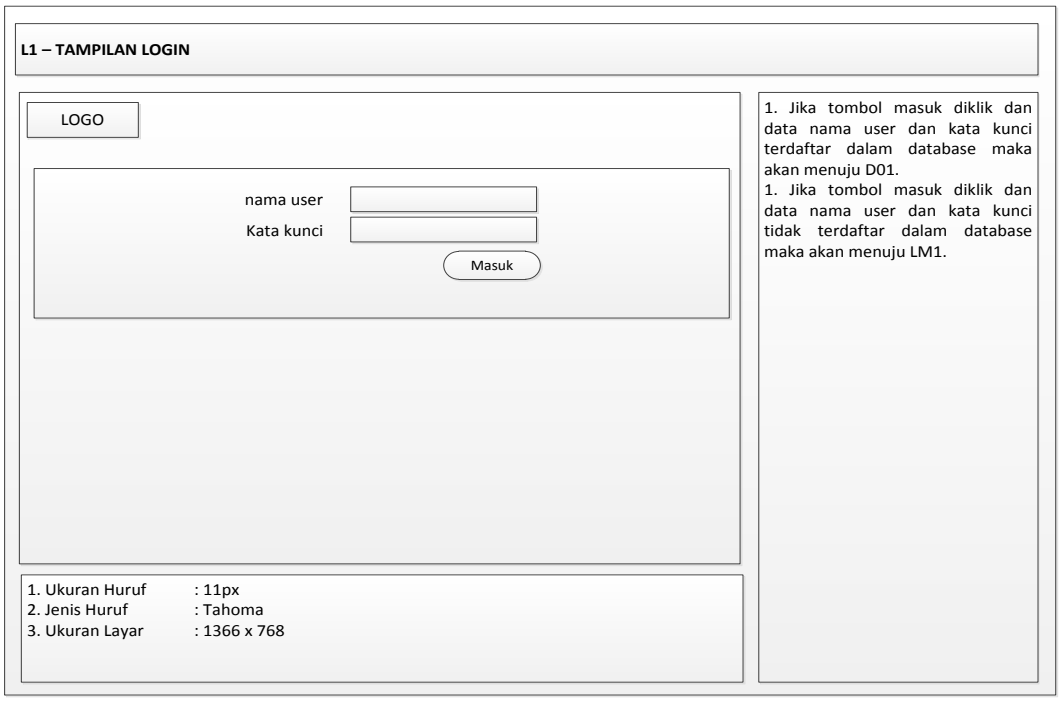


Gambar 3.38 Tampilan Gagal Lebih Lanjut 2

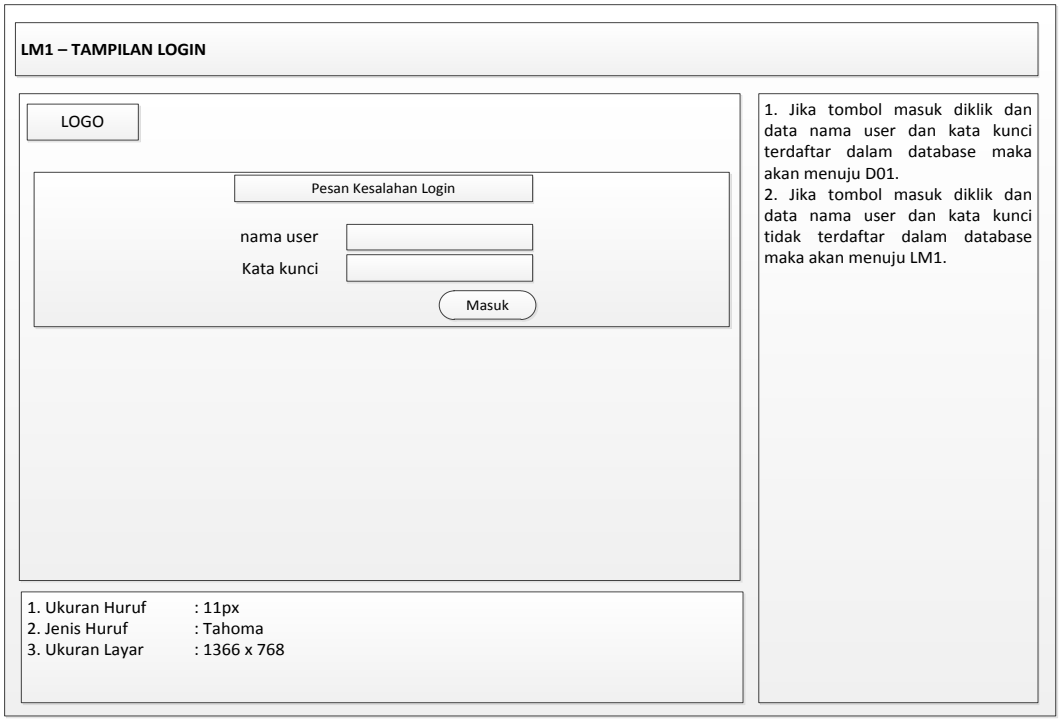


Gambar 3.39 Tampilan Deskripsi Buku

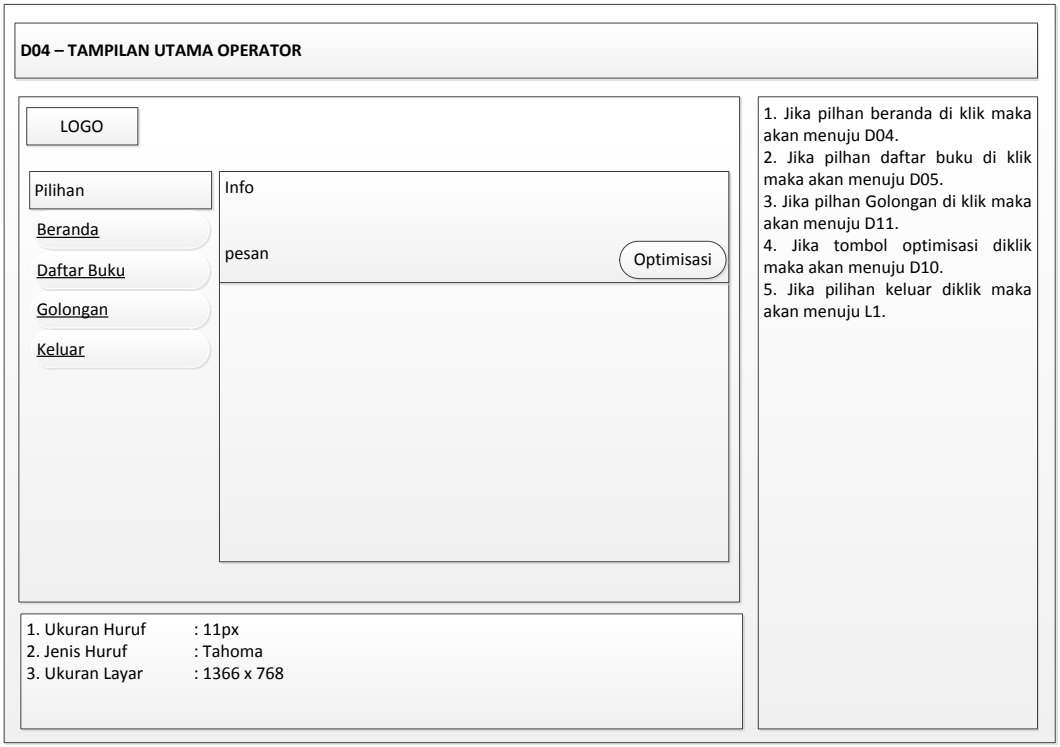
3.2.1.2 Perancangan Antarmuka Operator



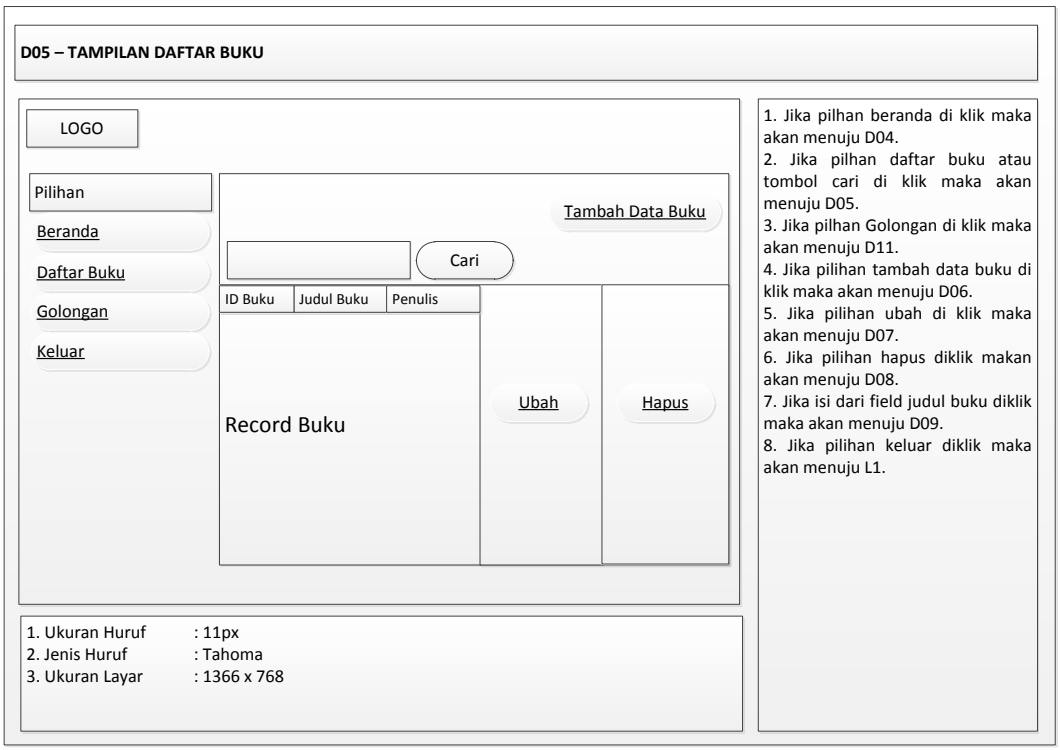
Gambar 3.40 Tampilan Login



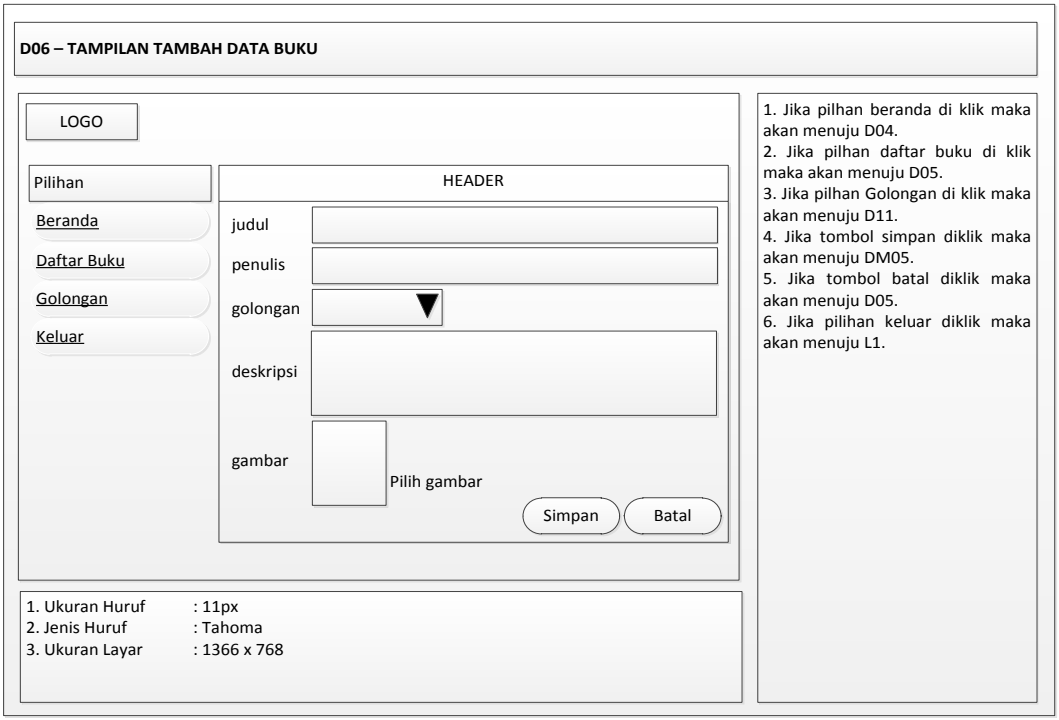
Gambar 3.41 Tampilan Kesalahan Login



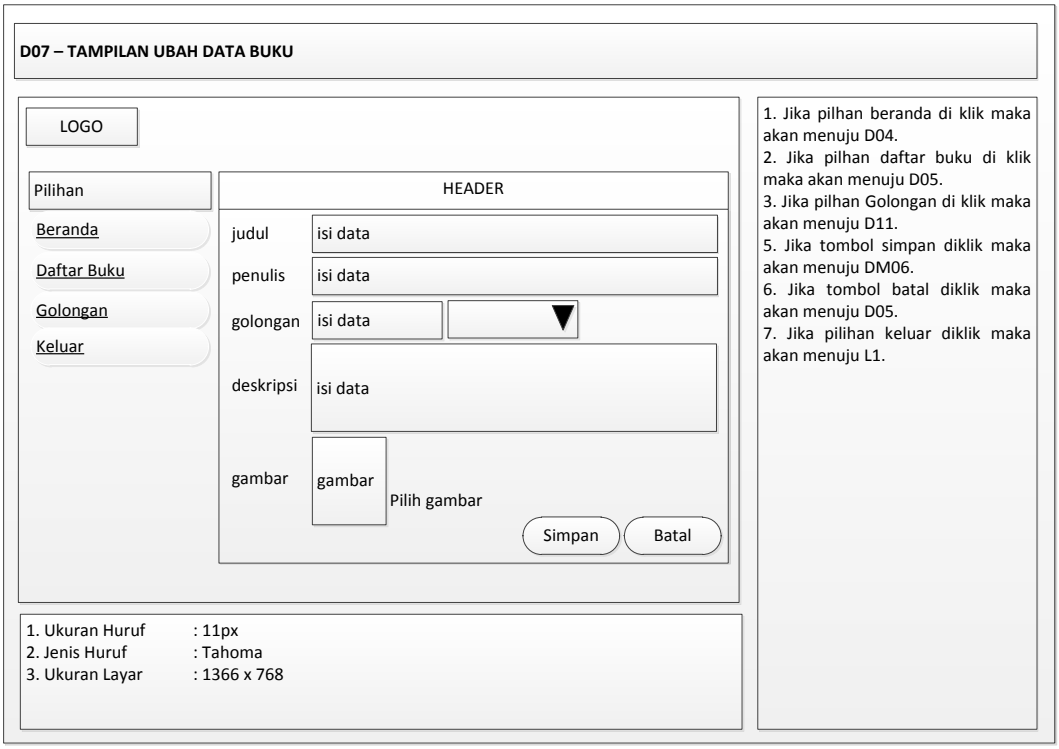
Gambar 3.42 Tampilan Awal Operator



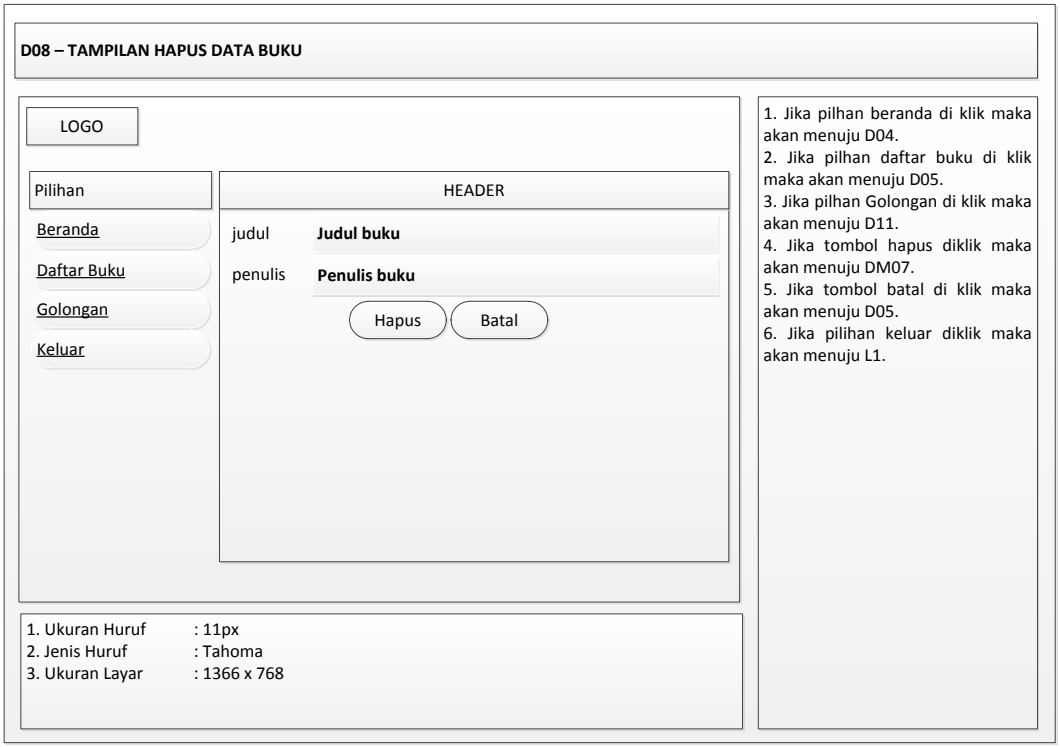
Gambar 3.43 Tampilan Daftar Buku



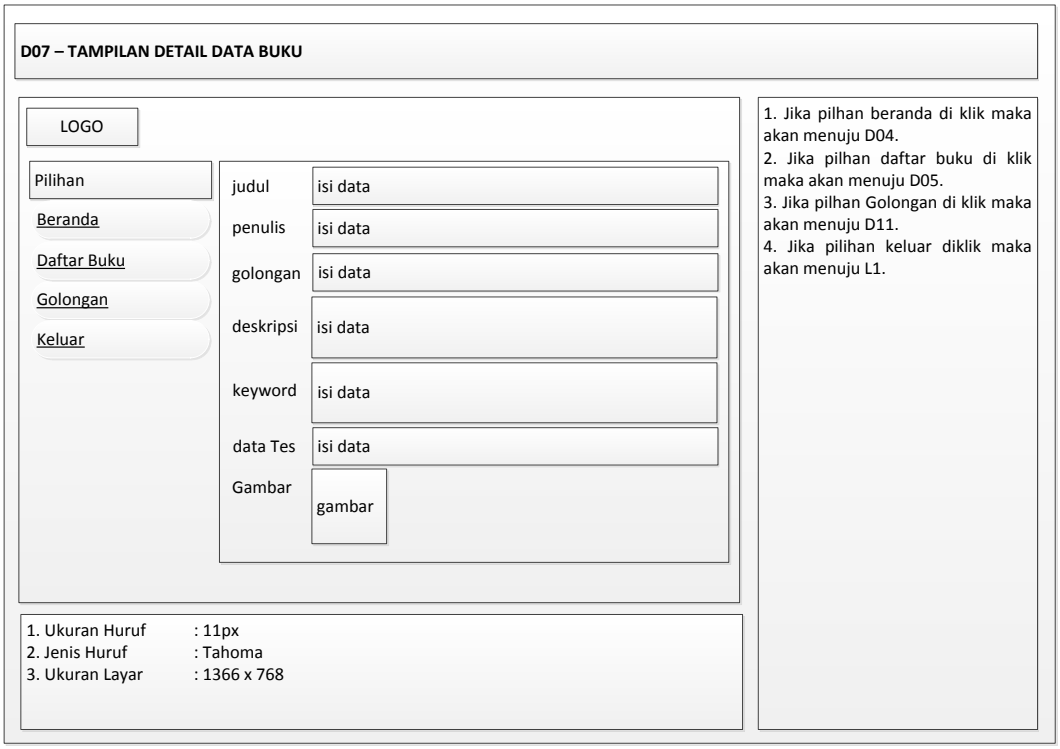
Gambar 3.44 Tampilan Tambah Data Buku



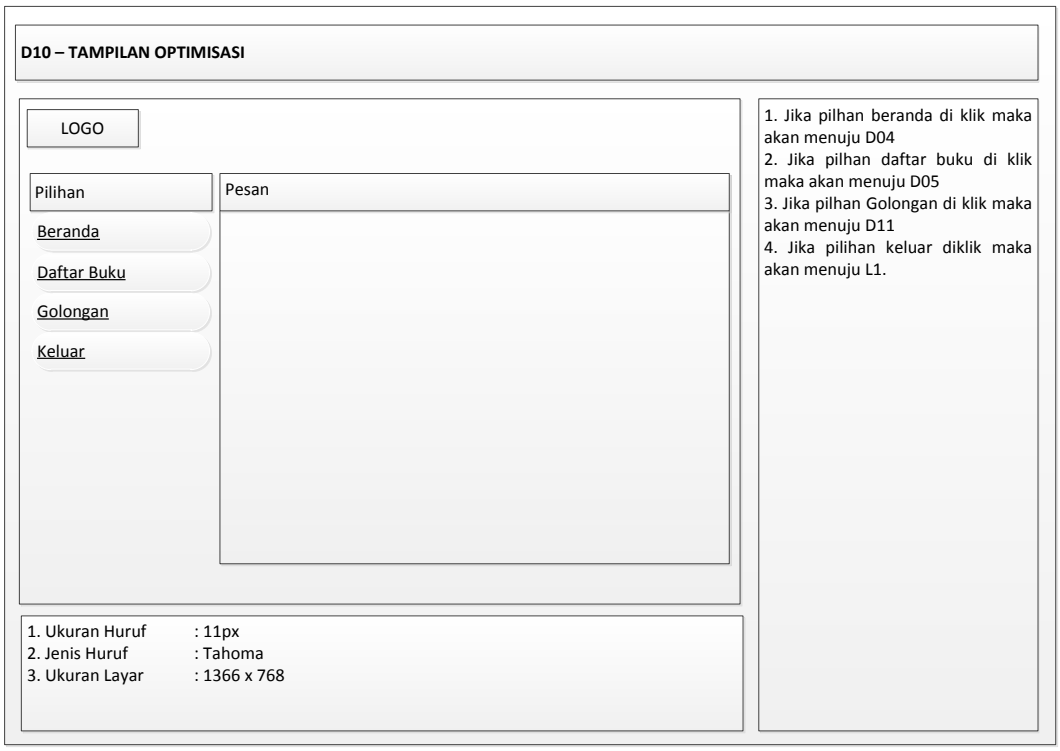
Gambar 3.45 Tampilan Ubah Data Buku



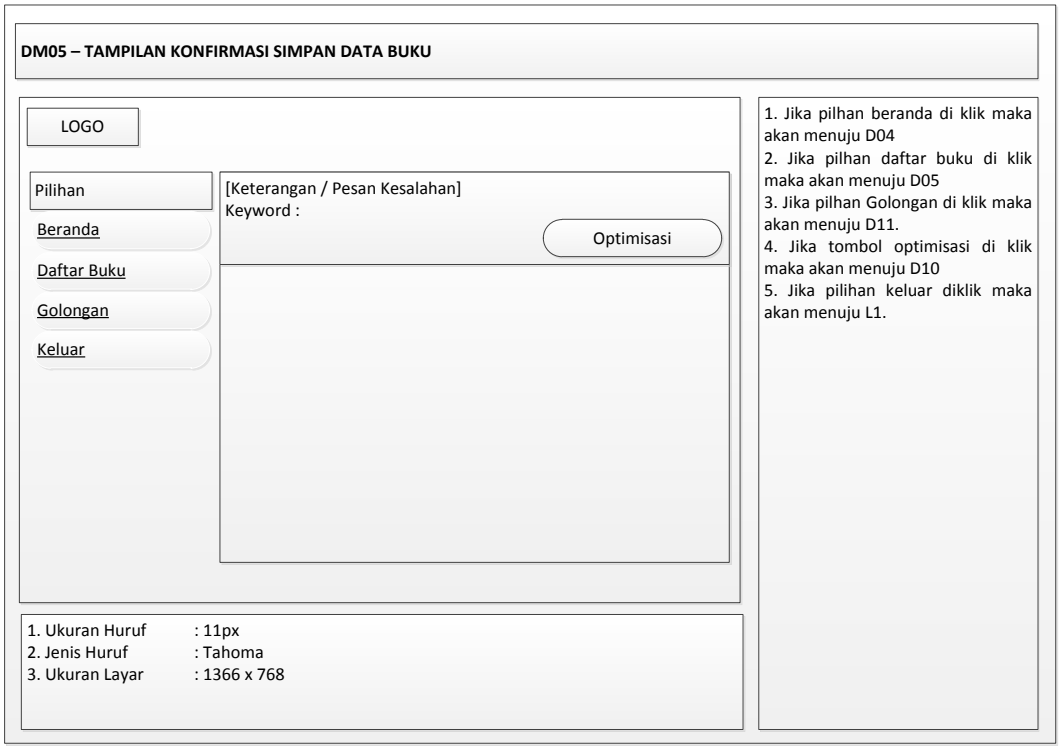
Gambar 3.46 Tampilan Hapus Data Buku



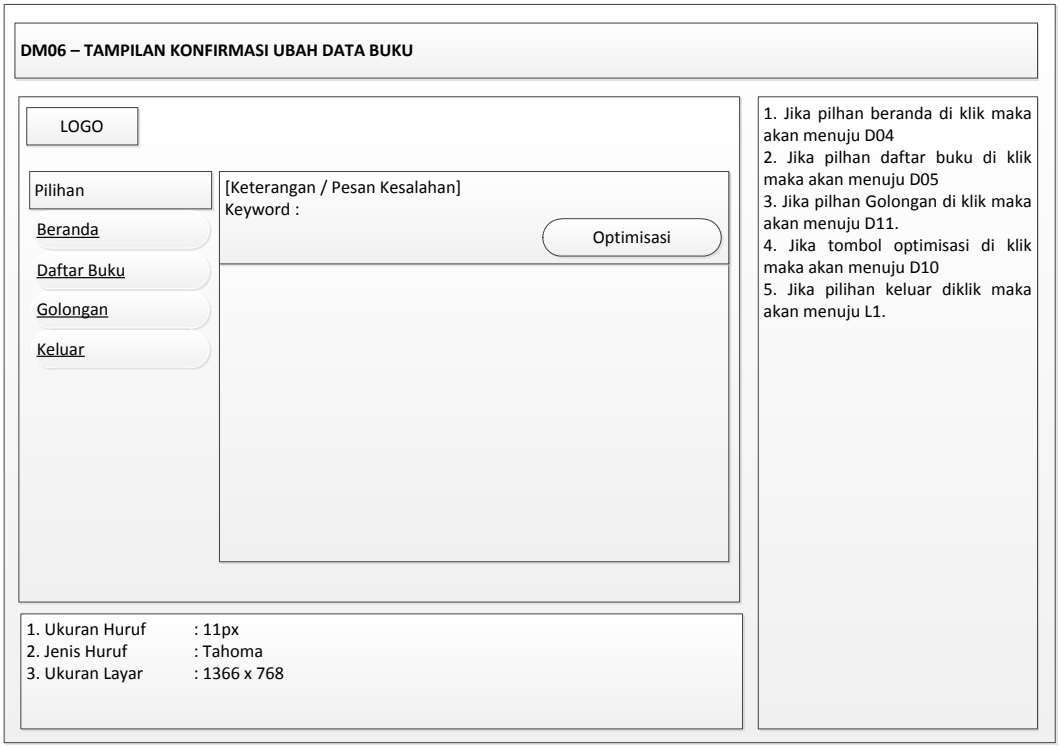
Gambar 3.47 Tampilan Detail Data Buku



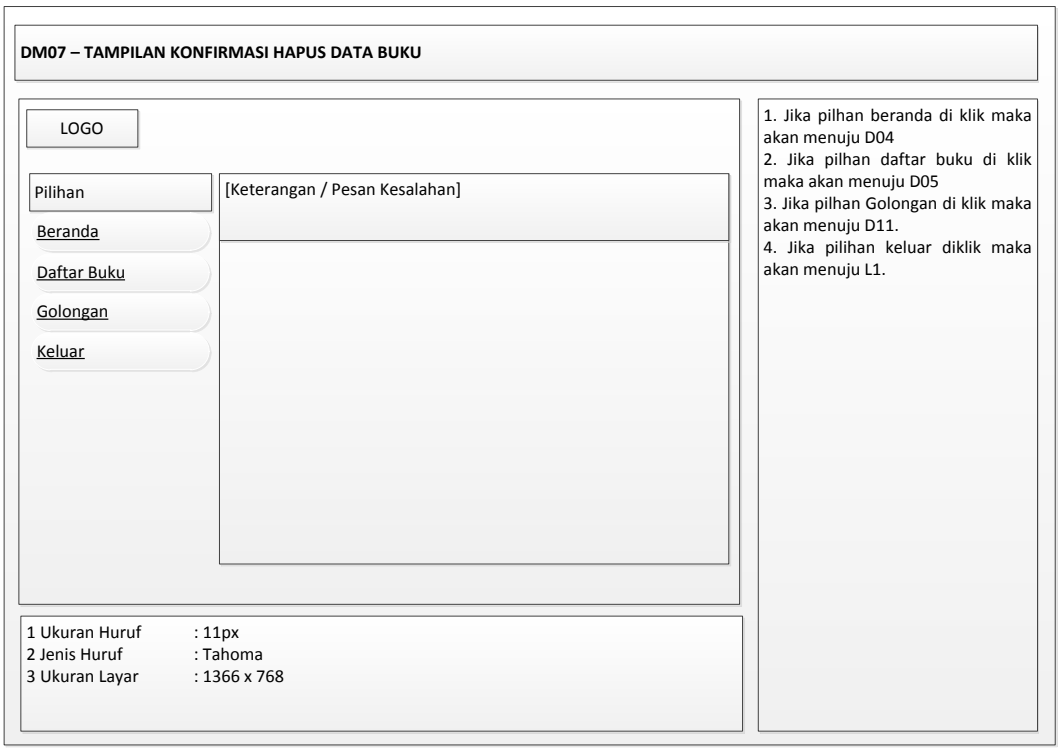
Gambar 3.48 Tampilan Optimisasi



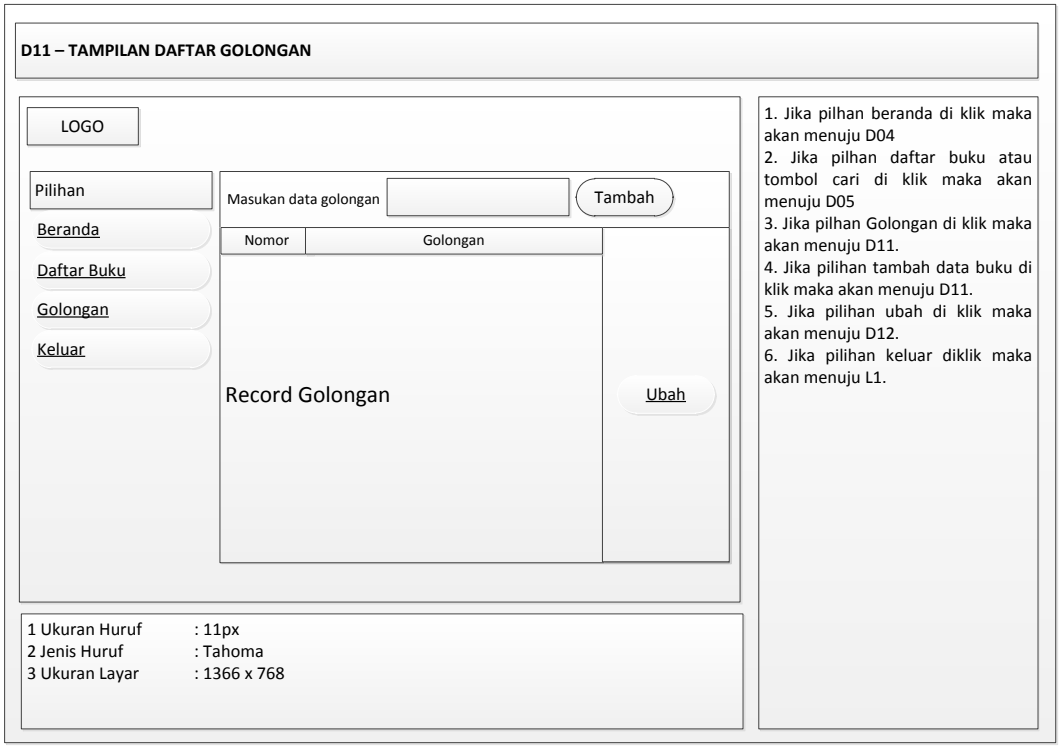
Gambar 3.49 Tampilan Konfirmasi Simpan Data Buku



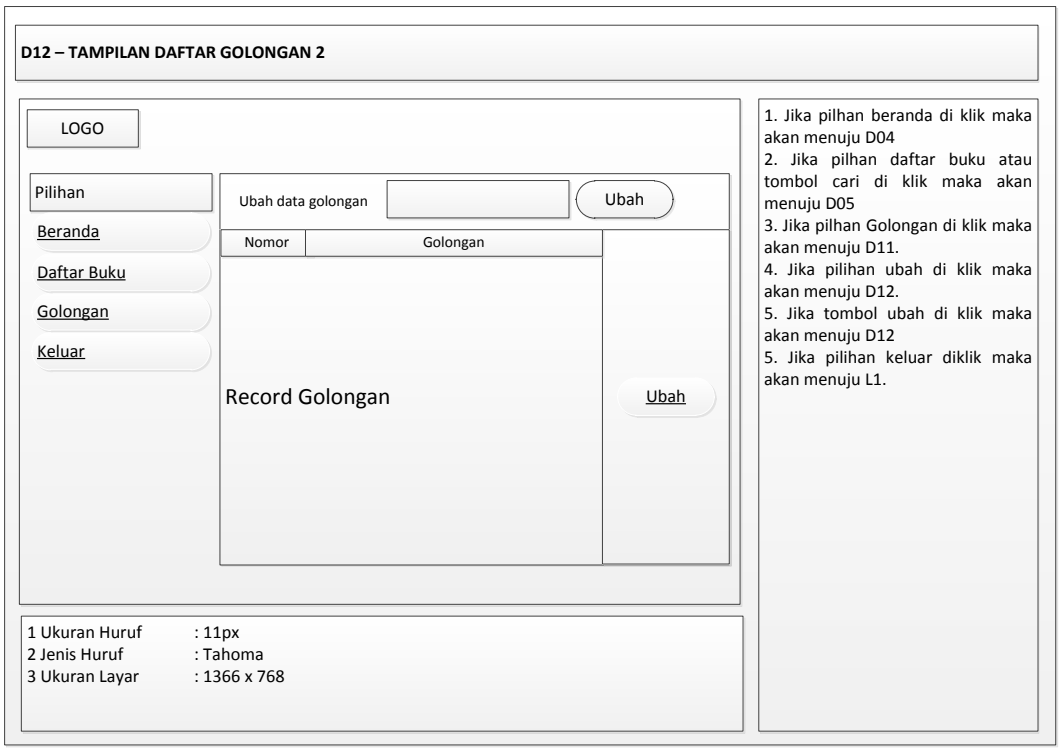
Gambar 3.50 Tampilan Konfirmasi Ubah Data Buku



Gambar 3.51 Tampilan Konfirmasi Hapus Data Buku



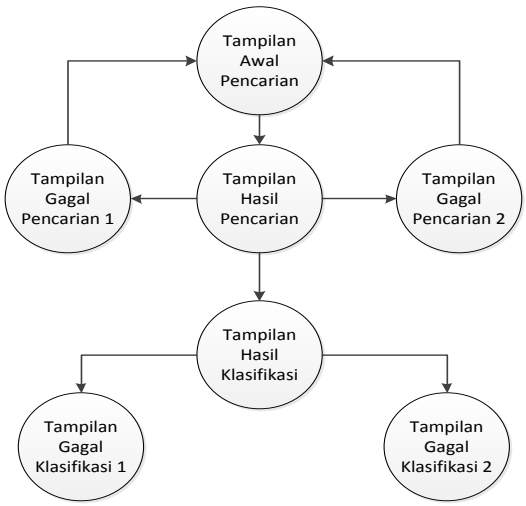
Gambar 3.52 Tampilan Daftar Golongan



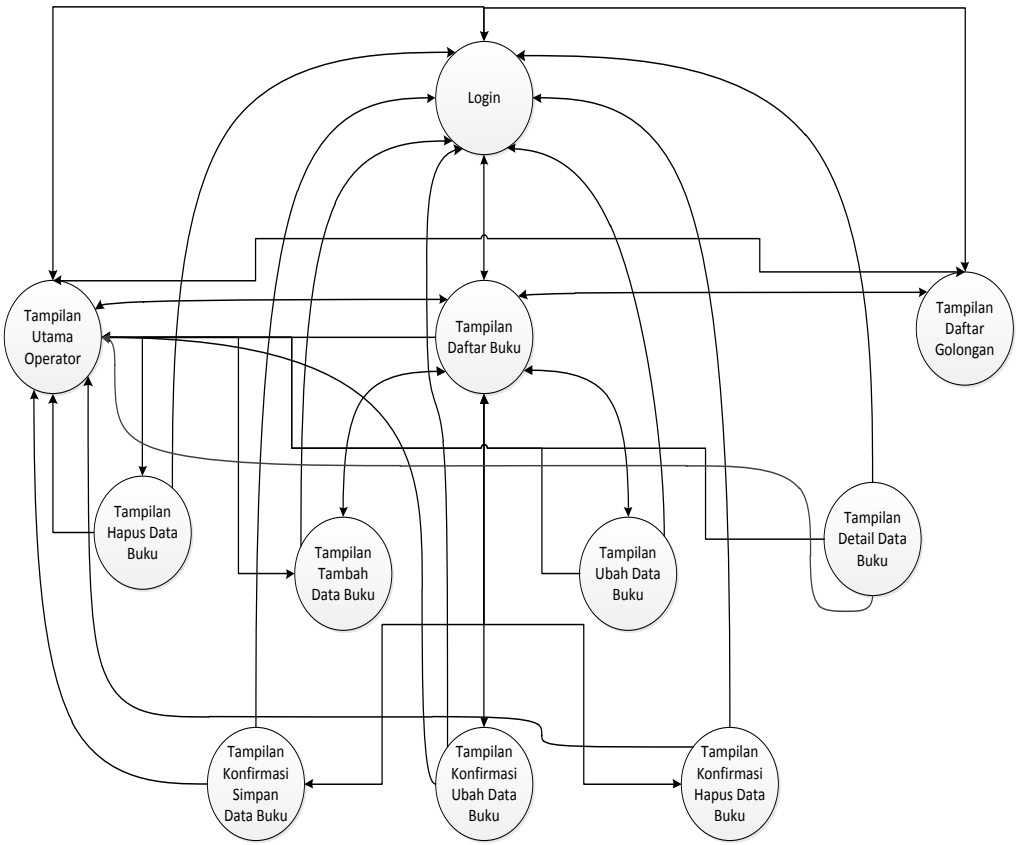
Gambar 3.53 Tampilan Daftar Golongan 2

3.2.2 Jaringan Semantik

Alur antarmuka yang terdapat pada sistem pencarian informasi buku, dapat digambarkan dalam jaringan dibawah ini.



Gambar 3.54 Jaringan Semantik - Pengunjung



Gambar 3.55 Jaringan Semantik – Operator

3.2.3 Perancangan Method

Sebagai langkah terakhir dalam perancangan yaitu merancang method yang akan diimplementasikan ke dalam sistem. Method ini akan digunakan sebagai algoritma dasar dalam mengkodekan prosedur yang ada. Adapun perancangan method untuk sistem pakar yang akan dibangun adalah sebagai berikut :

3.2.3.1 Method Pembuatan Keyword

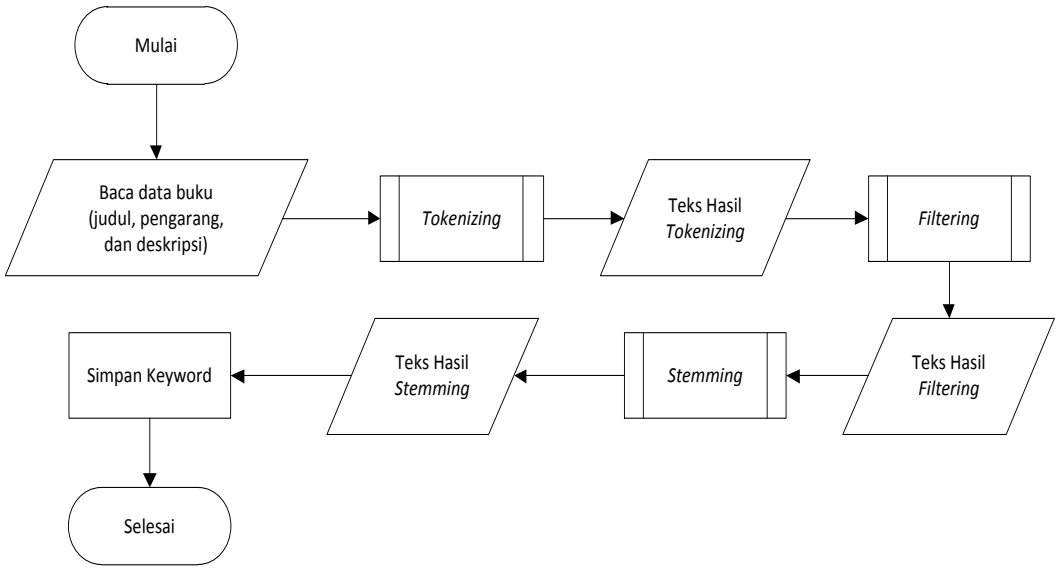
Method pencarian data buku digunakan untuk mencari informasi buku yang relevan terhadap *query* yang dimasukkan pengunjung.

A. Deskripsi Layanan

Operasi-operasi yang ada dalam method pencarian data buku adalah :

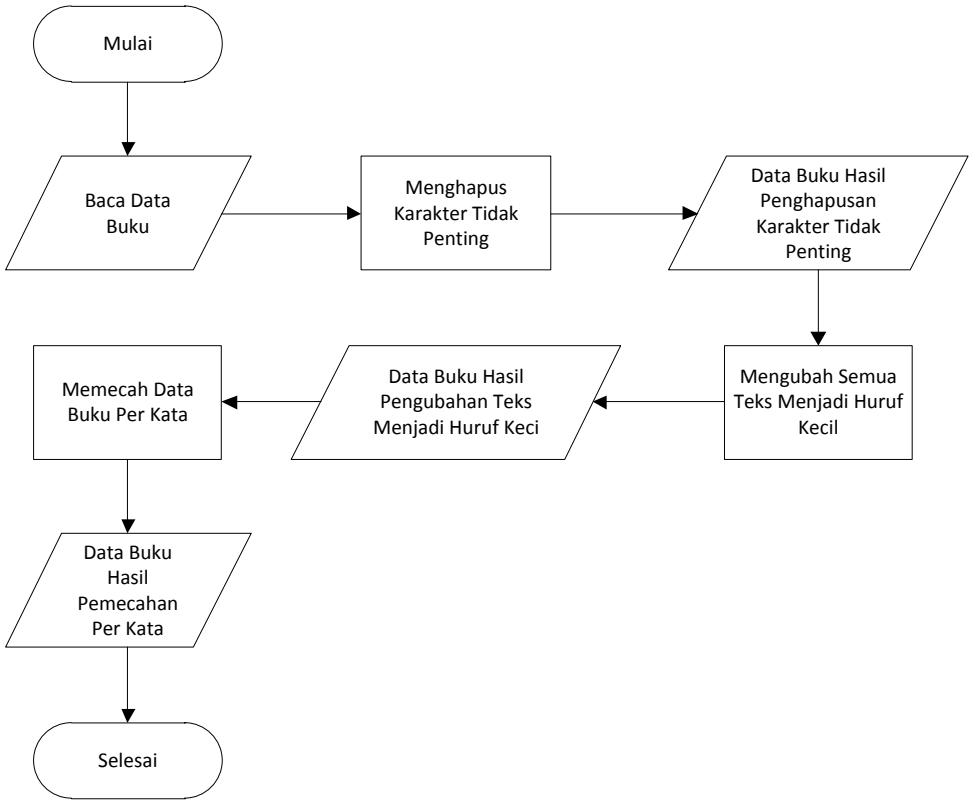
1. Method Tokenizing digunakan untuk melakukan pemotongan kata dan menghapus karakter yang tidak penting.
2. Method Filtering digunakan untuk menghapus kata-kata yang tidak penting.
3. Method Stemming digunakan untuk mencari kata dasar disetiap kata.

B. *Flowchart* Pembuatan Keyword



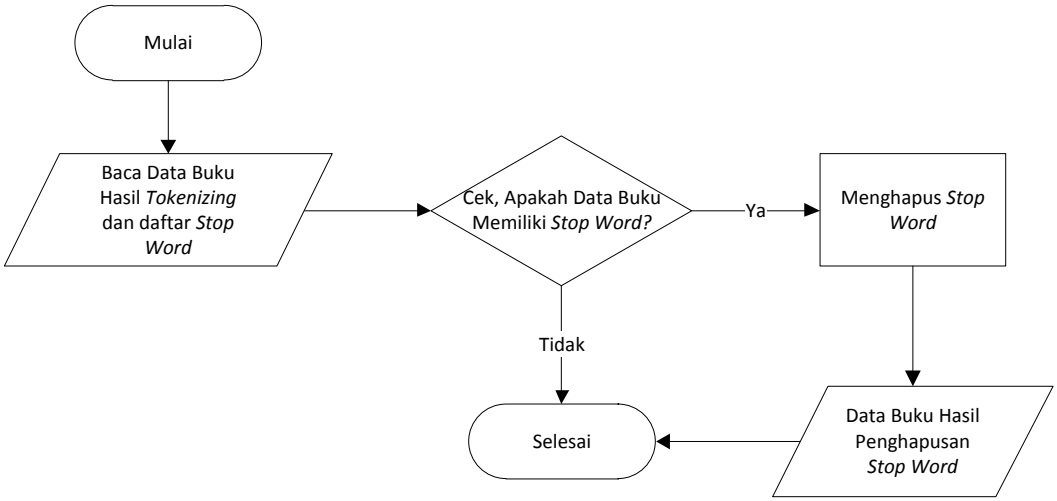
Gambar 3.56 *Flowchart* Pembuatan Keyword

C. *Flowchart Tokenizing*



Gambar 3.57 *Flowchart Tokenizing*

D. *Flowchart Filtering*



Gambar 3.58 Flowchart Filtering

E. *Flowchart Stemming*

Flowchart stemming bisa dilihat di BAB II halaman 19.

3.2.3.2 Method Pencarian Data Buku

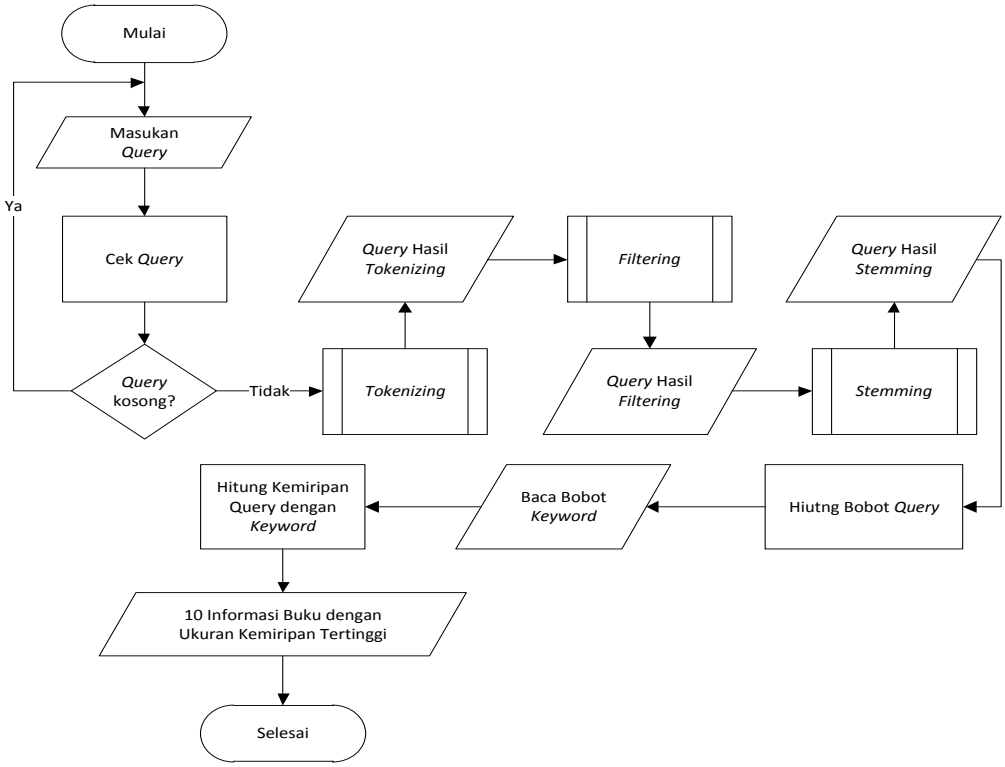
Method pencarian data buku digunakan untuk mencari informasi buku yang relevan terhadap *query* yang dimasukkan pengunjung.

A. Deskripsi Layanan

Operasi-operasi yang ada dalam method pencarian data buku adalah :

4. Method Tokenizing digunakan untuk melakukan pemotongan kata dan menghapus karakter yang tidak penting pada *query*.
5. Method Filtering digunakan untuk menghapus kata-kata yang tidak penting pada *query*.
6. Method Stemming digunakan untuk mencari kata dasar disetiap kata pada *query*.
7. Method Similiarity digunakan untuk menghitung tingkat kemiripan antara *query* dengan *keyword*.

B. *Flowchart* Method Pencarian Data Buku



Gambar 3.59 *Flowchart* Pencarian Data Buku

3.2.3.3 Method Pencarian Data Buku Lebih Lanjut

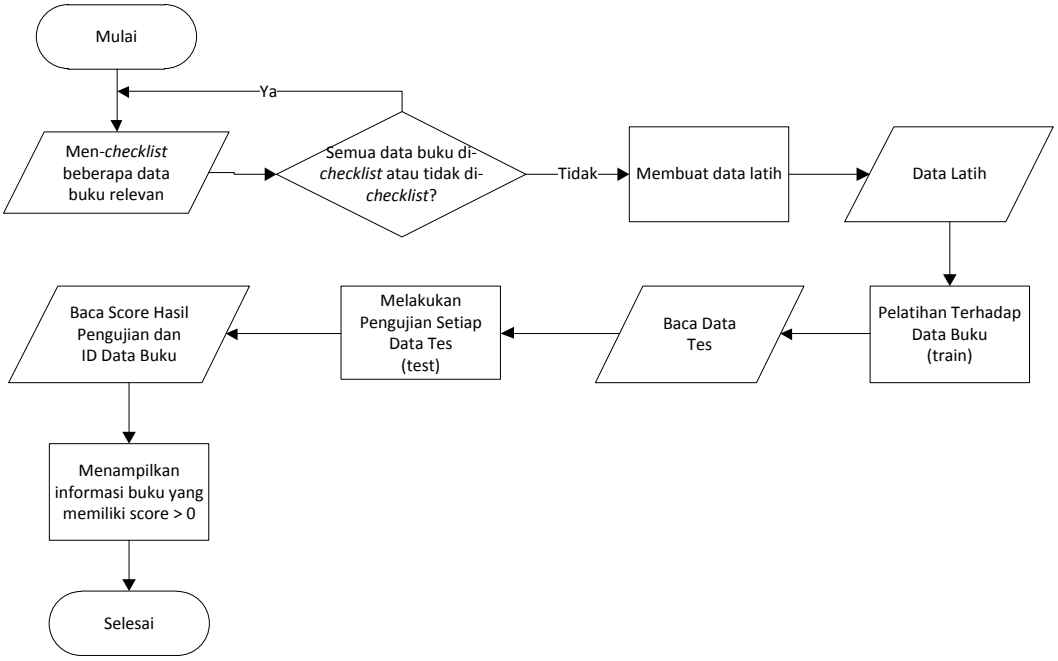
Method pencarian data buku lebih lanjut digunakan untuk mencari informasi buku yang lebih relevan dari hasil pencarian data buku.

A. Deskripsi Layanan

Operasi-operasi yang ada dalam method pencarian data buku lebih lanjut adalah :

1. Method Train digunakan untuk melakukan proses pembelajaran pada sistem pencarian.
2. Method Test digunakan untuk mengklasifikasi data buku dengan cara menghitung score pada setiap data tes.

B. *Flowchart* Method Pencarian Data Buku Lebih Lanjut



Gambar 3.60 *Flowchart* Pencarian Data Buku Lebih Lanjut

3.2.3.4 Method Optimisasi Sistem Pencarian

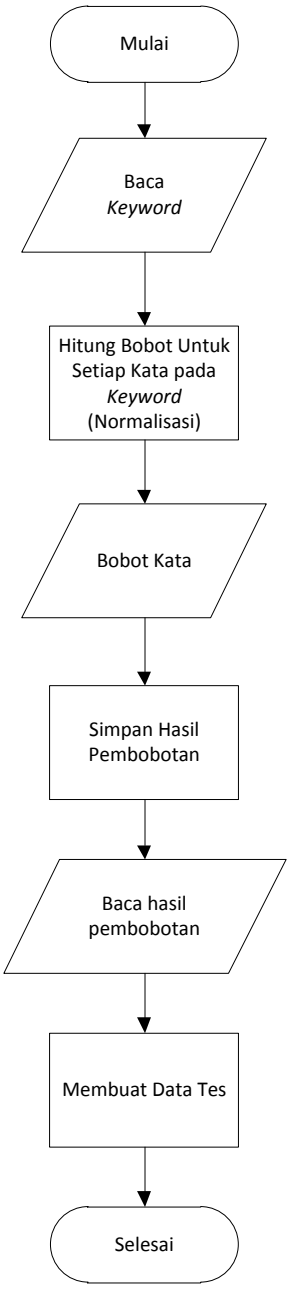
Method optimisasi sistem pencarian digunakan untuk mengubah data buku kedalam bentuk vektor dan mebuat data tes dari masing-masing data buku.

A. Deskripsi Layanan

Operasi-operasi yang ada dalam method optimisasi sistem pencarian adalah :

1. Method Normalisasi digunakan untuk mengubah *keyword* menjadi bentuk vektor dengan cara menghitung bobot setiap kata pada *keyword*.

B. *Flowchart* Optimisasi Sistem Pencarian



Gambar 3.61 *Flowchart* Optimisasi Sistem Pencarian Informasi