

**KLASIFIKASI PENERIMAAN BERAS MISKIN  
(RASKIN) MENGGUNAKAN METODE  
*K-NEAREST NEIGHBOR***

**(Studi Kasus : Kelurahan Tenda)**

**Oleh**

**MUKTI ALI MOHAMMAD**

**T3116126**

**SKRIPSI**



**PROGRAM SARJANA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS ICHSAN GORONTALO  
2023**

**PERSETUJUAN SKRIPSI**

**KLASIFIKASI PENERIMAAN BERAS MISKIN  
(RASKIN) MENGGUNAKAN METODE  
K-NEAREST NEIGHBOR**

(Studi kasus:kelurahan Tenda)

Oleh

**MUKTI ALI MOHAMMAD**

**T3116126**

**SKRIPSI**

Untuk memenuhi salah satu syarat ujian

Guna memperoleh gelar Sarjana Program

Studi Teknik Informatika

Telah disetujui dan siap untuk di seminarkan

Gorontalo, Maret 2022

**Pembimbing Utama**



**Asmaul Husna Nasrullah, M.Kom**

**NIDN. 0911108602**

**Pembimbing Pendamping**



**Rofiq Harun, M.Kom**

**NIDN. 0919048404**

**PENGESAHAN SKRIPSI**

**KLASIFIKASI PENERIMAAN BERAS MISKIN  
(RASKIN) MENGGUNAKAN METODE  
*K-NEAREST NEIGHBOR***

**(Studi Kasus : KELURAHAN TENDA)**

**Oleh**

**MUKTI ALI MOHAMMAD**

**T3116126**

Diperiksa oleh Panitia Ujian Strata Satu(S1)

Universitas Ichsan Gorontalo

1. Ketua Penguji

**Zohrahayaty, M.Kom**

2. Anggota

**Hastuti Dalai, M.Kom**

3. Anggota

**Roys Pakaya, M.Kom**

4. Anggota

**Asmaul Husnah Nasrullah, M.Kom**

5. Anggota

**Rofiq Harun, M.Kom**

.....  
.....  
.....  
.....  
.....  
.....

Mengetahui

Dekan Fakultas Ilmu Komputer

  
**Irvan Abraham Salihi, S.Kom, M.Kom**

**NIDN : 0928028101**

Ketua Program Studi

  
**Sudirman S. Panna, M.Kom**

**NIDN : 0924038205**

## PERNYATAAN SKRIPSI

**Dengan ini saya menyatakan bahwa :**

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk meendapatkan gelar akademik (Sarjana) baik di Universitas ichsan Gorontalo maupun diperguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari tim pembimbing.
3. Karyatulis (Skripsi) saya ini tidak lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar Pustaka.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbeneran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar, yang telah diperoleh karena karya tulis ini, serta sanksi lainnya dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, April 2023

Membuat Pernyataan  
  
Muku Ali Mohammad



## **ABSTRACT**

### **MUKTI ALI MOHAMMAD. T3116126. THE RECEIPTS CLASSIFICATION OF RICE FOR THE POOR USING THE K-NEAREST NEIGHBOR METHOD**

*Rice is the staple food of most of Indonesia's population. Rice for the poor is a staple food subsidy in the form of rice intended for poor families as an effort from the government to improve food security and provide protection to poor families. Therefore, in 2002 the Indonesian government launched the rice for the poor program as an implementation of the government's consistency. The rice for the poor program is stipulated in the Presidential Regulation of the Republic of Indonesia No. 15 of 2010 on the Acceleration of Poverty Reduction and Presidential Instruction No. 3 of 2010 on Equitable Development Programs. This program aims to reduce the expenditure burden of target households by meeting some of their basic needs in the form of rice. In addition, the program, rice for the poor, aims to increase and open access to family food through the sale of rice to beneficiary families with a predetermined amount. One of the efforts to overcome these problems is to implement one of the methods in data mining, namely classification, which can group data more accurately following the level of similarity of the data characteristics. In this research, data mining analysis is carried out with classification techniques using the K-Nearest Neighbor method. The variables applied in this study are government, education, income, housing conditions, employment, and government assistance card holders.*

*Keywords: classification, KNN method, rice for poor*



## ABSTRAK

### **MUKTI ALI MOHAMMAD. T3116126. KLASIFIKASI PENERIMAN BERAS MISKIN (RASKIN) MENGGUNAKAN METODE *K-NEAREST NEIGHBOR***

Beras merupakan makanan pokok sebagian besar penduduk Indonesia. Beras miskin (Raskin) merupakan subsidi pangan pokok dalam bentuk beras diperuntukkan bagi keluarga miskin sebagai upaya dari pemerintah untuk meningkatkan ketahanan pangan dan memberikan perlindungan pada keluarga miskin. Oleh karena itu, pada tahun 2002 pemerintahan Indonesia meluncurkan program beras miskin (Raskin) yang merupakan implementasi dari konsistensi pemerintah. Beras miskin (Raskin) telah dalam pelaturan presiden republik Indonesia nomor 15 tahun 2010 tentang percepatan penanggulangan kemiskinan dan instruksi presiden nomor 3 tahun 2010 tentang program pembangunan yang berkeadilan. Program ini bertujuan untuk mengurangi beban pengeluaran rumah tangga sasaran (RTS) melalui memenuhi sebagian kebutuhan pokok dalam bentuk beras. Selain itu, beras miskin bertujuan untuk meningkatkan dan membuka akses pangan keluarga melalui penjualan beras kepada keluarga penerima manfaat dengan jumlah telah ditentukan. Salah satu upaya untuk mengatasi permasalahan tersebut adalah dengan mengimplementasikan salah satu metode dalam data mining, yaitu klasifikasi, yang dapat mengelompokkan data secara lebih akurat sesuai dengan tingkat kemiripan dari karakteristik datanya. Dalam penelitian ini, analisa data mining dilakukan dengan Teknik klasifikasi menggunakan metode K-Nearest Neighbor. Variable yang akan diterapkan pada penelitian ini adalah pemerintah, pendidikan, penghasilan, kondisi rumah, pekerjaan, pemegang kartu bantuan pemerintah. Kata kunci: klasifikasi, metode KNN, beras miskin



## KATA PENGANTAR

Segala Puji Bagi ALLAH SWT Sebab dengan Taufiq Dan Hidayah-Nya lah sehingga penulis dapat menyelesaikan usulan penelitian proposal ini dengan judul **“Klasifikasi Penerimaan Beras Miskin (RASKIN ) Menggunakan Metode K-NEAREST NEIGHBOR** Untuk Memenuhi Salah Satu Syarat Penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo Sebagai Syarat Ujian Akhir Guna Memperoleh Gelar Sarjana Komputer.

Penulis menyadari sepenuhnya Bahwa Skripsi Ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terimah kasih dan penghargaan yang setinggi-tingginya kepada:

1. Ibu Dr. Hj. Juriko Abdussamad, M.Si, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo.
2. Bapak Dr. Abd Gaffar Latjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo.
3. Bapak Irvan Abraham Salihi, S.Kom, M.Kom, Selaku Dekan Fakultas Ilmu Komputer
4. Bapak Sudirman Melanggi, M.Kom, selaku Wakil Dekan I bidang Akademik
5. Ibu Irma Surya Kumala, M.Kom, selaku Wakil Dekan II bidang Administrasi umum dan keuangan.
6. Bapak Sudirman S. Panna, M.Kom, selaku ketua Program Studi Teknik Informatika
7. Ibu Asmaul Husna Nasrullah, M.Kom, selaku Pembimbing I yang telah banyak membimbing penulisan selama menyusun Skripsi ini
8. Bapak Rofig Harun, M.Kom, selaku Pembimbing II yang telah banyak membimbing penulisan selama menyusun Skripsi ini

9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis
10. Ucapan terima kasih kepada Kedua Orang Tua saya yang tercinta, atas segala kasih sayang, Jerih Payah, Motivasi dan doa yang di berikan pada penulis
11. Ucapan terima kasih pada Teman seperjuangan, khususnya Reguler C, Rivaldi Dondo, Moh. Yasir S.Kom, Agung Haipi, Didit Setyawan, Rifai Gobel S.Kom. yang telah banyak memberikan bantuan kepada penulis
12. Rekan-rekan seperjuangan yang telah memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
13. Kepada semua pihak yang ikut membantu dalam penyelesaian proposal ini yang tak sempat penulis sebutkan satu persatu;

Semoga Allah SWT, melimpahkan balasan atas jasa-jasa mereka kepada Kami.

Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat banyak kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang dicapai ini dapat bermanfaat untuk kita semua. Aamiin

Gorontalo, April 2023

**Penulis**



## DAFTAR ISI

	Halaman
<b>HALAMAN PERSETUJUAN .....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iii</b>
<b>PERNYATAAN.....</b>	<b>v</b>
<b>ABSTRAK .....</b>	<b>vi</b>
<b>KATA PENGANTAR .....</b>	<b>vii</b>
<b>DAFTAR ISI .....</b>	<b>viii</b>
<b>DAFTAR TABEL .....</b>	<b>x</b>
<b>DAFTAR GAMBAR .....</b>	<b>xi</b>
<b>BAB I    PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Identifikasi Masalah .....	4
1.3. Rumusan Masalah .....	4
1.4. Tujuan Penelitian .....	4
1.5. Manfaat Penelitian .....	4
<b>BAB II    LANDASAN TEORI .....</b>	<b>6</b>
2.1. Tinjauan Studi .....	6
2.2. Tinjauan Pustaka .....	7
2.2.1 Masyarakat Miskin .....	7
2.2.2 Beras Miskin .....	7
2.2.3 Klasifikasi .....	8
2.2.4 Metode <i>K-Nearest Neighbors</i> .....	9
2.2.5 Contoh Penggunaan <i>K-Nearest Neighbor</i> Untuk Klasifikasi .....	11
2.2.6 Siklus Hidup Pengembangan Sistem.....	14
2.2.7 Desain Sistem .....	17
2.2.1.1. Use Case Diagram .....	17
2.2.1.2. Class Diagram .....	19

2.2.1.3.Activity Diagram .....	20
2.2.1.4.Sequence Diagram .....	22
2.3. Teknik Pengujian Sistem .....	23
2.3.1.1.White Box Testing .....	24
2.3.1.2.Black Box Testing.....	28
2.4. Kerangka Pemikiran .....	31
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>32</b>
3.1. Jenis, Metode, Subjek, Waktu dan Lokasi Penelitian .....	32
3.2. Pengumpulan Data .....	32
3.3. Pemodelan .....	33
3.4. Pengembangan Model .....	33
3.5. Konstruksi Sistem .....	33
3.6 Analisa Sistem .....	33
3.7 Tahap Pengujian .....	34
<b>BAB IV HASIL PENELITIAN .....</b>	<b>35</b>
4.1 Hasil Pengumpulan Data .....	35
4.2 Hasil Pemodelan .....	36
4.3 Hasil Pengembangan Sistem .....	38
4.3.1 Desain Sistem Secara Umum .....	38
4.3.1.1 Diagram Konteks.....	38
4.3.1.2 Diagram Berjenjang.....	39
4.3.1.3 Diagram Arus Data.....	40
4.3.1.3.1 DAD Level 0 .....	40
4.3.1.3.2 DAD Level 1 Proses 1 .....	41
4.3.1.3.3 DAD Level 1 Proses 2 .....	42
4.3,1,4 Kamus Data .....	43
4.3.1.5 Desain Output Secara umum .....	44
4.3.1.6 Desain Input Secara umum.....	44
4.3.1.7 Desain Database Secara Umum.....	45
4.3.2 Desain Arsitektuk .....	45
4.3.3 Desain Interface.....	46

4.3.3.1 Mekanisme user .....	46
4.3.4 Desain Data.....	46
4.3.4.1 Struktur Data.....	46
4.3.4.2 Relasi .....	47
4.3.5 Flowgraph Untuk Pengujian white Box .....	48
4.3.6 Perhitungan CC Pada pengujian white Box .....	49
4.3.7 Path Pada Pengujian White Box .....	49
4.3.8 Pengujian Black Box .....	50
<b>BAB V HASIL PENELITIAN HASIL DAN PEMBAHASAN.....</b>	<b>52</b>
1.1 Hasil Penelitian.....	52
1.1.1 Hasil Pengujian Sistem.....	52
1.1.1.1 Pengujian White Box.....	52
1.1.1.2 Pengujian Black Box .....	55
5.2 Pembahasan .....	56
5.2.1 Deskripsi Kebutuhan Hardware/Software.....	56
5.2.2 Langkah-Langkah Menjalankan Sistem.....	57
5.2.2.1 Tampilan Halaman Login Admin .....	57
5.2.2.2 Tampilan Halaman Home Admin .....	58
5.2.2.3 Tampilan Halaman View Data Testing .....	59
5.2.2.4 Tampilan From Tambah Data Testing .....	60
5.2.2.5 Tampilan From Edit Data Testing.....	61
5.2.2.6 Tampilan Halaman Uji Data Testing .....	61
5.2.2.7 Tampilan Halaman View Data Training .....	62
5.2.2.8 Tampilan From Tambah Data Training .....	62
5.2.2.9 Tampilan From Edit Data Training.....	63
5.2.2.10 Tampilan Halaman Uji Data Training.....	64
5.2.2.11 Tampilan Halaman View Data User .....	64
5.2.2.12 Tampilan Halaman Laporan hasil Analisa .....	65
<b>BAB VI KESIMPULAN DAN SARAN .....</b>	<b>66</b>
1.1 Kesimpulan .....	66
1.2 Saran.....	66
<b>Daftar Pustaka.....</b>	<b>6</b>

## DAFTAR GAMBAR

Gambar 2.1 Framework of Classification .....	9
Gambar 2.2 Ilustrasi kedekatan kasus .....	10
Gambar 2.3 Use Case Diagram .....	18
Gambar 2.4. <i>Activity Diagram</i> .....	21
Gambar. 2.5. <i>Sequence Diagram</i> .....	23
Gambar 2.6. Bagan Air .....	26
Gambar 2.7 Flowgraph.....	27
Gambar 2.8 Kerangka Pemikiran .....	31
Gambar 3.1 Pemodelan .....	33
Gambar 4.1 Diagram Konteks.....	38
Gambar 4.2 Diagram Berjenjang .....	39
Gambar 4.3 DAD Level 0 .....	40
Gambar 4.4 DAD Level 1 Proses 1.....	41
Gambar 4.5 DAD Level 1 Proses 2.....	42
Gambar 4.6 Desain Relasi Antar Tabel.....	47
Gambar 4.7 Flowgraph untuk Pengujian White Box .....	48
Gambar 5.2 Flowgraph Proses Klasifikasi Penerima beras miskin .....	53
Gambar 5.3 Tampilan From Login Admin .....	57
Gambar 5.4 Tampilan Halaman Home Admin .....	58
Gambar 5.5 Tampilan Halaman View Data Testing.....	59
Gambar 5.6 Tampilan Halaman From Tambah Data Testing.....	60
Gambar 5.7 Tampilan From Edit Data Testing.....	61
Gambar 5.8 Tampilan Halaman Uji Data Testing .....	61
Gambar 5.9 Tampilan Halaman View Data Training .....	62
Gambar 5.10 Tampilan From Tamabah Data Training.....	62
Gambar 5.11 Tampilan From Edit Data Training.....	63
Gambar 5.12 Tampilan Halaman Uji Data Training.....	64
Gamabr 5.13 Tampilan Halaman View Data User .....	64
Gambar 5.13 Tampilan Halaman Laporan Hasil Analisa .....	65

## DAFTAR TABEL

Tabel 1.1. Data keluarga penerima beras miskin (Raskin) .....	2
Tabel 2.1. Penelitian Tentang kalsifikasi Menggunakan Metode <i>K-Nearest Neighbor</i> .....	6
Tabel 2.2. Hasil Pemasangan Kelas sesuai K .....	11
Tabel 2.3. Confusion Matrix K = 3 .....	12
Tabel 2.4. Hasil KNN dimana K=3 .....	12
Tabel 2.5. Confusion Matrix K = 4 .....	12
Tabel 2.6. Hasil KNN dimana K=4 .....	13
Tabel 2.7. Confusion Matrix K = 5 .....	13
Tabel 2.8. Hasil KNN dimana K=5.....	13
Tabel 2.9. Notasi Use Case Diagram .....	18
Tabel 2.10 Notasi Class Diagram.....	20
Tabel 2.11 Notasi Diagram Activity .....	22
Tabel 2.12. Notasi Diagram Sequence .....	23
Tabel 2.13. Hubungan antara Cyclomatic Complexity <i>dan Resiko</i> .....	28
Tabel 3.1 Atribut Data .....	32
Tabel 4.1 Hasil Pengumpulan Data.....	35
Tabel 4.2 Sampel Data Set .....	36
Tabel 4.3 Hasil Perhitungan Jarak (Distance).....	37
Tabel 4.4 Kamus Data Akurasi .....	43
Tabel 4.5 Kamus Data Atribut .....	43
Tabel 4.6 Daftar Output Yang Didesain .....	44
Tabel 4.7 Daftar Input Yang Didesain .....	44
Tabel 4.8 Daftar File Yang Didesain .....	45
Tabel 4.9 Interface Desain – Mekanisme User .....	46
Tabel 4.10 Data Desain : Struktur Data – User.....	46
Tabel 4.11 Data Desain : Stuktur Data – Data Atribut .....	47
Tabel 4.12 Path Pengujian White Box .....	49

Tabel 4.13 Hasil Pengujian Black Box Terhadap Beberapa proses.....	50
Tabel 5.1 Tabel Pengujian Black Box.....	55



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Beras merupakan makanan pokok sebagian besar penduduk Indonesia. Beras Miskin (Raskin) merupakan subsidi pangan pokok dalam bentuk beras yang diperuntukkan bagi keluarga miskin sebagai upaya dari pemerintah untuk meningkatkan ketahanan pangan dan memberikan perlindungan pada keluarga miskin. Oleh karena itu, pada tahun 2002 pemerintah Indonesia meluncurkan Program Beras Miskin (Raskin) yang merupakan implementasi dari konsistensi pemerintah dalam rangka memenuhi hak pangan masyarakat [1].

Permasalahan strategis di pemerintahan Provinsi Gorontalo Khususnya Kota Gorontalo yakni masih tingginya angka kemiskinan jika dibandingkan dengan provinsi lain. Data Badan Pusat Statistik Provinsi Gorontalo untuk jumlah penduduk miskin di Kota Gorontalo pada tahun 2020 berjumlah 12.460 jiwa, penduduk miskin ini tersebar di 9 Kecamatan dan 50 Kelurahan. Oleh karena itu salah satu program pemerintah Kota Gorontalo adalah pemeberian bantuan beras miskin (Raskin) 15 Kg kepada keluarga miskin yang ada di wilayah Kota Gorontalo.

Beras Miskin (Raskin) telah diatur dalam Peraturan Presiden Republik Indonesia Nomor 15 Tahun 2010 tentang Percepatan Penanggulangan Kemiskinan dan Instruksi Presiden Nomor 3 Tahun 2010 tentang Program Pembangunan yang Berkeadilan. Program ini bertujuan untuk mengurangi beban pengeluaran Rumah Tangga Sasaran (RTS) melalui pemenuhan sebagian kebutuhan pokok dalam bentuk beras. Selain itu, Beras Miskin (Raskin) bertujuan untuk meningkatkan dan membuka akses pangan keluarga melalui penjualan beras kepada keluarga penerima manfaat dengan jumlah yang telah ditentukan [2].

Kelurahan Tenda adalah salah satu kelurahan yang berada di Kecamatan Hulontalo yang menjalankan program bantuan beras miskin (Raskin) 15 Kg yang di programkan oleh pemerintah Kota Gorontalo, jumlah masyarakat penerima

bantuan beras miskin (Raskin) 15 Kg di Kelurahan Tenda untuk tahun 2020 berjumlah 118 kepala keluarga, data dari Kelurahan Tenda bahwa penerima beras miskin belum keseluruhan dari keluarga miskin di kelurahan Tenda yang berjumlah 251 Kepala keluarga. Berikut ini data keluarga miskin penerima beras miskin (Raskin) :

**Tabel 1.1** Data keluarga penerima beras miskin (Raskin)

Tahun	Jumlah Keluarga Miskin	Penerima Raskin
2017	279	137
2018	273	119
2019	265	170
2020	251	118

(Sumber: Kelurahan Tenda, 2021)

Tabel diatas menunjukan bahwa banyak keluarga miskin yang tidak termasuk penerima bantuan beras miskin dari pemerintah Kota Gorontalo, setiap tahunnya pemerintah Kelurahan Tenda menerima alokasi tambahan data penerima beras miskin. Permasalahan terjadi saat proses penetapan penerima bantuan raskin masih membutuhkan waktu yang lama karena masih berupa lembaran-lembaran berkas hasil survey lapangan dan data-data informasi dari pejabat-pejabat kelurahan yang tidak konsisten, dan ditambah dengan keterbatasan petugas yang berwenang dalam menganalisis variable-variabel yang menjadi prasyarat penerima beras miskin. Hal ini menjadi awal untuk merancang sebuah aplikasi terkomputerisasi yang dapat mengetahui masyarakat miskin yang layak dan tidak layak mendapatkan bantuan beras miskin dari pemerintah.

Salah satu upaya untuk mengatasi permasalahan tersebut adalah dengan mengimplementasikan salah satu metode dalam fungsi *data mining*, yaitu klasifikasi, yang dapat mengelompokkan data secara lebih akurat sesuai dengan tingkat kemiripan dari karakteristik datanya. Pada penelitian ini analisa data mining dilakukan dengan Teknik klasifikasi menggunakan metode *K-Nearest Neighbor*. Dengan menggunakan Teknik ini, data-data yang telah didapatkan dapat dikelompokkan kedalam beberapa cluster berdasarkan kemiripan dari data-data tersebut, sehingga data-data yang memiliki karakteristik yang sama dikelompokkan

dalam satu *cluster* dan yang memiliki karakteristik yang berbeda dikelompokkan dalam *cluster* yang lain yang memiliki karakteristik yang sama [3].

*K-Nearest Neighbor* adalah metode klasifikasi dengan mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (*neighbor*) terdekatnya dalam data pelatihan. Penentuan nilai terbaik dapat ditentukan dengan optimasi parameter, misalnya dengan menggunakan *K-Nearest Neighbor* yang merupakan salah satu metode yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut input yang acak [4].

Tujuan penelitian ini adalah untuk melakukan pengujian K-Nearest Neighbors dalam klasifikasi penerimaan bantuan beras miskin dengan menggunakan kriteria-kriteria yang sudah ditentukan untuk mengetahui penerima dana layak atau tidak layak untuk menerima bantuan beras miskin. Adapun variable yang akan diterapkan pada penelitian ini adalah pendidikan, penghasilan, kondisi rumah, pekerjaan, pemegang kartu bantuan pemerintah seperti kartu KIS (kartu indonesia sehat), BPNT (Bantuan Pangan Non Tunai) atau kartu PKH (Program kerluarga Harapan) dan sudah termasuk di daftar BDT (Basis Data Terpadu Kementerian Sosial RI).

Penelitian sebelumnya yang dilakukan oleh Hasran, 2020. Dengan judul Klasifikasi Penyakit Jantung Menggunakan Metode *K-Nearest Neighbor*. Penelitian ini mencakup pengukuran performa (akurasi, presisi, *recall* dan *f-measure*) metode KNN dengan nilai K 3 hingga 9 pada objek 1000 data pasien penyakit jantung yang diperoleh dari pusat dataset UCI Machine Learning Repository. Hasil dari pengukuran performa diperoleh nilai K terbaik adalah 6 dimana nilai akurasi 85%, presisi 78%, *recall* 93% dan *f-measure* sebesar 85% [5].

Berdasarkan hal tersebut di atas, maka penulis akan mengangkat Judul Penelitian tentang **“Klasifikasi Penerimaan Beras Miskin (Raskin) Menggunakan Metode *K-Nearest Neighbors*”** (Studi kasus Kelurahan Tenda).

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas maka permasalahan yang di ambil yaitu:

1. Penentuan bantuan beras miskin (Raskin) belum tepat sasaran.
2. Belum adanya sistem terkomputerisasi untuk menentukan masyarakat miskin yang menerima bantuan beras miskin (Raskin).

## 1.2 Rumusan Masalah

Dari uraian latar belakang masalah diatas, dapat dirumuskan masalah pokok yang berkaitan yaitu :

1. Bagaimana hasil klasifikasi penerimaan beras miskin (Raskin) dengan Menggunakan Metode *K-Nearest Neighbor*?
2. Bagaimana hasil penerapan metode *K-Nearest Neighbor* untuk klasifikasi penerimaan beras miskin (Raskin)?

## 1.3 Tujuan Penelitian

Adapun tujuan dari pembuatan software ini yaitu:

1. Untuk mengetahui hasil klasifikasi penerimaan beras miskin (Raskin) dengan Menggunakan Metode *K-Nearest Neighbor*.
2. Untuk mengetahui hasil penerapan metode *K-Nearest Neighbor* untuk klasifikasi penerimaan beras miskin (Raskin).

## 1.4 Manfaat Penelitian

Penelitian ini dimohonkan membawa faedah secara teoretis dan praktis kepada pihak terkait, sebagai pertimbangan, masukan dan pedoman serta evaluasi.

1. Secara Teoritis, Penelitian ini diharapkan dapat mengembangkan ilmu pengetahuan bagi akademis sebagai bahan masukan pemikiran mengenai permasalahan klasifikasi penerimaan beras miskin (Raskin) dengan efektif dan efisien, dan memberikan sumber informasi bagi mahasiswa apabila melakukan penelitian yang sejenis.
2. Secara Praktis, penelitian ini diharapkan dapat mengembangkan ilmu pengetahuan bagi perusahaan sebagai bahan informasi agar dapat

mengoptimalkan dan mengelola penerimaan beras miskin (Raskin) dengan terperinci secara terus menerus.

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Studi

Beberapa penelitian yang melekat tentang klasifikasi dan penerapan metode *K-Nearest Neighbor*, seperti di bawah ini:

**Tabel 2.1.** Penelitian Tentang klasifikasi Menggunakan Metode *K-Nearest Neighbor*

Peneliti	Judul	Hasil
Hasran, 2020. [5]	Klasifikasi Penyakit Jantung Menggunakan Metode <i>K-Nearest Neighbor</i> .	Penelitian ini mencakup pengukuran performa (akurasi, presisi, recall dan f-measure) metode KNN dengan nilai K 3 hingga 9 pada objek 1000 data pasien penyakit jantung yang diperoleh dari pusat dataset UCI Machine Learning Repository. Hasil dari pengukuran performa diperoleh nilai K terbaik adalah 6 dimana nilai akurasi 85%, presisi 78%, recall 93% dan f-measure sebesar 85%
Setiyo Arif Purnomo Aji, Hardian Oktavianto, Qurrota A'yun, 2020. [6]	Klasifikasi Penerima Bantuan Dana Desa Menggunakan Metode KNN ( <i>K-Nearest Neighbor</i> ) (Studi Kasus : Desa Andongsari Kecamatan Ambulu Kabupaten Jember)	Data yang digunakan merupakan data penerima bantuan langsung tunai dana desa tahun 2020 desa andongsari kecamatan ambulu kabupaten jember. Hasil yang di dapat dari penelitian ini berupa nilai akurasi tertinggi sebesar 90,79% dan K-optimal yaitu K-3 dengan akurasi yang tertinggi 90,79%. Maka dapat di simpulkan bahwa algoritma <i>K-Nearest Neighbor</i> dapat diterapkan dalam mengklasifikasi data penerima bantuan langsung tunai dana desa dengan Cukup akurat.
Esty Purwaningsih, Ela Nurelasari, 2021. [7]	Penerapan <i>K-Nearest Neighbor</i> Untuk Klasifikasi Tingkat Kelulusan Pada Siswa	<i>K-Nearest Neighbor</i> sering digunakan pada klasifikasi 'kinerja siswa karena kesederhanaannya juga dapat



		memberikan hasil yang signifikan dan kompetitif. Hasil dari prediksi tingkat kelulusan siswa dengan metode KNN didapat rata-rata akurasi dengan nilai sebesar 96,49%. Pengolahan data ‘dilakukan dengan ‘menggunakan tools rapid miner. Output dari implementasi pada ‘prediksi ‘tingkat ‘kelulusan dapat ‘dijadikan sebagai ‘acuan bagi siswa untuk ‘meningkatkan ‘prestasi dan predikat studi ‘lanjut dimasa yang akan datang.
--	--	--

## 2.2 Tinjauan Pustaka

### 2.2.1 Masyarakat Miskin

Permasalahan kemiskinan memang merupakan permasalahan yang kompleks dan bersifat perspektif. Oleh karena itu, upaya pengntasan kemiskinan harus dilakukan secara menyeluruh, yang mencakup berbagai aspek kehidupan di masyarakat, dan dilaksanakan secara terpadu. Deskripsi dari masyarakat miskin secara umumnya sebagai keadaan dari sekelompok orang yang tidak mampu memenuhi hak dasarnya demi mempertahankan kehidupan yang layak dan bermartabat. Deskripsi yang sangat luas ini mengisyaratkan bahwa kemiskinan merupakan masalah perspektif, sehingga tidak mudah untuk mengukur kemiskinan dan perlu kesepakatan pendekatan pengukuran yang dipakai [3].

### 2.2.2 Beras Miskin

Beras miskin (RASKIN) pada dasarnya adalah beras murah yaitu yang harga jualnya kepada masyarakat telah disubsidi oleh pemerintah, yang diberikan kepada keluarga prasejahtera dan sejahtera satu. Penetapan jumlah keluarga miskin yang berhak menerima RASKIN adalah sesuai dengan ketentuan pemerintah dalam hal ini Menko Kesra yaitu berdasarkan data dari BPS dan BKKBN. Kebijakan ini diambil oleh pemerintah agar dalam memberikan subsidi dan mengupayakan bantuan, dapat disalurkan tepat mencapai sasaran. Secara kriteria BKKBN telah memiliki standar keluarga yang masuk kategori miskin yaitu keluarga prasejahtera

dan sejahtera satu [2]. Keluarga prasejahtera adalah keluarga yang memiliki ciri-ciri sebagai berikut :

1. Anggota keluarga hanya makan dua kali sehari;
2. Anggota keluarga mempunyai pakaian yang berbeda terbatas untuk di rumah, bekerja/bersekolah dan bepergian;
3. Lantai rumah maksimal terbuat dari plester.

Sementara itu keluarga sejahtera satu memiliki ciri-ciri sebagai berikut:

1. Makan daging atau telur hanya seminggu sekali
2. Setiap anggota keluarga hanya mampu membeli pakaian baru setahun sekali;
3. Luas lantai rumah per penghuni hanya 8 meter persegi.

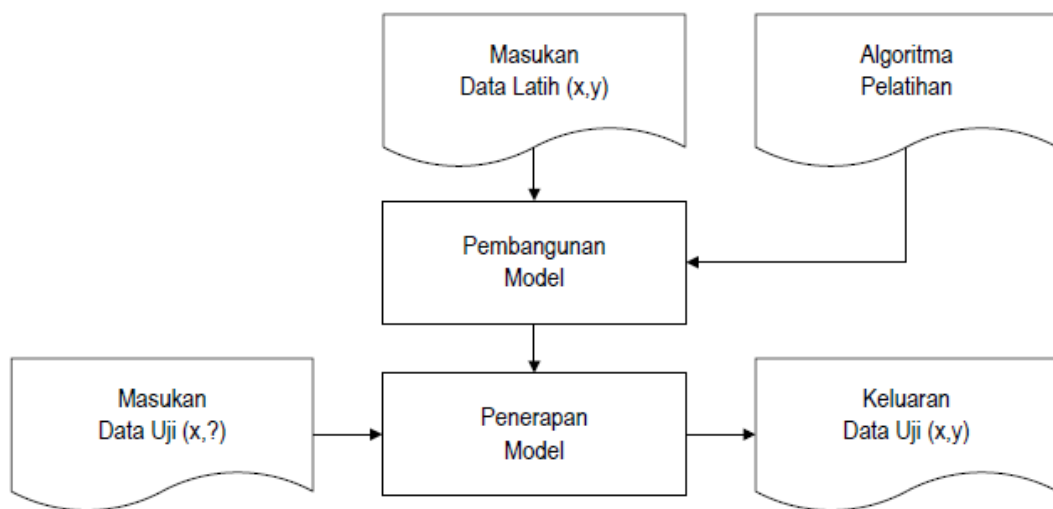
### 2.2.3 Klasifikasi

*Pattern recognition* melakukan pemetaan suatu data ke dalam konsep tertentu yang telah didefinisikan sebelumnya. Konsep tertentu yang dimaksud adalah *class* atau *category*. *Classification* (klasifikasi) merupakan konsep/teknologi yang cukup penting di dalam ilmu AI. Beberapa metode/algorithm yang umum dikenal dalam *classification* adalah: *Linear Discrimination Analysis* (LDA), *Hidden Markov Model*, *Artificial Neural Network* (ANN), *Fuzzy*, *Support Vector Machines* (SVM), *Bayesian*, *Boosting*, *Nearest Neighbor*, *Naïve Bayes*, dll [6].

Klasifikasi merupakan kata serapan dari bahasa Belanda, *classificatie*, yang sendirinya berasal dari bahasa Prancis *classification*. Istilah ini menunjuk kepada sebuah metode untuk menyusun data secara sistematis atau menurut beberapa aturan atau kaidah yang telah ditetapkan. Secara harafiah bisa pula dikatakan bahwa klasifikasi adalah pembagian sesuatu menurut kelas-kelas. Dengan demikian, klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia [6].

Dalam klasifikasi, ada 2 pekerjaan utama yang dilakukan, yaitu: (1) Pembangunan model sebagai prototipe untuk disimpan sebagai memori; (2) Penggunaan model tersebut untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam

model yang sudah disimpannya. Model dalam klasifikasi mempunyai arti yang sama dengan kotak hitam, di mana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut, dan memberikan jawaban sebagai keluaran dari hasil pemikirannya. *Framework* klasifikasi ditunjukkan pada gambar di bawah ini. Pada gambar tersebut disediakan sejumlah data latih  $(x,y)$  untuk digunakan sebagai data pembangunan model. Model tersebut kemudian digunakan untuk memprediksi kelas dari data uji  $(x,?)$  sehingga diketahui kelas  $y$  yang sesungguhnya [8].



**Gambar 2.1:** Framework of Classification

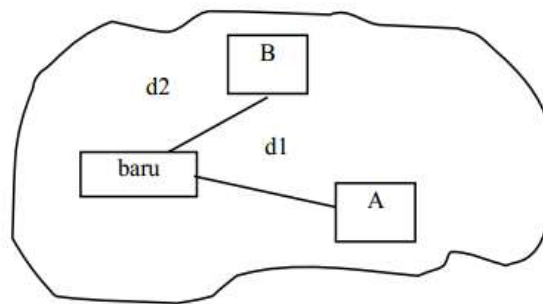
Kerangka kerja seperti yang ditunjukkan pada gambar di atas meliputi 2 langkah proses, yaitu induksi dan deduksi. Induksi merupakan langkah untuk membangun model klasifikasi dari data latih yang diberikan, disebut juga proses pelatihan. Sedangkan deduksi merupakan langkah untuk menerapkan model tersebut pada data uji sehingga kelas yang sesungguhnya dari data uji dapat diketahui, disebut juga proses prediksi/uji [8].

#### 2.2.4 Metode *K-Nearest Neighbors*

Algoritma *K-Nearest Neighbor* (KNN) adalah merupakan sebuah metode untuk melakukan klasifikasi terhadap obyek baru berdasarkan (K) tetangga terdekatnya. KNN termasuk algoritma *supervised learning*, dimana hasil dari *query instance* yang baru, diklasifikasikan berdasarkan mayoritas dari kategori pada

KNN. Kelas yang paling banyak muncul yang akan menjadi kelas hasil klasifikasi [8]

*Nearest Neighbor* adalah suatu pendekatan untuk menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada. Ilustrasi kedekatan kasus pada Gambar 2.3 memberikan gambaran tentang proses mencari solusi terhadap seorang pasien baru dengan menggunakan mengacu pada solusi dari pasien terdahulu. Untuk mencari kasus pasien mana yang akan digunakan, maka dihitung kedekatan anatara kasus pasien baru dengan semua kasus pasien lama. Kasus pasien lama dengan kedekatan terbesar-lah yang akan diambil solusinya untuk digunakan pada kasus pasien baru.



**Gambar 2.2** Ilustrasi kedekatan kasus

Rumus untuk melakukan penghitungan kedekatan antara kedua dua kasus adalah sebagai berikut:

$$\frac{\sum_{i=1}^n f(T_i, S_i) * W_i}{W_i} \dots\dots\dots \text{Persamaan 2.1}$$

Similarity (T, S):

Dimana:

T : kasus baru

S : kasus dalam penyimpanan

n : jumlah atribut tiap kasus

i : atribut individu 1 s/d n

$f$  : fungsi kesamaan atribut  $i$  antara kasus  $T$  dan  $S$

$w$  : bobot pada atribut yang ke  $i$

Langkah-langkah untuk menghitung metode *Algoritma K-Nearest Neighbor*:

1. Menentukan Parameter  $K$  (Jumlah tetangga paling dekat).
2. Menghitung kuadrat jarak *Euclid (query instance)* masing-masing objek terhadap data sampel yang diberikan.
3. Kemudian mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak *Euclid* terkecil.
4. Mengumpulkan kategori  $Y$  (*Klasifikasi Nearest Neighbor*)
5. Dengan menggunakan kategori *Nearest Neighbor* yang paling mayoritas maka dapat iprediksi nilai *query instance* yang telah dihitung.

### 2.2.5 Contoh Penggunaan *K-Nearest Neighbor* Untuk Klasifikasi

Penggunaan *K-Nearest Neighbor* Untuk Klasifikasi pada penelitian Andi Maulida Argina, 2020. Penerapan Metode Klasifikasi *K-Nearest Neighbor* pada Dataset Penderita Penyakit Diabetes [9]. penelitian ini adalah dengan melakukan pembagian data training dan data testing, data yang digunakan sebanyak 77 data, dengan pembagian sebesar 90% sebagai data training dan 10% sebagai data tesing. Tahapan selanjutnya adalah menerapkan metode KNN, pemilihan nilai  $K$  pada penelitian ini yaitu nilai  $K=3,4$  dan 5. Tabel 2.2 Menunjukkan hasil percobaan metode knn pada  $k=3,4$  dan 5

**Tabel 2.2:** Hasil Pemasangan Kelas sesuai  $K$

K=3		K=4		K=5	
1	TP	1	TP	1	TP
0	FN	0	FN	0	FN
1	TP	1	TP	1	TP
0	FN	0	FN	0	FN
0	FP	1	TP	0	FP
...	...	...	...	...	...
0	FN	0	FN	0	FN

0	FN	0	FN	0	FN
0	FN	0	FN	0	FN

Berdasarkan Tabel 2.2 Tahap selanjutnya adalah meneruskan hasil tersebut ke dalam bentuk confusion matrix, Tabel 2.3 menunjukkan confusion matrix pada  $k=3$ .

**Tabel 2.3** Confusion Matrix  $K = 3$

<b>n = 77</b>	Predicted : 1		Predicted :
Actual : 1	TP = 20		FN = 36
Actual : 0	FP = 11		TN = 10

Setelah diterapkan kedalam confusion matrix, performa metode dapat diukur, Tabel 2.4 Menunjukkan performa metode K-nn pada nilai  $K=3$ , dimana performa yang diukur adalah akurasi, presisi serta recall

**Tabel 2.4** Hasil KNN dimana  $K=3$

Akurasi	39%
Presisi	65%
Recall	36%
F-Measure	46%

Berdasarkan Tabel 2.2 Tahap selanjutnya adalah meneruskan hasil tersebut ke dalam bentuk confusion matrix, Tabel 2.5 menunjukkan confusion matrix pada  $k=4$ .

**Tabel 2.5** Confusion Matrix  $K = 4$

<b>n = 77</b>	Predicted : 1	Predicted : 0
Actual : 1	TP = 18	FN = 37
Actual : 0	FP = 13	TN = 9

Setelah diterapkan kedalam confusion matrix, performa metode dapat diukur, Tabel 2.4 Menunjukkan performa metode K-NN pada nilai  $K=4$ , dimana performa yang diukur adalah akurasi, presisi serta recall



**Tabel 2.6** Hasil KNN dimana K=4

Akurasi	35%
Presisi	58%
Recall	33%
F-Measure	42%

Berdasarkan Tabel 2.2 Tahap selanjutnya adalah meneruskan hasil tersebut ke dalam bentuk confusion matrix, Tabel 2.5 menunjukkan confusion matrix pada k=5.

**Tabel 2.7** Confusion Matrix K = 5

n = 77	Predicted : 1	Predicted : 0
Actual : 1	TP = 20	FN = 39
Actual : 0	FP = 11	TN = 7

Setelah diterapkan kedalam confusion matrix, performa metode dapat di ukur, Tabel 2.4. Menunjukkan performa metode K-nn pada nilai K=5, dimana performa yang diukur adalah akurasi, presisi serta recall

**Tabel 2.8** Hasil KNN dimana K=5

Akurasi	35%
Presisi	65%
Recall	34%
F-Measure	44%

Dari hasil perhitungan algoritma K-Nearest Neighbor (KNN) di atas, maka telah mendapatkan hasil akurasi tertinggi yaitu 39% pada K=3, presisi tertinggi yaitu 65% pada K=3 dan K=5, *recall* tertinggi yaitu 36% pada K=3, dan *F-Measure* tertinggi yaitu 46% pada K=3. Nilai yang diperoleh tidak cukup baik dikarenakan jumlah data yang digunakan cukup kecil. Saran untuk penelitian selanjutnya adalah melakukan percobaan yang sama dengan menambahkan jumlah data serta menerapkan crossvalidation

### 2.2.6 Siklus Hidup Pengembangan Sistem

Analisa sistem (*System Analisa*) dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya [10].

Analisa sistem adalah spesialis yang mempelajari masalah dan kebutuhan sebuah organisasi untuk menentukan bagaimana orang, data, proses dan teknologi informasi dapat mencapai kemajuan terbaik untuk bisnis.

Analisis sistem adalah *Stakeholder* yang berperan sebagai fasilitator atau pelatih, menjembatani jurang komunikasi yang dapat secara alamiah berkembang antara pemilik dan pengguna *system nonteknis* atau desainer dan perkembangan sistem teknis.

Whitten, et al. [11] mengungkapkan “*System analysis* adalah study domain masalah bisnis untuk merekomendasikan perbaikan dan menspesifikasi persyaratan dan prioritas bisnis untuk solusi”.

Impak teknologi objek sangat berarti dalam dunia analisis dan desain sistem. Sebelum ada teknologi objek, kebanyakan bahasa pemrograman didasarkan pada apa yang disebut metode yang terstruktur (*structured method*). Contohnya COBOL bahasa yang domain 0, C, Fortran, Pascal, dan PL/i. Maka, metode analisis dan desain berorientasi objek telah muncul sebagai pendekatan terpilih untuk membangun kebanyakan sistem informasi saat ini.

Sebagai tambahan keahlian analisis dan desain sistem formal, seorang analis harus mengembangkan atau memiliki keahlian lain, pengetahuan, dan karakter untuk menyelesaikan pekerjaan. Hal ini termasuk:

1. Pengalaman dan keahlian pemrograman komputer.

Sulit untuk membayangkan bagaimana para analisis sistem dapat dengan cukup mempersiapkan bisnis dan spesifikasi teknis untuk programmer jika mereka tidak memiliki pengalaman programan. Kebanyakan analis system harus menguasai satu atau lebih bahasa pemrograman tingkat tinggi.

2. Pengetahuan umum proses dan teknologi bisnis.

Analisis sistem harus mampu berkomunikasi dengan para ahli bisnis untuk memperoleh pemahaman masalah dan kebutuhan mereka. Untuk analisis, paling tidak sebagian dari pengetahuan ini datang hanya dari pengalaman. Pada saat yang sama analisis yang terinspirasi harus mengambil manfaat dari setiap kesempatan untuk menyelesaikan mata kuliah teori bisnis dasar.

Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan didalam tahap ini akan menyebabkan juga kesalahanditahap selanjutnya. Tahap analisa sistem mencakup studi kelayakan analisis kebutuhan.

a. Studi Kelayakan.

Studi kelayakan digunakan untuk menentukan kemungkinan keberhasilan solusi yang diusulkan. Tahapan berguna untuk memastikan bahwa solusi yang diusulkan tersebut benar-benar dapat dicapai dengan sumber daya dan dengan memperhatikan kendala yang terdapat pada perusahaan serta dampak terhadap lingkungan sekeliling. Tugas-tugas yang tercakup dalam studi kelayakan meliputi:

1. Penentuan masalah dan peluang yang dituju sistem.
2. Pembentukan sasaran sistem baru secara keseluruhan.
3. Pengidentifikasian para pemakai sistem.
4. Pembentukan lingkup sistem.

Selain itu, selama dalam tahapan studi kelayakan sistem analisis juga melakukan tugas-tugas sebagai berikut:

1. Pengusulan perangkat lunak dan perangkat keras untuk sistem baru.
2. Pembuatan analisis untuk membuat atau membeli aplikasi.
3. Pembuatan analisis biaya/manfaat.
4. Pengkajian terhadap resiko proyek.

Studi kelayakan diukur dengan memperhatikan aspek teknologi, ekonomi, faktor organisasi dan kendala hukum, etika, dan yang lain [10].

b. Analisis kebutuhan.

Analisis kebutuhan dilakukan untuk menghasilkan spesifikasi kebutuhan (disebut juga spesifikasi fungsional). Spesifikasi kebutuhan adalah spesifikasi

yang rinci tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan. Spesifikasi ini sekaligus dipakai untuk membuat kesepakatan antara pengembang sistem, pemakai yang kelak akan menggunakan sistem, manajemen, dan mitra kerja yang lain (misalnya auditor internal).

Analisis kebutuhan ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi keluaran, volume data yang akan ditangani sistem, jumlah pemakai dan kategori pemakai, serta kontrol terhadap sistem.

Didalam tahap analisis ini sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem, yaitu sebagai berikut [12] :

1. *Identify*, yaitu mengidentifikasi masalah.

Mengidentifikasi (mengenai) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah (*problems*) dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Tahap indentifikasi sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Tahap identifikasi masalah sangat penting karena akan menentukan keberhasilan pada langkah-langkah selanjutnya.

2. *Understand*, yaitu memahami kerja dari sistem yang ada.

Langkah kedua dari tahap analisis sistem adalah memahami kerja dari sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari operasi dari sistem ini diperlukan data yang dapat diperoleh dengan cara melakukan penelitian.

3. *Analyze*, yaitu menganalisis sistem tanpa report.

Langkah ini dilakukan berdasarkan data yang telah diperoleh dari hasil penelitian yang telah dilakukan.

4. *Report*, yaitu membuat laporan hasil analisis.

Tujuan utama dari pembuatan laporan hasil dilakukan;

- a. Pelaporan bahwa analisi telah selesai dilakukan.
- b. Meluruskan kesalah pengertian mengenai apa yang telah ditemukan dan dianalisis oleh analisis sistem tetapi tidak sesuai menurut manajemen

### 2.2.7 Desain Sistem

Setelah tahap analisis sistem selesai dilakukan, maka analisis sistem telah mendapat gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analisis sistem untuk memikirkan bagaimana membentuk sistem tersebut atau biasa disebut sebagai desain sistem (*system design*). Dalam desain sistem dibutuhkan alat bantu. Salah satu alat bantu yang dapat digunakan dalam pembuatan sistem adalah *Unified Modeling Language* (UML).

Menurut Whitten & Bentley (2007:371), *Unified Modeling Language* (UML) adalah sebuah bahasa pemodelan yang digunakan untuk menentukan atau mendeskripsikan sebuah sistem *software* berdasarkan objek-objek yang ada di sistem tersebut. UML tidak menentukan metode apa yang harus digunakan dalam mengembangkan suatu sistem, namun hanya menentukan notasi-notasi standar yang biasa digunakan untuk *object modeling* [13].

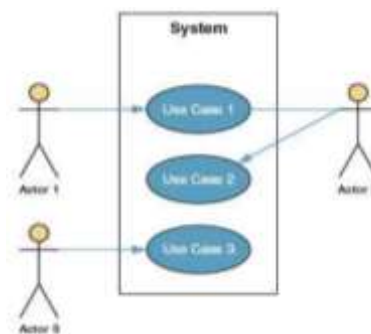
Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C [13].

#### 2.2.7.1 Use Case Diagram

Menurut Whitten & Bentley (2007:246) *Use-case diagram* adalah sebuah diagram yang mendeskripsikan interaksi antara sistem dengan bagian eksternal dari sistem serta dengan *user*. Secara grafis, *Use-case diagram* ini mendeskripsikan siapa yang akan menggunakan sistem yang ada dan bagaimana ekspektasi *user* saat berinteraksi dengan sistem tersebut [13].




Use case diagram memiliki unsur yang harus dipenuhi, yaitu:

- Use Cases*, yaitu sekumpulan fungsi yang terdapat dalam sistem dimana fungsi-fungsi tersebut dapat dilakukan oleh *actor* (*user*) untuk melakukan pekerjaannya dengan sistem yang ada.
- Actors*, yaitu segala sesuatu yang berinteraksi dengan sistem untuk bertukar informasi, baik *user* maupun sistem dari luar.
- Relationships*, yaitu garis yang menghubungkan antara *actors* dengan *use cases* yang dapat menggambarkan hubungan antara *actors* dengan *use cases* itu sendiri.



**Gambar 2.3.** Use Case Diagram: Whitten & Bentley (2007:246) [13].

**Tabel 2.9.** Notasi Use Case Diagram

Nama Komponen	Keterangan	Simbol
<i>Use Case</i>	<i>Use case</i> digambarkan sebagai lingkaran elips dengan nama use case dituliskan di dalam elips tersebut	
<i>Actor</i>	<i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan <i>input</i> atau memberikan <i>output</i> , maka aplikasi tersebut juga dianggap sebagai <i>actor</i>	
<i>Association</i>	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara Use case named <i>actor</i> dengan <i>use case</i>	

(Sumber: Whitten & Bentley, 2007:246) [13].



### 2.2.7.2 Class Diagram

Menurut Whitten & Bentley (2007:382), *class diagram* adalah sebuah diagram menggambarkan struktur objek dari sistem yang ada, dimana *class diagram* ini memperlihatkan *object class* yang menyusun diagram ini beserta hubungan antara *object class* tersebut [13].

Menurut Whitten & Bentley (2007:400-405), Terdapat beberapa tahap pembentukan *class diagram*, antara lain [13]:

1. Mengidentifikasi asosiasi dan keberagaman dari *class* yang ada dari objek.

Pada tahapan ini, kita akan mengidentifikasi asosiasi yang ada dari *class object* yang ada. Asosiasi yang dimaksud di sini adalah mengenai informasi apa yang perlu diketahui antara sebuah objek dengan objek lainnya.

2. Mengidentifikasi hubungan yang general dan hubungan khusus atas *class*.

Setelah kita mengetahui asosiasi dan keberagaman dari *class* yang ada, kita perlu mengetahui apakah hubungan antar *class* tersebut termasuk hubungan umum atau hubungan khusus. Hubungan umum atau khusus yang dimaksud di sini adalah klasifikasi dari sebuah hierarki, sebuah hubungan berdasarkan *supertype class* (*abstract / parent*) dan *subtype class* (*concrete / child*).

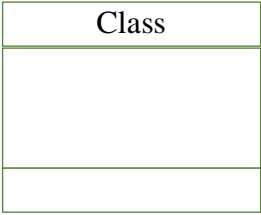




3. Mengidentifikasi hubungan agregasi / komposisi dari suatu *class*.

Pada tahap ini, kita harus menentukan apakah ada hubungan agregasi / komposisi yang terjadi antar *class* yang ada. Hubungan agregasi yang dimaksud adalah jenis hubungan yang unik dari suatu objek yang merupakan bagian dari objek tertentu.

4. Menyiapkan *class diagram* itu sendiri.

Pada tahap ini, kita menyusun *class diagram* berdasarkan informasi mengenai hubungan antar *class* yang ada, baik hubungan asosiasi, hubungan general / khusus, maupun hubungan agregasi yang terjadi antar *class* tersebut.

**Tabel 2.10.** Notasi Class Diagram

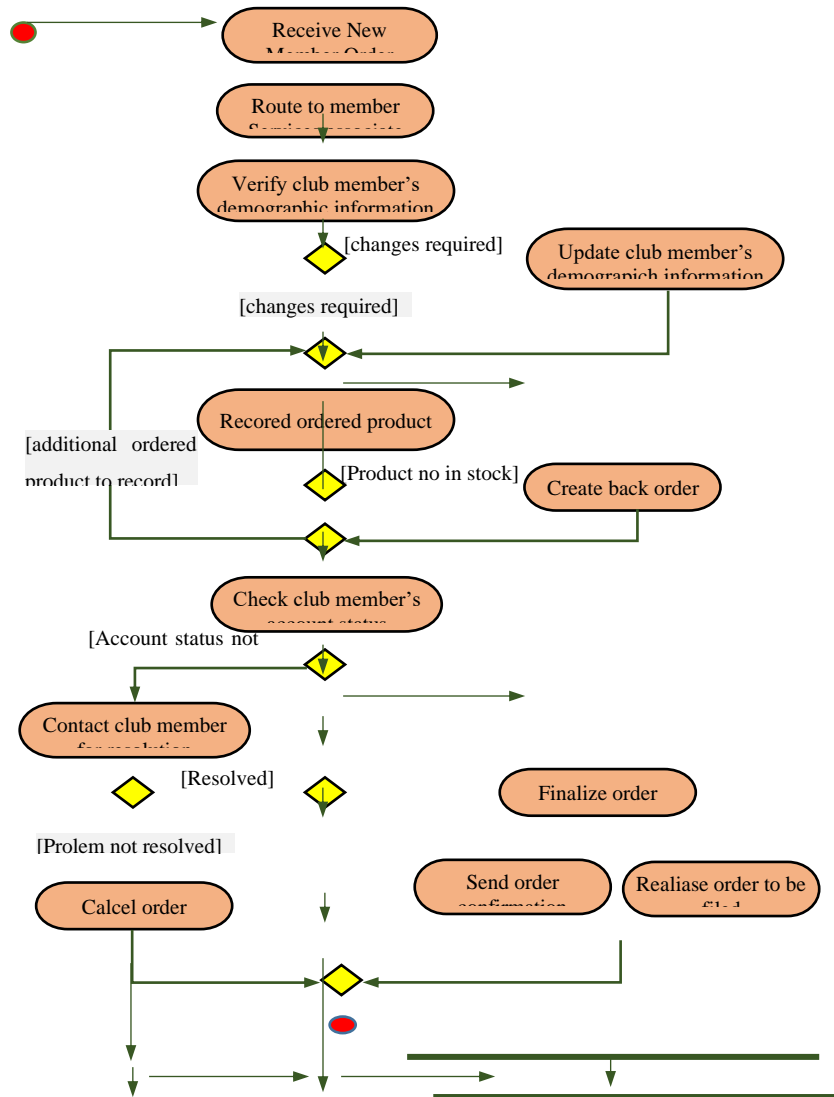
Symbol	Penjelasan
	<p><b>Class:</b></p> <p>Deskripsi dari objek terbagi atas 3 bagian, yaitu nama class pada bagian atas, atribut pada bagian tengah dan operasi pada bagian bawah.</p>
	<p><b>Aggregation:</b></p> <p>Bentuk spesial dari hubungan asosiasi yang memiliki hubungan secara spesifik antar kumpulan dan sebuah bagian. Agregasi digambarkan dengan wajik tidak berisi.</p>
	<p><b>Association:</b></p> <p>Menggambar hubungan terstruktur antar class yang saling berelasi.</p>
	<p><b>Generalization:</b></p> <p>Relasi yang memperhatikan suatu kelas dapat lebih general atau lebih spesifik dari kelas lainnya.</p>
<p><i>contains</i></p> 	<p><b>Multiplicity:</b></p> <p>Mengambarkan jumlah objek yang berpartisipasi dalam hubungan antar class.</p>

(Sumber: Whitten & Bentley, 2007:406) [13]

### 2.2.7.3 Activity Diagram


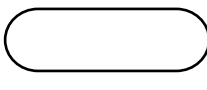
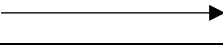
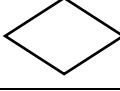
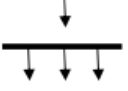
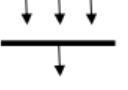

Menurut Whitten & Bentley (2007:394), *activity diagram* adalah sebuah diagram yang bisa digunakan untuk menggambarkan secara grafis alur dari sebuah proses bisnis, langkah-langkah dari sebuah *use-case*, atau logika dari sebuah objek. *Activity diagram* sangat berguna untuk model *action* yang akan dikerjakan ketika sebuah operasi dieksekusi serta hasil dari *action* tersebut [13].

Tidak semua *use-case* harus digambarkan dalam sebuah *activity diagram*. *Activity diagram* biasanya digunakan untuk *use-case* yang memiliki logika yang cukup kompleks. *Activity diagram* bisa membantu kita untuk berpikir tentang logika dari sebuah sistem.



**Gambar 2.4.** Activity Diagram: (Whitten & Bentley, 2007:392) [13].

**Tabel 2.11.** Notasi Diagram Activity

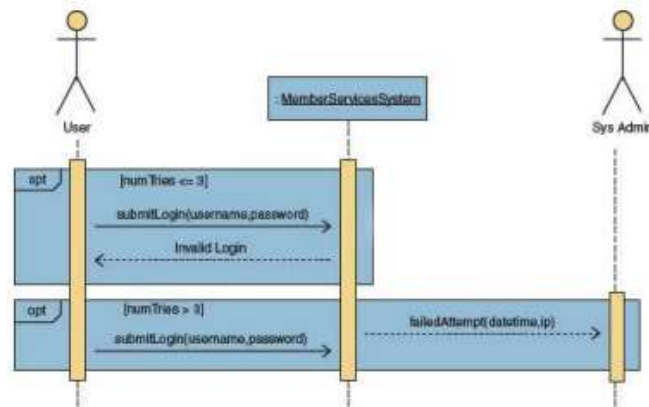
Komponen	Simbol	Penjelasan
Initial Node		Merupakan awal dari proses.
Action		Merupakan langkah-langkah individu yang membentuk aktivitas total yang ditunjukkan melalui diagram.
Flow		Menunjukkan perkembangan tindakan.
Decission		Menunjukkan kegiatan pemilihan yang menghasilkan keputusan.
Fork		Menunjukkan tindakan dilakukan secara bersamaan.
Join		Menandakan akhir dan penggabungan pross yang berlangsung bersamaan.
Activity Final		Merupakan akhir dari proses.

(Sumber: Whitten & Bentley, 2007:392) [13].

#### 2.2.7.4 Sequence Diagram

Menurut Whitten & Bentley (2007:394), *sequence diagram* adalah sebuah diagram yang menggambarkan interaksi antara *actor* dan *system* untuk sebuah skenario *use-case*. Pada tahap pembuatan *sequence diagram*, kita belum menganalisa lebih lanjut *individual object class*, namun hanya memikirkan keseluruhan system yang ada [13].

*Sequence diagram* membantu kita untuk mengidentifikasi setiap data yang masuk dan keluar dari sebuah sistem. Pada *sequence diagram* hanya sebuah skenario dari sebuah *use-case*, sehingga sebuah *use-case* dapat memiliki beberapa *sequence diagram* untuk menggambarkan keseluruhan *use-case* tersebut [13]



Gambar. 2.5. *Sequence Diagram*: (Whitten & Bentley, 2007:396) [13]

**Tabel 2.12.** Notasi Diagram Sequence

Simbol	Nama	Keterangan
	Object Lifeline	Menyatakan kehidupan suatu objek.
	Actor	Orang atau divisi yang terlibat dalam suatu sistem.
	Message	Menyatakan arah tujuan antara <i>Object Lifeline</i> .
	Message (return)	Menyatakan arah kembali dalam 1 <i>Object Lifeline</i> .
	Message (return)	Menyatakan arah kembali antara <i>Object Lifeline</i> .
	Activation	Menyatakan objek dalam keadaan aktif dan berinteraksi.

(Sumber: Whitten & Bentley, 2007:396) [13]

### 2.2.8 Teknik Pengujian Sistem

Pada pendekatan berorientasi objek, pengujian merupakan suatu persoalan yang lebih kompleks dibanding dengan pendekatan konvensional, karena keberadaan pewarisan, polymorphism, dan pengkapsulan pada

pengembangan sistem berorientasi objek menimbulkan suatu persoalan yang baru untuk perancangan kasus pengujian dan analisis hasil.

Hariyanto [14] mengungkapkan bahwa: fitur-fitur berikut berpengaruh dalam teknik-teknik pengujian yang perlu dilakukan:

- Pengkapsulan (*encapsulation*)
- Penyusunan objek-objek (*object composition*)
- Pewarisan (*inheritance*)
- Interaksi (*interaction*)
- *Polymorphism*
- Pengikatan dinamis (*dynamic binding*)
- Guna ulang (*reuse*)
- *Genericity* dan kelas abstrak

Dari kompleksnya fitur –fitur yang mempengaruhi dalam pengujian sistem berorientasi objek maka strategi pengujian dilakukan pada:

1. Pengujian unit, dimana pengujian unit dilakukan hingga beberapa level dengan alasan adanya konsep pewarisan. Pengujian unit ini bertujuan untuk menjamin setiap unit memenuhi spesifikasi. Kelas-kelas merupakan sasaran pengujian unit.
2. Pengujian integrasi, pengujian ini dilakukan untuk memverifikasi implementasi dari satu use case yang telah bekerja seperti yang diharapkan. Pengujian validitas, pengujian ini dilakukan untuk menjamin fungsi-fungsi sistem/aplikasi telah dilakukan secara benar, pengujian di eksekusi ketika satu sistem (subsistem) yang lengkap telah di rakit. Pengujian validasi ini meliputi rincian-rincian objek yang tidak tampak, fokus pada masukan dan keluaran yang tampak oleh pemakai.

#### **2.2.8.1 WhiteBox**

*White Box Testing* atau pengujian *glass box* adalah metode desain *test case* menggunakan struktur kontrol desain prosedural untuk mendapatkan *test case*. Dengan menggunakan metode *White Box* analisis sistem akan memperoleh Test Case yang:

- a) Menjamin seluruh *Independent Path* di dalam modul yang dikerjakan sekurang-kurangnya sekali.
- b) Mengerjakan seluruh keputusan logical
- c) Mengerjakan seluruh *loop* yang sesuai dengan batasannya
- d) Mengerjakan seluruh struktur data internal yang menjamin validitas [15].

Untuk melakukan proses pengujian *Test Case* terlebih dahulu dilakukan penerjemahan *flowchart* kedalam notasi *flowgraph* (aliran kontrol). Ada beberapa cara istilah saat pembuatan *flowgraph*, yaitu:

1. *Node* yaitu lingkaran pada *flowgraph* yang menggambarkan satu atau lebih perintah prosedural.
2. *Edge* yaitu tanda panah yang menggambarkan aliran kontrol dari setiap *node* harus mempunyai tujuan *node*.
3. *Region* yaitu daerah yang dibatasi oleh *node* dan *edge* dan untuk menghitung daerah diluar *flowgraph* juga harus dihitung.
4. *Predicate Node* yaitu kondisi yang terdapat pada *node* dan mempunyai karakteristik dua atau lebih *edge* lainnya.
5. *Cyclomatic Complexity* yaitu metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kekompleksan logikal program dan dapat digunakan untuk mencari jumlah path dalam suatu *flowgraph*.
6. *Independen Path* yaitu jalur melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru [15].

Rumus-rumus untuk menghitung jumlah *Independen Path* dalam suatu *flowgraph* yaitu:

1. Jumlah *region flowrgaph* mempunyai hubungan dengan *Cyclomatic Complexity (CC)*.
2.  $V(G)$  untuk *flowgraph* dapat dihitung dengan rumus :

$$a) \quad V(G) = E - N + 2$$

Dimana :

$E$  = Jumlah *edge* pada *flowrgaph*

$N$  = Jumlah *node* pada *flowrgaph*

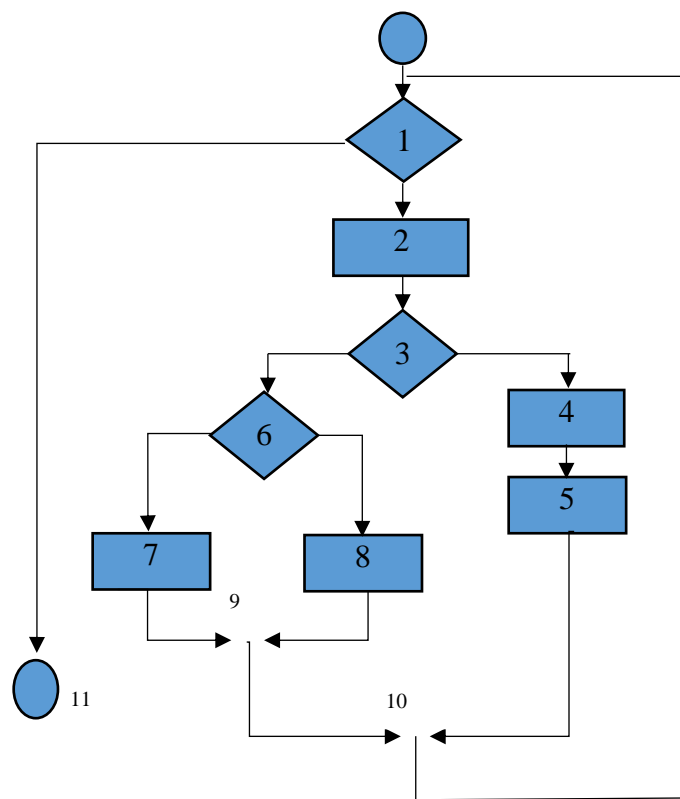
b)  $V(G) = P + 1$

Dimana :

$P$  = Jumlah *predicate node* pada *flowgraph*

Teknik pelaksanaan pengujian *White Box* ini mempunyai tiga langkah yaitu:

1. Menggambar *flowgraph* yang ditransfer oleh flowchart
2. Menghitung *Cyclomatic Complexity* untuk *flowgraph* yang telah dibuat
3. Menentukan jalur pengujian dari *flowgraph* yang berjumlah sesuai dengan *Cyclomatic Complexity* yang telah ditentukan.

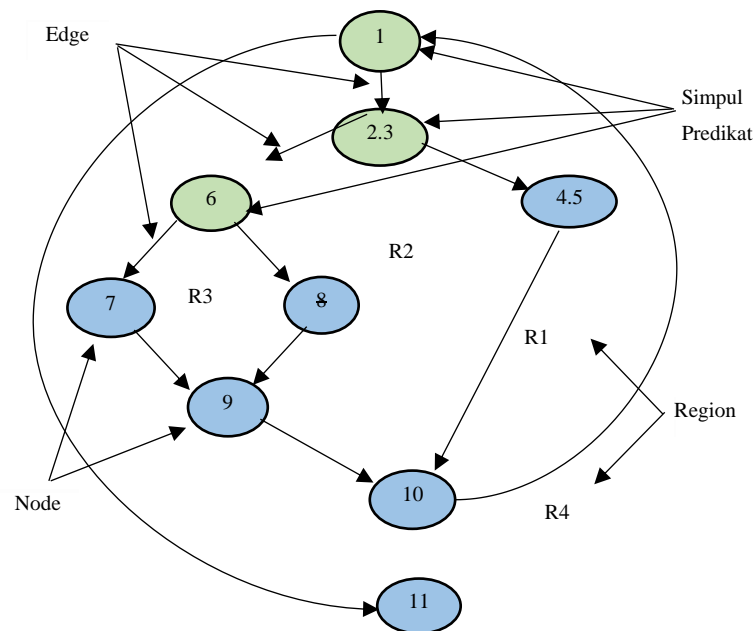


**Gambar 2.6.** Bagan Air: Roger S. Pressman [15].

Bagan alir digunakan untuk menggambarkan struktur kontrol program dan untuk menggambarkan grafik alir, harus memperhatikan representasi desain prosedural pada bagan alir. Pada gambar dibawah ini, grafik alir memetakan bagan alir tersebut ke dalam grafik alir yang sesuai (dengan mengasumsikan bahwa tidak ada kondisi senyawa yang diisikan di dalam diamond keputusan dari bagan alir tersebut). Masing-masing lingkaran, yang disebut simpul grafik alir, merepresentasikan satu atau lebih statemen prosedural. Urutan kotak proses dan



permata keputusan dapat memetakan simpul tunggal. Anak panah tersebut yang disebut edges atau links, merepresentasikan aliran kontrol dan analog dengan anak panah bagan alir. Edge harus berhenti pada suatu simpul, meskipun bila simpul tersebut tidak merepresentasikan statemen procedural [15].



**Gambar 2.6.** Flowgraph: Roger S. Pressman [15].

Dari gambar *flowgraph* di atas didapat:

Path 1 = 1 – 11

Path 2 = 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11

Path 3 = 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Path 4 = 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Path 1,2,3,4 yang telah didefinisikan diatas merupakan basis set untuk diagram alir.

*Cyclomatic complexity* digunakan untuk mencari jumlah path dalam satu flowgraph. Dapat dipergunakan rumusan sebagai berikut:

1. Jumlah region grafik alir sesuai dengan *cyclomatic complexity*.
2. *Cyclomatic complexity*  $V(G)$  untuk grafik alir dihitung dengan rumus:

$$V(G) = E - N + 2 \dots\dots\dots (1)$$

Dimana:

E = jumlah edge pada grafik alir

N = jumlah node pada grafik alir

3. *Cyclomatic complexity*  $V(G)$  juga dapat dihitung dengan rumus:

$$V(G) = P + 1 \dots\dots (2)$$

Dimana P = jumlah *predicate node* pada grafik alir

Dari Gambar di atas dapat dihitung *cyclomatic complexity*:

1. *Flowgraph* mempunyai 4 region
2.  $V(G) = 11 \text{ edge} - 9\text{node} + 2 = 4$
3.  $V(G) = 3 \text{ predicate node} + 1 = 4$

Jadi *cyclomatic complexity* untuk *flowgraph* adalah 4

*Cyclomatic Complexity* yang tinggi menunjukkan prosedur kompleks yang sulit untuk dipahami, diuji dan dipelihara. Ada hubungan antara *Cyclomatic Complexity* dan resiko dalam suatu prosedur.

**Tabel 2.13.** Hubungan antara Cyclomatic Complexity dan Resiko [15]

CC	Type of Procedure	Risk
1-4	A simple procedure	Low
5-10	A well structured and stable procedure	Low
11-20	A more complex procedure	Moderate
21-50	A complex procedure, alarming	High
>50	An error-prone, extremely troublesome, untestable procedure	Very high

#### 2.2.8.2 BlackBox

Menurut Pressman [15] *Black-Box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black-Box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan perilaku (*behavior*) atau kesalahan kinerja
5. Inisialisasi dan pemutusan kesalahan

Tes ini dirancang untuk menjawab beberapa pertanyaan-pertanyaan berikut ini:

- a. Bagaimana validitas fungsional diuji?
- b. Bagaimana perilaku dan kinerja sistem diuji?
- c. Apa kelas *input* akan membuat kasus uji yang baik?
- d. Apakah sistem *sensitive* terhadap nilai input tertentu?
- e. Bagaimana batas-batas kelas data yang terisolasi?
- f. Kecepatan dan volume data seperti apa yang dapat ditolerir sistem?
- g. Efek apakah yang akan menspesifikasikan kombinasi data dalam sistem operasi?

#### 1. Ciri-Ciri Black Box Testing

- a. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
- b. *Black box testing* bukan teknik alternatif daripada *white box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.
- c. *Black box testing* melakukan pengujian tanpa pengetahuan detil struktur internal dari sistem atau komponen yang dites. juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*

#### 2. Jenis teknik *design* tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan.

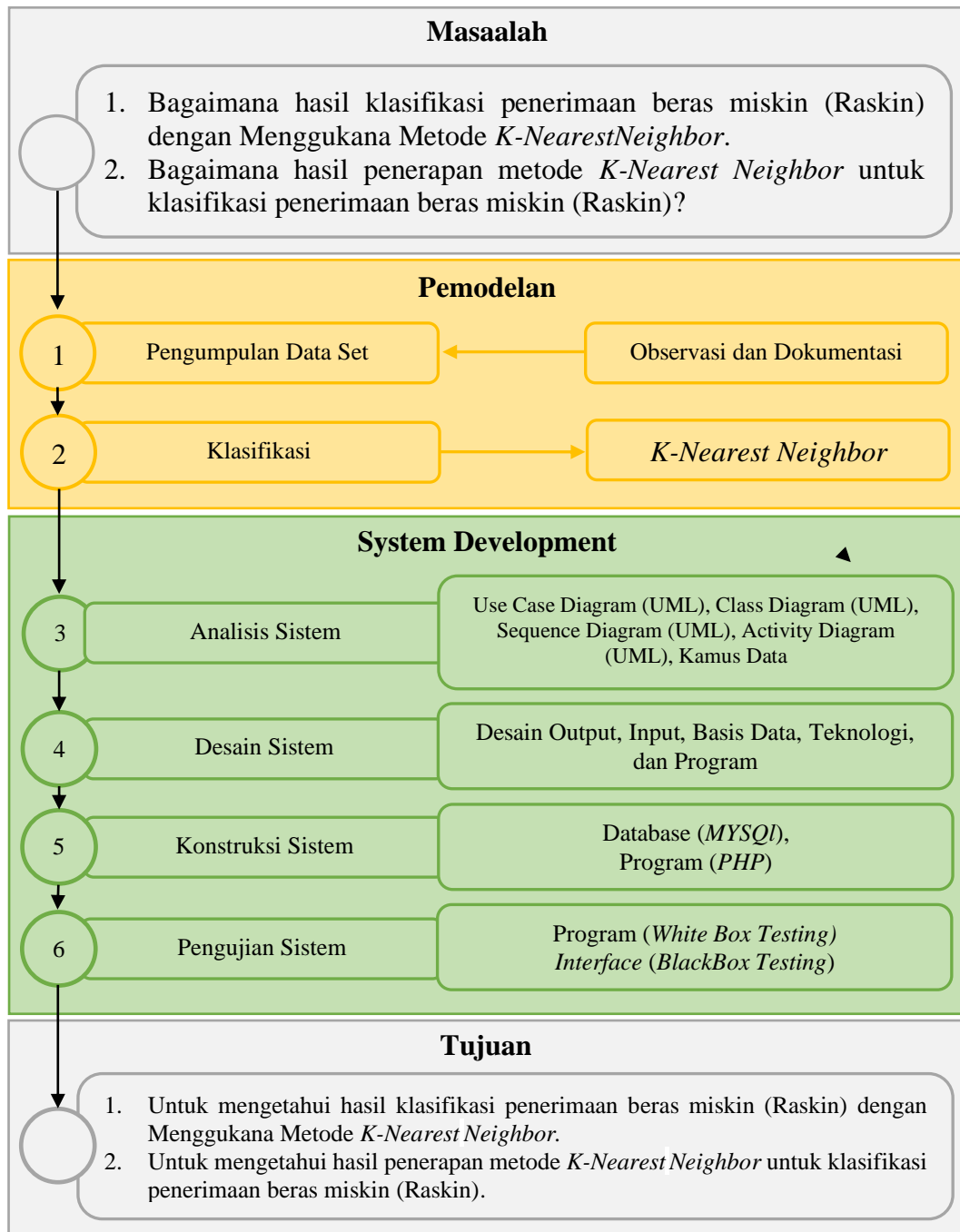
- a. *Equivalence Class Partitioning*
- b. *Boundary Value Analysis*
- c. *State Transitions Testing*

d. *Cause-Effect Graphing*

3. Kategori *error* yang akan diketeahui melalui *black box testing*
  - a. Fungsi yang hilang atau tak benar
  - b. *Error* dari antar-muka
  - c. *Error* dari struktur data atau akses eksternal database
  - d. *Error* dari kinerja atau tingkah laku

*Error* dari inisialisasi dan terminasi

### 2.3 Kerangka Pemikiran



**Gambar 2.7:** Kerangka Pemikiran

## BAB III

### METODE PENELITIAN

#### 3.1 Jenis, Metode dan Objek Penelitian

Dipandang dari tingkat penerapannya penelitian ini merupakan penelitian terapan sedangkan metode penelitian yang digunakan adalah metode deskriptif. Yang menjadi objek penelitian pada penelitian ini adalah penerimaan beras miskin (Raskin). Penelitian ini di mulai pada Januari 2022 sampai dengan April 2022 di Kelurahan Tenda.

#### 3.2 Pengumpulan Data

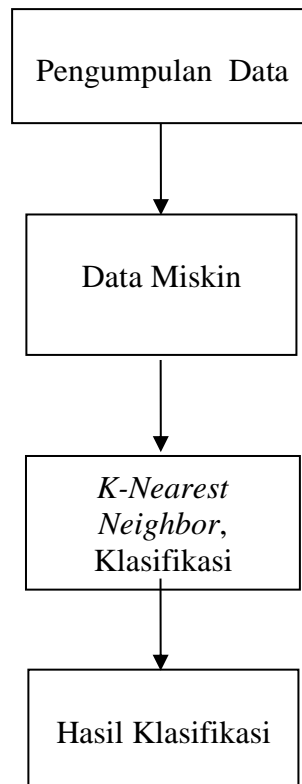
Data primer penelitian ini adalah data yang dikumpulkan langsung oleh peneliti pada kantor Kelurahan Tenda Kota Gorontalo. Data yang peneliti dapatkan berupa data jumlah penduduk miskin daftar peberima bantuan pemerintah Kota Gorontalo, sedangkan data dari penelitian ini adalah metode kepustakaan, yaitu telaah dari teori-teori yang sudah ada berupa teori-teori tentang Klasifikasi, *K-Nearest Neighbor* maupun tentang masyarakat penerima beras miskin (Raskin).

Adapun variabel inputan dengan tipe datanya masing-masing ditunjukkan pada tabel 3.1 berikut :

**Tabel 3.1:** Atribut Data

No	Nama	Type	Keterangan
1	Pangan	Varchar	Variable Input
2	Sandang	Varchar	Variabel Input
3	Papan	Varchar	Variabel Input
4	Penghasilan	Integer	Variabel Input
5	Kesehatan	Varchar	Variabel Input
6	Pendidikan	Varchar	Variabel Input
7	Kepemilikan Rumah	Varchar	Variabel Input
8	Air Bersih	Integer	Variabel Input
9	Penerangan Rumah	Integer	Variabel Input
10	Kartu Bantuan	Varchar	Variabel Input

### 3.3 Pemodelan



**Gambar: 3.1** Pemodelan

### 3.4 Pengembangan Model

Prosedur atau langkah pokok dalam klasifikasi menggunakan metode *K-Nearest Neighbor* untuk klasifikasi masyarakat miskin yang berhak menerima beras miskin.

### 3.5 Konstruksi Sistem

Pada tahap ini menerjemahkan hasil kedalam kode-kode program kemudian membangun sistemnya. Alat bantu yang digunakan pada tahap ini adalah *PHP* dengan menggunakan database.

### 3.6 Analisa Sistem

Analisis sistem menggunakan pendekatan berorientasi *procedural/structural*:

- a) Use Case Diagram, menggunakan alat bantu UML
- b) Class Diagram, menggunakan alat bantu UML
- c) Sequence Diagram menggunakan alat bantu UML
- d) Activity Diagram menggunakan alat bantu UML

Kamus Data menggunakan alat bantu Ms. Word.

### **3.7 Tahap Pengujian**

Tahap ini dilakukan setelah semua model selesai dibuat, dan program dapat berjalan, di mana seluruh perangkat lunak, program tambahan, dan semua program yang terlibat dalam pembangunan sistem diuji untuk memastikan sistem dapat berjalan sesuai dengan rancangan atau belum, pengujian yang dilakukan dengan dua teknik pengujian, yaitu:

#### *a. Whitebox Testing*

Dalam pengujian *whitebox* dengan membuat bagan alir program, *tracing* program, grafik alir, pengujian *basis path* serta perhitungan *cyclomatic complexity*.

#### *b. Blackbox Testing*

Pengujian *blackbox* yang termasuk dalam tahap ini yaitu menguji antarmuka sistem, apakah sebuah sistem setelah diberikan ke pengguna dapat dioperasikan atau tidak.



## BAB IV

### HASIL PENELITIAN

#### 4.1 Hasil Pengumpulan Data

Tabel 4. 1 : Hasil Pengumpulan Data

NO	NAMA	ALAMAT	F.MAKAN DALAM 1 HARI	F.KONSUMSI DAGING/AYAM/ SUSU	PENGHASIL AN PER- BULAN	JUMLAH TANGGU NGAN
1	RASEMAN ABDULLAH	KEL. TENDA	3	2	900000	1
2	TONI	KEL. TENDA	1	1	900000	3
3	OTEN	KEL. TENDA	2	1	900000	2
4	UDIN DALI	KEL. TENDA	3	3	900000	1
5	IMAM NASARU	KEL. TENDA	3	2	900000	2
6	MUH. KARIM	KEL. TENDA	2	1	900000	3
7	ABD. KASIM	KEL. TENDA	2	1	900000	3
8	SUCIPTO. B	KEL. TENDA	3	3	900000	4
9	MARJUN.S	KEL. TENDA	2	3	900000	2
10	MUSTAPA.U.J	KEL. TENDA	2	2	900000	1
11	SALIM	KEL. TENDA	2	1	900000	0
12	ISMAIL.P	KEL. TENDA	3	2	900000	2
13	MUH. NOLVAN. A	KEL. TENDA	3	3	900000	4
14	ABD. KADIR. NOU	KEL. TENDA	2	2	900000	0
15	MURI.GAGA	KEL. TENDA	3	2	900000	1
16	FIKRI	KEL. TENDA	3	2	900000	0
17	SANDI.N	KEL. TENDA	3	3	900000	2
18	IDRIS. M	KEL. TENDA	2	1	900000	1
19	MISWAR BAU	KEL. TENDA	3	2	900000	0
20	SUWITO.M	KEL. TENDA	3	2	900000	3
73	SULEMAN	KEL. TENDA	3	2	900000	1



Tabel 4. 2 : Hasil Normalisasi Data

No	Frekuensi Makan Dalam 1 Hari	Konsumsi Daging/Susu/Ayam Dalam 1 Minggu	Penghasilan Per-Bulan	Jumlah Tanggungan Anak	Status Kepemilikan Rumah	Kemampuan Berobat	Label
1	1	0.5	0.8333	0.25	1	0.5	0
2	0	0	0.6667	0.75	1	0.5	1
3	0.5	0	0.5833	0.5	1	0.5	1
4	1	1	0.9167	0.25	1	0.5	0
5	1	0.5	0.5833	0.5	1	0.5	1
6	0.5	0	0.3333	0.75	1	0.5	1
7	0.5	0	0.3333	0.75	1	0.5	1
8	1	1	0.9167	1	1	1	0
9	0.5	1	0.8333	0.5	1	0.5	1
10	0.5	0.5	0.6667	0.25	1	1	0
11	0.5	0	0.0833	0	1	0	1
12	1	0.5	0.9167	0.5	1	0.5	0
13	1	1	0.9167	1	0	1	0
14	0.5	0.5	0.8333	0	1	0.5	1
15	1	0.5	0.3333	0.25	1	0.5	1
16	1	0.5	0.3333	0	1	0.5	1
17	1	1	0.9167	0.5	1	0.5	0
18	0.5	0	0.3333	0.25	1	0.5	1
19	1	0.5	0.9167	0	1	1	0
20	1	0.5	0.8333	0.75	0.5	0.5	1
73	1	0.5	0.3333	0.25	1	0.5	1

#### 4.2 Hasil Pemodelan

##### a. Menentukan data training dan data testing

Berdasarkan pengumpulan dataset di atas, sebelum dilakukan perhitungan metode K-NN terlebih dahulu dilakukan pembagian data training dan data testing, dari 73 data yang dikumpulkan diambil 50 sebagai data training (set) dan 23 data sebagai data testing (uji). Berikut

sampel perhitungan dengan menggunakan 3 data uji yaitu pada tabel berikut :

Tabel 4. 2 : Sampel Data Uji

No	Frekuensi Makan Dalam 1 Hari	Konsumsi Daging/Susu/Ayam Dalam 1 Minggu	Penghasilan Per-Bulan	Jumlah Tanggungan Anak	Status Kepemilikan Rumah	Kemampuan Berobat	Label
1	0.5	0.5	0.3333	0.5	1	0.5	?
2	0.5	0	0.3333	0.5	1	0.5	?
3	1	1	0.5000	0.5	0	0.5	?

- b. Menghitung kedekatan kasus antara data Training dan data testing dengan menggunakan persamaan *Euclidian Distance* sebagai berikut:

$$\text{dist}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Jarak Data Uji terhadap data training 1 adalah:

$$d1 = \sqrt{(1 - 0.5)^2 + (0.5 - 0.5)^2 + (0.8333 - 0.3333)^2 + (0.25 - 0.5)^2 + (1 - 1)^2 + (0.5 - 0.5)^2}$$

$$d1 = \sqrt{0.25 + 0 + 0.25 + 0.0625}$$

$$d1 = \sqrt{0.5625} = 0.750000$$

Jarak Data Uji 1 terhadap data training 1 adalah = 0.750000

Demikian seterusnya untuk menentukan jarak data uji 1 terhadap data training 2 hingga data training 50. Hasil perhitungan selengkapnya disajikan pada tabel berikut.

Tabel 4. 3 : Hasil perhitungan jarak data uji 1 terhadap data training (Distance)

Nomor	Data	Euclidian Distance
1	D1, D51	0.750000

2	D2, D51	0.820738
3	D3, D51	0.559017
4	D4, D51	0.950146
5	D5, D51	0.559017
6	D6, D51	0.559017
7	D7, D51	0.559017
8	D8, D51	1.157704
9	D9, D51	0.707107
10	D10, D51	0.650854
...	...	...
50	D50, D51	1.118034

Tabel 4.4 : Penentuan nilai K

Data Training	Label	Euclidian Distance	Minimum Euclidian Distance	K=1	K=3	K=5	K=7
D31	LAYAK	0.263523138	1	LAYAK	LAYAK	LAYAK	LAYAK
D45	LAYAK	0.416666667	2		LAYAK	LAYAK	LAYAK
D27	LAYAK	0.5	3		LAYAK	LAYAK	LAYAK
D34	LAYAK	0.5	3		LAYAK	LAYAK	LAYAK
D35	LAYAK	0.5	3		LAYAK	LAYAK	LAYAK
D47	LAYAK	0.5	3		LAYAK	LAYAK	LAYAK
D3	LAYAK	0.559016994	4			LAYAK	LAYAK
D5	LAYAK	0.559016994	4			LAYAK	LAYAK
D6	LAYAK	0.559016994	4			LAYAK	LAYAK
D7	LAYAK	0.559016994	4			LAYAK	LAYAK
D15	LAYAK	0.559016994	4			LAYAK	LAYAK
D18	LAYAK	0.559016994	4			LAYAK	LAYAK
D32	LAYAK	0.559016994	4			LAYAK	LAYAK
D41	LAYAK	0.559016994	4			LAYAK	LAYAK
D43	LAYAK	0.559016994	4			LAYAK	LAYAK
D10	TIDAK LAYAK	0.65085414	5			TIDAK LAYAK	TIDAK LAYAK
D25	LAYAK	0.65085414	5			LAYAK	LAYAK
D9	LAYAK	0.707106781	6				LAYAK

D14	LAYAK	0.707106781	6				LAYAK
D16	LAYAK	0.707106781	6				LAYAK
D30	TIDAK LAYAK	0.707106781	6				TIDAK LAYAK
D46	TIDAK LAYAK	0.707106781	6				TIDAK LAYAK
D1	TIDAK LAYAK	0.75	7				TIDAK LAYAK
D37	TIDAK LAYAK	0.75	7				TIDAK LAYAK
D48	TIDAK LAYAK	0.75	7				TIDAK LAYAK
D51	?			LAYAK	LAYAK	LAYAK	LAYAK

Nilai  $k$  pada algoritma KNN mendefinisikan berapa banyak tetangga yang akan diperiksa untuk menentukan klasifikasi. Misalnya, dari data D51 terhadap data set yang ada jika  $k=1$ , maka tetangga terdekat yang di temukan terdapat satu orang saja atau bisa dikatakan dengan nilai  $k =1$  tidak mendapatkan hasil yang baik. Dan Ketika diujikan dengan menggunakan  $k= 3$  maka akan ditugaskan ke kelas yang sama dengan tetangga terdekatnya. Dan mendapatkan lebih banyak hasil rekomendasi yang layak tanpa rekomendasi tidak layak. Sementara pada pengujian  $k= 5$  dan  $7$  Nilai  $k$  yang lebih tinggi dapat memiliki varian yang terlalu tinggi, tetapi terdapat nilai rekomendasi yang tidak layak. Sedangkan nilai  $k$  yang lebih besar dapat menyebabkan bias yang tinggi.

### 3.8

### 4.3 Hasil Pengembangan Sistem

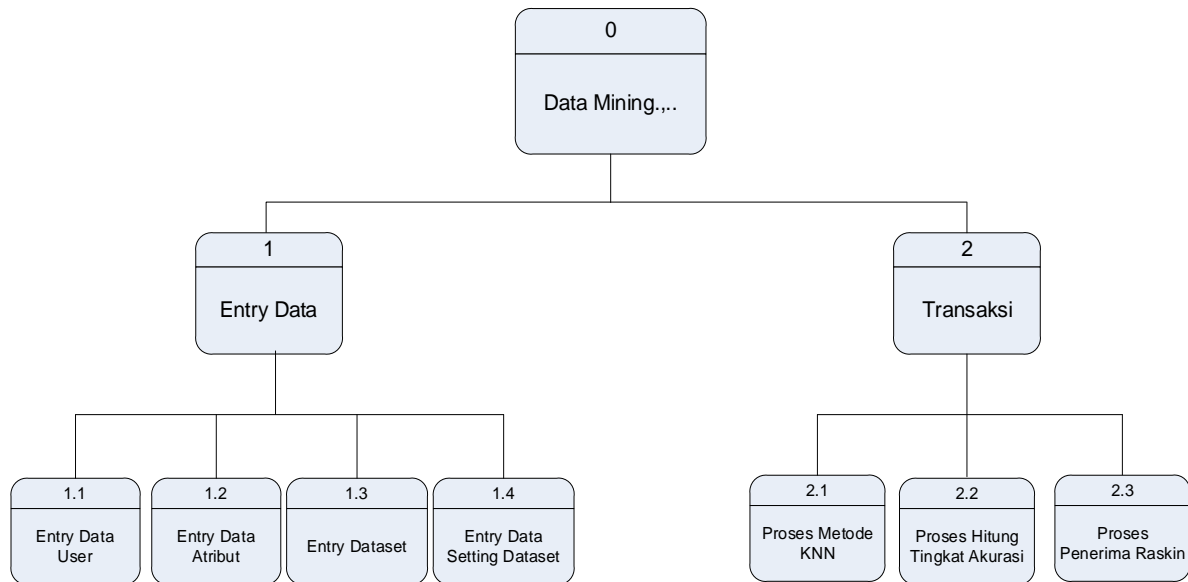
#### 4.3.1 Desain Sistem Secara Umum

##### 4.3.1.1 Diagram Konteks



**Gambar 4.1:** Diagram Konteks

#### 4.3.1.2 Diagram Berjenjang

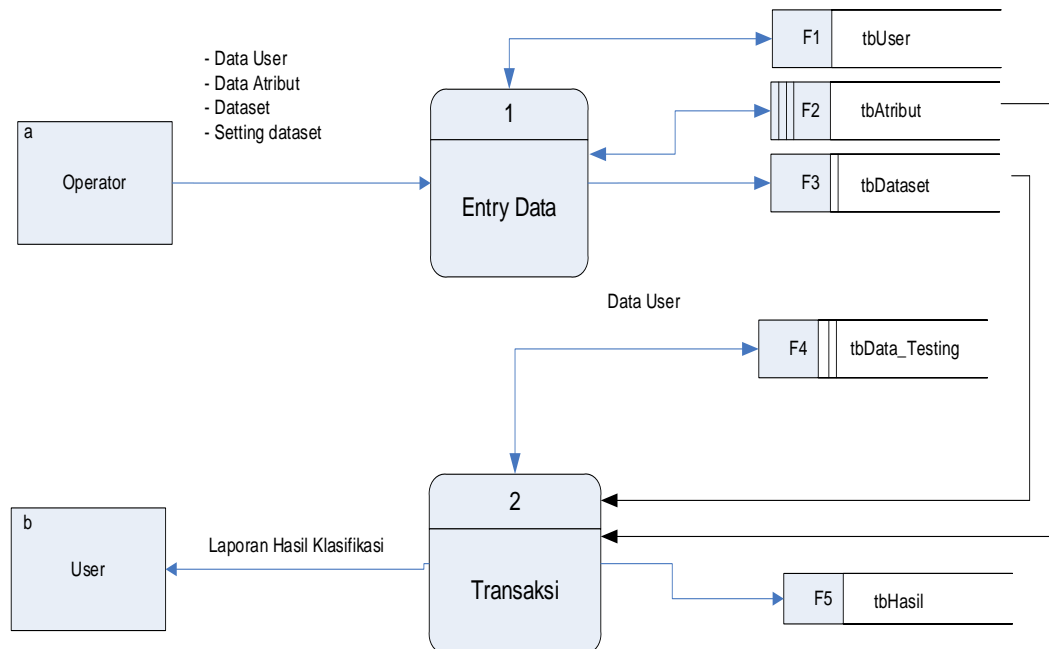


**Gambar 4.2:** Diagram Berjenjang



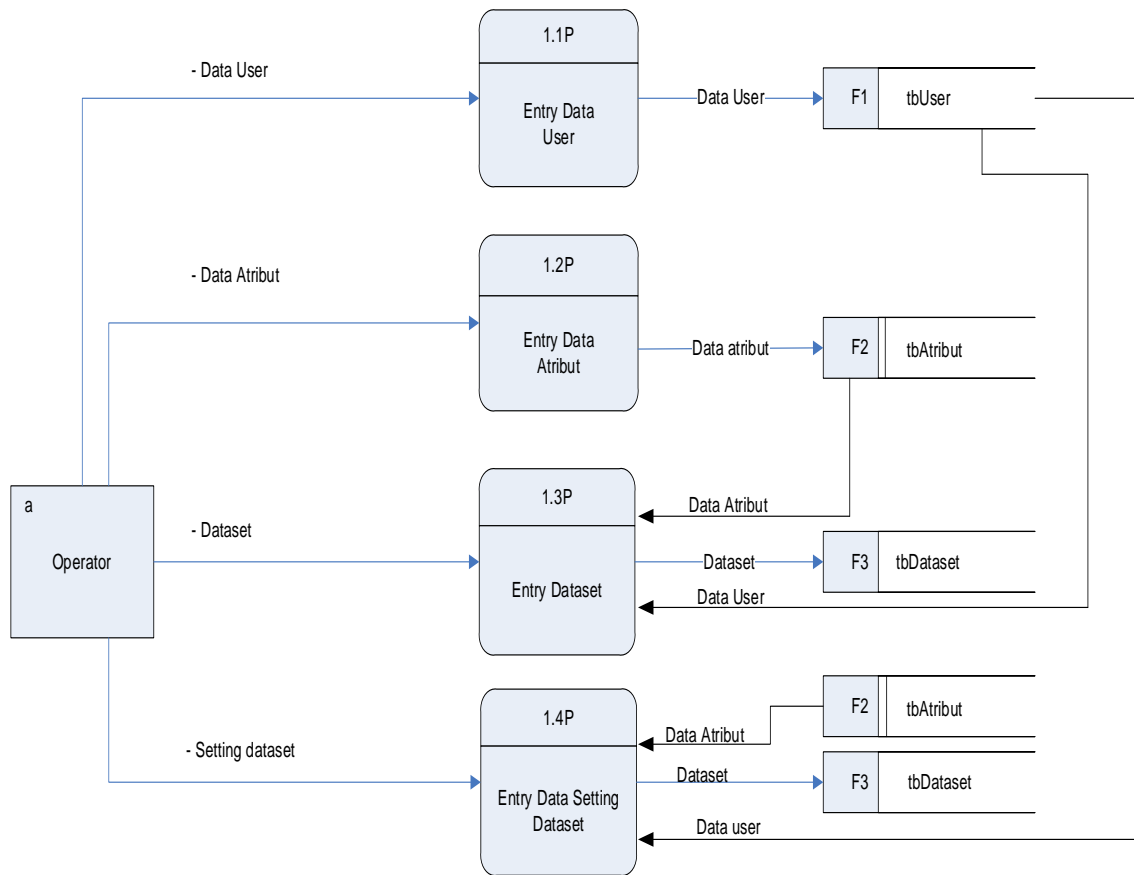
### 4.3.1.3 Diagram Arus Data

#### 4.3.1.3.1 DAD Level 0



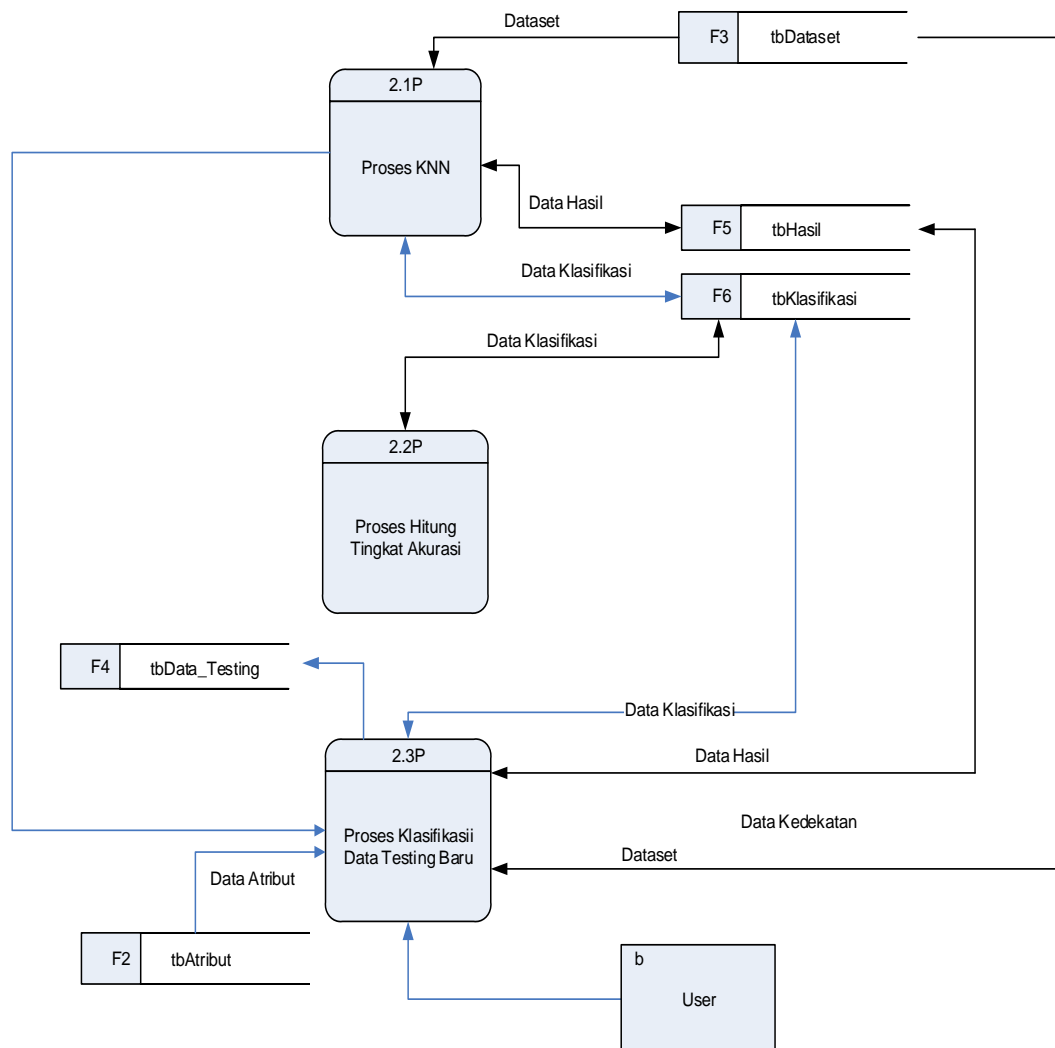
**Gambar 4.3:** DAD Level 0

#### 4.3.1.3.2 DAD Level 1 Proses 1



**Gambar 4.4:** DAD Level 1 Proses 1

#### 4.3.1.3 DAD Level 1 Proses 2



**Gambar 4.5:** DAD Level 1 Proses 2

#### 4.3.1.4 Kamus Data

Kamus Data atau *Data Dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem pendukung keputusan. Kamus data digunakan untuk merancang input, file-file/database dan output. Kamus data dibuat berdasarkan arus data yang mengalir pada DAD, dimana didalamnya terdapat struktur dari arus data secara detail.

Tabel 4. 4 : Kamus Data Akurasi

Nama Arus Data : Data Akurasi Penjelasan : Input Data Akurasi Periode : Setiap ada penambahan data Akurasi Bentuk Data : Dokumen Arus Data : a-2, a-2.2P,3.3P-a				
No	Field Name	Type	Size	Ket
1	Nomor	N	5	Nomor
2	Nilai K	N	1	Nilai K
3	Klasifikasi	C	5	Klasifikasi
4	Keterangan	C	15	Keterangan

Tabel 4. 5 : Kamus Data Atribut

Nama Arus Data : Data Atribut Penjelasan : Input Data Atribut Periode : Setiap ada penambahan data Atribut Bentuk Data : Dokumen Arus Data : a-1,1-F2,F2-1,F2-2, F2-3, a-1.2P, 1.2P-F2,F2-1.3P,F2-1.4P F2-2.3P				
No	Field Name	Type	Size	Ket
1	Kode_Atribut	C	3	Kode_Atribut
2	Nama_Atribut	C	100	Nama_Atribut
3	Jenis_Atribut	C	15	Jenis_Atribut

#### 4.3.1.5 Design Output Secara Umum

##### Daftar Output Yang Didesain

Untuk : Kantor Kelurahan tenda

Tahap : Rancangan sistem secara umum

Tabel 4. 6 : Daftar Output Yang Didesain

Kode Output	Nama Output	Tipe Output	Format Output	Media Output	Alat Output	Distribusi	Periode
O-001	Laporan Dataset	Internal	Tabel	Kertas	Printer	Admin	Non Periodik
O-002	Laporan Hasil Klasifikasi	Internal	Tabel	Kertas	Printer	Admin	Non Periodik

#### 4.3.1.6 Desain Input Secara Umum

##### Daftar Input Yang Didesain

Untuk : Kantor Kelurahan Tenda

Tahap : Rancangan sistem secara umum

Tabel 4. 7 : Daftar Input Yang Di Desain

Kode Input	Nama Input	Sumber Input	Periode
I-001	Entry Data Atribut	Admin	Non Periodik
I-002	Entry Dataset	Admin	Non Periodik
I-003	Entry Seting Dataset	Admin	Non Periodik
I-004	Entry Pemodelan KNN	Admin	Non Periodik
I-005	Entry Hitung Akurasi	Admin	Non Periodik
I-006	Penerima Raskin	Admin	Non Periodik
I-007	Laporan Dataset	Admin	Non Periodik
I-008	Laporan Hasil Klasifikasi	Admin	Non Periodik
I-009	Hasil Akurasi	Admin	Non Periodik

#### 4.3.1.7 Desain Database Secara Umum

##### **DAFTAR FILE YANG DI DESAIN**

Untuk : Kantor Kelurahan Tenda

Tahap : Rancangan sistem secara umum

Tabel 4. 8 : Daftar File Yang Di desain

<b>Kode File</b>	<b>Nama File</b>	<b>Tipe File</b>	<b>Media File</b>	<b>Organisasi File</b>	<b>Field Kunci</b>
F1	Tbuser	Master	Hard Disk	Index	User_Id
F2	Tbatribut	Master	Hard Disk	Index	Kode_Atribut,
F3	Tbdataset	Master	Hard Disk	Index	Nomor
F4	tbdata_testing	Proses/Pemodelan KNN	Hard Disk	Index	Nomor
F5	Tbakurasi	Proses/Laporan Akurasi	Hard Disk	Index	Nomor,+ Nilai K
F6	Tbhasil	Laporan	Hard Disk	Index	Nomor, + No_Testing,

#### 4.3.2 Desain Arsitektur

Agar sistem dapat berjalan secara maksimal maka disarankan untuk menggunakan perangkat hardware dan software sebagai berikut :

1. Processor Intel 600 MHz
2. Ram Minimal 2 GB
3. VGA minimal 16 Bit
4. Harddisk minimal ruang kosong 100 MB
5. Operating Sistem minimal Windows 7 ke atas
6. Tools : Xampp, MySql Conector ODBC, CRRedist2010

### 4.3.3 Desain Interface

#### 4.3.3.1 Mekanisme User

Tabel 4. 9 : Interface Design – Mekanisme User

Users	Kategori	Akses Input	Akses Output
Admin	Administrator	All	All
User	Operator	<ul style="list-style-type: none"> <li>- Prediksi Datatesting baru</li> <li>- Lap. Hasil</li> <li>- Utility</li> </ul>	Lap. Hasil

### 4.3.4 Desain Data

Data yang diperoleh pada sistem .....ini menggunakan format :

1. Microsoft Excel (.xlsx) sebagai tempat penyimpanan external
2. Database MySql untuk mengolah dan menyimpan data

Keduanya dihubungkan dan dimanupulasi dengan teknik *disconnected* data

#### 4.3.4.1 Struktur Data

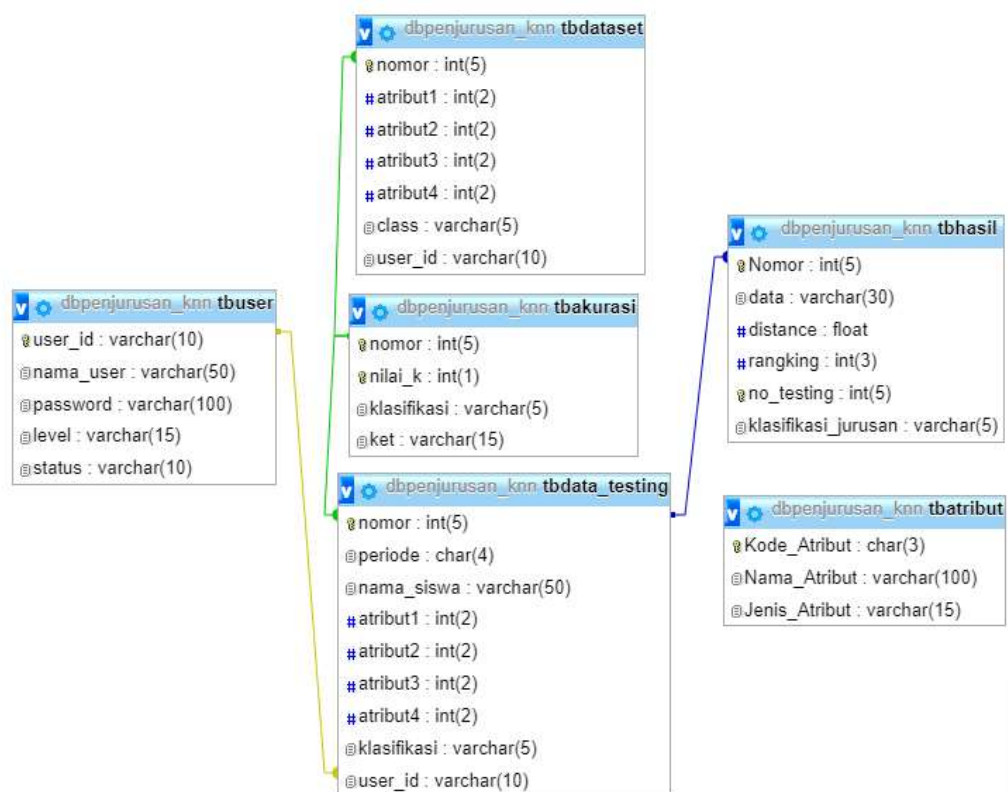
Tabel 4. 10 : Data Desain : Struktur Data - Data User

Nama File : tbUser Tipe File : Master Primary Key : User_Id Forigen Key : - Media : Harddisk Fungsi : Merupakan data pengguna aplikasi Struktur Data :				
No	Field Name	Type	Size	Ket
1	User_Id	Varchar	10	User Id
2	Username	Varchar	50	User Name
3	Password	Varchar	100	Password
4	Level	Varchar	15	Level
5	Status	Varchar	10	Status

Tabel 4. 11 : Data Desain : Struktur Data - Data Atribut

Nama File : tbAtribut Tipe File : Master Primary Key : Kode_Atribut Forigen Key : - Media : Harddisk Fungsi : Merupakan data Atribut Struktur Data :				
No	Field Name	Type	Size	Ket
1	Kode_Atribut	Char	3	Kode Atribut
2	Nama_Atribut	Varchar	100	Nama Atribut
3	Jenis_Atribut	Varchar	15	Jenis Atribut

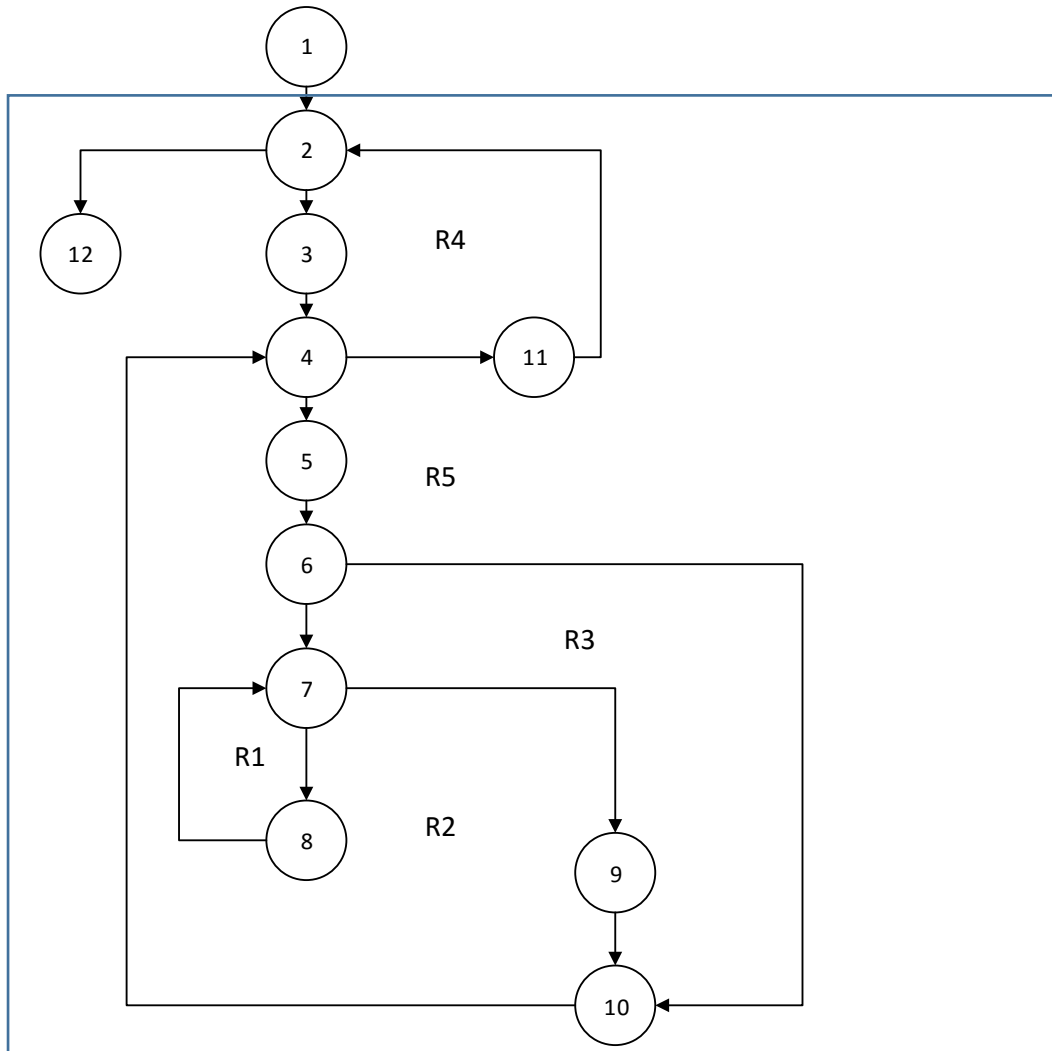
#### 4.3.4.2 Relasi



Gambar 4.6: Desain Relasi Antar Tabel



#### 4.3.5 Flowgraph Untuk Pengujian White Box



**Gambar 4.6:** Flowgraph untuk Pengujian White Box

#### 4.3.6 Perhitungan CC Pada Pengujian White Box

Dari *flowgraph* diatas, maka didapatkan :

Diketahui :

Region (R) = 5

Node (N) = 12

Edge (E) = 1

Predicate Node (P) = 4

Rumus :

$V(G) = (E - N) + 2$  atau

$V(G) = P + 1$

Penyelesaian :

$V(G) = (15 - 12) + 2 = 5$

$V(G) = 4 + 1 = 7$

(R1, R2, R3, R4, R5, R6, R7)

#### 4.3.7 Path Pada Pengujian White Box

Tabel 4. 12 : Path Pengujian White Box

No	Path	Ket
1	1-2-3-4-5-6-7-8-7-...	Ok
2	1-2-3-4-5-6-7-9-10-4-...	Ok
3	1-2-3-4-5-6-10-4-...	Ok
4	1-2-3-4-11-2-12	Ok
5	1-2-12	Ok

#### 4.3.8 Pengujian *Black Box*

Tabel 4. 13 : Hasil Pengujian Black Box Terhadap Beberapa Proses

Input/Event	Fungsi	Hasil yg Diharapkan	Hasil Uji
Input nama user dan password yg benar	Menampilkan halaman menu utama	Halaman menu utama tampil	Sesuai
Input nama user yg salah	Menampilkan pesan kesalahan “maaf user id salah”	Pesan kesalahan input nama user tampil	Sesuai
Input password yg salah	Menampilkan pesan kesalahan “maaf password salah”	Pesan kesalahan input password tampil	Sesuai
Klik Master Data User	Menampilkan Form Data User	Halaman form Daftar Data User	Sesuai
Klik Master Data Atribut	Menampilkan form data atribut	Halaman form list data atribut	Sesuai
Klik Master Dataset	Menampilkan form Dataset	Halaman form Dataset	Sesuai
Klik Master Seting Dataset	Menampilkan form Seting Dataset	Halaman form Seting Dataset	Sesuai
Klik Proses Pemodelan KNN	Menampilkan form Proses Pemodelan KNN	Halaman form Proses Pemodelan KNN	Sesuai
Klik tombol Proses Hitung Akurasi	Menampilkan Proses Hitung Akurasi (confision matrix)	Menampilkan Proses Hitung Akurasi (confision matrix)	Sesuai

<b>Input/Event</b>	<b>Fungsi</b>	<b>Hasil yg Diharapkan</b>	<b>Hasil Uji</b>
Klik Proses Penerima Raskin	Menampilkan form Penerima Raskin	Halaman form proses Penerima Raskin	Sesuai
Klik Laporan Dataset	Menampilkan form Laporan Dataset	Halaman form Laporan Dataset	Sesuai
Klik Laporan Hasil Klasifikasi	Menampilkan from Hasil Klasifikasi	Seluruh Hasil Klasifikasi	Sesuai
Klik Laporan Hasil Akurasi	Menampilkan from Laporan Hasil Akurasi	Laporan Hasil Akurasi	Sesuai
Keluar	Menampilkan halaman “Benar ingin keluar dari sistem ?”	Keluar dari program	Sesuai

## BAB V

### HASIL PENELITIAN DAN PEMBAHASAN

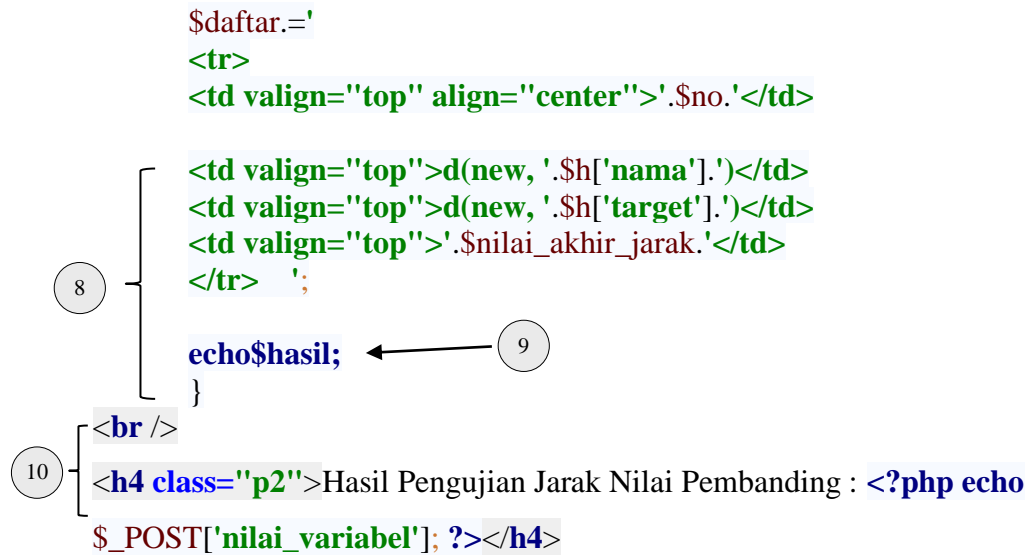
#### 1.1 Hasil Penelitian

##### 1.1.1 Hasil Pengujian Sistem

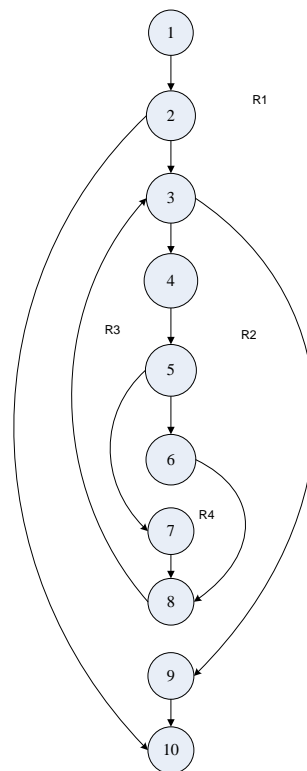
##### 1.1.1.1 Pengujain White Box

##### 1. PseudocodeProses Klasifikasi Penerima Beras Raskin





## 2. Flowgraph Proses Klasifikasi Penerima Beras Raskin



**Gambar 5.1** Flowgraph Proses Klasifikasi Penerima Beras Raskin

### a. Menghitung Nilai *Cyclomatic Complexity* (CC)

Dimana :

$$\begin{aligned}
 \text{Region(R)} &= 4 \\
 \text{Node(N)} &= 10 \\
 \text{Edge(E)} &= 12 \\
 \text{Predicate Node(P)} &= 3 \\
 V(G) &= E - N + 2 \\
 &= 12 - 10 + 2 \\
 &= 4 \\
 V(G) &= P + 1 \\
 &= 3 + 1 \\
 &= 4
 \end{aligned}$$

Jadi *cyclomatic complexity* untuk *flowgraph* proses klasifikasi Penerima Beras Raskin adalah 4. Berdasarkan tabel hubungan antara *cyclomatic complexity* dan Resiko menurut Mc Cabe, menunjukkan bahwa nilai CC 4 masuk dalam *type of procedure a well structured and stable procedure* (strukturnya baik dan prosedur stabil) serta resikonya *Low* (rendah)

**b. Menentukan Basis Path :**

**Basis set yang dihasilkan dari jalur independent path secara linier adalah jalur sebagai berikut :**

**Jalur 1 : 1-2-10**

**Jalur 2 : 1-2-3-9-10**

**Jalur 3 : 1-2-3-4-5-7-8-3-...**

**Jalur 4 : 1-2-3-4-5-6-8-3-...**

Ketika aplikasi dijalankan, maka terlihat bahwa semua basis path yang dihasilkan telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, system ini telah memenuhi syarat.

### 1.1.1.2 Pengujian Black Box

**Tabel 5.1**Tabel Pengujian *Black Box*

<b>Input/Event</b>	<b>Fungsi</b>	<b>Hasil</b>	<b>HasilUji</b>
Masukkan password salah, klik login	Menguji validasi password	Tampil pesan‘Username dan password salah’.	Sesuai
Masukkan username salah, klik Login	Menguji validasi username	Tampil pesan ‘Username dan password salah’.	Sesuai
Masukkan nama username dan password yang benar, klik login	Menguji validasi username dan password	Tampil ‘Selamat Datang admin’	Sesuai
Klik menu Testing, Data Testing	Menampilkan data testing	Tampil klasifikasi data testing	Sesuai
Klik menu Testing, Tambah Data	Menambahkan data testing	Tampil form Tambah Data Testing	Sesuai
Input data testing, klik tombol Simpan	Menguji proses penyimpanan data testing	Tampil pesan ‘Data Sudah Tersimpan’	Sesuai
Klik menu Testing, Uji Data	Menampilkan data uji testing	Tampil Hasil Normalisasi	Sesuai
Klik menu Training, Data Training	Menampilkan data training	Tampil klasifikasi data training	Sesuai
Klik menu Training, Tambah Data	Menambahkan data training	Tampil form Tambah Klasifikasi Data Training	Sesuai
Input data training, klik tombol Simpan	Menguji proses penyimpanan data training	Tampil pesan ‘Data Sudah Tersimpan’	Sesuai
Klik menu Training, Uji Data	Menampilkan data uji training	Tampil pengujian jarak data training dan data testing	Sesuai



<b>Pilih nilai pembanding, klik tombol Proses</b>	<b>Mekakukan proses uji data training</b>	<b>Tampil hasil pengujian jarak nilai pembanding</b>	<b>Sesuai</b>
<b>Klik menu Laporan Hasil, Hasil Analisa</b>	<b>Melakukan proses klasifikasi</b>	<b>Tampil Tampilkan berdasarakan kedekatan</b>	<b>Sesuai</b>
<b>Klik Ubah Password</b>	<b>Melakukan perubahan password</b>	<b>Tampil form Ubah Password</b>	<b>Sesuai</b>
<b>Klik menu Data User</b>	<b>Menampilkan data user</b>	<b>Tampil Data User Login</b>	<b>Sesuai</b>
<b>Klik + Tambah Data, pada menu Data User</b>	<b>Menambahkan data user login yang baru</b>	<b>Tampil form Tambah Data User Login</b>	<b>Sesuai</b>
<b>Klik Logout</b>	<b>Menguji proses logout</b>	<b>Tampil halaman login</b>	<b>Sesuai</b>

Ketika aplikasi dijalankan, maka terlihat bahwa semua pengujian black box yang dihasilkan telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan aplikasi, sistem ini telah memenuhi syarat.

## **5.2 Pembahasan**

### **5.2.1 Deskripsi Kebutuhan Hardware/Software**

Penulis dalam mengembangkan Website ini menggunakan bahasa pemrograman PHP(*Hypertext Preprocessor*) dan Basis Data MySQL.

Pada dasarnya, untuk implementasi sistem ini membutuhkan beberapa konfigurasi dasar, diantaranya :

#### **1. *Hardware dan Software***

**Spesifikasi yang disarankan untuk komputer**

- a. Processor setara Pentium IV 1.8 Ghz atau lebih
- b. RAM (Memory) 256 MB atau lebih

- c. HDD 40 GB atau lebih.
- d. Monitor SVGA dengan Resolusi 1024 X 768
- e. Dan Peralatan I/O Lainnya
- f. Windows XP, Vista atau Windows 7
- g. Browser Mozilla Firefox, Internet Explorer dan Opera untuk membuka Web

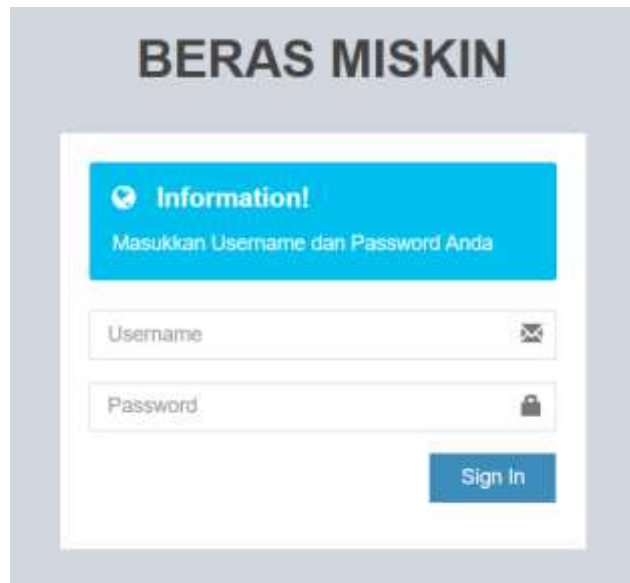
## **2. *Brainware***

Yaitu sumber daya manusia yang terlibat di dalam mengoperasikan serta mengatur sistem komputer. Sumber daya yang dibutuhkan dengan karakteristik sebagai berikut memiliki kemampuan dasar tentang komputer dan proses yang berlangsung di dalamnya

### **5.2.2 Langkah-Langkah Menjalankan Sistem**

Untuk menjalankan program cukup dengan mengetikkan alamat pada tab address.

### 5.2.2.1 Tampilan Halaman Login Admin



**Gambar 5.2** Tampilan Form Login Admin

Pada tampilan halaman login ini, user menginput username dan password untuk masuk ke halaman adminweb. Apabila salah maka akan tampil Pesan "Username dan password salah", ulangi lagi dengan mengisi username dan password yang benar kemudian klik tombol Login.

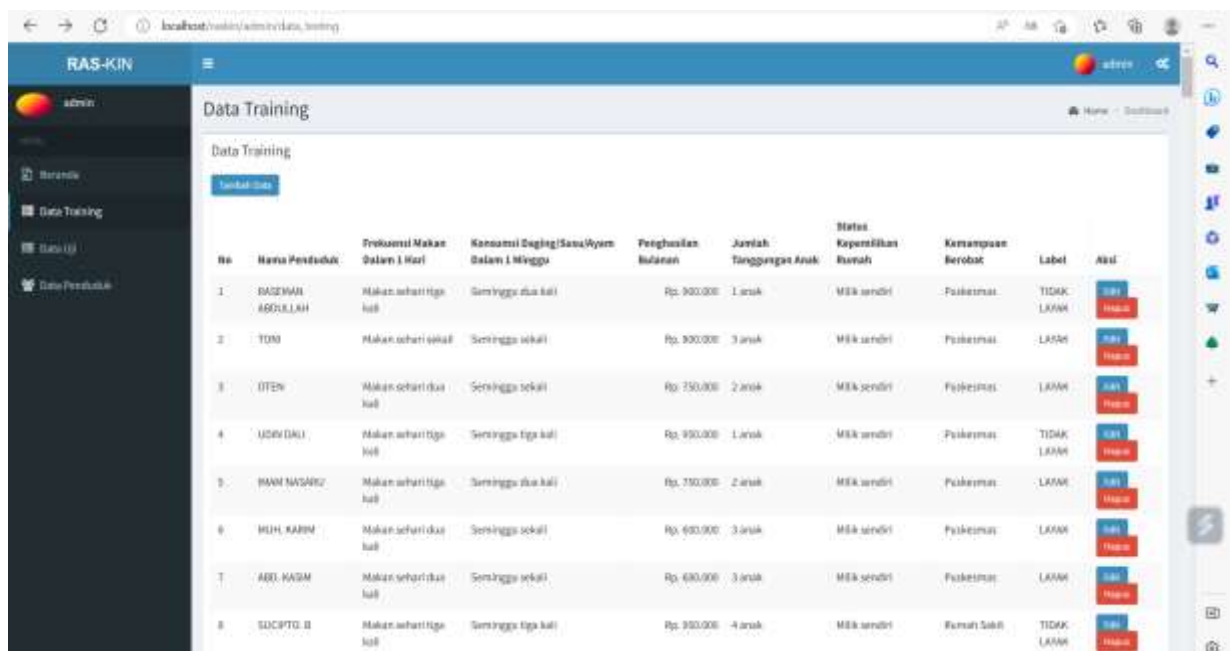
### 5.2.2.2 Tampilan Halaman Home Admin



**Gambar 5.3** Tampilan Halaman Home Admin

Halaman ini berfungsi untuk menampilkan Halaman Home dari admin setelah melakukan proses login sebagai admin. Terdiri atas menu-menu yang terdapat di lajur atas yaitu Home, Testing, Training, Data User, Logout. Masing-masing menu tersebut memiliki kegunaan yang berbeda-beda.

### 5.2.2.3 Tampilan Halaman View Data Testing



No	Nama Penduduk	Frekuensi Makan Dalam 1 Hari	Konsumsi Daging/Susu/Keju Dalam 1 Minggu	Penghasilan Bulanan	Jumlah Tanggungan Anak	Status Kepemilikan Rumah	Kemampuan Berotat	Label	Aksi
1	BASZAWI ABDULLAH	Makan sehari tiga kali	Seringnya dua kali	Rp. 900.000	1 anak	Milik sendiri	Puskesmas	TIDAK LAYAN	<a href="#">Edit</a> <a href="#">Hapus</a>
2	TOM	Makan sehari sekali	Seringnya sekali	Rp. 900.000	3 anak	Milik sendiri	Puskesmas	LAYAN	<a href="#">Edit</a> <a href="#">Hapus</a>
3	OTEN	Makan sehari dua kali	Seringnya sekali	Rp. 750.000	2 anak	Milik sendiri	Puskesmas	LAYAN	<a href="#">Edit</a> <a href="#">Hapus</a>
4	USW DAU	Makan sehari tiga kali	Seringnya tiga kali	Rp. 950.000	1 anak	Milik sendiri	Puskesmas	TIDAK LAYAN	<a href="#">Edit</a> <a href="#">Hapus</a>
5	RAMI NABARZ	Makan sehari tiga kali	Seringnya dua kali	Rp. 750.000	2 anak	Milik sendiri	Puskesmas	LAYAN	<a href="#">Edit</a> <a href="#">Hapus</a>
6	MUH. KARIM	Makan sehari dua kali	Seringnya sekali	Rp. 600.000	3 anak	Milik sendiri	Puskesmas	LAYAN	<a href="#">Edit</a> <a href="#">Hapus</a>
7	ABD. KAM	Makan sehari dua kali	Seringnya sekali	Rp. 600.000	3 anak	Milik sendiri	Puskesmas	LAYAN	<a href="#">Edit</a> <a href="#">Hapus</a>
8	SUCIPTO. D	Makan sehari tiga kali	Seringnya tiga kali	Rp. 950.000	4 anak	Milik sendiri	Rumah Saki	TIDAK LAYAN	<a href="#">Edit</a> <a href="#">Hapus</a>

**Gambar 5.4** Tampilan Halaman View Data Testing

Halaman ini digunakan untuk melihat data-data testing. Untuk menambahkan data testing yang baru, klik +Tambah Data. Untuk menghapus data klik Control Hapus. Untuk mengubah data, klik Control Edit.

#### 5.2.2.4 Tampilan Form Tambah Data Testing

The screenshot displays the 'Data Training' interface of the RAS-KIN system. It features a sidebar menu with options like 'Beranda', 'Data Training', 'Data Uj', and 'Data Penjualan'. The main content area is titled 'Data Training' and contains four identical data entry forms labeled 'Data Ke-1' through 'Data Ke-4'. Each form includes the following fields:

- Nama Pemilik:
- Frekuensi Makan:
- Frekuensi Konsumsi Daging/Susu:
- Penghasilan Bulanan:
- Jumlah Tanggungan Anak:
- Kepemilikan Rumah:
- Kesehatan:
- Label:

**Gambar 5.5** Tampilan Form Tambah Data Testing

Halaman ini digunakan untuk menginput data testing yang baru. Dimulai dengan mengisi data Nama, Frekuensi Makan, Frekuensi Konsumsi Daging/Susu, Penghasilan Bulanan, Jumlah Tanggungan Anak, Kepemilikan Rumah, Kesehatan.. Untuk menyimpan data, klik tombol Simpan. Untuk membatalkan proses, klik tombol Batal.

#### 5.2.2.5 Tampilan Form Edit Data Testing

The screenshot displays the 'Edit Data Testing' interface of the RAS-KIN system. It features a sidebar menu with options like 'Beranda', 'Data Training', 'Data Uj', and 'Data Penjualan'. The main content area is titled 'Edit Data Testing' and contains a single data entry form with the following pre-filled values:

- Nama Pemilik: IKSEMAN ABDULLAH
- Frekuensi Makan: 3
- Frekuensi Konsumsi Daging/Susu/Ayam: 2
- Penghasilan Bulanan: 300000
- Jumlah Tanggungan: 1
- Kepemilikan Rumah: 0
- Kemampuan Berobat: 2
- Label: 8

At the bottom right of the form, there are two buttons: 'Simpan' (Save) and 'Batal' (Cancel).

**Gambar 5.6** Tampilan Form Edit Data Testing

Halaman ini digunakan untuk mengubah data testing, dimulai dengan mengisi data Nama, Frekuensi Makan, Frekuensi Konsumsi Daging/Susu, Penghasilan Bulanan, Jumlah Tanggungan Anak, Kepemilikan Rumah, Kesehatan yang baru. Untuk menyimpan perubahan data klik tombol Simpan. Untuk membatalkan proses perubahan klik tombol Batal.

#### 5.2.2.6 Tampilan Halaman Uji Data Testing

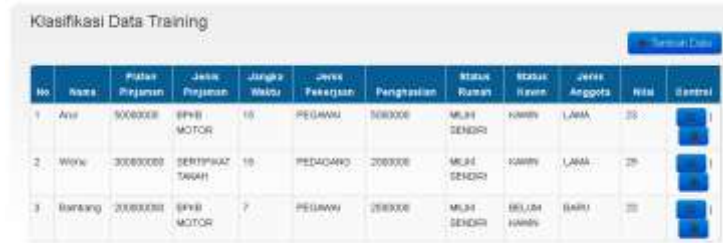


No	Nama Penduduk	Frekuensi Makan Dalam 1 Hari	Konsumsi Daging/Susu/Ayam Dalam 1 Minggu	Penghasilan Bulanan	Jumlah Tanggungan Anak	Status Kepemilikan Rumah	Kemampuan Berobat	Label
1.	Asnita Jabbar	0,5	0,5	0,5833	0,5	1	0,5	LAYAK
2.	Krisna	0	1	1	0,25	0,5	1	TIDAK LAYAK
3.	Evi	1	0,5	0,5833	1	0,5	0,5	LAYAK

**Gambar 5.7** Tampilan Halaman Uji Data Testing

Halaman ini digunakan untuk menampilkan data uji testing, uji data testing yang ditampilkan yaitu Nilai minimum dataset, nilai maksimum, range nilai, No, Nama, Nilai, Keterangan, dan Hasil Normalisasi.

### 5.2.2.7 Tampilan Halaman View Data Training



No	Nama	Plafon Pinjaman	Jenis Pinjaman	Jangka Waktu	Jenis Pekerjaan	Penghasilan	Status Rumah	Status Kawin	Jenis Anggota	Nilai	Kontrol
1	Andi	5000000	SPIN MOTOR	18	PEKERJA	500000	MUDA SENDER	KAWIN	JAWA	25	[Edit] [Hapus]
2	Wahyu	30000000	SEKIPINJAT TAWAN	18	PEDAKAWIN	200000	MUDA SENDER	KAWIN	JAWA	25	[Edit] [Hapus]
3	Bambang	20000000	SPIN MOTOR	7	PEKERJA	250000	MUDA SENDER	SEKAWIN	BARU	25	[Edit] [Hapus]

**Gambar 5.8** Tampilan Halaman View Data Training

Halaman ini digunakan untuk melihat data-data training. Data training yang ditampilkan yaitu No, Nama, Plafon Pinjam, Jenis Pinjam, Jangka Waktu, Jenis Pekerjaan, Penghasilan, Status Rumah, Status Kawin, Jenis Anggota, Status Kredit, dan Nilai. Untuk menambahkan data training yang baru, klik +Tambah Data. Untuk menghapus data klik Control Hapus. Untuk mengubah data, klik Control Edit.

### 5.2.2.8 Tampilan Form Tambah Data Training



**Gambar 5.9** Tampilan Form Tambah Data Training

Halaman ini digunakan untuk menginput data Training yang baru. Dimulai dengan mengisi dataNama, Plafon Pinjam, Jenis Pinjam, Jangka Waktu, Jenis Pekerjaan, Penghasilan, Status Rumah, Status Kawin, Jenis Anggota, Status Kredit,

dan Nilai. Untuk menyimpan data Training, klik tombol Simpan. Untuk membatalkan proses, klik tombol Batal.

### 5.2.2.9 Tampilan Form Edit Data Training

**Gambar 5.10** Tampilan Form Edit Data Training

Halaman ini digunakan untuk mengubah data training, dimulai dengan mengisi data Nama, Plafon Pinjam, Jenis Pinjam, Jangka Waktu, Jenis Pekerjaan, Penghasilan, Status Rumah, Status Kawin, Jenis Anggota, Status Kredit, dan Nilai yang baru. Untuk menyimpan perubahan data klik tombol Simpan. Untuk membatalkan proses pengubahan klik tombol Batal.

### 5.2.2.10 Tampilan Halaman Uji Data Training

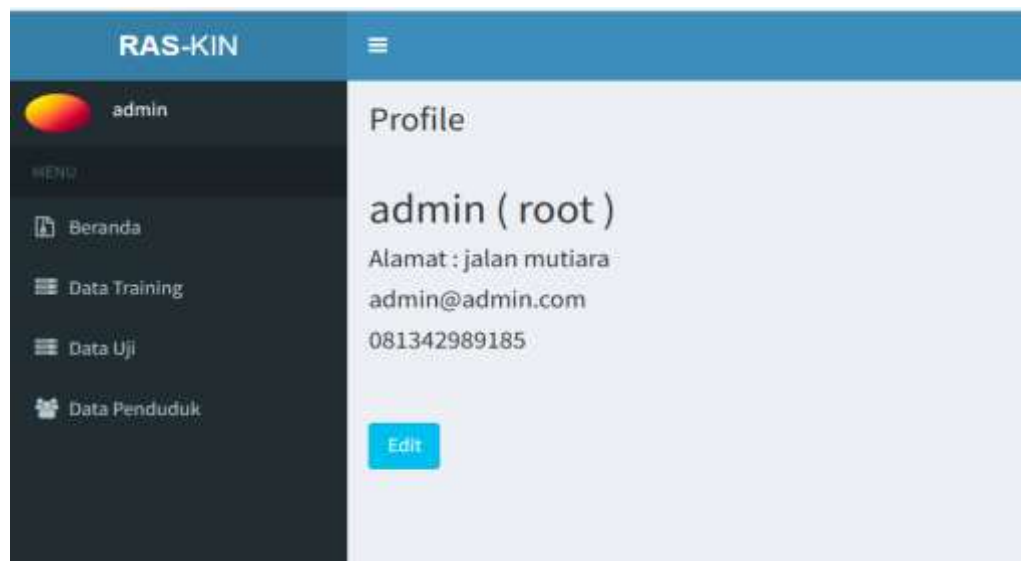
No	Objek	Target	Jumlah Aritama Variabel
1	d(nek, Marjan Hirda)	d(nek, LAMCAR)	0
2	d(nek, Herina nur)	d(nek, LAMCAR)	-0.222222222222222
3	d(nek, Rosana sangoni)	d(nek, LAMCAR)	0.777777777777778



### Gambar 5.11 Tampilan Halaman Uji Data Training

Halaman ini digunakan untuk menampilkan data uji training, uji data training yang ditampilkan yaitu Hasil pengujian jarak nilai pembandingan lyang terdiri dari Object, Target, dan Jarak Antara Variabel.

#### 5.2.2.11 Tampilan Halaman View Data User



**Gambar 5.12**Tampilan Halaman View Data User

Halaman ini digunakan untuk melihat data-data User. Data User yang ditampilkan yaitu No, Nama, Username, dan Password. Untuk menambahkan data User yang baru, klik +Tambah Data. Untuk menghapus data User klik Control Hapus.

### 5.2.2.12 Tampilan Halaman Laporan Hasil Analisa

KNN

Nilai K : 1

Tetangga pada label "LAYAK"

No	Nama	Jarak	Label
1	ANTON	0.25	LAYAK

Hasil Data Uji

No	Nama Penduduk	Frekuensi Makan Dalam 1 Hari	Konsumsi Daging/Susu/Ayam Dalam 1 Minggu	Penghasilan Per-Bulan	Jumlah Tanggungan Anak	Status Kepemilikan Rumah	Kemampuan Bersobat	Label
1	Asma Jabbar	0.5	0.5	0.5833	0.5	1	0.5	LAYAK

**Gambar 5.13**Tampilan Halaman Laporan Hasil Analisa

Halaman ini digunakan untuk menampilkan hasil analisa/ hasil klasifikasi, dimuali dengan memasukkan nilai kedekatan dan mengklik tombol proses. Data hasil yang ditampilkan yaitu Hasil analisa kedekatan, hasil analisa confusion matrix, serta kesimpulan hasil klasifikasi.

## BAB VI KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan pada Kantor Kelurahan tenda Kota Gorontalo dan pembahasan yang telah diuraikan sebelumnya, maka dapat ditarik suatu kesimpulan bahwa:

1. Klasifikasi Penerima Beras Raskin dengan Menggunakan Metode *K-Nearest Neighbor* dapat direkayasa, sehingga membantu dan memudahkan pihak Kantor Kelurahan Tenda dalam mengklasifikasikan Penerima Beras Raskin.
2. Peneliti juga dapat mengetahui hasil penerapan metode K-Nearest Neighbor (KNN) dalam mengklasifikasi bahwa nilai  $K=3$ . Dapat ditugaskan kedalam hasil rekomendasi yang layak tanpa rekomendasi tidak layak.

### 1.2 Saran

Setelah Melakukan Penelitian dan pembuatan Klasifikasi Penerima Beras Raskin dengan Menggunakan Metode K-Nearest Neighbor, ada beberapa saran yang perlu diperhatikan mencapai tujuan yang diharapkan, yaitu sebagai berikut:

1. Penulis berharap kepada pihak kantor Kantor Kelurahan Tenda Kota Gorontalo untuk dapat Menggunakan aplikasi ini yaitu Klasifikasi Penerima Beras Raskin dengan Menggunakan Metode K-Nearest Neighbor agar lebih mempermudah dalam proses klasifikasi Penerima beras miskin.
2. Perlu dilakukan bimbingan teknis dala, pengguna Aplikasi Klasifikasi Penerima beras Raskin dengan Menggunakan Metode K-Nearest Neighbor.

3. Penelitian ini diharapkan dapat direkayasa Kembali dengan menggunakan metode lainnya seperti Naïve Bayes Classifier.

## DAFTAR PUSTAKA

- [1] R. F. Damanik, T. Supriana, and T. Sebayang, “Analisis Efektivitas Distribusi Beras Miskin (Raskin),” *J. Sos. Econ. Agric. Agribus.*, vol. 3, no. 2, pp. 1–13, 2014.
- [2] A. B. Cakti, “SISTEM PENDUKUNG KEPUTUSAN SELEKSI PENERIMA BERAS UNTUK KELUARGA MISKIN,” p. 2012, 2012, [Online]. Available: <http://digilib.uinsgd.ac.id/id/eprint/1954>
- [3] L. R. Ananda, “Clustering Untuk Menentukan Calon Mahasiswa Berprestasi,” *Jiti*, vol. 1, no. 2, pp. 16–19, 2018.
- [4] A. Heriyanto, “Penerapan Metode K-Nearest Neighbor (KNN) Untuk Klasifikasi Stunting Pada Balita,” *Publ. Ilm. Univ. Muhammadiyah Jember*, 2021.
- [5] Hasran, “Klasifikasi Penyakit Jantung Menggunakan Metode K-Nearest Neighbor,” *Indones. J. Data Sci.*, vol. 1, no. 1, pp. 1–4, 2020.
- [6] S. A. P. Aji, H. Oktavianto, and Q. A’yun, “Klasifikasi Penerima Bantuan Dana Desa Menggunakan Metode KNN (K-Nearest Neighbor) (Studi Kasus : Desa Andongsari Kecamatan Ambulu Kabupaten Jember),” 2020.
- [7] E. Purwaningsih and E. Nurelasari, “Penerapan K-Nearest Neighbor Untuk Klasifikasi Tingkat Kelulusan Pada Siswa,” *Syntax J. Inform.*, vol. 10, no. 01, pp. 46–55, 2021, [Online]. Available: <https://journal.unsika.ac.id/index.php/syntax/article/download/5173/2749>
- [8] Y. Yahya and W. Puspita Hidayanti, “Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Efektivitas Penjualan Vape (Rokok Elektrik) pada ‘Lombok Vape On,’” *Infotek J. Inform. dan Teknol.*, vol. 3, no. 2, pp. 104–114, 2020, doi: 10.29408/jit.v3i2.2279.
- [9] A. M. Argina, “Penerapan Metode Klasifikasi K-Nearest Neighbor pada Dataset Penderita Penyakit Diabetes,” *Indones. J. Data Sci.*, vol. 1, no. 2, pp. 29–33, 2020, doi: 10.33096/ijodas.v1i2.11.
- [10] Sutarbi, Tata. 2013. *Analisis Sistem Informasi*. Yogyakarta: Andi.
- [11] Witten, Jeffrey L, et all, *Metode Desain & Analisis Sistem*, Edisi 6, Edisi International, Mc Graw Hill, Andi, Yogyakarta: 2004.
- [12] Kadir, Abdul. 2003. *Pengenalan Sistem Informasi*. Edisi I. Yogyakarta. Andi Yogyakarta.
- [13] Bently, Lonnie D, Jeffrey L Whitten, (2007). *Systems Analysis and Design for the Global Enterprise Seventh Edition*, New York: McGraw-Hill.
- [14] Hariyanto, Bambang, 2004. *Sistem Informasi Basis Data: Pemodelan, Perancangan, dan Terapannya*. Informatika, Bandung.
- [15] S. R. Presman, *Rekayasa Perangkat Lunak : Pendekatan Praktis*. Yogyakarta: Andi, 2002

# LAMPIRAN



Similarity Report ID: old:25211:34427174

## PAPER NAME

SKRIPSI\_T3116126\_MUKTI ALI MOHAM  
MAD.docx

## AUTHOR

SKRIPSI\_T3116126\_MUKTI ALI MOH mu  
ktisemma@gmail.com

## WORD COUNT

8659 Words

## CHARACTER COUNT

53024 Characters

## PAGE COUNT

63 Pages

## FILE SIZE

2.4MB

## SUBMISSION DATE

Apr 24, 2023 9:57 PM GMT+8

## REPORT DATE

Apr 24, 2023 9:58 PM GMT+8

## ● 26% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

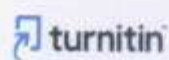
- 26% Internet database
- 11% Publications database
- Crossref database
- Crossref Posted Content database
- 8% Submitted Works database

## ● Excluded from Similarity Report

- Cited material
- Small Matches (Less than 30 words)







Similarity Report ID: oic:25211:34427174

**26% Overall Similarity**

Top sources found in the following databases:

- 26% Internet database
- 11% Publications database
- Crossref database
- Crossref Posted Content database
- 8% Submitted Works database

**TOP SOURCES**

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	jurnal.yoctobrain.org	4%
	Internet	
2	adoc.pub	4%
	Internet	
3	andl.ddns.net	3%
	Internet	
4	123dok.com	3%
	Internet	
5	id.scribd.com	3%
	Internet	
6	watiindra314.blogspot.com	2%
	Internet	
7	repository.stikes-yrsds.ac.id	1%
	Internet	
8	slideshare.net	1%
	Internet	

Defined('BASEPATH') OR exit('No direct script access allowed');

```

class Profile extends Admin_Controller {

    public function __construct(){
        parent::__construct();
        // $this->output->enable_profiler(TRUE);
        $this->load->model("m_user");
        $this->load->model("m_log");
        $this->load->helper('array');
        $this->load->library("pagination");
        $this->load->library('upload');
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data['page_title'] = "Profile ";

        $data['user']      =      $this->m_user->getUser(      $this->session-
        >userdata('user_id') );
        $this->load->view("_admin/_template/header");
        $this->load->view("_admin/_template/sidebar_menu");
        $this->load->view("_admin/profile/View",$data);
        $this->load->view("_admin/_template/footer");

    }

    public function edit()
    {
        $data['page_title'] = "edit ";
    }

```

```

    $this->form_validation-
>set_rules('user_profile_fullname','user_profile_fullname','trim|required');
    $this->form_validation-
>set_rules('user_profile_email','user_profile_email','trim|required');
    $this->form_validation-
>set_rules('user_profile_address','user_profile_address','trim|required');
    $this->form_validation-
>set_rules('user_profile_phone','user_profile_phone','trim|required');

    $message="lengkapi semua form !!";
    if ($this->form_validation->run() === true)
    {
        $data_user_profile['user_profile_fullname']    =    $this->input-
>post('user_profile_fullname');
        $data_user_profile['user_profile_email']        =    $this->input-
>post('user_profile_email');
        $data_user_profile['user_profile_address']      =    $this->input-
>post('user_profile_address');
        $data_user_profile['user_profile_phone']        =    $this->input-
>post('user_profile_phone');

        // echo var_dump( $data_store );
        $name= $_FILES['document_file']['name'];
        $size= $_FILES['document_file']['size'];
        $type= $_FILES['document_file']['type'];
        $tmp_name = $_FILES['document_file']['tmp_name'];

        $data_user_profile['user_profile_image_path']    =    $this->input-
>post('user_old_images') ;

        $location = './upload/user/';

```

```

$source_file =
$location.$data_user_profile['user_profile_image_path'];
$im = ImageCreateFromJpeg($source_file);

$message="terjadi kesalahan saat mengirim data";
if(!empty( $name ) && $size < 16777216 ){
    $location = './upload/user/';
    $data_user_profile['user_profile_image_path'] =time()."-
".$name;
    if(move_uploaded_file($tmp_name,
$location.$data_user_profile['user_profile_image_path']))
    {
        $old_photo = $this->input->post('user_old_images') ;
        @unlink($location.$old_photo );
    }else{
        if( $result )
        {
            $this->session->set_flashdata('info', array(
                'from' => 0,
                'message' => 'foto tidak terupload'
            ));
            redirect(site_url('admin/profile'));
            return;
        }
    }
}

$data_user_profile_param['user_id'] = $this->session-
->userdata('user_id');

if( !empty( $this->input->post('user_password') ) )

```

```

    {
        $data_user['user_password'] = md5( $this->input-
>post('user_password') );
        $this->m_user->update( $data_user, $data_user_profile_param );
    }
    $result = $this->m_user->update_profile( $data_user_profile,
$data_user_profile_param );
    if( $result )
    {
        $data = array(
}

```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

**UNIVERSITAS ICHSAN GORONTALO**

**FAKULTAS ILMU KOMPUTER**

**SK MENDIKNAS NOMOR 84/D/O/2001**



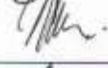


JL. Achmad Nadjamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo

**Berita Acara Perbaikan/Revisi Ujian SKRIPSI**

Pada hari ini, Rabu 3 Mei 2023, Pukul 13.00-15.00 Wita. Telah dilaksanakan Ujian SKRIPSI mahasiswa Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Nama : Mukti Ali Mohammad  
 Nim : T3116126  
 Pembimbing I : Asmaul Husnah Nasrullah, M.Kom  
 Pembimbing II : Rofiq Harun, M.Kom  
 Judul SKRIPSI : Klasifikasi Penerima Beras Miskin (Raskin) Menggunakan Metode K-Nearest Neighbor

Oleh Komite Seminar sebagai berikut :

No	Komite Seminar	Status	Tanda Tangan
1	Zohrahayaty, M.Kom	Ketua	
2	Hastuti Dalai, M.Kom	Anggota	
3	Roys Pakaya, M.Kom	Anggota	
4	Asmaul Husnah Nasrullah, M.Kom	Anggota	
5	Rofiq Harun, M.Kom	Anggota	

## RIWAYAT HIDUP



**Nama** : Mukti Ali Mohammad  
**NIM** : T311626  
**Tempat dan Tanggal Lahir** : Palu, 14 April 1997  
**Agama** : Islam  
**E-Mail** : [muktisemma@gmail.com](mailto:muktisemma@gmail.com)

### **Riwayat Pendidikan :**

1. Peneliti Tahun 2010, Telah Menyelesaikan Pendidikan di Sekolah Dasar Negeri 1 Tangeban Kecamatan Lamala, Kabupaten Banggai, Provinsi Sulawesi Tengah
2. Peneliti Tahun 2013, Telah Menyelesaikan Pendidikan di Sekolah Madrasah Tsanawiyah Negeri Lamala, Kabupaten Banggai, Provinsi Sulawesi Tengah
3. Peneliti Tahun 2016, Telah Menyelesaikan Pendidikan di Sekolah Madrasah Aliyah Al-Ittihaad Padangon, Kecamatan Masama, Kabupaten Banggai, Provinsi Sulawesi Tengah
4. Peneliti Tahun 2016, Telah di terima Menjadi Mahasiswa Di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo