

BAB III

METODELOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Penelitian ini dilaksanakan di Prodi Teknik Informatika IIB Darmajaya yang berada di Kec. Rajabasa, Kota Bandar Lampung, Provinsi Lampung.

3.1.2 Waktu Penelitian

Penelitian ini dilakukan selama 3 bulan, adapun detail kegiatan tertera pada table dibawah ini :

3.2 Populasi dan Sampel Penelitian.

3.2.1 Populasi Penelitian

Populasi penelitian yang diambil dari data mahasiswa dan dosen di Prodi Teknik Informatika.

3.2.2 Sampel Penelitian

Sampel pada penelitian ini adalah berisi data dosen dan data mahasiswa yang sedang mengerjakan skripsi di Prodi Teknik Informatika IIB Darmajaya.

3.3 Metode Pengumpulan Data

Dalam melakukan perancangan perangkat sistem rekomendasi dosen pembimbing skripsi berdasarkan kompetensi sebagai berikut:

a. Studi Pustaka

Peneliti menggunakan Studi pustaka dalam proses pengumpulan data dan informasi. Studi pustaka yang dilakukan peneliti bersumber dari berbagai sumber seperti Buku, Literatur, Jurnal Ilmiah terdahulu yang berkaitan dengan penelitian dan semua sumber terpercaya lainnya yang dapat menunjang dalam dalam penelitian ini.

b. Observasi

Metode ini digunakan dengan cara terjun langsung dan mengamati apa saja yang di perlukan untuk menemukan informasi dan pengetahuan yang diperlukan untuk bahan penelitian.

c. Wawancara

Metode wawancara dilakukan dengan cara menyampaikan sejumlah pertanyaan dari pewawancara untuk di jawab oleh narasumber dari pihak jurusan untuk mendapatkan suatu informasi.

3.4 Metode Pengembangan Perangkat Lunak

Dalam pembuatan perangkat sistem rekomendasi dosen pembimbing skripsi berdasarkan kompetensi menggunakan metode scrum:

3.4.1 *Product Backlog*

Pada tahap ini akan dilakukan pengumpulan data.

1. Jadwal Penelitian

Penelitian ini dilakukan selama 3 bulan

2. Hasil Survey dan Wawancara

Pada tahapan survey ini dilakukan ke lokasi penelitian yaitu Prodi Teknik Informatika IIB Darmajaya. Tujuan dilakukan survey ini adalah untuk mengetahui sistem rekomendasi dosen pembimbing skripsi. Sedangkan tahapan wawancara dilakukan untuk mengajukan pertanyaan mengenai sistem rekomendasi dosen pembimbing skripsi.

a. Peminatan apa saja yang ada di Prodi Teknik Informatika IIB Darmajaya?

b. Bagaimana cara mahasiswa untuk mendapatkan dosen pembimbing skripsi?

3. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang akan di terapkan pada sistem dan menjelaskan kebutuhan yang dibutuhkan sistem agar berjalan dengan baik. Adapun proses yang akan di hasilkan oleh sistem adalah:

- a. Admin dapat membuka sistem tanpa masalah.
- b. Dosen dapat mengetahui mahasiswa yang akan mengajukan skripsi.
- c. Mahasiswa mendapatkan dosen pembimbing skripsi sesuai dengan peminatannya.

4. Kebutuhan Non Fungsional

a. Perangkat Keras

- Laptop
- Ram 8 GB
- Prosesor intel I3-6006U
- HDD 1 TB

b. Perangkat Lunak

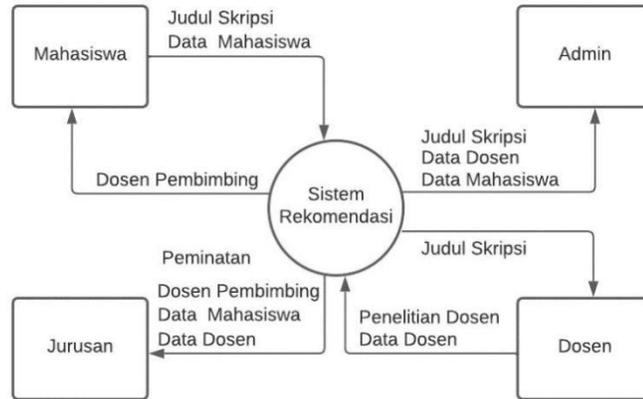
- Sistem operasi windows 11
- Visual studio code
- Android studio
- Figma

3.4.2 *Sprint Backlog*

Sprint backlog yang berisi daftar kegiatan yang telah dilakukan, pekerjaan yang sedang dilakukan dan pekerjaan yang akan dilakukan.

3.4.2.1 *Data Flow Diagram (DFD)*

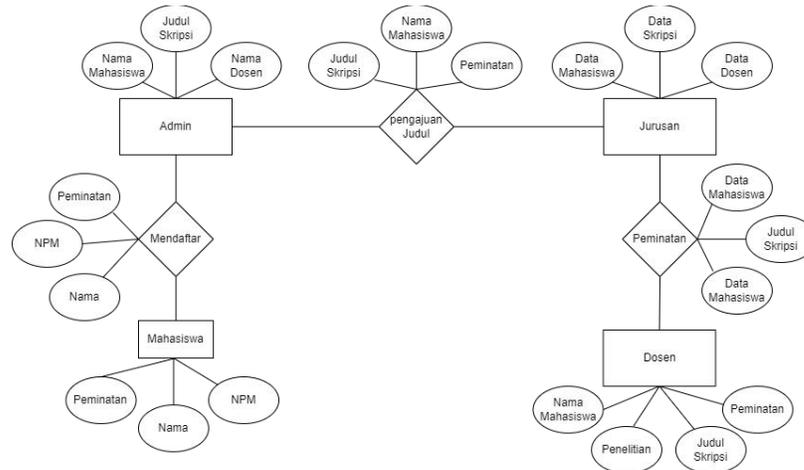
Berikut merupakan data flow diagram dari sistem rekomendasi dosen pembimbing skripsi berdasarkan kompetensi.



Gambar 3.1 Data Flow Diagram (DFD)

3.4.2.2 Entity Relationship Diagram (ERD)

Berikut merupakan entity relationship diagram dari sistem rekomendasi dosen pembimbing skripsi berdasarkan kompetensi.



Gambar 3.2 Entity Relationship Diagram (ERD)

3.4.2.3 Struktur Database

Database yang digunakan untuk membuat perangkat lunak sistem rekomendasi dosen pembimbing skripsi berdasarkan kompetensi adalah *MySql*. Maka rancangan database untuk perangkat lunak ini sebagai berikut:

1. Tabel Mahasiswa

Nama Tabel : Mahasiswa

Kunci Utama (*Primary Key*) : idmahasiswa

Kunci Tamu (*Foreign Key*) : -

Table 3.2 Database data mahasiswa

<i>Field Name</i>	<i>Type</i>	<i>Size</i>	<i>Description</i>
Idmahasiswa	int	10	Id
Nama mahasiswa	varchar	50	Nama mahasiswa
Judul Skripsi	varchar	50	Judul Skripsi

2. Tabel Dosen

Nama Tabel : Dosen

Kunci Utama (*Primary Key*) : Iddosen

Kunci Tamu (*Foreign Key*) : -

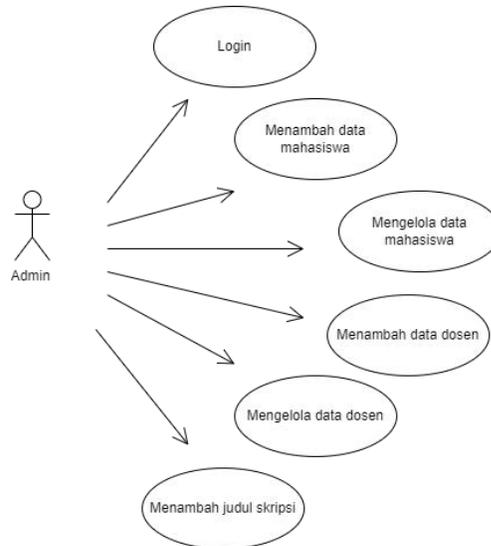
Table 3.3 Database data dosen

<i>Field Name</i>	<i>Type</i>	<i>Size</i>	<i>Description</i>
Iddosen	int	10	Id
Nama Dosen	varchar	100	Nama Dosen
Kompetensi Dosen	varchar	100	Kompetensi Dosen

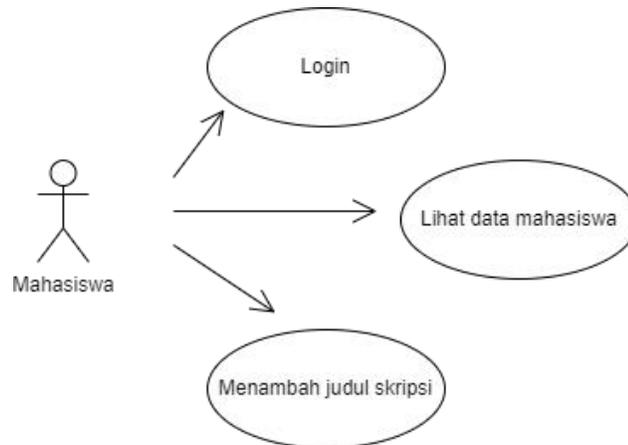
3.4.2.4 Use Case Diagram

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan

“bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. Berikut adalah use case diagram yang dibangun pada tugas akhir ini.



Gambar 3.3 Use Case Admin Jurusan

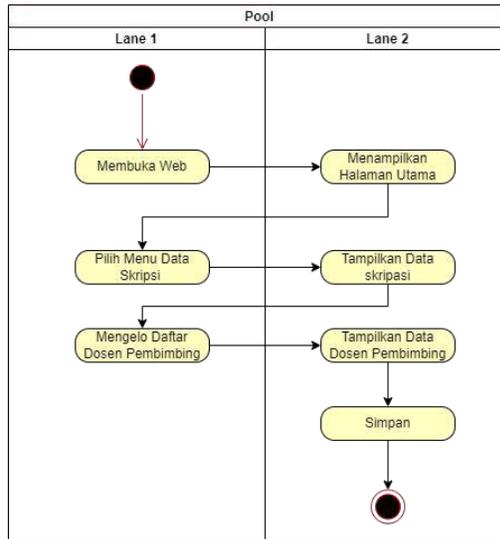


Gambar 3.4 Use Case Mahasiswa

3.4.2.5 Activity Diagram

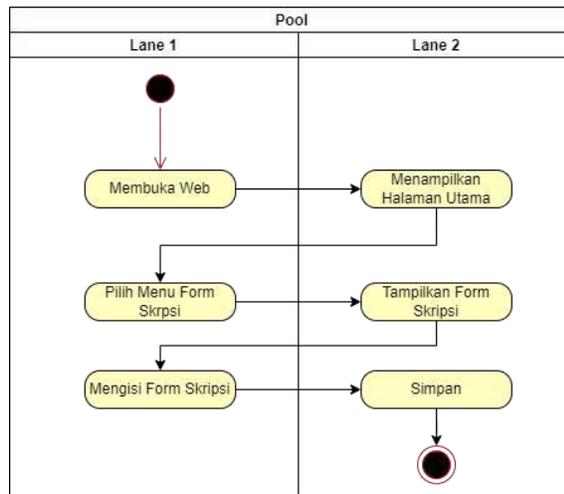
Activity Diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, yang mungkin terjadi, dan decision bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut adalah activity diagram yang terdapat pada tugas akhir yang dibangun:

1. Activity Admin Jurusan



Gambar 3.5 Activity Diagram Admin Jurusan

2. Activity Mahasiswa



Gambar 3.6 Activity Diagram Mahasiswa

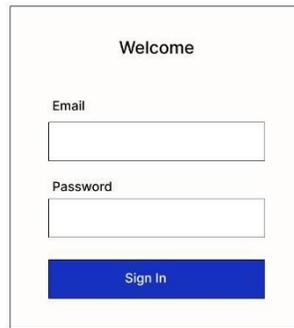
3.4.2.6 Tampilan Antarmuka

Rancangan *interface* perangkat lunak sistem rekomendasi dosen pembimbing skripsi berdasarkan kompetensi adalah sebagai berikut :

a. Desain Tampilan Admin

1. Halaman *Login*

Desain tampilan halaman *login* admin adalah seperti gambar 3.7.



Welcome

Email

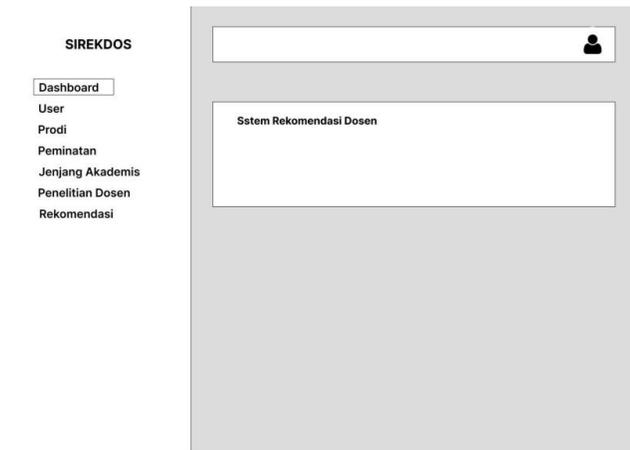
Password

Sign In

Gambar 3.7. Desain Tampilan Awal *Login* (Admin)

2. Halaman *Dashboard*

Desain tampilan halaman *dashboard* admin adalah seperti gambar 3.8.



SIREKDOS

Dashboard

User

Prodi

Peminatan

Jenjang Akademis

Penelitian Dosen

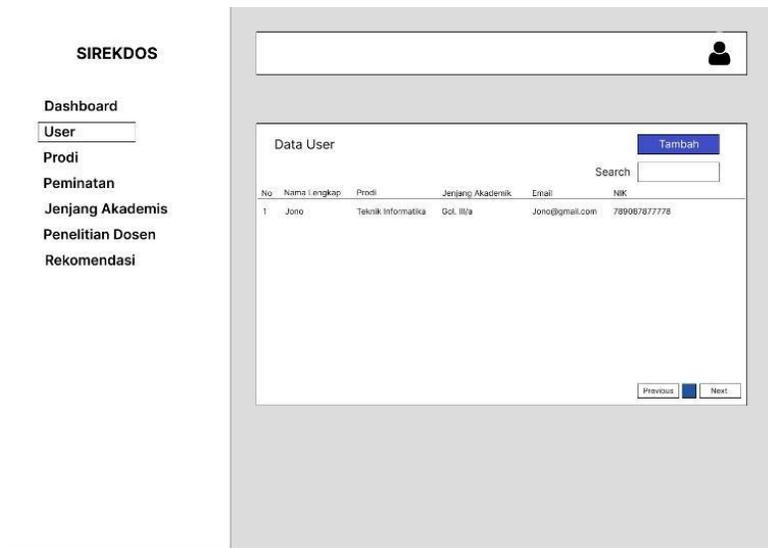
Rekomendasi

Sistem Rekomendasi Dosen

Gambar 3.8. Desain Tampilan Awal *Dashbord* (Admin)

3. Halaman *User*

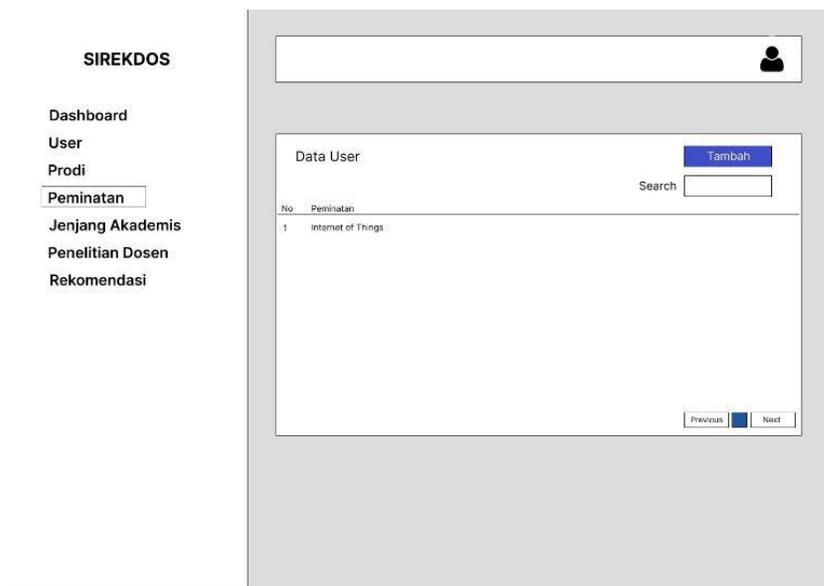
Desain tampilan halaman *user* adalah seperti gambar 3.9.



Gambar 3.9. Desain Tampilan *User* (Admin)

4. Halaman Peminatan

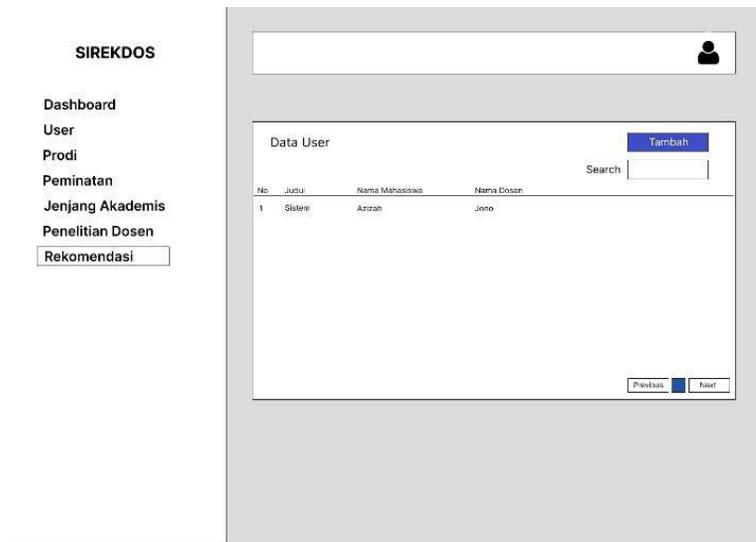
Desain tampilan halaman peminatan admin adalah seperti gambar 3.10.



Gambar 3.10. Desain Tampilan Peminatan (Admin)

5. Halaman Rekomendasi

Desain tampilan halaman rekomendasi admin adalah seperti gambar 3.11.

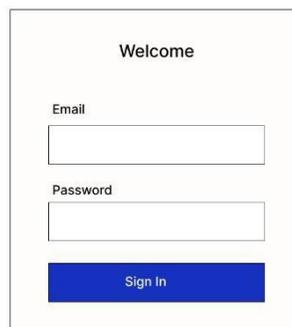


Gambar 3.11. Desain Tampilan Rekomendasi

b. Desain Tampilan Dosen

1. Halaman Login

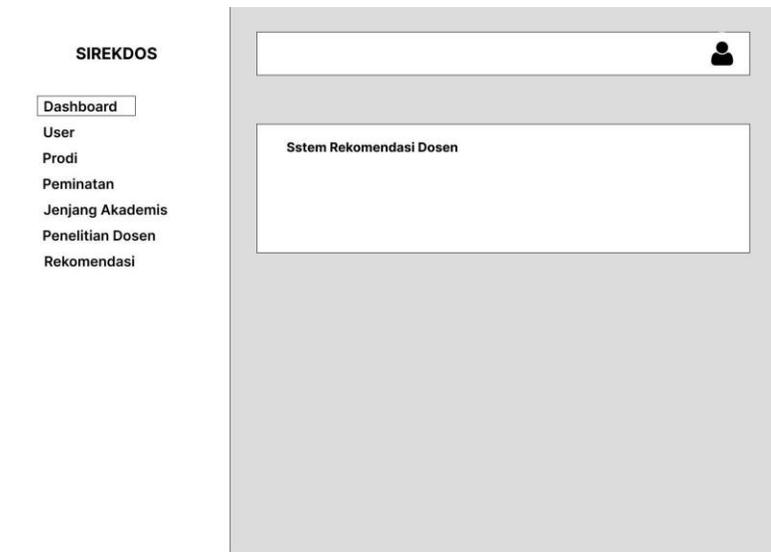
Desain tampilan halaman *login* dosen adalah seperti gambar 3.12.



Gambar 3.12. Desain Tampilan *Login* Dosen

2. Halaman *Dashboard*

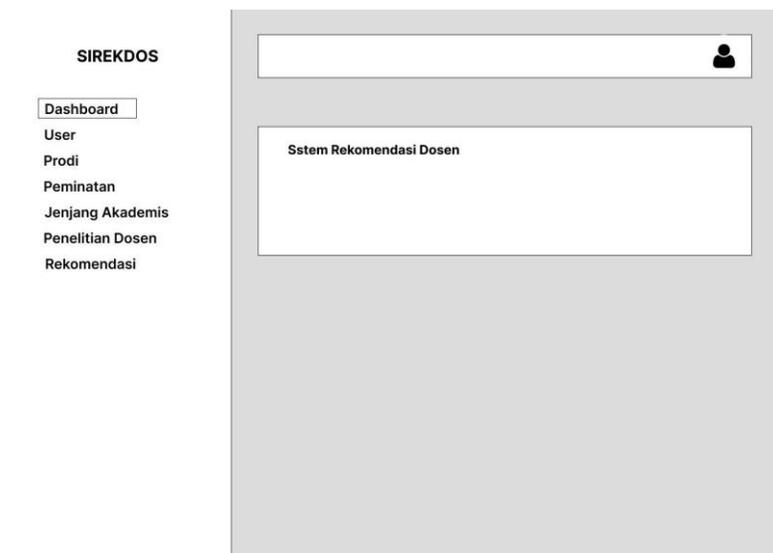
Desain tampilan halaman *dashboard* dosen adalah seperti gambar 3.13.



Gambar 3.13. Desain Tampilan *Dashboard* Dosen

3. Tampilan Mahasiswa

Desain tampilan halaman mahasiswa adalah seperti gambar 3.14.



Gambar 3.14. Desain Tampilan Mahasiswa

c. Halaman Mahasiswa

1. Halaman *Login*

Desain tampilan halaman *login* mahasiswa adalah seperti gambar 3.14.



Gambar 3.14. Desain Tampilan *Login* Dosen

2. Halaman Skripsi

Desain tampilan halaman skripsi adalah seperti gambar 3.14.

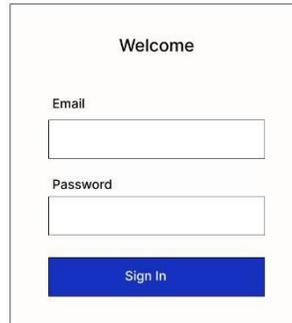


Gambar 3.14. Desain Tampilan Skripsi

d. Halaman Jurusan

1. Halaman *Login*

Desain tampilan halaman *login* jurusan adalah seperti gambar 3.15.



Welcome

Email

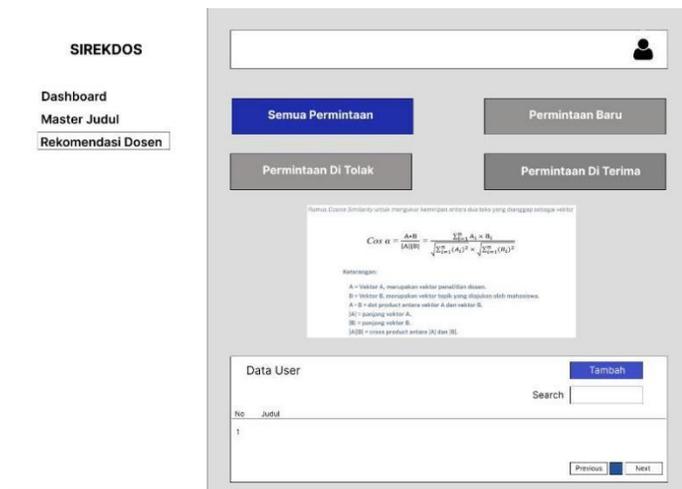
Password

Sign In

Gambar 3.15. Desain Tampilan Awal *Login*

2. Halaman Rekomendasi

Desain tampilan halaman rekomendasi adalah seperti gambar 3.16.



SIREKDOS

Dashboard
Master Judul
Rekomendasi Dosen

Semua Permintaan Permintaan Baru
Permintaan Di Tolak Permintaan Di Terima

Rumus Cosine Similarity untuk mengukur kemiripan antara dua buku yang dianggap sebagai vektor

$$\text{Cos } \theta = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}}$$

Keterangan:
A = vektor A, merupakan vektor penandaan buku.
B = vektor B, merupakan vektor topik yang diabaikan oleh mahasiswa.
A · B = dot product antara vektor A dan vektor B.
|A| = panjang vektor A.
|B| = panjang vektor B.
|A||B| = cross product antara |A| dan |B|.

Data User Tambah

Search

No	Judul
1	

Previous

Gambar 3.15. Desain Tampilan Rekomendasi

3.4.3 Implementasi

Penerapan untuk melihat kompleksitas algoritma sequential search dengan menggunakan notasi Big O, Algoritma ini biasa dinyatakan dengan notasi sequential atau linear $O(n)$. Penjelasan sebagai berikut dalam suatu algoritma itu biasanya terdapat satu set data, dan banyaknya data tersebut adalah n . Jadi kalau algoritma $O(1)$ artinya seberapa banyak data yang tersedia maka waktu yang diperlukan algoritma tidak bergantung pada banyaknya data. Namun untuk sequential atau linear $O(n)$ artinya waktu yang diperlukan itu sebanding (berbanding lurus) dengan banyaknya data. Aturan untuk menentukan kompleksitas waktu asimptotik pada algoritma pencarian beruntun atau sequential search, Jika kompleksitas waktu $T(n)$ dari algoritma diketahui adalah sebagai berikut:

1. $T_{\min}(n) = 1 = O(1)$.
2. $T_{\max}(n) = n = O(n)$.
3. $T_{\text{avg}}(n) = (n + 1)/2 = O(n)$

```
String key = [data yang dicari];
String data() = [larik data];
Boolean flag = False;
for (int index = 0; index < data.length; index++) {
    If (data[index] == key) {
        flag = True; break;
    }
}

If (flag) {
    System.out.println("Data ditemukan");
}
Else {
    System.out.println("Data tidak ditemukan");
}
```

Gambar 3.16 Source Code Algoritma Sequential Search

Berdasarkan source code program metode pencarian sekuensial di atas terlihat bahwa inti algoritma pencarian sekuensial adalah bagian perulangan (loop) yaitu For dengan kondisi `index < data.length` yang mengontrol agar perulangan jangan sampai melewati batas `data.length`.

Terdapat fungsi If (`data[index] == key`) yang mengontrol pencarian apabila data sudah ditemukan maka pencarian tidak perlu lagi dilanjutkan. Hal yang mengakibatkan proses pencarian keluar dari bagian perulangan adalah baris `flag = True;` dan `break;`. Setelah itu jika nilai `flag = True` maka data ditemukan, selain itu maka data tidak ditemukan.

