

BAB V

IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari analisis kebutuhan dan proses perancangan perangkat lunak yang dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi tiap klas pada file program, dan implementasi algoritma.

5.1 Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada Bab 4 menjadi acuan untuk melakukan implementasi menjadi sistem yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Pengembangan sistem pendukung keputusan pemilihan konsentrasi di Teknik Informatika Universitas Brawijaya menggunakan sebuah komputer dengan spesifikasi perangkat keras yang dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
Prosesor	Intel (R) Pentium(R) M processor 1,60 GHz
Memori(RAM)	504MB
Hardisk	Fujitsu MHW208DAT
Kartu Grafis	Mobile Intel(R) 915GM/GMS,910 GML Express Chipset Family

Sumber: Implementasi

5.1.2 Spesifikasi Perangkat Lunak

Pengembangan sistem pendukung keputusan pemilihan konsentrasi di Teknik Informatika Universitas Brawijaya menggunakan perangkat lunak dengan spesifikasi yang dijelaskan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak Komputer

Sistem Operasi	Microsoft Windows Professional version 2002 Service Pack 2
Bahasa Pemrograman	Java
Tools pemrograman	JDK 1.6.0_02
Lingkungan pemrograman	Java(TM) 2 Runtime Environment, Standard Edition (build 1.6.0_06) , Apache MINA Framework 2.00-M3
IDE (<i>Integrated Development Environment</i>)	Netbeans 6.9.1

Sumber: Implementasi

5.2 Batasan-Batasan Implementasi

Beberapa batasan dalam mengimplementasikan sistem adalah sebagai berikut:

- *Input* yang diterima oleh sistem berupa nilai mata kuliah dari mahasiswa.
- *Output* yang diterima *user* berupa analisis sistem dan prosentase kecocokan dengan salah satu konsentrasi.
- IDE (*Integrated Development Environment*) yang digunakan adalah Netbeans 6.9.
- Platform pengembangan yang digunakan adalah JSE 6 (*Java Standard Edition 6*).
- *Database* akan disimpan dalam My SQL.
- Metode yang digunakan yaitu *K- Nearest Neighbor*.
- *Data training* yang digunakan adalah nilai mata kuliah prasyarat masing-masing konsentrasi dari mahasiswa 2007-2008 yang telah menempuh konsentrasi yang dipilih.
- Konsentrasi yang ditentukan ada tiga, yaitu Rekayasa Perangkat Lunak (RPL), Komputasi Cerdas dan Visualisasi (KCV), dan Komputasi Berbasis Jaringan (KBJ).
- Sistem ini terbatas hanya untuk mahasiswa yang akan melakukan penjurusan dan telah menempuh mata kuliah prasyarat dari konsentrasi RPL, KCV, dan KBJ.

- Sistem ini bersifat dinamis, yaitu menyediakan fasilitas *insert* dan *update*.

5.3 Implementasi Kelas pada File Program

Setiap kelas yang telah dirancang pada proses perancangan direalisasikan pada sebuah file program *.java. Tabel 5.3 menjelaskan mengenai pasangan antara kelas dengan file program yang digunakan untuk mengimplementasikannya.

Tabel 5.3 Implementasi kelas pada kode program *.java

No	Paket	Nama Kelas Manager	Nama File Program Manager .java
1.	Koneksi	Koneksi	Koneksi.java
2.		Koneksi1	Koneksi1.java
3.	Spk	Knn	Knn.java
4.	User	PendukungKeputusan	PendukungKeputusan.java
5.		TesMhs	TesMhs.java
6.	user1	Data_hasiltes	Data_hasiltes.java
7.		Data_mahasiswa	Data_mahasiswa.java
8.		Data_pelatihan	Data_pelatihan.java
9.		User	User.java
10.		Menu	Menu.java
11.		Data_Account	Data_Account.java

Sumber: Implementasi

5.4 Implementasi Algoritma

Sistem Pendukung Keputusan ini mempunyai beberapa proses utama, yaitu proses login, proses memasukkan data, proses perhitungan jarak terdekat dengan *K-Nearest Neighbor*, dan proses perhitungan kecocokan.

5.4.1 Implementasi Algoritma Perhitungan Jarak *Euclidean*

Proses perhitungan jarak *euclidean* dilakukan dengan menghitung jarak antara data pelatihan dengan data tes. Gambar 5.1 merupakan implementasi dari algoritma Perhitungan Jarak *Euclidean*.

```

1. public void euclideanDistance(double aps, double grafkom,
2. double pbo, double rpl, double sismik, double pl, double
3. kb, double bdl, double pcd, double pp, double probstat,
4. double jarkom) throws SQLException {
5.     ResultSet rs = koneksi.executeSelect("select * from
6.     data_training");
7.     int i = 0;
8.     while (rs.next()) {
9.         double d1 = Double.valueOf(rs.getString("aps"));
10.        double d2 = Double.valueOf(rs.getString("grafkom"));
11.        double d3 = Double.valueOf(rs.getString("pbo"));
12.        double d4 = Double.valueOf(rs.getString("rpl"));
13.        double d5 = Double.valueOf(rs.getString("sismik"));
14.        double d6 = Double.valueOf(rs.getString("pl"));
15.        double d7 = Double.valueOf(rs.getString("kb"));
16.        double d8 = Double.valueOf(rs.getString("bdl"));
17.        double d9 = Double.valueOf(rs.getString("pcd"));
18.        double d10 = Double.valueOf(rs.getString("pp"));
19.        double d11 = Double.valueOf(rs.getString("probstat"));
20.        double d12 = Double.valueOf(rs.getString("jarkom"));
21.        double distance = Math.sqrt(Math.pow((d1 - aps), 2) +
22.        Math.pow((d2 - grafkom), 2)+ Math.pow((d3 - pbo), 2) +
23.        Math.pow((d4 - rpl), 2) + Math.pow((d5 - sismik), 2) +
24.        Math.pow((d6 - pl), 2)+ Math.pow((d7 - kb), 2) +
25.        Math.pow((d8 - bdl), 2) + Math.pow((d9 - pcd), 2) +
26.        Math.pow((d10 - pp), 2) + Math.pow((d11 - probstat),
27.        2)+ Math.pow((d12 - jarkom),2));
28.        System.out.println(distance);
29.        arrayList.add(i, distance);
30.        konsentrasi.add(i,rs.getString("konsentrasi"));
31.        i++;
32.    }
33. }

```

Gambar 5.1 Implementasi Algoritma Perhitungan Jarak *Euclidean*
Sumber: Implementasi

Penjelasan algoritma Proses Perhitungan Jarak *Euclidean* untuk operasi `euclideanDistance()` pada Gambar 5.3 yaitu:

1. Baris 1-7 merupakan pendeklarasian variabel yang dibutuhkan pada operasi `euclideanDistance()`
2. Baris 5-6 merupakan perintah pengeksekusian query untuk mengambil data pelatihan yang ada dalam basis data
3. Baris 7 merupakan inisialisasi nilai `i=0` untuk iterasi pertama, dimana `i` merupakan variabel yang menunjukkan data ke-

4. Baris 8 merupakan operasi perulangan yang dilakukan jika data pelatihan ada dalam basis data
5. Baris 9-20 merupakan perintah untuk menyimpan nilai mata kuliah yang ada dalam basis data ke dalam variabel baru yang bertipe double.
6. Baris 21-27 merupakan perintah untuk menghitung jarak *Euclidean* antara data pelatihan dengan data tes.
7. Baris 28 merupakan perintah yang digunakan untuk menampilkan hasil perhitungan jarak *Euclidean*
8. Baris 29-30 merupakan perintah untuk menyimpan hasil perhitungan jarak *Euclidean* kedalam ArrayList
9. Baris 31 merupakan perintah untuk menambahkan nilai iterasi $i=i+1$

5.4.2 Implementasi Algoritma Pemilihan Tetangga Terdekat

Setelah jarak euclidean dari semua data pelatihan diperoleh, maka selanjutnya dilakukan proses pemilihan tetangga terdekat. Gambar 5.2 merupakan implementasi dari algoritma Pemilihan Tetangga Terdekat.

```
1. public void nearestNeighbor() {
2.     arrayList = insertSort(arrayList);
3.     for (int i = 0; i < k; i++) {
4.         voting.add(i, (String)konsentrasi.get(i));
5.         int x = i+1;
6.         fp.getOutputStream().append("\n Tetangga "+x+" \n jarak
7. = "+arrayList.get(i)+" \n konsentrasi = "+konsentrasi.get(i));
8.     }
9.     String hasil = "\nhasil \t= MAX {";
10.    for (int i = 0; i < voting.size(); i++) {
11.        if (i == voting.size() - 1) {
12.            hasil += voting.get(i);
13.        }
14.        else {
15.            hasil += voting.get(i) + ", ";
16.        }
17.    }
18.    hasil += "};";
19.    System.out.println(hasil);
20.    fp.getOutputStream().append(hasil);
21. }
```

Gambar 5.2 Implementasi Algoritma Pemilihan Tetangga Terdekat
Sumber: Implementasi

Penjelasan algoritma Pemilihan Tetangga Terdekat untuk operasi `nearestNeighbor()` pada Gambar 5.2 yaitu:

1. Baris 2 merupakan perintah untuk mengurutkan jarak *euclidean* dari data pelatihan yang disimpan dalam `ArrayList arrayList`. Pengurutan dilakukan mulai dari jarak *euclidean* terkecil hingga terbesar.
2. Baris 3-9 merupakan operasi perulangan untuk memilih tetangga terdekat berdasarkan nilai *k* yang telah dimasukkan *user*. Nilai *k* merupakan bilangan yang dimasukkan *user* dalam menentukan jumlah tetangga terdekat.
3. Baris 4 merupakan perintah untuk menyimpan konsentrasi dari distance yang telah diurutkan ke dalam `ArrayList voting`.
4. Baris 5 merupakan perintah untuk memberikan nilai variabel $x=i+1$, dimana *x* merupakan variabel yang menunjukkan tetangga ke-.
5. Baris 6-7 merupakan perintah untuk menampilkan tetangga=*x* dengan jarak *euclidean* = `arrayList`, dan konsentrasi = `konsentrasi`.

- Baris 10-21 merupakan perintah untuk menampilkan konsentrasi-konsentrasi dari jumlah k yang dipilih.

5.4.3 Implementasi Algoritma Pengambilan Keputusan

Setelah diperoleh beberapa tetangga terdekat, maka selanjutnya dilakukan proses pengambilan keputusan. Pengambilan keputusan sistem didasarkan pada mayoritas konsentrasi yang ada dalam k . Gambar 5.3 merupakan implementasi dari algoritma Pengambilan Keputusan.

```

1. public String keputusan() {
2.     rpl = 0;
3.     kcv = 0;
4.     jaringan = 0;
5.
6.     for (int i = 0; i < voting.size(); i++) {
7.         if (voting.get(i).toString().equals("RPL")) {
8.             rpl++;
9.         }
10.        if (voting.get(i).toString().equals("KCV")) {
11.            kcv++;
12.        }
13.        if(voting.get(i).toString().equals("KBJ")) {
14.            jaringan++;
15.        }
16.    }
17.    ArrayList list = new ArrayList();
18.    list.add(rpl + "; RPL");
19.    list.add(kcv + "; KCV");
20.    list.add(jaringan + "; KBJ");
21.    String keputusan[] =
22.    String.valueOf(Collections.max(list)).split(";");
23.    hasil = keputusan[1];
24.    max = Integer.parseInt(keputusan[0]);
25.    return "\nkeputusan: " + keputusan[1];
26.    }

```

Gambar 5.3 Implementasi Algoritma Pengambilan Keputusan

Sumber: Implementasi

Penjelasan algoritma Pengambilan Keputusan untuk operasi keputusan() pada Gambar 5.3 yaitu:

- Baris 1-4 merupakan pendeklarasian variabel yang dibutuhkan dalam operasi keputusan().

2. Baris 6-16 merupakan operasi perulangan yang dilakukan untuk menghitung jumlah konsentrasi yang ada dalam array list voting.
3. Baris 7-9 merupakan operasi seleksi apakah data yang ada dalam array list voting adalah konsentrasi "RPL". Jika benar maka akan dijalankan baris 8.
4. Baris 10-12 merupakan operasi seleksi apakah data yang ada dalam array list voting adalah konsentrasi "KCV". Jika benar maka akan dijalankan baris 11.
5. Baris 13-15 merupakan operasi seleksi apakah data yang ada dalam array list voting adalah konsentrasi "KBJ". Jika benar maka akan dijalankan baris 14.
6. Baris 17 merupakan perintah membuat objek array list baru yang bernama list.
7. Baris 18-20 merupakan perintah untuk menyimpan hasil penyeleksian ke dalam list.
8. Baris 21-25 merupakan perintah untuk mengambil nilai terbanyak dari data yang ada pada list kemudian dilakukan pemisahan data yaitu `split[0]= jumlah konsentrasi` dan `split[1]= konsentrasi`.
9. Baris ke 26 perintah untuk menampilkan hasil keputusan

5.4.4 Implementasi Algoritma Perhitungan Kecocokan Minat

Setelah diperoleh hasil analisis keputusan dari sistem, maka diperlukan suatu operasi yang digunakan untuk mengukur tingkat kecocokan antara minat mahasiswa dengan hasil analisis sistem. Gambar 5.4 merupakan implementasi algoritma Perhitungan Kecocokan Minat.

```
1. public double HitungMinat(String minat) {
2.     double hasil=0;
3.     if (minat.equals("RPL")) {
4.         hasil = Double.valueOf(rpl)/Double.valueOf(k)*100;
5.     }
6.     if (minat.equals("KCV")) {
7.         hasil = Double.valueOf(kcv)/Double.valueOf(k)*100;
8.     }
9.     if (minat.equals("KBJ")) {
10.        hasil =
11.            Double.valueOf(jaringan)/Double.valueOf(k)*100;
12.    }
13.    System.out.println(hasil);
14.    return hasil;
15. }
```

Gambar 5.4 Implementasi Algoritma Perhitungan Kecocokan Minat
Sumber: Implementasi

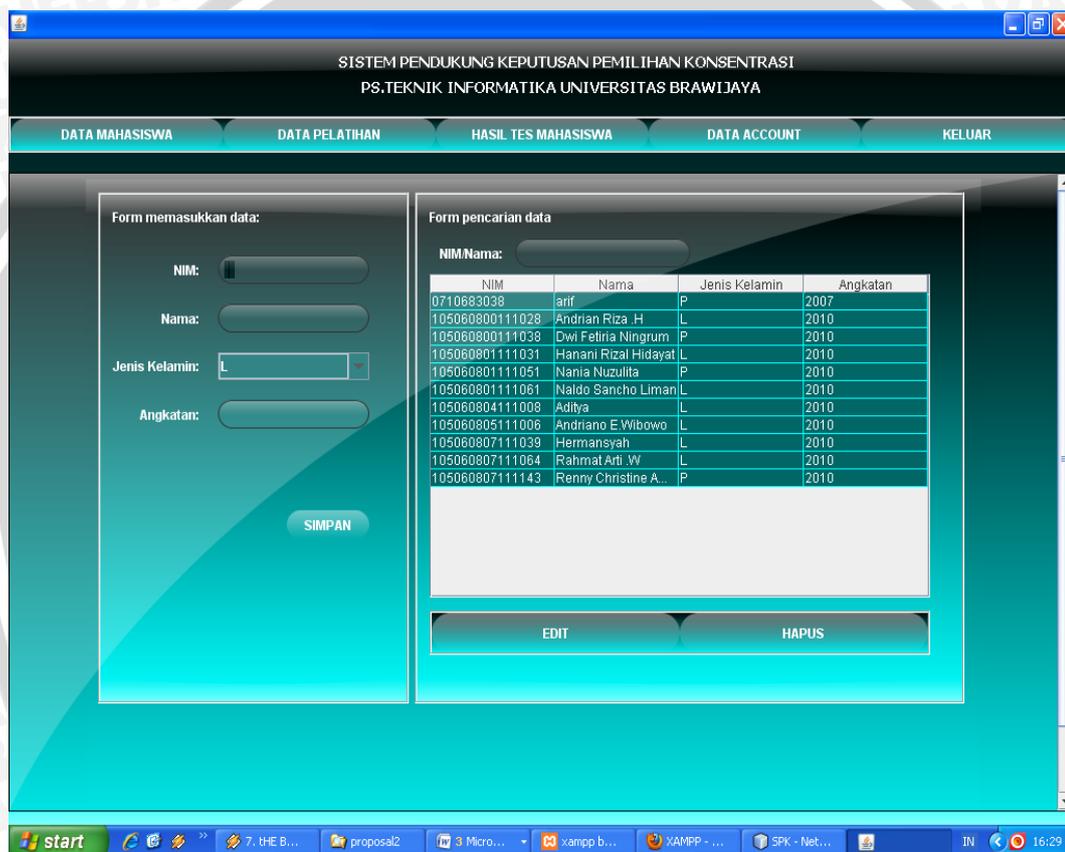
Penjelasan algoritma Perhitungan Kecocokan Minat untuk operasi HitungMinat() pada Gambar 5.4 yaitu:

1. Baris 2 merupakan pendeklarasian variabel yang dibutuhkan dalam operasi HitungMinat().
2. Baris 3-5 merupakan operasi seleksi, apakah minat yang dimasukkan="RPL", jika benar maka baris 4 akan dijalankan
3. Baris 4 merupakan perintah untuk menghitung prosentase kecocokan antara minat yang dipilih user dengan hasil analisis sistem.
4. Baris 6-8 merupakan operasi seleksi, apakah minat yang dimasukkan="KCV", jika benar maka baris 7 akan dijalankan
5. Baris 9-12 merupakan operasi seleksi, apakah minat yang dimasukkan="RPL", jika benar maka baris 10 akan dijalankan
6. Baris 13-14 perintah untuk menampilkan hasil prosentase kecocokan.

5.5 Implementasi Antar Muka

5.5.1 Tampilan Halaman Data Mahasiswa

Pada halaman admin terdapat beberapa halaman antara lain halaman Login, halaman Data Mahasiswa, halaman Data Pelatihan, halaman Hasil Tes, dan halaman Data *Account*. Gambar 5.5 merupakan tampilan dari halaman Data Mahasiswa.



Gambar 5.5 Tampilan Halaman Data Mahasiswa
Sumber: Implementasi

Pada halaman Data Mahasiswa, admin atau operator dapat melakukan penyimpanan, pengubahan, penghapusan dan pencarian data mahasiswa. Untuk melakukan penyimpanan admin dapat mengisi data mahasiswa yang akan dimasukkan pada form Memasukkan Data kemudian menekan tombol “Simpan”. Data mahasiswa yang telah dimasukkan akan diperlihatkan pada tabel. Untuk mengubah atau menghapus data mahasiswa, admin atau operator memilih data yang akan diubah atau dihapus pada tabel, kemudian admin atau operator

menekan tombol “Hapus”, jika akan mengubah data, dan “Edit” jika akan menghapus data. Admin atau operator dapat melakukan pencarian data dengan memasukkan Nim atau Nama mahasiswa yang akan dicari pada form Pencarian Data. Data yang dicari akan ditampilkan pada tabel.

5.5.2 Tampilan Halaman Data Pelatihan

Pada halaman Data Pelatihan, admin dapat melakukan penyimpanan, perubahan, penghapusan dan pencarian data pelatihan. Untuk melakukan penyimpanan admin dapat mengisi data pelatihan yang akan dimasukkan pada form Memasukkan Data kemudian menekan tombol “Simpan”. Data mahasiswa yang telah dimasukkan akan diperlihatkan pada tabel. Untuk mengubah atau menghapus data pelatihan, admin memilih data yang akan diubah atau dihapus pada tabel, kemudian admin menekan tombol “Edit” jika akan mengubah data, dan “Hapus” jika akan menghapus data. Admin dapat melakukan pencarian data dengan memasukkan kode pelatihan yang akan dicari pada form Pencarian Data Pelatihan. Data yang dicari akan ditampilkan pada tabel.

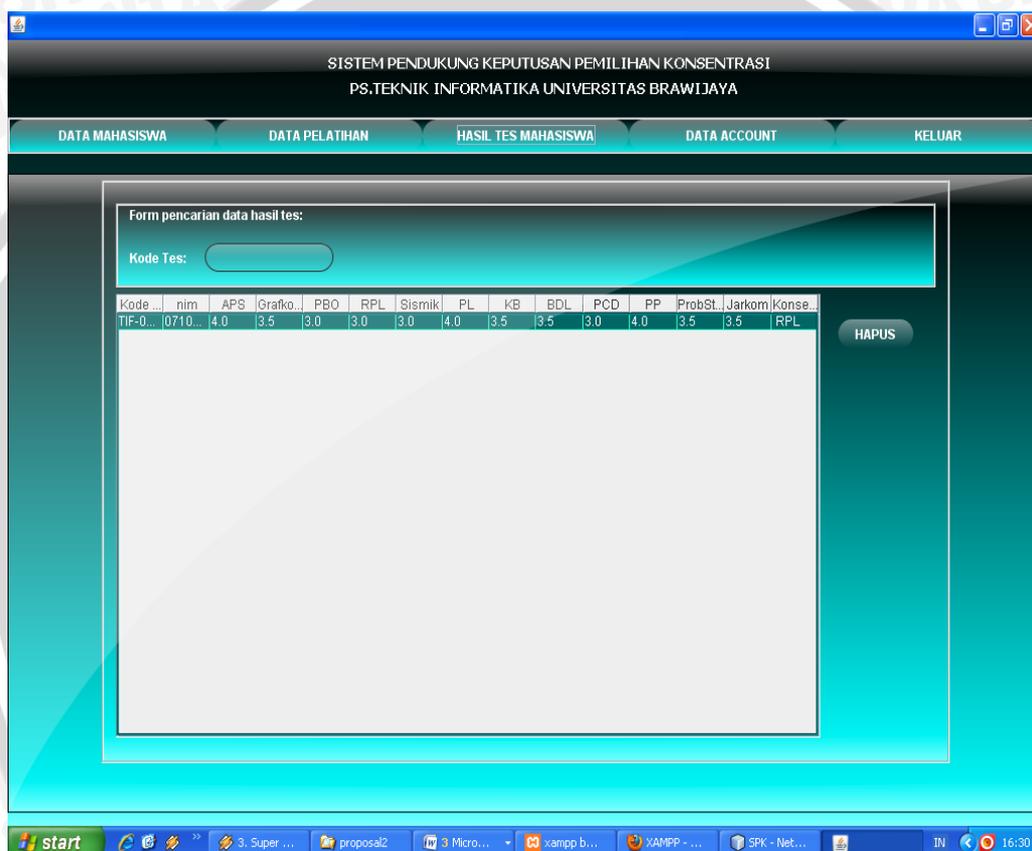
The screenshot displays the 'SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KONSENTRASI' interface. The main section is 'DATA PELATIHAN', which includes 14 dropdown menus for selecting subjects. Below this is a 'Form pencarian data pelatihan' which contains a table of training data.

Kode P...	APS	Grafikom	PBO	RPL	Sismik	PL	KG	BDL	PCD	PP	ProbStat	Jarkom	Konse...	Minat
Data 1	4	3,5	4	3,5	3,5	4	4	4	3	4	3,5	4	RPL	RPL
Data 2	4	3,5	4	3,5	4	4	3	4	3	4	3	4	RPL	KCV
Data 3	4	3,5	3,5	3,5	4	3	3,5	4	3	4	3,5	4	RPL	KCV
Data 4	3	3	3	2	3	4	2,5	3,5	2,5	3	3	3	RPL	KCV
Data 5	4	3,5	3,5	4	4	4	3	3,5	3	4	3	3	RPL	RPL
Data 6	3	3,5	3	2,5	3,5	4	3	4	3	3,5	3	2,5	RPL	KBJ
Data 7	3,5	3,5	4	4	3,5	4	2,5	4	3	3,5	3	3	RPL	RPL
Data 8	3,5	3,5	3	3,5	4	4	3,5	4	3	3	3	3,5	KBJ	KBJ
Data 9	3,5	3,5	3	2,5	4	3	3	3,5	3	4	3	4	KBJ	KBJ
Data 10	4	3,5	2,5	3,5	2	4	3	3,5	2,5	4	4	4	KBJ	KBJ
Data 11	3	3,5	2,5	2	3	3,5	3,5	2	3	3	3,5	3	KBJ	KBJ
Data 12	3,5	3,5	3	2	4	4	4	3,5	2,5	3,5	3	4	KBJ	KBJ
Data 13	3,5	3,5	4	3	4	3	3,5	3,5	3,5	3,5	4	3	RPL	RPL
Data 14	3,5	3	3	3,5	3,5	3	3,5	3	2,5	4	3	3	RPL	RPL

Gambar 5.6 Tampilan Halaman Data Pelatihan
Sumber: Implementasi

5.5.3 Tampilan Halaman Hasil Tes

Pada halaman Hasil Tes, admin atau operator akan dihadapkan pada tabel yang berisi daftar mahasiswa yang telah melakukan tes keputusan. Admin dapat menghapus data hasil tes dengan memilih data yang akan dihapus pada tabel, kemudian menekan tombol “Hapus”. Admin juga dapat melakukan pencarian data hasil tes dengan memasukkan kode tes pada form Pencarian Data Hasil Tes.



Gambar 5.7 Tampilan Halaman Hasil Tes

Sumber: Implementasi

5.5.4 Tampilan Halaman Login

Pada halaman *Login*, admin mengisi *username* dan *password* agar dapat masuk ke sistem. Kemudian admin atau operator menekan tombol “Masuk”.

SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KONSENTRASI
PS.TEKNIK INFORMATIKA UNIVERSITAS BRAWIJAYA

Silahkan Login disini:

Username:

Password:

MASUK

Gambar 5.8 Tampilan Halaman *Login*
Sumber: Implementasi

5.5.5 Tampilan Halaman *Data Account*

Pada halaman *Data Account*, admin dihadapkan pada form input data, dimana admin dapat menambah dan mengubah data *account*. Admin dapat melakukan perubahan data *account* dengan memilih *account* yang ditampilkan pada tabel kemudian menekan tombol “Edit”. Kemudian admin dapat melakukan perubahan *account* pada form input data, kemudian menekan tombol “Simpan”.

SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KONSENTRASI
PS.TEKNIK INFORMATIKA UNIVERSITAS BRAWIJAYA

DATA MAHASISWA DATA PELATIHAN HASIL TES MAHASISWA DATA ACCOUNT KELUAR

Form set password:

Nip:

Nama:

Username:

Password:

Status:

SIMPAN

Nip	Nama	Username	Password	Status
0710683038	arif	admin	admin	admin
0710683039	farid	operator	operator	operator

EDIT HAPUS

Gambar 5.9 Tampilan Halaman *Data Account*
Sumber: Implementasi

5.5.6 Tampilan Halaman Tes Mahasiswa

Pada halaman Tes Mahasiswa, mahasiswa dihadapkan pada form pengisian nim dan nama. Kemudian mahasiswa memasukkan nilai mata kuliah yang telah ditempuh sesuai dengan nama mata kuliah yang disediakan oleh sistem. Untuk melakukan perhitungan, mahasiswa dapat menekan tombol “Proses”.

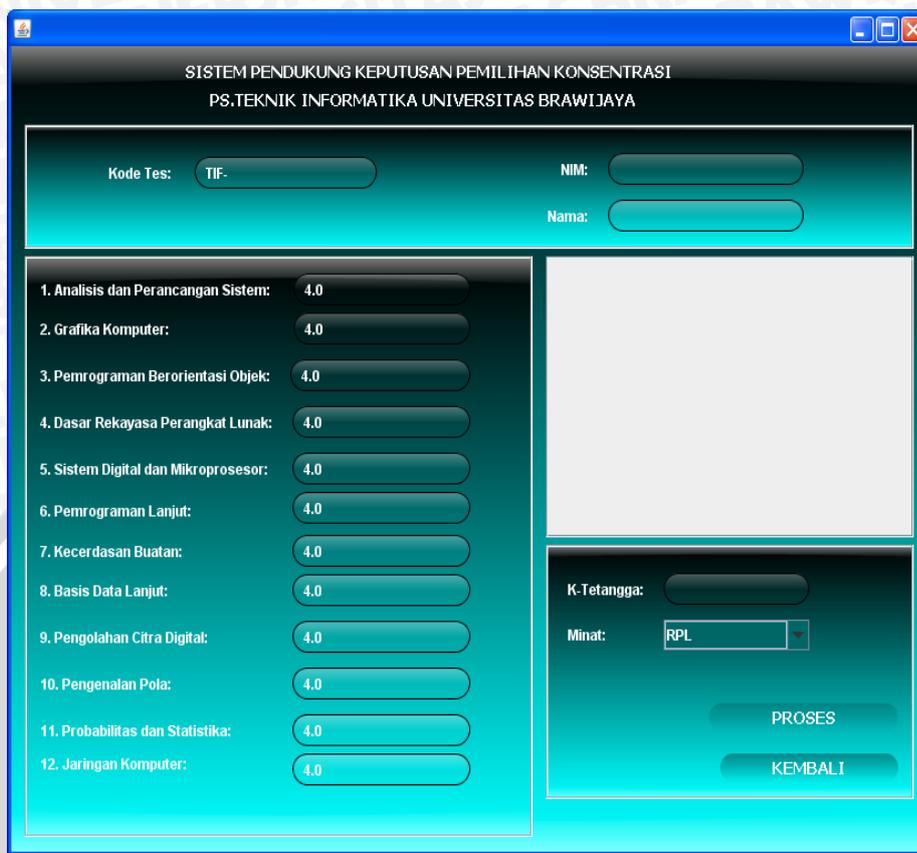
SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN KONSENTRASI PS.TEKNIK INFORMATIKA UNIVERSITAS BRAWIJAYA			
Kode Tes:	<input type="text"/>	NIM:	<input type="text"/>
		Nama:	<input type="text"/>
Masukkan nilai mata kuliah anda:			
1. Analisis dan Perancangan Sistem:	<input type="text" value="A"/>	7. Kecerdasan Buatan:	<input type="text" value="A"/>
2. Grafika Komputer:	<input type="text" value="A"/>	8. Basis Data Lanjut:	<input type="text" value="A"/>
3. Pemrograman Berorientasi Objek:	<input type="text" value="A"/>	9. Pengolahan Citra Digital:	<input type="text" value="A"/>
4. Dasar Rekayasa Perangkat Lunak:	<input type="text" value="A"/>	10. Pengenalan Pola:	<input type="text" value="A"/>
5. Sistem Digital dan Mikroprosesor:	<input type="text" value="A"/>	11. Probabilitas dan Statistika:	<input type="text" value="A"/>
6. Pemrograman Lanjut:	<input type="text" value="A"/>	12. Jaringan Komputer:	<input type="text" value="A"/>
<input type="button" value="PROSES"/>			

Gambar 5.10 Tampilan Halaman Tes Mahasiswa

Sumber: Implementasi

5.5.7 Tampilan Halaman Pendukung Keputusan

Pada halaman Penentuan Keputusan *user* akan dihadapkan pada form yang berisi tabel data pelatihan, form yang berisi kode tes, nim, nama, dan nilai mata kuliah yang telah diasukkan. Untuk mengetahui hasil analisis sistem, mahasiswa dapat mengisikan nilai “K”, dimana nilai “K” merupakan jumlah tetangga terdekat antara data tes dan data pelatihan. Kemudian mahasiswa menekan tombol ”Proses” dan hasil analisa sistem akan keluar. Apabila ingin melakukan tes ulang, mahasiswa dapat menekan tombol “Kembali”.



Gambar 5.11 Tampilan Halaman Pendukung Keputusan
Sumber: Implementasi

5.6 Penerapan *K-Nearest Neighbor* dalam Pemilihan Konsentrasi

Pada sistem pendukung keputusan ini digunakan metode *K-Nearest Neighbor* untuk mengambil keputusan konsentrasi mahasiswa berdasarkan nilai matakuliah yang telah dimasukkan. Proses perhitungan dilakukan dengan cara menghitung jarak terpendek data masukan terhadap data yang ada pada basis pengetahuan. Misalnya ingin diketahui konsentrasi dari salah satu mahasiswa dengan perincian nilai sebagai berikut:

Tabel 5.4 Contoh Data Tes

1	2	3	4	5	6	7	8	9	10	11	12	Minat
B	B+	C+	C	C	C+	B	C+	C+	C+	B	B	RPL

Sumber: Implementasi

Keterangan:

- 1. Analisis dan Perancangan Sistem



2. Grafika Komputer
3. Pemrograman Berorientasi Objek
4. Rekayasa Perangkat Lunak
5. Sistem Digital dan Mikroprosesor
6. Pemrograman Lanjut
7. Kecerdasan Buatan
8. Basis Data Lanjut
9. Pengolahan Citra Digital
10. Pengenalan Pola
11. Probabilitas dan Statistika
12. Jaringan Komputer

Berdasarkan algoritma *K-Nearest Neighbor*, dapat dilakukan perhitungan sebagai berikut:

1. Menentukan K , yaitu dengan memilih sejumlah K tetangga terdekat, misalnya $K=7$.
2. Menghitung jarak antara atribut setiap data tes dengan data pelatihan yang terdapat pada Tabel 5.5 dengan persamaan (2.3):

Tabel 5.5 Data Pelatihan

Data ke	1	2	3	4	5	6	7	8	9	10	11	12	Konsentrasi
Data ke-1	A	B+	A	B+	B+	A	A	A	B	A	B+	A	RPL
Data ke-2	A	B+	A	B+	A	A	B	A	B	A	B	A	RPL
Data ke-3	A	B+	B+	B+	A	B	B+	A	B	A	B+	A	RPL
Data ke-4	B	B	B	C	B	A	C+	B+	C+	B	B	B	RPL
Data ke-5	A	B+	B+	A	A	A	B	B+	B	A	B	B	RPL
Data ke-6	B	B+	B	C+	B+	A	B	A	B	B+	B	C+	RPL
Data ke-7	B+	B+	A	A	B+	A	C+	A	B	B+	B	B	RPL
Data ke-8	B+	B	B	B+	B+	B	B+	B	B	C+	A	B	RPL

Data ke-9	B+	B+	A	A	A	B	A	B+	B+	B	A	A	RPL
Data ke-10	B+	B+	A	B	A	B	B+	B+	B+	B+	A	B	RPL
Data ke-11	B+	B	A	B+	B+	B	A	A	B	A	A	C+	RPL
Data ke-12	B+	B+	B	B+	B	B	A	A	B	A	A	C	KCV
Data ke-13	B	B	A	B	B+	B	A	B+	B	A	A	C+	KCV
Data ke-14	B+	B+	B	A	B+	B	A	A	B	A	A	B	KCV
Data ke-15	B+	B	B	A	A	B	A	A	B	A	A	C+	KCV
Data ke-16	B+	B	A	B+	B	B	A	B+	B	A	A	B+	KCV
Data ke-17	B+	B	A	A	B	B	B+	A	B	B+	A	A	KCV
Data ke-18	B+	B	A	B+	A	B	B+	B	B+	B+	A	B	KCV
Data ke-19	B+	B	A	A	B+	B	A	A	B+	A	A	B	KCV
Data ke-20	B+	B	B	B	B	B	C+	B+	B+	B	A	C	KCV
Data ke-21	B+	B+	B	B+	A	A	B+	A	B	B	B	B+	KBJ
Data ke-22	B+	B+	B	C+	A	B	B	B+	B	A	B	A	KBJ
Data ke-23	A	B+	C+	B+	C	A	B	B+	C+	A	A	A	KBJ
Data ke-24	B	B+	C+	C	B	B	B+	C	B	B	B+	B	KBJ
Data ke-25	B+	B+	B	C	A	A	A	B+	C+	B+	B	A	KBJ
Data ke-26	B+	B+	A	B	B	B	B+	A	A	A	A	B+	KBJ
Data ke-27	B+	B	A	B	B+	B	B	B	B	C+	A	B+	KBJ
Data ke-28	B+	B	B	B+	B	B	A	B	B	B	C	B+	KBJ
Data ke-29	B+	B	B	B	A	B	B	B	B	B+	C+	B	KBJ
Data ke-30	B+	B+	B+	B	A	B	A	B+	B	A	A	B	KBJ
Data ke-31	B+	B	B	B+	B	B	A	A	B	C+	A	B+	KCV
Data ke-32	B+	B	B	C	B	B	B	B	B	C+	C	B+	KCV

ke-32													
Data	A	B+	B	A	B	A	B	A	B	B	C+	B	KBJ
ke-33													

Sumber: Nilai Mata Kuliah angkatan 2007-2008

Berdasarkan perhitungan *euclidean distance* diperoleh jarak antara data tes dengan data pelatihan, kemudian dilakukan pengurutan dari jarak terkecil sampai jarak terbesar sebagai berikut:

Tabel 5.6 Pengurutan Jarak Data Pelatihan

Tetangga Ke-	Jarak	Konsentrasi
1.	1.8027756377319946	KBJ
2.	1.9364916731037085	KCV
3.	2.29128784747792	RPL
4.	2.692582403567252	KBJ
5.	2.692582403567252	RPL
6.	2.7386127875258306	KCV
7.	2.7838821814150108	KBJ
8.	2.8284271247461903	KBJ
9.	2.958039891549808	RPL
10.	3.0	KCV
11.	3.082207001484488	KBJ
12.	3.278719262151	KBJ
13.	3.278719262151	KBJ
14.	3.391164991562634	KBJ
15.	3.427827300200522	KCV
16.	3.427827300200522	KCV
17.	3.4641016151377544	KBJ
18.	3.4641016151377544	KCV
19.	3.4641016151377544	RPL
20.	3.5	KBJ
21.	3.570714214271425	KCV
22.	3.605551275463989	KBJ
23.	3.7080992435478315	KCV
24.	3.7080992435478315	KCV
25.	3.8078865529319543	RPL
26.	3.840572873934304	RPL
27.	3.840572873934304	RPL
28.	3.968626966596886	RPL
29.	4.0	KCV
30.	4.0	RPL
31.	4.092676385936225	KCV
32.	4.123105625617661	RPL

33.	4.183300132670378	RPL
-----	-------------------	-----

Sumber: Implementasi

- Pengambilan tetangga terdekat sejumlah K=7.

Tabel 5.7 Pengambilan Tetangga Terdekat dengan K=7

Tetangga Ke-	Jarak	Konsentrasi
1.	1.8027756377319946	KBJ
2.	1.9364916731037085	KCV
3.	2.29128784747792	RPL
4.	2.692582403567252	KBJ
5.	2.692582403567252	RPL
6.	2.7386127875258306	KCV
7.	2.7838821814150108	KBJ

Sumber: Implementasi

- Dari hasil pengambilan data sejumlah 7 data, diperoleh konsentrasi terdekat sebagai berikut:

- RPL = 2
- KCV = 2
- KBJ = 3

- Setelah didapatkan jumlah konsentrasi dari 7 tetangga terdekat, kemudian diambil konsentrasi dengan jumlah terbanyak sebagai hasil keputusan dari sistem, yaitu “KBJ”.

- Perhitungan kecocokan antara minat dari data tes dengan hasil keputusan sistem, yaitu:

$$\% \text{ kecocokan} = (\text{jumlah data yang cocok dari } K / K) * 100$$

$$= \frac{2}{7} \times 100\%$$

$$= 28,57\%$$

Berdasarkan hasil perhitungan, diperoleh prosentase kecocokan antara keputusan sistem dengan minat data tes yaitu sebesar 28,57%.

