

BAB IV

PERANCANGAN

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan yang dilakukan meliputi dua tahap. Proses analisis kebutuhan dilakukan pada tahap pertama dan tahap kedua adalah proses perancangan perangkat lunak. Tahap analisis kebutuhan terdiri dari dua langkah yaitu membuat daftar kebutuhan *user* dan menggunakan pemodelan *use case diagram* untuk menggambarkan kebutuhan tersebut. Proses perancangan perangkat lunak mempunyai dua tahap, yaitu perancangan umum dan perancangan detail sistem pendukung keputusan pemilihan konsentrasi di Program Studi Teknik Informatika Universitas Brawijaya. Perancangan umum menggambarkan relasi antar paket (*package*) dan kelas (*class*) sebagai pemodelan sistem secara keseluruhan. Perancangan detail menggunakan *class diagram* dan *sequence diagram* sebagai pemodelan perangkat lunak.

4.1 Analisis Kebutuhan (*Requirements Analysis*)

Pada analisis kebutuhan ini diawali dengan identifikasi aktor yang terlibat dengan sistem, penjabaran kebutuhan dan kemudian memodelkannya ke dalam suatu *use case diagram*. Analisis kebutuhan ini ditujukan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

4.1.1 Identifikasi Aktor

Tahap ini mempunyai tujuan untuk melakukan identifikasi terhadap aktor yang akan berinteraksi dengan sistem. Tabel 4.1 memperlihatkan sebuah aktor beserta penjelasannya yang merupakan hasil dari proses identifikasi aktor.

Tabel 4.1 Identifikasi Aktor

Actor	Deskripsi Aktor
Admin	Admin merupakan aktor pengguna yang bertugas untuk memantau dan memantain sistem, seperti mengolah data

	mahasiswa, mengolah data pelatihan, dan mengolah data hasil tes mahasiswa.
Operator	Operator merupakan aktor pengguna yang bertugas untuk membantu admin dalam mengolah data mahasiswa dan mengolah data hasil tes mahasiswa.
Mahasiswa	Mahasiswa merupakan aktor pengguna yang menggunakan sistem untuk mendukung keputusan yang akan diambil dalam pemilihan konsentrasi yang akan ditempuh.

Sumber: Perancangan

4.1.2 Daftar Kebutuhan

Daftar kebutuhan ini terdiri dari sebuah kolom yang menguraikan kebutuhan yang harus disediakan oleh sistem, dan pada kolom yang lain akan menunjukkan nama *use case* yang akan menunjukkan fungsionalitas masing-masing kebutuhan tersebut. Daftar kebutuhan fungsional dan non fungsional keseluruhan sistem ditunjukkan pada Tabel 4.2.

Tabel 4.2 Daftar Kebutuhan Fungsional

ID	Requirements	Aktor	Nama Use Case
SRS_OO1_01	Sistem harus menyediakan antarmuka untuk melakukan <i>Log In</i>	Admin, Operator	<i>Log In</i>
SRS_OO1_02	Sistem harus menyediakan antarmuka untuk melakukan <i>Log Out</i>	Admin, Operator	<i>Log Out</i>
SRS_OO1_03	Sistem harus menyediakan antarmuka untuk mengolah data <i>account</i> .	Admin	Mengolah Data <i>Account</i>
SRS_OO1_04	Sistem harus	Admin,	Mengolah Data

	menyediakan antarmuka mengolah mahasiswa	Operator	Mahasiswa
SRS_OO1_05	Sistem menyediakan antarmuka mengolah pelatihan	harus Admin	Mengolah Data Pelatihan
SRS_OO1_06	Sistem menyediakan antarmuka mengolah data tes	harus Admin, Operator	Mengolah Data Hasil Tes
SRS_OO1_07	Sistem menyediakan antarmuka melakukan tes	harus Mahasiswa	Melakukan Tes
SRS_OO1_08	Sistem menyediakan antarmuka menampilkan dari perhitungan dilakukan sistem.	harus Mahasiswa	Mendapatkan Hasil

Sumber: Perancangan

Daftar kebutuhan non-fungsional aplikasi sistem pendukung keputusan diperlihatkan pada Tabel 4.3.

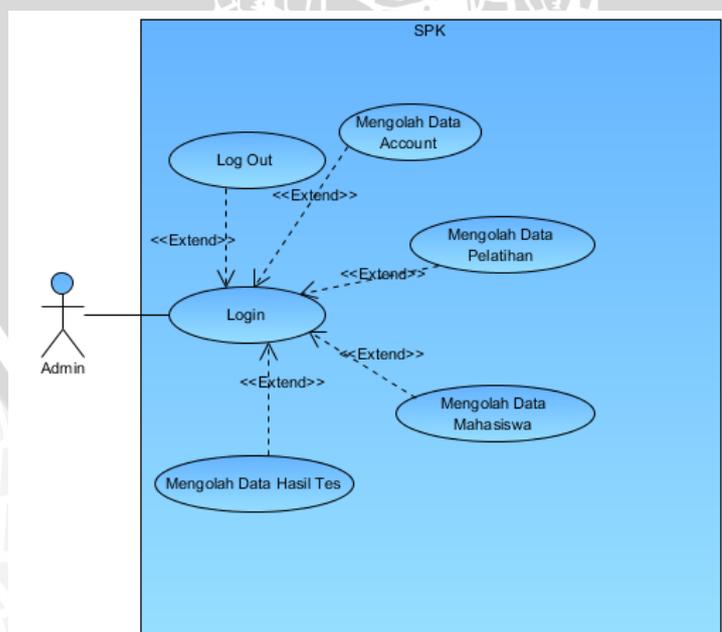
Tabel 4.3 Daftar Kebutuhan Non-Fungsional

Parameter	Deskripsi Kebutuhan
<i>Availability</i>	Aplikasi ini harus dapat beroperasi selama waktu yang ditentukan.
<i>Response Time</i>	Aplikasi ini harus cepat dalam melakukan proses penyimpanan data, pengubahan data, pencarian data, penghapusan data, dan penghitungan data.
<i>Security</i>	Aplikasi ini harus aman, karena terdapat data penting. <i>Security</i> pada sistem ini menggunakan fungsi Login.
<i>Memory</i>	Aplikasi ini harus ringan dan tidak membutuhkan <i>memory</i> yang besar.

Sumber: Perancangan

4.1.3 Diagram Use Case

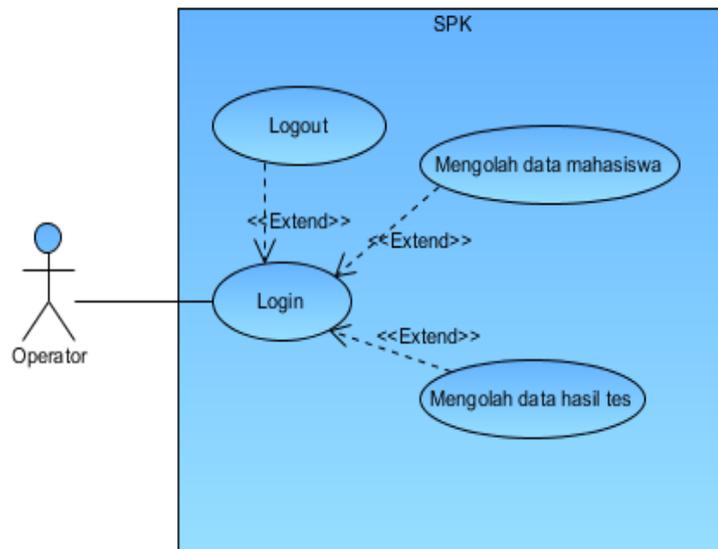
Kebutuhan-kebutuhan fungsional yang diperlukan oleh pengguna dan harus disediakan oleh sistem akan dimodelkan dalam diagram *use case*. Secara keseluruhan sistem ini memiliki 8 buah *use case*, yaitu *Log In*, *Log Out*, Mengolah Data *Account*, Mengolah Data Mahasiswa, Mengolah Data Pelatihan, Mengolah Data Hasil Tes, Melakukan Tes, dan Mendapatkan Hasil.



Gambar 4.1 Diagram Use Case untuk Admin

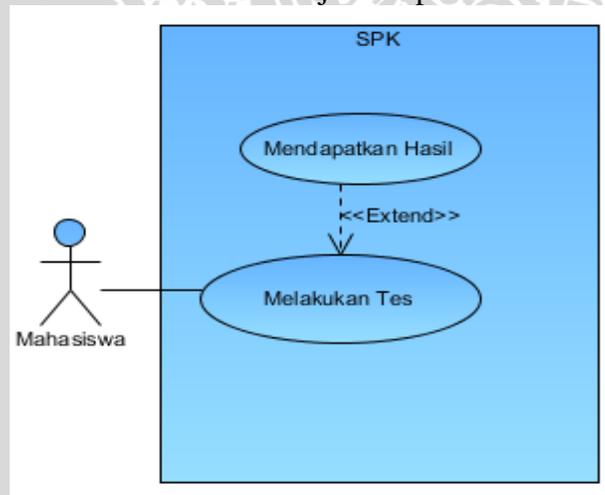
Sumber: Perancangan

Diagram *use case* untuk Operator ditunjukkan pada Gambar 4.2.



Gambar 4.2 Diagram *Use Case* untuk Operator
Sumber: Perancangan

Diagram *use case* untuk Mahasiswa ditunjukkan pada Gambar 4.3.



Gambar 4.3 Diagram *Use Case* untuk Mahasiswa
Sumber: Perancangan

4.1.4 Skenario *Use Case*

Secara lebih mendetail, masing-masing *use case* yang terdapat pada diagram *use case*, dijabarkan dalam skenario *use case*. Di dalam skenario *use case*, akan diberikan uraian nama *use case*, aktor yang berhubungan dengan *use case* tersebut, deskripsi global tentang *use case*, pra-kondisi yang harus dipenuhi dan kondisi akhir yang diharapkan setelah berjalannya fungsional *use case*.

Sistem juga akan diberikan ulasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang diberikan oleh aktor (aliran utama), serta kejadian alternatif.

Tabel 4.4 merupakan skenario *use case Log In*.

Tabel 4.4 Skenario *Use Case Log In*

Identifikasi	
Nama	<i>Log In</i>
Deskripsi	Use case ini menjelaskan proses melakukan <i>Log In</i>
Aktor	Admin, Operator
Pra-kondisi	Sistem siap untuk dijalankan aktor
Proses Skenario	
Aksi Aktor	Reaksi Sistem
1. Aktor menjalankan sistem	2. Sistem menampilkan halaman Login untuk memasukkan <i>username</i> dan <i>password</i>
3. Aktor memasukkan <i>username</i> dan <i>password</i> , kemudian memilih untuk melakukan proses login	4. Sistem melakukan pengecekan pada basis data, jika <i>username</i> dan <i>password</i> valid, maka sistem akan menampilkan halaman utama
Eksespsi Jika Gagal <i>Log In</i>	
	5. Sistem menampilkan pesan peringatan “ <i>Username</i> atau <i>password</i> yang anda masukkan salah!”.
Kondisi Akhir	Sistem menampilkan halaman utama

Sumber: Perancangan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk *log out*. Kebutuhan tersebut direpresentasikan oleh *use case Log Out*. Tabel 4.5 merupakan skenario *use case Log Out*.

Tabel 4.5 Skenario *Use Case Log Out*

Identifikasi	
Nama	<i>Log Out</i>
Deskripsi	Use case ini menjelaskan proses melakukan <i>Log Out</i>
Aktor	Admin, Operator
Pra-kondisi	Sistem telah menampilkan halaman utama
Proses Skenario	
Aksi Aktor	Reaksi Sistem
1. Aktor memilih untuk melakukan proses <i>log out</i>	2. Sistem menutup halaman utama
	3. Sistem menampilkan kembali halaman Login
Kondisi Akhir	Sistem menampilkan halaman Login

Sumber: Perancangan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan mengubah *password*. Kebutuhan tersebut direpresentasikan oleh *use case* Mengolah Data *Account*. Tabel 4.6 merupakan skenario *use case* Mengolah Data *Account*.

Tabel 4.6 Skenario *Use Case Mengolah Data Account*

Identifikasi	
Nama	Mengolah Data <i>Account</i>
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana proses memasukkan <i>account</i> baru, mengubah <i>account</i> , dan menghapus <i>account</i> .
Aktor	Admin
Pra-kondisi	Sistem telah menampilkan halaman utama

Skenario: Memasukkan <i>Account</i> Baru	
Aksi Aktor	Reaksi Sistem
1. Aktor memilih untuk menuju halaman <i>Data Account</i>	2. Sistem menampilkan halaman <i>Data Account</i>
3. Aktor memilih untuk memasukkan data <i>account</i> baru dan memilih untuk melakukan proses penyimpanan.	4. Sistem melakukan proses penyimpanan data <i>account</i> baru dalam basis data.
	5. Sistem menampilkan pesan "Data account telah tersimpan!"
6. Aktor menutup pesan	7. Sistem menampilkan kembali halaman <i>Data Account</i> .
Skenario: Mengubah <i>Data Account</i>	
1. Aktor memilih <i>account user</i> yang akan diubah. Kemudian melakukan proses perubahan data <i>account</i> .	2. Sistem melakukan proses penyimpanan data <i>account</i> baru dalam basis data.
	3. Sistem menampilkan pesan "Data account telah tersimpan!"
4. Aktor menutup pesan.	5. Sistem menampilkan kembali halaman <i>Data Account</i> .
Skenario: Menghapus <i>Data Account</i>	
1. Aktor memilih <i>account user</i> yang akan dihapus. Kemudian melakukan proses penghapusan data <i>account</i> .	2. Sistem melakukan proses penghapusan data <i>account</i> dari basis data.
	3. Sistem menampilkan pesan "Data account telah terhapus!"
4. Aktor menutup pesan	5. Sistem menampilkan kembali halaman <i>Data Account</i>
Kondisi Akhir	Sistem akan menyimpan hasil pengolahan data <i>account user</i> ke dalam basis data.

Sumber: Perancangan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan mengolah data mahasiswa. Kebutuhan tersebut direpresentasikan oleh *use case* Mengolah Data Mahasiswa. Tabel 4.7 merupakan skenario *use case* Mengolah Data Mahasiswa.

Tabel 4.7 Skenario *Use Case* Mengolah Data Mahasiswa

Identifikasi	
Nama	Mengolah Data Mahasiswa
Deskripsi	<i>Use case</i> ini menjelaskan proses mengolah data mahasiswa, yang terdiri atas memasukkan, mengubah, menghapus, dan mencari data mahasiswa
Aktor	Admin, Operator
Pra-kondisi	Sistem telah menampilkan halaman utama
Skenario Memasukkan Data Mahasiswa	
Aksi Aktor	Reaksi Sistem
1. Aktor memilih untuk menuju halaman Data Mahasiswa	2. Sistem menampilkan halaman Data Mahasiswa yang terdiri atas form input data, form pencarian data, dan tabel untuk menampilkan data.
3. Aktor memasukkan nim, nama, jenis kelamin, dan angkatan pada form input data. Kemudian memilih untuk melakukan proses penyimpanan	4. Sistem melakukan pengecekan pada basis data, jika data belum ada, maka data akan disimpan ke dalam basis data.
	5. Sistem menampilkan pesan "Data telah tersimpan!"
6. Aktor memilih untuk menutup kotak dialog pesan.	7. Sistem menampilkan kembali halaman Data Mahasiswa
Eksespsi : Jika Gagal Memasukkan Data	
	8. Sistem akan menampilkan pesan "Data sudah ada dalam

basis data!”

Skenario Mencari Data Mahasiswa

- | | |
|--|--|
| <p>1. Aktor memasukkan nim mahasiswa yang dicari pada form pencarian data. Kemudian memilih untuk melakukan proses pencarian</p> | <p>2. Sistem akan melakukan pengecekan pada basis data, jika data ada, maka data akan ditampilkan pada tabel</p> |
|--|--|

Eksepsi Jika Gagal Melakukan Pencarian Data

3. Sistem akan menampilkan pesan “Data tidak ada dalam basis data!”

Skenario Mengubah Data Mahasiswa

- | | |
|---|---|
| <p>1. Aktor memilih data mahasiswa yang diubah pada tabel. Kemudian memilih untuk melakukan proses <i>edit</i> data.mahasiswa</p> | <p>2. Sistem menampilkan data yang akan diubah pada form input data</p> |
| <p>3. Aktor mengubah nim, nama, jenis kelamin, atau angkatan. Kemudian memilih untuk melakukan proses penyimpanan data</p> | <p>4. Sistem menyimpan data yang diubah dalam basis data</p> |

5. Sistem menampilkan pesan “Data telah tersimpan!”

- | | |
|--|---|
| <p>6. Aktor memilih untuk menutup kotak dialog pesan</p> | <p>7. Sistem menampilkan kembali halaman Data Mahasiswa</p> |
|--|---|

Skenario Menghapus Data Mahasiswa

- | | |
|--|--|
| <p>1. Aktor memilih data mahasiswa yang akan dihapus pada tabel. Kemudian memilih untuk melakukan proses penghapusan data.</p> | <p>2. Sistem akan menghapus data dari basis data</p> |
|--|--|

3. Sistem akan menampilkan pesan “Data telah terhapus!”

- | | |
|--|---|
| <p>4. Aktor memilih untuk menutup kotak dialog pesan</p> | <p>5. Sistem menampilkan kembali halaman Data Mahasiswa</p> |
|--|---|



Kondisi Akhir	Sistem akan menyimpan hasil pengolahan data mahasiswa ke dalam basis data.
----------------------	--

Sumber: Perancangan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk mengolah data pelatihan. Kebutuhan tersebut direpresentasikan oleh *use case* Mengolah Data Pelatihan. Tabel 4.8 merupakan skenario *use case* Mengolah Data Pelatihan.

Tabel 4.8 Skenario *Use Case* Mengolah Data Pelatihan

Identifikasi	
Nama	Mengolah Data Pelatihan
Deskripsi	<i>Use case</i> ini menjelaskan proses mengolah data pelatihan, yang terdiri atas memasukkan, mengubah, menghapus, mencari, dan melakukan pengujian data pelatihan
Aktor	Admin
Pra-kondisi	Sistem telah menampilkan halaman utama
Skenario Memasukkan Data Pelatihan	
Aksi Aktor	Reaksi Sistem
1. Aktor memilih halaman Data Pelatihan	2. Sistem menampilkan halaman Data Pelatihan terdiri atas form input data, form pencarian data, dan tabel untuk menampilkan data.
3. Aktor memasukkan nilai data pelatihan dan konsentrasi pada form input data. Kemudian memilih untuk melakukan proses penyimpanan data	4. Sistem melakukan pengecekan pada basis data, jika data belum ada, maka data akan disimpan ke dalam basis data.
	5. Sistem menampilkan pesan "Data telah tersimpan!"
6. Aktor memilih untuk menutup	7. Sistem menampilkan kembali

kotak dialog pesan	halaman Data Pelatihan
Eksepsi : Jika Gagal Memasukkan Data	
	8. Sistem akan menampilkan pesan “Data sudah ada dalam basis data!”
Skenario Mencari Data Pelatihan	
1. Aktor memasukkan kode pelatihan yang dicari pada form pencarian data. Kemudian memilih untuk melakukan proses pencarian	2. Sistem akan melakukan pengecekan pada basis data, jika data ada, maka data akan ditampilkan pada tabel
Eksepsi Jika Gagal Melakukan Pencarian Data	
	3. Sistem akan menampilkan pesan “Data tidak ada dalam basis data!”
Skenario Mengubah Data Pelatihan	
1. Aktor memilih data pelatihan yang diubah pada tabel. Kemudian memilih untuk melakukan proses <i>edit</i> data pelatihan	2. Sistem menampilkan data yang akan diubah pada form input data
3. Aktor mengubah nilai data pelatihan atau konsentrasi. Kemudian memilih untuk melakukan proses penyimpanan data pelatihan	4. Sistem menyimpan data yang diubah dalam basis data
	5. Sistem menampilkan pesan “Data telah tersimpan!”
6. Aktor memilih untuk menutup kotak dialog pesan	7. Sistem menampilkan kembali halaman Data Pelatihan
Skenario Menghapus Data Pelatihan	
1. Aktor memilih data pelatihan yang akan dihapus pada tabel. Kemudian memilih untuk melakukan proses hapus data pelatihan.	2. Sistem akan menghapus data dari basis data

	3. Sistem akan menampilkan pesan “Data telah terhapus!”
4. Aktor memilih untuk menutup kotak dialog pesan	5. Sistem menampilkan kembali halaman Data Pelatihan
Skenario: Menguji Data Pelatihan	
1. Aktor memilih data pelatihan yang akan diuji. Kemudian memilih untuk melakukan proses penghitungan	2. Sistem akan melakukan proses dan menampilkan hasil perhitungan.
3. Aktor memilih untuk menyimpan hasil perhitungan ke dalam suatu file.	4. Sistem melakukan penyimpanan hasil perhitungan ke dalam file.
Kondisi Akhir	Sistem akan menyimpan hasil pengolahan data pelatihan ke dalam basis data.

Sumber: Perancangan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk mengolah data hasil tes. Kebutuhan tersebut direpresentasikan oleh *use case* Mengolah Data Hasil Tes. Tabel 4.9 merupakan skenario *use case* Mengolah Data Hasil Tes.

Tabel 4.9 Skenario *Use Case* Mengolah Data Hasil Tes

Identifikasi	
Nama	Mengolah Data Hasil Tes
Deskripsi	<i>Use case</i> ini menjelaskan proses mengolah data hasil tes, yang terdiri atas menghapus dan mencari data hasil tes
Aktor	Admin, Operator
Pra-kondisi	Sistem telah menampilkan halaman utama
Skenario Mencari Data Hasil Tes	
Aksi dari Aktor	Reaksi Sistem
1. Aktor memilih halaman Data	2. Sistem menampilkan halaman



Hasil Tes	Data Hasil Tes terdiri atas form pencarian data dan tabel untuk menampilkan data.
3. Aktor memasukkan kode tes yang dicari pada form pencarian data. Kemudian memilih untuk melakukan proses pencarian data	4. Sistem akan melakukan pengecekan pada basis data, jika data ada, maka data akan ditampilkan pada tabel
Eksepsi Jika Gagal Melakukan Pencarian Data	
	5. Sistem akan menampilkan pesan "Data tidak ada dalam basis data!"
Skenario Menghapus Data Hasil Tes	
1. Aktor memilih data hasil tes yang akan dihapus pada tabel. Kemudian memilih untuk melakukan proses hapus data	2. Sistem akan menghapus data dari basis data
	3. Sistem akan menampilkan pesan "Data telah terhapus!"
4. Aktor memilih untuk menutup kotak dialog pesan	5. Sistem menampilkan kembali halaman Data Hasil Tes
Hasil Akhir	Sistem akan menyimpan hasil pengolahan data hasil tes ke dalam basis data.

Sumber: Perancangan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk melakukan tes. Kebutuhan tersebut direpresentasikan oleh *use case* Melakukan Tes. Tabel 4.10 merupakan skenario *use case* Melakukan Tes.

Tabel 4.10 Skenario *Use Case* Melakukan Tes

Identifikasi	
Nama	Melakukan Tes
Deskripsi	<i>Use case</i> ini menjelaskan proses

	melakukan tes
Aktor	Mahasiswa
Pra-kondisi	Sistem telah menampilkan halaman Tes Mahasiswa
Skenario	
Aksi Aktor	Reaksi Sistem
1. Aktor memasukkan nim dan nama	2. Sistem menampilkan kode tes dan tanggal tes
3. Aktor memilih untuk memasukkan nilai mata kuliah	4. Sistem menyimpan nilai mata kuliah dalam basis data
Hasil Akhir	Jawaban disimpan dalam basis data untuk dilakukan proses perhitungan

Sumber: Perancangan

Kebutuhan fungsional selanjutnya yang harus disediakan oleh sistem adalah kebutuhan untuk mendapatkan hasil. Kebutuhan tersebut direpresentasikan oleh *use case* Mendapatkan Hasil. Tabel 4.11 merupakan skenario *use case* Mendapatkan Hasil.

Tabel 4.11 Skenario *Use Case* Mendapatkan Hasil

Identifikasi	
Nama	Mendapatkan Hasil
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana proses mendapatkan hasil keputusan
Aktor	Mahasiswa
Pra-kondisi	Sistem telah menampilkan halaman Tes Mahasiswa
Skenario	
Aksi Aktor	Reaksi Sistem
1. Aktor memilih untuk melakukan proses perhitungan	2. Sistem menampilkan halaman Pendukung Keputusan
3. Aktor memasukkan nilai K dan	4. Sistem akan melakukan proses

memilih salah satu konsentrasi yang diminati. Kemudian memilih untuk melakukan proses perhitungan	dan menampilkan hasil perhitungan
Hasil Akhir	Menampilkan hasil analisis sistem

Sumber: Perancangan

4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan dalam dua tahap, yaitu perancangan umum dan perancangan detail. Perancangan umum meliputi diagram blok sistem dan diagram blok sistem pendukung keputusan. Perancangan detail meliputi diagram kelas, diagram sekuensial dan diagram aktivitas.

4.2.1 Perancangan Umum

Perancangan umum dalam penelitian ini meliputi perancangan sistem secara keseluruhan dan perancangan sistem pendukung keputusan.

4.2.1.1 Diagram Blok Sistem

Secara umum diagram blok sistem secara keseluruhan digambarkan pada Gambar 4.4 berikut:



Gambar 4.4 Diagram Blok Sistem

Sumber: Perancangan

- *User*

User merupakan pengguna yang berinteraksi dengan sistem, seperti admin yang bertugas untuk mengolah sistem dan mahasiswa yang menggunakan sistem untuk mendukung keputusan.

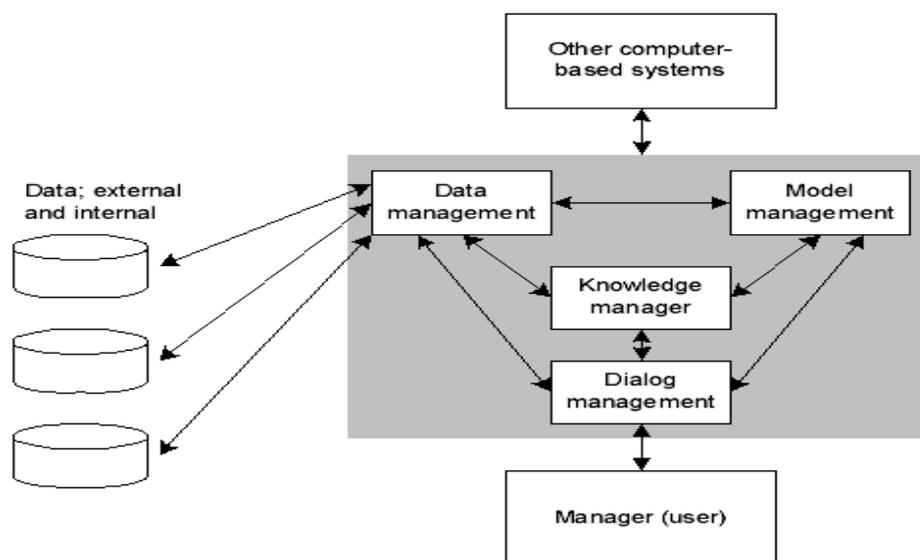
- Sistem Pendukung Keputusan

Dalam aplikasi ini, sistem pendukung keputusan berperan sebagai sistem yang membantu mahasiswa dalam menentukan keputusan. Metode yang dikembangkan dalam sistem ini yaitu *K-Nearest Neighbor*.

- My SQL

My SQL merupakan tempat penyimpanan basis data yang diperlukan oleh sistem, seperti basis data *user*, data mahasiswa, data pelatihan, dan data hasil tes.

4.2.1.2 Perancangan Sistem Pendukung Keputusan



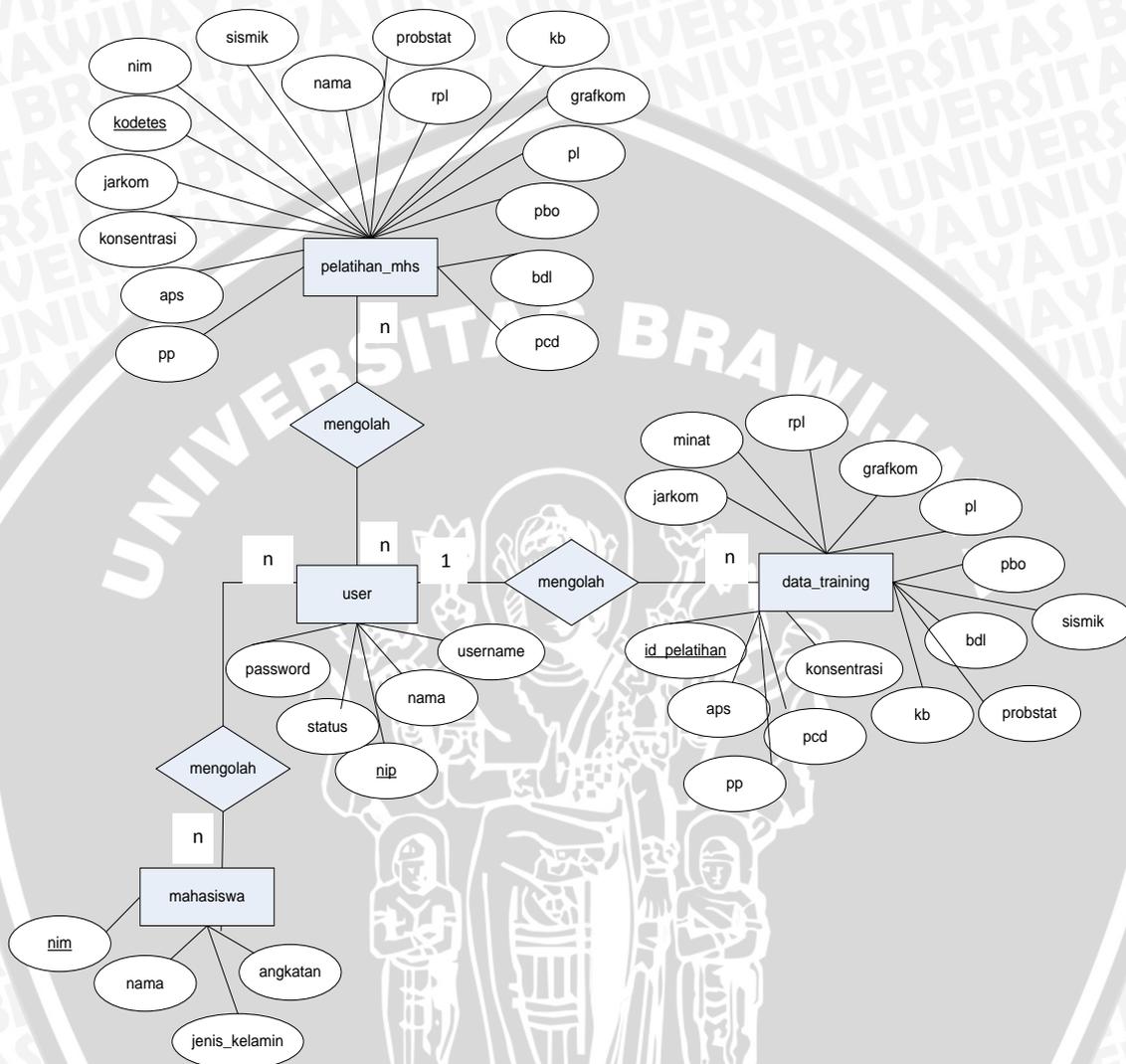
Gambar 4.5 Diagram Konseptual Sistem Pendukung Keputusan

Sumber: [SUB-02:21]

- *Data Management*

Data management, termasuk *basis data*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System (DBMS)*. Dalam sistem ini DBMS yang digunakan yaitu My Sql. Pada perancangan basis data sistem ini menggunakan lima tabel yaitu meliputi tabel user, tabel mahasiswa, tabel kuesioner, tabel data_training, dan tabel

pelatihan_mhs. Adapun perancangan tabel *Entity Relationship Diagram* basis data sistem ini diperlihatkan pada Gambar 4.6.



Gambar 4.6 *Entity Relationship Diagram*

Sumber: Perancangan

- **Model Management**

Model management, melibatkan model finansial, statistikal, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen *software* yang diperlukan. Pada sistem pendukung keputusan ini, pemodelan yang digunakan yaitu pemodelan

kuantitatif dengan memanfaatkan algoritma dari *K-Nearest Neighbor*.

- *Communication (Dialog Subsystem)*

User dapat berkomunikasi dan memberikan perintah pada Sistem Pendukung Keputusan melalui sub sistem ini. Ini berarti menyediakan antarmuka. Perancangan *user interface* sistem ini dijelaskan pada sub bab Perancangan Antarmuka.

- *Knowledge Management*

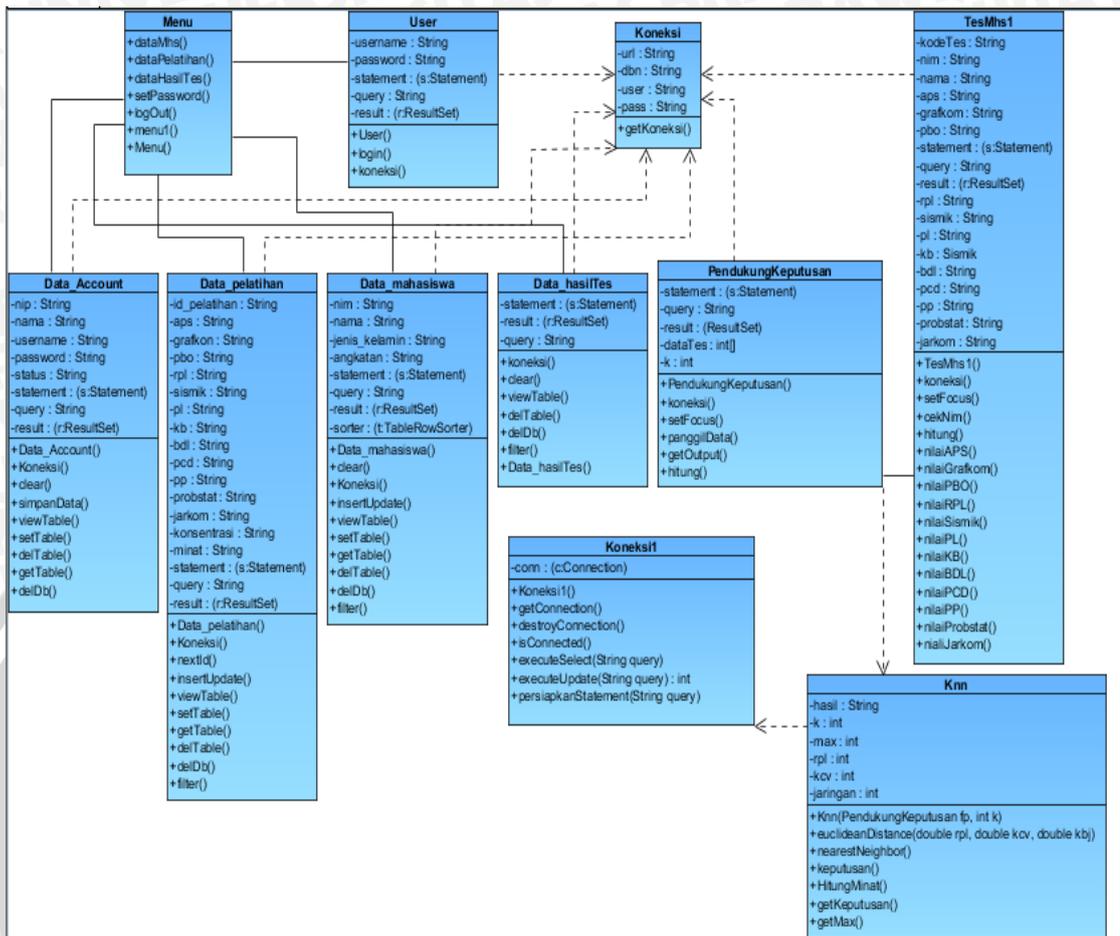
Subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri. Komponen *knowledge management* terdiri dari satu atau beberapa *expert system*. Pada sistem ini tidak mengikutsertakan komponen *knowledge management* sebagai bagian dari sistem pendukung keputusan.

4.2.2 Perancangan Detail

Perancangan detail menggunakan diagram kelas (*class diagram*) dan diagram sekuensial (*sequential diagram*). Perancangan detail menjelaskan mengenai pola hubungan antar komponen-komponen detail (kelas dan objek), sehingga mampu membentuk sebuah fungsi yang mampu memberikan pelayanan terhadap kebutuhan pengguna.

4.2.2.1 Diagram Kelas (*Class Diagram*)

Diagram kelas memberikan gambaran pemodelan elemen-elemen kelas yang membentuk sebuah sistem perangkat lunak mulai dari atribut-atribut serta operasi-operasinya dan hubungannya dengan kelas-kelas lain dalam sebuah sistem. Pada sistem ini terdapat sejumlah kelas yang saling membentuk relasi. Diagram kelas dari sistem ini ditunjukkan pada Gambar 4.7.



Gambar 4.7 Diagram Kelas Sistem
Sumber: Perancangan

Deskripsi atribut dan operasi dari kelas Koneksi diperlihatkan pada tabel 4.12 berikut:

Tabel 4.12 Kelas Koneksi

Nama Kelas: Koneksi
Deskripsi: Kelas Koneksi merupakan kelas yang digunakan oleh kelas lain untuk melakukan koneksi dengan basis data.
Atribut
Nama Atribut: url
Deskripsi: atribut ini digunakan untuk menyimpan alamat dari tempat penyimpanan basis data.



Nama Atribut: user
Deskripsi: atribut ini digunakan untuk menyimpan nama dari <i>user</i> basis data.
Nama Atribut: dbn
Deskripsi: atribut ini digunakan untuk menyimpan nama dari basis data.
Nama Atribut: pass
Deskripsi: atribut ini digunakan untuk menyimpan <i>password</i> basis data.
Operation
Nama Operasi: getKoneksi()
Deskripsi: merupakan operasi yang digunakan untuk melakukan koneksi dengan basis data.

Sumber: Perancangan

Deskripsi atribut dan operasi dari kelas Koneksi1 diperlihatkan pada tabel 4.13 berikut:

Tabel 4.13 Kelas Koneksi1

Nama kelas: Koneksi1
Deskripsi: Kelas Koneksi1 merupakan kelas yang digunakan oleh kelas Knn untuk melakukan koneksi dengan basis data, selain itu kelas ini memiliki operasi-operasi yang lebih detail jika dibandingkan dengan kelas Koneksi.
Operasi
Nama Operasi: getConnection()
Deskripsi: merupakan operasi yang digunakan untuk melakukan koneksi dengan basis data.
Nama Operasi: destroyConnection()
Deskripsi: merupakan operasi yang digunakan untuk menghapus koneksi dengan basis data.
Nama Operasi: isConnected()
Deskripsi: merupakan operasi yang digunakan untuk menentukan apakah telah terkoneksi dengan basis data.

Nama Operasi: executeSelect()

Deskripsi: merupakan operasi yang digunakan untuk mengeksekusi *select query* yang dilakukan.

Nama Operasi: executeUpdate()

Deskripsi: merupakan operasi yang digunakan untuk mengeksekusi *update query* yang dilakukan.

Nama Operasi: persiapkanStatement()

Deskripsi: merupakan operasi yang digunakan untuk mempersiapkan statement eksekusi query.

Sumber: Perancangan

Deskripsi atribut dan operasi dari kelas Menu diperlihatkan pada tabel 4.14 berikut:

Tabel 4.14 Kelas Menu

Nama Kelas: Menu
Deskripsi: merupakan kelas yang digunakan untuk mengakses menu yang disediakan sistem
Operasi
Nama operasi: dataMhs() Deskripsi: merupakan operasi yang digunakan untuk menampilkan halaman Data Mahasiswa
Nama operasi: dataNilai() Deskripsi: merupakan operasi yang digunakan untuk menampilkan halaman Data Nilai
Nama operasi: dataPelatihan() Deskripsi: merupakan operasi yang digunakan untuk menampilkan halaman Data Pelatihan
Nama operasi: dataHasilTes() Deskripsi: merupakan operasi yang digunakan untuk menampilkan halaman Data Hasil Tes
Nama operasi: dataAccount() Deskripsi: merupakan operasi yang digunakan untuk menampilkan halaman Data Account

Nama operasi: menu1()

Deskripsi: merupakan operasi yang digunakan untuk menon-aktifkan halaman khusus untuk operator

Sumber: Perancangan

Deskripsi atribut dan operasi dari kelas Data_mahasiswa diperlihatkan pada tabel 4.15 berikut:

Tabel 4.15 Kelas Data_mahasiswa

Nama Kelas: Data_mahasiswa	
Deskripsi:	Merupakan kelas yang digunakan untuk mengelola data yang berhubungan dengan data mahasiswa.
Atribut	
Nama Atribut: nim	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nim mahasiswa
Nama Atribut: nama	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nama mahasiswa
Nama Atribut: jenis_kelamin	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan jenis kelamin mahasiswa
Nama Atribut: angkatan	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan angkatan mahasiswa
Nama Atribut: statement	
Deskripsi:	merupakan atribut yang digunakan untuk melakuka pengekseskuan query
Nama Atribut: result	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan hasil eksekusi query
Nama Atribut: query	

Deskripsi: merupakan atribut yang digunakan untuk menyimpan perintah query

Nama Atribut: sorter

Deskripsi: merupakan atribut yang digunakan untuk melakukan pengurutan data dalam proses pencarian data mahasiswa.

Operation

Nama operasi: clear()

Deskripsi: merupakan operasi yang digunakan untuk mengatur agar form input data diatur kosong.

Nama operasi: koneksi()

Deskripsi: merupakan operasi yang digunakan untuk melakukan koneksi dengan basis data.

Nama operasi: insertUpdate()

Deskripsi: merupakan operasi yang dipanggil ketika memasukkan dan mengubah data mahasiswa.

Nama operasi: setTable()

Deskripsi: merupakan operasi yang digunakan untuk memasukkan data ke dalam tabel

Nama operasi: delTable()

Deskripsi: merupakan operasi yang digunakan untuk menghapus data dari tabel

Nama operasi: viewTable()

Deskripsi: merupakan operasi yang digunakan untuk menampilkan data pada tabel.

Nama operasi: getTable()

Deskripsi: merupakan operasi yang digunakan untuk mengubah data yang dipilih dari tabel.

Nama operasi: delDb()

Deskripsi: merupakan operasi yang digunakan untuk menghapus data dari basis data.

Nama operasi: filter()

Deskripsi: merupakan operasi yang digunakan untuk melakukan pencarian di

tabel.

Sumber: Perancangan

Deskripsi atribut dan operasi dari kelas Data_pelatihan diperlihatkan pada tabel 4.16 berikut:

Tabel 4.16 Kelas Data_pelatihan

Nama Kelas: Data_pelatihan	
Deskripsi:	Merupakan kelas yang digunakan untuk mengelola data pelatihan.
Atribut	
Nama Atribut: id_pelatihan	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan id pelatihan.
Nama Atribut: aps	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nilai Analisis dan Perancangan Sistem.
Nama Atribut: grafkom	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nilai Grafika Komputer..
Nama Atribut: pbo	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nilai Pemrograman Berorientasi Objek.
Nama Atribut: rpl	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nilai Rekayasa Perangkat Lunak.
Nama Atribut: sismik	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nilai Sistem Mikroprosesor.
Nama Atribut: pl	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nilai Pemrograman Lanjut.

Nama Atribut: kb

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Kecerdasan Buatan.

Nama Atribut: bdl

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Basis Data Lanjut.

Nama Atribut: pcd

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Pengolahan Citra Digital.

Nama Atribut: pp

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Pengenalan Pola.

Nama Atribut: probstat

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Probabilitas dan Statistika.

Nama Atribut: jarkom

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Jaringan Komputer.

Nama Atribut: statement

Deskripsi: merupakan atribut yang digunakan untuk melakukan pengeksekusian query.

Nama Atribut: query

Deskripsi: merupakan atribut yang digunakan untuk menyimpan operasi query yang dijalankan.

Nama Atribut: result

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai dari eksekusi query.

Nama Atribut: sorter

Deskripsi: merupakan atribut yang digunakan untuk melakukan pengurutan dalam pencarian data.

Operation

<p>Nama operasi: koneksi()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk melakukan koneksi dengan basis data.</p>
<p>Nama operasi: nextId()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk melakukan penambahan jumlah pada “id pelatihan”.</p>
<p>Nama operasi: insertUpdate()</p> <p>Deskripsi: merupakan operasi yang dipanggil ketika memasukkan dan mengubah data.</p>
<p>Nama operasi: setTable()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk memasukkan data ke dalam tabel.</p>
<p>Nama operasi: delTable()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk menghapus data dari tabel.</p>
<p>Nama operasi: viewTable()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk menampilkan data dalam tabel.</p>
<p>Nama operasi: getTable()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk mengubah data yang dipilih dari tabel.</p>
<p>Nama operasi: delDb()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk menghapus data dari basis data.</p>
<p>Nama operasi: filter()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk melakukan pencarian di tabel.</p>

Sumber: Perancangan

Deskripsi atribut dan operasi dari kelas Data_hasiltes diperlihatkan pada Tabel 4.17:

Tabel 4.17 Kelas Data_hasiltes

Nama Kelas: Data_hasiltes
<p>Deskripsi:</p> <p>Merupakan kelas yang digunakan untuk mengelola data yang berhubungan dengan data hasil tes.</p>
Atribut
<p>Nama Atribut: statement</p> <p>Deskripsi: merupakan atribut yang digunakan untuk melakukan pengekseskuan query.</p>
<p>Nama Atribut: query</p> <p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan operasi query yang dijalankan.</p>
<p>Nama Atribut: result</p> <p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai dari eksekusi query.</p>
Operation
<p>Nama operasi: clear()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk mengatur agar form input data diatur kosong.</p>
<p>Nama operasi: koneksi()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk melakukan koneksi dengan basis data.</p>
<p>Nama operasi: delTable()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk menghapus data dari tabel.</p>
<p>Nama operasi: viewTable()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk menampilkan data dalam tabel.</p>
<p>Nama operasi: delDb()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk menghapus data dari basis data.</p>

Nama operasi: filter()

Deskripsi: merupakan operasi yang digunakan untuk melakukan pencarian di tabel.

Sumber: Perancangan

Deskripsi atribut dan operasi dari kelas Data_Account diperlihatkan pada tabel 4.18 berikut:

Tabel 4.18 Kelas Data_Account

Nama Kelas:Data_Account	
Deskripsi:	Merupakan kelas yang digunakan untuk mengolah data <i>account</i> .
Atribut	
Nama Atribut: nip	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nip user.
Nama Atribut: nama	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nama user.
Nama Atribut: username	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan usernamr user.
Nama Atribut: password	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan password user.
Nama Atribut: status	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan status user.
Nama Atribut: statement	
Deskripsi:	merupakan atribut yang digunakan untuk melakukan pengekseskuan query.
Nama Atribut: query	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan operasi query yang dijalankan.
Nama Atribut: result	

<p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai dari eksekusi query.</p>
<p>Operation</p>
<p>Nama operasi: clear()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk mengatur agar form input data diatur kosong.</p>
<p>Nama operasi: koneksi()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk melakukan koneksi dengan basis data.</p>
<p>Nama operasi: simpanData()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk menyimpan <i>account</i> user baru</p>
<p>Nama operasi: viewTable()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk menampilkan data dalam tabel.</p>
<p>Nama operasi: setTable()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk memasukkan data ke dalam tabel</p>
<p>Nama operasi: getTable()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk mengubah data yang dipilih dari tabel.</p>
<p>Nama operasi: delDb()</p> <p>Deskripsi: merupakan operasi yang digunakan untuk menampilkan data dalam tabel.</p>

Sumber: Perancangan

Deskripsi atribut dan operasi dari kelas User diperlihatkan pada tabel 4.19 berikut:

Tabel 4.19 Kelas User

<p>Nama Kelas: User</p>
<p>Deskripsi:</p>

Merupakan kelas yang digunakan untuk masuk ke sistem.

Atribut

Nama Atribut: username

Deskripsi: merupakan atribut yang digunakan untuk menyimpan username yang dimasukkan user.

Nama Atribut: password

Deskripsi: merupakan atribut yang digunakan untuk menyimpan password yang dimasukkan user.

Nama Atribut: statement

Deskripsi: merupakan atribut yang digunakan untuk melakukan pengekseskuan query

Nama Atribut: query

Deskripsi: merupakan atribut yang digunakan untuk menyimpan operasi query yang dijalankan.

Nama Atribut: result

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai dari eksekusi query.

Operation

Nama operasi: clear()

Deskripsi: merupakan operasi yang digunakan untuk mengatur agar form input data diatur kosong.

Nama operasi: koneksi()

Deskripsi: merupakan operasi yang digunakan untuk melakukan koneksi dengan basis data.

Nama operasi: login()

Deskripsi: merupakan operasi yang digunakan untuk mengecek apakah *username* dan *password* yang dimasukkan *user* benar.

Sumber: Perancangan

Deskripsi atribut dan operasi dari kelas Knn diperlihatkan pada tabel 4.20.

Tabel 4.20 Kelas Knn

Nama kelas: Knn
<p>Deskripsi:</p> <p>Kelas Knn merupakan kelas yang berisi operasi-operasi untuk menghitung <i>K-Nearest Neighbor</i>.</p>
Atribut
<p>Nama atribut: k</p> <p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan jumlah kumpulan tetangga terdekat.</p>
<p>Nama atribut: max</p> <p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan mayoritas dari tetangga terdekat.</p>
<p>Nama Atribut: aps</p> <p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Analisis dan Perancangan Sistem.</p>
<p>Nama Atribut: grafkom</p> <p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Grafika Komputer..</p>
<p>Nama Atribut: pbo</p> <p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Pemrograman Berorientasi Objek.</p>
<p>Nama Atribut: rpl</p> <p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Rekayasa Perangkat Lunak.</p>
<p>Nama Atribut: sismik</p> <p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Sistem Mikroprosesor.</p>
<p>Nama Atribut: pl</p> <p>Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai</p>

Pemrograman Lanjut.
Nama Atribut: kb
Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Kecerdasan Buatan.
Nama Atribut: bdl
Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Basis Data Lanjut.
Nama Atribut: pcd
Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Pengolahan Citra Digital.
Nama Atribut: pp
Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Pengenalan Pola.
Nama Atribut: probstat
Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Probabilitas dan Statistika.
Nama Atribut: jarkom
Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Jaringan Komputer.
Nama atribut: hasil
Deskripsi: merupakan atribut yang digunakan untuk menampilkan hasil analisis sistem
Operasi
Nama operasi: euclideanDistance()
Deskripsi: merupakan operasi yang digunakan untuk menghitung jarak <i>euclidean</i> antara data pelatihan dengan data tes.
Nama operasi: nearestNeighbor()
Deskripsi: merupakan operasi yang digunakan untuk menentukan jarak terdekat berdasarkan jarak <i>euclidean</i> yang telah dihitung.
Nama operasi: keputusan()
Deskripsi: merupakan operasi yang digunakan untuk menampilkan hasil analisis

sistem berdasarkan perhitungan *K-Nearest Neighbor*.

Nama operasi: HitungMinat()

Deskripsi: merupakan operasi yang digunakan untuk menghitung prosentase kecocokan antara mahasiswa dengan hasil analisis sistem

Sumber:Perancangan

Deskripsi atribut dan operasi dari kelas TesMhs diperlihatkan pada tabel 4.21 berikut:

Tabel 4.21 Kelas TesMhs

Nama Kelas: TesMhs	
Deskripsi:	Merupakan kelas yang digunakan user untuk melakukan tes.
Atribut	
Nama Atribut: kodeTes	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan kode tes mahasiswa
Nama Atribut: nim	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nim mahasiswa
Nama Atribut: nama	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nama mahasiswa
Nama Atribut: aps	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nilai Analisis dan Perancangan Sistem.
Nama Atribut: grafkom	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nilai Grafika Komputer..
Nama Atribut: pbo	
Deskripsi:	merupakan atribut yang digunakan untuk menyimpan nilai Pemrograman Berorientasi Objek.

Nama Atribut: rpl

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Rekayasa Perangkat Lunak.

Nama Atribut: sismik

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Sistem Mikroprosesor.

Nama Atribut: pl

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Pemrograman Lanjut.

Nama Atribut: kb

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Kecerdasan Buatan.

Nama Atribut: bdl

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Basis Data Lanjut.

Nama Atribut: pcd

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Pengolahan Citra Digital.

Nama Atribut: pp

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Pengenalan Pola.

Nama Atribut: probstat

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Probabilitas dan Statistika.

Nama Atribut: jarkom

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai Jaringan Komputer.

Nama Atribut: statement

Deskripsi: merupakan atribut yang digunakan untuk melakukan pengeksekusian query.

Nama Atribut: query

Deskripsi: merupakan atribut yang digunakan untuk menyimpan operasi query yang dijalankan.

Nama Atribut: result

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai dari eksekusi query.

Operation

Nama operasi: setFocus()

Deskripsi: merupakan operasi yang digunakan untuk mengatur agar form input jawaban kosong.

Nama operasi: koneksi()

Deskripsi: merupakan operasi yang digunakan untuk melakukan koneksi dengan basis data.

Nama operasi: cekNim()

Deskripsi: merupakan operasi yang digunakan untuk mengecek data nim yang dimasukkan pada basis data.

Nama operasi: nilaiAPS()

Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Aplikasi dan Perancangan Sistem.

Nama operasi: nilaiGrafkomp()

Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Grafika Komputer.

Nama operasi: nilaiPBO()

Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Pmrograman Berorientasi Objek.

Nama operasi: nilaiRPL()

Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Rekayasa Perangkat Lunak.

Nama operasi: nilaiSismik()

Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Sistem Mikroprosesor.

Nama operasi: nilaiPL()

Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Pemrograman Lanjut.
Nama operasi: nilaiKB()
Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Kecerdasan Buatan.
Nama operasi: nilaiBDL()
Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Basis Data Lanjut.
Nama operasi: nilaiPCD()
Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Pengolahan Citra Digital.
Nama operasi: nilaiPP()
Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Pengenalan Pola.
Nama operasi: nilaiProbstat()
Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Probabilitas dan Statistika.
Nama operasi: nilaiJarkom()
Deskripsi: merupakan operasi yang digunakan untuk menampilkan nilai Jaringan Komputer.
Nama operasi: hitung()
Deskripsi: merupakan operasi yang digunakan untuk melakukan proses perhitungan nilai.

Sumber: Perancangan

Deskripsi atribut dan operasi dari kelas PendukungKeputusan diperlihatkan pada tabel 4.22 berikut:

Tabel 4.22 Kelas PendukungKeputusan

Nama Kelas: PendukungKeputusan
Deskripsi: Merupakan kelas yang digunakan untuk melakukan proses perhitungan nilai

dengan *K-Nearest Neighbor* dan menampilkan hasil analisis sistem.

Atribut

Nama Atribut: statement

Deskripsi: merupakan atribut yang digunakan untuk melakukan pengekseskuan query

Nama Atribut: query

Deskripsi: merupakan atribut yang digunakan untuk menyimpan operasi query yang dijalankan.

Nama Atribut: result

Deskripsi: merupakan atribut yang digunakan untuk menyimpan nilai dari eksekusi query.

Nama Atribut: dataTes

Deskripsi: merupakan atribut yang digunakan untuk menyimpan data tes mahasiswa.

Operation

Nama operasi: koneksi()

Deskripsi: merupakan operasi yang digunakan untuk melakukan koneksi dengan basis data.

Nama operasi: setFocus()

Deskripsi: merupakan operasi yang digunakan untuk mengatur letak kursor pada form input data.

Nama operasi: viewTable()

Deskripsi: merupakan operasi yang digunakan untuk menampilkan data dalam tabel.

Nama operasi: panggilData()

Deskripsi: merupakan operasi yang digunakan untuk menampilkan data tes yang ada pada kelas TesMhs.

Nama operasi: hitung()

Deskripsi: merupakan operasi yang digunakan untuk menghitung jawaban tes dengan algoritma *K Nearest Neighbor*.

Nama operasi: getOutput()

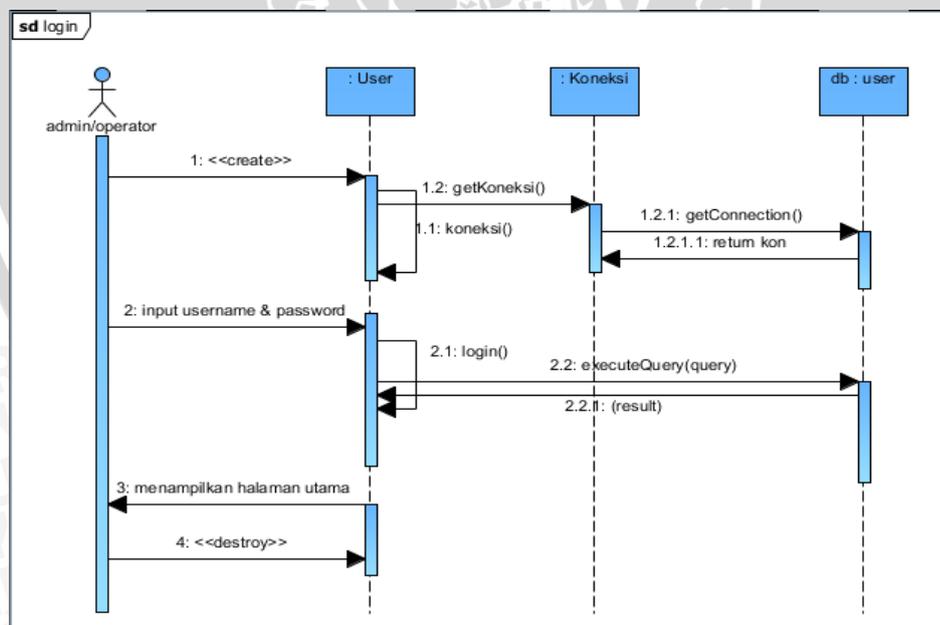
Deskripsi: merupakan operasi yang digunakan untuk menampilkan hasil perhitungan.

Sumber: Perancangan

4.2.2.2 Diagram Sekuensial (Sequence Diagram)

Dalam metodologi berorientasi objek untuk menunjukkan bagaimana suatu objek berkomunikasi dengan objek yang lain dengan memperhatikan urutan waktu yang dimodelkan didalam *sequence diagram*. *Sequence diagram* terdiri dari objek-objek yang dipresentasikan dalam pembelajaran didalamnya memuat nama dari objek, pengenalan objek, baca objek yang direpresentasikan dengan garis panah dan *time* (waktu) yang direpresentasikan dengan *vertical progression*. Berikut ini merupakan diagram sekuensial yang ada pada Sistem Pendukung Keputusan Pemilihan Konsentrasi di PS.Teknik Informatika Universitas Brawijaya.

- Diagram Sekuensial *Login*



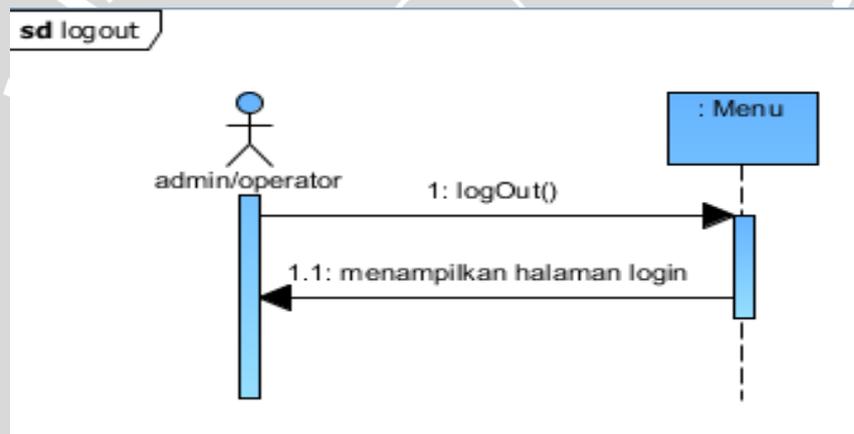
Gambar 4.8 Diagram Sekuensial *Login*

Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.8 sebagai berikut:

1. Admin/operator menjalankan program, kemudian sistem akan menampilkan halaman login dan sistem melakukan koneksi ke basis data dengan menggunakan operasi koneksi().
2. Admin/operator memasukkan *username* dan *password*. Kemudian sistem akan menjalankan operasi login() yang berisi perintah query ke basis data. Kemudian basis data akan mengirimkan hasil dari eksekusi query.
3. Sistem akan menampilkan halaman utama.
4. Sistem akan menutup halaman login.

- Diagram Sekuensial *Log Out*

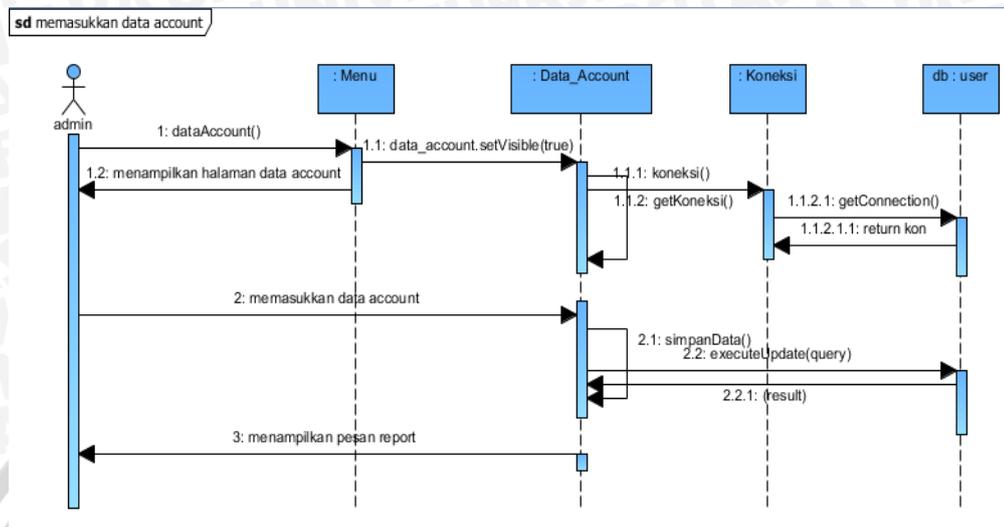


Gambar 4.9 Diagram Sekuensial Log Out
Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.9 sebagai berikut:

1. Admin/operator memilih untuk logout
2. Sistem menampilkan kembali halaman login

- Diagram Sekuensial Memasukkan Data *Account*

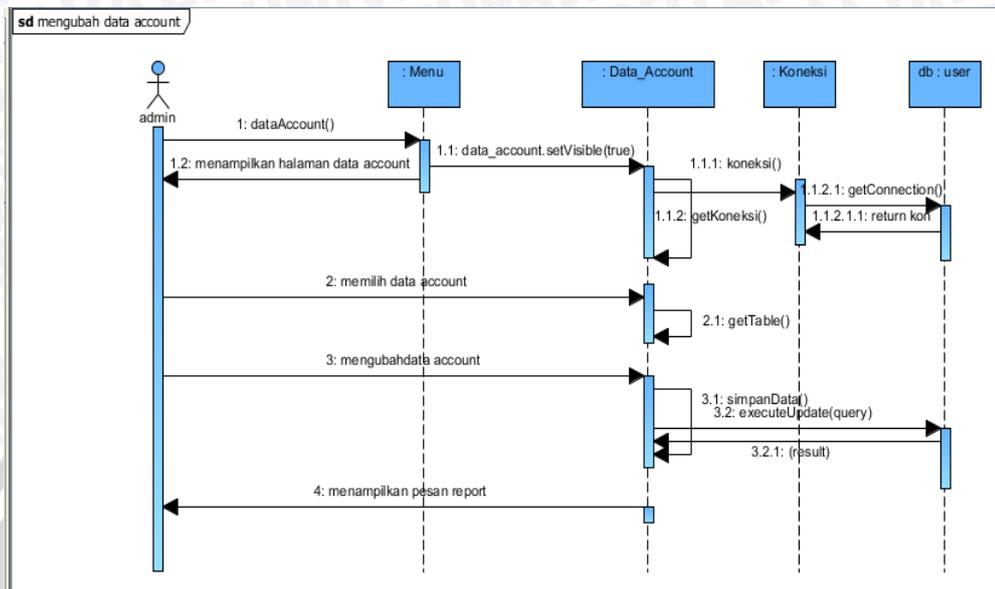


Gambar 4.10 Diagram Sekuensial Memasukkan Data *Account*
Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.10 sebagai berikut:

1. Admin memilih halaman *Data Account*, kemudian sistem menampilkan halaman *Data Account*. Sistem melakukan koneksi dengan basis data melalui operasi `koneksi()`. Di dalam operasi `koneksi()` sistem memanggil operasi `getKoneksi()` dan `getConnection()` untuk melakukan koneksi dengan basis data.
2. Admin memasukkan data *account*, kemudian sistem akan menjalankan operasi `simpanData()` yang berisi query untuk memasukkan data. Kemudian basis data akan mengirimkan result kepada sistem.
3. Sistem akan memberikan informasi bahwa data telah tersimpan.

- Diagram Sekuensial Mengubah Data *Account*

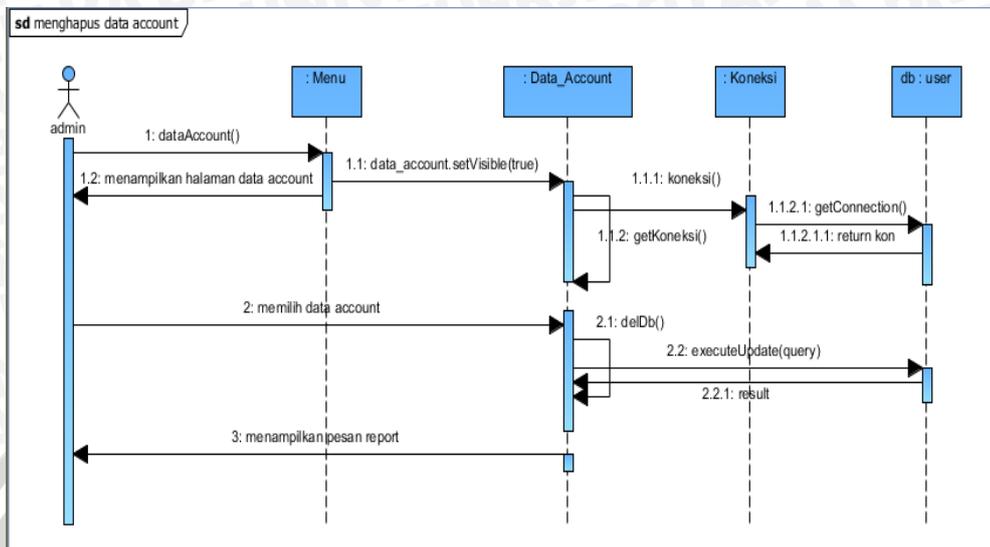


Gambar 4.11 Diagram Sekuensial Mengubah Data *Account*
Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.11 sebagai berikut:

1. Admin memilih halaman *Data Account*, kemudian sistem menampilkan halaman *Data Account*. Sistem melakukan koneksi dengan basis data melalui operasi `koneksi()`. Di dalam operasi `koneksi()` sistem memanggil operasi `getKoneksi()` dan `getConnection()` untuk melakukan koneksi dengan basis data.
2. Admin memilih data *account* yang akan diubah, kemudian sistem akan menjalankan operasi `getTable()` yang berisi perintah untuk menampilkan data yang dipilih pada form input data
3. Admin mengubah data *account*. Kemudian sistem akan menjalankan operasi `simpanData()` yang berisi perintah eksekusi query ke basis data.
4. Sistem akan menampilkan informasi bahwa data telah tersimpan

- Diagram Sekuensial Menghapus Data *Account*

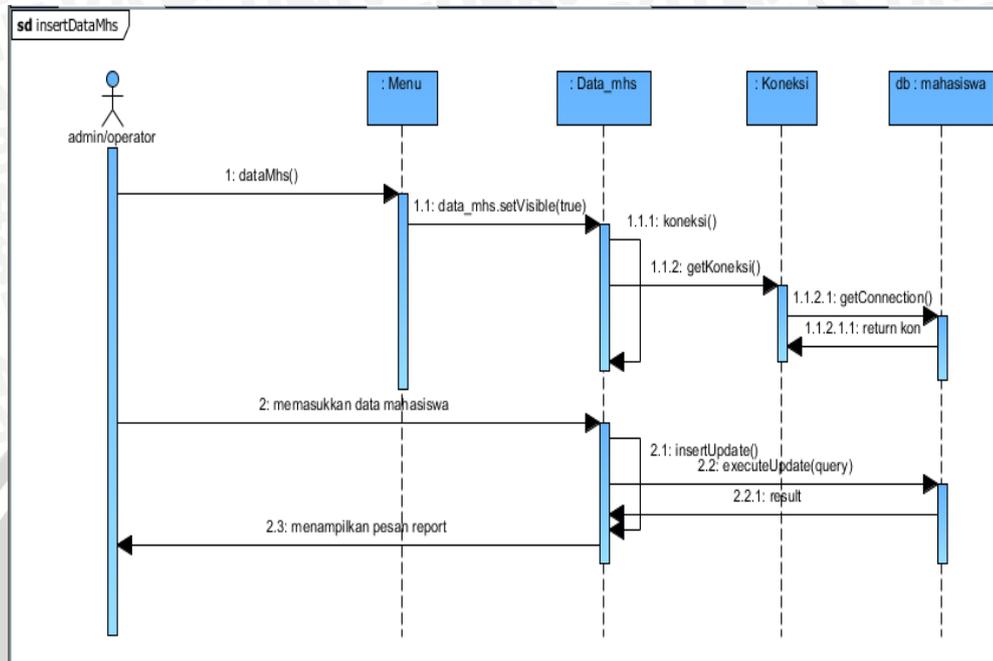


Gambar 4.12 Diagram Sekuensial Menghapus Data *Account*
Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.12 sebagai berikut:

1. Admin memilih halaman *Data Account*, kemudian sistem menampilkan halaman *Data Account*. Sistem melakukan koneksi dengan basis data melalui operasi koneksi(). Di dalam operasi koneksi() sistem memanggil operasi getKoneksi() dan getConnection() untuk melakukan koneksi dengan basis data.
2. Admin memilih data *account* yang akan dihapus, kemudian sistem akan menjalankan operasi delDb() yang berisi perintah query untuk menghapus data dari basis data
3. Sistem akan menampilkan informasi bahwa data telah terhapus

- Diagram Sekuensial Memasukkan Data Mahasiswa



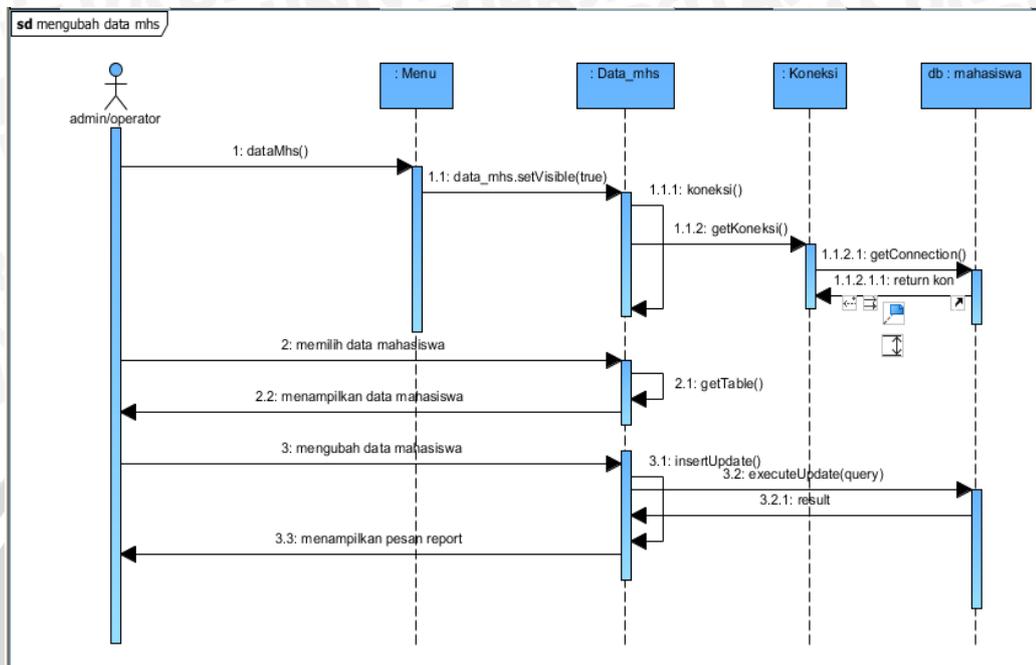
Gambar 4.13 Diagram Sekuensial Memasukkan Data Mahasiswa

Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.13 sebagai berikut:

1. Admin/operator memilih halaman Data Mahasiswa, kemudian sistem menampilkan halaman Data Mahasiswa. Sistem melakukan koneksi dengan basis data melalui operasi koneksi(). Di dalam operasi koneksi() sistem memanggil operasi getKoneksi() dan getConnection() untuk melakukan koneksi dengan basis data.
2. Admin/operator memasukkan data mahasiswa, kemudian sistem akan menjalankan operasi insertUpdate() yang berisi query untuk memasukkan data. Kemudian basis data akan mengirimkan result kepada sistem. Sistem akan memberikan informasi bahwa data telah tersimpan.

- Diagram Sekuensial Mengubah Data Mahasiswa



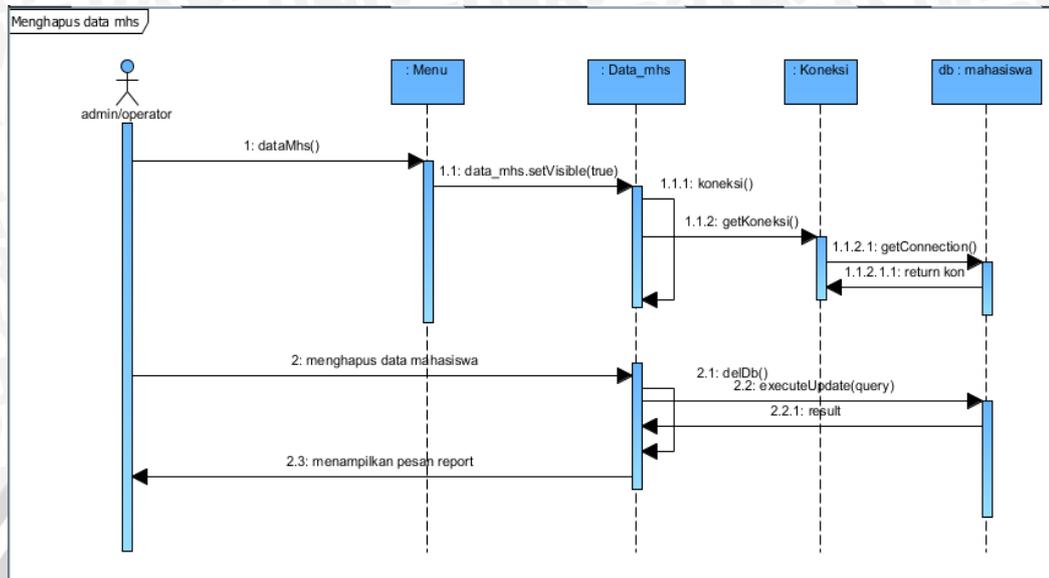
Gambar 4.14 Diagram Sekuensial Mengubah Data Mahasiswa

Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.14 sebagai berikut:

1. Admin/operator memilih halaman Data Mahasiswa, kemudian sistem menampilkan halaman Data Mahasiswa. Sistem melakukan koneksi dengan basis data melalui operasi koneksi(). Di dalam operasi koneksi() sistem memanggil operasi getKoneksi() dan getConnection() untuk melakukan koneksi dengan basis data.
2. Admin/operator memilih data mahasiswa yang akan diubah, kemudian sistem akan menjalankan operasi getTable() yang berisi perintah untuk menampilkan data yang dipilih pada form input data
3. Admin/operator mengubah data mahasiswa. Kemudian sistem akan menjalankan operasi insertUpdate() yang berisi perintah eksekusi query ke basis data. Kemudian sistem akan menampilkan informasi bahwa data telah tersimpan

- Diagram Sekuensial Menghapus Data Mahasiswa



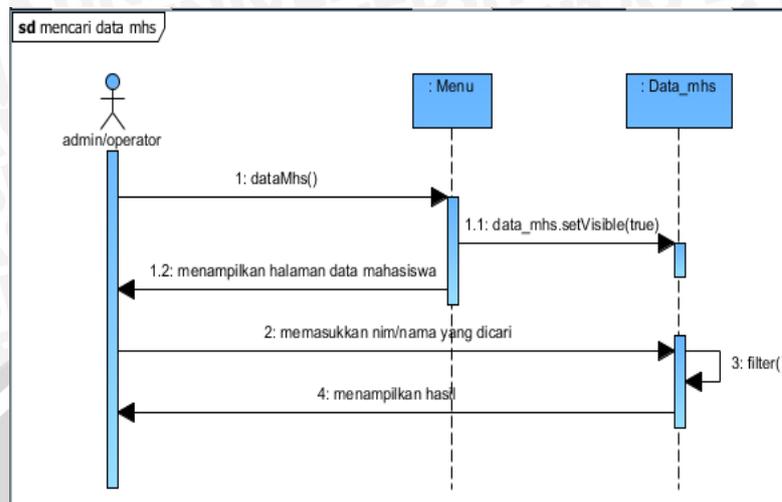
Gambar 4.15 Diagram Sekuensial Menghapus Data Mahasiswa

Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.15 sebagai berikut:

1. Admin/operator memilih halaman Data Mahasiswa, kemudian sistem menampilkan halaman Data Mahasiswa. Sistem melakukan koneksi dengan basis data melalui operasi koneksi(). Di dalam operasi koneksi() sistem memanggil operasi getKoneksi() dan getConnection() untuk melakukan koneksi dengan basis data.
2. Admin/operator memilih data mahasiswa yang akan dihapus, kemudian sistem akan menjalankan operasi delDb() yang berisi perintah query untuk menghapus data dari basis data. Kemudian sistem akan menampilkan informasi bahwa data telah terhapus

- Diagram Sekuensial Mencari Data Mahasiswa



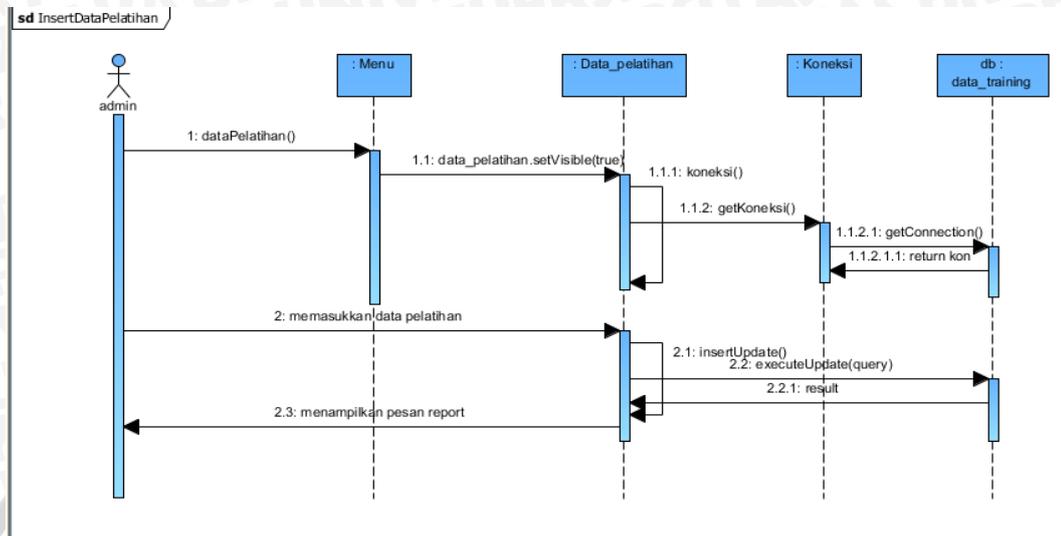
Gambar 4.16 Diagram Sekuensial Mencari Data Mahasiswa

Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.16 sebagai berikut:

1. Admin/operator memilih Data Mahasiswa, kemudian sistem akan menampilkan halaman Data Mahasiswa.
2. Admin/operator memasukkan nim atau nama yang dicari
3. Sistem akan melakukan proses pencarian dengan menggubakannoperasi `filter()`.
4. Sistem akan menampilkan hasil proses pencarian.

- Diagram Sekuensial Memasukkan Data Pelatihan

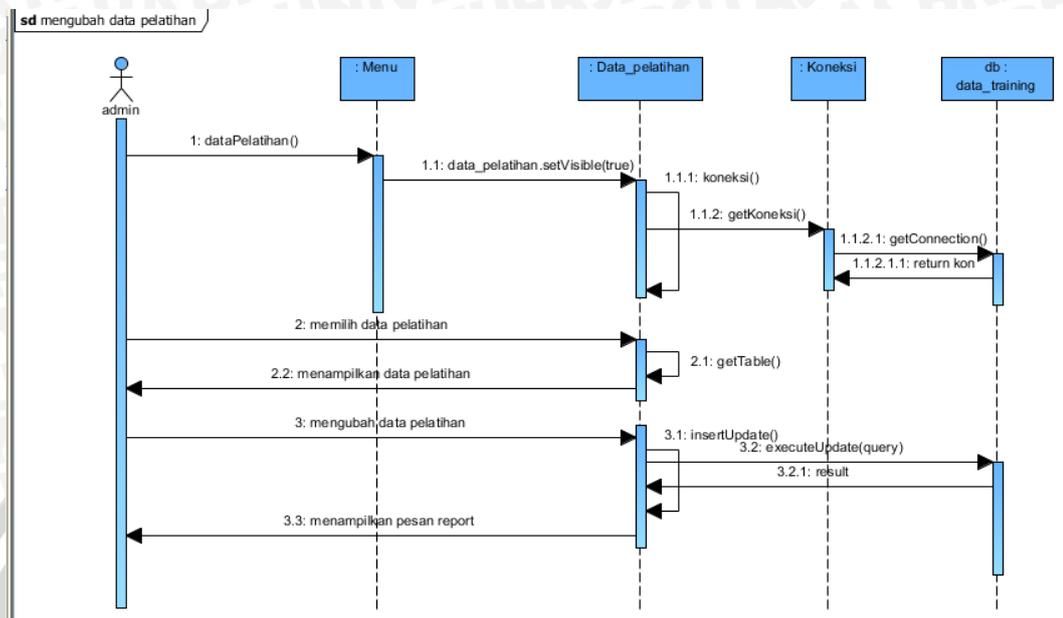


Gambar 4.17 Diagram Sekuensial Memasukkan Data Pelatihan
Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.17 sebagai berikut:

1. Admin memilih halaman Data Pelatihan, kemudian sistem menampilkan halaman Data Pelatihan. Sistem melakukan koneksi dengan basis data melalui operasi koneksi(). Di dalam operasi koneksi() sistem memanggil operasi getKoneksi() dan getConnection() untuk melakukan koneksi dengan basis data.
2. Admin memasukkan data pelatihan, kemudian sistem akan menjalankan operasi insertUpdate() yang berisi query untuk memasukkan data. Kemudian basis data akan mengirimkan result kepada sistem. Sistem akan memberikan informasi bahwa data telah tersimpan.

- Diagram Sekuensial Mengubah Data Pelatihan



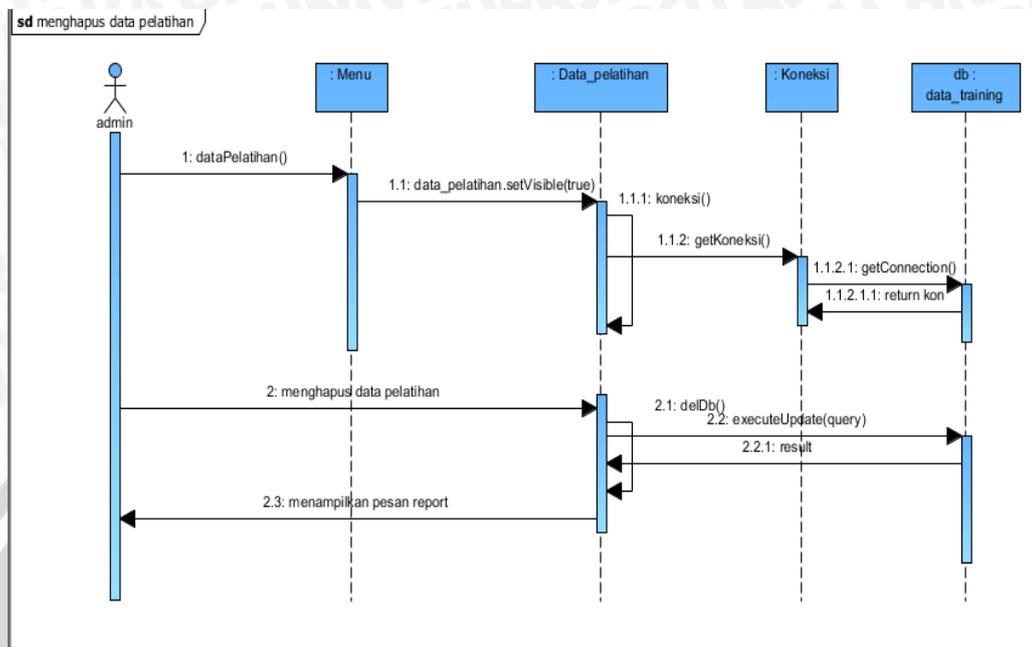
Gambar 4.18 Diagram Sekuensial Mengubah Data Pelatihan
Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.18 sebagai berikut:

1. Admin memilih halaman Data Pelatihan, kemudian sistem menampilkan halaman Data Pelatihan. Sistem melakukan koneksi dengan basis data melalui operasi koneksi(). Di dalam operasi koneksi() sistem memanggil operasi getKoneksi() dan getConnection() untuk melakukan koneksi dengan basis data.
2. Admin memilih data pelatihan yang akan diubah, kemudian sistem akan menjalankan operasi getTable() yang berisi perintah untuk menampilkan data yang dipilih pada form input data
3. Admin mengubah data pelatihan. Kemudian sistem akan menjalankan operasi insertUpdate() yang berisi perintah eksekusi query ke basis data. Kemudian sistem akan menampilkan informasi bahwa data telah tersimpan.



- Diagram Sekuensial Menghapus Data Pelatihan



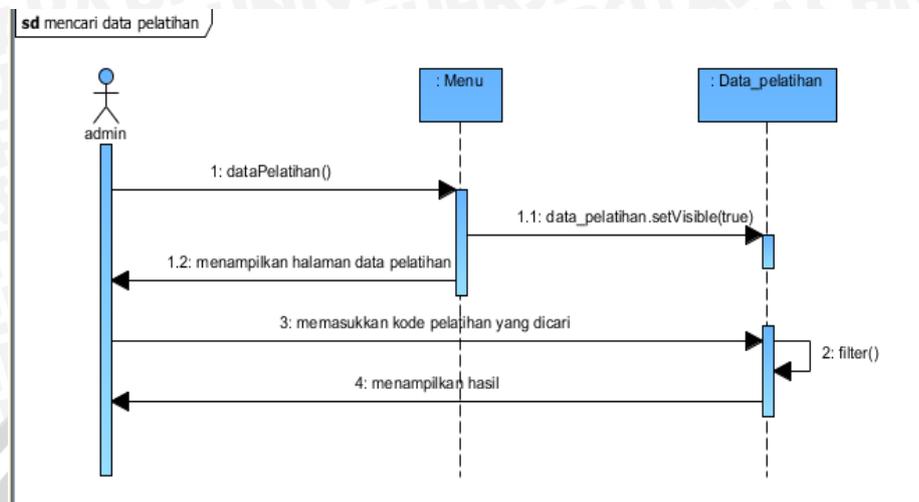
Gambar 4.19 Diagram Sekuensial Menghapus Data Pelatihan

Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.19 sebagai berikut:

1. Admin memilih halaman Data Pelatihan, kemudian sistem menampilkan halaman Data Pelatihan. Sistem melakukan koneksi dengan basis data melalui operasi koneksi(). Di dalam operasi koneksi() sistem memanggil operasi getKoneksi() dan getConnection() untuk melakukan koneksi dengan basis data.
2. Admin memilih data pelatihan yang akan dihapus, kemudian sistem akan menjalankan operasi delDb() yang berisi perintah query untuk menghapus data dari basis data. Kemudian sistem akan menampilkan informasi bahwa data telah terhapus.

- Diagram Sekuensial Mencari Data Pelatihan

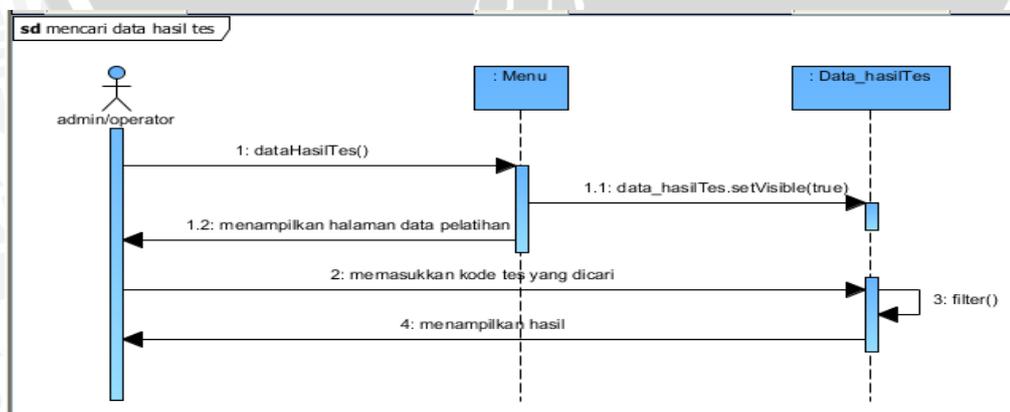


Gambar 4.20 Diagram Sekuensial Mencari Data Pelatihan
Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.20 sebagai berikut:

1. Admin memilih Data Pelatihan, kemudian sistem akan menampilkan halaman Data Pelatihan.
2. Admin memasukkan kode pelatihan yang dicari
3. Sistem akan melakukan proses pencarian dengan menggubakannoperasi filter(),
4. Sistem akan menampilkan hasil proses pencarian.

- Diagram Sekuensial Mencari Data Hasil Tes

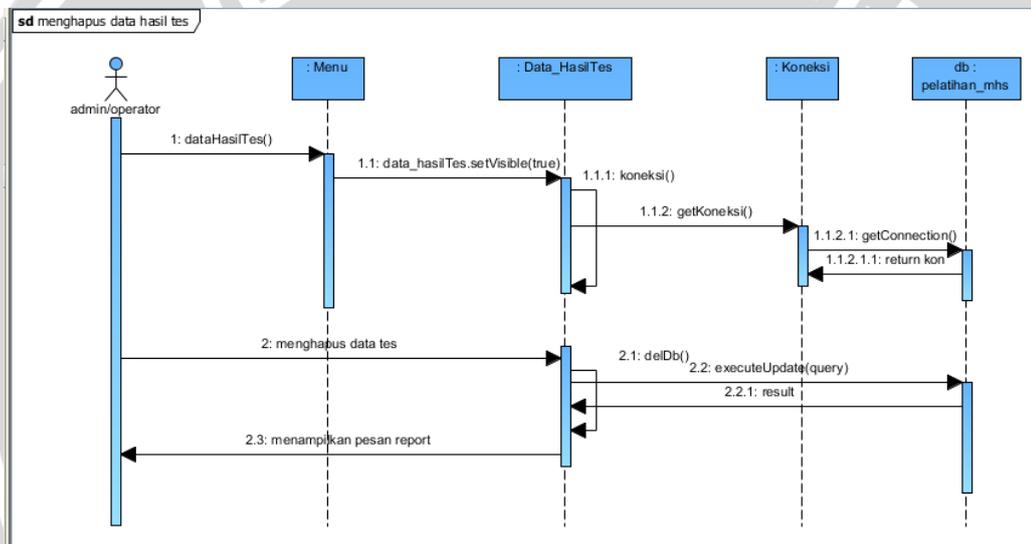


Gambar 4.21 Diagram Sekuensial Mencari Data Hasil Tes
Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.21 sebagai berikut:

1. Admin/operator memilih Data Hasil Tes, kemudian sistem akan menampilkan halaman Data Hasil Tes.
2. Admin/operator memasukkan kode tes yang dicari
3. Sistem akan melakukan proses pencarian dengan menggubakannoperasi `filter()`.
4. Sistem akan menampilkan hasil proses pencarian.

- Diagram Sekuensial Menghapus Data Hasil Tes



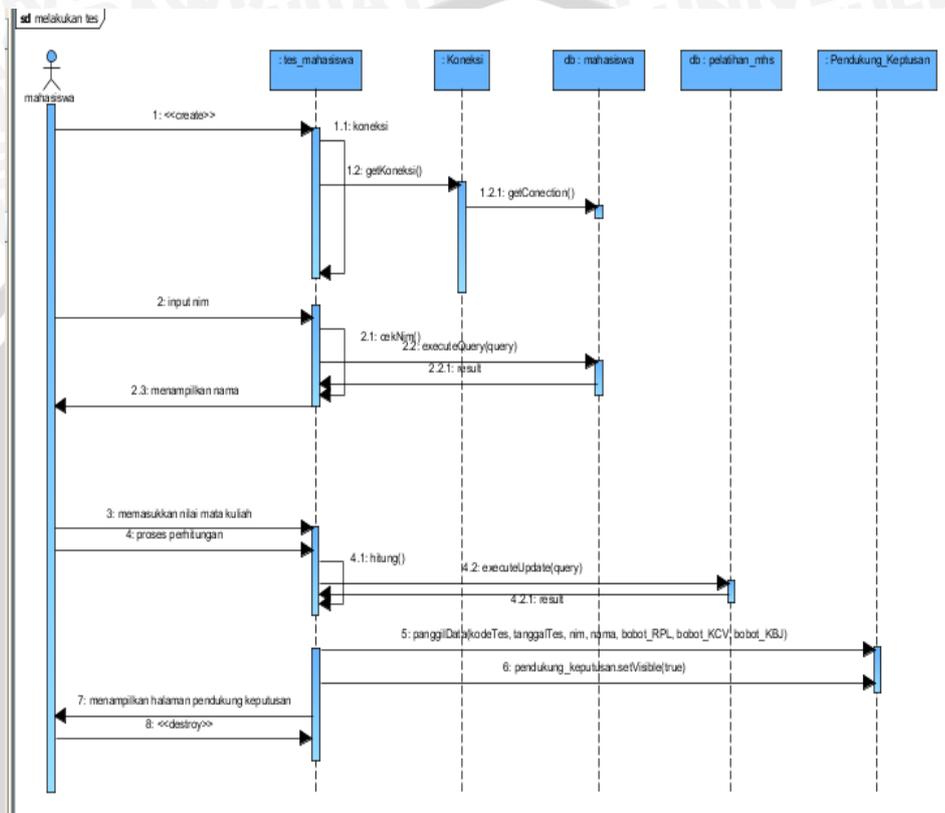
Gambar 4.22 Diagram Sekuensial Menghapus Data Hasil Tes
Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.22 sebagai berikut:

1. Admin/operator memilih halaman Data Hasil Tes, kemudian sistem menampilkan halaman Data Hasil Tes. Sistem melakukan koneksi dengan basis data melalui operasi `koneksi()`. Di dalam operasi `koneksi()` sistem memanggil operasi `getKoneksi()` dan `getConnection()` untuk melakukan koneksi dengan basis data.
2. Admin/operator memilih data hasil tes yang akan dihapus, kemudian sistem akan menjalankan operasi `delDb()` yang berisi

perintah query untuk menghapus data dari basis data. Kemudian sistem akan menampilkan informasi bahwa data telah terhapus.

- Diagram Sekuensial Melakukan Tes



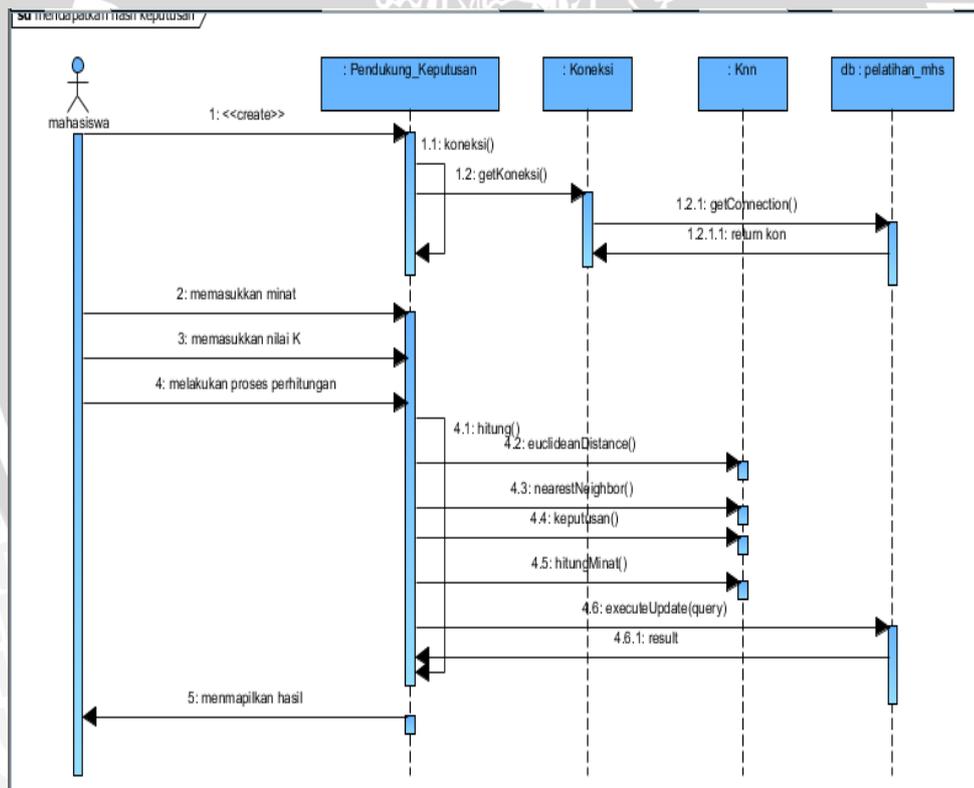
Gambar 4.23 Diagram Sekuensial Melakukan Tes
Sumber: Perancangan

Penjelasan diagram sekuensial dari Gambar 4.23 sebagai berikut:

1. Mahasiswa dihadapkan pada halaman Tes Mahasiswa, sistem akan melakukan koneksi ke basis data melalui kelas Koneksi dengan menggunakan operasi koneksi ().
2. Mahasiswa memasukkan nim, kemudian sistem akan memeriksa nim dengan menggunakan operasi cekNim() yang berisi perintah *query* untuk mengambil data nama mahasiswa dari basis data berdasarkan nim yang dimasukkan.
3. Mahasiswa memasukkan nilai mata kuliah pada form yang telah disediakan oleh sistem.

4. Mahasiswa memilih untuk melakukan proses perhitungan, kemudian sistem akan memanggil operasi `hitung()` yang berisi perintah untuk melakukan perhitungan jawaban dan menyimpan data hasil perhitungan sementara ke dalam basis data.
5. Kemudian sistem akan mengirimkan hasil perhitungan jawaban dengan memanggil operasi `panggilData()` yang ada pada kelas `PendukungKeputusan`.
6. Sistem akan memanggil kelas `PendukungKeputusan`.
7. Sistem menampilkan halaman `Pendukung Keputusan`
8. Sistem menutup halaman `Tes Mahasiswa`.

- Diagram Sekuensial Mendapatkan Hasil



Gambar 4.24 Diagram Sekuensial Mendapatkan Hasil

Sumber: Perancangan

Penjelasan diagram sekuensial Gambar 4.24 sebagai berikut:

1. Mahasiswa dihadapkan pada halaman Pendukung Keputusan, kemudian sistem akan melakukan koneksi ke basis data dengan menggunakan operasi koneksi ().
2. Mahasiswa memilih salah satu konsentrasi yang diminati.
3. Mahasiswa menentukan nilai K.
4. Mahasiswa memilih untuk melakukan proses penghitungan. Kemudian sistem akan melakukan proses perhitungan yang diawali dengan memanggil operasi yang ada pada kelas Knn yaitu `euclideanDistance()` untuk menghitung jarak *euclidean*, memanggil operasi `nearestNeighbor()` untuk menentukan tetangga terdekat, memanggil operasi `keputusan()` untuk menentukan keputusan dari sistem, memanggil operasi `hitungMinat()` untuk menghitung kecocokan hasil keputusan sistem dengan minat mahasiswa. Kemudian sistem akan menyimpan hasil dalam basis data tabel `pelatihan_mhs`.
5. Sistem akan menampilkan hasil keputusan.

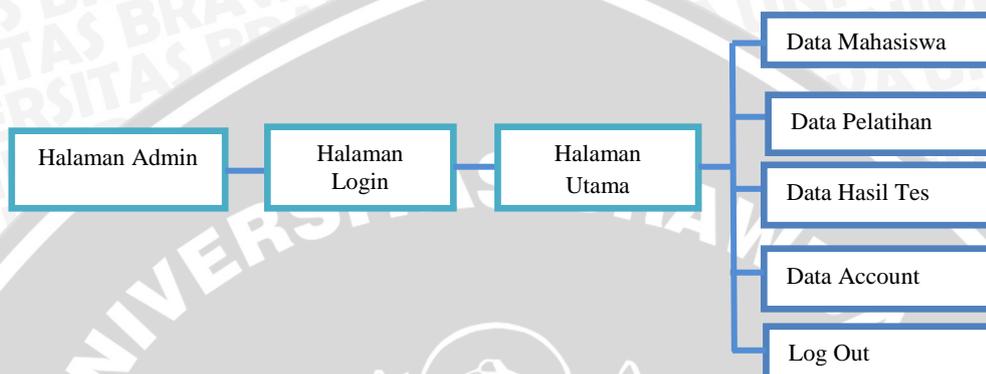
4.3 Perancangan Antar Muka

Perancangan antar muka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun. Sistem pendukung keputusan ini dibagi menjadi tiga halaman otoritas, yaitu halaman untuk admin, halaman untuk operator, dan halaman untuk mahasiswa. Halaman untuk admin dan operator terdiri atas halaman login, dan halaman utama. Halaman untuk mahasiswa terdiri atas halaman tes dan halaman pendukung keputusan. Berikut ini gambaran antarmuka yang ditunjukkan dengan *site map* halaman admin, operator dan halaman mahasiswa

4.3.1 Perancangan Antar Muka Halaman Admin dan Operator

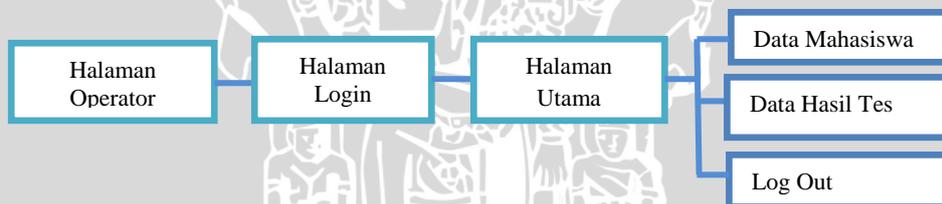
Halaman admin merupakan halaman yang disediakan sistem untuk admin. Sedangkan halaman operator merupakan halaman yang disediakan sistem untuk operator. Pada sistem ini, halaman admin dan operator terdiri atas halaman login

dan halaman utama. Halaman utama sendiri terdiri atas halaman Data Mahasiswa, halaman Data Pelatihan, halaman Hasil Tes, halaman Data *Account*, dan *Log out*. Halaman Data Pelatihan dan halaman Data *Account* akan di non-aktifkan bagi operator. *Site map* halaman admin ditunjukkan pada Gambar 4.25.



Gambar 4.25 *Site Map* Halaman Admin
Sumber: Perancangan

Site map halaman operator ditunjukkan pada Gambar 4.26.



Gambar 4.26 *Site Map* Halaman Operator
Sumber: Perancangan

1. Halaman *Login*

Pada halaman *Login*, admin atau operator dapat memulai *login* dengan memasukkan *username* dan *password*. Perancangan tampilan untuk halaman *Login* ditunjukkan pada Gambar 4.27.

SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KONSENTRASI
PS.TEKNIK INFORMATIKA UNIVERSITAS BRAWIJAYA

Form untuk masuk ke sistem:

Username: → 1

Password: → 2

MASUK → 3

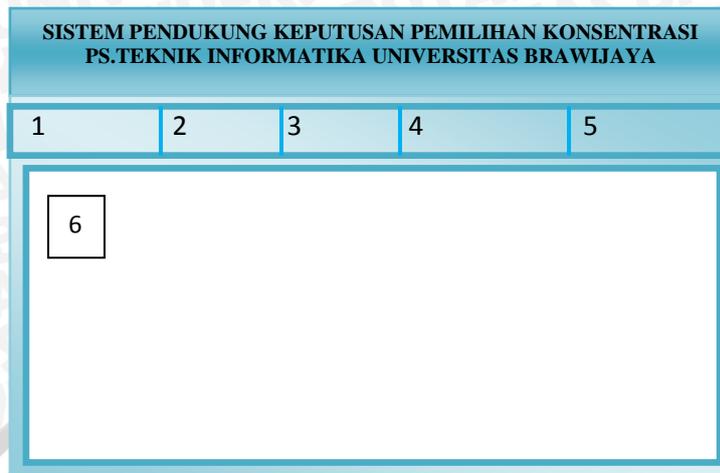
Gambar 4.27 Perancangan Tampilan Halaman *Login*
Sumber: Perancangan

Keterangan Gambar 4.27:

1. Untuk memasukkan *username*
2. Untuk memasukkan *password*
3. Tombol untuk *login*

2. Halaman Utama

Setelah berhasil melakukan login, admin atau operator dihadapkan pada halaman utama yang terdiri atas halaman Data Mahasiswa, halaman Data Pelatihan, halaman Data Hasil Tes, halaman Data *Account*, dan halaman *Log Out*. Halaman utama untuk operator hanya terdiri atas halaman Data Mahasiswa, halaman Data *Nilai*, dan halaman Data Hasil Tes. Perancangan tampilan untuk halaman utama ditunjukkan pada Gambar 4.28.



Gambar 4.28 Perancangan Tampilan Halaman Utama

Sumber: Perancangan

Keterangan Gambar 4.28:

1. Tombol untuk menuju halaman Data Mahasiswa
2. Tombol untuk menuju halaman Data Pelatihan
3. Tombol untuk menuju halaman Data Hasil Tes
4. Tombol untuk menuju halaman Data *Account*
5. Tombol untuk menuju melakukan *Logout*
6. Tempat untuk menampilkan halaman dari tombol yang dipilih

3. Halaman Data Mahasiswa

Halaman Data Mahasiswa akan ditampilkan ketika admin atau operator memilih tombol Data Mahasiswa. Pada halaman Data Mahasiswa, admin atau operator dihadapkan pada form pengisian data dan form untuk pencarian data. Admin atau operator dapat melakukan pengolahan data mahasiswa sebagaimana yang telah dijelaskan di *use case diagram* dan *sequence diagram*. Perancangan Tampilan dari halaman Data Mahasiswa ditunjukkan pada Gambar 4.29.

SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KONSENTRASI
PS.TEKNIK INFORMATIKA UNIVERSITAS BRAWIJAYA

DATA MAHASISWA

Form pengisian data:

Nim:

Nama:

Jenis Kelamin:

Angkatan:

SIMPAN

Form pencarian data:

Nim/nama:

EDIT **DELETE**

Gambar 4.29 Perancangan Tampilan Halaman Data Mahasiswa
Sumber: Perancangan

Keterangan Gambar 4.29:

1. Untuk memasukkan nim
2. Untuk memasukkan nama
3. Untuk memasukkan jenis kelamin
4. Untuk memasukkan angkatan
5. Tombol untuk menyimpan data
6. Untuk memasukkan nim atau yang akan dicari
7. Tabel untuk menampilkan data mahasiswa
8. Tombol untuk mengubah data mahasiswa
9. Tombol untuk menghapus data mahasiswa

4. Halaman Data Pelatihan

Halaman Data Pelatihan akan ditampilkan ketika admin memilih menu Data Pelatihan. Pada halaman data pelatihan, admin dihadapkan pada form pengisian data dan form untuk pencarian data. Admin dapat melakukan pengolahan data pelatihan sebagaimana yang telah dijelaskan di *use case*

diagram dan *sequence diagram*. Perancangan Tampilan halaman Data Pelatihan ditunjukkan pada Gambar 4.30.

The screenshot shows a web application interface for training data management. The title bar reads "SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KONSENTRASI" and "PS.TEKNIK INFORMATIKA UNIVERSITAS BRAWIJAYA". The main content area contains a form for entering training data, a search field for training codes, and buttons for saving, editing, and deleting data. Numbered callouts (1-6) point to specific UI elements:

- 1: Form untuk memasukkan data pelatihan:
- 2: SIMPAN button
- 3: Kode pelatihan: input field
- 4: Data display area
- 5: HAPUS button
- 6: EDIT button

Gambar 4.30 Perancangan Tampilan Halaman Data Pelatihan

Sumber: Perancangan

Keterangan Gambar 4.30:

1. Untuk memasukkan data pelatihan
2. Tombol untuk menyimpan data pelatihan yang dimasukkan
3. Untuk memasukkan kode pelatihan yang dicari
4. Untuk menampilkan data pelatihan yang telah dimasukkan
5. Tombol untuk menghapus data pelatihan yang akan dihapus
6. Tombol untuk mengubah data pelatihan yang akan diubah

5. Halaman Data Hasil Tes

Halaman Data Hasil Tes akan ditampilkan ketika admin atau operator memilih tombol Data Hasil Tes. Pada halaman Data Hasil Tes, admin atau operator dihadapkan pada form pencarian data. Admin atau operator dapat melakukan pengolahan data pelatihan sebagaimana yang telah dijelaskan di *use case diagram* dan *sequence diagram*. Perancangan tampilan dari halaman Data Hasil Tes ditunjukkan pada Gambar 4.37.

SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KONSENTRASI
PS.TEKNIK INFORMATIKA UNIVERSITAS BRAWIJAYA

Form pencarian data hasil tes mahasiswa:

Kode tes: → 1

--

HAPUS → 3

→ 2

Gambar 4.31 Perancangan Tampilan Halaman Data Hasil Tes
Sumber: Perancangan

Keterangan Gambar 4.31:

1. Untuk memasukkan kode tes yang akan dicari
2. Tabel untuk menampilkan data hasil tes
3. Tombol untuk menghapus data hasil tes yang dipilih

6. Halaman Data Account

Halaman Data Account akan ditampilkan ketika admin memilih tombol Data Account. Pada halaman Data Account, admin dihadapkan pada form perubahan password yang berisi username, password lama, dan password baru. Admin dapat melakukan perubahan password sebagaimana yang telah dijelaskan di use case diagram dan sequence diagram. Perancangan tampilan halaman Data Account ditunjukkan pada Gambar 4.32.



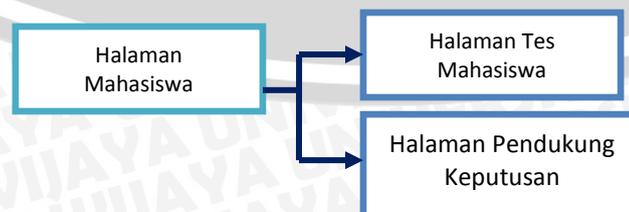
Gambar 4.32 Perancangan Tampilan Halaman Data Account
Sumber: Perancangan

Keterangan Gambar 4.32:

1. Form untuk memasukkan *accunt* baru
2. Tombol untuk memasukkan *account* baru
3. Tabel untuk menampilkan daftar *account*
4. Tombol untuk melakukan perubahan data *account*
5. Tombol untuk melakukan penghapusan data *account*

4.3.2 Perancangan Antar Muka Halaman Mahasiswa

Halaman mahasiswa merupakan halaman yang disediakan untuk mahasiswa yang digunakan untuk melakukan tes dalam mendukung keputusan. Halaman mahasiswa terdiri atas halaman tes mahasiswa dan halaman pendukung keputusan. *Site map* menu mahasiswa ditunjukkan pada Gambar 4.33.



Gambar 4.33 Site Map Halaman Mahasiswa
Sumber: Perancangan

1. Halaman Tes Mahasiswa

Halaman Tes Mahasiswa merupakan halaman yang digunakan mahasiswa untuk memasukkan nilai mata kuliah. Pada halaman tes mahasiswa terdapat form untuk memasukkan data diri dan form memasukkan data nilai mata kuliah.. Perancangan tampilan halaman Tes Mahasiswa

The image shows a web interface for a student test. At the top, it says "SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KONSENTRASI" and "PS.TEKNIK INFORMATIKA UNIVERSITAS BRAWIJAYA". Below this is a form for personal data with fields for "Kode Tes:", "Nim:", and "Nama:". A red arrow labeled "1" points to this section. Below that is a section titled "Masukkan nilai mata kuliah anda:" with a list of subjects and input boxes: "1. Analisi dan Perancangan Sistem:", "2. Grafika Komputer:", "3. Pemrograman Berorientasi Objek:", "4. Rekayasa Perangkat Lunak:", "5. ..", "6. ...", and "7. ...". A red arrow labeled "2" points to this section. At the bottom right is a button labeled "PROSES" with a red arrow labeled "3" pointing to it.

Gambar 4.34 Perancangan Tampilan Halaman Tes Mahasiswa
Sumber: Perancangan

Keterangan Gambar 4.34:

1. Form Untuk memasukkan data diri
2. Form Untuk memasukkan nilai mata kuliah
3. Tombol untuk melakukan proses perhitungan

2. Halaman Pendukung Keputusan

Halaman Pendukung Keputusan merupakan menu yang disediakan sistem untuk mahasiswa dalam mendukung keputusan. Pada halaman Pendukung Keputusan terdapat form untuk menampilkan data yang telah dimasukkan pada halaman Tes Mahasiswa. Perancangan tampilan halaman Pendukung Keputusan ditunjukkan pada Gambar 4.35.

SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KONSENTRASI
PS.TEKNIK INFORMATIKA UNIVERSITAS BRAWIJAYA

Kode Tes: Nim:
Nama:

1. Analisis dan Perancangan Sistem:
2. Grafika Komputer:
3. PBO:
4. Rekayasa Perangkat Lunak:
5.
6.

K-tetangga:
Minat:

PROSES
KEMBALI

Gambar 4.35 Perancangan Tampilan Halaman Pendukung Keputusan
Sumber: Perancangan

Keterangan Gambar 4.35:

1. Form untuk menampilkan data diri
2. Untuk menampilkan hasil perhitungan sistem
3. Untuk memasukkan jumlah tetanga terdekat
4. Untuk memasukkan minat

5. Untuk menampilkan nilai mata kuliah yang telah dimasukkan
6. Tombol untuk melakukan proses perhitungan
7. Tombol untuk kembali ke halaman sebelumnya

4.4 Perancangan Algoritma

Sistem pendukung keputusan ini memiliki beberapa proses perancangan algoritma, antara lain algoritma *login*, perancangan algoritma memasukkan data (data mahasiswa, data pelatihan, dan data *account*), perancangan algoritma mengubah data (data mahasiswa, data pelatihan, dan data *account*), perancangan algoritma menghapus data (data mahasiswa, data pelatihan, dan data *account*), perancangan algoritma mencari data (data mahasiswa, data pelatihan, data hasil tes dan data *account*), perancangan algoritma *K-Nearest Neighbor*. Pada penulisan penelitian ini hanya dicantumkan perancangan algoritma untuk algoritma dari beberapa proses (operasi) saja (tidak untuk keseluruhan operasi).

4.4.1 Perancangan Algoritma Proses *K-Nearest Neighbor*

Proses perhitungan *K-Nearest Neighbor* terdiri atas dua proses yaitu perhitungan jarak *euclidean* dan pemilihan tetangga terdekat. Proses perhitungan jarak *euclidean* digunakan untuk menghitung jarak antara data tes dengan data *training*. Algoritma yang diterapkan dari jarak *euclidean* ditunjukkan pada Gambar 4.36.

Nama algoritma: euclidean distance

Deklarasi:

- Integer: *i*
- Double: *d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, d12, aps, grafkom, pbo, rpl, sismik, pl, kb, bdl, pcd, pp, probstat, jarkom, distance*

Deskripsi:

- Input: *aps, grafkom, pbo, rpl, sismik, pl, kb, bdl, pcd, pp, probstat, jarkom, distance*
- Proses:
 1. Melakukan pengecekan data training yang ada pada database
 2. $i=0$
 3. selama data training ada maka
 4. $d1=$ nilai dari mata kuliah *aps*
 5. $d2=$ nilai dari mata kuliah *grafkom*

7. d3= nilai dari mata kuliah pbo
8. d4= nilai dari mata kuliah rpl
9. d5= nilai dari mata kuliah sismik
10. d6= nilai dari mata kuliah pl
11. d7= nilai dari mata kuliah kb
12. d8= nilai dari mata kuliah bdl
13. d9= nilai dari mata kuliah pcd
14. d10= nilai dari mata kuliah pp
15. d11= nilai dari mata kuliah probstat
16. d12= nilai dari mata kuliah jarkom
17. Menghitung jarak antara data training dengan data tes:


```
double distance = Math.sqrt(Math.pow((d1 - aps), 2) +
        Math.pow((d2 - grafkom), 2) + Math.pow((d3 - pbo), 2) +
        Math.pow((d4 - rpl), 2)+ Math.pow((d5 - sismik), 2) +
        Math.pow((d6 - pl), 2)+ Math.pow((d7 - kb), 2) +
        Math.pow((d8 - bdl), 2) + Math.pow((d9 - pcd), 2)+
        Math.pow((d10 - pp), 2) + Math.pow((d11 - probstat), 2)
        + Math.pow((d12 - jarkom), 2) );
```
18. Tambahkan nilai $i=i+1$, kembali ke langkah 3
19. Simpan semua distance ke dalam arraylist
 - Output: tampilkan semua nilai distance

Gambar 4.36 Perancangan Algoritma Perhitungan Jarak *Euclidean*

Sumber: Perancangan

Perancangan selanjutnya dari proses perhitungan *K-Nearest Neighbor* yaitu pemilihan tetangga terdekat. Pemilihan tetangga terdekat dilakukan berdasarkan jarak *euclidean* yang telah diperoleh sebelumnya. Algoritma yang diterapkan dari pemilihan tetangga terdekat adalah sebagai berikut:

Nama algoritma: nearest neighbor
Deklarasi:

- Integer: i, k, x
- ArrayList: voting

Deskripsi:

- Input: k
- Proses:
 1. distance yang disimpan dalam arrayList diurutkan berdasarkan jarak yang terkecil sampai yang terbesar
 2. Untuk $i=0$ sampai $i<k$ lakukan:
 3. Split data distance menjadi dua yaitu $split[0]=$ jarak dan $split[1]=$ konsentrasi
 4. Simpan data distance $split[1]$ dalam arrayList voting
 5. $x= i+1$
 6. $i=i+1$ kembali ke proses 2
- Output: tampilkan tetangga(x), jarak(split[0]), dan konsentrasi (split[1])

Gambar 4.37 Perancangan Algoritma Pemilihan Tetangga Terdekat

Sumber: Perancangan

4.4.2 Perancangan Algoritma Pengambilan Keputusan

Proses pengambilan keputusan sistem didasarkan pada mayoritas kelas dari “k” yang telah dipilih. Jika dari nilai “k” yang dipilih mayoritas masuk ke kelas “RPL”, maka keputusan sistem adalah “RPL”. Perancangan algoritma keputusan ditunjukkan pada Gambar 4.38.

<p>Nama Algoritma: keputusan</p> <p><u>Deklarasi:</u></p> <ul style="list-style-type: none"> • integer: rpl, kcv, jaringan, i • String: hasil, keputusan • ArrayList: voting, list <p><u>Deskripsi:</u></p> <ul style="list-style-type: none"> • Input: data distance yang telah disimpan dalam arrayList voting sebelumnya • Proses: <ol style="list-style-type: none"> 1. Set nilai awal rpl=0 kcv=0, dan jaringan=0 2. Untuk i=0 sampai i= jumlah data yang ada dalam arrayList voting lakukan: 3. Jika data voting == "RPL" maka 4. rpl=rpl+1, sebaliknya 5. jika data voting == "KCV" maka 6. kcv=kcv+1, sebaliknya 7. rpl=rpl+1 8. i=i+1, kembali ke langkah 2 9. simpan dalam arrayList list 10. keputusan = data terbanyak dari list yang disimpan 11. hasil=keputusan • Output: tampilkan hasil keputusan
--

Gambar 4.38 Perancangan Algoritma Pengambilan Keputusan
Sumber: Perancangan

4.4.3 Perancangan Algoritma Perhitungan Kecocokan Minat

Proses perhitungan kecocokan minat dilakukan dengan membandingkan minat yang dipilih oleh user dengan hasil analisis sistem yang terdiri atas hasil perhitungan jarak terdekat sebanyak “k” yang telah ditentukan sebelumnya. Algoritma yang diterapkan untuk menghitung prosentase kecocokan ditunjukkan pada Gambar 4.39.

Nama Algoritma: hitung keminatan

Deklarasi:

- String: minat
- Double: hasil

Deskripsi:

- Input: minat
- Proses:
 1. Mengecek dan mengambil data minat dari item comboBox yang dipilih
 2. hasil=0
 3. jika minat=="RPL" maka
 4. hasil= (rpl/k)*100, sebaliknya
 5. jika minat=="KCV" maka
 6. hasil= (kcv/k)*100, sebaliknya
 7. hasil= (jaringan/k)*100
- Ouput: tampilkan hasil

Gambar 4.39 Perancangan Algoritma Perhitungan Kecocokan Minat

Sumber: Perancangan

