



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini antara lain adalah sebagai berikut.

a. Studi Literatur

Melakukan studi mengenai teori-teori dan konsep yang berkaitan dengan kegiatan penelitian, seperti teori mengenai algoritma *Answer Graph Generation*, perhitungan nilai *similarity* dengan *vector space model*, *Bloom Filter*, *stemming* kata berbahasa Indonesia menggunakan algoritma *Nazief-Adriani*, pengembangan sistem menggunakan *framework* CodeIgniter, dan berbagai konsep pendukung lainnya. Referensi-referensi yang digunakan dapat berupa buku, jurnal ilmiah, artikel, dan lain-lain.

b. Perancangan

Melakukan perancangan awal terhadap sistem yang akan dibangun, meliputi perancangan alur sistem dalam bentuk *flowchart*, *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD), dan rancangan antarmuka.

c. Implementasi

Melakukan pembangunan sistem dengan mengimplementasikan rancangan dan algoritma yang telah didefinisikan sebelumnya dengan menggunakan bahasa pemrograman yang telah ditentukan.

d. Uji coba dan evaluasi

Melakukan uji coba terhadap sistem disertai dengan evaluasi hasil yang didapatkan.

3.2 Analisis

Penelitian bertujuan membangun sebuah jurnal online dimana terdapat beberapa fitur utama seperti pencarian jurnal dan *content management system*. Fitur pencarian jurnal menggunakan beberapa algoritma, antara lain algoritma *Stemming Nazief-Adriani*, *Bloom Filter*, *Answer Graph Generation*, dan perhitungan nilai *similarity* menggunakan konsep *vector space model*. Sedangkan fitur *content management system* berupa operasi CRUD (*Create, Read, Update* dan *Delete*) konten jurnal.

Masalah utama yang dihadapi dalam penelitian ini adalah besarnya beban pemrosesan pada *ranking* dan *indexing*. Pada masalah *ranking*, penelitian mencoba menggunakan konsep *cache* untuk mengurangi pemrosesan yang sama. Sedangkan pada *indexing*, penelitian menggunakan algoritma *Bloom Filter* sehingga proses tersebut menjadi lebih cepat.

3.3 Perancangan

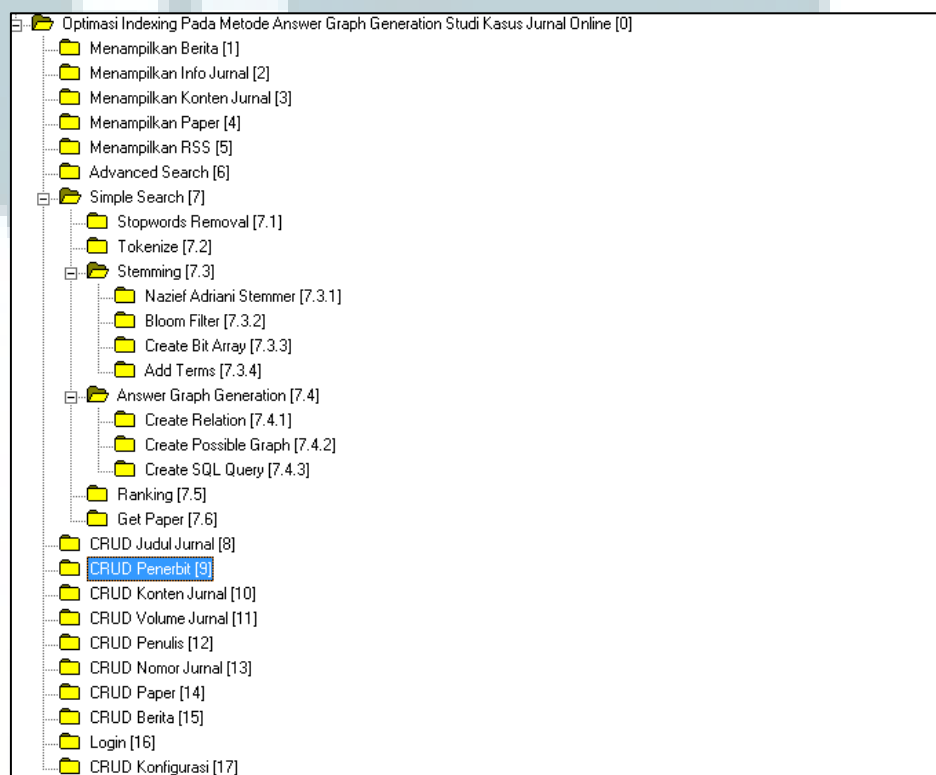
Pada subbab ini dipaparkan hasil perancangan dalam dua bagian, yaitu Pemodelan Sistem dan Sketsa Antarmuka Sistem.

3.3.1 Pemodelan Sistem

Pemodelan sistem dimulai dengan pembuatan model menggunakan diagram. Subbab ini dibagi menjadi empat bagian, yakni perancangan *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD), Diagram Alir (*flowchart*), dan struktur tabel.

A. Data Flow Diagram

Keseluruhan struktur sistem jurnal *online* memiliki *process tree* seperti yang diilustrasikan pada gambar 3.1.



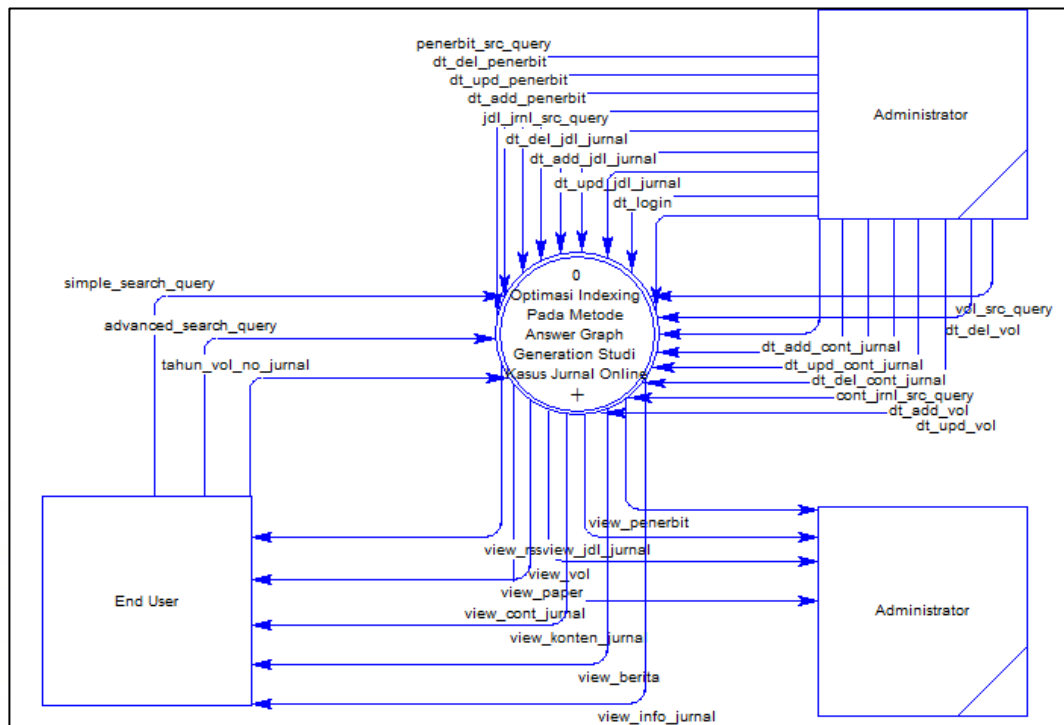
Gambar 3.1 *Process Tree Jurnal Online*

Sistem ini dibangun berdasarkan dua sisi kebutuhan pengguna, yaitu kebutuhan sebagai pengguna akhir dan administrator. Pada gambar 3.2-3.3 diilustrasikan bahwa sistem terdiri dari dua buah entitas eksternal (administrator

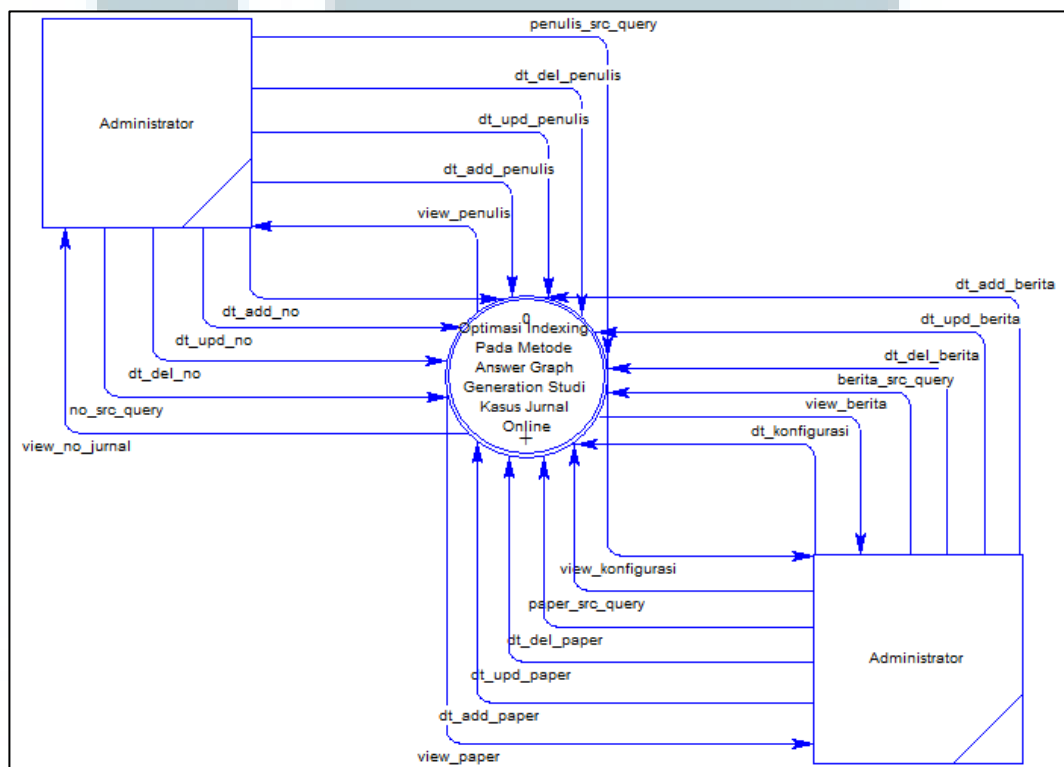
dan *end user*) dan satu buah proses utama. Entitas *end user* dapat melakukan berbagai macam kegiatan seperti melihat RSS, berita, *paper*, konten jurnal, informasi jurnal dan melakukan pencarian *paper* dengan *simple search* atau *advanced search*. Sedangkan kegiatan dari administrator lebih bersifat sebagai penyedia konten. Oleh karena itu, dibuatkanlah suatu antarmuka *content management system* untuk memudahkannya dalam memasukkan konten.

Untuk dapat masuk ke dalam *content management system*, seorang administrator harus melalui tahapan *login* terlebih dahulu sebagai salah satu prosedur keamanan. Setelah sampai di dalam halaman *content management system*, administrator dapat melakukan berbagai macam kegiatan utama seperti operasi CRUD (*create, read, update, delete*) judul jurnal, penerbit, konten jurnal, volume jurnal, nomor jurnal, penulis, *paper*, berita, dan kegiatan untuk mengubah konfigurasi sistem. Keseluruhan kegiatan tersebut dijelaskan lebih detil di dalam *context diagram* pada gambar 3.2-3.3.

U
M
N



Gambar 3.2 Context Diagram Jurnal Online



Gambar 3.3 Context Diagram Jurnal Online (lanjutan)

Berdasarkan DFD *level* 1 pada gambar 3.4, entitas pengguna akhir memiliki tujuh buah proses, yang dapat dijelaskan sebagai berikut.

1. Proses menampilkan berita.

Proses ini dimulai dari pengguna akhir memilih berita kemudian proses akan mengambil data dari dalam tabel *news_tb* lalu mengolahnya. Pada akhir proses, pengguna akhir akan disajikan sebuah halaman berisi konten berita yang telah dipilihnya.

2. Proses menampilkan informasi jurnal

Proses ini akan berjalan ketika pengguna akhir memilih salah satu jurnal dari daftar jurnal publikasi. Proses akan mengambil data dari tabel *journal_content_tb* dan *publisher_tb* untuk mengolahnya menjadi sebuah halaman.

3. Proses menampilkan konten jurnal

Proses ini akan berjalan ketika pengguna akhir berada pada halaman informasi jurnal kemudian dia memilih konten jurnal berdasarkan tahun, volume dan nomor tertentu dari *dropdown menu* yang telah disediakan.

Proses akan mengambil data dari tabel *journal_content_tb*, *publisher_tb*, *author_tb*, *paper_tb*, *journal_volume_tb*, *volume_issue_tb*, *journal_title_tb*, dan *paper_author_tb* untuk mengolahnya menjadi sebuah halaman.

4. Proses menampilkan *paper*

Proses ini akan berjalan ketika pengguna akhir berada pada halaman konten jurnal kemudian dia memilih salah satu *paper* dari daftar. Proses akan mengambil data dari tabel *journal_content_tb*, *publisher_tb*, *author_tb*,

paper_tb, *journal_volume_tb*, *volume_issue_tb*, *journal_title_tb*, dan *paper_author_tb* untuk mengolahnya menjadi sebuah halaman.

5. Proses menampilkan RSS

Proses ini akan berjalan ketika pengguna akhir memilih menu *RSS Feeds* kemudian proses akan memanggil proses menampilkan *paper* untuk membuatkan *paper_feeds* yang nantinya akan diolah menjadi format XML.

6. Proses *Advanced Search*

Proses ini akan berjalan ketika pengguna akhir memasukkan suatu *query* ke dalam fitur *advanced search*. *Query* yang dimasukkan akan diolah oleh proses ini hingga menampilkan *paper* hasil pencarian tersebut.

7. Proses *Simple Search*

Proses ini akan berjalan ketika pengguna akhir memasukkan suatu *query* ke dalam fitur *simple search*. *Query* yang dimasukkan akan diolah kemudian proses akan mengambil data dari tabel *journal_content_tb*, *publisher_tb*, *author_tb*, *paper_tb*, *journal_volume_tb*, *volume_issue_tb*, *journal_title_tb*, dan *paper_author_tb* untuk mengolahnya menjadi sebuah halaman.

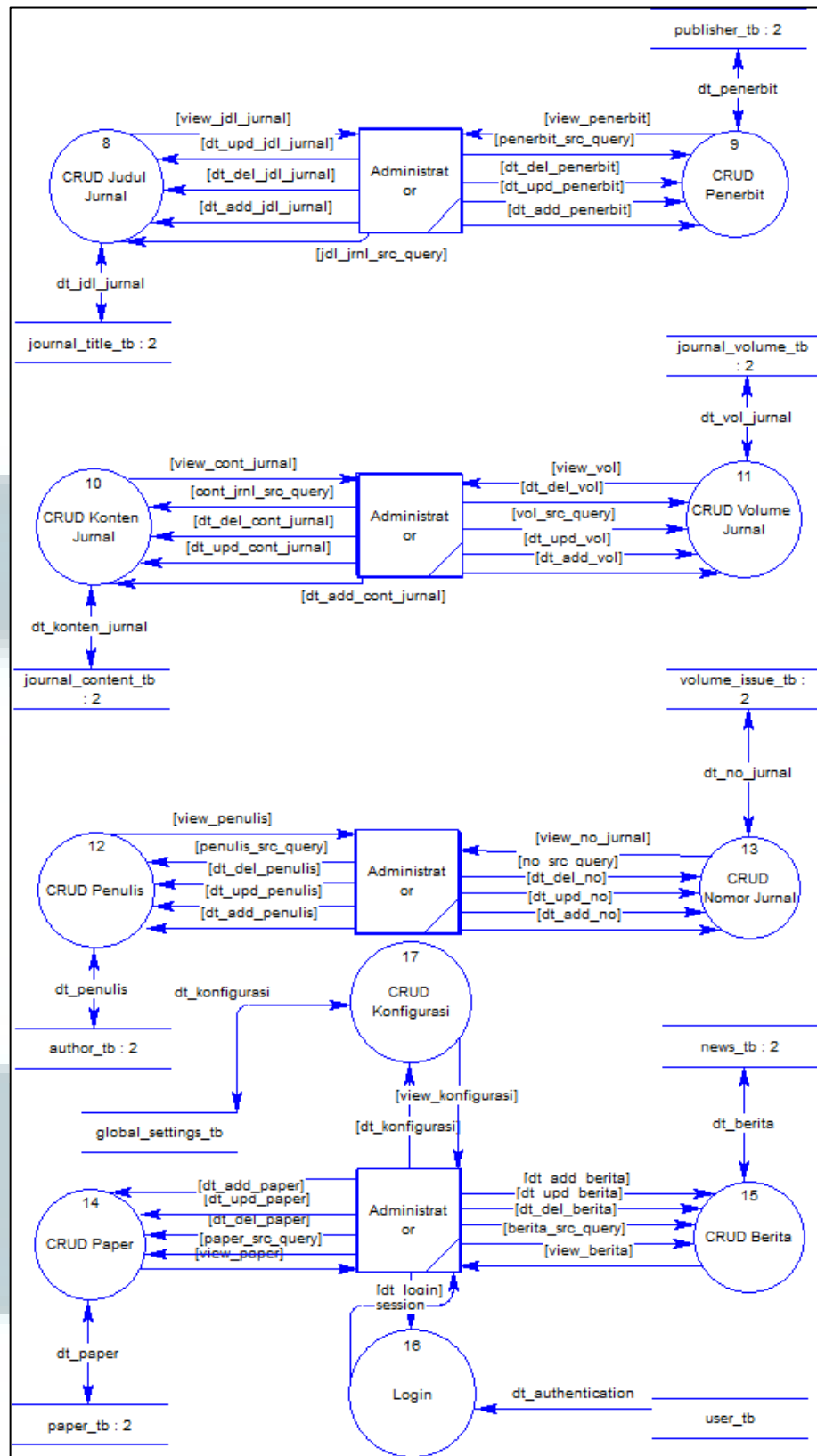
U
M
N

dalam tabel `user_tb`. Jika data tersebut valid maka proses *login* akan membuatkan sebuah *session* lalu mengarahkan administrator ke halaman utama *content management system*.

2. Proses CRUD judul jurnal, penerbit, konten jurnal, volume jurnal, nomor jurnal, penulis, paper, berita, konfigurasi.

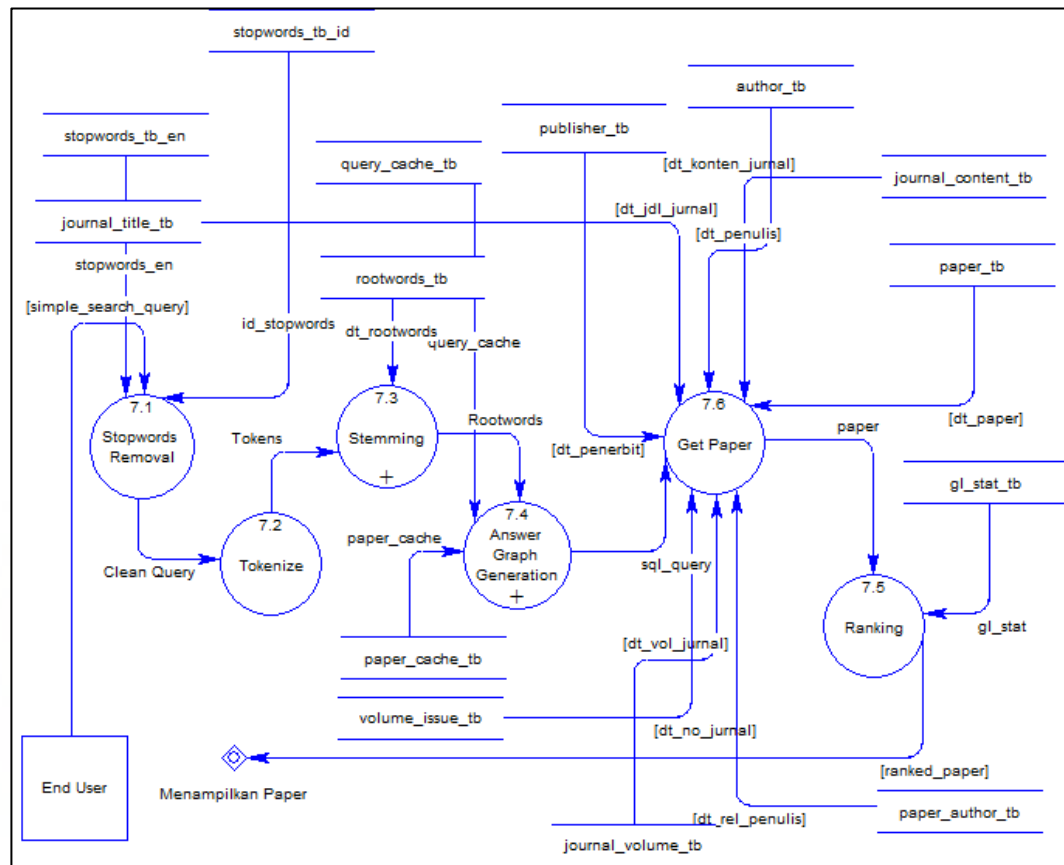
Kesembilan proses ini memiliki operasi utama yang sama, yaitu operasi *create*, *read*, *update*, dan *delete* dimana masing-masing operasi membutuhkan data masukan berupa `dt_add`, `dt_upd`, `dt_del`, `src_query` dan data keluaran berupa *view*. Beberapa tabel yang terlibat dalam sembilan proses ini antara lain `journal_title_tb`, `publisher_tb`, `journal_content_tb`, `journal_volume_tb`, `author_tb`, `volume_issue_tb`, `news_tb`, `paper_tb`, `user_tb`, dan `global_settings_tb`.

UMMN



Gambar 3.5 DFD Level 1 Jurnal Online dari Sisi Administrator

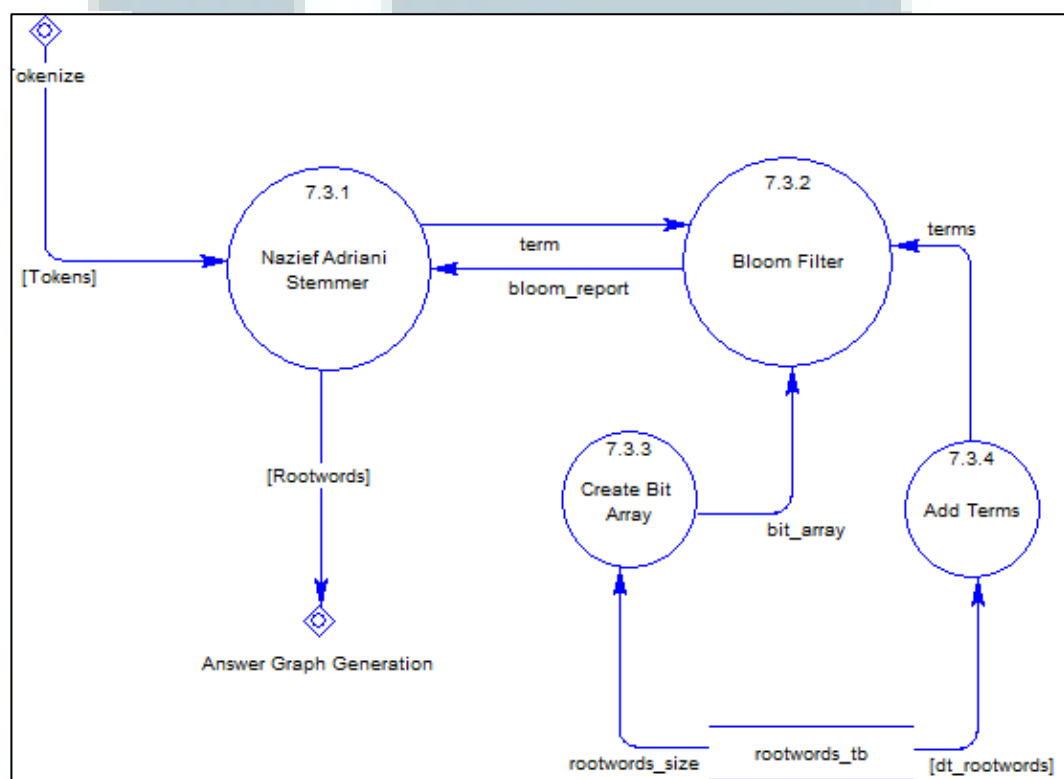
Berdasarkan DFD *level 2* pada gambar 3.6, proses *simple search* memiliki satu buah entitas (*end user*) dan enam buah proses pendukung dimana seluruh proses ini dimulai ketika entitas *end user* memasukkan *simple_search_query* ke dalam sistem. *simple_search_query* ini kemudian diolah oleh proses *stopwords removal* untuk membuang kata-kata yang tidak dibutuhkan (kata ganti orang, kata penghubung, dll.) dalam proses *retrieval* dengan melibatkan tabel *stopwords_tb_en* dan *stopwords_tb_id*. Hasil dari proses ini disebut dengan *clean_query*. Kemudian proses *tokenize* akan memecah *clean_query* yang masih berupa *string* ke dalam bentuk yang lebih kecil lagi yaitu bentuk kata atau *token*. *Token-token* inilah yang kemudian diproses oleh proses *stemming* untuk mendapatkan kata dasarnya atau *rootwords*. *Rootwords* ini akan diolah oleh proses *answer graph generation* untuk menghasilkan suatu *query* dalam bentuk *graph* sehingga proses *get paper* dapat menarik *paper* dari dalam tabel *author_tb*, *publisher_tb*, *journal_title_tb*, *journal_content_tb*, *journal_volume_tb*, *volume_issue_tb*, *paper_tb*, dan *paper_author_tb*. Proses terakhir adalah *ranking* dimana *paper-paper* yang sudah didapatkan dari proses *get paper* dihitung untuk mendapatkan peringkat kemiripan antara *query* dengan *paper* yang diambil oleh sistem. *Paper* yang memiliki peringkat paling tinggi akan berada di urutan pertama, sedangkan yang lain berada dibawahnya.



Gambar 3.6 DFD Level 2 Proses Simple Search

Proses *stemming* dapat dipecah menjadi beberapa subproses seperti yang diilustrasikan pada gambar 3.7. Pada gambar tersebut proses *stemming* memiliki empat buah subproses, yaitu proses *Nazief-Adriani Stemmer*, *Bloom Filter*, *create bit array*, dan *add terms*. Proses ini dimulai ketika proses *Nazief-Adriani Stemmer* menerima *tokens* dari proses *tokenize*. *Token-token* ini kemudian diolah dengan menggunakan aturan pembuangan imbuhan dan pencocokan kata dengan data yang ada di dalam kamus atau basis data. Adanya kelemahan pada proses ini yaitu lamanya waktu yang dibutuhkan untuk melakukan pencocokan kata, maka ditambahkan satu buah proses tambahan sebagai salah satu solusi, yaitu dengan menggunakan proses *Bloom Filter*. Proses *Bloom Filter* ini dimulai dengan proses

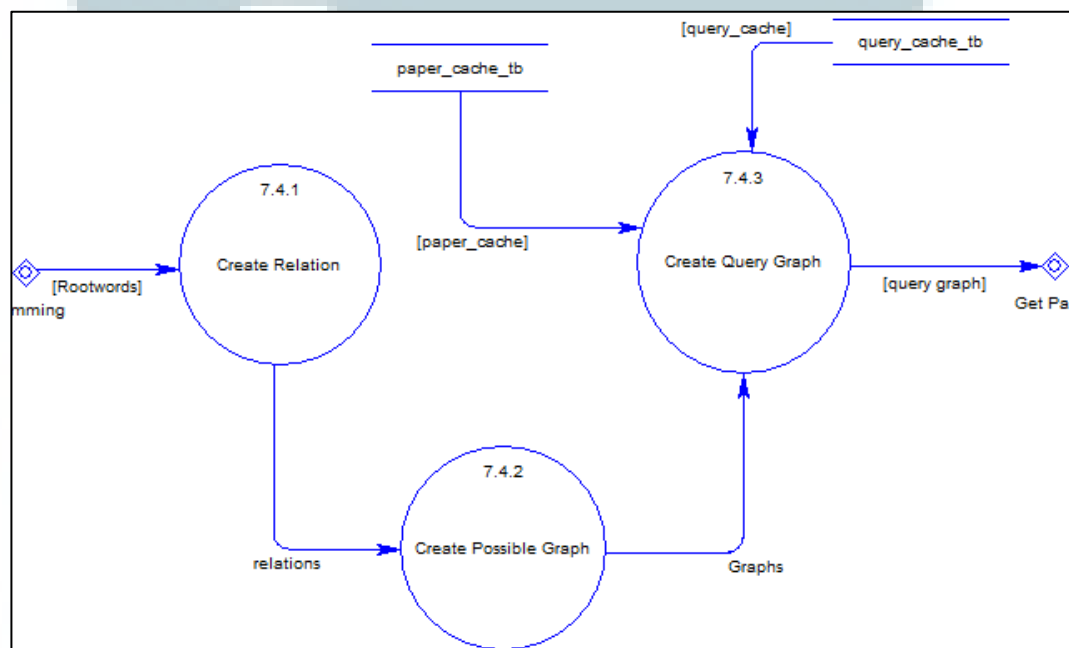
create bit array untuk membuat suatu ruang penyimpanan di dalam memori yang memiliki ukuran sesuai dengan jumlah data dari tabel *rootwords_tb*. Kemudian *bit array* ini akan diisi data-data berupa *terms* dari proses *add terms*. Setelah itu proses *Bloom Filter* dapat digunakan oleh proses *Nazief-Adriani Stemmer* untuk melakukan pencocokan kata dasar.



Gambar 3.7 DFD Level 3 Proses Stemming

Proses *answer graph generation* dapat dipecah menjadi beberapa subproses pendukung seperti yang diilustrasikan pada gambar 3.8. Proses ini terdiri dari proses *create relation*, *create possible graph*, dan *create SQL query*. Proses *create relation* akan berjalan ketika mendapatkan masukkan berupa kata dasar atau *rootwords*. Hasil keluaran dari proses ini disebut dengan *relations*.

Kemudian proses *create possible graph* akan membuat suatu *graph* berdasarkan *relations* yang diterima dari proses sebelumnya. Hasil keluaran dari proses ini berupa *graphs* yang nantinya akan diproses oleh proses *create SQL query* untuk menghasilkan *SQL query* agar dapat digunakan untuk menarik data dalam bentuk *paper*. Proses ini sebelumnya akan melakukan pemeriksaan ke dalam tabel *query_cache_tb* dan *paper_cache_tb* untuk mencari tahu apakah *query* tersebut sudah pernah dimasukkan ke dalam sistem atau belum. Jika *query* tersebut sudah pernah ada, maka tidak perlu dilakukan perhitungan *ranking* kembali.

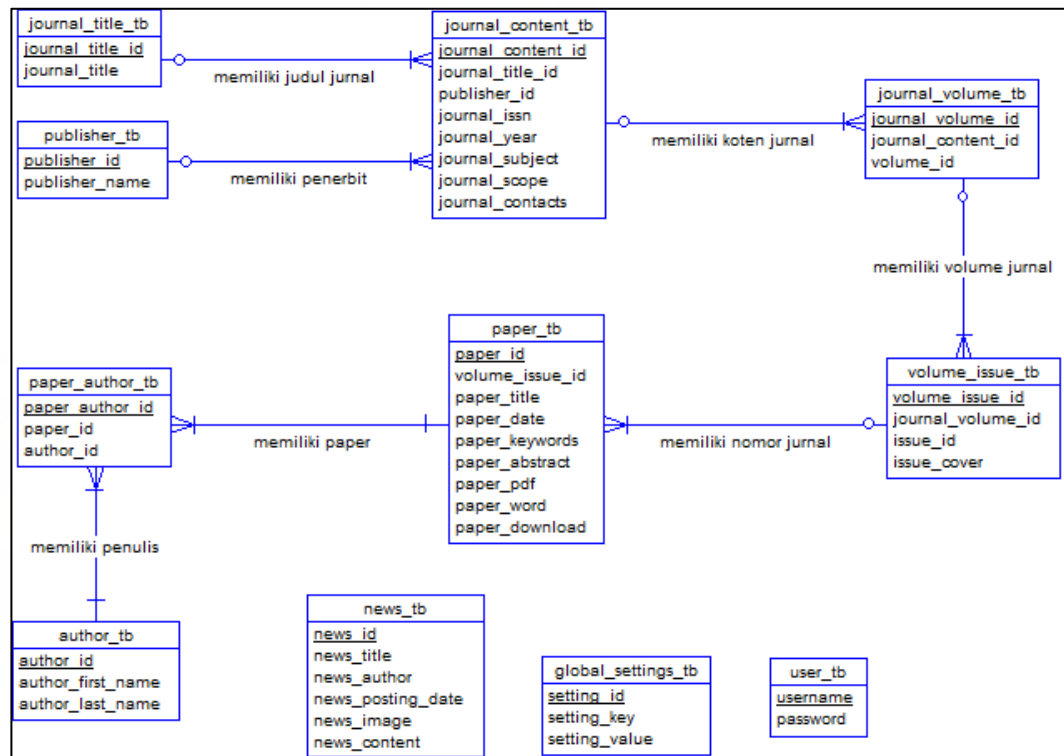


Gambar 3.8 DFD Level 3 Proses Answer Graph Generation

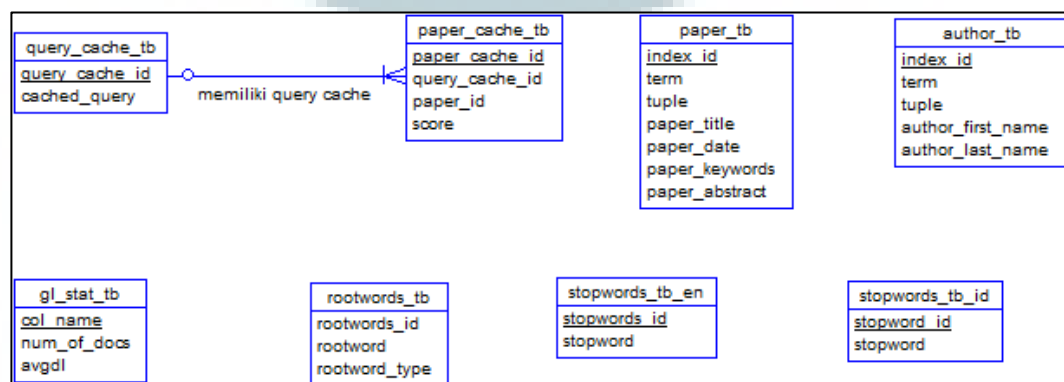
B. Entity Relationship Diagram

Dari penjelasan mengenai DFD dari gambar 3.2 sampai 3.8 terdapat beberapa *data store* yang terbagi menjadi dua basis data, yaitu *journal_db* dan

index_db. Masing-masing data store tersebut dapat diilustrasikan melalui gambar 3.9 dan 3.10.



Gambar 3.9 ERD journal_db



Gambar 3.10 ERD index_db

C. Diagram Alir

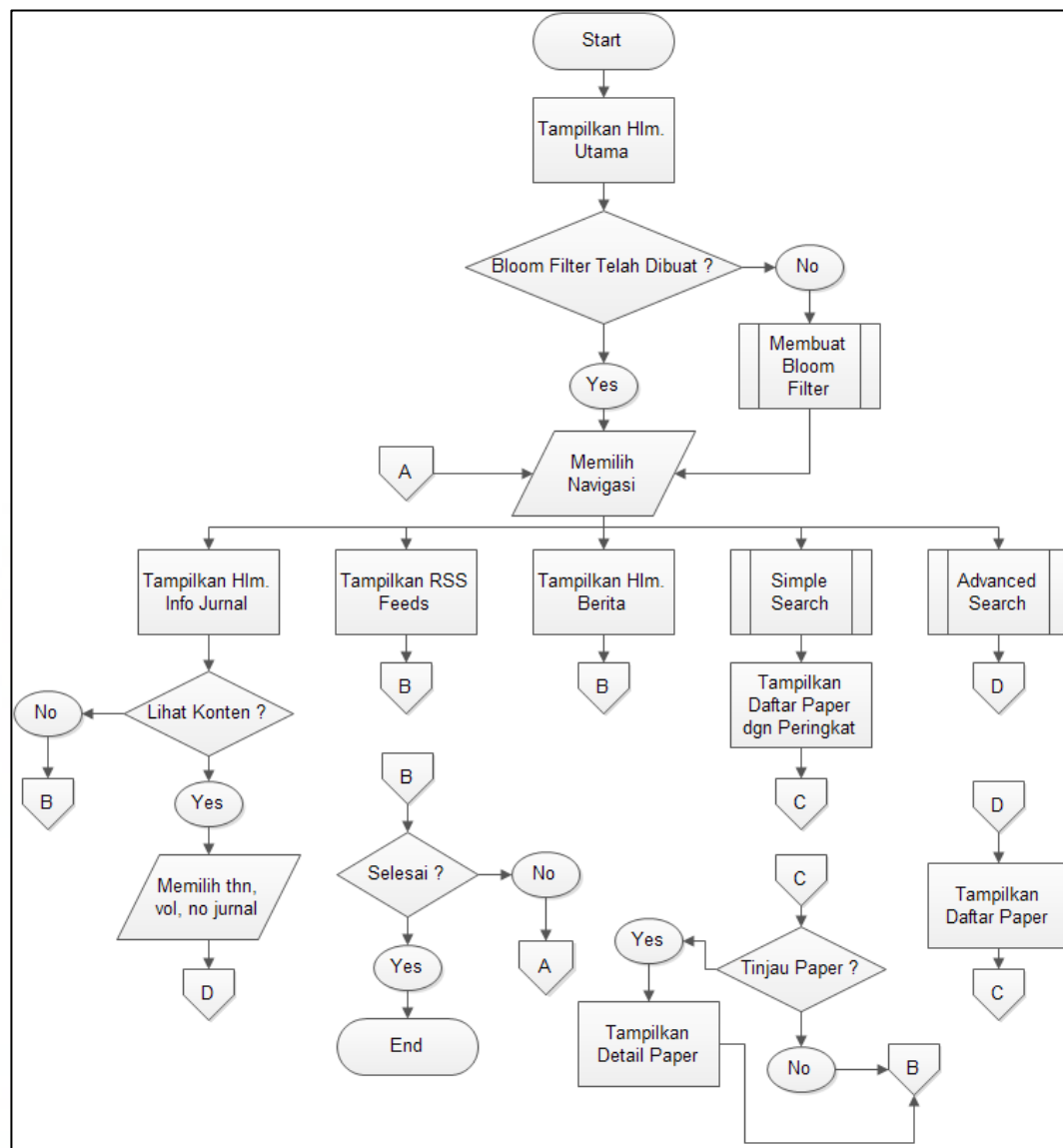
Diagram alir (*flowchart*) merupakan bagan-bagan yang memiliki arus yang menggambarkan langkah-langkah dalam menyelesaikan suatu masalah.

C.1 Flowchart Sistem (*End User*)

Ketika pengguna mengakses *website* jurnal *online*, sistem akan menampilkan halaman utama. Pada halaman inilah awal mula terjadinya interaksi antara pengguna dengan sistem. Setelah halaman utama berhasil ditampilkan, sistem akan melakukan pemeriksaan status dari *Bloom Filter*. Jika *Bloom Filter* belum pernah dibuat untuk *session* pengguna tersebut maka subproses membuat *Bloom Filter* akan dijalankan. Subproses ini dieksekusi oleh sistem secara otomatis dan *asynchronous* sehingga tidak mengganggu kegiatan pengguna.

Di dalam halaman utama pengguna disajikan beberapa navigasi seperti melihat informasi jurnal, RSS, berita dan beberapa fitur pencarian *paper* seperti *simple search* dan *advanced search*. Untuk melihat daftar *paper*, pengguna dapat menggunakan fitur pencarian atau melalui lihat konten pada halaman informasi jurnal. Untuk detail subproses *simple search* dan *advanced search* akan dijelaskan lebih detail pada bagian selanjutnya.

U
M
N



Gambar 3.11 *Flowchart Sistem (End User)*

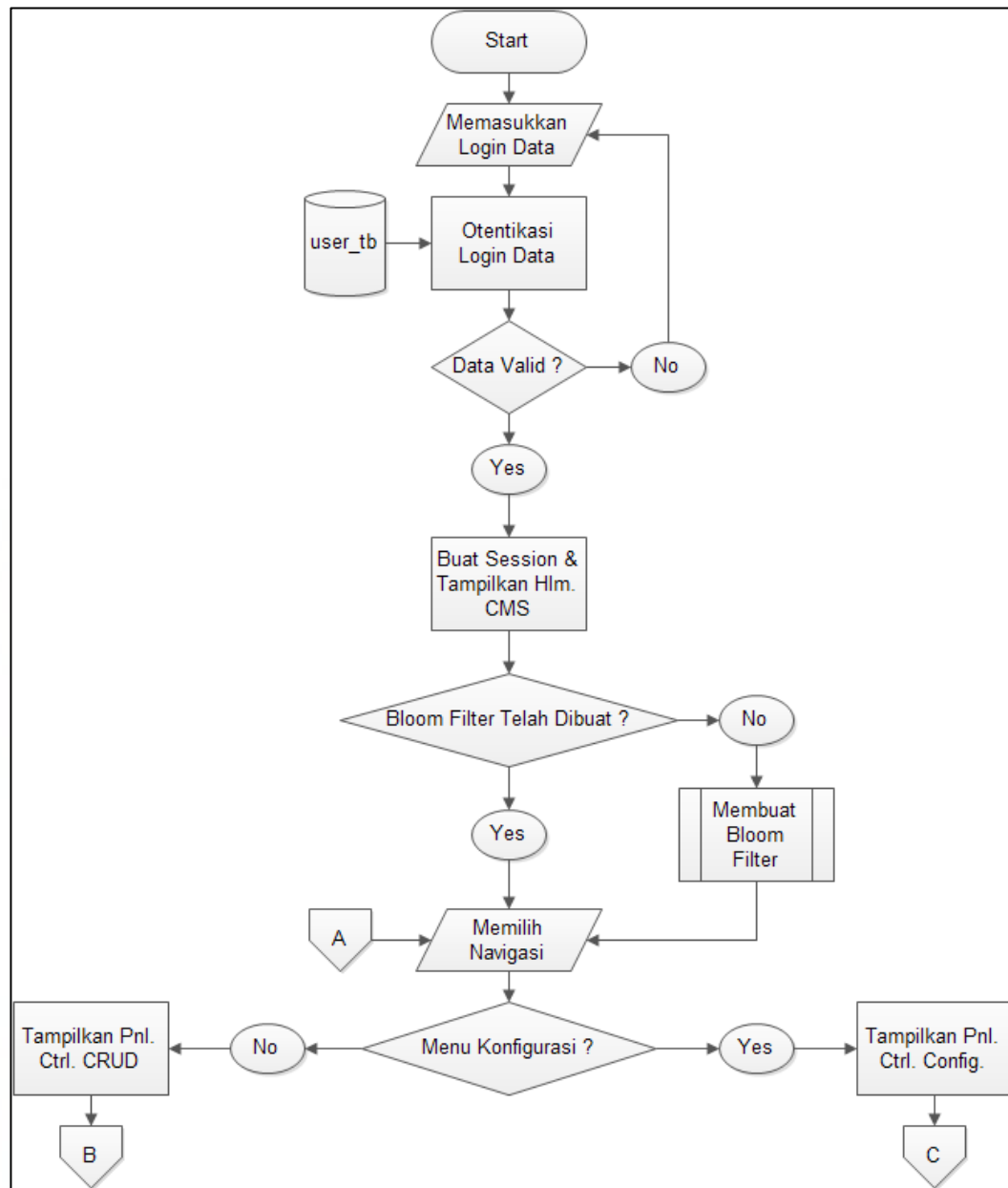
C.2 Flowchart Sistem (Administrator)

Flowchart sistem administrator sangat berbeda sekali dengan *flowchart* pada pengguna karena berupa *content management system*. Dengan adanya sistem ini memudahkan administrator dalam memasukkan konten jurnal. Untuk dapat mengakses *content management system* administrator diharuskan melakukan tahapan *login* terlebih dahulu. Jika otentikasi tersebut berhasil, maka administrator

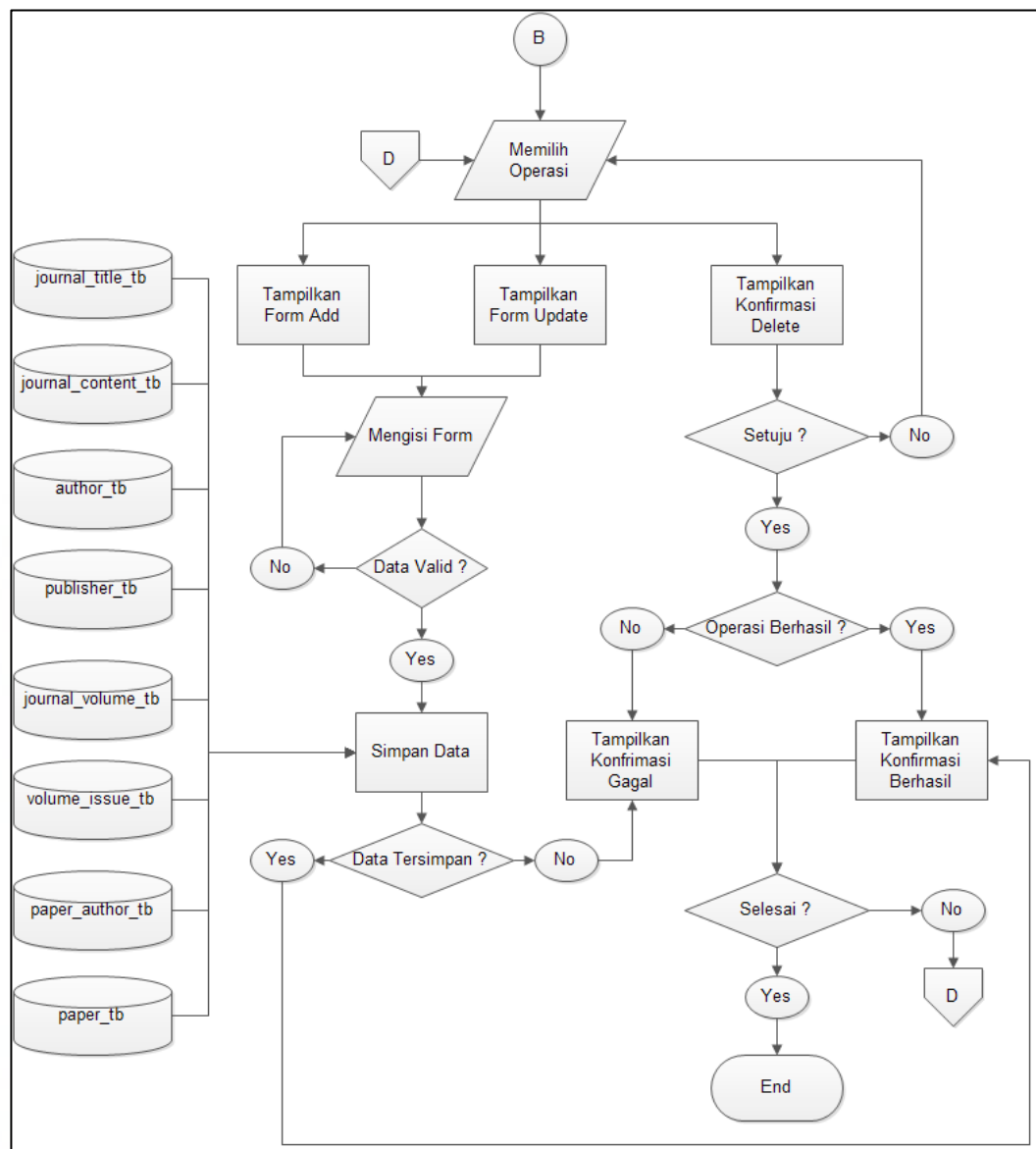
akan diarahkan pada halaman *content management system* dan dibuatkan sebuah *session*. Sama seperti pada penjelasan sistem sebelumnya, setelah halaman *content management system* berhasil ditampilkan sistem akan melakukan pemeriksaan status dari *Bloom Filter*. Jika *Bloom Filter* untuk *session* administrator belum pernah dibuat, maka subproses membuat *Bloom Filter* akan dijalankan.

Menu navigasi terbagi menjadi 2 bagian secara umum, yaitu bagian operasi CRUD dan konfigurasi yang diilustrasikan melalui gambar 3.12 sampai 3.14. Setiap kali administrator memilih operasi CRUD, sistem akan melakukan validasi terlebih dahulu pada semua data yang dimasukkan oleh administrator kemudian akan ditampilkan suatu pesan konfirmasi diakhir proses untuk memberitahu administrator apakah operasi tersebut berhasil atau tidak.

Pada bagian konfigurasi, administrator dapat menggunakan beberapa fitur seperti *change password*, *preferences*, *indexing*, *optimize tables*, dan *run janitor*. Ketika administrator memilih fitur *change password* atau *preferences*, sistem akan menampilkan suatu *form*. Isi *form* tersebut akan divalidasi terlebih dahulu sebelum dimasukkan ke dalam basis data. Untuk fitur *indexing*, *optimize tables* dan *run janitor*, administrator akan diberikan pesan konfirmasi yang memberitahukan kepada administrator apakah dia setuju proses tersebut akan dijalankan. Jika administrator meyetujuinya, operasi tersebut akan berjalan dan diakhiri dengan pesan konfirmasi lagi yang memberitahu administrator apakah operasi berjalan dengan lancar atau tidak.

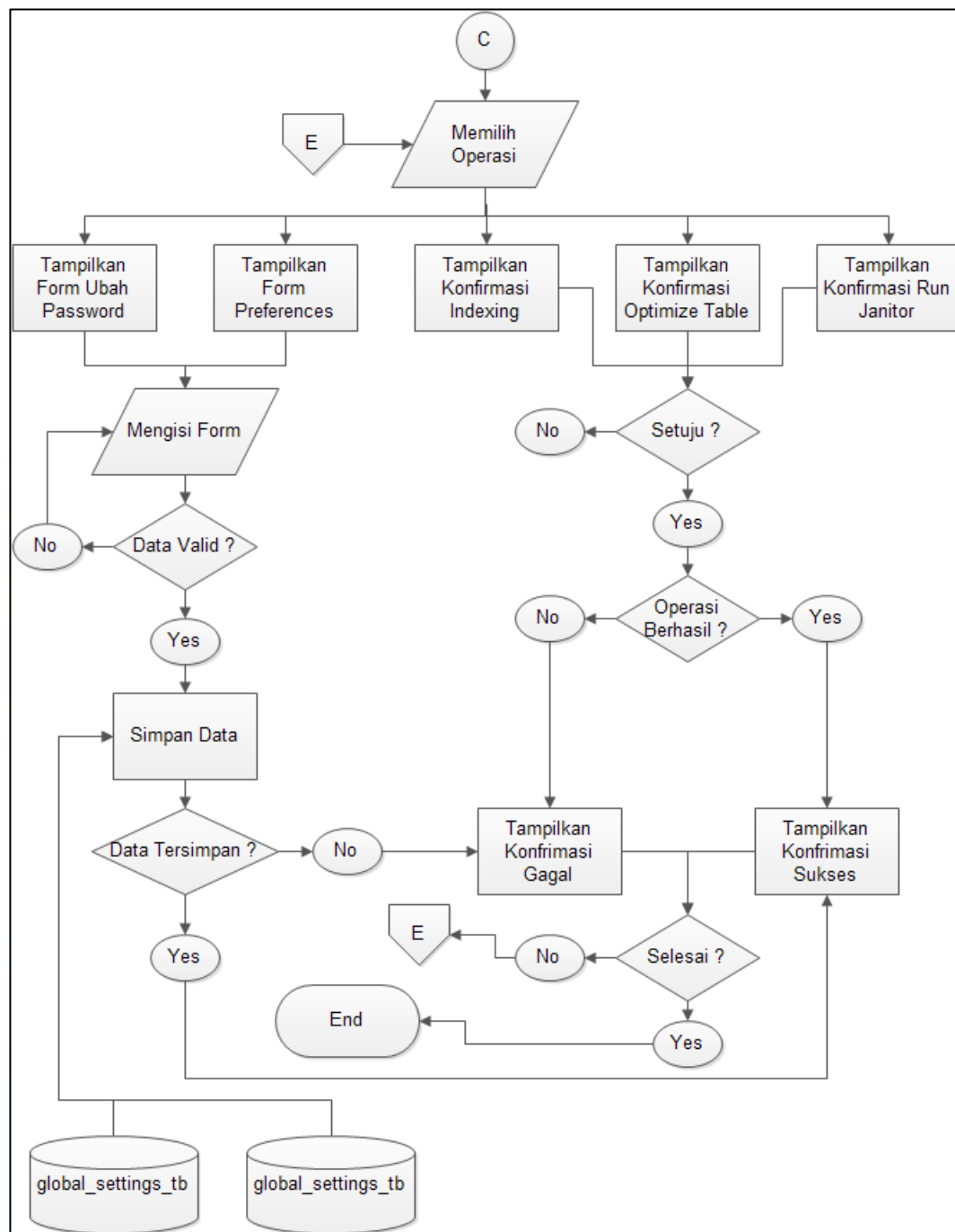


Gambar 3.12 *Flowchart* Sistem (Administrator)



Gambar 3.13 Flowchart Sistem (Administrator - lanjutan)

U
M
M
N

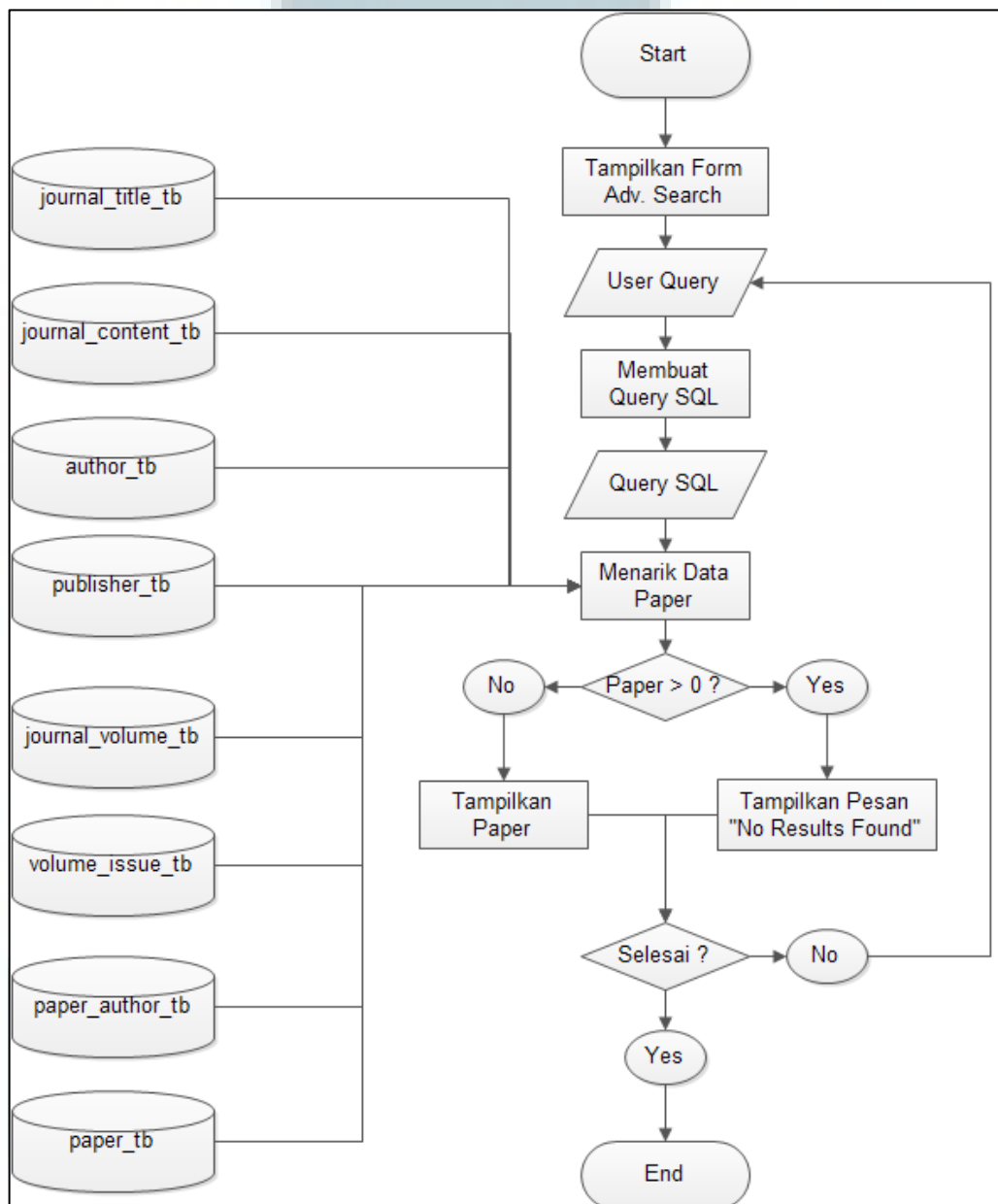


Gambar 3.14 Flowchart Sistem (Administrator - lanjutan)

C.3 Flowchart Subproses Advanced Search

Kegiatan ini dimulai dari pengguna memasukkan *query* beserta operator pembantu kedalam sistem. Sistem akan menerjemahkan *query* dan operator

tersebut kedalam *SQL query*. Jika *query* tersebut tidak menghasilkan *paper* sama sekali, maka sistem akan menampilkan pesan bahwa tidak ada *paper* yang terambil dan sebaliknya.

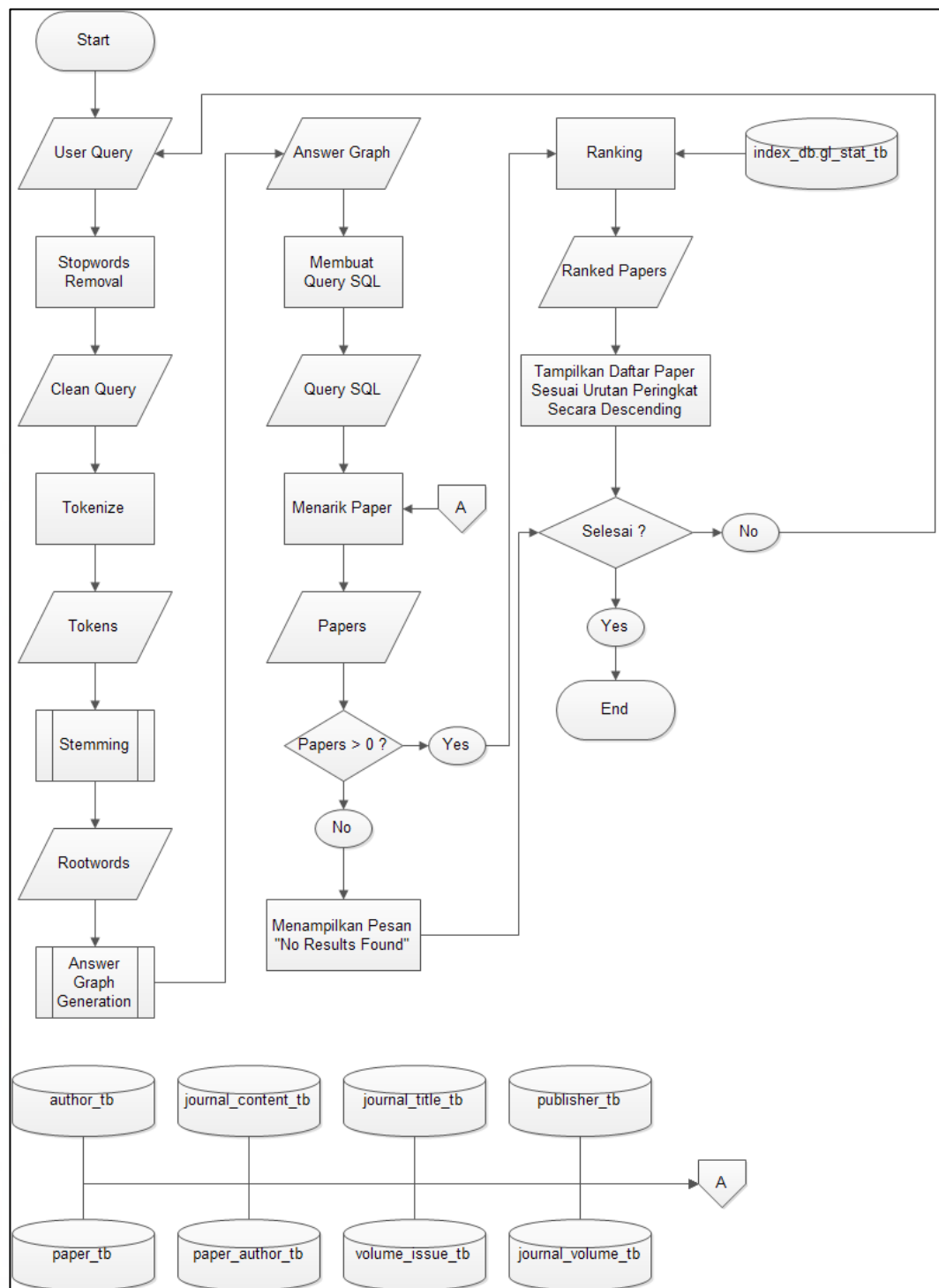


Gambar 3.15 Flowchart Subproses Advanced Search

C.4 Flowchart Subproses Simple Search

Flowchart subproses *simple search* dimulai dengan adanya masukan *query* dari pengguna ke dalam sistem. *Query* tersebut pertama kali akan diolah oleh proses *stopwords removal* untuk membuang kata-kata yang tidak dibutuhkan dalam proses *retrieval*. Hasil dari proses ini disebut dengan *clean query*, untuk memudahkan proses selanjutnya *query* ini dipecah menjadi beberapa bagian yang lebih kecil lagi yaitu *token* melalui proses *tokenize*. *Token-token* inilah yang nantinya akan diproses oleh subproses *stemming* untuk mendapatkan kata dasarnya atau *rootwords*.

Rootwords kemudian digunakan oleh subproses *answer graph generation* untuk membuat *answer graph*. *Answer graph* ini sangat membantu sekali dalam pembuatan *SQL query*. *Paper-paper* kemudian diambil menggunakan *SQL query* dari dalam database dan pemeriksaan *paper* dilakukan untuk menentukan apakah perlu dilakukan *ranking* atau tidak. Jika tidak ada *paper* yang terambil, maka sistem tidak perlu melakukan proses *ranking* dan langsung menampilkan ke pengguna bahwa *query* tersebut tidak menghasilkan jawaban. Sedangkan jika *query* tersebut menghasilkan beberapa *paper*, maka proses *ranking* akan dieksekusi dan hasil dari proses tersebut ditampilkan ke dalam suatu daftar *paper* yang diurutkan berdasarkan peringkat tertinggi.



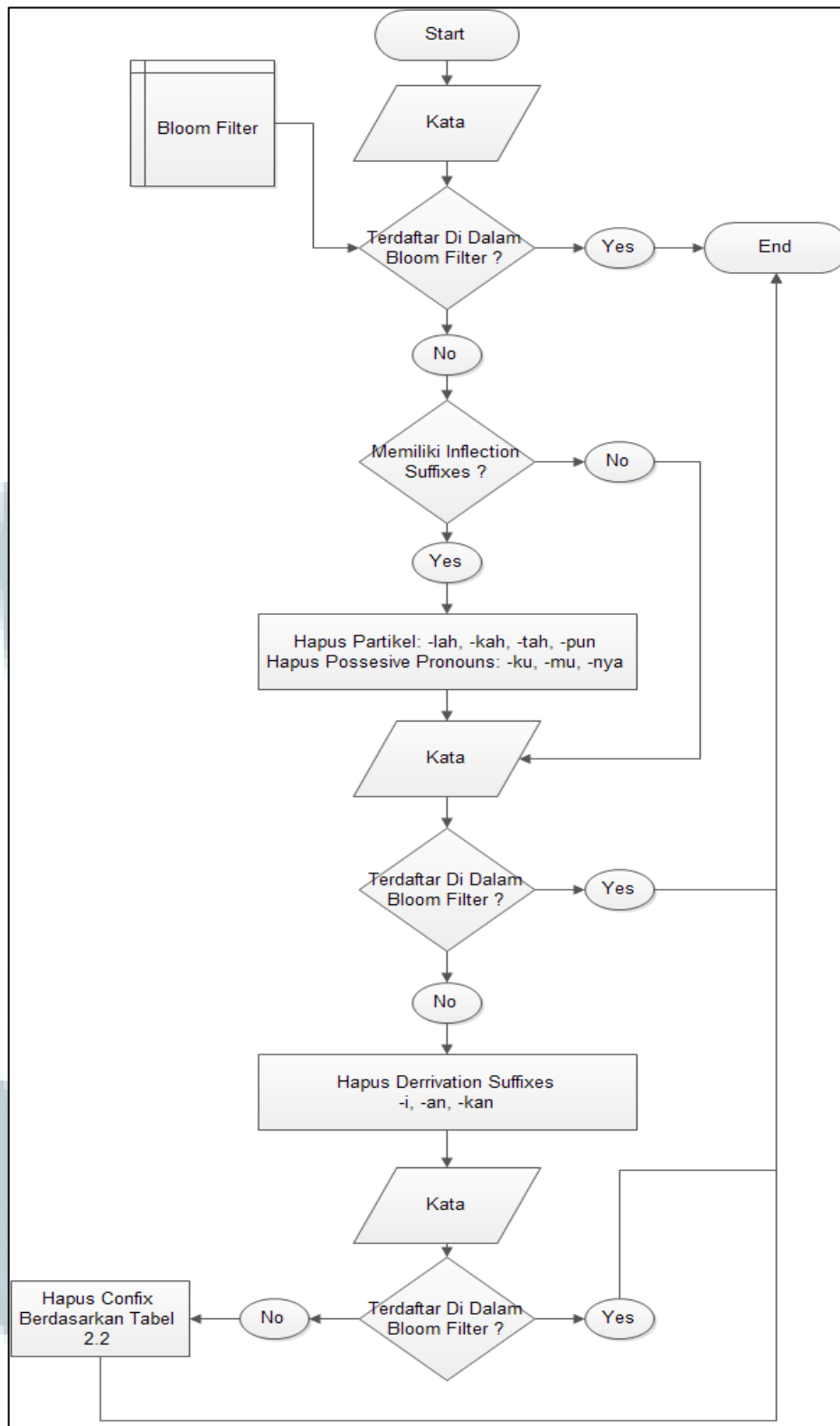
Gambar 3.16 Flowchart Subproses Simple Search

C.5 Flowchart Subproses Stemming Kata Nazief-Adriani

Proses ini merupakan modifikasi dari algoritma *Nazief-Adriani* dimana sebelumnya proses ini menggunakan kamus data setiap kali ada pemeriksaan kata dasar. Namun karena adanya kelemahan disisi waktu akibat proses tersebut, maka salah satu solusi yang memungkinkan adalah dengan menggunakan algoritma *Bloom Filter* yang lebih efisien.

Setiap kata yang masuk ke dalam proses ini akan diperiksa terlebih dahulu apakah kata tersebut telah terdaftar di dalam *Bloom Filter*, jika ternyata sudah terdaftar maka kata tersebut dapat diasumsikan sebagai kata dasar dan proses dianggap telah selesai. Jika belum, maka *inflection suffixes* (-lah, -kah, -tah, -pun, -ku, -mu, -nya) dari kata tersebut akan dihapus.

Kata ini kemudian akan diperiksa kembali ke dalam *Bloom Filter*, jika terdaftar maka proses dianggap telah selesai. Proses berikutnya adalah membuang *derrivation suffixes* (-i, -an, -kan). Kata ini kemudian akan diperiksa kembali ke dalam *Bloom Filter*, jika terdaftar maka proses dianggap telah selesai. Proses yang terakhir adalah menghapus *confix* yang terdapat pada kata. Dalam menghapus *confix*, sistem harus mengikuti beberapa aturan yang telah dijelaskan pada tabel 2.2. Di dalam proses ini juga terjadi pemeriksaan menggunakan *Bloom Filter* sehingga diakhir proses ini kata tersebut diasumsikan sebagai kata dasar.



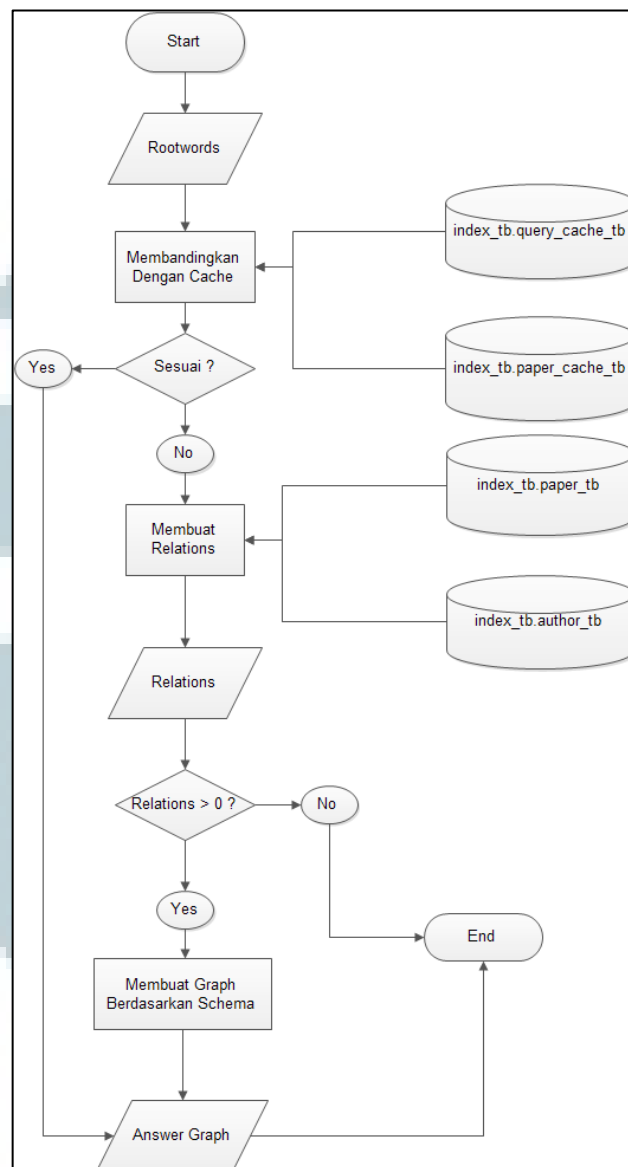
Gambar 3.17 *Flowchart Subproses Stemming Nazief-Adriani*

C.6 Flowchart Subproses Answer Graph Generation

Pada proses ini sebelum membuat *relation*, kata dasar dicocokkan terlebih dahulu dengan *query cache* yang ada di dalam sistem. Hal ini dimaksudkan agar sistem lebih cepat dalam menampilkan hasil karena *query* yang sudah pernah dimasukkan oleh pengguna lain tidak perlu lagi dicari nilai peringkatnya. Jika ternyata belum terdaftar di dalam *cache*, maka sistem akan memulai membuat *relation* dengan membandingkan dengan kata-kata yang telah diindeks sebelumnya. Apabila ternyata kata-kata tersebut terdapat di dalam indeks, maka *relation* akan terbentuk. Jika tidak ada *relation* yang terbentuk sama sekali, maka proses ini akan berakhir.

Relation inilah yang merupakan komponen utama dalam membuat *answer graph*. Pembuatan *answer graph* dimulai dengan menghubungkan setiap *relation* berdasarkan *schema graph* yang telah dibuat sebelumnya. Setiap *relation* dianggap sebagai sebuah *node* sedangkan hubungan antar satu *node* dengan lainnya dianggap suatu *edge*. Hasil akhir berupa *answer graph* akan dikirim ke proses selanjutnya.

UMMN



Gambar 3.18 *Flowchart Subproses Answer Graph Generation*

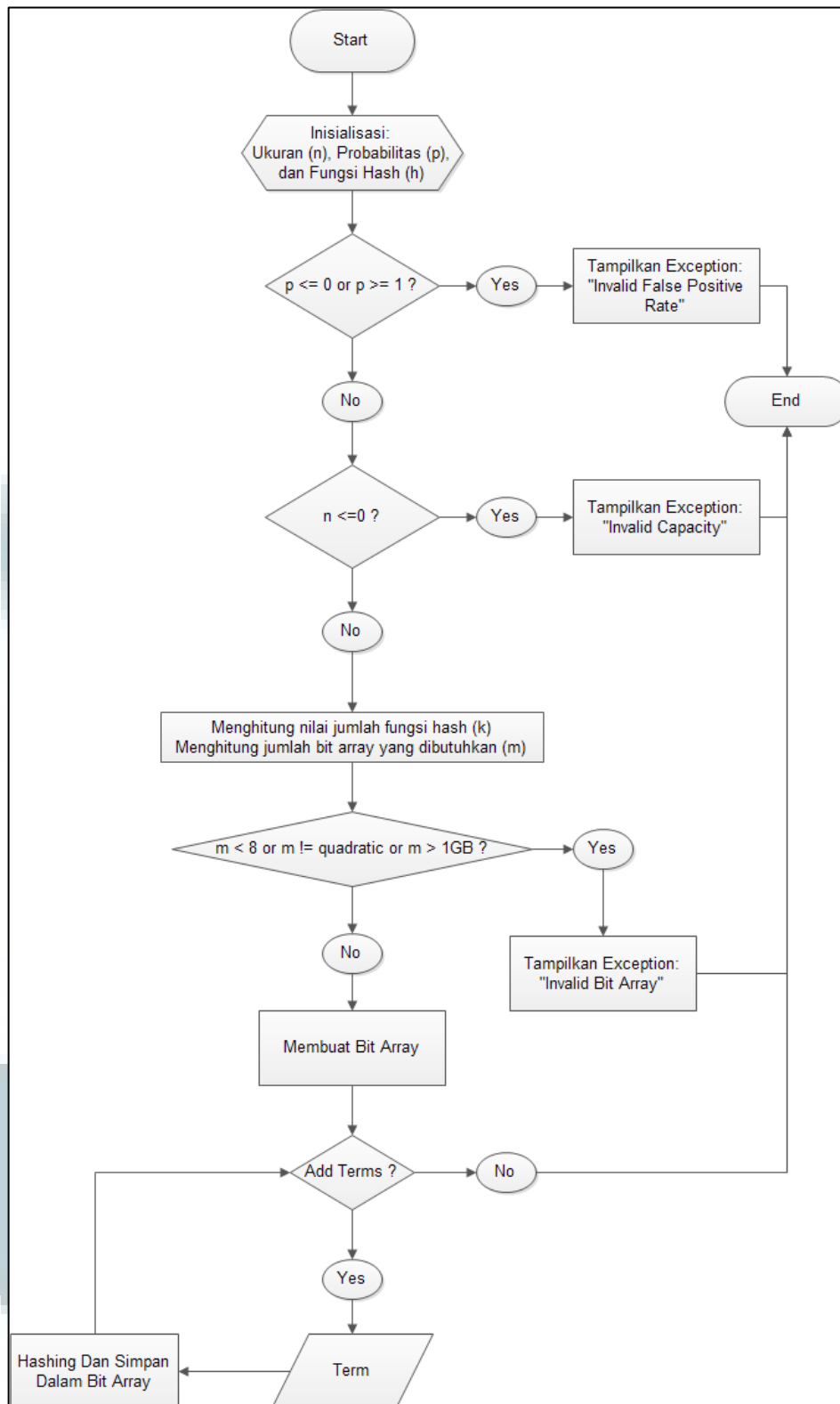
C.7 Flowchart Subproses Bloom Filter

Subproses *Bloom Filter* diawali dengan inisialisasi beberapa variabel seperti ukuran *filter* (n), probabilitas yang diinginkan (p), dan fungsi *hash* (h) yang digunakan. Kemudian proses akan melakukan pemeriksaan terhadap nilai variabel tersebut. Jika tidak sesuai dengan ketentuan, maka proses akan menampilkan pesan *exception*.

Tahapan berikutnya adalah mencari tahu berapa jumlah fungsi hash optimal (k) dan banyaknya kapasitas optimal *filter* (m). Sebelum proses berlanjut ke tahapan selanjutnya kedua variabel ini akan diperiksa terlebih dahulu. Jika tidak sesuai dengan ketentuan, maka proses akan menampilkan pesan *exception*. Ketika semua kebutuhan sudah terpenuhi proses akan membuat suatu ruang penyimpanan yang disebut *bit array*. *Bit array* inilah yang nantinya digunakan untuk memasukkan istilah dan melakukan perbandingan untuk mengetahui keberadaan suatu istilah.



UMN



Gambar 3.19 Flowchart Subproses Bloom Filter

D. Struktur Tabel

Sistem ini dibangun dengan menggunakan dua buah basis data yaitu `journal_db` dan `index_db`. Kedua basis data tersebut dibuat menggunakan MySQL server yang terdiri dari beberapa tabel sebagai berikut.

D.1 Basis data `journal_db`

1. Tabel Penulis

Nama tabel : `author_tb`

Fungsi : menyimpan nama-nama penulis paper.

Tabel 3.1 Struktur tabel `author_db`

Nama Kolom	Tipe Data	Panjang	Informasi
<code>author_id</code>	INT	11	<i>primary key, not null, auto increment</i>
<code>author_first_name</code>	VARCHAR	100	nama depan penulis
<code>author_last_name</code>	VARCHAR	100	nama belakang penulis

2. Tabel Penerbit

Nama tabel : `publisher_tb`

Fungsi : menyimpan nama-nama penerbit jurnal.

Tabel 3.2 Struktur tabel `publisher_tb`

Nama Kolom	Tipe Data	Panjang	Informasi
<code>publisher_id</code>	INT	11	<i>primary key, not null, auto increment</i>
<code>publisher_name</code>	VARCHAR	50	nama penerbit

3. Tabel Judul Jurnal

Nama tabel : `journal_title_tb`

Fungsi : menyimpan judul jurnal.

Tabel 3.3 Struktur tabel journal_title_tb

Nama Kolom	Tipe Data	Panjang	Informasi
journal_title_id	INT	11	<i>primary key, not null, auto increment</i>
journal_title	VARCHAR	50	judul jurnal

4. Tabel Konten Jurnal

Nama tabel : journal_content_tb

Fungsi : menyimpan informasi konten dari setiap jurnal.

Tabel 3.4 Struktur tabel journal_content_tb

Nama Kolom	Tipe Data	Panjang	Informasi
journal_content_id	INT	11	<i>primary key, not null, auto increment</i>
journal_title_id	INT	11	<i>foreign key, not null</i>
publisher_id	INT	11	<i>foreign key, not null</i>
journal_issn	VARCHAR	30	nomor issn
journal_year	YEAR	4	tahun publikasi
journal_subject	LONGTEXT	-	subyek atau topik jurnal
journal_scope	LONGTEXT	-	ruang lingkup jurnal
journal_contacts	LONGTEXT	-	kontak jurnal

5. Tabel Volume Jurnal

Nama tabel : journal_volume_tb

Fungsi : menyimpan volume setiap jurnal

Tabel 3.5 Struktur tabel journal_volume_tb

Nama Kolom	Tipe	Panjang	Informasi
journal_volume_id	INT	11	<i>primary key, not null, auto increment</i>
journal_content_id	INT	11	<i>foreign key, not null</i>
volume_id	INT	11	volume jurnal

6. Tabel Nomor Jurnal

Nama tabel : volume_issue_tb

Fungsi : menyimpan nomor jurnal

Tabel 3.6 Struktur tabel volume_issue_tb

Nama Kolom	Tipe Data	Panjang	Informasi
volume_issue_id	INT	11	<i>primary key, not null, auto increment</i>
journal_volume_id	INT	11	<i>foreign key, not null</i>
issue_id	INT	11	nomor jurnal
issue_cover	VARCHAR	100	gambar sampul jurnal

7. Tabel Paper

Nama tabel : paper_tb

Fungsi : menyimpan informasi *paper*.

Tabel 3.7 Struktur tabel paper_tb

Nama Kolom	Tipe Data	Panjang	Informasi
paper_id	INT	11	<i>primary key, not null, auto increment</i>
volume_issue_id	INT	11	<i>foreign key, not null</i>
paper_title	VARCHAR	500	judul paper
paper_date	DATE	-	tanggal paper
paper_keywords	LONGTEXT	-	kata kunci yang berkaitan dengan isi paper
paper_abstract	LONGTEXT	-	abstrak paper
paper_pdf	VARCHAR	100	paper dalam bentuk file
paper_word	VARCHAR	100	paper dalam bentuk file
paper_download	BOOLEAN	-	izin untuk mengunduh

8. Tabel Penulis Paper

Nama tabel : paper_author_tb

Fungsi : menyimpan relasi penulis dengan *paper* yang dibuatnya.

Tabel 3.8 Struktur tabel paper_author_tb

Nama Kolom	Tipe	Panjang	Informasi
paper_author_id	INT	11	<i>primary key, not null, auto increment</i>
paper_id	INT	11	<i>foreign key, not null</i>
author_id	INT	11	<i>foreign key, not null</i>

9. Tabel User

Nama tabel : user_tb

Fungsi : menyimpan informasi untuk otentikasi administrator jurnal.

Tabel 3.9 Struktur tabel user_tb

Nama	Tipe Data	Panjang	Informasi
username	VARCHAR	30	<i>primary key, not null, auto increment</i>
password	VARCHAR	32	<i>not null</i>

10. Tabel Konfigurasi

Nama tabel : global_settings_tb

Fungsi : menyimpan konfigurasi jurnal.

Tabel 3.10 Struktur tabel global_settings_tb

Nama Kolom	Tipe Data	Panjang	Informasi
setting_id	INT	11	<i>primary key, not null, auto increment</i>
setting_key	VARCHAR	100	nama konfigurasi
setting_value	VARCHAR	30	isi konfigurasi

D.2 Basis data index_db

1. Tabel indeks penulis

Nama tabel : author_tb

Fungsi : menyimpan indeks istilah dari tabel journal_db.author_tb

Tabel 3.11 Struktur tabel author_tb

Nama Kolom	Tipe Data	Panjang	Informasi
index_id	INT	11	<i>primary key, not null, auto increment</i>
term	VARCHAR	30	istilah yang diindeks
tuple	INT	11	tuple id dimana istilah tersebut muncul
author_first_name	INT	11	panjang istilah pada nama depan
author_last_name	INT	11	panjang istilah pada nama belakang

2. Tabel indeks paper

Nama tabel : paper_tb

Fungsi : menyimpan indeks istilah dari tabel journal_db.paper_tb

Tabel 3.12 Struktur tabel paper_tb

Nama Kolom	Tipe Data	Panjang	Informasi
index_id	INT	11	<i>primary key, not null, auto increment</i>
term	VARCHAR	30	istilah yang diindeks
tuple	INT	11	tuple id dimana istilah tersebut muncul
paper_title	INT	11	panjang istilah pada judul paper
paper_date	INT	11	panjang istilah pada tanggal paper
paper_keywords	INT	11	panjang istilah pada kata kunci paper
paper_abstract	INT	11	panjang istilah pada abstrak paper

3. Tabel kata dasar bahasa Indonesia

Nama tabel : rootwords_tb

Fungsi : menyimpan kata dasar bahasa Indonesia

Tabel 3.13 Struktur tabel rootwords_tb

Nama Kolom	Tipe Data	Panjang	Informasi
rootwords_id	INT	11	<i>primary key, not null, auto increment</i>
rootword	VARCHAR	20	kata dasar
rootwords_type	VARCHAR	20	jenis kata dasar

4. Tabel stopwords bahasa Indonesia

Nama tabel : stopwords_tb_id

Fungsi : menyimpan stopwords bahasa Indonesia

Tabel 3.14 Struktur tabel stopwords_tb_id

Nama Kolom	Tipe Data	Panjang	Informasi
stopword_id	INT	10	<i>primary key, not null, auto increment</i>
stopword	VARCHAR	50	<i>stopword</i>

5. Tabel stopwords bahasa Inggris

Nama tabel : stopwords_tb_en

Fungsi : menyimpan stopwords bahasa Inggris

Tabel 3.15 Struktur tabel stopwords_tb_en

Nama Kolom	Tipe Data	Panjang	Informasi
stopword_id	INT	10	<i>primary key, not null, auto increment</i>
stopword	VARCHAR	50	<i>stopword</i>

6. Tabel statistik global

Nama tabel : gl_stat_tb

Fungsi : menyimpan data statistik global dari setiap tabel yang diindeks.

Tabel 3.16 Struktur tabel gl_stat_tb

Nama Kolom	Tipe Data	Panjang	Informasi
col_name	VARCHAR	200	Nama kolom dari tabel yang diindeks
num_of_docs	INT	11	Jumlah baris dari tabel yang diindeks
avgdl	DOUBLE	-	Panjang rata-rata istilah dari kolom tabel yang diindeks

7. Tabel paper cache

Nama tabel : paper_cache_tb

Fungsi : menyimpan hasil perhitungan nilai similarity dari paper yang sudah pernah dicari untuk meningkatkan performa sistem.

Tabel 3.17 Struktur tabel paper_cache_tb

Nama Kolom	Tipe Data	Panjang	Informasi
paper_cache_id	INT	11	<i>primary key, not null, auto increment</i>
query_cache_id	INT	11	<i>foreign key, not null</i>
paper_id	INT	11	<i>foreign key, not null</i>
score	DOUBLE	-	<i>nilai similarity</i>

8. Tabel query cache

Nama tabel : query_cache_tb

Fungsi : menyimpan query-query masukkan dari pengguna.

Tabel 3.18 Struktur tabel query_cache_tb

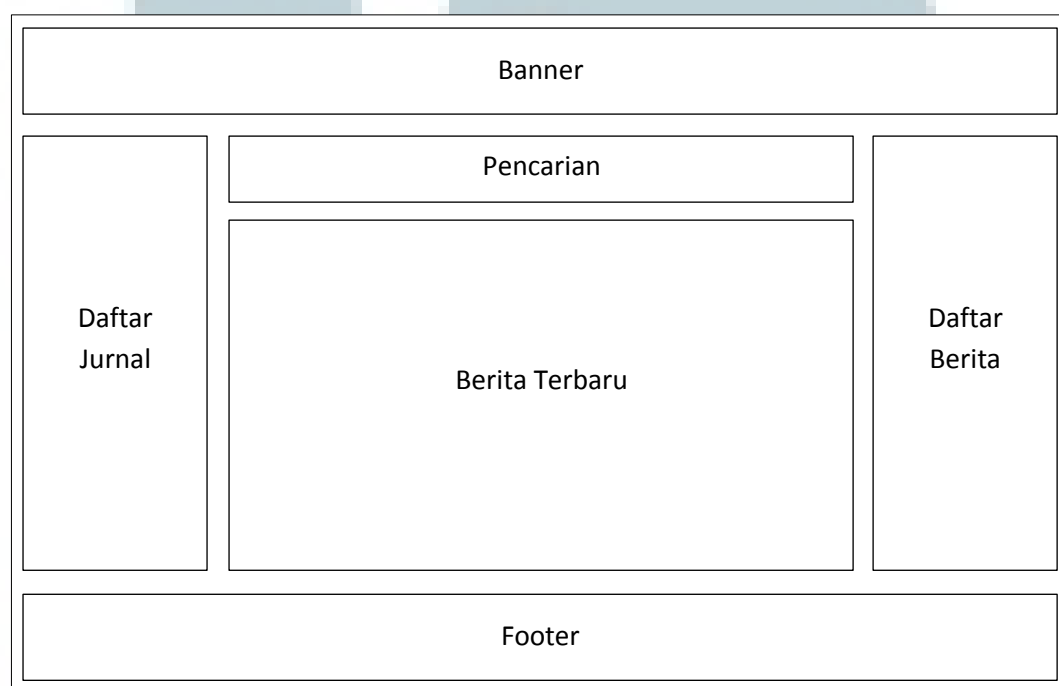
Nama Kolom	Tipe Data	Panjang	Informasi
query_cache_id	INT	11	<i>primary key, not null, auto increment</i>
cached_query	LONGTEXT	-	<i>query pengguna</i>

3.3.2 Sketsa Antarmuka

A. Sketsa Antarmuka Pengguna Akhir

Pada subbab berikut ditampilkan sketsa antarmuka sistem dari sisi pengguna akhir. Setiap antarmuka memiliki beberapa komponen pendukung yang secara konsisten ditampilkan pada halaman-halaman lainnya. Komponen tersebut antara lain adalah *banner*, *footer*, pencarian, daftar jurnal, dan daftar berita.

1. Halaman Utama

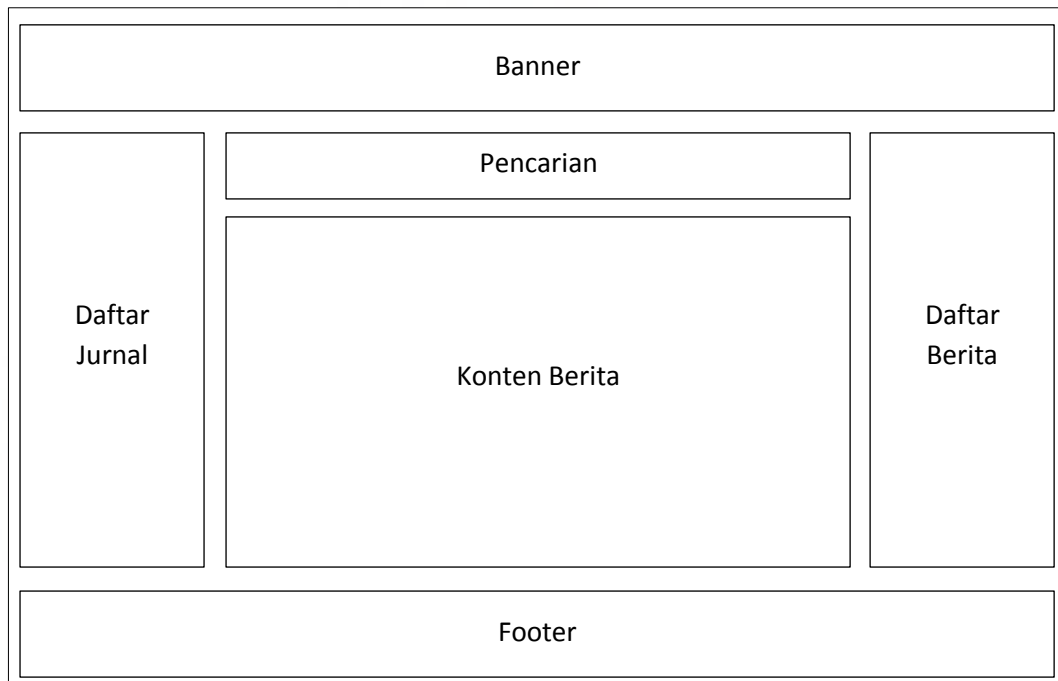


Gambar 3.20 Sketsa Antarmuka Halaman Utama

2. Halaman Berita

Halaman ini dapat diakses melalui dua cara, yaitu melalui komponen berita terbaru ataupun dengan komponen daftar berita. Di halaman ini pengguna dapat melihat mengenai isi berita, penulis berita, tanggal *posting* berita, dan *user path*. *User path* selain digunakan untuk memberitahu lokasi pengguna saat ini,

komponen ini juga dapat digunakan sebagai navigasi pengguna ke halaman-halaman sebelumnya.

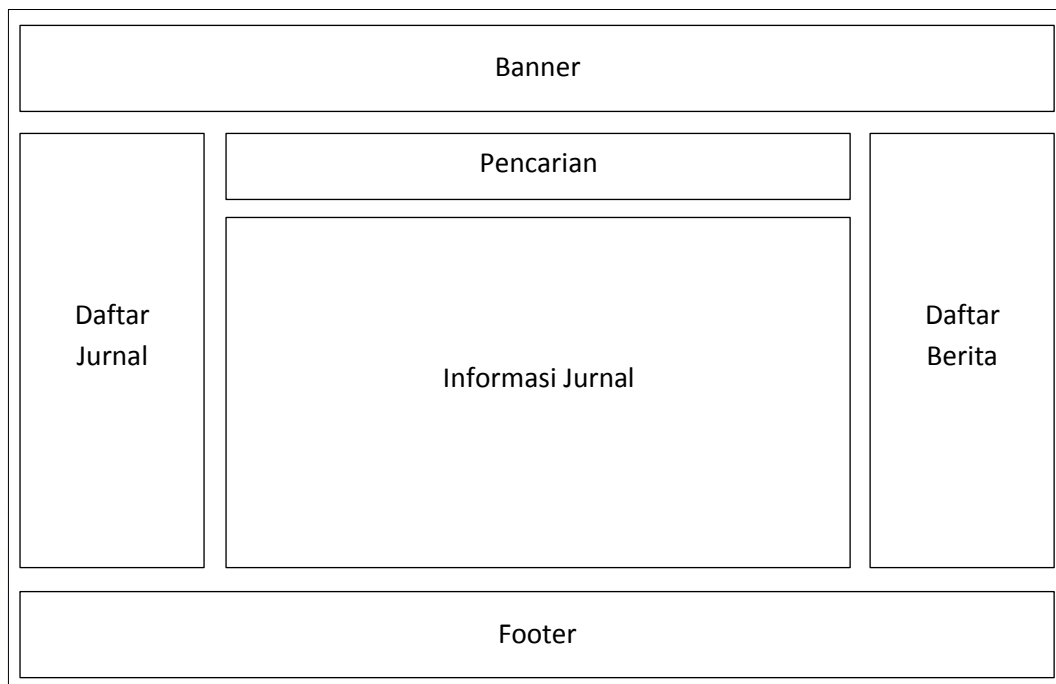


Gambar 3.21 Sketsa Antarmuka Halaman Berita

3. Halaman Informasi Jurnal

Halaman ini menyajikan informasi mengenai topik, ruang lingkup, kontak, dan *cover* jurnal. Selain itu pengguna dapat mengubah informasi yang ditampilkan berdasarkan tahun terbit, volume, dan nomor jurnal dengan memilih opsi dari *dropdown menu* yang telah disediakan.

Untuk mengakses konten dari jurnal, pengguna dapat menekan tombol “lihat konten”. Pengguna akan diarahkan ke halaman konten jurnal yang berisi *paper-paper* sesuai dengan tahun terbit, volume, dan nomor jurnal yang dipilihnya.

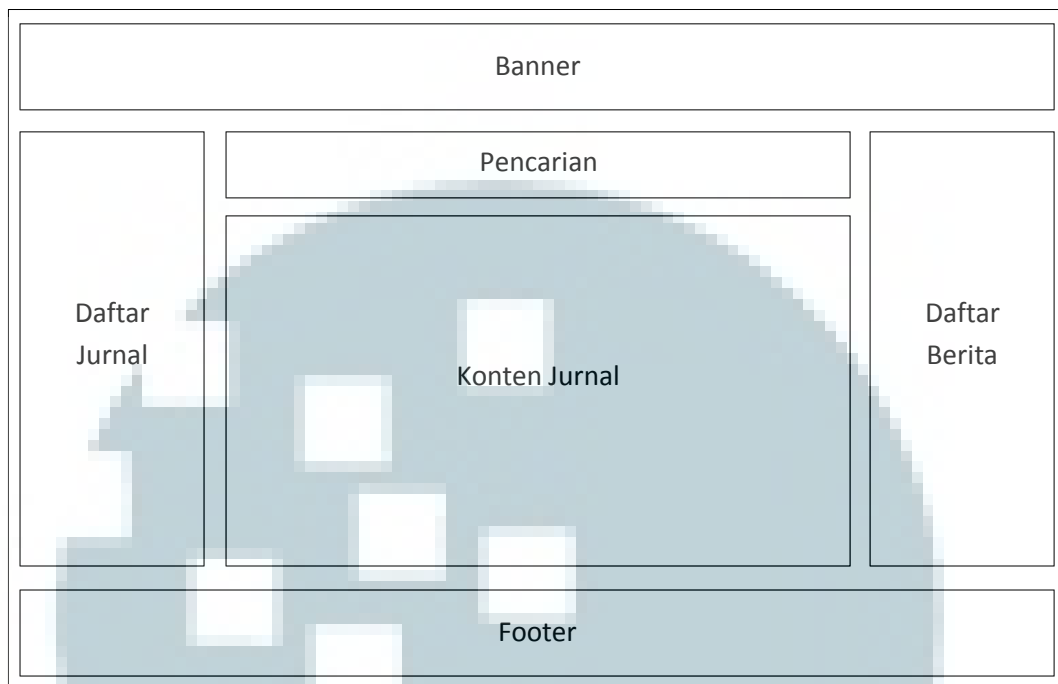


Gambar 3.22 Sketsa Informasi Jurnal

4. Halaman Konten Jurnal

Halaman ini akan menampilkan *paper* sesuai dengan tahun terbit, volume dan nomor jurnal yang telah dipilih oleh pengguna pada halaman sebelumnya (halaman jurnal). Informasi mengenai *paper* hanya terbatas pada judul *paper*, nama penulis, tanggal publikasi, kata kunci, dan sebagian abstrak *paper*.

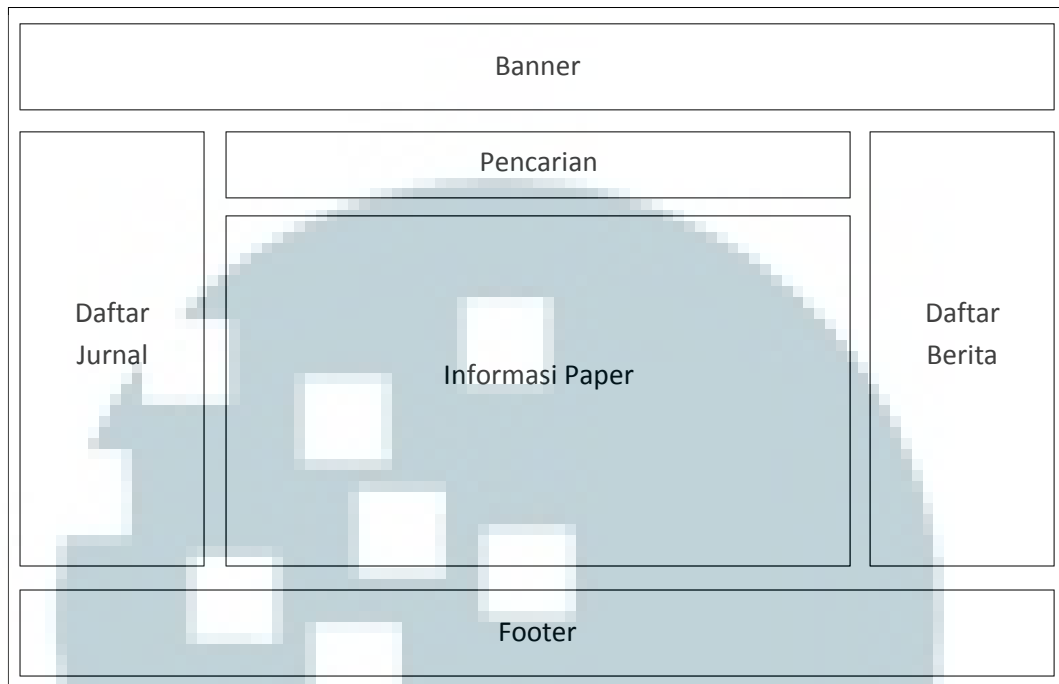
Setiap halaman secara *default* hanya menampilkan lima buah *paper*, namun jumlah banyaknya *paper* yang ditampilkan dapat dikonfigurasi oleh administrator pada menu *content management system*. Untuk menampilkan daftar *paper* yang lain, pengguna dapat mengaksesnya melalui menu navigasi *pagination* yang berada di sebelah kanan bawah judul jurnal.



Gambar 3.23 Sketsa Antarmuka Konten Jurnal

5. Halaman *Paper*

Halaman ini menyajikan informasi mengenai judul *paper*, penulis *paper*, kata kunci, abstrak, dan menu untuk mengunduh *paper*. Pada menu unduh *paper*, pengguna dapat mengunduh *paper* dalam dua buah format *file* berbeda, yaitu microsoft word dan pdf. Administrator dapat menentukan apakah *file paper* tersebut dapat diunduh oleh pengguna atau tidak melalui menu *content management system*.

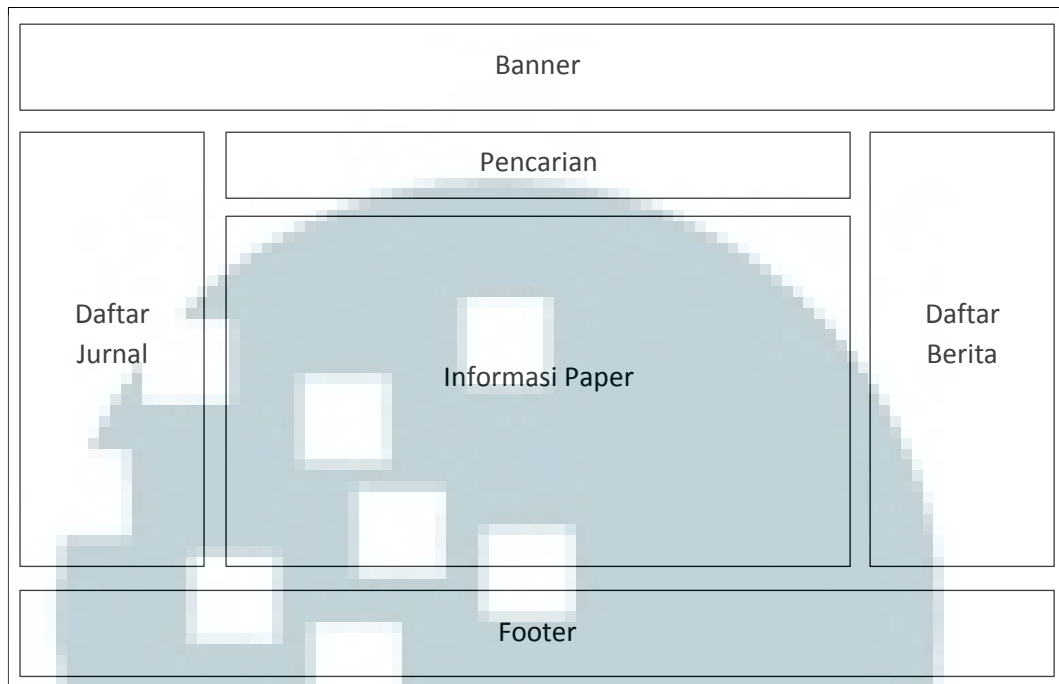


Gambar 3.24 Sketsa Antarmuka Halaman *Paper*

6. Halaman *Search*

Halaman ini dapat diakses melalui dua cara, yaitu dengan menggunakan fitur *simple search* dan *advanced search*. *Paper* yang ditampilkan adalah *paper-paper* yang berkaitan dengan *query* yang diberikan oleh pengguna. Jumlah *paper* yang ditampilkan pada halaman ini sama seperti pada halaman konten jurnal.

UMMN



Gambar 3.25 Sketsa Antarmuka Halaman Search

7. Halaman RSS

Halaman ini dapat diakses melalui *icon* RSS yang terletak di sebelah kanan atas daftar jurnal. Ketika *icon* ini ditekan, pengguna akan diarahkan ke halaman berikutnya yang berisi informasi mengenai 50 *paper* terbaru dalam format XML. Beberapa *browser* dapat langsung menampilkan menu berlangganan RSS sehingga pengguna selalu mendapatkan informasi terbaru mengenai *paper-paper* yang dipublikasikan.



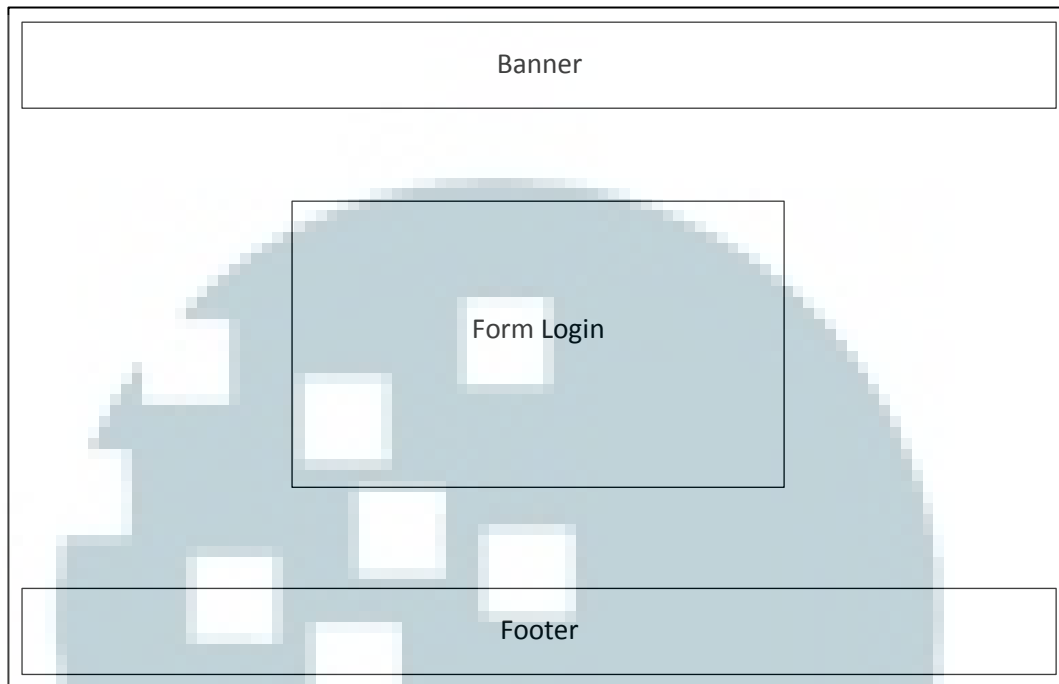
Gambar 3.26 Sketsa Antarmuka Halaman RSS

B. Sketsa Antarmuka Administrator

Pada subbab berikut ditampilkan sketsa antarmuka sistem dari sisi administrator. Setiap antarmuka memiliki beberapa komponen pendukung yang secara konsisten ditampilkan pada halaman-halaman lainnya. Komponen tersebut antara lain adalah *banner*, *footer*, *panel navigasi*, dan *panel kontrol*.

1. Halaman Login

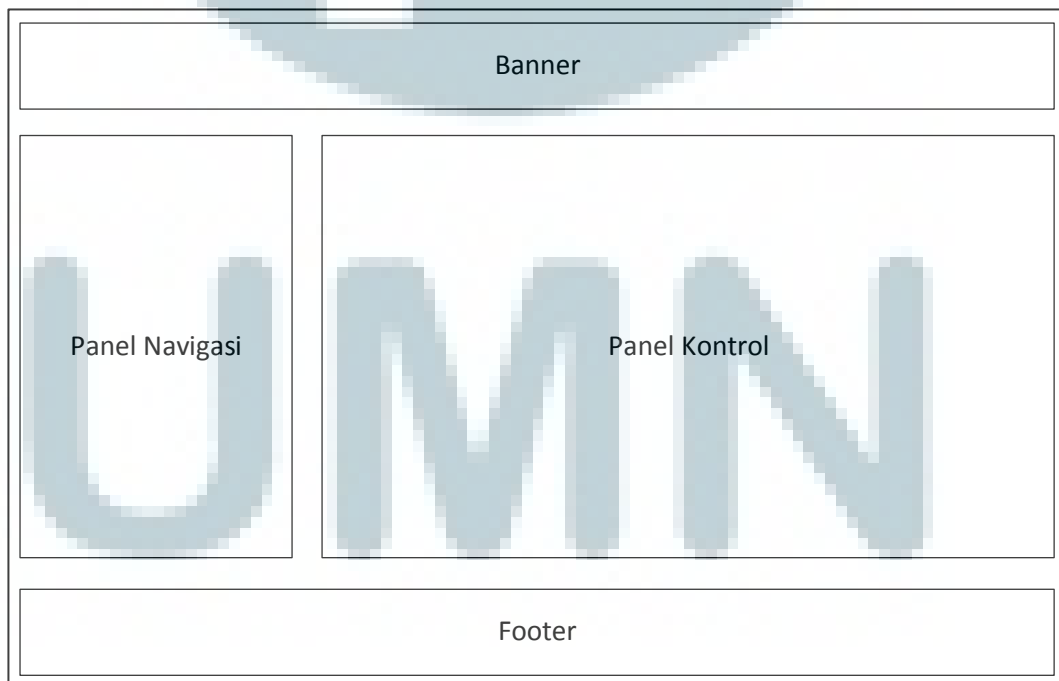
Di halaman ini akan tampil sebuah *form login* dimana seorang administrator harus melakukan otentikasi terlebih dahulu agar dapat mengakses menu *content management system*.



Gambar 3.27 Sketsa Antarmuka Halaman Login

2. Halaman *Content Management System*

Pada halaman ini terdapat menu operasi CRUD dan konfigurasi.



Gambar 3.28 Sketsa Antarmuka Halaman CMS

3. Halaman Judul Jurnal, Penerbit, Konten Jurnal, Volume, Nomor, Penulis, *Paper*, dan Berita

Pada halaman ini administrator dapat memilih beberapa menu kontrol seperti tambah, ubah dan hapus (gambar 3.29-3.31). Halaman hapus hanya berupa pesan konfirmasi yang menyakinkan administrator bahwa apakah prosedur ini akan dieksekusi. Sedangkan halaman tambah dan ubah, berupa *form* isian dimana ketika tombol *submit* ditekan data-data tersebut akan divalidasi terlebih dahulu kemudian dimasukkan ke dalam *database*. Diakhir proses, sistem akan menampilkan pesan konfirmasi yang memberitahu administrator apakah operasi tersebut berjalan dengan lancar atau tidak.

4. Halaman Konfigurasi

Halaman konfigurasi dapat diakses melalui panel navigasi yang ada di sebelah kiri halaman *content management system*. Ketika pengguna memilihnya panel kontrol akan mengisi konten secara dinamis. Di dalam panel kontrol, Administrator dapat mengakses lima buah kontrol utama untuk melakukan penghapusan berkas otomatis (*run janitor*), perubahan *password* (*change password*), pengindeksan (*indexing*), perubahan konfigurasi (*preferences*), dan optimasi tabel (*optimize tables*) yang akan dijelaskan lebih detil sebagai berikut.

4.1 Halaman Penghapus Berkas Otomatis

Halaman ini dapat diakses dengan menekan tombol *run janitor* dan akan memunculkan sebuah pesan konfirmasi untuk meyakinkan administrator apakah prosedur ini akan dieksekusi (gambar 3.32). Ketika tombol *yes* ditekan prosedur penghapusan *file* akan dijalankan dan diakhiri dengan munculnya pesan

konfirmasi yang memberitahukan apakah prosedur tersebut berjalan dengan lancar atau tidak.

4.2 Halaman Ubah Password

Halaman ini dapat diakses dengan menekan tombol *change password* dan akan memunculkan sebuah *form* yang terdiri dari satu buah label berisi *username*, tiga buah *textfield* untuk memasukkan *password* lama, *password* baru, dan konfirmasi *password* baru, serta satu buah tombol *submit* untuk menyimpan data tersebut (gambar 3.33). Ketika tombol *submit* ditekan, akan muncul pesan konfirmasi yang memberitahukan apakah data tersebut berhasil disimpan atau tidak.

4.3 Halaman *Indexing*

Halaman ini dapat diakses dengan menekan tombol *indexing* dan akan memunculkan sebuah pesan konfirmasi untuk meyakinkan administrator apakah prosedur ini akan dieksekusi (gambar 3.32). Ketika tombol *yes* ditekan prosedur *indexing database* akan dijalankan dan diakhiri dengan munculnya pesan konfirmasi yang memberitahukan apakah prosedur tersebut berjalan dengan lancar atau tidak.

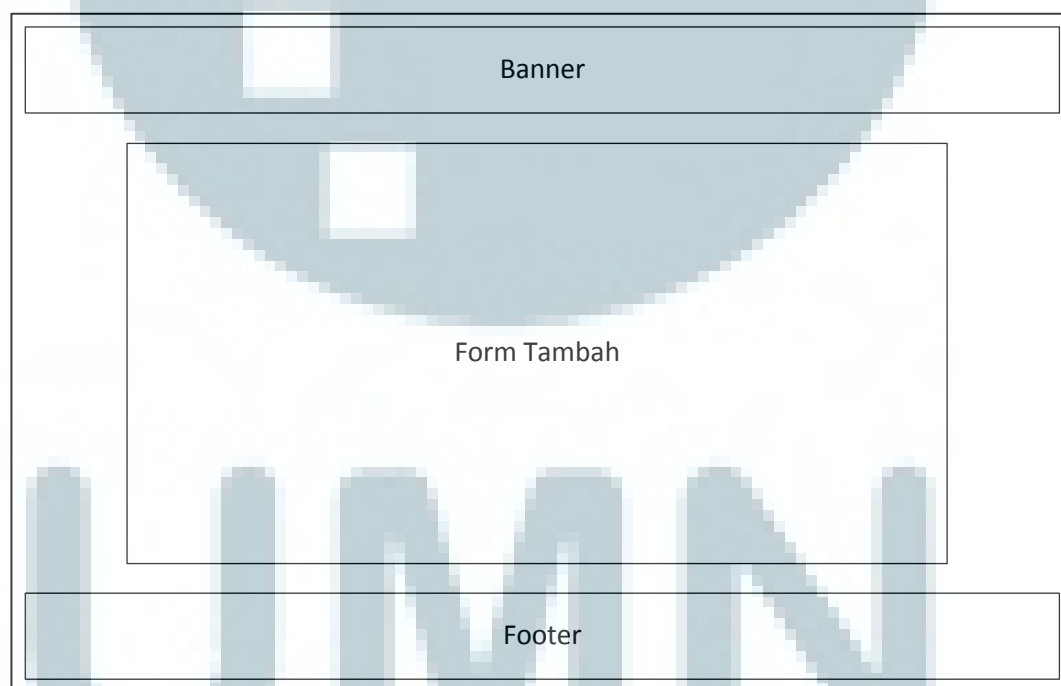
4.4 Halaman *Preferences*

Halaman ini dapat diakses dengan menekan tombol *preferences* dan akan memunculkan sebuah *form* yang terdiri dari tiga buah *textfield* untuk memasukkan *jQuery Timeout*, *jQuery Animation Speed*, dan *Pagination*, empat buah *radio button* untuk menentukan apakah fitur *profiler* dan *similarity* ingin ditampilkan atau tidak, serta satu buah tombol *submit* untuk menyimpan data tersebut (gambar

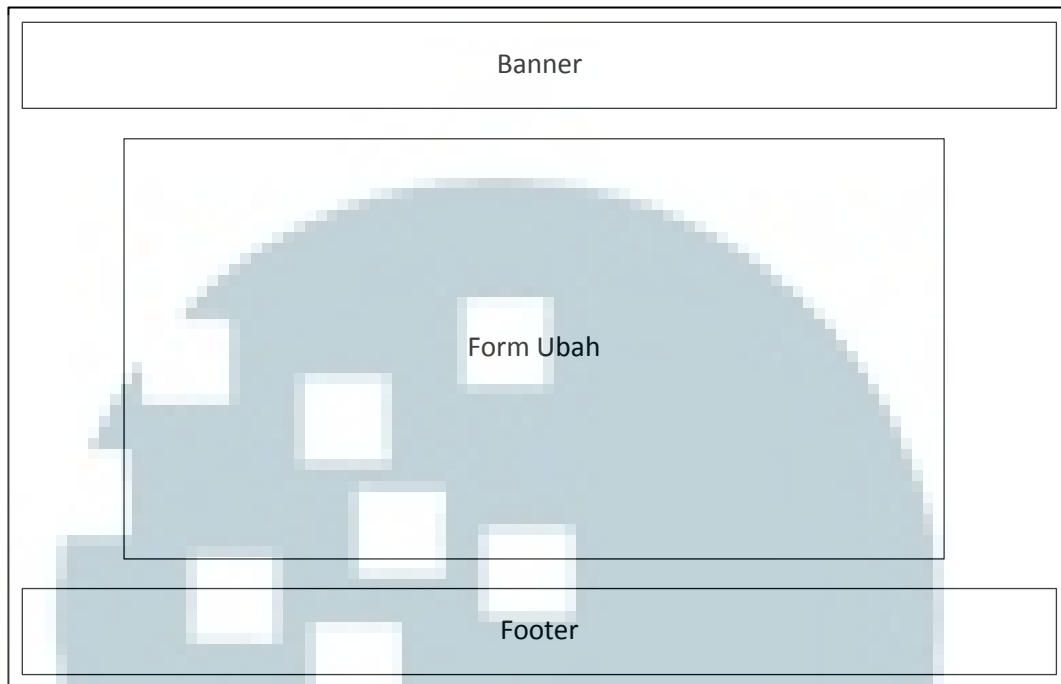
3.34). Ketika tombol *submit* ditekan, akan muncul pesan konfirmasi yang memberitahukan apakah data tersebut berhasil disimpan atau tidak.

4.5 Halaman Optimasi Tabel

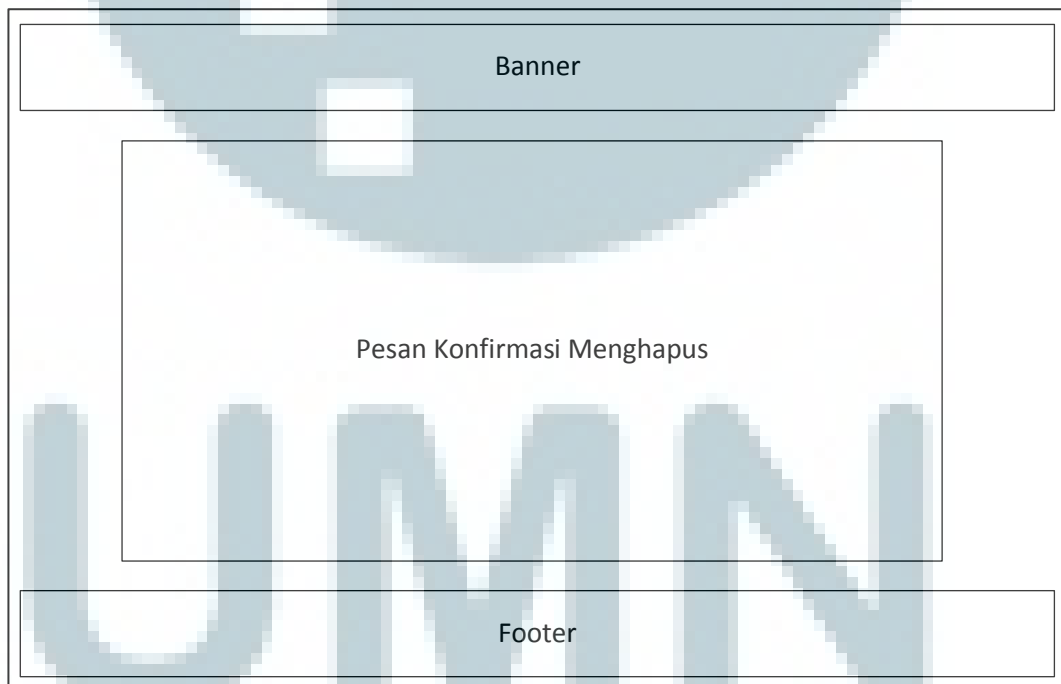
Halaman ini dapat diakses dengan menekan tombol *optimize tables* dan akan memunculkan sebuah pesan konfirmasi untuk meyakinkan administrator apakah prosedur ini akan dieksekusi (gambar 3.32). Ketika tombol *yes* ditekan prosedur optimasi tabel akan dijalankan dan diakhiri dengan munculnya pesan konfirmasi yang memberitahukan apakah prosedur tersebut berjalan dengan lancar atau tidak.



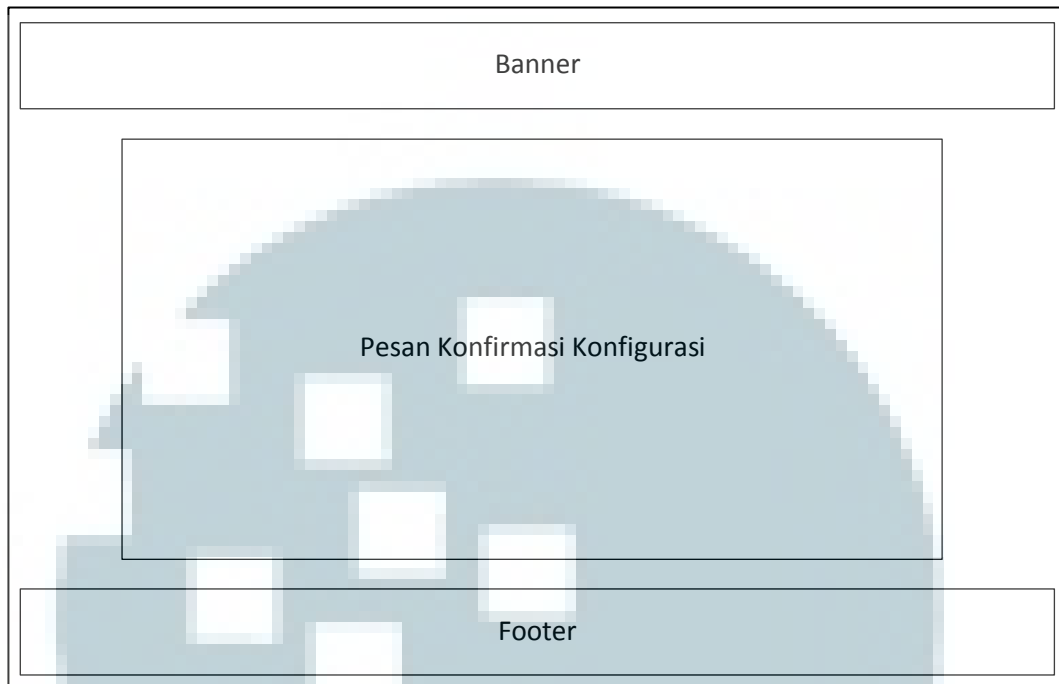
Gambar 3.29 Sketsa Antarmuka Tambah



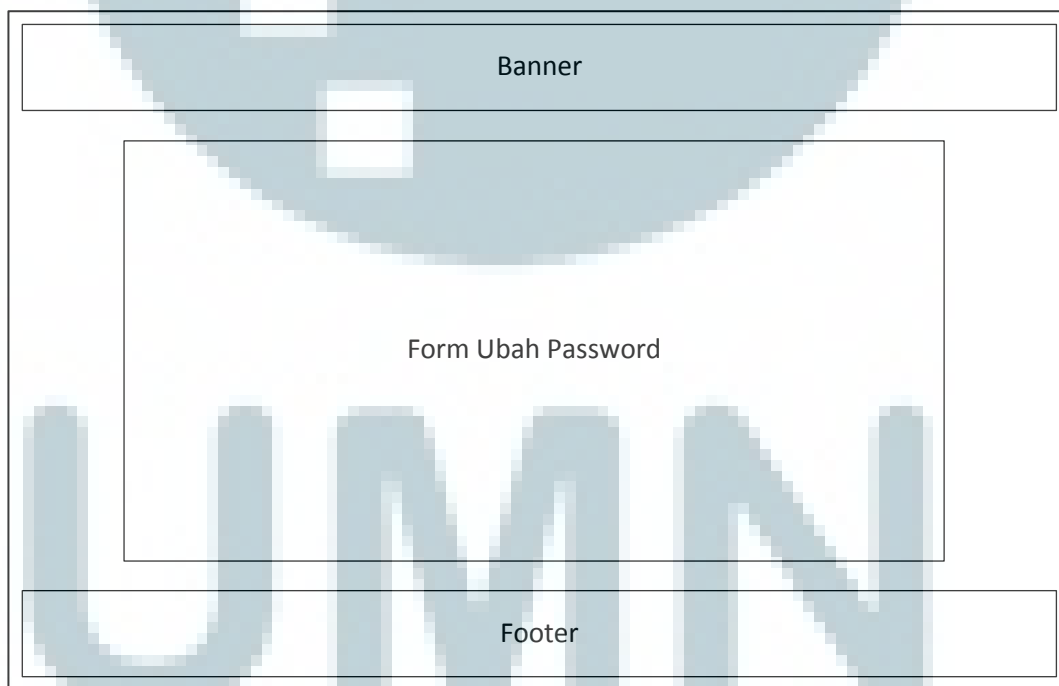
Gambar 3.30 Sketsa Antarmuka Ubah



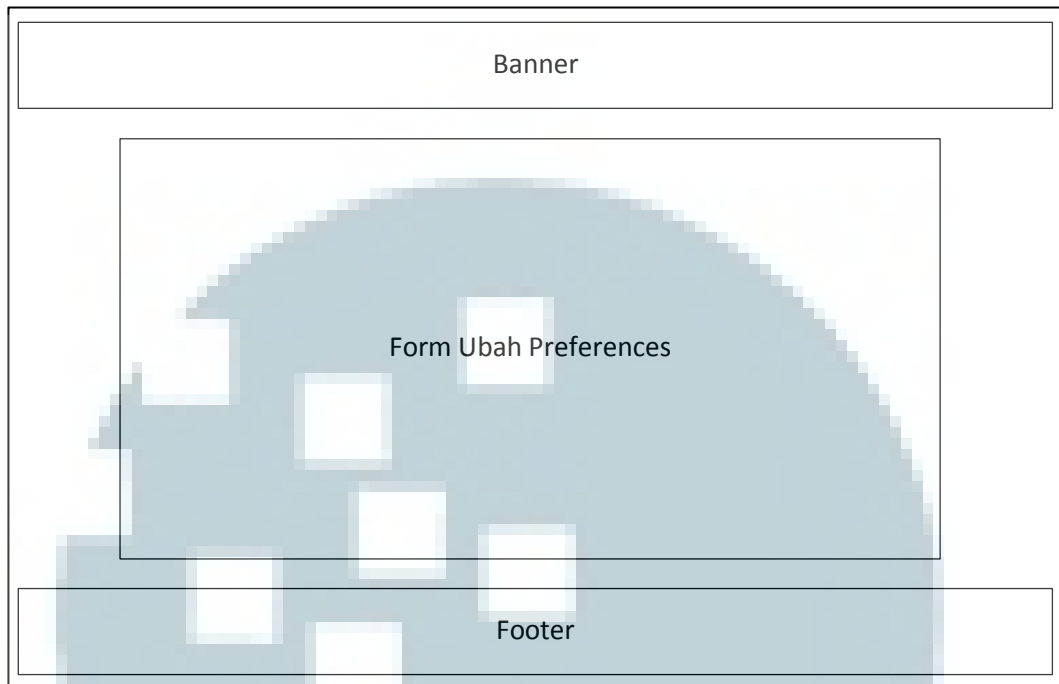
Gambar 3.31 Sketsa Antarmuka Hapus



Gambar 3.32 Sketsa Antarmuka Konfirmasi Konfigurasi



Gambar 3.33 Sketsa Antarmuka Ubah password



Gambar 3.34 Sketsa Antarmuka Pengubahan Preferences

UMMN