

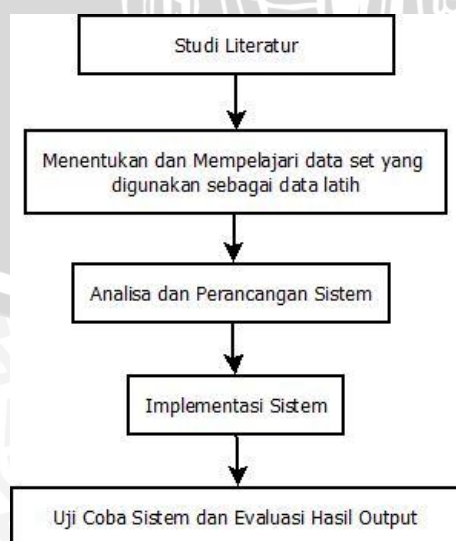
BAB III

METODE PENELITIAN

Bab metodologi penelitian ini akan dibahas metode yang digunakan, perancangan sistem, dan langkah-langkah yang dilakukan dalam penelitian tentang implementasi *K-nearest neighbor* dan *fuzzy k-nearest neighbor* untuk mendiagnosa diabetes pada pasien. Penelitian dilakukan dengan tahapan-tahapan sebagai berikut :

1. Mempelajari literatur yang memuat *K-nearest neighbor*, *fuzzy K-nearest neighbor* dan diabetes Indian Pima.
2. Mempelajari dataset yang digunakan untuk data latih , yaitu dataset diabetes Indian Pima.
3. Melakukan analisa dan perancangan sistem menggunakan algoritma *K-nearest neighbor* dan *fuzzy K-nearest neighbor*.
4. Membangun perangkat lunak berdasarkan analisa dan perancangan yang telah dilakukan (implementasi)
5. Melakukan uji coba dan mengevaluasi hasil output yang dihasilkan dari sistem.

Adapun tahapan penelitian dapat digambarkan dalam bentuk diagram alir yang ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.1 Studi literatur

Dalam penelitian ini dibutuhkan studi literatur untuk merealisasikan tujuan dan penyelesaian masalah. Teori-teori mengenai datadiabetes Indian Pima, himpunan fuzzy, metode *K-Nearest Neighbor* (K-NN) dan metode *fuzzy K-nearest neighbor* (FK-NN) sebagai dasar penelitian yang diperoleh dari buku, jurnal, dan browsing Internet serta literatur lain yang mendukung bab 2. Kemudian data penelitian diproses menggunakan K-NN dan FK-NN sehingga dapat digunakan untuk analisis. Setelah dianalisis maka dapat diimplementasikan ke dalam program.

3.2 Menentukan dan Mempelajari Dataset yang Digunakan

Data yang digunakan dalam penelitian ini adalah data yang diambil dari sumber data diabetes di alamat <http://archive.ics.uci.edu>. Pada *dataset* diabetes Indian Pima, terdapat 8 atribut, diantaranya jumlah hamil, 2 jam PP (OGTT), tekanan diastolik, indeks massa badan (IMB), riwayat diabetes keluarga (DPF) dan usia. Sedangkan untuk kelas output, yaitu:

0 = Negatif Diabetes Melitus (DM)

1 = Positif DM

Data diabetes Indian Pima berjumlah sebanyak 768 data klinis. Pada data ini, tidak semua atribut memiliki nilai yang lengkap dimana kelengkapan nilai atribut sangat berpengaruh pada hasil klasifikasi. Jumlah data tidak lengkap (*missing value*) pada masing-masing atribut dapat dilihat pada tabel 3.1.

Tabel 3.1. Jumlah *missing value* pada setiap atribut

Hamil	OGTT	Diastolik	TSFT	INS	IMB	DPF	Usia
111	5	35	227	373	11	0	0

Pada penelitian ini aturan untuk mengatasi *missing value* pada masing-masing atribut sebagai berikut:

1. Nilai nol pada atribut hamil dapat diasumsikan bahwa nilai tersebut menyatakan pasien belum pernah melahirkan, sehingga hal ini dimungkinkan sesuai kondisi sebenarnya.
2. Data dengan nilai nol pada atribut glukosa, DBP, dan BMI dapat dihilangkan karena jumlahnya tidak terlalu banyak sehingga tidak begitu mempengaruhi hasil klasifikasi.
3. Karena atribut TSFT dan INS memiliki jumlah nilai yang tidak ada sangat besar, maka kedua atribut ini tidak mungkin dihilangkan dan tidak mungkin dipakai dalam pengklasifikasian. Oleh karena itu, dalam penelitian ini atribut TSFT dan INS tidak digunakan.

Setelah proses penanganan *missing value* dilakukan sesuai dengan aturan di atas, maka didapatkan 724 data dari 768 data aslinya. Data yang digunakan pada penelitian ini berjumlah 290 record (154 record data negatif diabetes dan 136 record data positif diabetes) untuk diproses lebih lanjut.

3.3 Analisa dan Perancangan Sistem

Secara umum sistem yang dibangun adalah suatu perangkat lunak untuk melakukan perbandingan kinerja antara *K-nearest neighbor* dan *fuzzy K-nearest neighbor* dalam diagnosis penyakit Diabetes Melitus (DM). Sistem ini bertujuan untuk menentukan apakah seorang pasien positif atau negatif terkena DM berdasarkan beberapa atribut yang digunakan dan juga membuktikan klasifikasi terbaik antara *K-nearest neighbor* dan *fuzzy K-nearest neighbor*.

Pada tahap awal, sistem telah memiliki data yang digunakan sebagai parameter diagnosis dimana semua nilainya berupa data numerik sehingga proses *K-nearest neighbor* dan *fuzzy K-nearest neighbor* bisa dilakukan. *Dataset* diabetes Indian Pima terdiri dari 290 record dan tidak terdapat *missing value*.

Perancangan data yang dilakukan menggunakan variasi pada data latih dan data uji, di mana tiap pengujiannya menggunakan komposisi yang berubah yaitu 80, 130, 180, dan 230 data latih. sedangkan data uji yang digunakan data yang tetap yaitu 50 data.

Perangkat lunak ini akan menguji keakuratan hasil diagnosis *dataset* faktor penyakit diabetes melitus terhadap data sebenarnya. Parameter uji yang berkaitan dengan nilai k (tetangga) setiap kelas dan data latih yang berpengaruh terhadap tingkat akurasi dan pengaruh pemberian bobot (m) pada perhitungan yang dilakukan.

Pada perancangan sistem ini akan dipaparkan mengenai proses secara keseluruhan dalam membangun sistem ini. Sistem akan melakukan klasifikasi menggunakan algoritma *K-nearest neighbor* dan *fuzzy K-nearest neighbor* sesuai dengan parameter yang diinputkan lalu hasil akurasi dibandingkan untuk mengetahui algoritma terbaik untuk klasifikasi data.

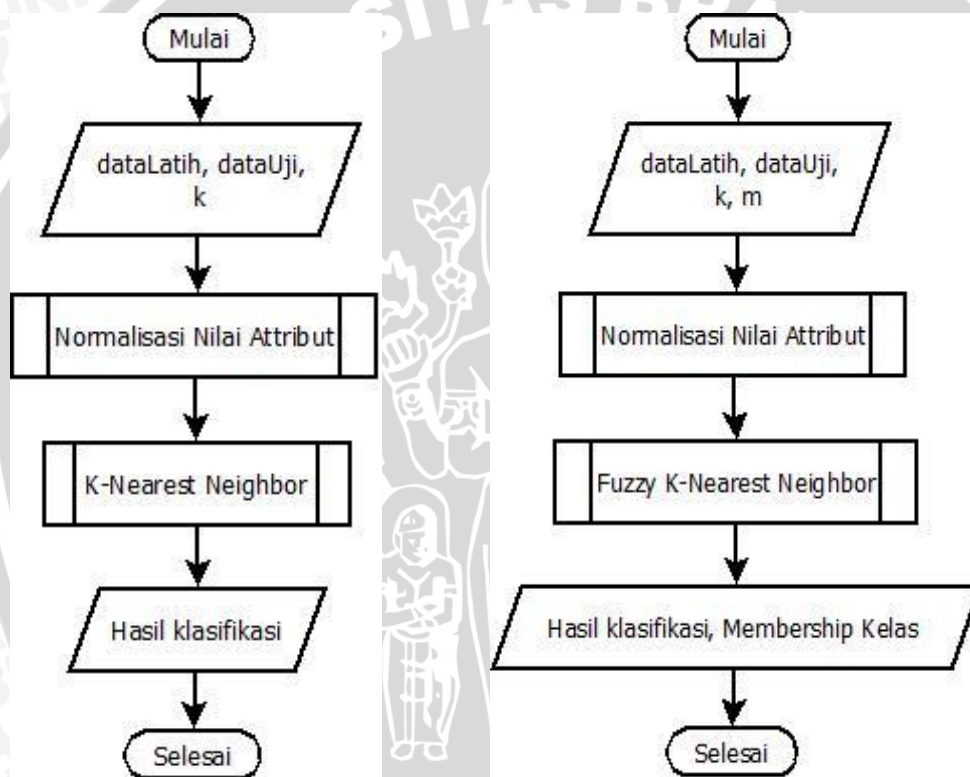
3.3.1 Proses Klasifikasi *K-Nearest Neighbor* dan *Fuzzy K-Nearest Neighbor*

Pada tahap ini, sistem akan melakukan proses klasifikasi pada data latih menggunakan *K-nearest neighbor* (KNN) dan *fuzzy K-nearest neighbor* (FK-NN). Tahapan dari proses ini sebagai berikut :

1. Proses input data uji dan data latih dari Microsoft excel.
2. Melakukan perhitungan normalisasi atribut menggunakan *min-max normalization*.
3. Menghitung *euclidean distance*.
4. Proses KNN yaitu mengambil mayoritas kelas pada K yang telah ditentukan sebagai kelas target pada data yang baru.
5. Proses FKNN yaitu menghitung nilai derajat keanggotaan dan mengambil nilai terbesar dari proses tersebut dan ditentukan kelas targetnya.
6. Perbandingan akurasi antara KNN dan FK-NN.

3.3.2 Flowchart Sistem

1. Melakukan input data latih dan data uji diabetes Indian Pima serta nilai k .
2. Proses normalisasi atribut menggunakan normalisasi *min-max*.
3. Melakukan klasifikasi diabetes Indian Pima menggunakan *K-nearest neighbor* dan *fuzzy K-nearest neighbor*.
4. Akurasi kedua sistem dihitung dan dilihat perbandingan kinerjanya.



Gambar 3.2 Alur proses sistem

Pada gambar 3.2, terlihat sistem ini memiliki 3 proses utama, di antaranya:

1. Normalisasi nilai atribut

Merupakan proses transformasi untuk membakukan skala pengaruh yang ada pada atribut terhadap hasil. Hal ini diperlukan karena pada perhitungan jarak euclidean, atribut berskala panjang dapat berpengaruh lebih besar daripada atribut berskala pendek.

2. K-Nearest Neighbor (K-NN)

Merupakan proses mengklasifikasikan dataset berdasarkan jarak terdekat antara 2 record dimana pada dataset ini terdapat 2 klasifikasi yaitu positif diabetes dan negatif diabetes.

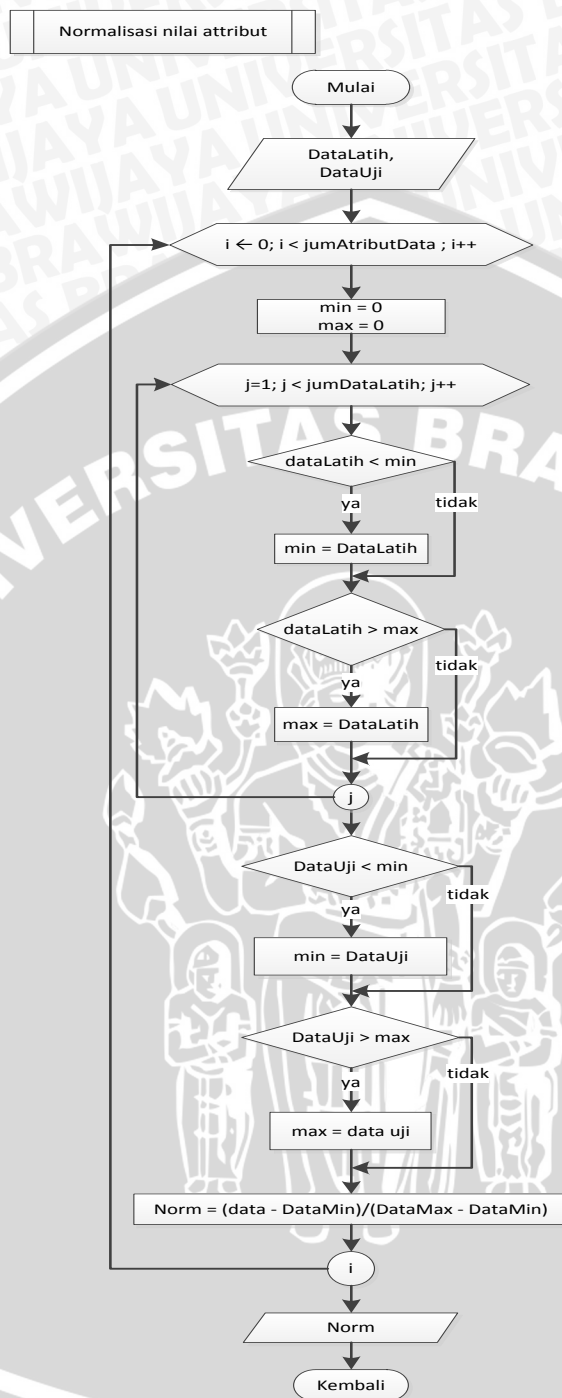
3. Fuzzy K-Nearest Neighbor

Dataset diklasifikasi berdasarkan nilai derajat keanggotaan terbesar yang dimiliki masing-masing kelas

3.3.3 Normalisasi Nilai Atribut

Proses ini berfungsi untuk membakukan skala pengaruh yang ada pada atribut terhadap hasil. Persamaan (2-1) digunakan untuk menghitung normalisasi data. Selanjutnya, data hasil normalisasi digunakan untuk proses selanjutnya. Untuk lebih jelasnya, bisa dilihat di gambar 3.3.





Gambar 3.3 Alur proses Normalisasi atribut

3.3.4 Klasifikasi Menggunakan *K-Nearest Neighbor* (K-NN)

Pada proses ini, sistem melakukan klasifikasi pada dataset ke dalam 2 kelas sesuai dengan data latih. Pada klasifikasi ini dihitung jarak

menggunakan *Euclidean Distance*. Langkah-langkah dalam proses K-NN sebagai berikut:

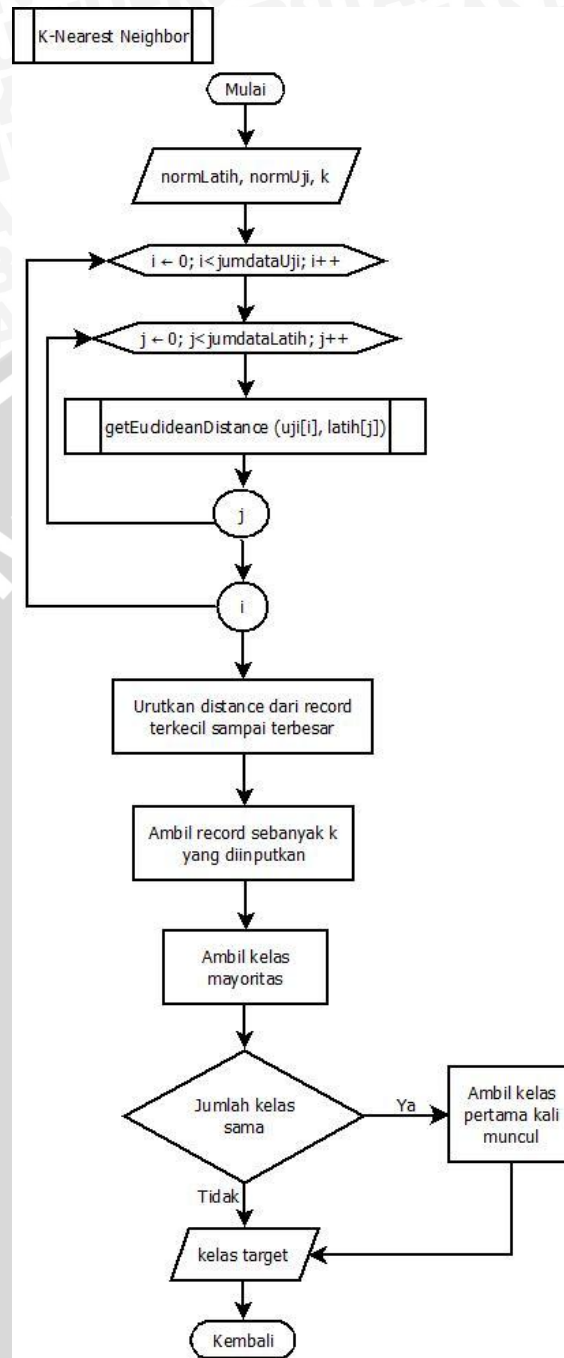
1. Data latih dan data uji diabetes Indian Pima yang telah dinormalisasi dan nilai k dimasukkan user
 2. Dilakukan iterasi untuk menghitung jarak antara *record* data baru dengan tiap *record* data latih menggunakan persamaan (2-2). Proses ini terlihat pada gambar 3.5.
 3. *Record* diurutkan dan diambil k sesuai masukan
 4. Penentuan keputusan diambil dari mayoritas kelas dalam k , jika jumlah kelas sama, maka yang diambil adalah kelas pertama yang muncul.
 5. Output berupa hasil kelas target diabetes Indian Pima pada data uji.
- Flowchart* untuk proses K-NN bisa dilihat pada gambar 3.4

3.3.5 Klasifikasi Menggunakan *Fuzzy K-Nearest Neighbor*

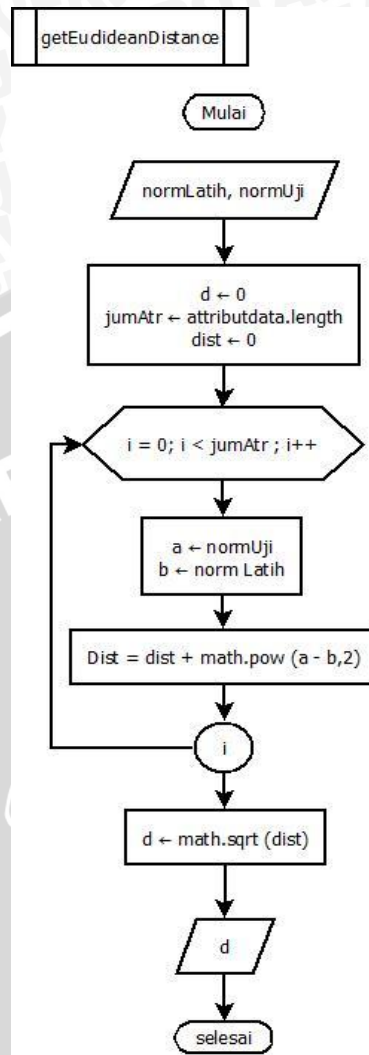
Pada proses ini, awalnya sama dengan K-NN, tetapi setelah proses penyortiran data dan penentuan K , Fuzzy k -nn memberikan nilai derajat keanggotaan pada setiap kelas untuk menentukan kelas data yang baru. Langkahnya sebagai berikut:

1. Masukan berupa jarak sejumlah k yang diambil pada proses K-NN ($\text{dataNN}[]$), k dan bobot pangkat ($m > 1$)
2. Hitung nilai keanggotaan tiap kelas dan diambil keanggotaan terbesar yang digunakan untuk menentukan kelas target yang baru
3. Output berupa hasil kelas target dari data uji dan nilai keanggotaan tiap kelas

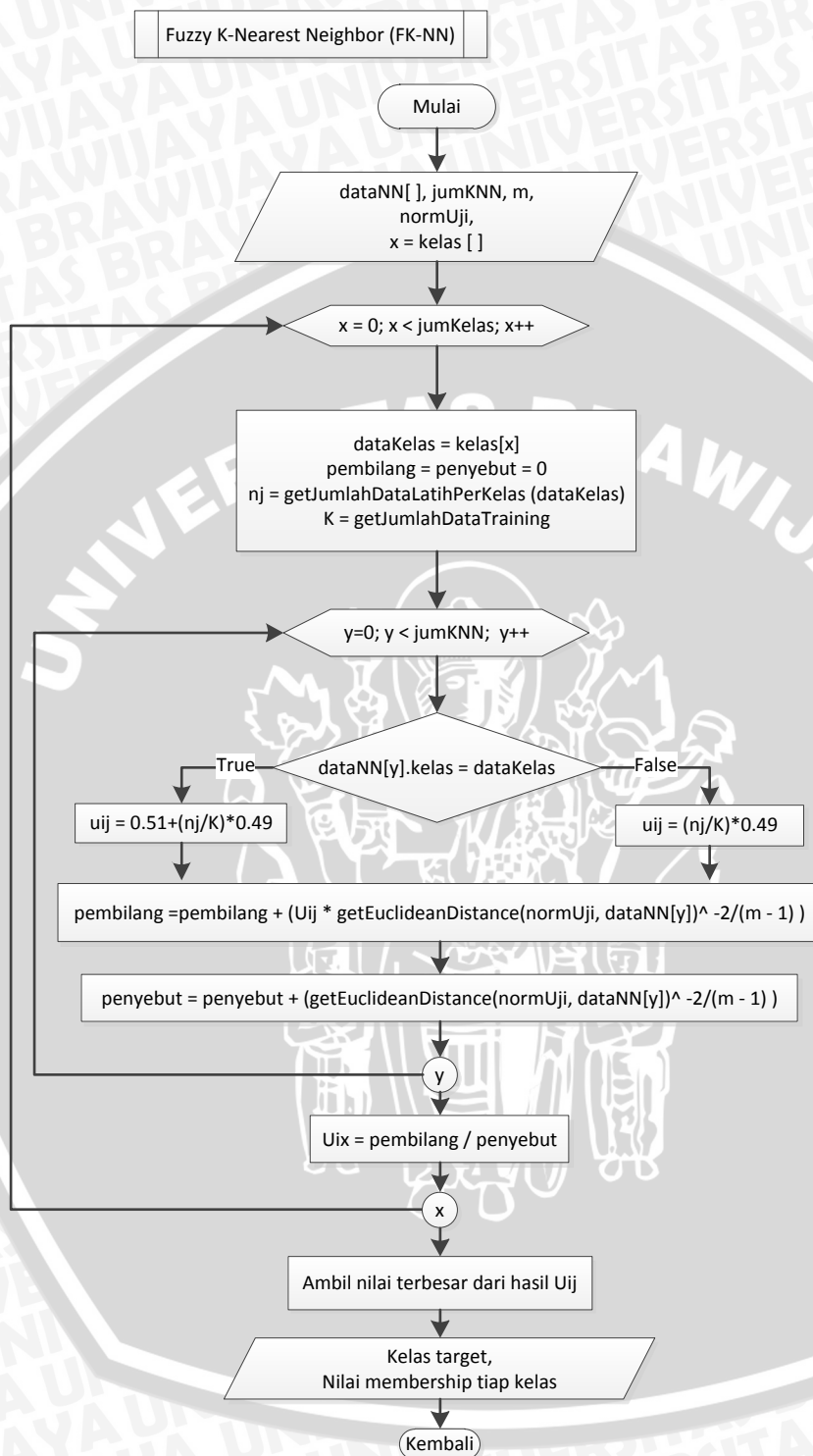
Flowchart untuk proses FK-NN bisa dilihat pada gambar 3.5



Gambar 3.4 Alur proses K-NN



Gambar 3.5 Proses *Euclidean Distance*



Gambar 3.6 Alur Proses FK-NN

3.4 Perhitungan Manual

Sampel data yang digunakan pada perhitungan ini diambil dari *dataset* diabetes Indian Pima yang menggunakan 8 atribut. Data yang diambil berjumlah 16 data. Pada perhitungan manual ini digunakan 15 data sebagai data latih dan 1 data untuk data uji. Tabel 3.2 menunjukkan data latih dan tabel 3.3 menunjukkan data uji.

Tabel 3.2 Contoh data latih sistem

No	Hamil	OGTT	Diastolik	IMB	DPF	Usia	Diagnosa
1	6	148	72	33.6	0.627	50	1
2	1	85	66	26.6	0.351	31	0
3	8	183	64	23.3	0.672	32	1
4	1	89	66	28.1	0.167	21	0
5	0	137	40	43.1	2.288	33	1
6	5	116	74	25.6	0.201	30	0
7	3	78	50	31.0	0.248	26	1
8	2	197	70	30.5	0.158	53	1
9	4	110	92	37.6	0.191	30	0
10	10	168	74	38.0	0.537	34	1
11	10	139	80	27.1	1.441	57	0
12	1	189	60	30.1	0.398	59	1
13	5	166	72	25.8	0.587	51	1
14	1	103	30	43.3	0.183	33	0
15	1	115	70	34.6	0.529	32	1

Tabel 3.3 Contoh data uji sistem

Hamil	OGTT	Diastolik	IMB	DPF	Usia	Diagnosa
3	126	88	39.3	0.704	27	?

3.4.1 Normalisasi Nilai Atribut

Pada penelitian ini menggunakan normalisasi *min-max* yang ditunjukkan pada persamaan (2-1). Nilai rentang (*range*) didapat dari nilai *max* dikurang nilai *min* data latih pada tabel 3.2. Berikut ini adalah contoh perhitungan normalisasi pada atribut *pregnant*.

$$\begin{aligned}
 V &= 6 \text{ (nilai record pertama atribut pregnant)} \\
 \max &= 10 \\
 \min &= 0 \\
 \text{range} &= 10 - 0 = 10
 \end{aligned}$$

Setelah diketahui nilai *max*, *min* dan *range*, kemudian lakukan perhitungan menggunakan persamaan (2-1).

$$\begin{aligned}
 V' &= \frac{v - \min A}{\text{range } A} \\
 &= \frac{6 - 0}{10} \\
 &= 0.6
 \end{aligned}$$

Dari perhitungan ini, didapatkan nilai normalisasi *record* pertama pada atribut *pregnant* adalah 0.6. Proses yang sama juga dilakukan pada semua *record* seluruh atribut. Hasil perhitungan normalisasi ditunjukkan tabel 3.5. Tabel 3.6 menunjukkan data uji yang telah dinormalisasikan.

Tabel 3.4 Range data

	Hamil	OGTT	Diastolik	IMB	DPF	Usia
max	10	197	92	43.3	2.288	59
min	0	78	30	23.3	0.158	21
range	10	119	62	20	2.13	38

Tabel 3.5 Data latih yang telah dinormalisasi

No	Hamil	OGTT	Diastolik	IMB	DPF	Usia
1	0.6	0.588	0.677	0.515	0.220	0.763
2	0.1	0.059	0.581	0.165	0.091	0.263
3	0.8	0.882	0.548	0	0.241	0.289
4	0.1	0.092	0.581	0.24	0.004	0
5	0	0.496	0.161	0.99	1	0.316
6	0.5	0.319	0.710	0.115	0.020	0.237
7	0.3	0	0.323	0.385	0.042	0.132
8	0.2	1	0.645	0.36	0	0.842
9	0.4	0.269	1	0.715	0.015	0.237

10	1	0.756	0.710	0.735	0.178	0.342
11	1	0.513	0.806	0.19	0.602	0.947
12	0.1	0.933	0.484	0.34	0.113	1
13	0.5	0.739	0.677	0.125	0.201	0.789
14	0.1	0.210	0	1	0.012	0.316
15	0.1	0.311	0.645	0.565	0.174	0.289

Tabel 3.6 Data uji yang telah dinormalisasi

Hamil	OGTT	Diastolik	IMB	DPF	Usia
0.3	0.403	0.935	0.8	0.256	0.158

3.4.2. Menghitung Jarak *Record* Baru Data Uji di setiap *Record* Data Latih

Proses perhitungan jarak *record* baru data uji dengan setiap *record* data latih menggunakan persamaan (2-2) sebagai berikut.

$$d(x_1, x_2) = \sqrt{\sum_{r=1}^n (a_r(x_1) - a_r(x_2))^2}$$

$$= \sqrt{(0.3 - 0.6)^2 + (0.403 - 0.588)^2 + (0.935 - 0.677)^2 + (0.8 - 0.515)^2 + (0.256 - 0.220)^2 + (0.158 - 0.763)^2}$$

$$= 0.7998$$

Proses perhitungan jarak dilanjutkan antara data uji ke semua *record* data latih menggunakan persamaan yang sama. Hasil perhitungan jarak ditunjukkan pada tabel 3.7

Tabel 3.7 Hasil perhitungan jarak dan diurutkan dari yang terkecil

Record	Jarak	Rank Minimum Distance	Target
9	0.3220	1	0
15	0.4606	2	1
6	0.7933	3	0
1	0.7998	4	1
4	0.8153	5	0
10	0.8425	6	1
2	0.8523	7	0
7	0.8701	8	1
13	1.0378	9	1
14	1.0379	10	0

8	1.0852	11	1
3	1.1344	12	1
5	1.1454	13	1
12	1.2106	14	1
11	1.2782	15	0

3.4.3 Menentukan k record terdekat

Misal pada proses awal diinputkan nilai k = 3, maka diambil 3- k terdekat yang ada pada data uji. Setelah dilakukan proses perhitungan jarak, dilakukan sorting dari jarak terkecil, dapat dilihat pada tabel 3.6, record 9, 15 dan 6 merupakan record yang terpilih.

3.4.4 K-Nearest Neighbor (K-NN)

Pada tabel 3.6 terlihat kelas 0 lebih dominan daripada kelas 1, maka data uji tersebut merupakan kelas 0 (tidak terdeteksi diabetes).

3.4.5 Proses Fuzzy K-Nearest Neighbor (FK-NN)

Setelah proses penentuan k terdekat, proses FK-NN dimulai dengan mencari nilai keanggotaan untuk tiap kelas j dengan menggunakan persamaan (2-7). K = 15 (jumlah data latih), n₀ = 6, n₁ = 9

$$\begin{aligned}
 u_{0(0)} &= 0.51 + \left(\frac{6}{15}\right) * 0.49 \\
 &= 0.51 + 0.4 * 0.49 \\
 &= 0.51 + 0.196 \\
 &= 0.706
 \end{aligned}$$

$$\begin{aligned}
 u_{0(1)} &= \left(\frac{6}{15}\right) * 0.49 \\
 &= 0.4 * 0.49 \\
 &= 0.196
 \end{aligned}$$



$$\begin{aligned}
 u_{1(1)} &= 0.51 + \left(\frac{9}{15}\right) * 0.49 \\
 &= 0.51 + 0.6 * 0.49 \\
 &= 0.51 + 0.294 \\
 &= 0.804
 \end{aligned}$$

$$\begin{aligned}
 u_{1(0)} &= \left(\frac{6}{15}\right) * 0.49 \\
 &= 0.6 * 0.49 \\
 &= 0.294
 \end{aligned}$$

Setelah didapatkan keanggotaan untuk setiap kelas j , dilakukan perhitungan nilai keanggotaan dari data yang baru pada masing-masing kelas menggunakan persamaan (2-6), pada perhitungan manual di sini digunakan $m = 2$.

$$\begin{aligned}
 u_0 &= \frac{\left(0.706 + \left(0.3220 \frac{2}{2-1}\right)\right) + \left(0.196 + \left(0.4606 \frac{2}{2-1}\right)\right) + \left(0.706 + \left(0.7933 \frac{2}{2-1}\right)\right)}{\left(0.3220 \frac{2}{2-1}\right) + \left(0.4606 \frac{2}{2-1}\right) + \left(0.7933 \frac{2}{2-1}\right)} \\
 &= \frac{6.8078 + 0.9240 + 1.1219}{9.6428 + 4.7143 + 1.5891} \\
 &= \frac{8.8538}{15.9463} \\
 &= 0.5552
 \end{aligned}$$

$$\begin{aligned}
 u_1 &= \frac{\left(0.804 + \left(0.3220 \frac{2}{2-1}\right)\right) + \left(0.196 + \left(0.4606 \frac{2}{2-1}\right)\right) + \left(0.804 + \left(0.7933 \frac{2}{2-1}\right)\right)}{\left(0.3220 \frac{2}{2-1}\right) + \left(0.4606 \frac{2}{2-1}\right) + \left(0.7933 \frac{2}{2-1}\right)} \\
 &= \frac{2.8350 + 3.7903 + 0.4672}{9.6428 + 4.7143 + 1.5891} \\
 &= \frac{7.0925}{15.9463} \\
 &= 0.4448
 \end{aligned}$$

Berdasarkan hasil perhitungan nilai keanggotaan didapat dua nilai keanggotaan, untuk menentukan kelas target, yang dipilih nilai keanggotaan terbesar yaitu 0.5552, sehingga kelas target diagnosisnya yaitu 0 (negatif diabetes). Jadi, diagnosis yang didapat ialah pasien tersebut tidak terkena diabetes.

3.5 Perancangan Antarmuka

Pada subbab ini akan dijelaskan antarmuka (*interface*) sistem yang dibuat. Tampilan antarmuka terdiri dari 3 form, form prediksi, form koleksi data uji dan form perbandingan akurasi.

3.5.1 Form Prediksi

Pada tahap ini bertujuan untuk pengujian dengan cara menginputkan data parameter secara manual yang kemudian akan diketahui hasil prediksinya. Antarmuka prediksi ditunjukkan pada gambar 3.7.

The screenshot shows a web application interface for diabetes classification. The title bar reads 'Aplikasi Klasifikasi Penyakit Diabetes' with 'Proses' and 'Keluar' buttons. The main content area is titled 'KLASIFIKASI PENYAKIT DIABETES' and includes the subtitle 'Menggunakan Fuzzy K-Nearest Neighbour dan K-Nearest Neighbour'. The interface is organized into several functional areas: a 'Data Training' section with a file selection input and button; an 'Atribut Data' section with multiple input fields for patient history and vitals; a 'Parameter Proses KNN' section with a 'K' value input and 'Proses'/'Reset' buttons; a 'KNN' section; and a 'Hasil Proses' section for displaying the system diagnosis. Red callouts 1 through 7 are placed over various UI elements to indicate their function or location.

Gambar 3.7a *Interface* form prediksi K-NN

.Gambar 3.7b Interface form prediksi FK-NN

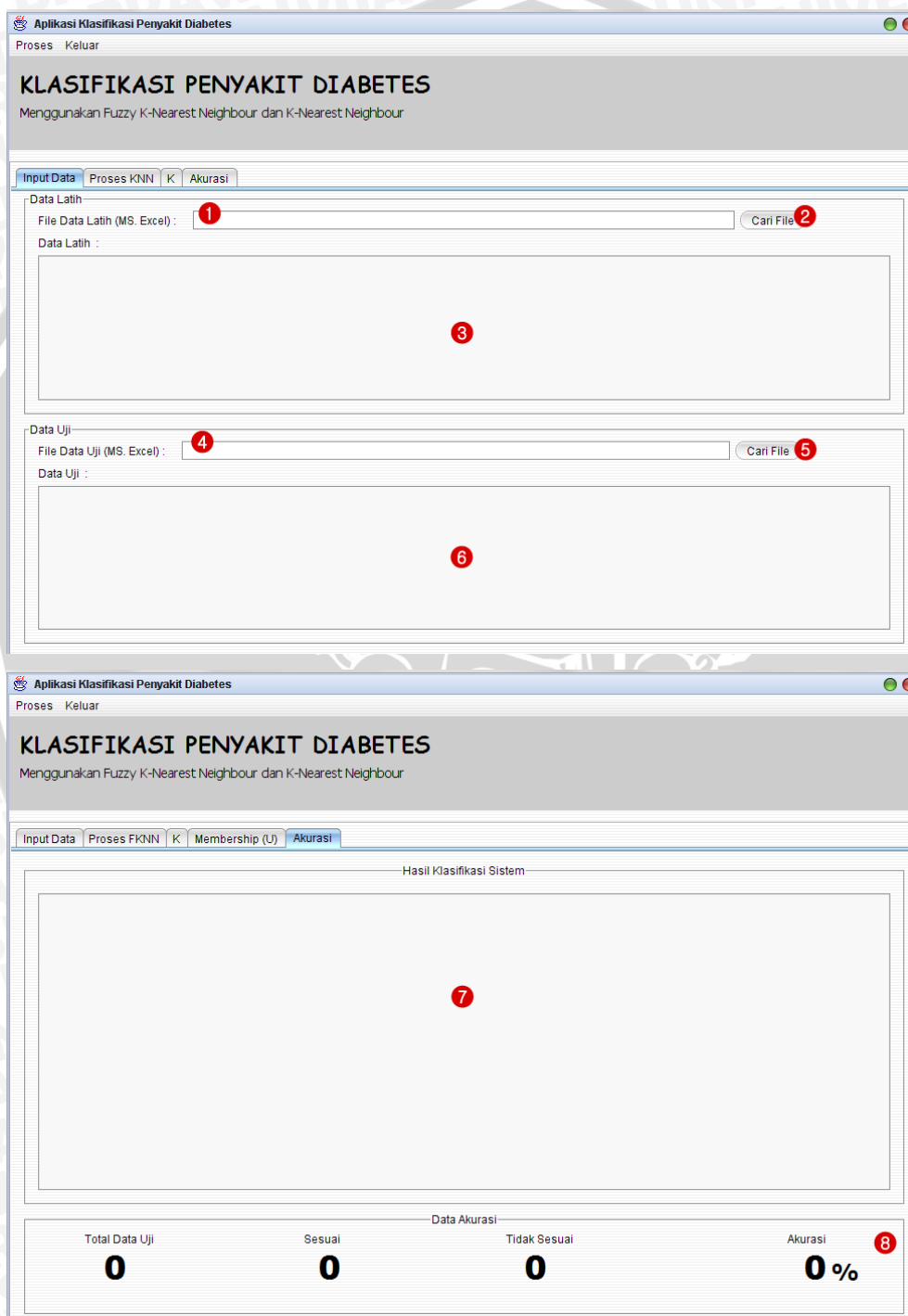
Pada gambar 3.7, *interface* sistem terdiri dari :

1. *Textbox* yang digunakan untuk menginputkan data latih yang berupa file
2. Tombol cari file digunakan untuk mencari data yang akan diinputkan ke dalam sistem
3. *Textbox* yang digunakan untuk menginputkan data uji sesuai dengan atribut pada data latih yaitu sejumlah 8 atribut
4. *Textbox* yang digunakan untuk menginputkan nilai parameter bobot (m) dan k yang dilakukan sebagai parameter proses
5. Tombol proses digunakan untuk memberi perintah pada sistem untuk memulai proses K-NN
6. Tombol *reset* untuk menghapus semua proses yang telah dilakukan sebelumnya
7. *LabelText* yang digunakan untuk menampilkan hasil dari proses klasifikasi K-NN dan fuzzy K-NN.

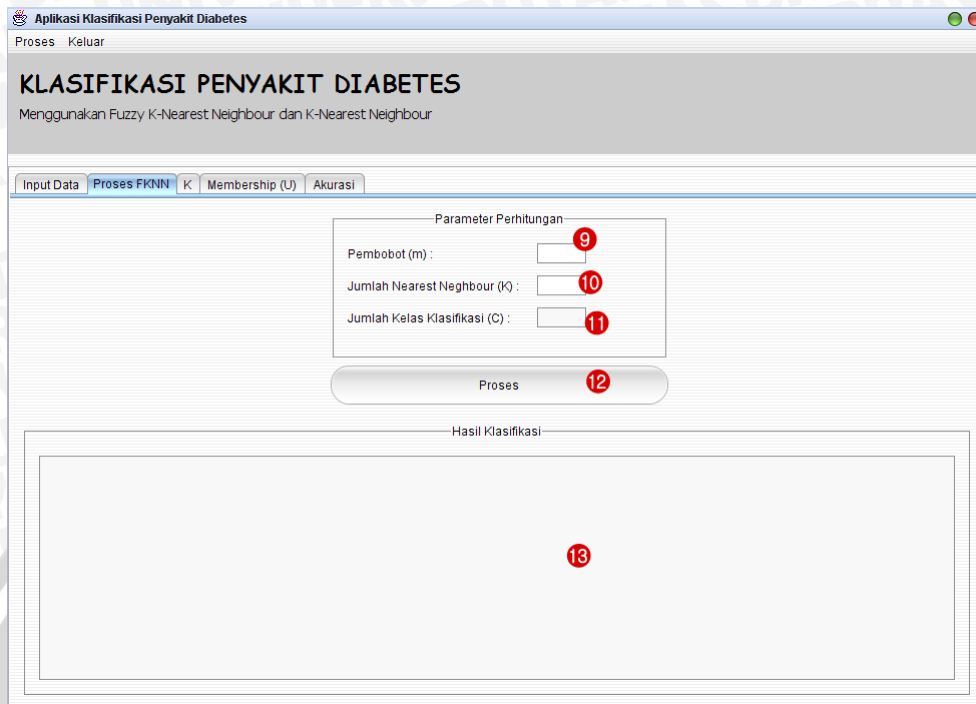
3.5.2 Form koleksi data uji

Tahap ini dilakukan untuk memproses akurasi sistem yang ditentukan oleh parameter nilai k dan bobot (m). Fitur antara K-NN

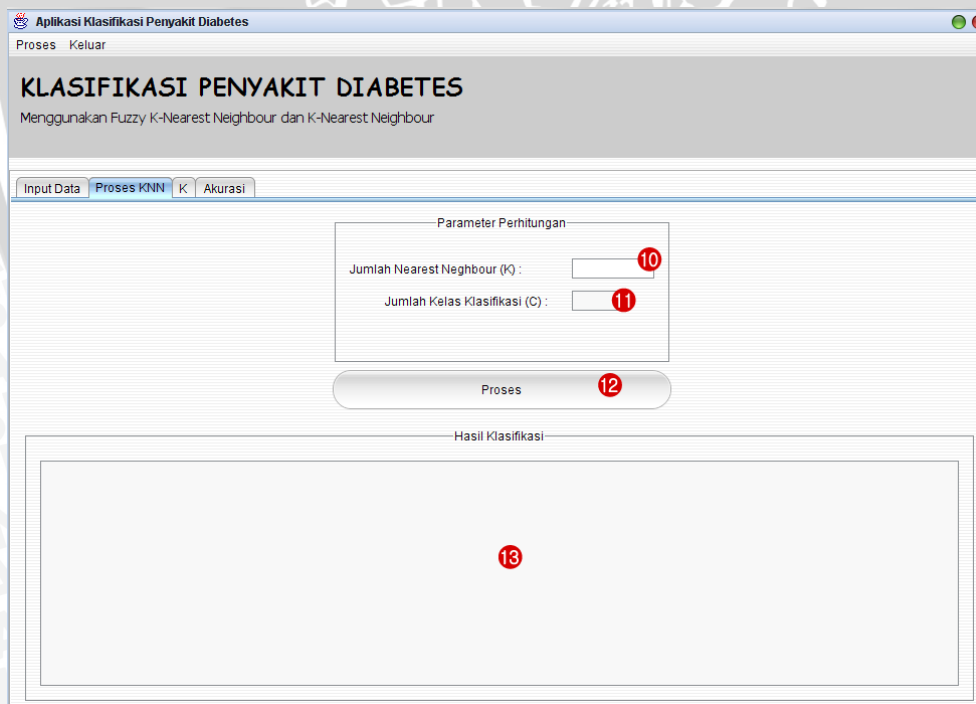
dan FK-NN hampir sama, hanya beda pada tab proses. Untuk K-NN, hanya atribut k yang diinputkan user. Sedangkan pada FK-NN, user menginputkan nilai bobot (m) dan k untuk diproses.



Gambar 3.8a Interface koleksi data uji K-NN dan FK-NN



Gambar 3.8b *Interface* koleksi data uji tab proses FK-NN

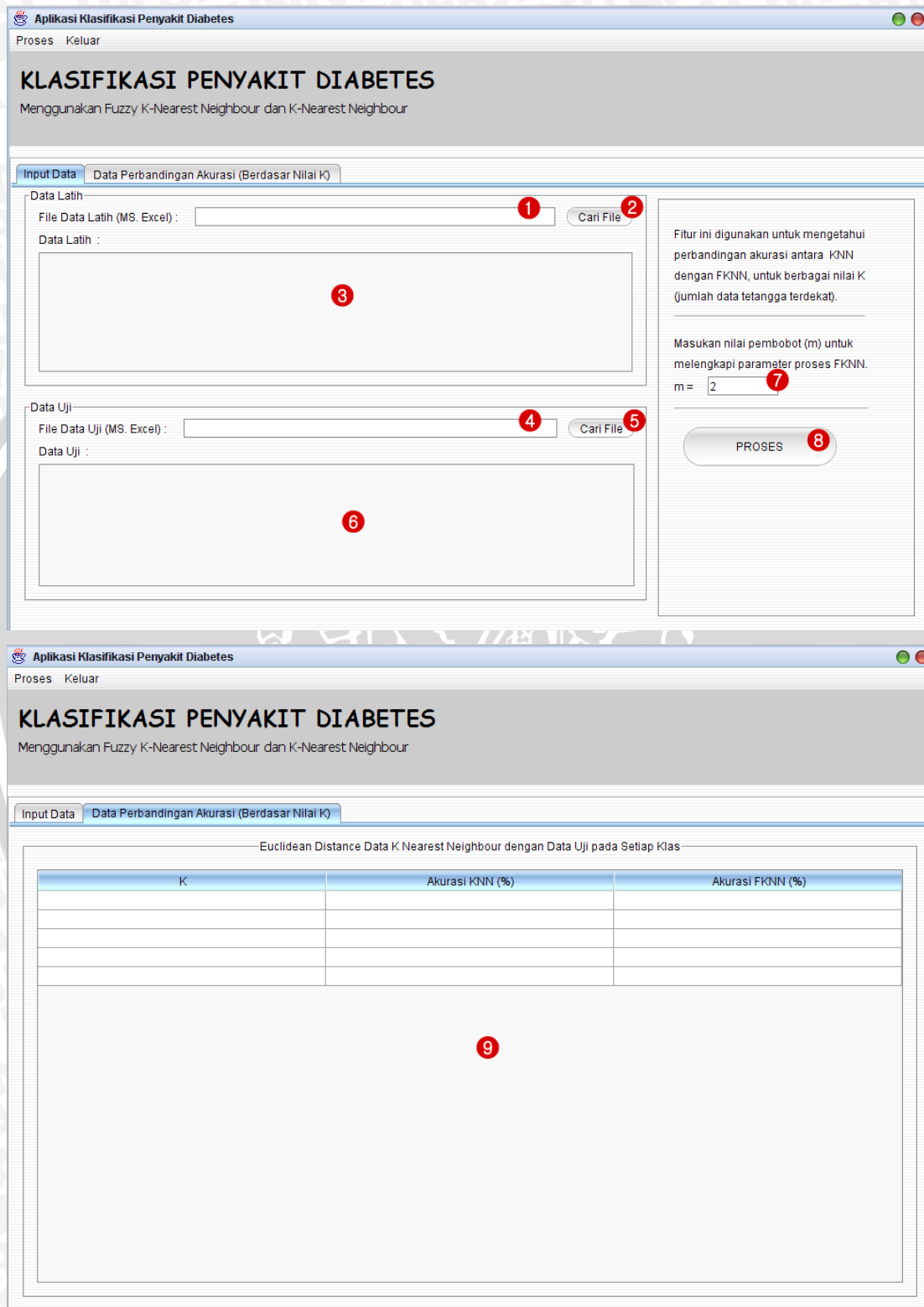


Gambar 3.8c *Interface* koleksi data uji tab proses K-NN

Pada gambar 3.8, *interface* sistem terdiri dari :

1. *Textbox* yang digunakan untuk menginputkan data latih yang berupa file
2. Tombol cari file digunakan untuk mencari data latih yang akan diinputkan ke dalam sistem
3. Tab data latih untuk menampilkan data latih dari *excel* yang digunakan pada sistem
4. *Textbox* yang digunakan untuk menginputkan data uji yang berupa file
5. Tombol cari file digunakan untuk mencari data uji yang akan diinputkan ke dalam sistem
6. Tab data uji untuk menampilkan data uji dari *excel* yang digunakan pada sistem
7. Tab K-NN dan FK-NN yang digunakan untuk menampilkan hasil klasifikasi diagnosa sistem
8. *Textbox* yang digunakan untuk menampilkan total data uji yang digunakan, jumlah data yang benar, jumlah data yang salah dan akurasi yang didapatkan pada sistem
9. *Textbox* yang digunakan untuk menginputkan nilai bobot (m)
10. *Textbox* yang digunakan untuk menginputkan nilai k
11. *Textbox* yang digunakan untuk mengidentifikasi jumlah kelas yang digunakan pada data. Pada bagian ini, sistem sendiri yang akan mengenali berapa jumlah kelas yang digunakan
12. Tombol proses yang digunakan untuk memulai proses K-NN dan FK-NN
13. Tab hasil proses yang digunakan untuk menampilkan semua atribut data uji dan hasil diagnosa sistem

3.5.2 Perbandingan Akurasi



Gambar 3.9 Form perbandingan akurasi K-NN dan F-KNN

Pada gambar 3.9, *interface* sistem terdiri dari :

1. *Textbox* yang digunakan untuk menginputkan data latih yang berupa file
2. Tombol cari file digunakan untuk mencari data latih yang akan diinputkan ke dalam sistem
3. Tab data latih untuk menampilkan data latih dari *excel* yang digunakan pada sistem
4. *Textbox* yang digunakan untuk menginputkan data uji yang berupa file
5. Tombol cari file digunakan untuk mencari data uji yang akan diinputkan ke dalam sistem
6. Tab data uji untuk menampilkan data uji dari *excel* yang digunakan pada sistem
7. *Textbox* yang digunakan untuk menginputkan nilai bobot (m) untuk FK-NN
8. Tombol proses yang digunakan untuk memulai proses akurasi berdasarkan nilai k antara algoritma K-NN dan FK-NN
9. Tampilan akurasi dari K-NN dan F-KNN berdasarkan nilai K . Nilai K ini merupakan sejumlah data latih yang digunakan.

3.6 Perancangan Uji Coba

Setelah sistem selesai dibuat, langkah selanjutnya adalah melakukan pengujian terhadap sistem tersebut. Pengujian ini dilakukan untuk mengetahui tingkat akurasi dari hasil diagnosa menggunakan *K-nearest neighbor* dan *fuzzy K-nearest neighbor*. Pada penelitian ini, terdapat 2 macam yang akan diuji, di antaranya:

1. Pengujian untuk mengetahui pengaruh nilai k (jumlah tetangga terdekat) pada setiap kelas dan jumlah data latih terhadap tingkat akurasi sistem
2. Pengujian untuk mengetahui pengaruh nilai m (bobot) pada proses FK-NN terhadap tingkat akurasi sistem

3.6.1 Uji pengaruh nilai k dan data latih terhadap tingkat akurasi

Pengujian dan analisa hasil akurasi diagnosa Diabetes Melitus digunakan dengan membandingkan hasil diagnosa sistem dengan diagnosa sebenarnya. Nilai akurasi sistem dilakukan dengan menggunakan persamaan (2-8). Dalam penelitian ini terdapat 10 kali uji coba untuk setiap nilai k. Uji coba 1 berarti pengambilan sebanyak n data latih pertama. Kemudian data di random lagi, dan diambil sebanyak n data latih sebagai data uji coba 2, begitu seterusnya hingga uji coba 10. Setelah itu hasil rata-ratanya digunakan untuk analisa. Uji coba dilakukan menggunakan nilai k=1 sampai k=30. Perancangan skenario uji coba sistem ditunjukkan pada tabel 3.8.

Tabel 3.8 Tabel Perancangan Uji Coba

K	Akurasi Sistem (%)										Rata-rata
	Uji Coba 1	Uji Coba 2	Uji Coba 3	Uji Coba 4	Uji Coba 5	Uji Coba 6	Uji Coba 7	Uji Coba 8	Uji Coba 9	Uji Coba 10	
1											
2											
3											
...											
60											

Uji pengaruh jumlah data latih terhadap tingkat akurasi diagnosa sistem dilakukan pada dataset diabetes Indian Pima. Pada pengujian ini, dilakukan pada 5 jenis data latih yang berbeda, yaitu 80, 130, 180, dan 230 data latih dengan data uji yang tetap yaitu 50 data. Perancangan hasil uji coba nilai k dan pengaruh terhadap akurasi sistem ditunjukkan pada tabel 3.9.

Tabel 3.9 Tabel uji pengaruh nilai k dan data latih terhadap akurasi sistem

nilai k	Nilai Akurasi (%) untuk Data Latih			
	80	130	180	230
1				
2				
3				
...				
60				

3.6.2 Uji pengaruh nilai m terhadap tingkat akurasi

Uji pengaruh nilai m (bobot) terhadap tingkat akurasi digunakan untuk mengetahui pengaruh pemberian bobot ketika proses FK-NN dilakukan pada persamaan (2-6).

Tabel 3.10 Tabel uji pengaruh nilai m (bobot) terhadap tingkat akurasi

k	Akurasi sistem (%) berdasarkan bobot (m)						
	m=1	m=2	m=3	m=4	m=5	...	n
1							
2							
3							
4							
5							
...							