

**IMPLEMENTASI DATA *MINING* UNTUK MEMPREDIKSI
KELULUSAN MAHASISWA DENGAN METODE KLASIFIKASI
DAN ALGORITMA K-NEAREST NEIGHBOR BERBASIS
DESKTOP (STUDI KASUS : FAKULTAS TEKNOLOGI
INFORMASI, PROGRAM STUDI TEKNIK INFORMATIKA –
UNIVERSITAS BUDI LUHUR)**

TUGAS AKHIR



**Oleh :
CHRISTIAN YONATHAN SILLUETA
1211501075**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
2016**

**IMPLEMENTASI DATA *MINING* UNTUK MEMPREDIKSI
KELULUSAN MAHASISWA DENGAN METODE KLASIFIKASI
DAN ALGORITMA K-NEAREST NEIGHBOR BERBASIS
DESKTOP (STUDI KASUS : FAKULTAS TEKNOLOGI
INFORMASI, PROGRAM STUDI TEKNIK INFORMATIKA –
UNIVERSITAS BUDI LUHUR)**

TUGAS AKHIR

**Diajukan untuk memenuhi salah satu persyaratan
memperoleh gelar Sarjana Komputer (S.Kom)**



Oleh :

**CHRISTIAN YONATHAN SILLUETA
1211501075**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
2016**



LEMBAR PENGESAHAN

Nama : Christian Yonathan Sillueta
Nomor Induk Mahasiswa : 1211501075
Jenjang Studi : Strata 1
Judul : IMPLEMENTASI DATA *MINING* UNTUK MEMPREDIKSI
KELULUSAN MAHASISWA DENGAN METODE KLASIFIKASI
DAN ALGORITMA K-NEAREST NEIGHBOR BERBASIS
DESKTOP (STUDI KASUS : FAKULTAS TEKNOLOGI
INFORMASI, PROGRAM STUDI TEKNIK INFORMATIKA –
UNIVERSITAS BUDI LUHUR)

Jakarta, 28 Desember 2015
Tim Penguji :

Tanda tangan :

Ketua,
M. Syafrullah, M.Kom, M.Sc

Anggota,
Purwanto, S.Si, M.Kom

Pembimbing,
Muhammad Ainur Rony, S.Kom, M.T.I

Ketua Program Studi

Muhammad Ainur Rony, S.Kom, M.T.I

ABSTRAK

NIM : 1211501075
Nama : Christian Yonathan Sillueta
Judul : IMPLEMENTASI DATA *MINING* UNTUK MEMPREDIKSI KELULUSAN MAHASISWA DENGAN METODE KLASIFIKASI DAN ALGORITMA K-NEAREST NEIGHBOR BERBASIS DESKTOP (STUDI KASUS : FAKULTAS TEKNOLOGI INFORMASI, PROGRAM STUDI TEKNIK INFORMATIKA – UNIVERSITAS BUDI LUHUR)

Saat ini Universitas Budi Luhur (UBL) berada dalam lingkungan yang sangat kompetitif, sehingga Fakultas Teknologi Informasi program studi Teknik Informatika - UBL berusaha untuk terus memperbaiki manajemen, yang berfungsi meningkatkan mutu pendidikan dan akreditasi, salah satu elemen penilaian akreditasi perguruan tinggi adalah mahasiswa yang lulus tepat waktu, dengan kriterianya : lulus cepat, lulus tepat, lulus terlambat, dan tidak lulus atau drop out. Berdasarkan data dari Forlap Dikti, UBL memiliki jumlah mahasiswa aktif sebanyak 15348. dimana 60% diantaranya merupakan mahasiswa (FTI), dimana pada program studi Teknik Informatika (TI) mempunyai mahasiswa aktif sebanyak 2720. Dari jumlah tersebut menunjukkan bahwa prodi TI-UBL merupakan salah satu program studi favorit bagi mahasiswa, hal ini juga ditunjukkan dengan jumlah mahasiswa yang diterima tiap tahun sekitar 450-550 mahasiswa. Dengan makin banyaknya jumlah mahasiswa aktif di program studi TI-UBL, maka manajemen membutuhkan aplikasi yang dapat memprediksi kelulusan mahasiswanya. Dengan makin banyaknya jumlah mahasiswa aktif di program studi TI-UBL, maka manajemen membutuhkan aplikasi yang dapat memprediksi kelulusan mahasiswanya. Pada penelitian ini, analisis yang digunakan merupakan analisis prediksi dengan metode Klasifikasi dan Algoritma K-Nearest Neighbor. Atribut yang paling penting pada penelitian ini meliputi nilai IPS mahasiswa 6 (enam) semester pertama. Aplikasi ini dibuat dengan bahasa pemrograman Java berbasis desktop menggunakan editor Netbeans. Setelah melakukan pengujian terhadap aplikasi yang sudah dibuat, didapatkan hasil yaitu pada pengujian tunggal memiliki akurasi tertinggi sebesar 70% dengan memiliki rata-rata 61.11%, lalu pada pengujian jamak memiliki akurasi tertinggi sebesar 75.36% dengan memiliki rata-rata 61.88%. Aplikasi mampu memprediksi kelulusan mahasiswa FTI-TI UBL dengan baik secara tunggal maupun jamak. Hasil prediksi kelulusan dari penelitian ini dapat membantu program studi untuk mengetahui kelulusan mahasiswa, sehingga bisa menjadi rekomendasi pengambilan mata kuliah bagi mahasiswa untuk semester berikutnya seperti skripsi dan kkp.

Kata kunci : Universitas Budi Luhur, Fakultas Teknologi Informasi – Program Studi Teknik Informatika, Data *Mining*, Algoritma K-NEAREST NEIGHBOR, Kelulusan Mahasiswa.

iii+53 Hal; 11 Tabel; 28 Gambar; 2 Lampiran

SURAT PERNYATAAN TIDAK PLAGIAT DAN PERSETUJUAN PUBLIKASI

Saya yang bertanda tangan dibawah ini :

Nama : Christian Yonathan Silweta
Nim : 1211501075
Program Studi : Teknik Informatika
Fakultas : Fakultas Teknologi Informasi

Menyatakan bahwa TUGAS AKHIR yang berjudul :

Implementasi Data Mining Untuk Memprediksi Kejurusan Mahasiswa dengan Metode Klasifikasi dan Algoritma K-Nearest Neighbor Berbasis Desktop (Studi Kasus : Fakultas Teknologi Informasi, Program Studi Teknik Informatika - Universitas Budi Luhur)

1. Merupakan hasil karya tulis ilmiah sendiri dan bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain,
2. Saya ijin untuk dikelola oleh Universitas Budi Luhur sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai aturan yang berlaku apabila dikemudian hari pernyataan ini tidak benar.

Jakarta, 10 Desember 2015



Christian Yonathan Silweta

KATA PENGANTAR

Puji Tuhan, Segala puji dan syukur penulis panjatkan kehadirat Tuhan Yesus Kristus yang telah memberikan hikmat dan akal sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Adapun skripsi ini disusun untuk memenuhi persyaratan dalam menyelesaikan tingkat pendidikan Strata Satu (S1) pada Program Studi Teknologi Informatika, Fakultas Teknologi Informasi Universitas Budi Luhur.

Dalam Kesempataan ini, penulis ingin mengucapkan terima kasih kepada semua pihak yang telah membimbing dan memberikan masukan berupa kritik serta saran kepada penulis dalam penyusunan tugas akhir ini, oleh karena itu, penulis menyampaikan rasa hormat dan ungkapan terima kasih kepada :

1. Tuhan Yesus Kristus, atas segala Petunjuk dan Hikmat sehingga pada akhirnya penulis dapat menyelesaikan tugas akhir ini.
2. Kepada Papa, Mama, Mas Arthur, Mas Gerald, Mbak Irene, Mbak Priska, Keponakan Kevin, dan Tante Ari terima kasih atas segala dukungan baik moral dan moril.
3. Bapak Prof. Ir. Suryo Hapsoro Tri Utomo, Ph.D selaku Rektor Universitas Budi Luhur.
4. Bapak Goenawan Brotosaputro, S.Kom, M.Sc. selaku Dekan Fakultas Teknologi Informasi, Universitas Budi Luhur.
5. Bapak Muhammad Ainur Rony, S.Kom, M.T.I., selaku Ketua Program Studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Budi Luhur dan Dosen Pembimbing yang selalu memberi penulis arahan dan motivasi.
6. Bapak Humisar Hasugian, M.Kom, selaku Kepala LAB ICT Terpadu Universitas Budi Luhur.
7. Kepada teman-teman Asisten Laboratorium Komputer 2012 : "Agung, Alim, Andrew, Devi, Diah, Diana, Fitri, Irfan, Kory, Nanang, Riri, Riyan, Angga, dan Yesih". Terima kasih atas dukungan, perhatian, dan semangatnya.
8. Kepada teman-teman KUTI 2012, terima kasih atas dukungan, perhatian, dan semangatnya.
9. Kepada teman-teman RPL 2012, terima kasih atas dukungan, perhatian, dan semangatnya.
10. Kepada teman-teman Asisten 2013 dan 2014 dan 2015 terima kasih atas dukungannya, Semoga kita semua tetap solid sebagai satu keluarga besar labkom.
11. Kepada semua alumni Asisten labkom UBL. Terima kasih atas segala perhatian kalian.
12. Semua pihak yang terkait, baik secara langsung maupun tidak langsung dalam penulisan laporan TA ini dari tahap awal sampai dengan tahap akhir penyusunan laporan TA ini.

Semoga Tuhan Yang Maha Esa melimpahkan karunia serta anugrah-Nya atas segala bantuan yang telah diberikan. Amin.

Penulis sadar bahwa penyusunan tugas akhir ini masih jauh dari sempurna, oleh karena itu penulis menerima kritik dan saran demi sempurnanya Tugas Akhir ini.

Akhir kata penulis memohon maaf atas segala kekurangan yang ada pada tugas akhir ini. Semoga tulisan ini dapat bermanfaat bagi semua pihak

Jakarta, September 2015

Penulis

DAFTAR TABEL

	Halaman
Tabel 2.5.1 Data Matriks.....	8
Tabel 2.5.2 Data Transaksi	8
Tabel 2.5.3 Data Istilah Dokumen.....	8
Tabel 2.8.1 Tipe Fitur	11
Tabel 2.10.1 Penelitian Terkait	14
Tabel 3.5.1 Penggunaan Data	19
Tabel 3.5.2 Tabel Transformasi Keterangan Lulus.....	20
Tabel 3.5.3 Tabel Transformasi Jenis Kelamin	20
Tabel 4.2.1 Pengujian Tunggal	47
Tabel 4.2.2 Pengujian Tunggal Lanjutan.....	48
Tabel 4.2.3 Pengujian Jamak	48

DAFTAR GAMBAR

	Halaman
Gambar 2.2.1 Akar Ilmu Data <i>Mining</i>	4
Gambar 2.2.2 <i>Business Intelligence</i>	5
Gambar 2.5.1 Contoh Data Grafik	9
Gambar 2.5.2 Prakiraan Musim Kemarau	9
Gambar 2.6.1 Proses Pekerjaan Klasifikasi	10
Gambar 2.8.1 Ilustrasi Kedekatan Kasus.....	13
Gambar 3.6.1 Arsitektur Aplikasi Usulan	21
Gambar 3.7.1 Rancangan Layar <i>Form</i> Halaman Utama.....	21
Gambar 3.7.2 Rancangan Layar <i>Form</i> Tunggal	22
Gambar 3.7.3 Rancangan Layar <i>Form</i> Jamak	23
Gambar 3.7.4 Rancangan Layar <i>Form</i> Tunggal Dengan <i>Message Box</i>	24
Gambar 3.7.4 Rancangan Layar <i>Form</i> Jamak Dengan <i>Message Box</i>	25
Gambar 3.8.1 <i>Flowchart</i> K-Nearest Neighbor.....	26
Gambar 3.8.2 <i>Flowchart</i> Menghitung Jarak <i>Euclidian</i>	27
Gambar 3.8.3 <i>Flowchart Form</i> Halaman Utama	28
Gambar 3.8.4 <i>Flowchart Form</i> Tunggal.....	29
Gambar 3.8.5 <i>Flowchart Form</i> Jamak	30
Gambar 3.8.6 <i>Flowchart</i> Proses Pada <i>Form</i> Tunggal	31
Gambar 3.8.7 <i>Flowchart</i> Proses Menyimpan Nilai <i>Training</i> Pada <i>Form</i> Tunggal.....	32
Gambar 3.8.8 <i>Flowchart</i> Proses Pada <i>Form</i> Jamak.....	33
Gambar 3.8.9 <i>Flowchart</i> Proses Menyimpan Nilai <i>Training</i> Pada <i>Form</i> Jamak	34
Gambar 4.1.1 Tampilan Layar <i>Form</i> Utama.....	40
Gambar 4.1.2 Tampilan Layar <i>Form</i> Tunggal.....	41
Gambar 4.1.3 Tampilan Layar <i>Form</i> Jamak	42
Gambar 4.1.4 Implementasi <i>Form</i> Tunggal	43
Gambar 4.1.5 Implementasi <i>Form</i> Jamak.....	44
Gambar 4.1.6 Implementasi <i>Form</i> Tunggal Dengan <i>Message Box</i>	45
Gambar 4.1.7 Implementasi <i>Form</i> Jamak Dengan <i>Message Box</i>	46

DAFTAR SIMBOL

1. Flowchart



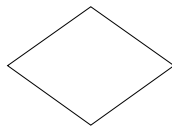
Terminator

Digunakan untuk menggambarkan kegiatan awal atau akhir suatu proses



Input/Output

Digunakan untuk menggambarkan suatu kegiatan masukan ataupun keluaran



Decision

Digunakan untuk mengambil suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu



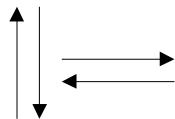
Process

Digunakan untuk menggambarkan suatu kegiatan proses penghubung



Predefined Process

Digunakan untuk modul yang tidak ditulis, karena sudah ada dalam sistem yang menggambarkan suatu proses



Line Connector

Digunakan untuk menghubungkan satu simbol dengan simbol lainnya yang menyatakan alur proses

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	ii
ABSTRAK	iii
SURAT PERNYATAAN TIDAK PLAGIAT DAN PERSETUJUAN PUBLIKASI.....	iv
KATA PENGANTAR.....	v
DAFTAR TABEL.....	vi
DAFTAR SIMBOL	viii
DAFTAR ISI.....	ix
BAB I PENDAHULUAN.....	1
1. Latar Belakang	1
2. Masalah	1
3. Tujuan Penulisan.....	1
4. Batasan Permasalahan.....	2
5. Metodologi Penyelesaian Masalah.....	2
a. Identifikasi Masalah	2
b. Observasi	2
c. Metode Pengembangan.....	2
1) Analisis	2
2) Desain	2
3) Implementasi	2
4) Perawatan.....	3
6. Sistematika Penulisan	3
BAB II LANDASAN TEORI	4
1. Pendahuluan.....	4
2. Data <i>Mining</i>	4
a. Perbedaan Data <i>Mining</i> Dengan Data <i>Warehouse</i>	4
3. Pekerjaan Dalam Data <i>Mining</i>	5
a. Model Prediksi (Prediction Modelling).....	5
b. Analisis Cluster (<i>Cluster Analysis</i>)	6
c. Analisis Asosiasi (<i>Association Analysis</i>).....	6
d. Deteksi Anomali (<i>Anomaly Detection</i>)	6
4. Proses Data Mining.....	6
a. Eksplorasi/pemrosesan awal data	6
b. Membangun model dan melakukan validasi terhadapnya	7
c. Penerapan	7
5. Set Data.....	7
6. Metode Klasifikasi.....	9
7. Tipe Fitur	11
8. Konsep K-Nearest Neighbor	12
9. Algoritma K-Nearest Neighbor.....	12
10. Penelitian Terkait	14
BAB III ANALISA MASALAH DAN RANCANGAN PROGRAM.....	17
1. Pendahuluan.....	17
2. Analisa Perangkat.....	17
a. Perangkat Keras	17
b. Perangkat Lunak	17
3. Permasalahan Dan Strategi Pemecahan Masalah.....	17
a. Analisa Masalah.....	17

b. Strategi Pemecahan Masalah	18
4. Deskripsi Proses	18
a. Proses Halaman Utama	18
b. Proses Tunggol	18
c. Proses Jamak.....	19
5. Analisa Data <i>Mining</i>	19
a. Sumber Data.....	19
b. Data Yang Digunakan	19
c. Proses Data Dalam Data <i>Mining</i>	19
d. Analisa Data	20
6. Analisa Aplikasi	20
a. Analisa Aplikasi Usulan	20
b. Sistem Kerja Aplikasi Usulan	21
7. Rancangan Layar.....	21
a. Rancangan Layar <i>Form</i> Halaman Utama.....	21
b. Rancangan Layar <i>Form</i> Tunggol	22
c. Rancangan Layar <i>Form</i> Jamak	23
d. Rancangan Layar <i>Form</i> Tunggol Dengan <i>Message Box</i>	24
e. Rancangan Layar <i>Form</i> Jamak Dengan <i>Message Box</i>	25
8. <i>Flowchart</i> Dan Algoritma	26
a. <i>Flowchart</i>	26
1) <i>Flowchart</i> K-Nearest Neighbor	26
2) <i>Flowchart</i> Menghitung Jarak <i>Euclidean</i>	27
3) <i>Flowchart Form</i> Halaman Utama.....	28
4) <i>Flowchart Form</i> Tunggol	29
5) <i>Flowchart Form</i> Jamak.....	30
6) <i>Flowchart</i> Proses Pada <i>Form</i> Tunggol	31
7) <i>Flowchart</i> Proses Menyimpan Nilai Training Pada <i>Form</i> Tunggol	32
8) <i>Flowchart</i> Proses Pada <i>Form</i> Jamak	33
9) <i>Flowchart</i> Proses Menyimpan Nilai Training Pada <i>Form</i> Jamak.....	34
b. Algoritma.....	35
1) Algoritma KNN.....	35
2) Algoritma Menghitung Jarak <i>Euclidian</i>	35
3) Algoritma <i>Form</i> Halaman Utama	35
4) Algoritma <i>Form</i> Tunggol	35
5) Algoritma <i>Form</i> Jamak.....	37
6) Algoritma Proses Pada <i>Form</i> Tunggol	38
7) Algoritma Proses Menyimpan Nilai Training Pada <i>Form</i> Tunggol.....	38
8) Algoritma Proses Pada <i>Form</i> Jamak.....	39
9) Algoritma Proses Menyimpan Nilai Training Pada <i>Form</i> Jamak.....	39
BAB IV IMPLEMENTASI DAN UJI COBA PROGRAM.....	40
1. Tampilan Layar dan Implementasi.....	40
a. Tampilan Layar	40
1) Tampilan Layar <i>Form</i> Utama	40
2) Tampilan Layar <i>Form</i> Tunggol	41
3) Tampilan Layar <i>Form</i> Jamak.....	42
b. Implementasi.....	43
1) Implementasi <i>Form</i> Tunggol.....	43
2) Implementasi <i>Form</i> Jamak	44

3) Implementasi <i>Form</i> Tunggal Dengan <i>Message Box</i>	45
4) Implementasi <i>Form</i> Jamak Dengan <i>Message Box</i>	46
2. Hasil Uji Program	47
a. Pengujian Tunggal.....	47
b. Pengujian Jamak	48
3. Evaluasi Program.....	49
a. Kelebihan Program	49
b. Kekurangan Program	49
BAB V PENUTUP	50
1. Kesimpulan.....	50
2. Saran.....	51
DAFTAR PUSTAKA	52
LAMPIRAN-LAMPIRAN	53

BAB I PENDAHULUAN

1. Latar Belakang

Saat ini Universitas Budi Luhur (UBL) berada dalam lingkungan yang sangat kompetitif, sehingga Fakultas Teknologi Informasi program studi Teknik Informatika - UBL berusaha untuk terus memperbaiki manajemen, yang berfungsi meningkatkan mutu pendidikan dan akreditasi, salah satu elemen penilaian akreditasi perguruan tinggi adalah mahasiswa yang lulus tepat waktu, dengan kriterianya : lulus cepat, lulus tepat, lulus terlambat, dan tidak lulus atau drop out.

Berdasarkan data dari Forlap Dikti, UBL memiliki jumlah mahasiswa aktif sebanyak 15348. dimana 60% diantaranya merupakan mahasiswa (FTI), dimana pada program studi Teknik Informatika (TI) mempunyai mahasiswa aktif sebanyak 2720. Dari jumlah tersebut menunjukkan bahwa prodi TI-UBL merupakan salah satu program studi favorit bagi mahasiswa, hal ini juga ditunjukkan dengan jumlah mahasiswa yang diterima tiap tahun sekitar 450-550 mahasiswa. Dengan makin banyaknya jumlah mahasiswa aktif di program studi TI-UBL, maka manajemen membutuhkan aplikasi yang dapat memprediksi kelulusan mahasiswanya.

Salah satu cabang ilmu komputer yang bisa melakukan prediksi adalah data *mining*, penggunaan ilmu *data mining* dapat digunakan sebagai pertimbangan dalam mengambil keputusan lebih lanjut tentang faktor yang mempengaruhi kelulusan khususnya faktor dalam data induk mahasiswa (Huda 2010). Pada penelitian sebelumnya, bahwa faktor yang paling berpengaruh dalam penentuan klasifikasi kinerja akademik mahasiswa adalah Indeks Prestasi Kumulatif (IPK), Indeks Prestasi Semester (IPS) 1 (satu) hingga 4 (empat) dan Jenis Kelamin (Ridwan dkk. 2013).

Pada tugas akhir ini, penulis ingin memprediksi kelulusan mahasiswa menggunakan data *mining* dengan metode klasifikasi dan algoritma K-Nearest Neighbor dan berbasis desktop sehingga bisa diketahui kriteria kelulusan mahasiswa.

2. Masalah

Dengan melihat ulasan dari latar belakang di atas, maka didapatkan beberapa permasalahan sebagai berikut :

- a. Atribut pada mahasiswa apa saja yang digunakan?
- b. Bagaimana proses utama dalam data *mining*?
- c. Berapa data yang digunakan dari total keseluruhan data?
- d. Berapa akurasi hasil prediksi?

3. Tujuan Penulisan

Adapun tujuan penulisan dari penelitian ini adalah :

- a. Dapat memahami Algoritma K-NEAREST NEIGHBOR dengan menggunakan rumus perhitungan jarak *Euclidian* dan menerapkan pada bahasa pemrograman Java.
- b. Mampu memprediksi mengenai kelulusan Mahasiswa Fakultas Teknologi Informasi, Program Studi Teknik Informatika – Universitas Budi Luhur, apakah bernilai lulus cepat, lulus tepat, lulus terlambat, dan tidak lulus atau DO.

4. Batasan Permasalahan

Untuk mencapai tujuan penelitian di atas, penulis memfokuskan batasan penelitian sebagai berikut :

- a. Data yang digunakan merupakan data mahasiswa Fakultas Teknologi Informasi, Program Studi Teknik informatika pada kelas Reguler di kampus Pusat.
- b. Atribut pada mahasiswa yang digunakan yaitu NIM, Nama, IPS 1, IPS 2, IPS 3, IPS 4, IPS 5, IPS 6, jumlah SKS hingga semester 6, dan Keterangan Lulus.
- c. Input file pada Data *training* dan *testing*, hanya untuk ekstensi .xls, atau .xlsx.
- d. Data mahasiswa yang digunakan haruslah sudah mempunyai nilai IPS dari 1 hingga 6.
- e. Format isi pada berkas *file excel* harus dengan urutan No, NIM, Nama, Jenis Kelamin, IPS 1, IPS 2, IPS 3, IPS 4, IPS 5, IPS 6, Jumlah SKS, dan Keterangan Lulus.
- f. Menggunakan Algoritma K-Nearest Neighbor dan untuk perhitungan jarak *Euclidian*.
- g. Menggunakan bahasa pemrograman Java.

5. Metodologi Penyelesaian Masalah

Beberapa metode pengembangan yang digunakan dalam mengembangkan aplikasi prediksi kelulusan mahasiswa yaitu meliputi :

a. Identifikasi Masalah

Fakultas Teknologi Informasi mempunyai 3 (tiga) program studi yaitu : Teknik Informatika (TI), Sistem Informasi (SI), dan Sistem Komputer (SK). Pada ketiga program studi tersebut, mahasiswa paling banyak berada pada program studi Teknik Informatika (TI). Pada Tugas Akhir ini penulis ingin memprediksi kelulusan mahasiswa berdasarkan parameter yang sudah dijelaskan sebelumnya pada batasan masalah.

b. Observasi

Meminta data mahasiswa Fakultas Teknologi Informasi, Program Studi Teknik Informatika Universitas Budi Luhur kepada Bapak Muhammad Ainur Rony. Meminta bantuan dari Bapak Muhammad Ainur Rony, untuk membaca data mahasiswa untuk info keterangan lulus mahasiswa berdasarkan tanggal teori dan beberapa parameter lainnya.

c. Metode Pengembangan

Dalam pengembangan aplikasi ini merujuk pada *the waterfall model*. Ada empat tahap utama dalam metode ini. Disebut *waterfall* karena *diagram* tahapan prosesnya mirip dengan air terjun yang bertingkat.

Tahapan dalam *The Waterfall* model secara ringkas menurut Pressman, 1997, adalah sebagai berikut :

1) Analisis

Tahapan analisis bertujuan untuk mencari kebutuhan pengguna dan organisasi serta menganalisa kondisi yang ada (sebelum diterapkan sistem yang baru).

2) Desain

Tahapan desain bertujuan untuk menentukan spesifikasi detail dari komponen-komponen program (manusia, perangkat keras, perangkat lunak, jaringan dan data) dan produk-produk informasi yang sesuai dengan hasil tahap analisis.

3) Implementasi

Tahapan ini merupakan tahapan untuk mendapatkan atau mengembangkan perangkat keras dan perangkat lunak (pengkodean program), melakukan pengujian, pelatihan dan perpindahan ke sistem yang baru.

4) **Perawatan**

Tahapan ini dilakukan ketika sistem sudah dioperasikan. Pada tahapan perawatan ini dilakukan proses monitoring, evaluasi dan perubahan bila diperlukan.

6. Sistematika Penulisan

Adapun sistematika penulisan Tugas Akhir ini dibagi ke dalam lima bab yang dapat dijelaskan sebagai berikut :

BAB I : PENDAHULUAN

Bab ini menjelaskan serta membahas mengenai latar belakang penelitian, masalah yang menjelaskan uraian singkat tentang masalah atau studi kasus yang akan dibahas secara detail pada bab IV, tujuan penulisan yang menggambarkan tentang apa saja yang ingin dicapai dalam penulisan laporan, batasan masalah yang menjelaskan tentang lingkup permasalahan yang dibahas dan batasan penyelesaian yang dilakukan data dan sistematika penulisan yang menjelaskan uraian singkat per babnya.

BAB II : LANDASAN TEORI

Bab ini menjelaskan mengenai teori-teori dan teknik-teknik untuk menunjang penulisan skripsi. Adapun yang dibahas dalam bab ini adalah teori yang berkaitan tentang metode Klasifikasi dan Algoritma K-NEAREST NEIGHBOR

BAB III : ANALISA MASALAH DAN RANCANGAN PROGRAM

Bab ini menjelaskan tentang analisa dan perancangan program, *Flowchart* dan algoritma.

BAB IV : IMPLEMENTASI DAN UJI COBA PROGRAM

Bab ini menjelaskan mengenai jalannya hasil uji coba program sistem yang dirancang, juga menjalankan tata cara uji coba sistem yang dirancang.

BAB V : PENUTUP

Bab ini menguraikan tentang kesimpulan yang didapat dari hasil analisa bab – bab sebelumnya, dan saran yang dapat berguna bagi pengembangan sistem ini selanjutnya.

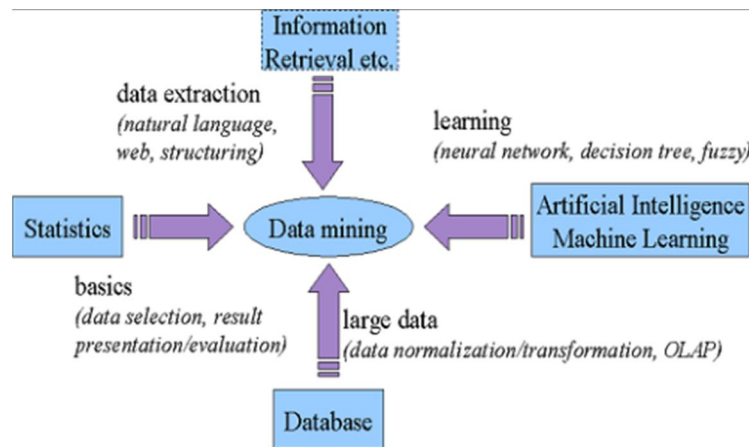
BAB II LANDASAN TEORI

1. Pendahuluan

Pada bab II ini, penulis akan menjelaskan mengenai teori, teknik dan juga definisi dari subyek penelitian yaitu mengenai teori umum tentang Data *Mining*, K-NEAREST NEIGHBOR, dan konsep klasifikasi, serta pekerjaan-pekerjaan yang berhubungan dengan penelitian yang dilakukan oleh penulis.

2. Data *Mining*

Nama data *Mining* sebenarnya mulai dikenal sejak tahun 1990, ketika pekerjaan pemanfaatan data menjadi sesuatu yang penting dalam berbagai bidang, mulai dari bidang akademik, bisnis, hingga medis (Gorunescu 2011). Data mining bisa diterapkan pada berbagai bidang yang mempunyai sejumlah data, tetapi karena wilayah penelitian dengan sejarah yang belum lama, dan belum melewati masa 'remaja', maka data *mining* masih menjadi perdebatan pada posisi bidang pengetahuan yang memilikinya. Maka, Daryl Pregibon menyakan bahwa "data *mining* adalah campuran dari statistik, kecerdasan buatan, dan riset basis data" yang masih berkembang (Gorunescu 2011).

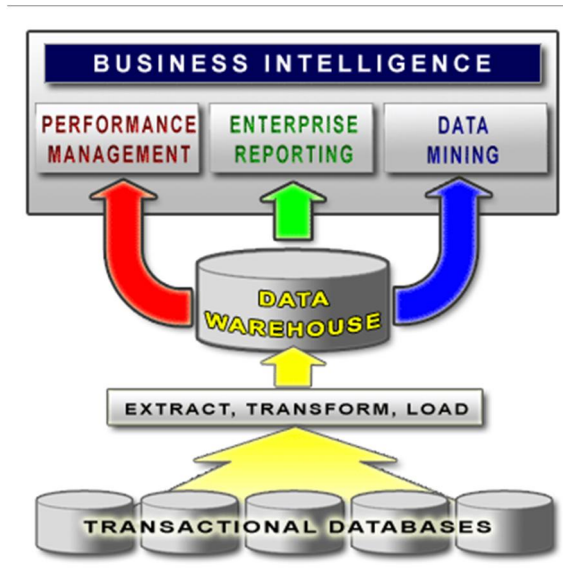


Gambar 2.2.1 Akar Ilmu Data *Mining*

(Sumber : <https://rezqiwati.wordpress.com/2009/04/24/data-mining-%E2%80%93-proses-tahapan-dan-penerapannya/>)

Jadi dapat disimpulkan bahwa data *mining* adalah serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak diketahui secara manual dari suatu *database*. Informasi yang dihasilkan diperoleh dengan cara mengekstraksi dan mengenali pola yang atau menarik data yang terdapat dalam *database*.

a. Perbedaan Data *Mining* Dengan Data *Warehouse*



Gambar 2.2.2 Business Intelligence

(Sumber : <http://elisa.ugm.ac.id/community/show/data-mining-dan-bisnis-intelijen-edi-winarko/>)

Dari gambar 2.2.2 dapat dilihat bahwa data *mining* adalah pengolahan data yang berasal dari Data Warehouse bersama dengan bidang yang menangani masalah pelaporan dan manajemen data. Sementara, data warehouse sendiri bertugas untuk menarik atau melakukan *query* data dari basis data mentah untuk memberikan hasil data yang nantinya digunakan oleh bidang yang menangani manajemen, pelaporan, dan data *mining*. Dengan data *mining* inilah, penggalian informasi baru dapat dilakukan dengan bekal data mentah yang diberikan oleh data warehouse. Hasil yang diberikan oleh ketiga bidang tersebut berguna untuk mendukung aktivitas bisnis cerdas (*business intelligence*).

3. Pekerjaan Dalam Data Mining

Pekerjaan yang berkaitan dengan data *mining* dapat dibagi menjadi 4 (empat) kelompok yaitu : model prediksi (*prediction modeling*), analisis cluster (*cluster analysis*), analisis asosiasi (*association analysis*), dan deteksi anomaly (*anomaly detection*).

a. Model Prediksi (Prediction Modelling)

Model Prediksi merupakan pekerjaan yang berkaitan dengan pembuatan suatu model yang dapat melakukan pemetaan dari setiap himpunan variable ke setiap targetnya, kemudian menggunakan model yang sudah ada tersebut untuk memberikan nilai target pada himpunan baru yang didapat. ada 2 jenis model prediksi, yaitu Klasifikasi dan Regresi. Klasifikasi digunakan untuk variabel target yang sifatnya diskret, sedangkan regresi digunakan untuk variabel target yang sifatnya kontinyu.

Contoh pekerjaan yang menggunakan jenis Klasifikasi adalah melakukan pendeteksian jenis penyakit pasien berdasarkan sejumlah nilai-nilai parameter penyakit yang diderita sebagai masukan. Pekerjaan ini termasuk jenis klasifikasi karena target yang diharapkan adalah bernilai diskret, hanya ada beberapa jenis kemungkinan nilai target yang didapatkan dan tidak ada nilai seri waktu (*time series*) yang harus didapatkan untuk mendapat target nilai akhir. Sementara

melakukan prediksi jumlah penjualan yang didapatkan pada 3 bulan ke depan itu termasuk jenis Regresi karena untuk mendapatkan nilai penjualan bulan ketiga harus mendapatkan nilai penjualan bulan pertama. Dalam hal ini ada seri waktu yang harus dihitung untuk sampai pada target akhir yang diinginkan dan ada nilai kontinu yang harus dihitung untuk mendapatkan nilai target akhir yang diinginkan.

b. Analisis Cluster (*Cluster Analysis*)

Contoh pekerjaan yang berkaitan dengan analisis cluster adalah bagaimana bisa mengetahui pola pembelian barang oleh konsumen pada waktu-waktu tertentu. Dengan mengetahui pola kelompok pembelian tersebut, maka perusahaan dapat menentukan jadwal promosi yang dapat diberikan sehingga dapat membantu meningkatkan omzet penjualan.

Analisis kelompok melakukan pengelompokan data ke dalam sejumlah kelompok berdasarkan kesamaan karakteristik masing-masing data pada kelompok-kelompok yang ada. Data yang masuk dalam batas kesamaan dengan kelompoknya akan bergabung dalam kelompok tersebut, dan akan terpisah dalam kelompok yang berbeda jika keluar dari batas kesamaan kelompok tersebut.

c. Analisis Asosiasi (*Association Analysis*)

Analisis asosiasi digunakan untuk menemukan pola yang menggambarkan kekuatan hubungan fitur dalam data. Pola yang ditemukan biasanya merepresentasikan bentuk aturan implikasi atau subset fitur. Tujuannya adalah untuk menemukan pola yang menarik dengan cara yang efisien.

Penerapan yang paling dekat dengan kehidupan sehari-hari adalah analisis data keranjang belanja. Jika ibu rumah tangga akan membeli barang kebutuhan rumah tangga (misalnya beras) di sebuah supermarket, maka sangat besar kemungkinan ibu rumah tangga tersebut juga akan membeli kebutuhan rumah tangga yang lain, misalnya minyak atau telur, dan tidak mungkin (atau jarang) membeli barang lain seperti topi atau buku. Dengan mengetahui hubungan yang lebih kuat antara beras dengan telur daripada beras dengan topi, maka retailer dapat menentukan barang-barang yang sebaiknya disediakan dalam jumlah yang cukup banyak.

d. Deteksi Anomali (*Anomaly Detection*)

Pekerjaan deteksi anomali berkaitan dengan pengamatan sebuah data dari sejumlah data yang secara signifikan mempunyai karakteristik yang berbeda dari sisa data yang lain. Data yang karakteristiknya menyimpang (berbeda) dari data yang lain disebut sebagai *outlier*. Algoritma deteksi anomaly yang baik harus mempunyai laju deteksi yang tinggi dan laju kesalahan yang rendah. Deteksi anomali dapat diterapkan pada sistem jaringan untuk mengetahui pola data yang memasuki jaringan sehingga dapat diketahui adanya penyusupan jika pola kerja yang datang berbeda ataupun perilaku kondisi cuaca yang mengalami anomaly juga dapat dideteksi dengan algoritma ini.

4. Proses Data Mining

Secara sistematis, ada 3 (tiga) langkah utama dalam data *mining* (Gorunescu 2011) yaitu :

a. Eksplorasi/pemrosesan awal data

Eksplorasi/pemrosesan awal data terdiri dari 'pembersihan' data, normalisasi data, transformasi data, penanganan data yang salah, reduksi dimensi, pemilihan subset fitur, dan sebagainya.

b. Membangun model dan melakukan validasi terhadapnya

Membangun model dan melakukan validasi terhadapnya berarti melakukan analisis berbagai model dan memilih model dengan kinerja prediksi yang terbaik. Dalam langkah ini digunakan metode-metode seperti Klasifikasi, Regresi, Analisis Cluster, Deteksi Anomali, Asosiasi, Analisis Pola Sekuensial, dan sebagainya. Dalam beberapa referensi, Deteksi Anomali juga masuk dalam langkah eksplorasi. Akan tetapi, Deteksi Anomali juga dapat digunakan sebagai algoritma utama, terutama untuk mencari data yang spesial.

c. Penerapan

Penerapan berarti menerapkan model pada data yang baru untuk menghasilkan perkiraan atau prediksi masalah yang diinvestigasi.

5. Set Data

Bukan data mining namanya jika tidak ada set data yang diolah di dalamnya. Kata 'data' dalam terminology statistic adalah kumpulan objek dengan atribut-atribut tertentu, dimana objek tersebut adalah individu berupa data dimana setiap data memiliki sejumlah atribut. Atribut tersebut berpengaruh pada dimensi dari data, semakin banyak atribut/fitur maka semakin besar dimensi data. Kumpulan data membentuk set data. Berikut 3 (tiga) jenis set data yang dikenal dan masing-masing penggolongannya yaitu :

a. Record

- 1) Matriks Data
- 2) Data Transaksi
- 3) Data Dokumen

b. Graph

- 1) World Wide Web (WWW)
- 2) Struktur Molekul

c. Ordered Data Set

- 1) Data Spasial
- 2) Data Temporal
- 3) Data Sekuensial
- 4) Data Urutan Genetik (*Genetic Sequence*)

Dalam set data berbentuk *record* data, tidak ada hubungan antara baris data dengan baris data yang lain dan juga tidak punya hubungan dengan set data yang lain. Setiap baris data berdiri sendiri sebagai sebuah data individu. Dalam sistem basis data, umumnya ada sejumlah tabel yang saling berhubungan menggunakan suatu kunci, tetapi dalam set data berbentuk *record* data, diasumsikan bahwa hanya ada satu tabel yang berisi sejumlah baris data. Maka dari itu, biasanya set data yang diolah dalam data *mining* adalah keluaran dari sistem data *warehouse* yang menggunakan *query* untuk melakukan pengambilan data dari sejumlah tabel dalam sistem basis data. Ada 3 (tiga) jenis record data, yaitu Matriks, Transaksi, dan Dokumen.

Set data yang berisi kumpulan data, dengan semua data mempunyai sejumlah atribut (fitur) numeric yang sama, dapat dipandang sebagai vektor (data) dalam wilayah multidimensi, dimana setiap dimensi (fitur) merepresentasikan atribut berbeda yang menggambarkan objek/data. Set data dengan model seperti itu disebut juga dengan data matriks karena semua nilainya berisi nilai numerik. Sekumpulan objek/data matriks dapat diinterpretasikan sebagai matriks $M \times N$. M adalah jumlah baris, satu baris menyatakan satu data, dan N adalah jumlah kolom, satu kolom menyatakan satu atribut/fitur. Data matriks merupakan jenis record data yang paling umum dan banyak digunakan pada aplikasi-aplikasi statistik. Contoh data matriks ditunjukkan pada tabel 2.5.1

Data keranjang belanja adalah contoh data transaksi. Setiap *record* (transaksi) berisi sejumlah item, dimana jumlah item untuk sebuah transaksi bisa berbeda dengan transaksi yang lain. Contohnya bisa dilihat pada kasus keranjang belanja di pasar atau supermarket. Setiap pembeli melakukan pembelian barang yang jumlah dan jenisnya bisa berbeda dengan pembeli yang lain. Data transaksinya berisi kumpulan item, tetapi dapat dilihat pada *field* atributnya asimetrik (urutan nilai dari kolom pertama sampai terakhir bisa berbeda antara satu transaksi dengan transaksi yang lain). Datanya biasanya digambarkan dengan nilai biner untuk setiap atribut dengan nilai 1 untuk adanya barang (atribut) yang dibeli dan 0 untuk barang yang tidak dibeli. Contoh data keranjang belanja ditunjukkan pada tabel 2.5.2

Untuk data yang menggambarkan dokumen, maka setiap dokumen dicatat dalam basis data menjadi vektor 'istilah'. Setiap istilah menjadi atribut/fitur dari vektor sehingga nilai yang diberikan berarti jumlah kemunculan kata dalam dokumen. Data seperti ini disebut dengan Data Dokumen. Contoh data data dokumen ditunjukkan pada tabel 2.5.3

Tabel 2.5.1 Data Matriks

Suhu (Celcius)	Nilai Mata Kuliah	Jarak (KM)	Frekuensi (HZ)
32	60	38	30
18	85	42	35
20	77	39	39
22	55	38	28
34	66	41	27
38	70	40	29

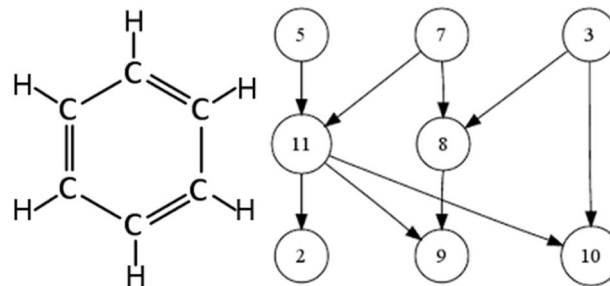
Tabel 2.5.2 Data Transaksi

IDT	Item
1	Buku, Pulpen, Pensil, Penggaris
2	Roti, Terigu, Gula, Merica
3	Minyak, Susu, Pemutih Pakaian, Selimut, Bantal
4	Meja, Guling, Selimut, Bantal
5	HP, Laptop, Speaker, HDD Eksternal
6	Tas, Kipas Angin, AC, Brankas

Tabel 2.5.3 Data Istilah Dokumen

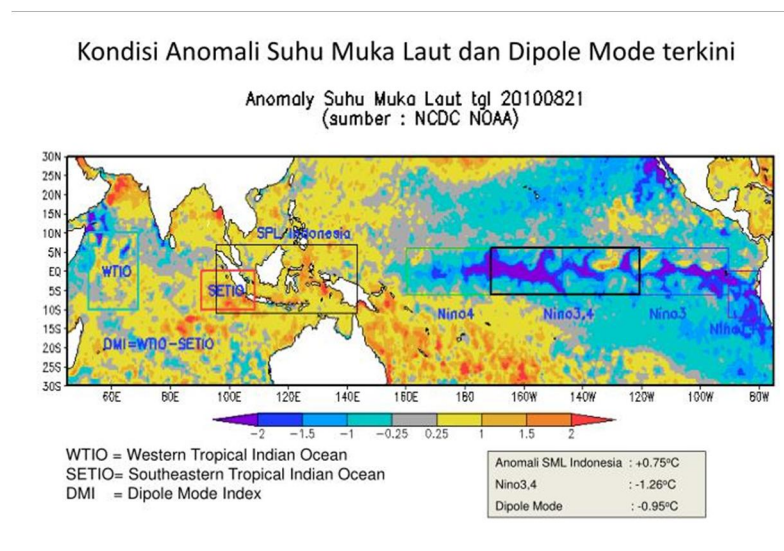
Dokumen	Geografis	Jaringan	Kecerdasan	Tiruan	Algoritma
Dokumen A	2	5	1	0	7
Dokumen B	4	0	3	9	0
Dokumen C	5	4	6	8	12
Dokumen D	0	3	0	0	9
Dokumen E	0	0	0	2	3
Dokumen F	1	2	8	0	0

Data grafik direpresentasikan dalam bentuk grafik (diagram). Seperti namanya, informasi diberikan dalam bentuk gambar dengan jenis tertentu, seperti rumus kimia, link HTML, struktur molekul, dan sebagainya. Contoh dapat dilihat pada gambar di bawah ini :



Gambar 2.5.1 Contoh Data Grafik
(Sumber : (Gorunescu 2011))

Ordered data adalah data yang tersusun dengan suatu cara dalam urutan atau aturan tertentu, misalnya data struktur DNA mempunyai urutan genetik tertentu, data atmosfer dari BMKG akan tersusun menjadi data spasial dengan koordinat tertentu, data rekam medis seorang pasien di puskesmas atau rumah sakit dengan pola terurut penyakit yang diderita, dan sebagainya. Contoh data spasial diberikan pada gambar 2.5.2



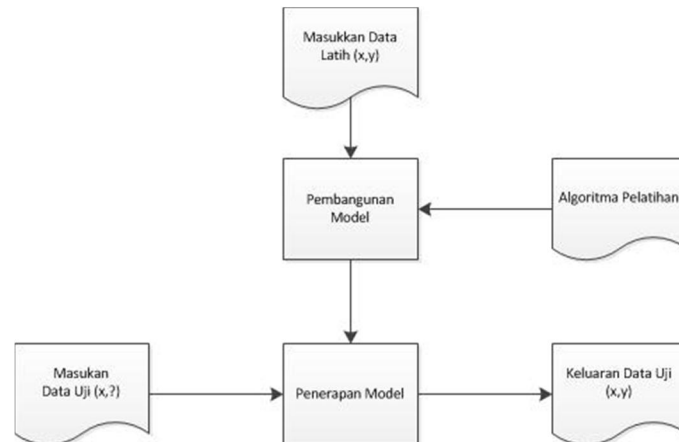
Gambar 2.5.2 Prakiraan Musim Kemarau
(Sumber : <http://www.slideserve.com/nodin/prediksi-bmkg-indonesia>)

6. Metode Klasifikasi

Klasifikasi didefinisikan secara detail sebagai suatu kegiatan atau pekerjaan yang melakukan pelatihan atau pembelajaran terhadap fungsi target f yang memetakan untuk setiap vektor (set fitur) x ke dalam satu dari sejumlah label kelas y yang tersedia. Pekerjaan pelatihan tersebut akan menghasilkan suatu model yang kemudian disimpan sebagai memori (Prasetyo 2013).

Model dalam Klasifikasi mempunyai arti yang sama dengan *black box*, dimana ada suatu model yang menerima masukan kemudian mampu melakukan pemikiran terhadap masukan tersebut dan memberikan jawaban sebagai keluaran dari hasil pemikirannya. Kerangka kerja klasifikasi ditunjukkan pada gambar 2.6.1. pada gambar tersebut, disediakan sejumlah data latih (x,y) untuk digunakan sebagai data pembangunan model

yang diinginkan, kemudian menggunakan model tersebut untuk memprediksi kelas dari data uji ($x, ?$) sehingga data uji ($x, ?$) diketahui kelas y yang seharusnya.



Gambar 2.6.1 Proses Pekerjaan Klasifikasi
(Sumber : (Prasetyo 2014))

Model yang sudah dibangun pada saat pelatihan kemudian dapat digunakan untuk memprediksi label kelas dari data yang belum diketahui label kelasnya. Dalam pembangunan model selama proses pelatihan tersebut diperlukan adanya suatu algoritma untuk membangunnya yang disebut sebagai algoritma pelatihan (*algorithm learning*). Ada banyak algoritma pelatihan yang sudah dikembangkan oleh para peneliti yaitu *Decision Tree*, *K-Nearest Neighbor*, *Artificial Neural Network*, *Support Vector Machine*, dan sebagainya. Setiap algoritma mempunyai kelebihan dan kekurangan masing-masing. Akan tetapi, semua algoritma mempunyai prinsip yang sama yaitu melakukan suatu pelatihan sehingga pada akhir pemodelan bisa memetakan (memprediksi) setiap vektor masukan ke label kelas keluaran yang benar. Kerangka kerja yang telah ditunjukkan pada gambar 2.6.1 meliputi 2 (dua) langkah proses yaitu Induksi dan Deduksi. Induksi merupakan suatu langkah membangun model klasifikasi dari data latih yang diberikan, disebut juga proses pelatihan, sedangkan deduksi merupakan suatu langkah untuk menerapkan model tersebut pada data uji sehingga data uji bisa mengetahui kelas yang sesuai atau disebut juga dengan prediksi.

Berdasarkan cara pelatihan, algoritma-algoritma klasifikasi dapat dibagi menjadi 2 (dua) macam, yaitu *Eager Learner* dan *Lazy Learner*. Algoritma-algoritma yang masuk dalam kategori *Eager Learner* dibuat untuk melakukan pembacaan atau pelatihan atau pembelajaran pada data latih untuk dapat memetakan dengan benar setiap vektor masukan ke dalam label kelas keluarannya sehingga di akhir proses pelatihan, model sudah dapat melakukan pemetaan dengan benar terhadap semua data latih ke label kelas keluarannya. Setelah proses pelatihan tersebut selesai, maka model (biasanya berupa bobot atau sejumlah nilai kuantitas tertentu) disimpan sebagai memori, sedangkan semua data latihnya dibuang. Proses prediksi dilakukan menggunakan model yang tersimpan dan tidak melibatkan data latih sama sekali. Cara ini mengakibatkan proses prediksi berjalan dengan cepat, namun harus member dengan proses pelatihan yang memakan waktu cukup lama. Algoritma-algoritma klasifikasi yang masuk kategori ini yaitu : *Artificial Neural Network*, *Support Vector Machine*, *Decision Tree*, *Bayesian*, dan sebagainya. Sementara algoritma-algoritma yang masuk kategori *Lazy Learner* hanya sedikit melakukan pelatihan

(atau bahkan tidak sama sekali). Algoritma-algoritma ini hanya menyimpan sebagian atau seluruh data pelatihan, kemudian menggunakan data latih tersebut ketika proses prediksi. Hal ini mengakibatkan proses prediksi menjadi lebih lama karena model harus membaca kembali semua data latihnya untuk dapat memberikan keluaran label kelas dengan benar pada data uji yang diberikan. Kelebihan algoritma seperti ini adalah proses pelatihan berjalan dengan cepat. Algoritma-algoritma klasifikasi yang masuk kategori ini yaitu *Rote Classifier*, *K-Nearest Neighbor* (K-NN), *Fuzzy K-Nearest Neighbor* (FK-NN), Regresi Linear, dan sebagainya.

7. Tipe Fitur

Fitur yang menjadi elemen setiap vektor mempunyai jenis yang beragam, misalnya pada data yang mengenai informasi fisik seorang manusia. Ada beberapa fitur tinggi badan yang menggunakan nilai yang sifatnya kuantitatif. Fitur ini mempunyai tipe numerik yang dapat membandingkan satu dengan yang lainnya. Sementara fitur warna kulit menggunakan nilai yang bersifat kualitatif sehingga tidak bisa dilakukan perbandingan. Umumnya tipe fitur ada dua, yaitu kategorikal (kualitatif) dan numerik (kuantitatif). Ada 4 (empat) sifat penting utama yang dimiliki secara umum, yaitu :

- Distinctness*, meliputi sama dengan ($=$) dan tidak sama dengan (\neq).
- Order*, meliputi lebih kecil ($<$), lebih kecil atau sama dengan (\leq), lebih besar ($>$), dan lebih besar atau sama dengan (\geq).
- Addition*, meliputi penjumlahan ($+$) dan pengurangan ($-$).
- Multiplication*, meliputi perkalian ($*$) dan pembagian ($/$).

Dari keempat sifat tersebut dapat diturunkan 4 (empat) tipe fitur, yaitu nominal, ordinal, interval, dan rasio. Lihatlah pada tabel 2.8.1, mengenai penjelasan ketiga tipe fitur dan kaitannya dengan sifat-sifat diatas.

Tabel 2.8.1 Tipe Fitur
(Sumber : (Prasetyo 2014))

Tipe Fitur		Penjelasan	Contoh
Kategorikal (Kualitatif)	Nominal	Nilai fitur bertipe nominal berupa nama, dengan nama inilah sebuah fitur membedakan dirinya pada vektor yang satu dengan yang lain ($=, \neq$).	Kode pos, nomer ktp, nomor induk mahasiswa, jenis kelamin.
	Ordinal	Nilai fitur bertipe ordinal mempunyai nilai berupa nama yang mempunyai arti informasi yang terurut ($<, \leq, >, \geq$).	Predikat kelulusan (cum laude, sangat memuaskan), suhu (dingin, normal, panas).
Numerik (Kuantitatif)	Interval	Nilai fitur dimana perbedaan diantara dua nilai mempunyai makna yang berarti.	Tanggal, suhu (Celcius atau Fahrenheit).
	Rasio	Nilai fitur dimana perbedaan diantara dua nilai dan rasio dua nilai mempunyai makna yang berarti ($*, /$).	Umur, panjang, tinggi, rata-rata.

Fitur nominal dan ordinal merupakan jenis kategorikal dan nilainya kualitatif, misalnya kode pos atau nomor KTP. Nilai ini sebenarnya adalah nilai simbolik, tidak

mungkin dilakukan operasi aritmatika seperti pada tipe numerik. Sementara fitur interval dan rasio merupakan jenis numerik dan nilainya kuantitatif. Nilai ini dapat dilakukan operasi aritmatika dan bisa merepresentasikan dengan nilai *integer* ataupun kontinu. Berdasarkan angka nilai, fitur juga dapat dibedakan menjadi 2 (dua), yaitu Diskrit atau Kontinu. Sebuah fitur dapat bernilai diskret jika mempunyai nilai pada himpunan jumlah yang terbatas. Jenis ini bisa ditemui pada fitur kategorikal yang memiliki beberapa variasi nilai, misalnya suhu. Suhu hanya mempunyai 3 (tiga) kemungkinan nilai yaitu (dingin, normal, dan panas). Contoh lain yang paling sederhana adalah jenis kelamin yang hanya Pria dan Wanita. Nilai ini kadang direpresentasikan dengan nilai biner seperti ya/tidak, benar/salah, pria/wanita, atau 0/1. Sementara fitur yang bernilai kontinu akan mempunyai jangkauan nilai real. Variabel panjang atau tinggi biasanya bernilai menggunakan representasi *floating point* (real). Akan tetapi, meskipun menggunakan representasi real, tetapi digunakan ukuran presisi jumlah angka dibelakang koma.

8. Konsep K-Nearest Neighbor

Nearest Neighbor (NN) menjadi salah satu metode dalam top 10 metode data *mining* yang paling populer (X & Kumar 2009). Metode KNN murni termasuk dalam klasifikasi yang *lazy learner* karena menunda proses pelatihan (atau bahkan tidak melakukan pelatihan sama sekali) sampai ada data uji yang ingin diketahui label kelasnya, maka metode baru akan menjalankan algoritmanya. Algoritma KNN melakukan klasifikasi berdasarkan kemiripan suatu data dengan data yang lain (Steinbach & Kumar 2006). Prinsip sederhana yang diadopsi oleh KNN adalah "jika seekor hewan berjalan seperti anjing, bersuara guk-guk seperti anjing, dan penampilannya seperti anjing, maka hewan itu mungkin anjing".

9. Algoritma K-Nearest Neighbor

Algoritma KNN menjadi salah satu metode yang paling tua dan populer. Nilai K yang digunakan disini menyatakan jumlah tetangga terdekat yang bisa dilibatkan dalam penentuan prediksi pada label kelas data uji. Sebanyak K tetangga terdekat yang terpilih kemudian, akan dilakukan pemungutan suara dari K tetangga terdekat tersebut. Kelas dengan jumlah suara tetangga terbanyak, maka akan dianggap sebagai label kelas hasil prediksi pada data uji tersebut (Steinbach & Kumar 2006).

Algoritma K-Nearest Neighbor (KNN) merupakan sebuah metode untuk melakukan klasifikasi terhadap obyek baru berdasarkan (K) tetangga terdekatnya (Gorunescu 2011). KNN termasuk algoritma *supervised learning*, yang mana hasil dari *query instance* baru, diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Kelas yang paling banyak muncul, yang akan menjadi kelas hasil klasifikasi (Gorunescu 2011).

Nearest Neighbor adalah suatu pendekatan untuk melakukan proses penghitungan kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan dari sejumlah fitur yang ada. Ilustrasi kedekatan kasus, seperti gambar di bawah ini. Memberikan gambar tentang proses mencari solusi terhadap kasus baru dengan mengacu pada solusi kasus lama. Pada kasus lama dengan kedekatan terbesar yang akan diambil solusinya untuk digunakan pada kasus baru.

Isu lain yang juga penting pada KNN, yaitu nilai K yang digunakan, apakah genap atau ganjil. Untuk K bernilai ganjil, dengan jumlah kelas genap akan memudahkan pengambilan suara karena dijamin tidak akan terjadi 2 (dua) kelas yang mendapat suara voting yang sama. Untuk kasus ketika ada 2 (dua) kelas atau lebih dengan pengambilan suara yang sama dan terbanyak maka *label* kelas yang diambil bisa sembarang di antara

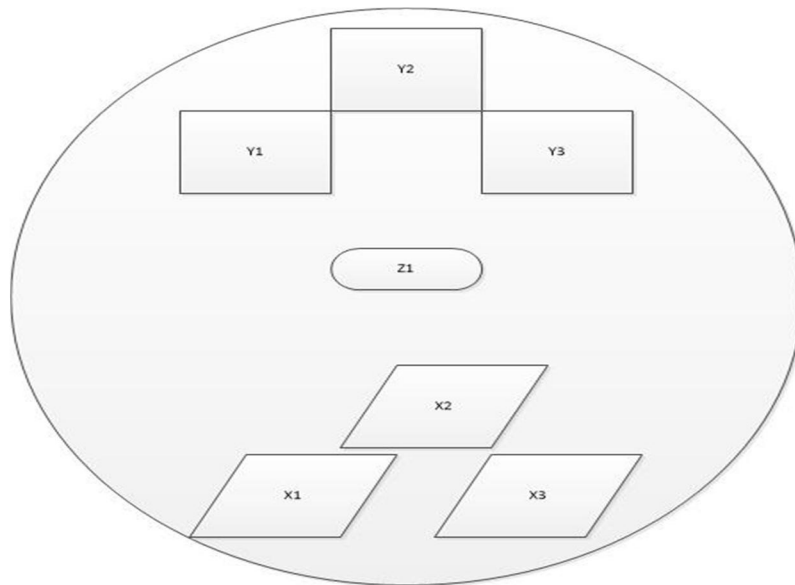
kelas-kelas tadi. Pada algoritma KNN terdapat 5 (lima) cara, untuk mencari tetangga terdekat (Prasetyo 2014) yaitu:

- a) Jarak *Euclidean*
- b) Jarak *Manhattan*
- c) Jarak *Cosine*
- d) Jarak *Correlation*
- e) Jarak *Hamming*

Seperti yang sudah dibahas sebelumnya, pada penulisan Tugas Akhir, penulis hanya menggunakan jarak *Euclidean*, maka rumus perhitungan jarak dengan *Euclidean* seperti di bawah ini (Sayad 2010):

$$\sqrt{\sum_{i=1}^K (X_i - Y_i)^2}$$

Nilai X_i merupakan nilai yang ada pada data *training*, sedangkan nilai Y_i merupakan nilai yang ada pada data *testing*. Nilai K merupakan dimensi atribut.



Gambar 2.8.1 Ilustrasi Kedekatan Kasus

10. Penelitian Terkait

Tabel 2.10.1 Penelitian Terkait

No	Judul	Nama	Tahun	Kesimpulan
1.	Implementasi Data Mining dengan Algoritma C 4.5 untuk Memprediksi Tingkat Kelulusan Mahasiswa	David Hartanto Kamagi, Seng Hansun	2014	Pada penelitian ini akurasi 87.5%. pada penelitian dapat membantu program studi mengetahui status kelulusan mahasiswa, sehingga bisa menjadi rekomendasi pengambilan mata kuliah bagi mahasiswa untuk semester berikutnya seperti KKP dan Skripsi.
2.	<i>Graduation Prediction Of Gunadarma University Students Using Algorithm And Naïve Bayes C 4.5 Algorithm</i>	Marselina Silvia Suhartinah, Ernastuti	2010	Proses pengklasifikasian nilai sangat penting karena dapat mengelompokkan nilai-nilai yang akan diuji. Dengan menggunakan algoritma C 4.5 kesalahan yang dihasilkan dalam proses prediksi lebih sedikit karena C 4.5 melakukan klasifikasi pada sejumlah record ke dalam kelas tujuan yang ada.
3.	Pengambilan Pola Kelulusan Tepat Waktu Pada Mahasiswa STMIK AMIKOM Yogyakarta Menggunakan Data Mining Algoritma C 4.5	Muchamad Henry Widiarto	2011	Dengan menggunakan data mining dapat digunakan untuk membantu manajemen STMIK AMIKOM Yogyakarta dalam menentukan keputusan tepat mahasiswa tepat

No	Judul	Nama	Tahun	Kesimpulan
				pada waktunya sesuai pola yang terbentuk.
4.	Prediksi Mahasiswa Yang Tidak Menyelesaikan Pendidikan Tidak Tepat Waktu Dengan Menggunakan Data Mining Metode Backpropagation (Studi Kasus STMIK Kaputama Binjai)	Relita Buaton, Rahmawati	2012	Pada penelitian dengan menggunakan Jaringan Syaraf Tiruan mampu memprediksi akan menyelesaikan pendidikan atau tidak. Lalu, untuk mendapatkan hasil prediksi yang lebih akurat perlu memperhatikan jumlah simpul lapisan tersembunyi, laju pembelajaran, toleransi galat dan jumlah epoch.
5.	Perancangan Aplikasi Prediksi Kelulusan Tepat Waktu Bagi Mahasiswa Baru Dengan Teknik Data Mining	Muhammad Syukri Mustafa, I Wayan Simpen	2012	Dapat menjadi acuan untuk memprediksi kelulusan seorang mahasiswa baru apakah dapat menyelesaikan kuliahnya dengan tepat waktu atau tidak berdasarkan kedekatan data yang sudah ada. Akurasi yang diperoleh 83.36%
6.	Pembuatan Aplikasi Data Mining Untuk Memprediksi Masa Studi Mahasiswa Menggunakan Algoritma K-NEAREST NEIGHBORHOOD	Astrid Darmawan	2012	Aplikasi data memprediksi untuk 1 (satu) data mahasiswa dan banyak mahasiswa. Tingkat akurasi bervariasi berdasarkan jumlah nilai IPS yang digunakan dan nilai K nya. Tingkat akurasi prediksi dari 76.66% - 85%

No	Judul	Nama	Tahun	Kesimpulan
7.	Analisis Prediksi Tingkat Pengunduran Diri Mahasiswa Dengan Metode K-Nearest Neighbor	Ricky Imanuel Ndaumanu, Kusrini, M. Rudyanto Arief	2014	Hasil pengujian menggunakan <i>prototype</i> sistem pendukung keputusan dan dibandingkan secara manual yang menggunakan metode algoritma KNN dengan 4 variabel yaitu IPK, Pekerjaan Orang Tua, Jurusan, dan Semester adalah mendapatkan kecocokkan 79%.

BAB III

ANALISA MASALAH DAN RANCANGAN PROGRAM

1. Pendahuluan

Bab ini akan membahas mengenai analisa perangkat lunak yang digunakan, analisa masalah yang dihadapi dengan identifikasi masalah beserta strategi pemecahan masalah, analisa aplikasi yang akan dibangun beserta sistem kerja aplikasi, analisa data *mining*, deskripsi proses, rancangan layar yang akan dibuat, *flowchart* dan algoritma aplikasi. Analisa ini digunakan sebagai tahapan untuk mempermudah dalam penyelesaian masalah dan pembuatan aplikasi.

2. Analisa Perangkat

Pada tahap analisa kebutuhan perangkat adalah tahapan pengumpulan kebutuhan-kebutuhan dari semua elemen sistem perangkat yang akan digunakan dalam pembuatan program. Adapaun perangkat keras dan perangkat lunak yang digunakan antara lain adalah sebagai berikut :

a. Perangkat Keras

Perangkat keras komputer yang digunakan untuk membangun aplikasi ini terdiri dari satu buah komputer jinjing dengan spesifikasi sebagai berikut :

- 1) DELL INSPIRON 14-3000 SERIES
- 2) Processor : Intel ® Core i5 ® CPU 4 CPUs @ 2.2Ghz
- 3) Memori : 4GB DDR3
- 4) Storage : 500GB HDD

b. Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun aplikasi ini adalah sebagai berikut :

- 1) Sistem Operasi : Windows 8.1
- 2) Editor : Netbeans V.7.4
- 3) Bahasa Pemrograman : Java

3. Permasalahan Dan Strategi Pemecahan Masalah

a. Analisa Masalah

Universitas Budi Luhur merupakan salah satu perguruan tinggi swasta di Indonesia yang mempunyai jumlah mahasiswa yang sangat banyak. Universitas Budi Luhur, mempunyai 5 (lima) Fakultas yaitu : Fakultas Teknologi Informasi (FTI), Fakultas Ekonomi (FE), Fakultas Teknik (FT), Fakultas Ilmu Sosial dan Politik (FISIP), Fakultas Ilmu Komunikasi (FILKOM). Pada Fakultas Teknologi Informasi, terdapat 3 (tiga) Program Studi yaitu : Teknik Informatika, Sistem Informasi, dan Sistem Komputer. Untuk setiap tahun ajaran baru, pada 5 Fakultas tersebut yang paling banyak diminati oleh mahasiswa baru adalah Fakultas Teknologi Informasi terutama Program Studi Teknik Informatika. Namun, kendala yang sering terjadi adalah semakin banyak mahasiswa baru, akan tetapi sedikit mahasiswa yang lulus. Oleh karena itu, untuk meningkatkan kualitas pada perguruan tinggi Universitas Budi Luhur pada Fakultas Teknologi Informasi Program Studi Teknik Informatika, maka haruslah dilihat pola kelulusan mahasiswa pada setiap tahunnya. Penumpukan data mahasiswa yang sia-sia tanpa diambil informasi yang dapat digunakan sebagai analisa dalam mengambil kebijakan manajemen. Pada setiap mahasiswa mempunyai data yang banyak seperti biodata mahasiswa itu sendiri, mata kuliah yang pernah diambil, catatan pembayaran kuliah, dan sebagainya. Melalui data mahasiswa

tersebut akan digunakan untuk 'penggalian data' atau biasa dikenal dengan ilmu Data *Mining*. Data *Mining* merupakan salah satu cabang ilmu pengetahuan komputer yang tepat untuk membentuk pola-pola yang mungkin memberikan informasi yang bermanfaat pada data mahasiswa dalam jumlah yang besar. Pada Tugas Akhir ini, dapat dirumuskan sebuah permasalahan yang akan dijadikan sebagai acuan yaitu proses Algoritma K-NEAREST NEIGHBOR menentukan mahasiswa mana yang akan lulus cepat, lulus tepat, lulus terlambat, atau tidak lulus.

b. Strategi Pemecahan Masalah

Berdasarkan masalah yang telah diuraikan di atas maka pemecahan masalahnya adalah dengan membuat suatu aplikasi yang menggunakan Data *Mining* dengan Algoritma K-NEAREST NEIGHBOR untuk melihat setiap mahasiswa akan lulus cepat, lulus tepat, lulus terlambat, atau tidak lulus.

Algoritma Klasifikasi KNN adalah salah satu metode klasifikasi yang paling mendasar dan sederhana dan harus menjadi salah satu pilihan pertama untuk studi klasifikasi ketika ada sedikit atau tidak pengetahuan sebelumnya tentang distribusi data. Klasifikasi KNN dikembangkan dari kebutuhan untuk melakukan analisis diskriminan ketika estimasi parametrik diandalkan kepadatan probabilitas yang tidak diketahui sebelumnya atau sulit untuk ditentukan (Peterson 2009)

4. Deskripsi Proses

a. Proses Halaman Utama

Pada tahap ini akan menjelaskan bagaimana cara kerja pada *form* utama yaitu sebagai berikut :

- 1) Jika pengguna menekan tombol *form* tunggal, maka akan membuka *form* tunggal.
- 2) Jika pengguna menekan tombol *form* jamak, maka akan membuka *form* jamak.

b. Proses Tunggal

Pada tahap ini akan menjelaskan bagaimana cara kerja pada *form* tunggal yaitu sebagai berikut :

- 1) Jika pengguna menekan tombol cari file pada bagian "pilih file training", maka pengguna harus mencari berkas file *excel* yang akan digunakan sebagai data *training*.
- 2) Jika pengguna menekan tombol cari file pada bagian "pilih file testing", maka pengguna harus mencari berkas file *excel* yang akan digunakan sebagai data *testing*.
- 3) Jika pengguna menekan tombol tampilkan pada bagian "pilih file training", maka berkas file *excel* akan ditampilkan pada *control* tabelTraining.
- 4) Jika pengguna menekan tombol tampilkan pada bagian "pilih file testing", maka berkas file *excel* akan ditampilkan pada *control* tabelTesting.
- 5) Jika pengguna memilih kriteria *random*, berarti akan mengambil baris pada *control* tabelTesting lalu menampilkan datanya terhadap *control* yang lainnya kecuali *control* comboBoxJK dan labelK2.
- 6) Jika pengguna memilih kriteria *input*, berarti pengguna akan melakukan input seperti biasa.
- 7) Pengguna inputkan jumlah K, sesuai pilihan pada *control* comboBoxJK.
- 8) Jika pengguna menekan tombol proses, maka akan melakukan proses perhitungan hingga selesai dan menampilkan hasil prediksi pada *control* labelK2.

- 9) Jika pengguna menekan tombol bersih, maka membersihkan semua nilai atau pilihan pada bagian "kriteria".

c. Proses Jamak

Pada tahap ini akan menjelaskan bagaimana cara kerja pada *form* jamak yaitu sebagai berikut :

- 1) Jika pengguna menekan tombol cari file pada bagian "pilih file training", maka pengguna harus mencari berkas file *excel* yang akan digunakan sebagai data *training*.
- 2) Jika pengguna menekan tombol cari file pada bagian "pilih file testing", maka pengguna harus mencari berkas file *excel* yang akan digunakan sebagai data *testing*.
- 3) Jika pengguna menekan tombol tampilkan pada bagian "pilih file training", maka berkas file *excel* akan ditampilkan pada *control* tabelTraining.
- 4) Jika pengguna menekan tombol tampilkan pada bagian "pilih file testing", maka berkas file *excel* akan ditampilkan pada *control* tabelTesting.
- 5) Pengguna inputkan jumlah K, sesuai pilihan pada *control* comboBoxJK.
- 6) Jika pengguna menekan tombol proses, maka akan melakukan proses perhitungan hingga selesai dan menampilkan hasil prediksi pada *control* dgvHasilPrediksi.
- 7) Jika pengguna menekan tombol bersih, maka akan membersihkan semua nilai pada *control* textbox, combobox, dan table.

5. Analisa Data Mining

Pada tahap ini akan menjelaskan tentang sumber data, data yang digunakan, proses data dalam data *mining*, dan analisa data . Tidak semua data atribut pada mahasiswa akan digunakan untuk dijadikan atribut yang berkaitan dengan prediksi kelulusan mahasiswa. Hanya beberapa atribut yang sekiranya berguna dan terkait mengenai prediksi kelulusan mahasiswa.

a. Sumber Data

Sumber data yang digunakan berasal dari Fakultas Teknologi Informasi (FTI) program studi Teknik Informatika (TI) Universitas Budi Luhur.

b. Data Yang Digunakan

Atribut data yang digunakan untuk melakukan proses prediksi kelulusan mahasiswa yaitu : NIM, Nama, Jenis Kelamin, IPS 1, IPS 2, IPS 3, IPS 4, IPS 5, IPS 6, Total SKS, dan Keterangan Lulus.

Tabel 3.5.1 Penggunaan Data

No.	Angkatan	Jumlah Data
1.	2009	476
2.	2010	420
3.	2011	376

c. Proses Data Dalam Data Mining

Sebelum data siap digunakan, data tersebut harus melewati tahap *Pre-Processing*. *Pre-Processing* merupakan tahap untuk menyiapkan data mentah dengan tujuan agar data tersebut siap pakai untuk diolah selanjutnya. Proses ini dilakukan diluar pembuatan aplikasi dan masih dilakukan secara mandiri

menggunakan bantuan dari Microsoft Excel. Adapun tahap *Pre-Processing* sebagai berikut :

- 1) Memberikan nama atribut pada setiap data, sesuai dengan nilainya.
- 2) Menambahkan atribut no.
- 3) Menjumlahkan total sks yang didapat dari semester 1 (satu) hingga 6 (enam).
- 4) Menghilangkan atribut asal sma.
- 5) Menghilangkan data mahasiswa yang memiliki sks bernilai 0 (nol).
- 6) Menghilangkan data mahasiswa yang bukan kelas regular dan kampus pusat.
- 7) Melakukan pemberian keterangan, berdasarkan atribut tanggal lulus teori dan total sks.
- 8) Menghilangkan atribut tanggal lulus teori.
- 9) Menghilangkan atribut sks yang diambil mahasiswa dari semester 1 (satu) hingga 6 (enam).
- 10) Melakukan transformasi nilai yaitu :
 - a. Keterangan Lulus

Tabel 3.5.2 Tabel Transformasi Keterangan Lulus

Nilai Awal Keterangan	Nilai Baru Keterangan
Lulus Cepat	20
Lulus Tepat	30
Lulus Terlambat	40
Tidak Lulus/DO	50

- b. Jenis Kelamin

Tabel 3.5.3 Tabel Transformasi Jenis Kelamin

Nilai Awal Jenis Kelamin	Nilai Baru Jenis Kelamin
P (Pria)	11
W (Wanita)	12

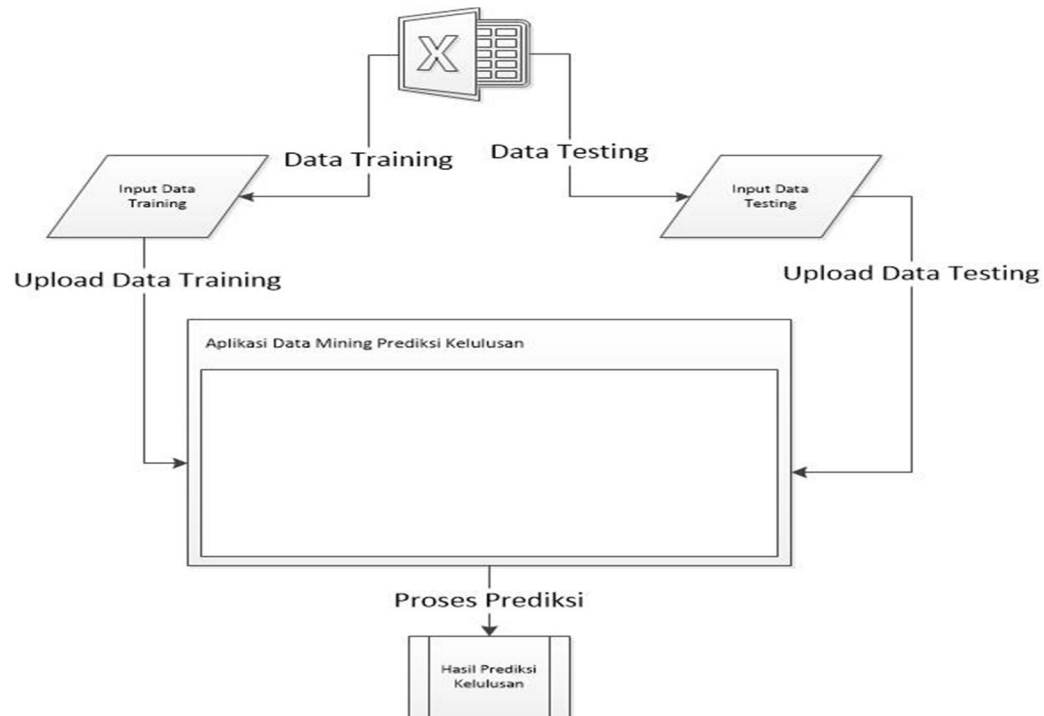
- d. Analisa Data

Setelah melewati tahapan *pre-processing* hingga selesai, maka ditemukan ketidaknormalan data yaitu pada Angkatan 2009 terutama semester 6 (enam) yang mana nilai IPS nya mayoritas bernilai 0 (nol). Sedangkan nilai IPS pada semester 6 (enam) untuk angkatan 2010 dan 2011 yang memiliki nilai 0 (nol) jumlahnya sedikit.

6. Analisa Aplikasi

- a. Analisa Aplikasi Usulan

Aplikasi yang penulis usulkan adalah aplikasi berbasis *desktop* dengan menggunakan bahasa pemrograman Java yang dapat dipasang pada komputer mana pun. Aplikasi ini dapat mengambil data dari *Microsoft Excel*. Data yang telah diambil pertama kali disebut juga sebagai data *training* yang nantinya dapat diproses menggunakan K-NEAREST NEIGHBOR dan dibandingkan dengan data *testing* sehingga menghasilkan prediksi kelulusan. Berikut Arsitektur Aplikasi terlihat pada gambar 3.6.1



Gambar 3.6.1 Arsitektur Aplikasi Usulan

b. Sistem Kerja Aplikasi Usulan

Berdasarkan masalah yang telah dijabarkan di atas, maka aplikasi tersebut harus mempunyai pengetahuan mengenai *Data Mining* dan algoritma K-NEAREST NEIGHBOR, yang mana aplikasi tersebut bisa menerima *input* dari *Microsoft Excel* lalu mengolah data berdasarkan algoritma K-NEAREST NEIGHBOR.

7. Rancangan Layar

a. Rancangan Layar *Form* Halaman Utama

Pada gambar 3.7.1 merupakan rancangan layar *form* halaman utama, yang mana ketika pertama kali dibuka pengguna akan memberikan pilihan apakah hendak melakukan pengujian secara tunggal atau jamak.



Gambar 3.7.1 Rancangan Layar *Form* Halaman Utama

b. Rancangan Layar *Form Tunggal*

Pada gambar 3.7.2 merupakan rancangan layar *form tunggal*, yang mana akan muncul jika pengguna memilih *form tunggal* saat membuka *form* halaman utama. Pada *form* ini, akan melakukan proses perhitungan dengan algoritma KNN dengan banyak data *training* berbanding 1 (satu) data *testing*. Yang mana data *testing* berasal dari inputan atau secara acak berdasarkan data *testing* yang terpilih.

Form Tunggal

Pilih File Training

Ambil File

-Browse-

Cari File Tampilkan

Pilih File Testing

Ambil File

-Browse-

Cari File Tampilkan

Tabel Data Training

<tampil no>	<tampil nim>	<tampil nama>	<tampil jk>	<tampil ips 1>	<tampil ips 2>	<tampil ips 3>	<tampil ips 4>	<tampil ips 5>	<tampil ips 6>	<tampil sks>	<tampil ket>
-------------	--------------	---------------	-------------	----------------	----------------	----------------	----------------	----------------	----------------	--------------	--------------

Tabel Data Testing

<tampil no>	<tampil nim>	<tampil nama>	<tampil jk>	<tampil ips 1>	<tampil ips 2>	<tampil ips 3>	<tampil ips 4>	<tampil ips 5>	<tampil ips 6>	<tampil sks>	<tampil ket>
-------------	--------------	---------------	-------------	----------------	----------------	----------------	----------------	----------------	----------------	--------------	--------------

Kriteria

Kriteria -Kriteria- IPS 3 x-3-x Jumlah K -K-

Total SKS x-3-x IPS 4 x-3-x Status -Status-

IPS 1 x-4-x IPS 5 x-3-x <TampilKeputusan>

IPS 2 x-4-x IPS 6 x-3-x Bersih

NIM x-10-x Proses

Gambar 3.7.2 Rancangan Layar *Form Tunggal*

c. Rancangan Layar *Form* Jamak

Pada gambar 3.7.3 merupakan rancangan layar *form* jamak, yang mana akan muncul jika pengguna memilih *form* jamak saat membuka *form* halaman utama. Pada *form* ini, akan melakukan proses perhitungan dengan algoritma KNN dengan banyak data *training* berbanding banyak data *testing*.

Form Jamak

Pilih File Training
Ambil File
-Browse-
Cari File Tampilkan

Pilih File Testing
Ambil File
-Browse-
Cari File Tampilkan

Tabel Data Training

<tampil no>	<tampil nim>	<tampil nama>	<tampil jk>	<tampil ips 1>	<tampil ips 2>	<tampil ips 3>	<tampil ips 4>	<tampil ips 5>	<tampil ips 6>	<tampil sks>	<tampil ket>
-------------	--------------	---------------	-------------	----------------	----------------	----------------	----------------	----------------	----------------	--------------	--------------

Tabel Data Testing

<tampil no>	<tampil nim>	<tampil nama>	<tampil jk>	<tampil ips 1>	<tampil ips 2>	<tampil ips 3>	<tampil ips 4>	<tampil ips 5>	<tampil ips 6>	<tampil sks>	<tampil ket>
-------------	--------------	---------------	-------------	----------------	----------------	----------------	----------------	----------------	----------------	--------------	--------------

Tabel Data Prediksi

<tampil no>	<tampil nim>	<tampil nama>	<tampil jk>	<tampil ips 1>	<tampil ips 2>	<tampil ips 3>	<tampil ips 4>	<tampil ips 5>	<tampil ips 6>	<tampil sks>	<tampil ket>	<tampil prediksi>
-------------	--------------	---------------	-------------	----------------	----------------	----------------	----------------	----------------	----------------	--------------	--------------	-------------------

Kriteria
Jumlah K ~JumlahK- v Proses Bersih

Gambar 3.7.3 Rancangan Layar *Form* Jamak

d. Rancangan Layar *Form Tunggal* Dengan *Message Box*

Pada gambar 3.7.4 merupakan rancangan layar pada *form* tunggal, yang mana *message box* akan tampil jika seluruh proses sudah selesai, dalam waktu sekian milidetik.

Form Tunggal

Pilih File Training

Ambil File

-Browse-

Cari File Tampilkan

Pilih File Testing

Ambil File

-Browse-

Cari File Tampilkan

Tabel Data Training

<tampil no>	<tampil nim>	<tampil nama>	<tampil skor>	<tampil ket>

Tabel Data Testing

<tampil no>	<tampil nim>	<tampil nama>	<tampil jk>	ips 1>	ips 2>	ips 3>	ips 4>	ips 5>	ips 6>	<tampil skor>	<tampil ket>

Informasi Hasil Testing

-Proses Testing Telah Selesai-

Waktu = XXX Milidetik

Kriteria

Kriteria -Kriteria- IPS 3 x-3-x Jumlah K -K-

Total SKS x-3-x IPS 4 x-3-x Status -Status-

IPS 1 x-4-x IPS 5 x-3-x <TampilKeputusan>

IPS 2 x-4-x IPS 6 x-3-x Bersih

NIM x-10-x Proses

Gambar 3.7.4 Rancangan Layar *Form Tunggal* Dengan *Message Box*

e. Rancangan Layar *Form Jamak* Dengan *Message Box*

Pada gambar 3.7.5 merupakan rancangan layar pada *form* jamak, yang mana *message box* akan tampil jika seluruh proses sudah selesai, dalam waktu sekian detik.

The screenshot displays the 'Form Jamak' application window. It features two file selection sections at the top: 'Pilih File Training' and 'Pilih File Testing', each with a 'Browse' button and a 'Tampilkan' button. Below these are three data tables: 'Tabel Data Training' (4 columns: <tampil no>, <tampil nim>, <tampil nama>, <tampil jk>), 'Tabel Data Testing' (4 columns: <tampil no>, <tampil nim>, <tampil nama>, <tampil jk>), and 'Tabel Data Prediksi' (12 columns: <tampil no>, <tampil nim>, <tampil nama>, <tampil jk>, <tampil ips 1>, <tampil ips 2>, <tampil ips 3>, <tampil ips 4>, <tampil ips 5>, <tampil ips 6>, <tampil sks>, <tampil ket>, <tampil prediksi>). A central message box titled 'Informasi Hasil Prediksi' is overlaid, displaying the following text: '-Proses Testing Telah Selesai-', 'Data Testing = <tampil>', 'Tepat Prediksi = <tampil>', 'Akurasi Prediksi = <tampil>%', and 'Waktu = <tampil> Detik'. At the bottom right, there is a 'Kriteria' section with a 'Jumlah K' label, a dropdown menu showing '~JumlahK-', and two buttons labeled 'Proses' and 'Bersih'.

Gambar 3.7.4 Rancangan Layar *Form* Jamak Dengan *Message Box*

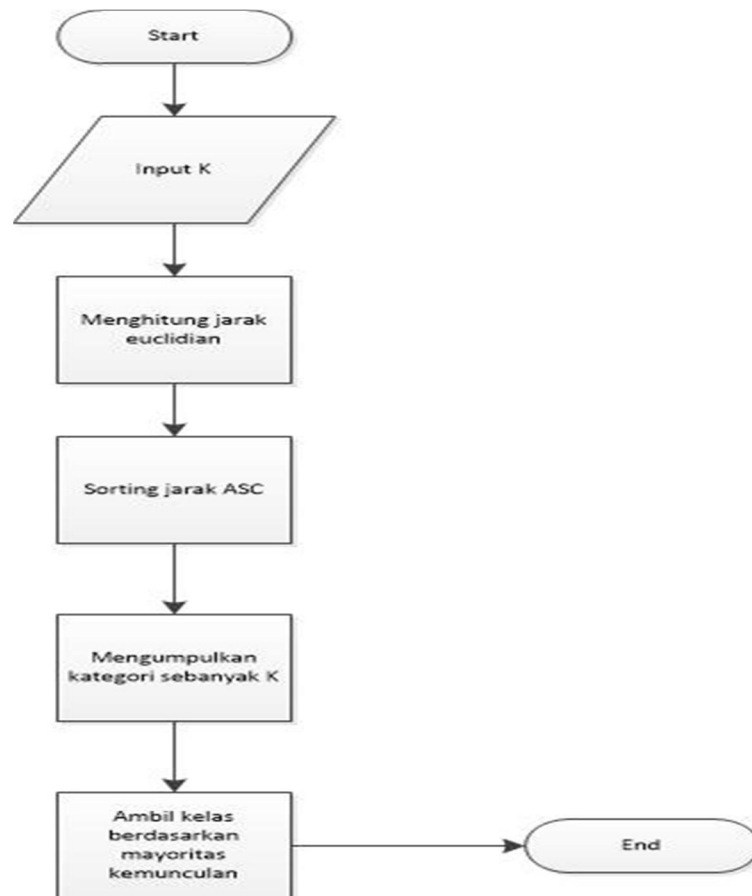
8. Flowchart Dan Algoritma

Pada tahap ini akan menjelaskan mengenai keseluruhan tahapan bagaimana membuat aplikasi prediksi kelulusan mahasiswa dengan menerapkan Algoritma KNN dan proses pada setiap *form* yang ada.

a. Flowchart

1) Flowchart K-Nearest Neighbor

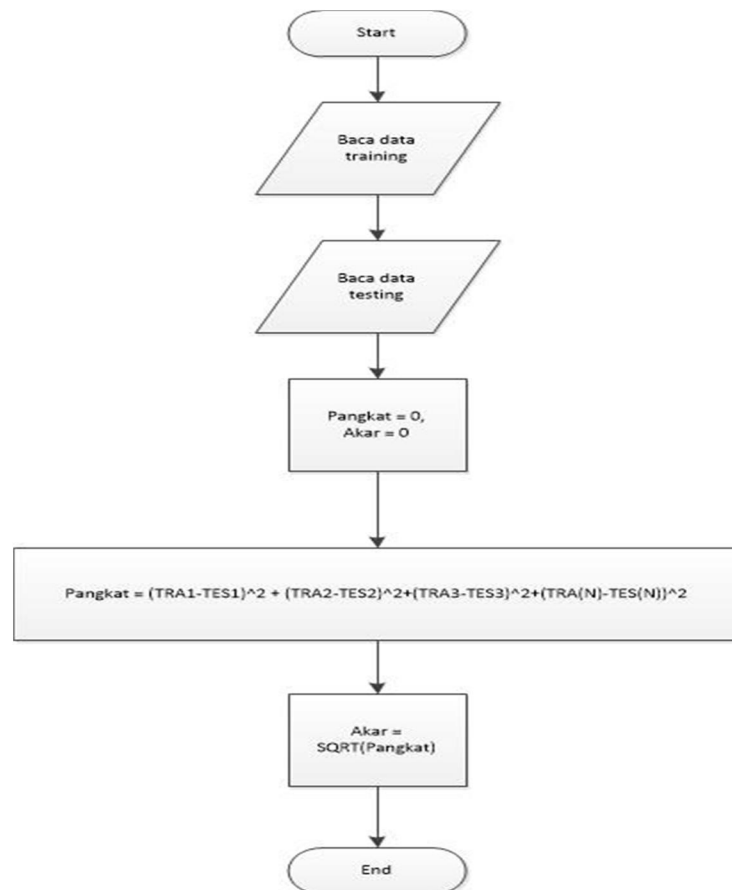
Pada gambar 3.8.1, akan menjelaskan bagaimana cara kerja dari Algoritma KNN itu sendiri. Langkah pertama inputkan nilai K, setelah itu hitunglah jarak menggunakan rumus *Euclidean Distance*. Selanjutnya, lakukan pengurutan dari jarak yang telah dihitung dengan konsep terkecil ke terbesar. Lalu, hasil yang sudah diurutkan dari terkecil ke terbesar diambil sebanyak K, dan langsung dihitung kemunculan dari setiap kelas, kelas dengan jumlah paling banyak kemunculannya maka menjadi hasil keputusan.



Gambar 3.8.1 Flowchart K-Nearest Neighbor

2) Flowchart Menghitung Jarak Euclidean

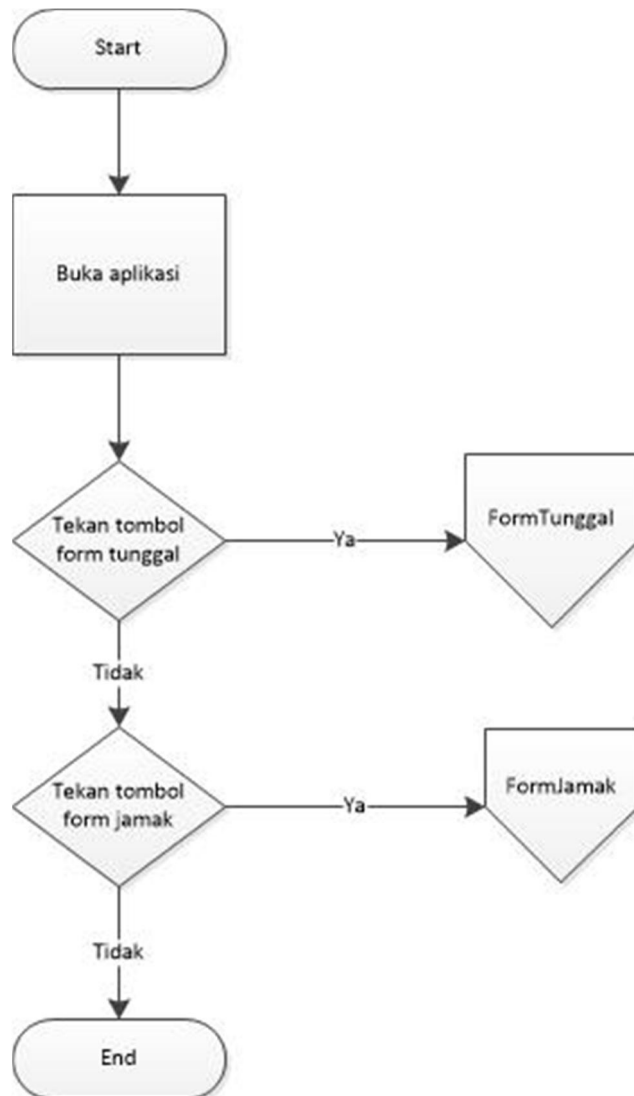
Pada gambar 3.8.2, akan menjelaskan bagaimana proses perhitungan menggunakan rumus pada *Euclidean Distance*. Langkahnya yaitu, baca data *training* dan *testing*, setelah itu masukkan ke dalam inti rumus untuk setiap data *training* – data *testing* lalu dipangkatkan 2 (dua) dan diakarkan.



Gambar 3.8.2 Flowchart Menghitung Jarak Euclidian

3) *Flowchart Form Halaman Utama*

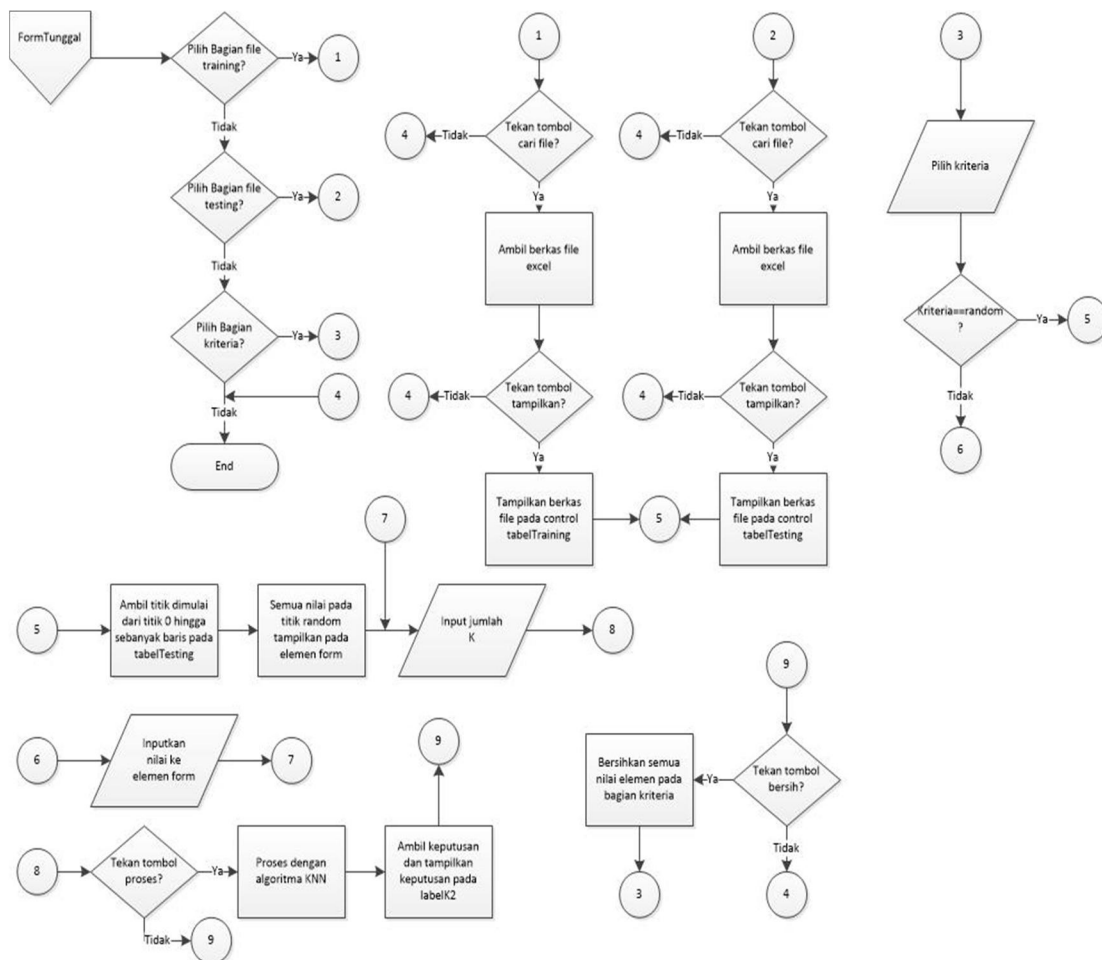
Pada gambar 3.8.3, akan menjelaskan bagaimana proses pada *form* halaman utama aplikasi data *mining* ini.



Gambar 3.8.3 *Flowchart Form Halaman Utama*

4) Flowchart Form Tunggal

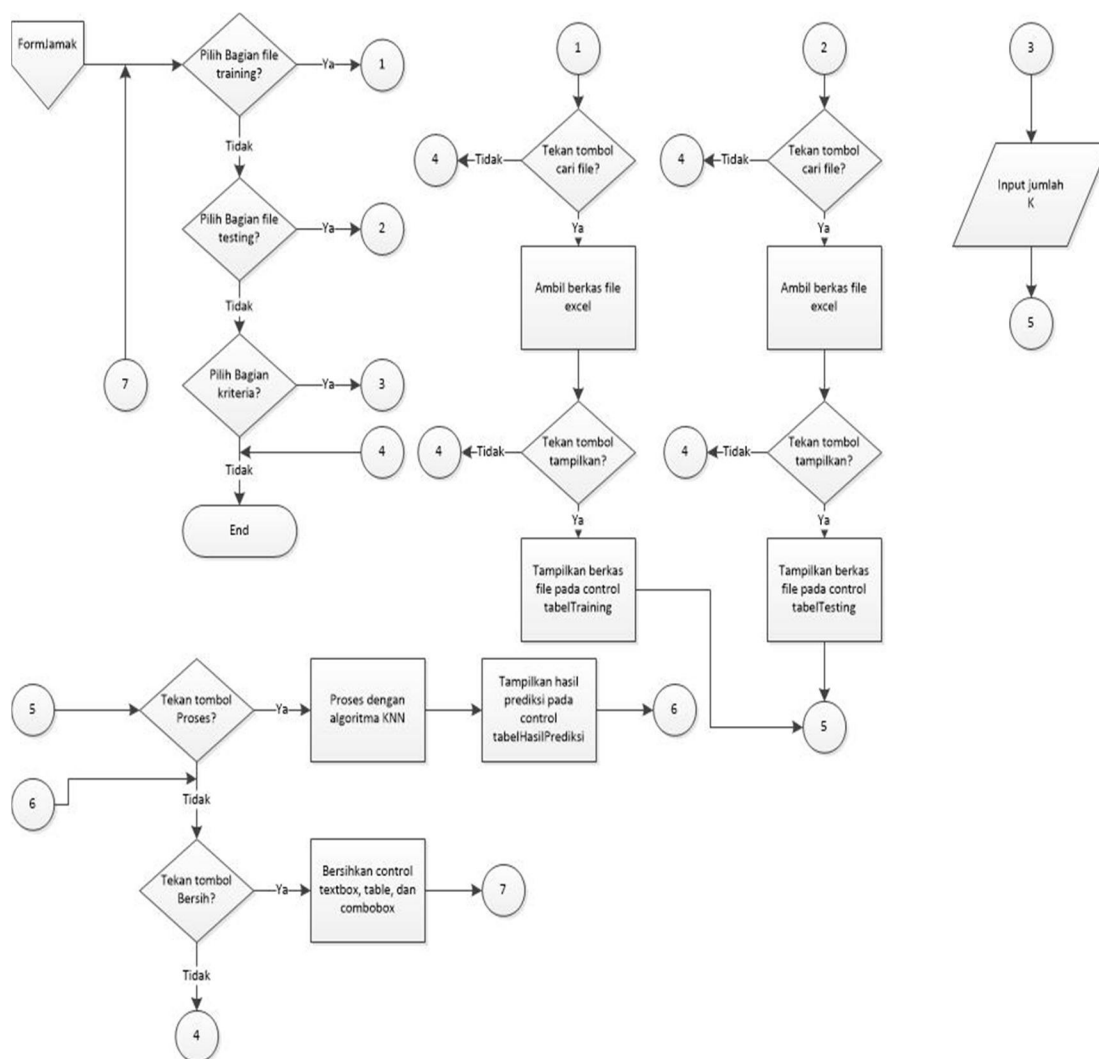
Pada gambar 3.8.4, akan menjelaskan bagaimana keseluruhan proses yang terjadi pada *form* tunggal. Terdapat 3 (tiga) bagian yaitu : bagian *Training*, bagian *Testing*, dan bagian Kriteria. Jika pengguna berada pada bagian *training*, dan menekan tombol cari file maka pengguna mencari berkas *file excel*, setelah itu menekan tombol tampilkan lalu ditampilkan pada control tabelTraining. Jika pengguna berada pada bagian *testing*, dan menekan tombol cari file maka pengguna mencari berkas *file excel*, setelah itu menekan tombol tampilkan lalu ditampilkan pada control tabelTesting. Jika pengguna berada pada bagian kriteria, dan memilih random maka akan diambil sebuah titik antara 0 hingga jumlah data *training* pada tabelTesting, lalu pengguna memasukkan jumlah K yang diinginkan dan menekan tombol proses sehingga menampilkan hasil prediksi pada *control* labelK2. Tetapi jika pengguna memilih kriteria untuk melakukan input, maka pengguna setelah menginput memilih jumlah K dan menekan tombol proses sehingga menampilkan hasil preidksi pada *control* labelK2. Jika pengguna menekan tombol bersih, maka pengguna bisa melakukan perubahan kriteria dari awal kembali.



Gambar 3.8.4 Flowchart Form Tunggal

5) Flowchart Form Jamak

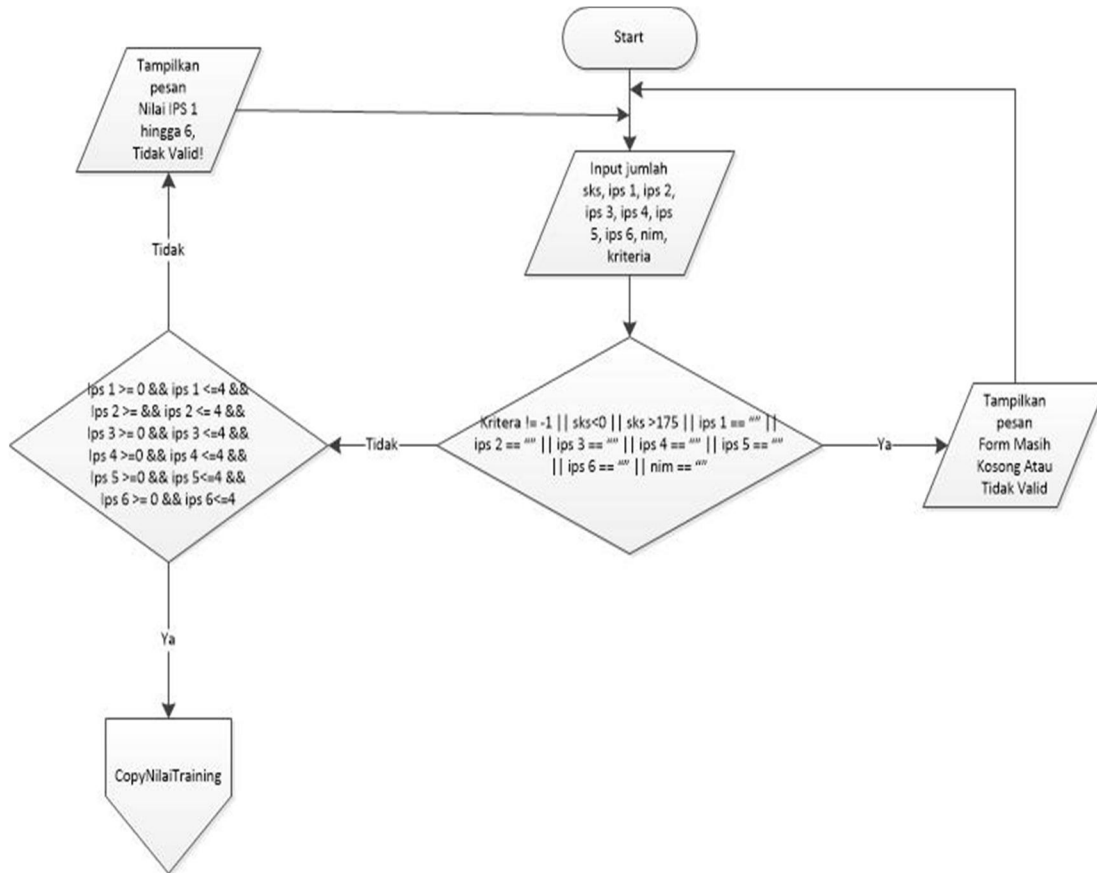
Pada gambar 3.8.5, akan menjelaskan bagaimana keseluruhan proses yang terjadi pada *form* jamak. Terdapat 3 (tiga) bagian yaitu : bagian *Training*, bagian *Testing*, dan bagian Kriteria. Jika pengguna berada pada bagian *training*, dan menekan tombol cari file maka pengguna mencari berkas *file excel*, setelah itu menekan tombol tampilkan lalu ditampilkan pada control tabelTraining. Jika pengguna berada pada bagian *testing*, dan menekan tombol cari file maka pengguna mencari berkas *file excel*, setelah itu menekan tombol tampilkan lalu ditampilkan pada control tabelTesting. Jika pengguna pada bagian kriteria, lalu melakukan input jumlah K nya. Setelah itu pengguna menekan tombol proses lalu dilakukan perhitungan dengan menggunakan algoritma KNN dan ditampilkan hasil prediksi pada *control* tabelPrediksi. Jika pengguna menekan tombol bersih, maka membersihkan nilai pada *control textbox*, *table*, dan *combobox* sehingga sama seperti ketika pertama kali *form* diload.



Gambar 3.8.5 Flowchart Form Jamak

6) *Flowchart* Proses Pada *Form* Tunggal

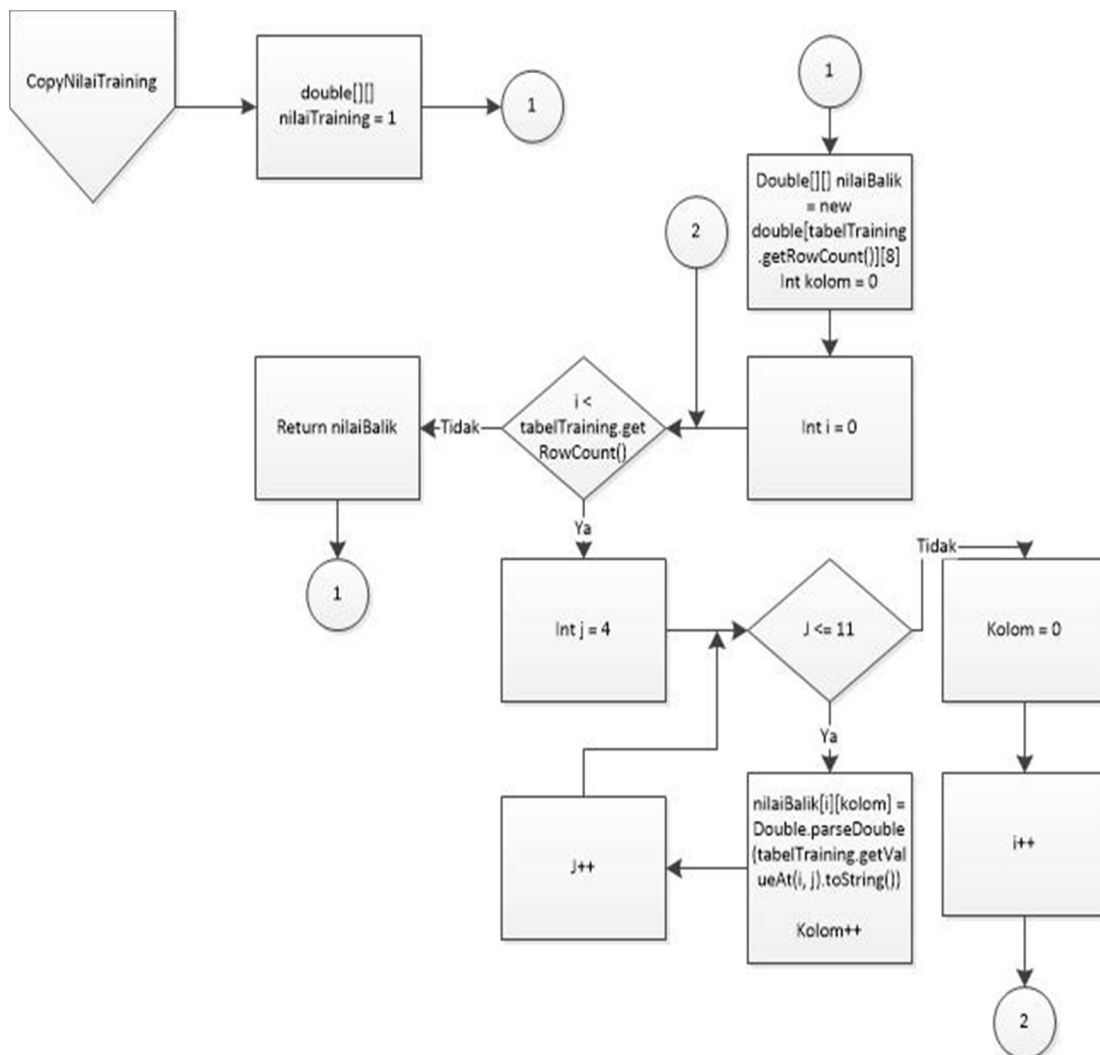
Pada gambar 3.8.6, akan menjelaskan tentang proses yang terjadi jika pengguna menekan tombol proses pada *form* tunggal, hanya saja yang akan dibahas pada *flowchart* di bawah bukan secara keseluruhan melainkan per-bagian. Bagian yang akan dijabarkan adalah bagian menyimpan nilai data *training* yang sumbernya berasal dari tabel data *training*. Proses keseluruhan untuk menyimpan data *training* dilakukan oleh fungsi *CopyNilaiTraining*.



Gambar 3.8.6 *Flowchart* Proses Pada *Form* Tunggal

7) *Flowchart* Proses Menyimpan Nilai Training Pada *Form* Tunggal

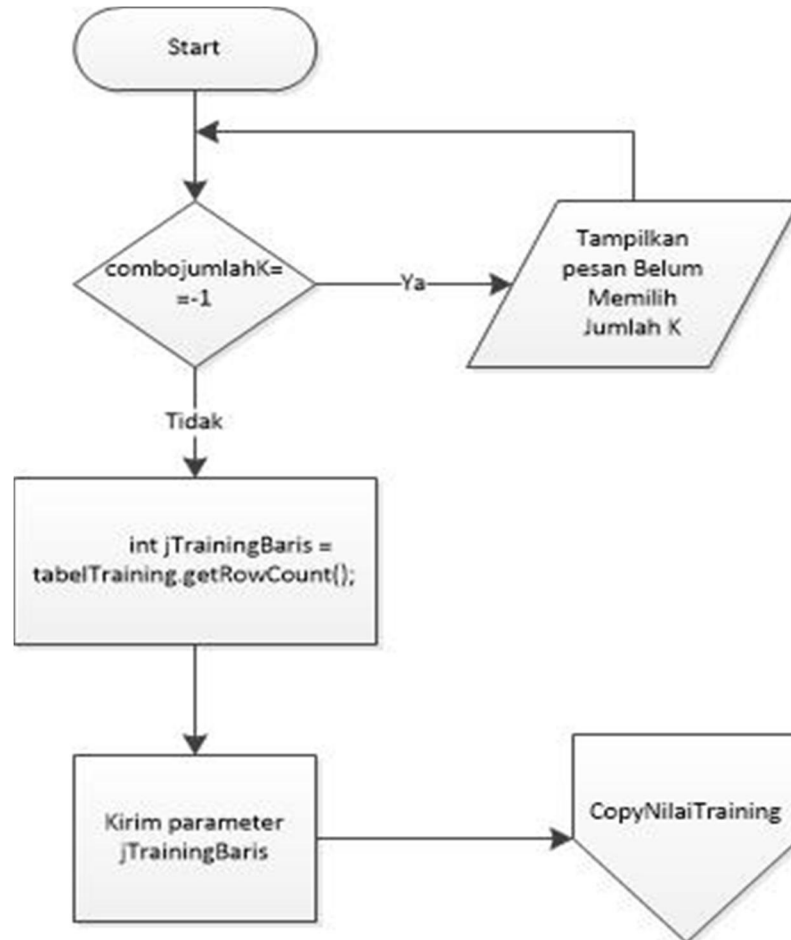
Pada gambar 3.8.7, akan menjelaskan tentang proses membuat fungsi bertipe data *double* larik 2 dimensi yang akan menyimpannya dan melemparkan nilainya pada variabel yang memanggilnya. Langkah selanjutnya buat sebuah larik dengan nama *nilaiBalik*, larik tersebut berfungsi sebagai penyimpanan semua nilai data *training* yang diperlukan seperti IPS 1 hingga 6, total sks, dan keterangan. Setelah itu lakukan perulangan yang pertama dari 0 hingga sebanyak jumlah data pada tabel *training*, lalu lakukan perulangan yang ke 2 mulai dari 4 hingga 11. Pada perulangan ke 2, dimulai pada titik 4 karena posisi IPS 1 berada ke 4 hingga keterangan pada titik 11, selanjutnya simpan pada *nilaiBalik* ke *i*, kolom.



Gambar 3.8.7 *Flowchart* Proses Menyimpan Nilai *Training* Pada *Form* Tunggal

8) *Flowchart* Proses Pada *Form* Jamak

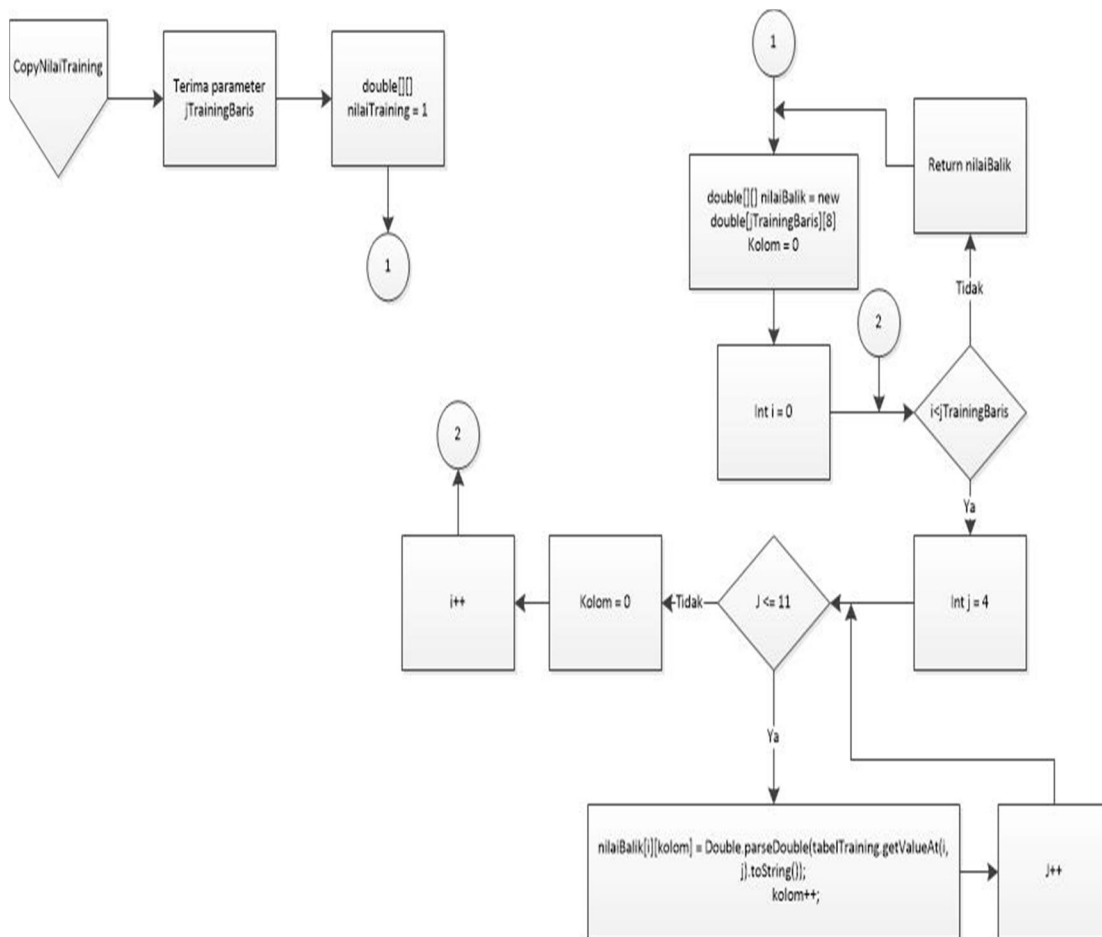
Pada gambar 3.8.8, akan menjelaskan tentang proses yang terjadi jika pengguna menekan tombol proses pada *form* jamak, hanya saja yang akan dibahas pada *flowchart* di bawah bukan secara keseluruhan melainkan per-bagian. Bagian yang akan dijabarkan adalah bagian menyimpan nilai data *training* yang sumbernya berasal dari tabel data *training*. Proses keseluruhan untuk menyimpan data *training* dilakukan oleh fungsi *CopyNilaiTraining*.



Gambar 3.8.8 *Flowchart* Proses Pada *Form* Jamak

9) *Flowchart* Proses Menyimpan Nilai Training Pada *Form* Jamak

Pada gambar 3.8.9, akan menjelaskan tentang proses membuat fungsi bertipe data *double* larik 2 dimensi yang akan menyimpannya dan melemparkan nilainya pada variabel yang memanggilnya. Pertama diterima terlebih dahulu nilai dari parameter *jTrainingBaris*, langkah selanjutnya buat sebuah larik dengan nama *nilaiBalik*, larik tersebut berfungsi sebagai penyimpanan semua nilai data *training* yang diperlukan seperti IPS 1 hingga 6, total sks, dan keterangan. Setelah itu lakukan perulangan yang pertama dari 0 hingga sebanyak *jTrainingBaris*, lalu lakukan perulangan yang ke 2 mulai dari 4 hingga 11. Pada perulangan ke 2, dimulai pada titik 4 karena posisi IPS 1 berada ke 4 hingga keterangan pada titik 11, selanjutnya simpan pada *nilaiBalik* ke *i*, kolom.



Gambar 3.8.9 *Flowchart* Proses Menyimpan Nilai *Training* Pada *Form* Jamak

b. Algoritma

1) Algoritma KNN

Pada Algoritma di bawah ini, akan menjelaskan bagaimana proses yang terjadi menggunakan Algoritma KNN yaitu :

1. Input nilai K
2. Hitung jarak dengan rumus Euclidian setiap data training terhadap data testing
3. Sorting nilai training berdasarkan jarak secara ASC
4. Mengumpulkan kategori nilai training sebanyak K
5. Ambil kelas berdasarkan mayoritas kemunculan, lalu hitung jumlah setiap kelas ada berapa

2) Algoritma Menghitung Jarak *Euclidian*

Pada Algoritma di bawah ini, akan menjelaskan bagaimana proses menghitung jarak menggunakan *Euclidian* yaitu :

1. Baca data training
2. Baca data testing
3. Pangkat = 0
4. Akar = 0
5. $\text{Pangkat} = (\text{TRA1}-\text{TES1})^2 + (\text{TRA2}-\text{TEST2})^2 + (\text{TRA(N)}-\text{TES(N)})^2$
6. $\text{Akar} = \text{SQRT}(\text{Pangkat})$

3) Algoritma *Form* Halaman Utama

Pada Algoritma di bawah ini, akan menjelaskan bagaimana proses yang terjadi pada *form* halaman utama yaitu :

1. Buka aplikasi
2. If tekan tombol tunggal then
3. Buka FormTunggal
4. Else If tekan tombol jamak then
5. Buka FormJamak
6. Else
7. Selesai
8. End if

4) Algoritma *Form* Tunggal

Pada Algoritma di bawah ini, akan menjelaskan bagaimana proses yang terjadi pada *form* tunggal yaitu :

1. Buka FormTunggal
2. If pilih bagian training then
3. If tekan tombol cari file then
4. Ambil berkas file excel
5. If tekan tombol tampilkan then
6. Tampilkan berkasi file pada control tabelTraining
7. Else

```
8.           Menjalankan baris 47
9.           End if
10.        Else
11.           Menjalankan baris 47
12.        End if
13. Else If pilih bagian testing then
14.     If tekan tombol cari file then
15.         Ambil berkas file excel
16.         If tekan tombol tampilkan then
17.             Tampilkan berkasi file pada control tabelTesting
18.         Else
19.             Menjalankan baris 47
20.         End if
21.     Else
22.         Selesai
23.     End if
24. Else if pilih bagian kriteria then
25.     Pilih tipe kriteria
26.     If kriteria == random then
27.         Ambil titik dari titik 0 hingga sebanyak baris pada tabelTesting
28.         Semua nilai pada titik random tampilkan pada elemen form
29.         Inputkan jumlah K
30.         If tekan tombol proses then
31.             Proses dengan algoritma KNN
32.             Ambil keputusan dan tampilkan keputusan pada labelK2
33.             Menjalankan baris 35
34.         Else
35.             If tekan tombol bersih then
36.                 Bersihkan semua nilai elemen pada bagian kriteria
37.                 Kembali pada baris 25
38.             Else
39.                 Menjalankan baris 47
40.             End if
41.         End if
42.     Else
43.         Inputkan nilai ke elemen form
44.         Menjalankan baris 29
45.     End if
46. Else
47.     Selesai
48. End if
```


5) Algoritma *Form* Jamak

Pada Algoritma di bawah ini, akan menjelaskan bagaimana proses yang terjadi pada *form* jamak yaitu :

```

1.  Buka FormJamak
2.  If pilih bagian training then
3.    If tekan tombol cari file then
4.      Ambil berkas file excel
5.      If tekan tombol tampilkan then
6.        Tampilkan berkas file pada control tabelTraining
7.      Else
8.        Menjalankan baris 37
9.      End if
10. Else
11.   Menjalankan baris 37
12. End if
13. Else if pilih bagian testing then
14.   If tekan tombol cari file then
15.     Ambil berkas file excel
16.     If tekan tombol tampilkan then
17.       Tampilkan berkas file pada control tabelTesting
18.     Else
19.       Menjalankan baris 37
20.     End if
21.   Else
22.     Menjalankan baris 37
23.   End if
24. Else if pilih bagian kriteria then
25.   Input jumlah K
26.   If tekan tombol proses then
27.     Proses dengan algoritma KNN
28.     Tampilkan hasil prediksi pada control tabelHasilPrediksi
29.     Menjalankan baris 30
30.   Else if tekan tombol bersih then
31.     Bersihkan control textbox, table, dan combobox
32.     Kembali pada baris 1
33.   Else
34.     Menjalankan baris 37
35.   End if
36. Else
37.   Selesai
38. End if

```

6) Algoritma Proses Pada *Form* Tunggal

Pada Algoritma di bawah ini, akan menjelaskan bagaimana proses yang terjadi pada *form* tunggal yaitu :

```

1. Input jumlah sks, IPS 1, IPS 2, IPS 3, IPS 4, IPS 5, IPS 6, nim, kriteria
2. If kriteria != -1 atau sks < 0 atau sks > 175 atau IPS 1 == "" atau IPS
   2 == "" atau IPS 3 == "" atau IPS 4 == "" atau IPS 5 == "" atau IPS 6
   == "" atau nim == "" then
3.     Tampilkan pesan form masih kosong atau tidak valid
4. Else
5.     If IPS 1>=0 dan IPS 1<=4 dan IPS 2>=0 dan IPS2 <=4 dan IPS
       3>=0 dan IPS 3<=4 dan IPS 4>=0 dan IPS 4<=0 dan IPS 5>=0
       dan IPS 5<=4 dan IPS 6>=0 dan IPS 6<=0 then
6.         CopyNilaiTraining
7.     Else
8.         Tampilkan pesan nilai IPS 1 hingga 6 tidak valid!
9.     End if
10. End if

```

7) Algoritma Proses Menyimpan Nilai Training Pada *Form* Tunggal

Pada Algoritma di bawah ini, akan menjelaskan bagaimana proses menyimpan nilai *training* yang berasal dari tabel *training* pada *form* tunggal yaitu :

```

1. nilaiBalik mempunyai baris sepanjang tabel training dan kolom
   sepanjang 8
2. kolom = 0
3. loop i mulai dari 0 hingga sebanyak baris pada tabel training
4.     loop j mulai dari 4 hingga 11
5.         nilaiBalik baris ke i dan kolom ke j = nilai tabel training pada baris
           ke i dan kolom ke j
6.         kolom bertambah satu
7.     end loop
8.     kolom = 0
9. end loop
10. return nilaiBalik

```

8) Algoritma Proses Pada *Form* Jamak

Pada Algoritma di bawah ini, akan menjelaskan bagaimana proses yang terjadi pada *form* jamak yaitu :

1. If comboJumlahK bernilai -1 then
2. Tampilkan pesan belum memilih jumlah K
3. Else
4. jTrainingBaris = jumlah baris pada tabel training
5. Kirim parameter jTrainingBaris
6. CopyNilaiTraining
7. End if

9) Algoritma Proses Menyimpan Nilai Training Pada *Form* Jamak

Pada Algoritma di bawah ini, akan menjelaskan bagaimana proses menyimpan nilai *training* yang berasal dari tabel *training* pada *form* jamak yaitu :

1. Terima parameter jTrainingBaris
2. nilaiBalik mempunyai baris sepanjang jTrainingBaris dan kolom sepanjang 8
3. kolom = 0
4. loop i mulai 0 hingga jTrainingBaris
5. loop j mulai 4 hingga 11
6. nilaiBalik baris ke i dan kolom ke j = nilai tabel training pada baris ke i dan kolom ke j
7. kolom bertambah satu
8. end loop
9. kolom = 0
10. end loop
11. return nilaiBalik

BAB IV

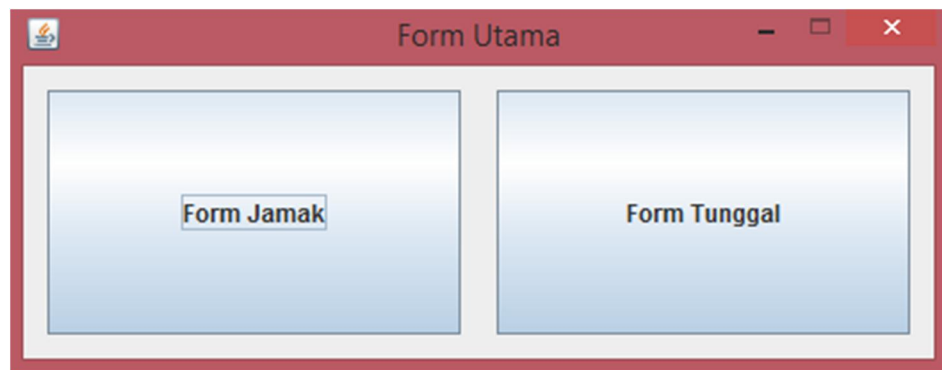
IMPLEMENTASI DAN UJI COBA PROGRAM

1. Tampilan Layar dan Implementasi

a. Tampilan Layar

1) Tampilan Layar *Form* Utama

Pada gambar 4.1.1 merupakan tampilan layar *form* utama ketika pertama kali ditampilkan. Pada *form* utama ini terdapat 2 (dua) tombol yaitu : tombol form jamak dan tombol form tunggal. Jika pengguna menekan tombol form jamak, maka akan membuka form jamak. Tapi, jika pengguna menekan tombol form tunggal, maka akan membuka form tunggal.



Gambar 4.1.1 Tampilan Layar *Form* Utama

2) Tampilan Layar *Form* Tunggal

Pada gambar 4.1.2 merupakan tampilan layar *form* tunggal, yang mana akan terbuka ketika pengguna menekan tombol *form* tunggal pada *form* utama. Pada *form* ini dibagi menjadi 3 (tiga) bagian yaitu : bagian training, bagian testing, dan bagian kriteria. Pada 3 (tiga) bagian tersebut, mempunyai peranan masing-masing.

Piilh File Training

Ambil File

Cari File Tampilkan

Piilh File Testing

Ambil File

Cari File Tampilkan

Tabel Data Training

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET

Tabel Data Testing

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET

Kriteria

Tipe Kriteria: Random

Total SKS:

IPS 1: IPS 2: IPS 3: IPS 4: IPS 5: IPS 6:

NIM:

Jumlah K:

Status:

Prediksi:

Proses Bersih

Gambar 4.1.2 Tampilan Layar *Form* Tunggal

3) Tampilan Layar *Form* Jamak

Pada gambar 4.1.3 merupakan tampilan layar *form* jamak, yang mana akan terbuka ketika pengguna menekan tombol form tunggal pada form utama. Pada *form* ini dibagi menjadi 3 (tiga) bagian yaitu : bagian training, bagian testing, dan bagian kriteria. Pada 3 (tiga) bagian tersebut, mempunyai peranan masing-masing.

Pilih File Training

Ambil File

Cari File Tampilkan

Pilih File Testing

Ambil File

Cari File Tampilkan

Tabel Data Training

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET

Tabel Data Testing

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET

Tabel Data Prediksi

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET	Prediksi

Kriteria

Jumlah K ▼

Gambar 4.1.3 Tampilan Layar *Form* Jamak

b. Implementasi

1) Implementasi *Form Tunggal*

Pada gambar 4.1.4 merupakan tampilan layar *form* tunggal, yang mana akan terbuka ketika pengguna menekan tombol *form* tunggal pada *form* utama. Pada *form* ini dibagi menjadi 3 (tiga) bagian yaitu : bagian training, bagian testing, dan bagian kriteria. Pada 3 (tiga) bagian tersebut, mempunyai peranan masing-masing.

Pilih File Training

Ambil File

Documents\Kumpulan\2010_15112015_1616.xlsx

Cari File Tampilkan

Pilih File Testing

Ambil File

Documents\Kumpulan\2011_15112015_1617.xls

Cari File Tampilkan

Tabel Data Training

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET
1.0	10115...	Akbar...	11.0	3.04	2.95	2.38	2.63	2.71	3.11	124.0	40.0
2.0	10115...	Erwin...	11.0	3.42	3.58	4.0	3.75	3.77	3.52	143.0	20.0
3.0	10115...	Dwi Y...	11.0	2.63	2.09	1.75	2.5	2.07	2.08	60.0	50.0
4.0	10115...	Yohan...	11.0	2.76	2.23	2.67	2.5	3.0	2.78	102.0	40.0

Tabel Data Testing

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET
1.0	11115...	Iskan...	11.0	2.95	0.0	0.0	0.0	0.0	0.0	32.0	50.0
2.0	11115...	Edwin...	11.0	2.45	0.37	0.5	0.0	0.0	0.68	20.0	50.0
3.0	11115...	Angga...	11.0	3.4	3.14	3.26	3.63	3.5	3.84	129.0	40.0
4.0	11115...	Adhiti...	11.0	3.4	2.77	3.42	3.5	3.77	3.88	138.0	20.0

Kriteria

Tipe Kriteria: Random

Total SKS: 72.0

IPS 1: 1.85

IPS 2: 2.65

NIM: 1111504104

IPS 3: 2.4

IPS 4: 2.0

IPS 5: 2.0

IPS 6: 2.94

Jumlah K: 20

Status: Tidak Lulus/DO

Prediksi: Tidak Lulus/DO

Proses Bersih

Gambar 4.1.4 Implementasi *Form Tunggal*

2) Implementasi *Form* Jamak

Pada gambar 4.1.5 merupakan tampilan layar *form* jamak, yang mana akan terbuka ketika pengguna menekan tombol *form* jamak pada *form* utama. Pada *form* ini dibagi menjadi 3 (tiga) bagian yaitu : bagian training, bagian testing, dan bagian kriteria. Pada 3 (tiga) bagian tersebut, mempunyai peranan masing-masing.

Pilih File Training

Ambil File

ian\Documents\Kumpulan\2010_15112015_1616.xlsx

Cari File Tampilkan

Pilih File Testing

Ambil File

tian\Documents\Kumpulan\2011_15112015_1617.xls

Cari File Tampilkan

Tabel Data Training

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET
1.0	10115...	Akbar ...	11.0	3.04	2.95	2.38	2.63	2.71	3.11	124.0	40.0
2.0	10115...	Erwin ...	11.0	3.42	3.58	4.0	3.75	3.77	3.52	143.0	20.0
3.0	10115...	Dwi Yo...	11.0	2.63	2.09	1.75	2.5	2.07	2.08	60.0	50.0
4.0	10115...	Yohan...	11.0	2.76	2.23	2.67	2.5	3.0	2.78	102.0	40.0
5.0	10115...	Andria...	11.0	2.71	2.55	2.57	2.5	2.89	2.76	101.0	40.0

Tabel Data Testing

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET
1.0	11115...	Iskand...	11.0	2.95	0.0	0.0	0.0	0.0	0.0	32.0	50.0
2.0	11115...	Edwin ...	11.0	2.45	0.37	0.5	0.0	0.0	0.68	20.0	50.0
3.0	11115...	Angga ...	11.0	3.4	3.14	3.26	3.63	3.5	3.84	129.0	40.0
4.0	11115...	Adhitia...	11.0	3.4	2.77	3.42	3.5	3.77	3.88	138.0	20.0
5.0	11115...	Abdul ...	11.0	3.75	3.45	3.29	3.71	3.5	3.42	138.0	20.0

Tabel Data Prediksi

No.	NIM	Nama	JK	IPS1	IPS2	IPS3	IPS4	IPS5	IPS6	SKS	KET	PRED...
1.0	11115...	Iskan...	11.0	2.95	0.0	0.0	0.0	0.0	0.0	32.0	50.0	50.0
2.0	11115...	Edwin...	11.0	2.45	0.37	0.5	0.0	0.0	0.68	20.0	50.0	50.0
3.0	11115...	Angga...	11.0	3.4	3.14	3.26	3.63	3.5	3.84	129.0	40.0	30.0
4.0	11115...	Adhiti...	11.0	3.4	2.77	3.42	3.5	3.77	3.88	138.0	20.0	20.0
5.0	11115...	Abdul ...	11.0	3.75	3.45	3.29	3.71	3.5	3.42	138.0	20.0	20.0

Kriteria

Jumlah K 20 Proses Bersih

Gambar 4.1.5 Implementasi *Form* Jamak

3) Implementasi *Form Tunggal* Dengan *Message Box*

Pada gambar 4.1.6 merupakan tampilan menggunakan *message box*, *message box* akan muncul jika seluruh proses pada *form tunggal* telah selesai.

Form Tunggal

Pilih File Training

Ambil File

Documents\Kumpulan\2010_15112015_1616.xlsx

Cari File Tampilkan

Pilih File Testing

Ambil File

Documents\Kumpulan\2011_15112015_1617.xls

Cari File Tampilkan

Tabel Data Training

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET
1.0	10115...	Akbar...	11.0	3.04	2.95	2.38	2.63	2.71	3.11	124.0	40.0
2.0	10115...	Erwin...	11.0	3.42	3.58	4.0	3.75	3.77	3.52	143.0	20.0
3.0	10115...	Dwi Y...	11						08	60.0	50.0
4.0	10115...	Yohan...	11						78	102.0	40.0

Tabel Data Testing

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET
1.0	11115...	Iskan...	11						0	32.0	50.0
2.0	11115...	Edwin...	11						68	20.0	50.0
3.0	11115...	Angga...	11						84	129.0	40.0
4.0	11115...	Adhiti...	11.0	3.4	2.77	3.42	3.5	3.77	3.88	138.0	20.0
5.0	11115...	Abdul...	11.0	3.75	3.45	3.20	3.71	3.5	3.42	138.0	20.0

Informasi Hasil Testing

-Proses Testing Telah Selesai-
Waktu = 15 Milidetik

OK

Kriteria

Tipe Kriteria: Random

Total SKS: 72.0

IPS 1: 1.85

IPS 2: 2.65

NIM: 1111504104

IPS 3: 2.4

IPS 4: 2.0

IPS 5: 2.0

IPS 6: 2.94

Jumlah K: 20

Status: Tidak Lulus/DO

Prediksi : Tidak Lulus/DO

Proses Bersih

Gambar 4.1.6 Implementasi *Form Tunggal* Dengan *Message Box*

4) Implementasi *Form* Jamak Dengan *Message Box*

Pada gambar 4.1.7 merupakan tampilan menggunakan *message box*, *message box* akan muncul jika seluruh proses pada *form* jamak telah selesai.

The screenshot shows the 'Form Jamak' application window. It contains two file selection sections at the top: 'Pilih File Training' and 'Pilih File Testing', each with an 'Ambil File' button and a file path input field. Below these are two data tables: 'Tabel Data Training' and 'Tabel Data Testing'. A message box titled 'Informasi Hasil Prediksi' is overlaid on the tables, displaying the following information:

- Proses Testing Telah Selesai-
- Data Testing = 376
- Tepat Prediksi = 253
- Akurasi Prediksi = 67.29
- Waktu = 3 Detik

At the bottom of the form is a 'Kriteria' section with a 'Jumlah K' input field set to 20, and 'Proses' and 'Bersih' buttons.

No	NIM	Nama	JK	IPS 1	IPS 2	IPS 3	IPS 4	IPS 5	IPS 6	SKS	KET
1.0	10115...	Akbar ...	11.0	3.04	2.95	2.38	2.63	2.71	3.11	124.0	40.0
2.0	10115...	Erwin ...	11.0						3.52	143.0	20.0
3.0	10115...	Dwi Yo...	11.0						2.08	60.0	50.0
4.0	10115...	Yohan...	11.0						2.78	102.0	40.0
5.0	10115...	Andria...	11.0						2.76	101.0	40.0

No	NIM	Nama	JK	IPS 6	SKS	KET
1.0	11115...	Iskand...	11.0	0.0	32.0	50.0
2.0	11115...	Edwin ...	11.0	0.68	20.0	50.0
3.0	11115...	Angga ...	11.0	3.84	129.0	40.0
4.0	11115...	Adhiti...	11.0	3.88	138.0	20.0
5.0	11115...	Abdul ...	11.0	3.42	138.0	20.0

No.	NIM	Nama	JK	IPS1	IPS2	IPS3	IPS4	IPS5	IPS6	SKS	KET	PRED...
1.0	11115...	Iskan...	11.0	2.95	0.0	0.0	0.0	0.0	0.0	32.0	50.0	50.0
2.0	11115...	Edwin...	11.0	2.45	0.37	0.5	0.0	0.0	0.68	20.0	50.0	50.0
3.0	11115...	Angga...	11.0	3.4	3.14	3.26	3.63	3.5	3.84	129.0	40.0	30.0
4.0	11115...	Adhiti...	11.0	3.4	2.77	3.42	3.5	3.77	3.88	138.0	20.0	20.0
5.0	11115...	Abdul ...	11.0	3.75	3.45	3.29	3.71	3.5	3.42	138.0	20.0	20.0

Gambar 4.1.7 Implementasi *Form* Jamak Dengan *Message Box*

2. Hasil Uji Program

Pada tahap ini, merupakan penyajian dalam bentuk tabel mengenai pembahasan pengujian program terhadap kasus yaitu :

a. Pengujian Tunggal

Pengujian dengan tunggal, hanya menggunakan tipe kriteria *Random*. Konsep perbandingan pada pengujian tunggal yaitu sebanyak data *training* berbanding dengan 1 (satu) data *testing*. Total data yang digunakan untuk uji coba, sebanyak 60 data, yang mana 60 data tersebut dibagi menjadi 6 (enam) untuk masing-masing nilai K nya. Sehingga pada saat pengujian untuk masing-masing K, diuji sebanyak 10 kali. Total pengujian tunggal yaitu sebanyak 360 kali. Dengan melakukan pengujian tunggal, maka nilai akurasi terbesar adalah 70% dan rata-rata akurasi sebesar 61.11%.

Tabel 4.2.1 Pengujian Tunggal

No	Angka tan Traini ng	Angkata n Testing	Nilai K	Sub Percobaa n	Akurasi Per Sub (%)	Akurasi Rata-Rata Per Sub (%)	Akurasi Keseluruha n (%)
1.	2010	2011	5	1 – 10	90	70	61.11
			10	11 – 20	50		
			15	21 – 30	60		
			20	31 – 40	70		
			25	41 – 50	80		
			30	51 – 60	70		
2.	2011	2010	5	1 – 10	70	70	
			10	11 – 20	50		
			15	21 – 30	80		
			20	31 – 40	70		
			25	41 – 50	60		
			30	51 - 60	90		
3.	2009	2010	5	1 – 10	60	68.33	
			10	11 – 20	70		
			15	21 – 30	70		
			20	31 – 40	80		
			25	41 – 50	50		
			30	51 - 60	80		

Tabel 4.2.2 Pengujian Tunggal Lanjutan

	Angkatan Training	Angkatan Testing	Nilai K	Sub Percobaan	Akurasi Per Sub (%)	Akurasi Rata-Rata Per Sub (%)	Akurasi Keseluruhan (%)
4.	2009	2011	5	1 – 10	30	53.33	
			10	11 – 20	70		
			15	21 – 30	60		
			20	31 – 40	60		
			25	41 – 50	50		
			30	51 – 60	50		
5.	2010	2009	5	1 – 10	70	60	
			10	11 – 20	50		
			15	21 – 30	50		
			20	31 – 40	60		
			25	41 – 50	50		
			30	51 - 60	80		
6.	2011	2009	5	1 – 10	40	45	
			10	11 – 20	50		
			15	21 – 30	70		
			20	31 – 40	20		
			25	41 – 50	60		
			30	51 - 60	30		

b. Pengujian Jamak

Pada pengujian jamak ini, memiliki konsep banyak data *training* berbanding dengan banyak data *testing*. Pada pengujian jamak, total percobaan dilakukan sebanyak 36 kali. Dengan pengujian jamak, maka nilai akurasi terbesar adalah 75.36% dan rata-rata akurasi sebesar 61.88%.

Tabel 4.2.3 Pengujian Jamak

No	Angkat an Trainin g	Angkata n Testing	Nilai K	Sub Percobaa n	Akurasi Per Sub (%)	Akurasi Rata-Rata Per Sub (%)	Akurasi Keseluruh an (%)
1.	2010	2011	5	1	66.49	66.62	
			10	2	66.49		
			15	3	67.55		
			20	4	67.29		
			25	5	65.96		
			30	6	65.96		
2.	2011	2010	5	1	73.57	75.36	
			10	2	73.81		
			15	3	76.19		
			20	4	76.19		
			25	5	76.19		

No	Angkat an Trainin g	Angkata n Testing	Nilai K	Sub Percobaa n	Akurasi Per Sub (%)	Akurasi Rata-Rata Per Sub (%)	Akurasi Keseluruh an (%)
			30	6	76.19		61.88
3.	2009	2010	5	1	63.10	72.42	
			10	2	73.81		
			15	3	74.05		
			20	4	74.52		
			25	5	74.52		
			30	6	74.52		
4.	2009	2011	5	1	38.83		
			10	2	49.73		
			15	3	52.39		
			20	4	54.79		
			25	5	52.93		
			30	6	53.72		
5.	2010	2009	5	1	65.34	64.01	
			10	2	64.71		
			15	3	64.29		
			20	4	63.24		
			25	5	63.45		
			30	6	63.03		
6.	2011	2009	5	1	39.71		
			10	2	40.13		
			15	3	43.91		
			20	4	43.91		
			25	5	43.49		
			30	6	43.49		

3. Evaluasi Program

Pada tahap ini, akan menjabarkan hasil uji coba program dengan berbagai jumlah data dan nilai K, maka perlu adanya evaluasi program yang telah diuji. Berdasarkan hasil uji coba, terdapat hasil sebagai berikut :

a. Kelebihan Program

- 1) Aplikasi dapat digunakan dengan mudah.
- 2) Ukuran berkas file *excel*/tidak ada batasan pada saat melakukan *upload*.
- 3) Aplikasi dapat diinstall pada komputer atau laptop dengan mudah.
- 4) Waktu yang diperlukan untuk melakukan prediksi termasuk cepat, yaitu kurang dari 10 detik pada pengujian jamak dan kurang dari 100 milidetik pada pengujian tunggal.
- 5) Aplikasi hanya menggunakan *memory* sebesar, 15.2 MB dari total 4 GB.

b. Kekurangan Program

- 1) Aplikasi berbasis desktop, sehingga hanya bisa digunakan pada komputer atau laptop yang sudah diinstall.
- 2) Aplikasi hanya menerima berkas dari *file excel* saja, ketika melakukan *upload*.

BAB V PENUTUP

1. Kesimpulan

Sesuai dengan pembahasan mengenai prediksi kelulusan mahasiswa berbasis *desktop* menggunakan algoritma KNN, maka kesimpulan yang dapat diambil oleh penulis adalah sebagai berikut:

- a. Pada akurasi pengujian dengan menggunakan nilai K yang berbeda pada setiap percobaan baik tunggal maupun jamak adalah sebagai berikut :
 - 1) Percobaan Tunggal
Mempunyai nilai akurasi terbesar dengan nilai 70% dan, ketika menggunakan data *training* pada angkatan 2010 dan data *testing* pada angkatan 2011. Nilai rata-rata akurasi setelah melakukan pengujian sebesar 61.11%.
 - 2) Percobaan Jamak
Mempunyai nilai akurasi terbesar dengan nilai 75.36%, ketika menggunakan data *training* pada angkatan 2011 dan data *testing* pada angkatan 2010. Nilai rata-rata akurasi setelah melakukan pengujian sebesar 61.88%.
- b. Pada dasarnya aplikasi dapat digunakan untuk memprediksi semua fakultas dan program studi, namun pada penulisan kali ini diberi batasan hanya pada fakultas teknologi informasi dan program studi teknik informatika.
- c. Penerapan algoritma KNN dapat diimplementasikan untuk memprediksi tingkat kelulusan mahasiswa dengan 4 (empat) kategori yaitu Lulus Cepat, Lulus Tepat, Lulus Terlambat, dan Tidak Lulus/DO. Atribut yang paling berpengaruh dalam hal prediksi adalah IPS Semester 6 (enam) dan Total SKS hingga Semester 6 (Enam).
- d. Hasil prediksi kelulusan dari penulisan ini dapat membantu program studi untuk mengetahui kelulusan mahasiswa. Sehingga hal ini dapat, menjadi rekomendasi pengambilan mata kuliah bagi mahasiswa untuk semester berikutnya seperti skripsi dan kkp. Dengan hal tersebut mahasiswa bisa lulus minimal tepat waktu.
- e. Pada saat menggunakan data *training* angkatan 2011 dan data *testing* angkatan 2009, akurasi menjadi turun secara signifikan karena pada angkatan 2011 pada semester 6 sudah mengalami perubahan nilai *grading* sehingga kemungkinan nilai IPS nya lebih dari 3 sangatlah besar, daripada angkatan 2009 pada semester 6 yang kebanyakan bernilai 0.
- f. Dengan menggunakan Algoritma KNN dan bahasa pemrograman Java, dapat digunakan untuk merancang aplikasi Data *Mining* untuk memprediksi kelulusan mahasiswa sehingga dapat menampilkan informasi tingkat kelulusan secara efektif.
- g. Dengan menggunakan berkas *file* terbesar yaitu pada angkatan 2009 sebagai pelatihan, lalu diujikan pada angkatan 2010 waktu yang diperlukan 4 detik dan pada angkatan 2011 waktu yang diperlukan 3 detik.
- h. Jelas sangat dibutuhkan IPS semester 1 hingga 6 karena mahasiswa yang memiliki pola IPS semakin baik atau stabil selama 6 semester mempunyai peluang lebih besar untuk lulus daripada hanya melihat semester 6 saja yang mungkin selama semester 1 hingga 5 IPS nya menurun, yang menandakan mahasiswa sangat kewalahan untuk bisa lulus.
- i. Hasil prediksi bisa meleset, karena dipengaruhi oleh faktor-faktor lain yang tidak dapat dijelaskan seperti gaya belajar (IPS makin baik akan tetapi, motivasi untuk lulus tepat waktu tidak tercapai, misalnya karena bekerja sebelum lulus atau menikah sehingga menjadi tidak fokus menyelesaikan studi), faktor psikologis (bahwa mahasiswa tersebut memiliki ipk tinggi tapi dipengaruhi pergaulan dan

solidaritas kepada teman satu angkatan yang juga belum lulus sehingga mahasiswa yang memiliki ipk tinggi ikut menunda kelulusan), faktor perbaikan nilai yang membuat waktu studi lebih lama.

2. Saran

Tugas akhir ini masih jauh dari sempurna dan masih perlu banyak perbaikan dan pengembangan supaya menjadi lebih baik lagi. Berikut ini saran untuk pengembang tugas akhir ini :

- a. Menggunakan algoritma selain KNN, untuk mengetahui tingkat perbedaan akurasi.
- b. Menambahkan atribut gaji orang tua dan pekerjaan orang tua, untuk menjadi acuan motivasi belajar mahasiswa tersebut.
- c. Menambahkan atribut mengikuti organisasi atau tidak, untuk mengetahui tingkat kesibukan mahasiswa selain kuliah.
- d. Menambahkan atribut anak ke berapa dan jumlah bersaudara, untuk mengetahui motivasi belajar mahasiswa tersebut.
- e. Penelitian ini dapat diimplementasikan oleh pengelola kampus, sehingga dapat diambil tindakan preventif terhadap mahasiswa yang diprediksi kelulusannya tidak tepat waktu.
- f. Dilakukannya pengembangan, sehingga sumber data tidak hanya berasal dari *file excel*/tapi juga bisa dari *database* seperti Oracle, MySQL, dan lainnya.
- g. Diharapkan data seperti nilai IPS 1 hingga 6 tidak bernilai 0, terutama pada IP semester 6.

DAFTAR PUSTAKA

- Gorunescu, F., 2011. *Data Mining: Concepts, Models and Techniques*, New York: Springer-Verlag.
- Huda, N.M., 2010. Aplikasi Data Mining Untuk Menampilkan Informasi Tingkat Kelulusan Mahasiswa.
- Peterson, L.E., 2009. K-Nearest Neighbor. Available at: http://www.scholarpedia.org/article/K-nearest_neighbor#Acknowledgments [Diakses November 11, 2015].
- Prasetyo, E., 2014. *Data Mining Mengolah Data Menjadi Informasi Menggunakan Matlab* S. Erwin, ed., Penerbit Andi.
- Prasetyo, E., 2013. *Konsep dan Aplikasi Menggunakan Matlab*, Yogyakarta: Andi Offset.
- Ridwan, M., Suyono, H. & Sarosa, M., 2013. Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier. , 7(1), pp.59–64.
- Sayad, S., 2010. K Nearest Neighbors. Available at: http://www.saedsayad.com/k_nearest_neighbors_reg.htm [Diakses November 12, 2015].
- Steinbach, T.P. & Kumar, M., 2006. *Introduction to Data Mining*, New York: Pearson Education.
- X, W. & Kumar, 2009. *The Top Ten Algorithms In Data Mining* C. P. T. & F. Group, ed., London.

LAMPIRAN-LAMPIRAN**Lampiran Surat Riset**

KAMPUS PUSAT : Jl. Raya Ciledug, Petukangan Utara, Jakarta Selatan 12260
Telp : (021) 5853753 (Hunting), (021) 7371164, Fax : (021) 5853753
Website : <http://www.budiluhur.ac.id>

SURAT KETERANGAN RISET
NOMOR : S/UBL/FTI/000/058/10/15

Yang bertandatangan dibawah ini :

NIP : 930008
NAMA : Goenawan Brotosaputro, S.Kom, M.Sc
JABATAN : Dekan Fakultas Teknologi Informasi

Menerangkan bahwa :

NIM : 1211501075
NAMA : Christian Yonathan Sillueta
PROGRAM STUDI : Teknik Informatika

Telah melaksanakan riset pada Fakultas Teknologi Informasi Universitas Budi Luhur sejak tanggal 11 September s/d 14 Oktober 2015 dengan baik.

Demikian surat keterangan ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

Jakarta, 26 Oktober 2015
Dekan Fakultas Teknologi Informasi
Universitas Budi Luhur


Goenawan Brotosaputro, S.Kom, M.Sc

KAMPUS ROXY MAS : Pusat Niaga Roxy Mas Blok E.2 No. 38-39 Telp : (021) 6328709, 6328710, Fax : (021) 6322872
KAMPUS SALEMBA MAS : Sentra Salemba Mas Blok S-T, Telp : (021) 3928688, 3928689, Fax : (021) 3161636

Lampiran Berita Acara Sidang



UNIVERSITAS BUDI LUHUR

FAKULTAS TEKNOLOGI INFORMASI | FAKULTAS EKONOMI | FAKULTAS ILMU SOSIAL & ILMU POLITIK
FAKULTAS TEKNIK | FAKULTAS ILMU KOMUNIKASI

Kampus Pusat : Jl. Raya Ciledug , Petukangan Utara - Jakarta Selatan 12260
Telp : 021-5853753 (hunting), Fax : 021-5853489, http://www.budiluhur.ac.id

BERITA ACARA SIDANG PENDADARAN TUGAS AKHIR

S/UBL/FTI/0038/XII/15

Pada hari ini, Senin tanggal 28 Desember 2015 telah dilaksanakan Ujian Sidang Pendadaran Tugas Akhir sebagai berikut :

Judul : IMPLEMENTASI DATA MINING UNTUK MEMPREDIKSI KELULUSAN MAHASISWA DENGAN METODE KLASIFIKASI DAN ALGORITMA K-NEAREST NEIGHBOR BERBASIS DESKTOP
(STUDI KASUS : FAKULTAS TEKNOLOGI INFORMASI, PROGRAM STUDI TEKNIK INFORMATIKA - UNIVERSITAS BUDI LUHUR)

Nama : Christian Yonathan Sillueta
NIM : 1211501075
Dosen Pembimbing : Muhammad Ainur Rony, S.Kom., M.T.I

Berdasarkan penilaian pada Penyajian, Penulisan, Program dan Penguasaan Materi, Maka Mahasiswa tersebut di atas dinyatakan :

L U L U S

dengan nilai angka : 91 huruf : A

Mahasiswa tersebut di atas wajib menyerahkan hasil perbaikan tulisan Tugas Akhir dalam bentuk terjilid sesuai dengan Panduan Perbaikan Tugas Akhir, selambat-lambatnya tanggal 11 Januari 2016.

Panitia Penguji :

1. Ketua

(M. Syafrullah, M.Kom, M.Sc)
2. Anggota

(Purwanto, S.Si, M.Kom)
3. Moderator :

(Muhammad Ainur Rony, S.Kom., M.T.I)

Keterangan :

Nilai huruf : A: 85-100 A-: 80-84,99 B+: 75-79,99 B: 70-74,99 B-: 65-69,99
C: 60-64,99 D: 45-59,99 E: 0-44,99

LISTING PROGRAM

FormUtama.java

```

package form1;

import java.awt.Dimension;
import java.awt.Toolkit;

public class FormUtama extends javax.swing.JFrame {

    public FormUtama() {
        initComponents();
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        setLocation(dim.width / 2 - getWidth() / 2, dim.height / 2 - getHeight() / 2);
    }

    // <editor-fold    defaultstate="collapsed"    desc="Generated    Code"> //GEN-BEGIN: initComponents
    private void initComponents() {

        tombolFormJamak = new javax.swing.JButton();
        tombolFormTunggal = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Form Utama");
        setResizable(false);

        tombolFormJamak.setText("Form Jamak");
        tombolFormJamak.setName(""); // NOI18N
        tombolFormJamak.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                tombolFormJamakActionPerformed(evt);
            }
        });

        tombolFormTunggal.setText("Form Tunggal");
        tombolFormTunggal.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                tombolFormTunggalActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
        javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(tombolFormJamak,
javax.swing.GroupLayout.PREFERRED_SIZE,                207,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(tombolFormTunggal,
javax.swing.GroupLayout.PREFERRED_SIZE,                207,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(tombolFormJamak,
javax.swing.GroupLayout.PREFERRED_SIZE,                122,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(tombolFormTunggal,
javax.swing.GroupLayout.PREFERRED_SIZE,                122,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

private void tombolFormJamakActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_tombolFormJamakActionPerformed
    FormJamak FJ = new FormJamak();
    FJ.show(true);
} // GEN-LAST: event_tombolFormJamakActionPerformed

private void tombolFormTunggalActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_tombolFormTunggalActionPerformed
    FormTunggal FT = new FormTunggal();
    FT.show(true);
} // GEN-LAST: event_tombolFormTunggalActionPerformed

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new FormUtama().setVisible(true);
        }
    }
}

```

```
    });  
}  
  
// Variables declaration - do not modify//GEN-BEGIN:variables  
private javax.swing.JButton tombolFormJamak;  
private javax.swing.JButton tombolFormTunggal;  
// End of variables declaration//GEN-END:variables  
}
```

FormTunggal.java

```

package form1;

import java.awt.Color;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.text.DecimalFormat;
import java.util.Random;
import javax.swing.JComboBox;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.table.DefaultTableModel;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.xssf.usermodel.XSSFCell;
import org.apache.poi.xssf.usermodel.XSSFRow;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class FormTunggal extends javax.swing.JFrame {

    private DecimalFormat df = new DecimalFormat("#.####");
    private Random RA = new Random();

    public FormTunggal() {
        initComponents();
        this.setSize(620, 615);
        this.setLocationRelativeTo(null);
        comboboxKriteria.addItem("Random");
        comboboxKriteria.addItem("Input");
        comboboxKriteria.setSelectedIndex(0);
        comboboxK.addItem(5);
        comboboxK.addItem(10);
        comboboxK.addItem(15);
        comboboxK.addItem(20);
        comboboxK.addItem(25);
        comboboxK.addItem(30);
        comboboxK.setSelectedIndex(-1);
        comboboxStatus.addItem("Lulus Cepat");
        comboboxStatus.addItem("Lulus Tepat");
        comboboxStatus.addItem("Lulus Terlambat");
        comboboxStatus.addItem("Tidak Lulus/DO");
        comboboxStatus.addItem("Tidak Ada Hasil");
        comboboxStatus.setSelectedIndex(-1);
        labelKeputusan.setText("Prediksi :");
    }

```

```

    }

    @SuppressWarnings("unchecked")
    // <editor-fold    defaultstate="collapsed"    desc="Generated    Code"> //GEN-BEGIN: initComponents
    private void initComponents() {

        jPanel2 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        textIsiFileTraining = new javax.swing.JTextField();
        tombolTampilkanTraining = new javax.swing.JButton();
        tombolCariFileTraining = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        textIsiFileTesting = new javax.swing.JTextField();
        tombolTampilkanTesting = new javax.swing.JButton();
        tombolCariFileTesting = new javax.swing.JButton();
        jScrollPane4 = new javax.swing.JScrollPane();
        tabelTraining = new javax.swing.JTable();
        jPanel3 = new javax.swing.JPanel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        comboboxKriteria = new javax.swing.JComboBox();
        textSKS = new javax.swing.JTextField();
        textIPS1 = new javax.swing.JTextField();
        textIPS2 = new javax.swing.JTextField();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        textIPS3 = new javax.swing.JTextField();
        textIPS4 = new javax.swing.JTextField();
        textIPS5 = new javax.swing.JTextField();
        textIPS6 = new javax.swing.JTextField();
        textNIM = new javax.swing.JTextField();
        jLabel11 = new javax.swing.JLabel();
        comboboxK = new javax.swing.JComboBox();
        jLabel12 = new javax.swing.JLabel();
        comboboxStatus = new javax.swing.JComboBox();
        tombolProses = new javax.swing.JButton();
        tombolBersih = new javax.swing.JButton();
        labelKeputusan = new javax.swing.JLabel();
        jLabel15 = new javax.swing.JLabel();
        jScrollPane3 = new javax.swing.JScrollPane();
        tabelTesting = new javax.swing.JTable();
        jLabel13 = new javax.swing.JLabel();
        jLabel14 = new javax.swing.JLabel();

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Form Tunggal");
setResizable(false);
getContentPane().setLayout(null);

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder("Pilih File
Training"));
jPanel2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jLabel2.setText("Ambil File");

tombolTampilkanTraining.setText("Tampilkan");
tombolTampilkanTraining.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tombolTampilkanTrainingActionPerformed(evt);
    }
});

tombolCariFileTraining.setText("Cari File");
tombolCariFileTraining.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tombolCariFileTrainingActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(tombolCariFileTraining)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(tombolTampilkanTraining)
                .addGap(10, 10, 10)
                .addComponent(textIsiFileTraining)
            )
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(tombolTampilkanTraining)
                .addGap(10, 10, 10)
                .addComponent(textIsiFileTraining)
            )
        )
        .addContainerGap())
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(tombolCariFileTraining)
        .addComponent(tombolTampilkanTraining)
        .addComponent(textIsiFileTraining)
    )
);
jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(tombolCariFileTraining)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(tombolTampilkanTraining)
                .addGap(10, 10, 10)
                .addComponent(textIsiFileTraining)
            )
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(tombolTampilkanTraining)
                .addGap(10, 10, 10)
                .addComponent(textIsiFileTraining)
            )
        )
        .addContainerGap())
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(tombolCariFileTraining)
        .addComponent(tombolTampilkanTraining)
        .addComponent(textIsiFileTraining)
    )
);
jPanel2Layout.setVerticalGroup(

```



```

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(jLabel2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(textIsiFileTraining,
javax.swing.GroupLayout.PREFERRED_SIZE, 37, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(tombolTampilkanTraining)
        .addComponent(tombolCariFileTraining))
        .addContainerGap())
    );

    getContentPane().add(jPanel2);
    jPanel2.setBounds(10, 10, 290, 131);

    jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Piilh           File
Testing"));
    jPanel1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

    jLabel1.setText("Ambil File");

    tombolTampilkanTesting.setText("Tampilkan");
    tombolTampilkanTesting.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            tombolTampilkanTestingActionPerformed(evt);
        }
    });

    tombolCariFileTesting.setText("Cari File");
    tombolCariFileTesting.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            tombolCariFileTestingActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(textIsiFileTesting)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addComponent(tombolCariFileTesting)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(tombolTampilkanTesting))

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addContainerGap(120, Short.MAX_VALUE)
        .addComponent(jLabel1)
        .addGap(114, 114, 114))
    );
jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(textIsiFileTesting,
javax.swing.GroupLayout.PREFERRED_SIZE, 37, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
    )
        .addComponent(tombolTampilkanTesting)
        .addComponent(tombolCariFileTesting))
        .addContainerGap())
    );

getContentPane().add(jPanel1);
jPanel1.setBounds(300, 10, 290, 131);

tabelTraining.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null, null}
    },
    new String [] {
        "No", "NIM", "Nama", "JK", "IPS 1", "IPS 2", "IPS 3", "IPS 4", "IPS 5", "IPS
6", "SKS", "KET"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
jScrollPane4.setViewportViewView(tabelTraining);

```

```

getContentPane().add(jScrollPane4);
jScrollPane4.setBounds(10, 160, 580, 93);

jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder("Kriteria"));

jLabel3.setText("Tipe Kriteria");

jLabel4.setText("Total SKS");

jLabel5.setText("IPS 1");

jLabel6.setText("IPS 2");

jLabel7.setText("IPS 3");

comboBoxKriteria.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        comboBoxKriteriaActionPerformed(evt);
    }
});

jLabel8.setText("IPS 4");

jLabel9.setText("IPS 5");

jLabel10.setText("IPS 6");

jLabel11.setText("Jumlah K");

jLabel12.setText("Status");

tombolProses.setText("Proses");
tombolProses.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tombolProsesActionPerformed(evt);
    }
});

tombolBersih.setText("Bersih");
tombolBersih.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tombolBersihActionPerformed(evt);
    }
});

jLabel15.setText("NIM");

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);

```

```

jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup())

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addGroup(jPanel3Layout.createSequentialGroup())
        .addComponent(jLabel15)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(textNIM, javax.swing.GroupLayout.PREFERRED_SIZE,
82, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(jPanel3Layout.createSequentialGroup())
        .addComponent(jLabel3)
        .addGap(53, 53, 53)
        .addComponent(comboboxKriteria,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(jPanel3Layout.createSequentialGroup())

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel4)
    .addComponent(jLabel5)
    .addComponent(jLabel6))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
    .addComponent(textSKS, javax.swing.GroupLayout.DEFAULT_SIZE,
46, Short.MAX_VALUE)
    .addComponent(textIPS1)
    .addComponent(textIPS2))))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addGroup(jPanel3Layout.createSequentialGroup())
        .addComponent(jLabel9)
        .addGap(18, 18, 18)
        .addComponent(textIPS5))
    .addGroup(jPanel3Layout.createSequentialGroup())
        .addComponent(jLabel7)

```

```

        .addGap(18, 18, 18)
        .addComponent(textIPS3,
javax.swing.GroupLayout.PREFERRED_SIZE,          55,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel3Layout.createSequentialGroup())
        .addComponent(jLabel8)
        .addGap(18, 18, 18)
        .addComponent(textIPS4,
javax.swing.GroupLayout.PREFERRED_SIZE,          55,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel3Layout.createSequentialGroup())
        .addComponent(jLabel10)
        .addGap(18, 18, 18)
        .addComponent(textIPS6)))
        .addComponent(tombolProses,
javax.swing.GroupLayout.PREFERRED_SIZE,          83,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(comboboxK, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel11)
        .addComponent(jLabel12)
        .addComponent(comboboxStatus,
javax.swing.GroupLayout.PREFERRED_SIZE,          javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tombolBersih,
javax.swing.GroupLayout.PREFERRED_SIZE, 82, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(labelKeputusan,
javax.swing.GroupLayout.PREFERRED_SIZE,          178,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(130, Short.MAX_VALUE))
    );
    jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup())
    .addContainerGap(14, Short.MAX_VALUE)

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup())
    .addComponent(jLabel11)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(comboboxK,
javax.swing.GroupLayout.PREFERRED_SIZE,          javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jLabel12)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(comboboxStatus,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(jPanel3Layout.createSequentialGroup())

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
    .addComponent(jLabel3)
    .addComponent(jLabel7)
    .addComponent(comboboxKriteria,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(textIPS3,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(10, 10, 10)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup())

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
    .addComponent(jLabel8)
    .addComponent(textIPS4,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(15, 15, 15)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
    .addComponent(jLabel9)
    .addComponent(textIPS5,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(labelKeputusan,
javax.swing.GroupLayout.PREFERRED_SIZE, 19, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
    .addComponent(jLabel10)

```

```

        .addComponent(textIPS6,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
        .addComponent(tombolProses)
        .addComponent(tombolBersih)))
.addGroup(jPanel3Layout.createSequentialGroup())

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
        .addComponent(textSKS,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel4))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
        .addComponent(textIPS1,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel5))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
        .addComponent(textIPS2,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel6))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
        .addComponent(textNIM,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel15))))))
.addContainerGap(13, Short.MAX_VALUE)
);

getContentPane().add(jPanel3);

```

```

jPanel3.setBounds(10, 380, 580, 200);

tabelTesting.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null}
    },
    new String [] {
        "No", "NIM", "Nama", "JK", "IPS 1", "IPS 2", "IPS 3", "IPS 4", "IPS 5", "IPS
6", "SKS", "KET"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
jScrollPane3.setViewportViewView(tabelTesting);

getContentPane().add(jScrollPane3);
jScrollPane3.setBounds(10, 280, 580, 94);

jLabel13.setText("Tabel Data Training");
getContentPane().add(jLabel13);
jLabel13.setBounds(10, 140, 140, 14);

jLabel14.setText("Tabel Data Testing");
getContentPane().add(jLabel14);
jLabel14.setBounds(10, 260, 160, 14);

pack();
}

private void tombolTampilkanTrainingActionPerformed(java.awt.event.ActionEvent
evt) {
    try {
        if (textIsiFileTraining.getText().equals("")) {
            JOptionPane.showMessageDialog(this, "Belum Memilih File Excel", "Informasi
Kesalahan Upload File", JOptionPane.ERROR_MESSAGE);
        } else {
            if (textIsiFileTraining.getText().endsWith(".xls")) {
                ekstensiXLSTraining();
            } else if (textIsiFileTraining.getText().endsWith(".xlsx")) {
                ekstensiXLSXTraining();
            }
        }
    }
}

```



```

        } else {
            JOptionPane.showMessageDialog(this, "Ekstensi Tidak Diketahui",
                "Informasi Kesalahan Upload File", JOptionPane.ERROR_MESSAGE);
        }
    }
} catch (Exception er1) {

}

} //GEN-LAST:event_tombolTampilkanTrainingActionPerformed

private void tombolCariFileTrainingActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_tombolCariFileTrainingActionPerformed
    try {
        JFileChooser JFC = new JFileChooser();
        FileNameExtensionFilter fFilter = new FileNameExtensionFilter("Excel", "xls",
            "xlsx");
        JFC.setFileFilter(fFilter);

        if (JFC.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
            textIsiFileTraining.setText(JFC.getSelectedFile().getAbsolutePath());
        }
    } catch (Exception er1) {
        JOptionPane.showMessageDialog(this, "Kesalahan Upload Excel", "Informasi
            Kesalahan Upload", JOptionPane.WARNING_MESSAGE);
    }
} //GEN-LAST:event_tombolCariFileTrainingActionPerformed

private void tombolTampilkanTestingActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_tombolTampilkanTestingActionPerformed
    try {
        if (textIsiFileTesting.getText().equals("")) {
            JOptionPane.showMessageDialog(this, "Belum Memilih File Excel", "Informasi
                Kesalahan Upload File", JOptionPane.ERROR_MESSAGE);
        } else {
            if (textIsiFileTesting.getText().endsWith(".xls")) {
                ekstensiXLSTesting();
            } else if (textIsiFileTesting.getText().endsWith(".xlsx")) {
                ekstensiXLSXTesting();
            } else {
                JOptionPane.showMessageDialog(this, "Ekstensi Tidak Diketahui",
                    "Informasi Kesalahan Upload File", JOptionPane.ERROR_MESSAGE);
            }
        }
    } catch (Exception er1) {

    }
} //GEN-LAST:event_tombolTampilkanTestingActionPerformed

```

```

private void tombolCariFileTestingActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_tombolCariFileTestingActionPerformed
    try {
        JFileChooser JFC = new JFileChooser();
        FileNameExtensionFilter fFilter = new FileNameExtensionFilter("Excel", "xls",
"xlsx");
        JFC.setFileFilter(fFilter);

        if (JFC.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
            textIsiFileTesting.setText(JFC.getSelectedFile().getAbsolutePath());
        }
    } catch (Exception er1) {
        JOptionPane.showMessageDialog(this, "Kesalahan Upload Excel", "Informasi
Kesalahan Upload", JOptionPane.WARNING_MESSAGE);
    }
}
//GEN-LAST:event_tombolCariFileTestingActionPerformed

private void tombolProsesActionPerformed(java.awt.event.ActionEvent evt) {
    //GEN-FIRST:event_tombolProsesActionPerformed
    if (comboBoxK.getSelectedIndex() == -1
        || (Double.parseDouble(textSKS.getText()) < 0
        || Double.parseDouble(textSKS.getText()) > 175)
        || textIPS1.getText().equals("") || textIPS2.getText().equals("")
        || textIPS3.getText().equals("") || textIPS4.getText().equals("")
        || textIPS5.getText().equals("") || textIPS6.getText().equals("")
        || textNIM.getText().equals("")) {
        JOptionPane.showMessageDialog(this, "Form Masih Kosong Atau Tidak Valid",
"Informasi Kesalahan Input", JOptionPane.WARNING_MESSAGE);
        bersih();
    } else {
        if ((Double.parseDouble(textIPS1.getText()) >= 0 &&
Double.parseDouble(textIPS1.getText()) <= 4)
        && (Double.parseDouble(textIPS2.getText()) >= 0 &&
Double.parseDouble(textIPS2.getText()) <= 4)
        && (Double.parseDouble(textIPS3.getText()) >= 0 &&
Double.parseDouble(textIPS3.getText()) <= 4)
        && (Double.parseDouble(textIPS4.getText()) >= 0 &&
Double.parseDouble(textIPS4.getText()) <= 4)
        && (Double.parseDouble(textIPS5.getText()) >= 0 &&
Double.parseDouble(textIPS5.getText()) <= 4)
        && (Double.parseDouble(textIPS6.getText()) >= 0 &&
Double.parseDouble(textIPS6.getText()) <= 4)) {

            long w1 = System.currentTimeMillis();

            prosesKNN_Euclidean(new double[][]{
                {
                    Double.parseDouble(textIPS1.getText()),
                    Double.parseDouble(textIPS2.getText()),

```

```

        Double.parseDouble(textIPS3.getText()),
        Double.parseDouble(textIPS4.getText()),
        Double.parseDouble(textIPS5.getText()),
        Double.parseDouble(textIPS6.getText()),
        Double.parseDouble(textSKS.getText()),
        90
    }
    });

    long w2 = System.currentTimeMillis(),
        w3 = (w2-w1);

    JOptionPane.showMessageDialog(this, "-Proses Testing Telah Selesai-\n"+
        "Waktu = "+w3+" Milidetik",
        "Informasi Hasil Testing",
        JOptionPane.INFORMATION_MESSAGE);
    } else {
        JOptionPane.showMessageDialog(this, "Nilai IPS 1 hingga 6, Tidak Valid!",
        "Informasi Kesalahan Input", JOptionPane.WARNING_MESSAGE);
        bersih();
    }
}
} //GEN-LAST:event_tombolProsesActionPerformed

private void tombolBersihActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_tombolBersihActionPerformed
    bersih();
} //GEN-LAST:event_tombolBersihActionPerformed

private void comboboxKriteriaActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_comboboxKriteriaActionPerformed
    if (tabelTesting.getRowCount() != 4 && comboboxKriteria.getSelectedIndex() != -1)
    {
        String JC = ((JComboBox) evt.getSource()).getSelectedItem().toString();
        if (JC.equals("Random")) {
            disabledEnabledKriteria(false, false, false, false, false, false, false, false,
            false, true, false);
            int TR = RA.nextInt(tabelTesting.getRowCount());

            double ips1 = Double.parseDouble(tabelTesting.getValueAt(TR,
            4).toString()),
                ips2 = Double.parseDouble(tabelTesting.getValueAt(TR, 5).toString()),
                ips3 = Double.parseDouble(tabelTesting.getValueAt(TR, 6).toString()),
                ips4 = Double.parseDouble(tabelTesting.getValueAt(TR, 7).toString()),
                ips5 = Double.parseDouble(tabelTesting.getValueAt(TR, 8).toString()),
                ips6 = Double.parseDouble(tabelTesting.getValueAt(TR, 9).toString()),
                sks = Double.parseDouble(tabelTesting.getValueAt(TR, 10).toString()),
                ket = Double.parseDouble(tabelTesting.getValueAt(TR,
            11).toString());

```

```

        textSKS.setText("" + sks);
        textIPS1.setText("" + ips1);
        textIPS2.setText("" + ips2);
        textIPS3.setText("" + ips3);
        textIPS4.setText("" + ips4);
        textIPS5.setText("" + ips5);
        textIPS6.setText("" + ips6);
        textNIM.setText(tabelTesting.getValueAt(TR, 1).toString());

        if (ket == 20) {
            comboboxStatus.setSelectedIndex(0);
        } else if (ket == 30) {
            comboboxStatus.setSelectedIndex(1);
        } else if (ket == 40) {
            comboboxStatus.setSelectedIndex(2);
        } else if (ket == 50) {
            comboboxStatus.setSelectedIndex(3);
        } else if (ket == 60) {
            comboboxStatus.setSelectedIndex(4);
        }
    } else {
        disabledEnabledKriteria(false, true, true, true, true, true, true, true, true,
true, true);
    }
}
}
} //GEN-LAST:event_comboboxKriteriaActionPerformed

void bersih() {
    disabledEnabledKriteria(true, true, true, true, true, true, true, true, true,
true);
    textSKS.setText("");
    textIPS1.setText("");
    textIPS2.setText("");
    textNIM.setText("");
    textIPS3.setText("");
    textIPS4.setText("");
    textIPS5.setText("");
    textIPS6.setText("");
    labelKeputusan.setText("");
    comboboxK.setSelectedIndex(-1);
    comboboxStatus.setSelectedIndex(-1);
    comboboxKriteria.setSelectedIndex(-1);
}

void prosesKNN_Euclidean(double[][] pNilaiTesting) {
    double[][] nilaiTraining = nilaiDefaultTraining();
    double[] nilaiKuadrat = nilaiKuadratEuclidean(pNilaiTesting, nilaiTraining);
}

```

```

        double[][] nilaiTrainingSorting = nilaiTrainingSortingEuclidian(nilaiKuadrat,
nilaiTraining);
        int jumlahK = Integer.parseInt(comboboxK.getSelectedItem().toString());

        double[][] nilaiTrainingK = nilaiTrainingSebanyakJumlahK(nilaiTrainingSorting,
jumlahK);
        int j20 = kemunculanKelas(nilaiTrainingK, 20),
            j30 = kemunculanKelas(nilaiTrainingK, 30),
            j40 = kemunculanKelas(nilaiTrainingK, 40),
            j50 = kemunculanKelas(nilaiTrainingK, 50),
            j60 = kemunculanKelas(nilaiTrainingK, 60),
            kelasMayoritas = nilaiMayoritasKelas(new int[]{j20, j30, j40, j50, j60});

        String cetak = "---Jumlah K = "+jumlahK+"---\n";
        cetak += "Jumlah 20 = "+j20+"\n";
        cetak += "Jumlah 30 = "+j30+"\n";
        cetak += "Jumlah 40 = "+j40+"\n";
        cetak += "Jumlah 50 = "+j50+"\n";
        cetak += "Jumlah 60 = "+j60+"\n";
        System.out.print(cetak);

        if (kelasMayoritas == 0) {
            labelKeputusan.setText("Prediksi : Lulus Cepat");
            labelKeputusan.setForeground(Color.BLUE);
        } else if (kelasMayoritas == 1) {
            labelKeputusan.setText("Prediksi : Lulus Tepat");
            labelKeputusan.setForeground(Color.BLUE);
        } else if (kelasMayoritas == 2) {
            labelKeputusan.setText("Prediksi : Lulus Terlambat");
            labelKeputusan.setForeground(Color.RED);
        } else if (kelasMayoritas == 3) {
            labelKeputusan.setText("Prediksi : Tidak Lulus/DO");
            labelKeputusan.setForeground(Color.RED);
        } else if (kelasMayoritas == 4) {
            labelKeputusan.setText("Prediksi : Tidak Ada Hasil");
            labelKeputusan.setForeground(Color.RED);
        }
    }

    void disabledEnabledTraining(boolean pText, boolean pCari, boolean pTampil) {
        textIsiFileTraining.enable(pText);
        tombolCariFileTraining.enable(pCari);
        tombolTampilkanTraining.enable(pTampil);
    }

    void disabledEnabledTesting(boolean pText, boolean pCari, boolean pTampil) {
        textIsiFileTesting.enable(pText);
        tombolCariFileTesting.enable(pCari);
        tombolTampilkanTesting.enable(pTampil);
    }

```

```

    }

    void disabledEnabledKriteria(boolean pKriteria, boolean pSKS, boolean pIPS1,
    boolean pIPS2, boolean pIPS3, boolean pIPS4, boolean pIPS5, boolean pIPS6, boolean
    pNIM, boolean pJumlahK, boolean pStatus) {
        comboboxKriteria.enable(pKriteria);
        textSKS.setEditable(pSKS);
        textIPS1.setEditable(pIPS1);
        textIPS2.setEditable(pIPS2);
        textIPS3.setEditable(pIPS3);
        textIPS4.setEditable(pIPS4);
        textIPS5.setEditable(pIPS5);
        textIPS6.setEditable(pIPS6);
        textNIM.setEditable(pNIM);
        comboboxK.enable(pJumlahK);
        comboboxStatus.enable(pStatus);
    }

    void ekstensiXLSTraining() {
        try {
            FileInputStream FIS = new FileInputStream(new
            File(textIsiFileTraining.getText()));
            HSSFWorkbook wb = new HSSFWorkbook(FIS);
            HSSFSheet ws = wb.getSheetAt(0);

            int baris = ws.getLastRowNum() + 1;
            int kolom = ws.getRow(0).getLastCellNum();
            String[][] data = new String[baris][kolom];
            String[] nKolom = new String[kolom];

            //ambil semua informasi yang berada pada sheet baik itu nama kolom maupun
            nilai pada baris nya
            for (int i = 0; i < baris; i++) {
                HSSFRow row = ws.getRow(i);
                for (int j = 0; j < kolom; j++) {
                    HSSFCell cell = row.getCell(j);
                    String value = nilaiCell(cell);
                    data[i][j] = value;
                }
            }

            //ambil nama kolom pada index data baris ke 0
            for (int a = 0; a < kolom; a++) {
                nKolom[a] = data[0][a];
            }

            DefaultTableModel DTM = new DefaultTableModel(data, nKolom);
            tabelTraining.setModel(DTM);
            DTM.removeRow(0);

```

```

    } catch (IOException er1) {
        System.out.print("Error : \n" + er1.toString());
    }
}

void ekstensiXLSTraining() {
    try {
        FileInputStream FIS = new FileInputStream(new
File(textIsiFileTraining.getText()));
        XSSFWorkbook wb = new XSSFWorkbook(FIS);
        XSSFSheet ws = wb.getSheetAt(0);

        int baris = ws.getLastRowNum() + 1;
        int kolom = ws.getRow(0).getLastCellNum();
        String[][] data = new String[baris][kolom];
        String[] nKolom = new String[kolom];

        for (int i = 0; i < baris; i++) {
            XSSFRow row = ws.getRow(i);
            for (int j = 0; j < kolom; j++) {
                XSSFCell cell = row.getCell(j);
                String value = nilaiCell(cell);
                data[i][j] = value;
            }
        }

        //ambil nama kolom pada index data baris ke 0
        for (int a = 0; a < kolom; a++) {
            nKolom[a] = data[0][a];
        }

        DefaultTableModel DTM = new DefaultTableModel(data, nKolom);
        tabelTraining.setModel(DTM);
        DTM.removeRow(0);
    } catch (Exception er1) {
        System.out.print("Error : \n" + er1.toString());
    }
}

void ekstensiXLSTesting() {
    try {
        FileInputStream FIS = new FileInputStream(new
File(textIsiFileTesting.getText()));
        HSSFWorkbook wb = new HSSFWorkbook(FIS);
        HSSFSheet ws = wb.getSheetAt(0);

        int baris = ws.getLastRowNum() + 1;
        int kolom = ws.getRow(0).getLastCellNum();
        String[][] data = new String[baris][kolom];

```

```

String[] nKolom = new String[kolom];

//ambil semua informasi yang berada pada sheet baik itu nama kolom maupun
nilai pada baris nya
for (int i = 0; i < baris; i++) {
    HSSFRow row = ws.getRow(i);
    for (int j = 0; j < kolom; j++) {
        HSSFCell cell = row.getCell(j);
        String value = nilaiCell(cell);
        data[i][j] = value;
    }
}

//ambil nama kolom pada index data baris ke 0
for (int a = 0; a < kolom; a++) {
    nKolom[a] = data[0][a];
}

DefaultTableModel DTM = new DefaultTableModel(data, nKolom);
tabelTesting.setModel(DTM);
DTM.removeRow(0);
} catch (IOException er1) {
    System.out.print("Error : \n" + er1.toString());
}
}

void ekstensiXLSXTesting() {
    try {
        FileInputStream FIS = new FileInputStream(new
File(textIsiFileTesting.getText()));
        XSSFWorkbook wb = new XSSFWorkbook(FIS);
        XSSFSheet ws = wb.getSheetAt(0);

        int baris = ws.getLastRowNum() + 1;
        int kolom = ws.getRow(0).getLastCellNum();
        String[][] data = new String[baris][kolom];
        String[] nKolom = new String[kolom];

        for (int i = 0; i < baris; i++) {
            XSSFRow row = ws.getRow(i);
            for (int j = 0; j < kolom; j++) {
                XSSFCell cell = row.getCell(j);
                String value = nilaiCell(cell);
                data[i][j] = value;
            }
        }

        //ambil nama kolom pada index data baris ke 0
        for (int a = 0; a < kolom; a++) {

```



```

        nKolom[a] = data[0][a];
    }

    DefaultTableModel DTM = new DefaultTableModel(data, nKolom);
    tabelTesting.setModel(DTM);
    DTM.removeRow(0);
} catch (Exception er1) {
    System.out.print("Error : \n" + er1.toString());
}
}

String nilaiCell(HSSFCell pCell) {
    int tipe = pCell.getCellType();
    Object nilaiBalik = null;

    if (tipe == 0) {
        nilaiBalik = pCell.getNumericCellValue();
    } else if (tipe == 1) {
        nilaiBalik = pCell.getStringCellValue();
    }

    return nilaiBalik.toString();
}

String nilaiCell(XSSFCell pCell) {
    int tipe = pCell.getCellType();
    Object nilaiBalik = null;

    if (tipe == 0) {
        nilaiBalik = pCell.getNumericCellValue();
    } else if (tipe == 1) {
        nilaiBalik = pCell.getStringCellValue();
    }

    return nilaiBalik.toString();
}

int nilaiMayoritasKelas(int[] pJumlahNilaiKelas) {
    int nilaiBalik = 0,
        nMax = pJumlahNilaiKelas[0];

    for (int i = 1; i < pJumlahNilaiKelas.length; i++) {
        if (nMax < pJumlahNilaiKelas[i]) {
            nMax = pJumlahNilaiKelas[i];
            nilaiBalik = i;
        }
    }

    return nilaiBalik;
}

```

```

    }

    int kemunculanKelas(double[][] pNilaiTraing, int pNilai) {
        int nilaiBalik = 0;
        for (int i = 0; i < pNilaiTraing.length; i++) {
            if (pNilaiTraing[i][7] == pNilai) {
                nilaiBalik++;
            }
        }
        return nilaiBalik;
    }

    double[][] nilaiDefaultTraining() {
        double[][] nilaiBalik = new double[tabelTraining.getRowCount()][8];

        int kolom = 0;
        for (int i = 0; i < tabelTraining.getRowCount(); i++) {
            for (int j = 4; j <= 11; j++) {
                nilaiBalik[i][kolom] = Double.parseDouble(tabelTraining.getValueAt(i, j).toString());
                kolom++;
            }
            kolom = 0;
        }
        return nilaiBalik;
    }

    double[][] nilaiTrainingSebanyakJumlahK(double[][] pNilaiTraining, int pJumlahK) {
        double[][] nilaiBalik = new double[pJumlahK][8];

        for (int i = 0; i < pJumlahK; i++) {
            for (int j = 0; j < 8; j++) {
                nilaiBalik[i][j] = pNilaiTraining[i][j];
            }
        }

        return nilaiBalik;
    }

    double[][] nilaiTrainingSortingEuclidian(double[] pNilaiKuadrat, double[][] pNilaiTraining) {
        double[][] tempTrain = new double[pNilaiTraining.length][8];
        double temp = 0;

        for (int i = 0; i < pNilaiTraining.length; i++) {
            for (int j = i + 1; j < pNilaiKuadrat.length; j++) {
                if (pNilaiKuadrat[j] < pNilaiKuadrat[i]) {
                    temp = pNilaiKuadrat[i];
                    pNilaiKuadrat[i] = pNilaiKuadrat[j];
                }
            }
        }
    }
}

```

```

        pNilaiKuadrat[j] = temp;
        for (int k = 0; k < 8; k++) {
            tempTrain[i][k] = pNilaiTraining[i][k];
            pNilaiTraining[i][k] = pNilaiTraining[j][k];
            pNilaiTraining[j][k] = tempTrain[i][k];
        }
    }
}
return pNilaiTraining;
}

double[] nilaiKuadratEuclidean(double[][] pTesting, double[][] pTraining) {
    double[] nilaiBalik = new double[pTraining.length];

    for (int i = 0; i < pTraining.length; i++) {
        double ips1 = Math.abs(pTraining[i][0] - pTesting[0][0]),
            ips2 = Math.abs(pTraining[i][1] - pTesting[0][1]),
            ips3 = Math.abs(pTraining[i][2] - pTesting[0][2]),
            ips4 = Math.abs(pTraining[i][3] - pTesting[0][3]),
            ips5 = Math.abs(pTraining[i][4] - pTesting[0][4]),
            ips6 = Math.abs(pTraining[i][5] - pTesting[0][5]),
            sks = Math.abs(pTraining[i][6] - pTesting[0][6]),
            pangkat = Math.pow(ips1, 2) + Math.pow(ips2, 2)
                + Math.pow(ips3, 2) + Math.pow(ips4, 2) + Math.pow(ips5, 2) +
Math.pow(ips6, 2)
                + Math.pow(sks, 2),
            akar = Double.parseDouble(df.format(Math.sqrt(pangkat)));
        nilaiBalik[i] = akar;
    }

    return nilaiBalik;
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JComboBox comboboxK;
private javax.swing.JComboBox comboboxKriteria;
private javax.swing.JComboBox comboboxStatus;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;

```

```

private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JLabel labelKeputusan;
private javax.swing.JTable tabelTesting;
private javax.swing.JTable tabelTraining;
private javax.swing.JTextField textIPS1;
private javax.swing.JTextField textIPS2;
private javax.swing.JTextField textIPS3;
private javax.swing.JTextField textIPS4;
private javax.swing.JTextField textIPS5;
private javax.swing.JTextField textIPS6;
private javax.swing.JTextField textIsiFileTesting;
private javax.swing.JTextField textIsiFileTraining;
private javax.swing.JTextField textNIM;
private javax.swing.JTextField textSKS;
private javax.swing.JButton tombolBersih;
private javax.swing.JButton tombolCariFileTesting;
private javax.swing.JButton tombolCariFileTraining;
private javax.swing.JButton tombolProses;
private javax.swing.JButton tombolTampilkanTesting;
private javax.swing.JButton tombolTampilkanTraining;
// End of variables declaration//GEN-END:variables
}

```

FormJamak.java

```

package form1;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.text.DecimalFormat;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.table.DefaultTableModel;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.xssf.usermodel.XSSFCell;
import org.apache.poi.xssf.usermodel.XSSFRow;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class FormJamak extends javax.swing.JFrame {

    private DecimalFormat df = new DecimalFormat("#.####");
    private int[][] nilaiKesimpulan = null;

    public FormJamak() {
        initComponents();
        this.setSize(650, 615);
        this.setLocationRelativeTo(null);
        comboJumlahK.addItem(5);
        comboJumlahK.addItem(10);
        comboJumlahK.addItem(15);
        comboJumlahK.addItem(20);
        comboJumlahK.addItem(25);
        comboJumlahK.addItem(30);
        comboJumlahK.setSelectedIndex(-1);
    }

    // <editor-fold    defaultstate="collapsed"    desc="Generated    Code"> //GEN-BEGIN: initComponents
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        textIsiFileTesting = new javax.swing.JTextField();
        tombolTampilkanTesting = new javax.swing.JButton();
        tombolCariFileTesting = new javax.swing.JButton();
        jPanel2 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();

```

```

textIsiFileTraining = new javax.swing.JTextField();
tombolTampilkanTraining = new javax.swing.JButton();
tombolCariFileTraining = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
tabelHasilPrediksi = new javax.swing.JTable();
jScrollPane3 = new javax.swing.JScrollPane();
tabelTesting = new javax.swing.JTable();
jScrollPane4 = new javax.swing.JScrollPane();
tabelTraining = new javax.swing.JTable();
jPanel3 = new javax.swing.JPanel();
jLabel3 = new javax.swing.JLabel();
comboJumlahK = new javax.swing.JComboBox();
tombolProses = new javax.swing.JButton();
tombolBersih = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Form Jamak");
setResizable(false);
getContentPane().setLayout(null);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Pilih File
Testing"));
jPanel1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

jLabel1.setText("Ambil File");

tombolTampilkanTesting.setText("Tampilkan");
tombolTampilkanTesting.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tombolTampilkanTestingActionPerformed(evt);
    }
});

tombolCariFileTesting.setText("Cari File");
tombolCariFileTesting.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tombolCariFileTestingActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(textIsiFileTesting,
javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addComponent(tombolCariFileTesting)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(tombolTampilkanTesting))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())
        .addContainerGap(138, Short.MAX_VALUE)
        .addComponent(jLabel1)
        .addGap(116, 116, 116))
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup())
    .addGap(6, 6, 6)
    .addComponent(jLabel1)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(textIsiFileTesting,
javax.swing.GroupLayout.PREFERRED_SIZE, 37, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(tombolCariFileTesting)
    .addComponent(tombolTampilkanTesting)))
    );

    getContentPane().add(jPanel1);
    jPanel1.setBounds(330, 0, 310, 130);

    jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder("Pilih File
Training"));
    jPanel2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

    jLabel2.setText("Ambil File");

    tombolTampilkanTraining.setText("Tampilkan");
    tombolTampilkanTraining.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            tombolTampilkanTrainingActionPerformed(evt);
        }
    });

    tombolCariFileTraining.setText("Cari File");
    tombolCariFileTraining.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {

```

[illegible]


```

        "No", "NIM", "Nama", "JK", "IPS 1", "IPS 2", "IPS 3", "IPS 4", "IPS 5", "IPS
6", "SKS", "KET", "Prediksi"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
jScrollPane1.setViewportViewView(tabelHasilPrediksi);

getContentPane().add(jScrollPane1);
jScrollPane1.setBounds(10, 410, 630, 100);

tabelTesting.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null}
    },
    new String [] {
        "No", "NIM", "Nama", "JK", "IPS 1", "IPS 2", "IPS 3", "IPS 4", "IPS 5", "IPS
6", "SKS", "KET"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false, false, false, true, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
jScrollPane3.setViewportViewView(tabelTesting);

getContentPane().add(jScrollPane3);
jScrollPane3.setBounds(10, 280, 630, 100);

tabelTraining.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null}
    },

```

```

        new String [] {
            "No", "NIM", "Nama", "JK", "IPS 1", "IPS 2", "IPS 3", "IPS 4", "IPS 5", "IPS
6", "SKS", "KET"
        }
    ) {
        boolean[] canEdit = new boolean [] {
            false, false, false, false, false, false, false, false, false, false,
        };

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });
jScrollPane4.setViewportView(tabelTraining);

getContentPane().add(jScrollPane4);
jScrollPane4.setBounds(10, 150, 630, 100);

jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder("Kriteria"));

jLabel3.setText("Jumlah K");

tombolProses.setText("Proses");
tombolProses.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tombolProsesActionPerformed(evt);
    }
});

tombolBersih.setText("Bersih");
tombolBersih.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tombolBersihActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addComponent(jLabel3)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
66, Short.MAX_VALUE)
        .addComponent(comboJumlahK,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(tombolProses)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(tombolBersih))
    );
    jPanel3Layout.setVerticalGroup(
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
)
        .addComponent(comboJumlahK,
javax.swing.GroupLayout.PREFERRED_SIZE, 32, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3))

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
)
        .addComponent(tombolProses)
        .addComponent(tombolBersih))
    );

    getContentPane().add(jPanel3);
    jPanel3.setBounds(250, 520, 390, 55);

    jLabel4.setText("Tabel Data Training");
    getContentPane().add(jLabel4);
    jLabel4.setBounds(10, 130, 140, 14);

    jLabel5.setText("Tabel Data Testing");
    getContentPane().add(jLabel5);
    jLabel5.setBounds(10, 260, 160, 14);

    jLabel6.setText("Tabel Data Prediksi");
    getContentPane().add(jLabel6);
    jLabel6.setBounds(10, 390, 170, 14);

    pack();
    setLocationRelativeTo(null);
} // </editor-fold> // GEN-END: initComponents

private void tombolCariFileTrainingActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_tombolCariFileTrainingActionPerformed
    try {
        JFileChooser JFC = new JFileChooser();
        FileNameExtensionFilter fFilter = new FileNameExtensionFilter("Excel", "xls",
"xlsx");
        JFC.setFileFilter(fFilter);

        if (JFC.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
            textIsiFileTraining.setText(JFC.getSelectedFile().getAbsolutePath());

```

```

    }
    } catch (Exception er1) {
        JOptionPane.showMessageDialog(this, "Kesalahan Upload Excel", "Informasi
Kesalahan Upload", JOptionPane.WARNING_MESSAGE);
    }
} //GEN-LAST:event_tombolCariFileTrainingActionPerformed

private void tombolTampilkanTrainingActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_tombolTampilkanTrainingActionPerformed
    try {
        if (textIsiFileTraining.getText().equals("")) {
            JOptionPane.showMessageDialog(this, "Belum Memilih File Excel", "Informasi
Kesalahan Upload File", JOptionPane.ERROR_MESSAGE);
        } else {
            if (textIsiFileTraining.getText().endsWith(".xls")) {
                ekstensiXLSTraining();
            } else if (textIsiFileTraining.getText().endsWith(".xlsx")) {
                ekstensiXLSXTraining();
            } else {
                JOptionPane.showMessageDialog(this, "Ekstensi Tidak Diketahui",
"Informasi Kesalahan Upload File", JOptionPane.ERROR_MESSAGE);
            }
        }
    } catch (Exception er1) {
    }
} //GEN-LAST:event_tombolTampilkanTrainingActionPerformed

private void tombolCariFileTestingActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_tombolCariFileTestingActionPerformed
    try {
        JFileChooser JFC = new JFileChooser();
        FileNameExtensionFilter fFilter = new FileNameExtensionFilter("Excel", "xls",
"xlsx");
        JFC.setFileFilter(fFilter);

        if (JFC.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
            textIsiFileTesting.setText(JFC.getSelectedFile().getAbsolutePath());
        }
    } catch (Exception er1) {
        JOptionPane.showMessageDialog(this, "Kesalahan Upload Excel", "Informasi
Kesalahan Upload", JOptionPane.WARNING_MESSAGE);
    }
} //GEN-LAST:event_tombolCariFileTestingActionPerformed

private void tombolTampilkanTestingActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_tombolTampilkanTestingActionPerformed
    try {
        if (textIsiFileTesting.getText().equals("")) {

```

```

        JOptionPane.showMessageDialog(this, "Belum Memilih File Excel", "Informasi
Kesalahan Upload File", JOptionPane.ERROR_MESSAGE);
    } else {
        if (textIsiFileTesting.getText().endsWith(".xls")) {
            ekstensiXLSTesting();
        } else if (textIsiFileTesting.getText().endsWith(".xlsx")) {
            ekstensiXLSXTesting();
        } else {
            JOptionPane.showMessageDialog(this, "Ekstensi Tidak Diketahui",
"Informasi Kesalahan Upload File", JOptionPane.ERROR_MESSAGE);
        }
    }
} catch (Exception er1) {

}

} //GEN-LAST:event_tombolTampilkanTestingActionPerformed

private void tombolProsesActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_tombolProsesActionPerformed
    if (comboJumlahK.getSelectedIndex() == -1) {
        JOptionPane.showMessageDialog(this, "Belum Memilih Jumlah K", "Informasi
Kesalahan Proses Prediksi", JOptionPane.WARNING_MESSAGE);
        comboJumlahK.requestFocus(true);
    } else {
        long w1 = System.currentTimeMillis();

        int jTrainingBaris = tabelTraining.getRowCount(),
            jTestingBaris = tabelTesting.getRowCount(),
            kolom = 0;

        double[][] nilaiTraining = nilaiDefaultTraining(jTrainingBaris),
            nilaiTesting = nilaiDefaultTesting(jTestingBaris),
            nilaiKuadrat = nilaiPerhitunganKuadrat(jTrainingBaris, jTestingBaris,
nilaiTraining, nilaiTesting),
            hasilTempS;

        double[][][] sortingNilaiTrainingS = new double[jTestingBaris][][];

        for (int i = 0; i < jTestingBaris; i++) {
            kolom = i;
            hasilTempS = sortingNilaiTraining(nilaiTraining, nilaiKuadrat, kolom)[0];
            sortingNilaiTrainingS[i] = hasilTempS;
        }

        int jumlahK = Integer.parseInt(comboJumlahK.getSelectedItem().toString());
        double[][][] nilaiTrainingK = nilaiTrainingSebanyakK(sortingNilaiTrainingS,
jumlahK);

        int[][][] jumlahSetiapKelas = jumlahTerbanyakKemunculan(nilaiTrainingK);

```

```

        double[][] nilaiTestingKesimpulan = kirimKesimpulanKeTesting(jTestingBaris,
        nilaiTesting);

        String[][] nilaiTestingKumpul = new String[jTestingBaris][13];
        //copy dari nim hingga keterangan
        for (int i = 0; i < jTestingBaris; i++) {
            for (int j = 0; j <= 11; j++) {
                nilaiTestingKumpul[i][j] = tabelTesting.getValueAt(i, j).toString();
            }
        }

        for (int i = 0; i < jTestingBaris; i++) {
            nilaiTestingKumpul[i][12] = String.valueOf(nilaiTestingKesimpulan[i][7]);
        }

        tabelHasilPrediksi.setModel(new DefaultTableModel(
            nilaiTestingKumpul,
            new String[]{
                "No.", "NIM", "Nama", "JK", "IPS1", "IPS2", "IPS3", "IPS4", "IPS5",
                "IPS6", "SKS", "KET", "PREDIKSI"
            }
        ));

        int jum = 1;
        for (int i = 0; i < jTestingBaris; i++) {
            double nStat = Double.parseDouble(tabelHasilPrediksi.getValueAt(i,
            11).toString()),
            nKet = Double.parseDouble(tabelHasilPrediksi.getValueAt(i,
            12).toString());
            if (nStat == nKet) {
                jum++;
            }
        }

        long w2 = System.currentTimeMillis(),
        w3 = (w2 - w1) / 1000;

        double rata1 = Double.parseDouble(df.format(jum /
        Double.parseDouble(String.valueOf(jTestingBaris)))) * 100;
        String info = "-Proses Testing Telah Selesai-\n";
        info += "Data Testing = " + jTestingBaris + "\n";
        info += "Tepat Prediksi = " + jum + "\n";
        info += "Akurasi Prediksi = " + df.format(rata1) + "\n";
        info += "Waktu = " + w3 + " Detik";
        JOptionPane.showMessageDialog(this, info, "Informasi Hasil Prediksi",
        JOptionPane.INFORMATION_MESSAGE);
    }
} //GEN-LAST:event_tombolProsesActionPerformed

```

```

        private void tombolBersihActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_tombolBersihActionPerformed
            tabelTraining.setModel(new DefaultTableModel());
            tabelTesting.setModel(new DefaultTableModel());
            tabelHasilPrediksi.setModel(new DefaultTableModel());
            comboJumlahK.setSelectedIndex(-1);
            textIsiFileTesting.setText("");
            textIsiFileTraining.setText("");
        }GEN-LAST:event_tombolBersihActionPerformed

        void ekstensiXLSTraining() {
            try {
                FileInputStream FIS = new FileInputStream(new
                File(textIsiFileTraining.getText()));
                HSSFWorkbook wb = new HSSFWorkbook(FIS);
                HSSFSheet ws = wb.getSheetAt(0);

                int baris = ws.getLastRowNum() + 1;
                int kolom = ws.getRow(0).getLastCellNum();
                String[][] data = new String[baris][kolom];
                String[] nKolom = new String[kolom];

                //ambil semua informasi yang berada pada sheet baik itu nama kolom maupun
                nilai pada baris nya
                for (int i = 0; i < baris; i++) {
                    HSSFRow row = ws.getRow(i);
                    for (int j = 0; j < kolom; j++) {
                        HSSFCell cell = row.getCell(j);
                        String value = nilaiCell(cell);
                        data[i][j] = value;
                    }
                }

                //ambil nama kolom pada index data baris ke 0
                for (int a = 0; a < kolom; a++) {
                    nKolom[a] = data[0][a];
                }

                DefaultTableModel DTM = new DefaultTableModel(data, nKolom);
                tabelTraining.setModel(DTM);
                DTM.removeRow(0);
            } catch (IOException er1) {
                System.out.print("Error : \n" + er1.toString());
            }
        }

        void ekstensiXLSTraining() {
            try {

```

```

        FileInputStream FIS = new FileInputStream(new
File(textIsiFileTraining.getText()));
        XSSFWorkbook wb = new XSSFWorkbook(FIS);
        XSSFSheet ws = wb.getSheetAt(0);

        int baris = ws.getLastRowNum() + 1;
        int kolom = ws.getRow(0).getLastCellNum();
        String[][] data = new String[baris][kolom];
        String[] nKolom = new String[kolom];

        for (int i = 0; i < baris; i++) {
            XSSFRow row = ws.getRow(i);
            for (int j = 0; j < kolom; j++) {
                XSSFCell cell = row.getCell(j);
                String value = nilaiCell(cell);
                data[i][j] = value;
            }
        }

        //ambil nama kolom pada index data baris ke 0
        for (int a = 0; a < kolom; a++) {
            nKolom[a] = data[0][a];
        }

        DefaultTableModel DTM = new DefaultTableModel(data, nKolom);
        tabelTraining.setModel(DTM);
        DTM.removeRow(0);
    } catch (Exception er1) {
        System.out.print("Error : \n" + er1.toString());
    }
}

void ekstensiXLSTesting() {
    try {
        FileInputStream FIS = new FileInputStream(new
File(textIsiFileTesting.getText()));
        HSSFWorkbook wb = new HSSFWorkbook(FIS);
        HSSFSheet ws = wb.getSheetAt(0);

        int baris = ws.getLastRowNum() + 1;
        int kolom = ws.getRow(0).getLastCellNum();
        String[][] data = new String[baris][kolom];
        String[] nKolom = new String[kolom];

        //ambil semua informasi yang berada pada sheet baik itu nama kolom maupun
        nilai pada baris nya
        for (int i = 0; i < baris; i++) {
            HSSFRow row = ws.getRow(i);
            for (int j = 0; j < kolom; j++) {

```



```

        XSSFCell cell = row.getCell(j);
        String value = nilaiCell(cell);
        data[i][j] = value;
    }
}

//ambil nama kolom pada index data baris ke 0
for (int a = 0; a < kolom; a++) {
    nKolom[a] = data[0][a];
}

DefaultTableModel DTM = new DefaultTableModel(data, nKolom);
tabelTesting.setModel(DTM);
DTM.removeRow(0);
} catch (IOException er1) {
    System.out.print("Error : \n" + er1.toString());
}
}

void ekstensiXLSXTesting() {
    try {
        FileInputStream FIS = new FileInputStream(new
File(textIsiFileTesting.getText()));
        XSSFWorkbook wb = new XSSFWorkbook(FIS);
        XSSFSheet ws = wb.getSheetAt(0);

        int baris = ws.getLastRowNum() + 1;
        int kolom = ws.getRow(0).getLastCellNum();
        String[][] data = new String[baris][kolom];
        String[] nKolom = new String[kolom];

        for (int i = 0; i < baris; i++) {
            XSSFRow row = ws.getRow(i);
            for (int j = 0; j < kolom; j++) {
                XSSFCell cell = row.getCell(j);
                String value = nilaiCell(cell);
                data[i][j] = value;
            }
        }

        //ambil nama kolom pada index data baris ke 0
        for (int a = 0; a < kolom; a++) {
            nKolom[a] = data[0][a];
        }

        DefaultTableModel DTM = new DefaultTableModel(data, nKolom);
        tabelTesting.setModel(DTM);
        DTM.removeRow(0);
    } catch (Exception er1) {

```

```

        System.out.print("Error : \n" + er1.toString());
    }
}

void cetak(String pInfo, double[][] pAR1) {
    System.out.println(pInfo);
    for (int i = 0; i < pAR1.length; i++) {
        for (int j = 0; j < pAR1[i].length; j++) {
            System.out.print(pAR1[i][j] + "\t");
        }
        System.out.println("");
    }
    System.out.println("");
}

void cetak(String pInfo, String[][] pAR1) {
    System.out.println(pInfo);
    for (int i = 0; i < pAR1.length; i++) {
        for (int j = 0; j < pAR1[i].length; j++) {
            System.out.print(pAR1[i][j] + "\t");
        }
        System.out.println("");
    }
    System.out.println("");
}

void cetak(String pInfo, double[][][] pAR1) {
    System.out.println(pInfo);
    for (int i = 0; i < pAR1.length; i++) {
        for (int j = 0; j < pAR1[i].length; j++) {
            for (int k = 0; k < pAR1[i][j].length; k++) {
                System.out.print(pAR1[i][j][k] + "\t");
            }
            System.out.println("");
        }
        System.out.println("");
    }
    System.out.println("");
}

String nilaiCell(HSSFCell pCell) {
    int tipe = pCell.getCellType();
    Object nilaiBalik = null;

    if (tipe == 0) {
        nilaiBalik = pCell.getNumericCellValue();
    } else if (tipe == 1) {
        nilaiBalik = pCell.getStringCellValue();
    }
}

```

```

        return nilaiBalik.toString();
    }

    String nilaiCell(XSSFCell pCell) {
        int tipe = pCell.getCellType();
        Object nilaiBalik = null;

        if (tipe == 0) {
            nilaiBalik = pCell.getNumericCellValue();
        } else if (tipe == 1) {
            nilaiBalik = pCell.getStringCellValue();
        }

        return nilaiBalik.toString();
    }

    double[][] nilaiDefaultTraining(int pBaris) {
        double[][] nilaiBalik = new double[pBaris][8];
        int kolom = 0;

        for (int i = 0; i < pBaris; i++) {
            for (int j = 4; j <= 11; j++) {
                nilaiBalik[i][kolom] = Double.parseDouble(tabelTraining.getValueAt(i,
j).toString());
                kolom++;
            }
            kolom = 0;
        }

        return nilaiBalik;
    }

    double[][] nilaiDefaultTesting(int pBaris) {
        double[][] nilaiBalik = new double[pBaris][8];
        int kolom = 0;

        for (int i = 0; i < pBaris; i++) {
            for (int j = 4; j <= 11; j++) {
                nilaiBalik[i][kolom] = Double.parseDouble(tabelTesting.getValueAt(i,
j).toString());
                kolom++;
            }
            kolom = 0;
        }

        return nilaiBalik;
    }

```

```

double[][] nilaiPerhitunganKuadrat(int pJTraining, int pJTesting, double[][] pTraining,
double[][] pTesting) {
    double[][] nilaiBalik = new double[pJTraining][pJTesting];

    for (int i = 0; i < pJTraining; i++) {
        for (int j = 0; j < pJTesting; j++) {
            double ips1 = Math.abs(pTraining[i][0] - pTesting[j][0]),
            ips2 = Math.abs(pTraining[i][1] - pTesting[j][1]),
            ips3 = Math.abs(pTraining[i][2] - pTesting[j][2]),
            ips4 = Math.abs(pTraining[i][3] - pTesting[j][3]),
            ips5 = Math.abs(pTraining[i][4] - pTesting[j][4]),
            ips6 = Math.abs(pTraining[i][5] - pTesting[j][5]),
            totalSKS = Math.abs(pTraining[i][6] - pTesting[j][6]),
            pangkat = Math.pow(ips1, 2) + Math.pow(ips2, 2)
            + Math.pow(ips3, 2) + Math.pow(ips4, 2)
            + Math.pow(ips5, 2) + Math.pow(ips6, 2)
            + Math.pow(totalSKS, 2),
            akar = Double.parseDouble(df.format(Math.sqrt(pangkat)));
            nilaiBalik[i][j] = akar;
        }
    }
    return nilaiBalik;
}

double[][][] sortingNilaiTraining(double nilaiTrain[][[]], double hasilKuadrat[][[]], int
indexTesting) {
    double[][] input = new double[nilaiTrain.length][[]], input2 = new
double[hasilKuadrat.length][[]];
    System.arraycopy(nilaiTrain, 0, input, 0, nilaiTrain.length);
    System.arraycopy(hasilKuadrat, 0, input2, 0, hasilKuadrat.length);
    double[][][] returnVal = new double[2][input.length][input[0].length];
    int counter, index;
    double temp, temp2[];

    for (counter = 0; counter < input.length - 1; counter++) {
        for (index = 0; index < input.length - 1 - counter; index++) {
            if (input2[index][indexTesting] > input2[index + 1][indexTesting]) {
                temp = input2[index][indexTesting];
                temp2 = input[index];

                input2[index][indexTesting] = input2[index + 1][indexTesting];
                input[index] = input[index + 1];

                input2[index + 1][indexTesting] = temp;
                input[index + 1] = temp2;
            }
        }
    }
}

```

```

    for (int i = 0; i < returnVal.length; i++) {
        for (int j = 0; j < input.length; j++) {
            for (int k = 0; k < input[0].length; k++) {
                if (i == 0) {
                    returnVal[i][j][k] = input[j][k];
                } else {
                    returnVal[i][j][k] = input2[j][k];
                }
            }
        }
    }
}

return returnVal;
}

double[][][] nilaiTrainingSebanyakK(double[][][] pNilaiTraining, int pJumlahK) {
    double[][][] nilaiBalik = new double[pNilaiTraining.length][pJumlahK][8];

    for (int i = 0; i < pNilaiTraining.length; i++) {
        for (int j = 0; j < pJumlahK; j++) {
            for (int k = 0; k < 8; k++) {
                nilaiBalik[i][j][k] = pNilaiTraining[i][j][k];
            }
        }
    }

    return nilaiBalik;
}

int[][][] jumlahTerbanyakKemunculan(double[][][] pNilaiK) {
    int[][][] nilaiBalik = new int[pNilaiK.length][1][1];
    int j30 = 0, j20 = 0, j40 = 0, j50 = 0, j60 = 0, tMaks = 0, tNilai = 0;
    this.nilaiKesimpulan = new int[pNilaiK.length][1];
    int[][] tempMaks = new int[pNilaiK.length][6];

    for (int a = 0; a < pNilaiK.length; a++) {
        for (int b = 0; b < pNilaiK[a].length; b++) {
            for (int c = 0; c < pNilaiK[a][b].length; c++) {
                if (pNilaiK[a][b][c] == 20) {
                    j20++;
                } else if (pNilaiK[a][b][c] == 30) {
                    j30++;
                } else if (pNilaiK[a][b][c] == 40) {
                    j40++;
                } else if (pNilaiK[a][b][c] == 50) {
                    j50++;
                } else if (pNilaiK[a][b][c] == 60) {
                    j60++;
                }
            }
        }
    }
}

```

```

    }
}
tempMaks[a][0] = j20;
tempMaks[a][1] = j30;
tempMaks[a][2] = j40;
tempMaks[a][3] = j50;
tempMaks[a][4] = j60;
j20 = 0;
j40 = 0;
j30 = 0;
j50 = 0;
j60 = 0;
}

int tunjuk = 0;
for (int a = 0; a < tempMaks.length; a++) {
    for (int b = 0; b < 4; b++) {
        if (tempMaks[a][b] > tMaks) {
            tMaks = tempMaks[a][b];
            tunjuk = b;
            if (tunjuk == 0) {
                tNilai = 20;
            } else if (tunjuk == 1) {
                tNilai = 30;
            } else if (tunjuk == 2) {
                tNilai = 40;
            } else if (tunjuk == 3) {
                tNilai = 50;
            } else if (tunjuk == 4) {
                tNilai = 60;
            }
        }
    }
    nilaiKesimpulan[a][0] = tNilai;
    tMaks = 0;
    tunjuk = 0;
}

return nilaiBalik;
}

double[][] kirimKesimpulanKeTesting(int pJumlahTesting, double[][] pNilaiTesting) {
    double[][] nilaiBalik = new double[pJumlahTesting][8];

    for (int i = 0; i < pJumlahTesting; i++) {
        for (int j = 0; j < 8; j++) {
            if (j == 7) {
                nilaiBalik[i][j] = (double) nilaiKesimpulan[i][0];
            } else {

```

```

        nilaiBalik[i][j] = pNilaiTesting[i][j];
    }
}

return nilaiBalik;
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JComboBox comboJumlahK;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JTable tabelHasilPrediksi;
private javax.swing.JTable tabelTesting;
private javax.swing.JTable tabelTraining;
private javax.swing.JTextField textIsiFileTesting;
private javax.swing.JTextField textIsiFileTraining;
private javax.swing.JButton tombolBersih;
private javax.swing.JButton tombolCariFileTesting;
private javax.swing.JButton tombolCariFileTraining;
private javax.swing.JButton tombolProses;
private javax.swing.JButton tombolTampilkanTesting;
private javax.swing.JButton tombolTampilkanTraining;
// End of variables declaration//GEN-END:variables
}

```