

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Tujuan Operasional Penelitian**

Tujuan dari penelitian ini adalah untuk mengembangkan dan mengintegrasikan sistem pencarian *full-text* karya akhir berdasarkan abstrak di sistem informasi perpustakaan Jurusan Teknik Elektro.

#### **3.2 Tempat dan Waktu Penelitian**

Penelitian dilakukan di laboratorium multimedia Jurusan Teknik Elektro Universitas Negeri Jakarta sejak bulan September 2012 hingga Desember 2012.

#### **3.3 Definisi Kebutuhan (*Requirements Definition*)**

Tahap *requirements definition* pada sistem pencarian *full-text* karya akhir berdasarkan abstrak di sistem informasi perpustakaan Jurusan Teknik Elektro memiliki fitur yang memudahkan mahasiswa dalam memperoleh informasi-informasi karya akhir berdasarkan abstrak. *Full-text searching* adalah fitur yang dapat mencari dengan cepat namun relevan.

Pada tahap definisi kebutuhan, langkah yang dilakukan adalah melakukan pengembangan sistem pencarian menggunakan *full-text*. Sistem pencarian *full-text* hanya digunakan oleh admin dan user (mahasiswa).

Kebutuhan fungsional harus mendefinisikan aksi dasar yang harus diambil oleh sistem untuk menerima dan memproses masukan dan menghasilkan keluaran.

Tabel 3.1 Daftar Kebutuhan Fungsional

Aspek	Kebutuhan
Content	Suatu aplikasi perangkat lunak untuk membantu dalam pencarian karya akhir berdasarkan abstrak pada sistem informasi perpustakaan.
	Memberikan informasi karya akhir berdasarkan abstrak kepada mahasiswa Jurusan Teknik Elektro.
User	Mahasiswa Jurusan Teknik Elektro dapat mencari karya akhir berdasarkan abstrak pada sistem informasi perpustakaan.
Admin	Admin dapat mengedit data, meng- <i>input</i> dan mengelola data karya akhir berdasarkan abstrak.
	Admin dapat memberikan hak akses kepada mahasiswa yang ingin mencari karya akhir berdasarkan abstrak

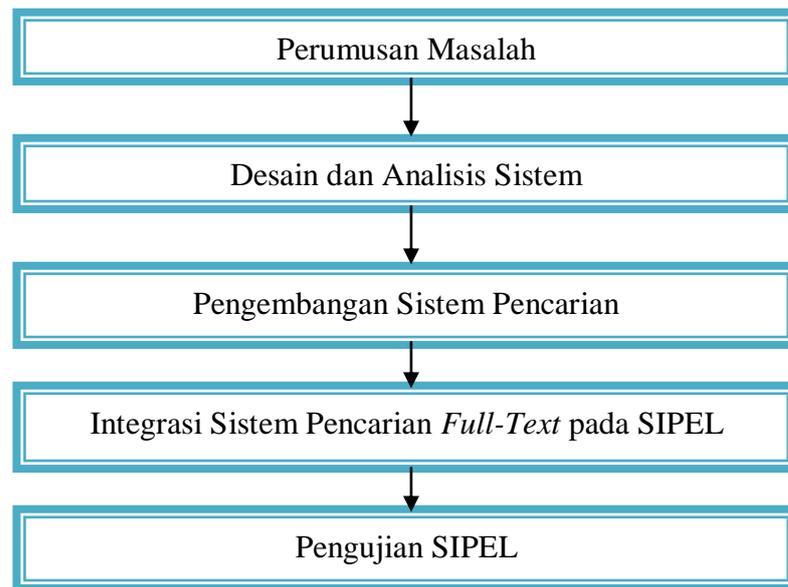
Pada pembuatan sistem pencarian *full-text* dibutuhkan perangkat keras dan perangkat lunak diantaranya adalah :

1. Perangkat Keras
  - a. Processor Intel® Core™ i3-350M Processor (2.26 GHz, Cache 3 MB).
  - b. Memori 2 GB DDR3
  - c. Harddisk 320 GB, *free space* 11 GB
  - d. Monitor 14"
2. Perangkat Lunak
  - a. *Operating System Zorin 6.1 for Linux*
  - b. *XAMPP for Linux (LAMPP) 1.7.7*
  - c. *Software Bluefish Editor 2.2*
  - d. *MySQL Database 5.0*

e. *PHP 5*

### 3.4 Metode Penelitian

Bahasan tahapan pengembangan terangkum dalam bagan berikut :



Gambar 3.1 Bagan Metode Penelitian

Perumusan masalah membahas latar belakang masalah dan konsep dasar penyelesaian masalah. Desain dan analisis sistem membahas kebutuhan detail dari sistem pencarian. Pengembangan sistem pencarian membahas perbandingan klausa pencarian pada bahasa SQL, penyusunan dan pengujian koleksi tester, analisis hasil. Integrasi Sistem Pencarian *Full-Text* pada SIPEL membahas fitur *full-text searching*, integrasi *database*, integrasi *interface* dan pengujian SIPEL.

### 3.4.1 Perumusan Masalah

Dengan berkembangnya jumlah koleksi yang ada di sistem perpustakaan elektro, semakin meningkatnya keperluan untuk mencari informasi karya akhir, khususnya bagi mahasiswa yang sudah mengambil karya akhir, dibutuhkan adanya mesin pencari yang dapat mencari dengan cepat namun relevan. Sistem informasi perpustakaan yang sudah dikembangkan menyediakan fasilitas pencarian dengan menggunakan teknik pencarian *database relational*. Dengan menggunakan perintah SQL klausa LIKE, ORDER dan WHERE, pencarian dengan teknik tersebut akan menampilkan seluruh informasi yang sesuai terurut berdasarkan urutan abjad bagi atribut-atribut skripsi yang dipilih seperti nama, penulis dan judul. Hal ini menyebabkan hasil pencarian tidak dapat segera mengarahkan pencari kepada informasi yang benar-benar dicari. Dengan membuat *program full-text searching* pada mesin pencari di sistem informasi perpustakaan, mesin pencari bisa mengarahkan *user* kepada dokumen atau informasi karya akhir yang benar-benar dicari dan terurut berdasarkan nilai relevansi, dimana mengambil semua dokumen yang relevan pada saat yang sama mungkin dengan mengambil beberapa non-relevan<sup>1</sup>, sehingga hasil pertama yang ditampilkan diharapkan paling relevan. Program *Full-text searching* adalah bagaimana menginterpretasikan permasalahan yang timbul di Perpustakaan Jurusan Teknik Elektro Universitas Negeri Jakarta sebagaimana yang telah penulis kemukakan pada Bab 1 pada sub bab perumusan masalah dan batasan masalah. Permasalahan yang timbul adalah bagaimana mengembangkan sistem pencarian karya akhir

---

<sup>1</sup> Van RIJSBERGEN, C.J., *Information Retrieval*. Department of Computing Science. University of Glasgow.hlm.3

berdasarkan abstrak menggunakan *Full-Text Searching* dan mengintegrasikan ke dalam sistem informasi perpustakaan Jurusan Teknik Elektro, agar dapat mencari karya akhir pada mesin pencarian dengan cepat namun relevan.

### **3.4.2 Desain dan Analisis Sistem ( *System Analysis and Design* )**

Pada tahap desain dan analisis sistem, bertujuan menentukan spesifikasi detail dari komponen-komponen sistem pencari menggunakan *full-text* (manusia, hardware, software, network dan datanya). Melakukan perancangan sistem berupa *Data Flow Diagram* (DFD) serta perancangan *Entity Relationship Diagram* (ERD) yang akan digunakan dalam pengembangan *databasenya*.

#### **3.4.2.1 Data Flow Diagram (DFD)**

DFD dapat digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk mempresentasikan aliran informasi atau fungsi yang lebih detail.

##### **a. Diagram Konteks (Level 0)**

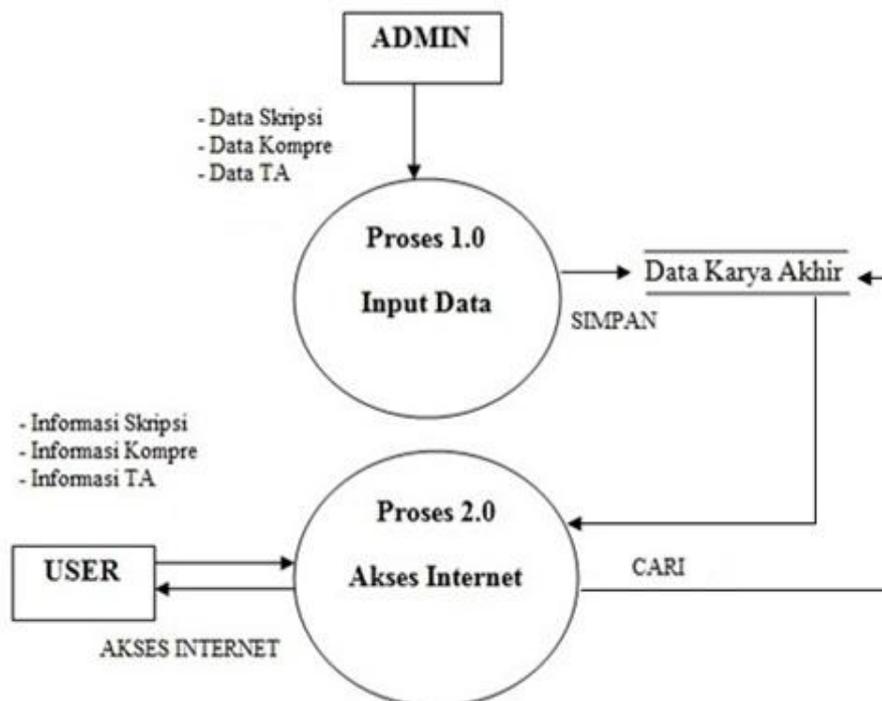
Diagram konteks menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. Diagram konteks dari sistem pencari *full-text* ditunjukkan pada gambar 3.2.



Gambar 3.2 Diagram Konteks (Level 0) Sistem Pencari *Full-Text*

Diagram diatas menunjukkan Sistem Pencari *Full-text*. Pada diagram konteks, terdapat dua entitas yaitu Admin dan User, dan terdapat 1 proses yaitu Sistem pencari *full-text*, terdapat 2 arus data yang saling berhubungan yaitu data informasi dan data karya akhir.

**b. Diagram Zero (Level 1)**



Gambar 3.3 Diagram Zero (Level 1) Sistem Pencari *Full-Text*

Keterangan :

a. Proses 1.0

Pada Proses 1.0 Pengolahan Data yang meliputi Data Skripsi, Data Kompre dan Data TA diolah Admin dan disimpan di dalam filenya masing-masing.

b. Proses 2.0

Proses 2.0 Akses Internet diakses oleh *User*, lalu *User* mendapatkan informasi yang berupa informasi skripsi, informasi kompre dan informasi TA.

**c. Diagram Rinci (Level 2)**

Pada Diagram Rinci (Level 2) proses ditunjukkan pada gambar 3.4, dengan pembahasan sebagai berikut :

a. Proses 1.1

Pada proses 1.1 Input Data Skripsi diolah Admin dan disimpan di dalam file Data Karya Akhir.

b. Proses 1.2

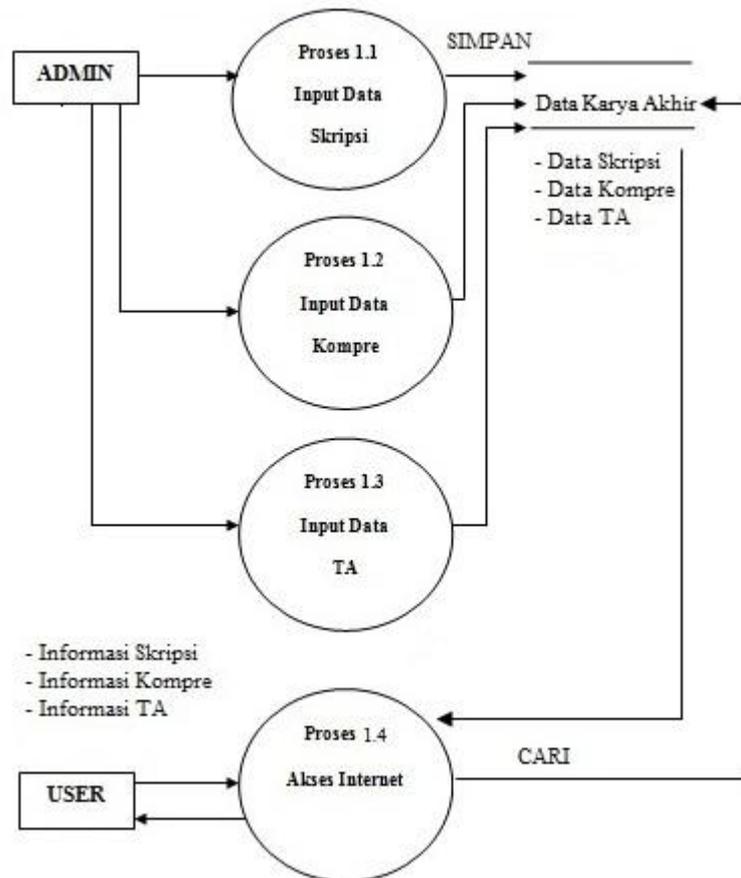
Pada proses 1.2 Input Data Kompre diolah Admin dan disimpan di dalam file Data Karya Akhir.

c. Proses 1.3

Pada proses 1.3 Input Data TA diolah Admin dan disimpan di dalam file Data Karya Akhir.

d. Proses 1.4

Proses 1.4 Akses Internet oleh *User*, lalu *User* mendapatkan informasi yang berupa informasi skripsi, informasi kompre dan informasi TA.

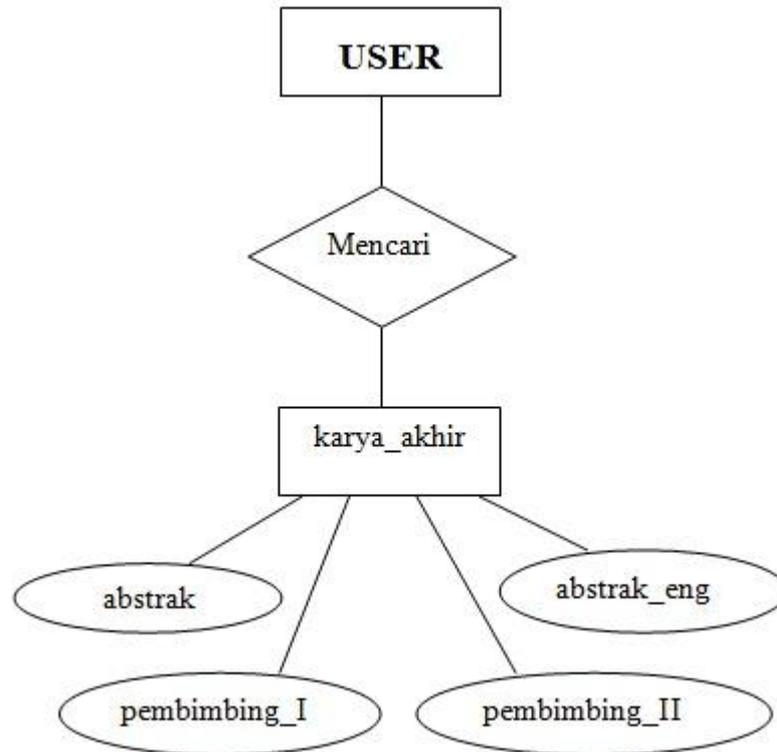


Gambar 3.4 Diagram Rinci (Level 2) Sistem Pencari *Full-Text*

### 3.4.2.2 Entity Relationship Diagram (ERD)

Pada Entity Relationship Diagram (ERD) terdapat entitas *user* dengan *username* sebagai *primary key*-nya dan *karya\_akhir* (Skripsi, Kompre, TA) dengan *id\_karya\_akhir* sebagai *primary key*. Terdapat penambahan atribut pada

entitas `karya_akhir` yaitu `abstrak`, `abstrak_eng`, `pembimbing_I` dan `pembimbing_II`.



Gambar 3.5 *Entity Relationship Diagram (ERD)*

### 3.4.3 Pengembangan Sistem Pencarian

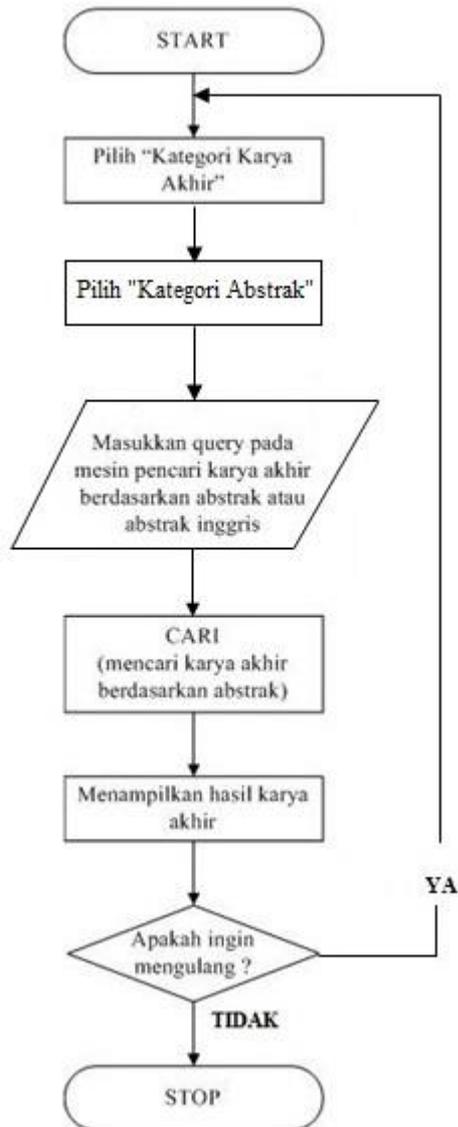
Pengembangan sistem pencarian terdiri dari beberapa tahap yaitu : perbandingan klausa pencarian pada bahasa SQL, penyusunan dan pengujian koleksi tester, analisis hasil.

#### 3.4.3.1 Perbandingan klausa pencarian pada bahasa SQL

Perbandingan klausa pencarian pada bahasa SQL dibuat untuk membandingkan program yang menggunakan LIKE tanpa ORDER, program LIKE dan ORDER dan program *Full-Text Searching* pada mesin pencari karya akhir berdasarkan abstrak guna membandingkan hasil pencarian yang sesuai dengan permintaan / query *user*. Agar dapat melakukan perbandingan, *fulltext searching* mencari satu frasa dengan standar minimum panjang empat karakter. Langkah-langkah perbandingan klausa pencarian pada bahasa SQL adalah sebagai berikut :

1. Abstrak-abstrak yang ada pada mesin pencari karya akhir dimasukan oleh seorang admin *website*.
2. Abstrak yang digunakan adalah abstrak yang ada pada Sistem Informasi Perpustakaan Jurusan Teknik Elektro Universitas Negeri Jakarta.
3. Pemakai (*user*) melakukan pencarian karya akhir melalui mesin pencari karya akhir berdasarkan abstrak.
4. Pemakai memilih kategori karya akhir dan kategori abstrak.
5. Pemakai mengisi query yang diinginkan pada mesin pencarian.

Langkah-langkah perbandingan klausa pencarian pada bahasa SQL juga ditunjukkan melalui *flowchart* pada gambar 3.6.



Gambar 3.6 *Flowchart* perbandingan klausa pencarian pada bahasa SQL

*User* melakukan pencarian karya akhir pada mesin pencari perbandingan klausa, maka *user* harus memilih “kategori karya akhir” dan “kategori abstrak”, *user* mengisi query pada mesin pencari kemudian klik tombol “Cari”, setelah itu *user* baru bisa mendapatkan informasi-informasi karya akhir berdasarkan abstrak. Apabila *user* ingin melakukan pencarian kembali maka klik tombol “Home”. Jika

memasukan query yang sama pada ketiga kriteria pencarian, maka hasilnya akan berbeda-beda sesuai dengan masing-masing program pada kriteria pencarian.

Perbandingan Klausula Pencarian Pada Bahasa SQL	
<a href="#">Home</a>	
<p>Cari Karya Akhir Dengan LIKE Tanpa ORDER</p> <p>Kategori Karya Akhir ▼</p> <p>Kategori Abstrak ▼</p> <p>Abstrak</p> <input type="text"/> <p>Cari</p>	
<p>Cari Karya Akhir Dengan LIKE Dan ORDER</p> <p>Kategori Karya Akhir ▼</p> <p>Kategori Abstrak ▼</p> <p>Abstrak</p> <input type="text"/> <p>Cari</p>	
<p>Cari Karya Akhir Dengan Fulltext Searching</p> <p>Kategori Karya akhir ▼</p> <p>Kategori Abstrak ▼</p> <p>Abstrak</p> <input type="text"/> <p>Cari</p>	
Copyright 2012 Erna Selviyanti	

Gambar 3.7 Rancangan halaman perbandingan klausa pencarian pada bahasa SQL

Membuat *layout* perbandingan klausa pencarian pada bahasa SQL. Halaman yang berisi menu mesin pencari dengan klausa LIKE Tanpa ORDER, LIKE Dan ORDER dan *FULLTEXT SEARCHING*, memiliki listbox kategori

karya akhir, listbox kategori abstrak, mesin pencari abstrak dan tombol klik “Cari”.

Adapun rangkaian pengujian halaman perbandingan klausa pencarian pada bahasa SQL terangkum dalam tabel 3.2.

Tabel 3.2 Kriteria pengujian halaman perbandingan klausa pencarian pada bahasa SQL

No.	Skenario proses	Sistem bekerja	Keterangan
<b>Cari Karya Akhir Dengan LIKE Tanpa ORDER</b>			
1.	<i>User</i> mengklik tombol ‘Cari’. Tidak memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak kosong.		
2.	<i>User</i> mengklik tombol ‘Cari’. Tidak memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak terisi.		
3.	<i>User</i> mengklik tombol ‘Cari’. Tidak memilih kategori karya akhir. Memilih kategori abstrak. Abstrak terisi.		
4.	<i>User</i> mengklik tombol ‘Cari’. Tidak memilih kategori karya akhir. Memilih kategori abstrak. Abstrak kosong.		
5.	<i>User</i> mengklik tombol ‘Cari’. Memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak terisi.		
6.	<i>User</i> mengklik tombol ‘Cari’. Memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak kosong.		
7.	<i>User</i> mengklik tombol ‘Cari’. Memilih kategori karya akhir. Memilih kategori abstrak. Abstrak kosong.		
8.	<i>User</i> mengklik tombol ‘Cari’. Memilih kategori karya akhir. Memilih kategori abstrak. Abstrak terisi.		
<b>Cari Karya Akhir Dengan LIKE Dan ORDER</b>			
9.	<i>User</i> mengklik tombol ‘Cari’. Tidak memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak kosong.		

No.	Skenario proses	Sistem bekerja	Keterangan
10.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak terisi.		
11.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Memilih kategori abstrak. Abstrak terisi.		
12.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Memilih kategori abstrak. Abstrak kosong.		
13.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak terisi.		
14.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak kosong.		
15.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Memilih kategori abstrak. Abstrak kosong.		
16.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Memilih kategori abstrak. Abstrak terisi.		
<b>Cari Karya Akhir Dengan Fulltext Searching</b>			
17.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak kosong.		
18.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak terisi.		
19.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Memilih kategori abstrak. Abstrak terisi.		
20.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak terisi.		
21.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak kosong.		
22.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak kosong.		

No.	Skenario proses	Sistem bekerja	Keterangan
23.	<i>User</i> mengklik tombol ‘Cari’. Memilih kategori karya akhir. Memilih kategori abstrak. Abstrak kosong.		
24.	<i>User</i> mengklik tombol ‘Cari’. Memilih kategori karya akhir. Memilih kategori abstrak. Abstrak terisi.		

### 3.4.3.2 Penyusunan dan Pengujian Koleksi Tester

Penyusunan dan Pengujian Koleksi Tester dibuat untuk mengukur tingkat relevansi pada program *full-text searching*. Koleksi Tester terdiri dari lima tester, dimana satu tester memiliki lima abstrak. Untuk mengukur tingkat relevansi, dibuat tester dengan kriteria:

1. Koleksi Tester pertama memiliki panjang kalimat abstrak yang sama dengan frasa “Perguruan Tinggi Universitas Negeri Jakarta” yang semakin banyak.
2. Koleksi Tester kedua memiliki panjang kalimat abstrak yang berbeda dengan frasa “Perguruan Tinggi Universitas Negeri Jakarta” yang sama banyak.
3. Koleksi Tester ketiga memiliki panjang kalimat abstrak yang sama dengan frasa “Perguruan”, “Tinggi”, “Universitas”, “Negeri” dan “Jakarta”.
4. Koleksi Tester keempat memiliki panjang kalimat abstrak yang sama dengan frasa “Perguruan”, “Tinggi”, “Universitas”, “Negeri”, “Jakarta” dan “Perguruan Tinggi Universitas Negeri Jakarta”.

5. Koleksi Tester kelima memiliki panjang kalimat abstrak yang sama dengan frasa “Perguruan Tinggi Universitas Negeri Jakarta” dimana diletakkan dengan berbeda-beda posisi pada setiap abstrak.

Abstrak-abstrak tester akan di panggil dengan query “Perguruan Tinggi Universitas Negeri Jakarta”. Koleksi Tester menggunakan program *full-text searching*. Penyusunan dan pengujian koleksi tester adalah sebagai berikut :

1. Abstrak-abstrak tester yang ada pada mesin pencari dimasukan oleh seorang admin *website*.
2. Pemakai (*user*) melakukan pencarian abstrak melalui mesin pencari berdasarkan abstrak tester.
3. Pemakai memilih “kategori tester”.
4. Pemakai mengisi query pada mesin pencarian.

Langkah-langkah penyusunan dan pengujian koleksi tester juga ditunjukkan melalui *flowchart* pada gambar 3.8.

*User* melakukan pencarian abstrak tester pada mesin pencari, maka *user* harus memilih “Kategori Tester”, *user* mengisi query “Perguruan Tinggi Universitas Negeri Jakarta” pada mesin pencari kemudian klik tombol “Cari”, setelah itu *user* baru bisa mendapatkan informasi-informasi abstrak tester sesuai dengan kategori tester yang dipilih.

Membuat *layout* penyusunan dan pengujian koleksi tester. Halaman yang berisi menu mesin pencari, memiliki listbox kategori tester, mesin pencari abstrak tester dan tombol klik “Cari”. Adapun rancangan halaman penyusunan dan pengujian koleksi tester ditunjukkan pada gambar 3.9



Gambar 3.8 *Flowchart* penyusunan dan pengujian koleksi tester

<b>KOLEKSI TESTER</b>	
<a href="#">Home</a>	
<div style="display: flex; align-items: center; gap: 10px;"> <div style="border: 1px solid black; padding: 2px 5px; display: flex; align-items: center;"> <span>Kategori Tester</span> <span style="font-size: 0.8em;">▼</span> </div> <input style="width: 60%; height: 20px; border: 1px solid black;" type="text"/> <div style="border: 1px solid black; padding: 2px 10px; font-weight: bold; cursor: pointer;">Cari</div> </div>	
Copyright 2012 Erna Selviyanti	

Gambar 3.9 Rancangan halaman penyusunan dan pengujian koleksi tester

Adapun rangkaian pengujian halaman penyusunan dan pengujian koleksi tester terangkum dalam tabel 3.3.

Tabel 3.3 Kriteria pengujian halaman koleksi tester.

No.	Skenario proses	Sistem bekerja	Kete Rangan
1.	User mengklik tombol 'Cari'. Tidak memilih kategori tester. Abstrak tester kosong.		
2.	User mengklik tombol 'Cari'. Tidak memilih kategori tester. Abstrak tester terisi.		
3.	User mengklik tombol 'Cari'. Memilih kategori tester. Abstrak tester kosong.		
4.	User mengklik tombol 'Cari'. Memilih kategori tester. Abstrak tester terisi.		

### 3.4.3.3 Analisis Hasil

Analisis hasil berupa pengujian perbandingan klausa pencarian pada bahasa SQL. Menganalisis perbandingan klausa pencarian pada bahasa SQL bertujuan untuk mengetahui perbedaan hasil pencarian dengan query yang sama. Query di batasi satu frasa. Agar dapat melakukan perbandingan, *fulltext searching* mencari satu frasa dengan standar minimum panjang empat karakter. Kategori karya akhir yang digunakan adalah 'skripsi'. Analisis hasil pengujian perbandingan klausa pencarian pada bahasa SQL dirinci dalam bentuk tabel.

Tabel 3.4 Pengujian perbandingan klausa pencarian pada bahasa SQL dengan satu frasa dan lima karakter, bahasa Indonesia.

No.	Kriteria Pencarian	Query	Hasil Pencarian Berdasarkan id_karya_akhir					Ket
			Perta ma	Kedua	Ketiga	Keem pat	Keli ma	
1.	LIKE tanpa ORDER	Media						

2.	LIKE dan ORDER	Media						
3.	Full-Text Searching	Media						

Tabel 3.5 Pengujian perbandingan klausa pencarian pada bahasa SQL dengan satu frasa dan delapan karakter, bahasa Inggris.

No.	Kriteria Pencarian	Query	Hasil Pencarian Berdasarkan id_karya_akhir					Ket
			Pertama	Kedua	Ketiga	Keempat	Kelima	
1.	LIKE tanpa ORDER	Learning						
2.	LIKE dan ORDER	Learning						
3.	Full-Text Searching	Learning						

Analisis hasil berupa pengujian koleksi tester. Menganalisis koleksi tester bertujuan untuk mengukur tingkat relevansi dengan menggunakan fitur *full-text searching*. Koleksi tester yang digunakan cenderung homogen dan heterogen. Analisis hasil pengujian koleksi tester dirinci dalam bentuk tabel.

Tabel 3.6 Pengujian koleksi tester.

No.	Tester	Query	Hasil Pencarian Berdasarkan id_tester dan nilai relevansi					Ket
			Pertama	Kedua	Ketiga	Keempat	Kelima	
1.	User memilih tester_1	Perguruan Tinggi Universitas Negeri Jakarta						
2.	User memilih tester_2	Perguruan Tinggi Universitas Negeri Jakarta						

3.	<i>User</i> memilih tester_3	Perguruan Tinggi Universitas Negeri Jakarta						
4.	<i>User</i> memilih tester_4	Perguruan Tinggi Universitas Negeri Jakarta						
5.	<i>User</i> memilih tester_5	Perguruan Tinggi Universitas Negeri Jakarta						

#### 3.4.4 Integrasi Sistem Pencarian *Full-Text* pada SIPEL

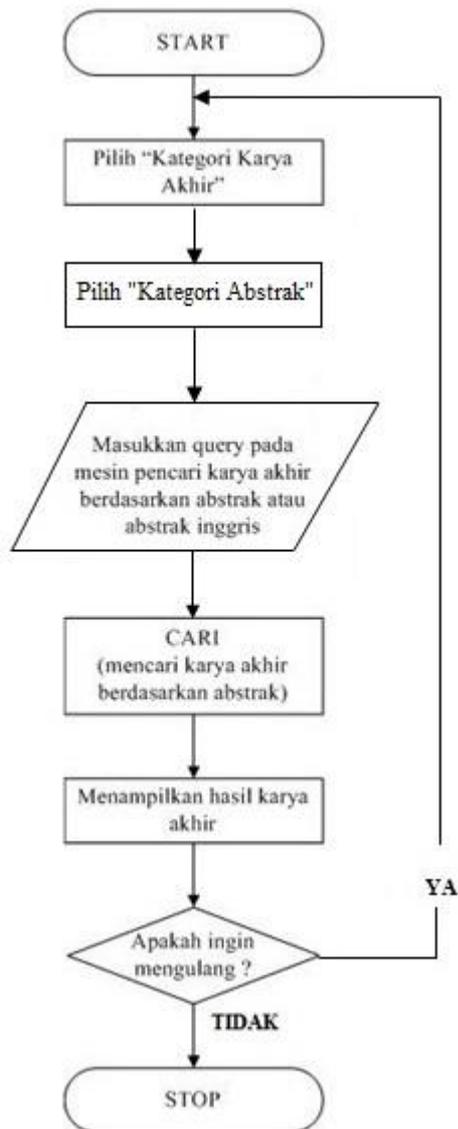
Integrasi sistem pencarian *full-text* pada SIPEL dengan menambahkan tabel *database* baru dan pencarian karya akhir berdasarkan abstrak menggunakan fitur *full-text searching*. Langkah-langkah integrasi sistem pencarian *full-text* pada SIPEL adalah sebagai berikut :

1. Abstrak-abstrak yang ada pada mesin pencari karya akhir Sistem Informasi Perpustakaan Jurusan Teknik Elektro Universitas Negeri Jakarta dimasukan oleh seorang admin *website*.
2. Pemakai (*user*) melakukan pencarian karya akhir melalui mesin pencari karya akhir berdasarkan abstrak pada Sistem Informasi Perpustakaan Jurusan Teknik Elektro Universitas Negeri Jakarta.
3. Pemakai mengisi query yang diinginkan pada mesin pencarian.
4. Pemakai dapat mengisi tahun karya akhir, pembimbing karya akhir dan judul karya akhir yang diinginkan dalam pencarian karya akhir melalui

mesin pencari karya akhir dengan menggunakan perintah SQL klausa LIKE .

5. Pemakai dapat mengisi query abstrak atau abstrak inggris yang diinginkan dalam pencarian karya akhir melalui mesin pencari karya akhir dengan menggunakan perintah SQL klausa *full-text searching*.

Langkah-langkah integrasi sistem pencarian *full-text* pada SIPEL ditunjukkan melalui flowchart pada gambar 3.10. dan langkah-langkah sistem pencarian menggunakan perintah SQL klausa LIKE ditunjukkan melalui flowchart pada gambar 3.11



Gambar 3.10 *Flowchart* integrasi sistem pencarian *full-text* pada SIPEL

Keterangan :

*User* melakukan pencarian karya akhir pada mesin pencari karya akhir, *user* harus memilih “kategori karya akhir” dan “kategori abstrak”, *user* harus mengisi query berdasarkan abstrak atau abstrak inggris dan klik tombol “Cari”, setelah itu *user* baru bisa mendapatkan informasi-informasi karya akhir berdasarkan abstrak atau abstrak inggris.



Gambar 3.11 *Flowchart* sistem pencarian menggunakan perintah SQL klausa LIKE

Keterangan :

*User* melakukan pencarian karya akhir pada mesin pencari karya akhir, *user* harus memilih “kategori karya akhir”, *user* harus mengisi query berdasarkan judul/ pembimbing 1/ pembimbing 2/ tahun dan klik tombol “Cari”, setelah itu *user* baru bisa mendapatkan informasi-informasi karya akhir berdasarkan judul atau pembimbing atau tahun.

Perencanaan *database* pada integrasi sistem pencarian *full-text* pada SIPEL dimana *database* sebagai penyedia data dan untuk memproses *query* dari parameter yang dikirim / diminta oleh *client*. Pada skripsi digunakan *database server* MySQL sebagai penyimpan data yang akan ditampilkan melalui *website*. Dalam perencanaan dan penambahan *database*, dibuat beberapa tabel, antara lain:

### 1. Tabel karya\_akhir

Tabel 3.7 Berisi data-data karya akhir yang ada di perpustakaan

Nama field	Type field	Panjang field
id_karya_akhir	Char	50
id_kategorikaryaakhir	Char	50
nama_karya_akhir	Char	200
penulis	Char	50
pembimbing_I	Char	50
pembimbing_II	Char	50
tahun	Char	50
keterangan	Text	-
abstrak	Text	-
abstrak_eng	Text	-
halaman	Char	50
no_rak	Char	50
jumlah	Int	-
stok	Int	-
tgl_masuk	Date	-
gambar	Char	50
dipinjam	Int	-

- a. id\_karya\_akhir (*character*), *field* ini berisi nomor karya akhir dan digunakan sebagai kunci utama (*Primary Key*), karena isi dari *field* tidak mungkin sama untuk setiap *recordnya*.
- b. id\_kategorikaryaakhir (*character*), *field* yang berisi nomor kategori karya akhir.
- c. nama\_karya\_akhir (*character*), *field* yang berisi nama karya akhir.

- d. penulis (*character*), *field* yang berisi nama penulis karya akhir.
- e. pembimbing\_I (*character*), *field* yang berisi nama pembimbing 1.
- f. pembimbing\_II (*character*), *field* yang berisi nama pembimbing 2.
- g. tahun (*character*), *field* yang berisi tahun karya akhir dibuat.
- h. keterangan (*text*), *field* yang berisi keterangan karya akhir.
- i. abstrak (*text*), *field* yang berisi abstrak karya akhir.
- j. abstrak\_eng (*text*), *field* yang berisi abstrak karya akhir dalam bahasa inggris.
- k. halaman (*character*), *field* yang berisi jumlah halaman karya akhir.
- l. no\_rak (*character*), *field* yang berisi tempat rak karya akhir disimpan.
- m. jumlah (*integer*), *field* yang berisi jumlah keseluruhan karya akhir yang ada di perpustakaan.
- n. stok (*integer*), *field* yang berisi jumlah *copy* karya akhir yang belum dipinjam.
- o. tgl\_masuk (*date*), *field* yang berisi tanggal masuk karya akhir.
- p. gambar (*character*), *field* yang berisi foto sampul karya akhir.
- q. dipinjam (*integer*), *field* yang berisi jumlah berapa kali karya akhir dipinjam.

## 2. Tabel kategorikaryaakhir

Tabel 3.8 Berisi kategori karya akhir yang ada di perpustakaan

Nama field	Type field	Panjang field
id_kategorikaryaakhir	Char	50
nama_grupkaryaakhir	Char	50
nama_kategorikaryaakhir	Char	50

- a. `id_kategorikaryaakhir` (*character*), *field* yang berisi nomor kategori karya akhir (Skripsi, TA, Kompre, Laporan PKL) dan dijadikan sebagai *primary key*.
- b. `nama_grupkaryaakhir` (*character*), *field* yang berisi nama-nama grup karya akhir berdasarkan pengkodean Perpustakaan Jurusan Teknik Elektro.
- c. `nama_kategorikaryaakhir` (*character*), *field* yang berisi nama-nama kategori karya akhir (Skripsi, TA, Kompre, Laporan PKL).

### 3. Tabel tester

Tabel 3.9 Berisi data tester

Nama field	Type field	Panjang field
<code>id_tester</code>	Int	-
<code>id_kategoritester</code>	Int	-
<code>abstrak_tester</code>	Text	-

- a. `id_tester` (*integer*), *field* yang berisi nomor tester, dan dijadikan sebagai *primary key*.
- b. `id_kategoritester` (*integer*), *field* yang berisi nomor kategori tester (`tester_1`, `tester_2`, `tester_3`, `tester_4` dan `tester_5`).
- c. `abstrak_tester` (*text*), *field* yang berisi abstrak tester.

### 4. Tabel kategoritester

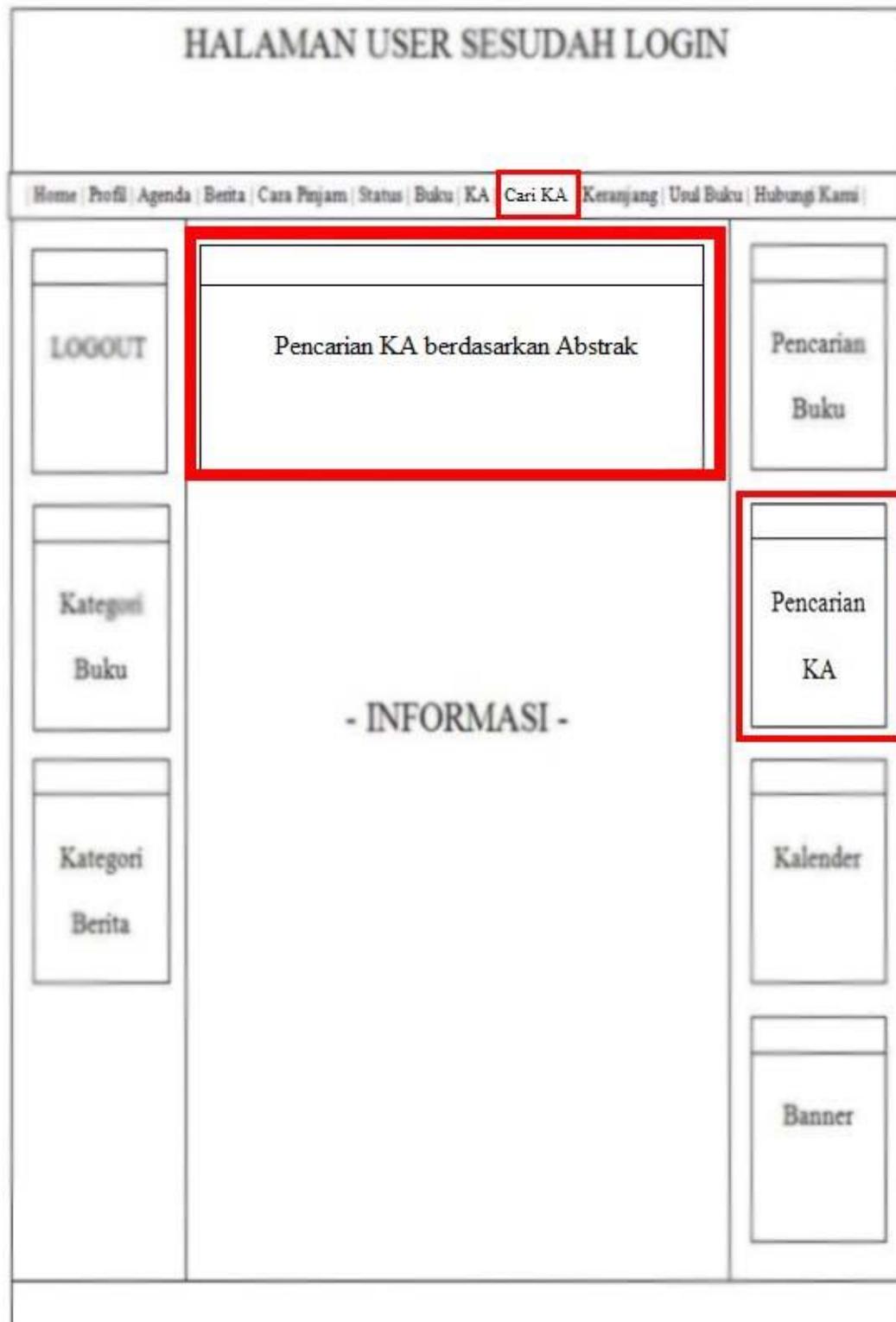
Tabel 3.10 Berisi kategori tester

Nama field	Type field	Panjang field
<code>id_kategoritester</code>	Int	-
<code>nama_kategoritester</code>	Char	50

- a. *id\_kategoritester* (*integer*), *field* yang berisi nomor kategori tester, dan dijadikan sebagai *primary key*.
- b. *nama\_kategoritester* (*Character*), *field* yang berisi nama kategori tester (*tester\_1*, *tester\_2*, *tester\_3*, *tester\_4* dan *tester\_5*).

Membuat integrasi sistem pencarian *full-text* pada SIPEL. Bagian kanan halaman terdapat menu mesin pencari, memiliki listbox kategori karya akhir, mesin pencari judul, mesin pencari pembimbing 1, mesin pencari pembimbing 2, mesin pencari tahun dan tombol klik “Cari”. Terdapat menu tambahan “Cari KA” yang berguna untuk mencari karya akhir berdasarkan abstrak. Bagian tengah atas halaman terdapat menu mesin pencari, memiliki listbox kategori karya akhir, memiliki listbox kategori abstrak, mesin pencari abstrak dan tombol klik “Cari”. Adapun rancangan halaman integrasi sistem pencarian *full-text* pada SIPEL ditunjukkan pada gambar 3.12.

Jika *layout* telah siap, selanjutnya dilakukan penambahan kode *script* untuk menampilkan data yang ada. *Script* PHP yang digunakan ditambahkan pada *tag* dan kode html-nya. Sehingga dari tampilan tersebut dapat terkoneksi ke *databasenya*



Gambar 3.12 Rancangan Halaman integrasi sistem pencarian *full-text* pada SIPEL

### 3.4.5 Pengujian SIPEL

Pengujian SIPEL terdiri dari dua pengujian sistem pencarian, dimana pengujian dirinci dalam bentuk tabel.

Tabel 3.11 Sistem pencarian (LIKE)

No.	Skenario proses	Sistem bekerja	Keterangan
1.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul kosong. Pembimbing I kosong. Pembimbing II kosong. Tahun kosong.		
2.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul kosong. Pembimbing I kosong. Pembimbing II kosong. Tahun terisi.		
3.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul kosong. Pembimbing I kosong. Pembimbing II terisi. Tahun terisi.		
4.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul kosong. Pembimbing I terisi. Pembimbing II terisi. Tahun terisi.		
5.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul terisi. Pembimbing I terisi. Pembimbing II terisi. Tahun terisi.		
6.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul terisi. Pembimbing I kosong. Pembimbing II kosong. Tahun kosong.		
7.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul terisi. Pembimbing I terisi. Pembimbing II kosong. Tahun kosong.		
8.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul terisi. Pembimbing I terisi. Pembimbing II terisi. Tahun kosong.		
9.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul kosong. Pembimbing I terisi. Pembimbing II kosong. Tahun kosong.		

No.	Skenario proses	Sistem bekerja	Keterangan
10.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul kosong. Pembimbing I terisi. Pembimbing II terisi. Tahun kosong.		
11.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul kosong. Pembimbing I kosong. Pembimbing II terisi. Tahun kosong.		
12.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul terisi. Pembimbing I kosong. Pembimbing II terisi. Tahun kosong.		
13.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Judul terisi. Pembimbing I terisi. Pembimbing II kosong. Tahun terisi.		
14.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul kosong. Pembimbing I kosong. Pembimbing II terisi. Tahun kosong.		
15.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul kosong. Pembimbing I terisi. Pembimbing II kosong. Tahun terisi.		
16.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul terisi. Pembimbing I terisi. Pembimbing II kosong. Tahun terisi.		
17.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul terisi. Pembimbing I kosong. Pembimbing II kosong. Tahun terisi.		
18.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul terisi. Pembimbing I kosong. Pembimbing II terisi. Tahun terisi.		
19.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul kosong. Pembimbing I kosong. Pembimbing II kosong. Tahun terisi.		
20.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul kosong. Pembimbing I kosong. Pembimbing II terisi. Tahun terisi.		

No.	Skenario proses	Sistem bekerja	Keterangan
21.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul kosong. Pembimbing I terisi. Pembimbing II terisi. Tahun terisi.		
22.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul kosong. Pembimbing I kosong. Pembimbing II kosong. Tahun kosong.		
23.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul terisi. Pembimbing I kosong. Pembimbing II kosong. Tahun kosong.		
24.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul terisi. Pembimbing I terisi. Pembimbing II kosong. Tahun kosong.		
25.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul terisi. Pembimbing I terisi. Pembimbing II terisi. Tahun kosong.		
26.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Judul terisi. Pembimbing I terisi. Pembimbing II terisi. Tahun terisi.		

Tabel 3.12 Sistem pencarian (*full-text*)

No.	Skenario proses	Sistem bekerja	Keterangan
1.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak kosong.		
2.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak terisi.		
3.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Memilih kategori abstrak. Abstrak terisi.		
4.	<i>User</i> mengklik tombol 'Cari'. Tidak memilih kategori karya akhir. Memilih kategori abstrak. Abstrak kosong.		

No.	Skenario proses	Sistem bekerja	Keterangan
5.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak terisi.		
6.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Tidak memilih kategori abstrak. Abstrak kosong.		
7.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Memilih kategori abstrak. Abstrak kosong.		
8.	<i>User</i> mengklik tombol 'Cari'. Memilih kategori karya akhir. Memilih kategori abstrak. Abstrak terisi.		