

**KLASIFIKASI INCOMPLETE DATA PENYAKIT LIVER PADA  
MANUSIA DENGAN MENGGUNAKAN ALGORITMA  
VOTING FEATURE INTERVAL-5 (VFI5)**

**SKRIPSI**

**KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI**

**Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer**



**Disusun Oleh :**

**REVIANGGA DIKA SATYATAMA  
NIM. 0910683077**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS BRAWIJAYA  
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER  
PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER  
MALANG  
2013**

## LEMBAR PERSETUJUAN

### KLASIFIKASI *INCOMPLETE DATA* PENYAKIT LIVER PADA MANUSIA DENGAN MENGGUNAKAN ALGORITMA VOTING *FEATURE INTERVAL-5 (VF15)*

#### SKRIPSI

Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Komputer



Disusun oleh:

**REVIANGGA DIKA SATYATAMA**

**NIM. 0910683077**

Telah diperiksa dan disetujui untuk diuji dengan dosen pembimbing :

1. Indriati, ST., M.Kom
2. Budi Dharma Setiawan, S.Kom., M.Cs

Malang, 5 Juli 2013

Pembimbing I

Pembimbing II,

**Indriati, ST., M.Kom**  
**NIK. 831013 06 1 2 0035**

**Budi Dharma Setiawan, S.Kom., M.Cs**  
**NIP. 841015 06 1 1 0090**

**LEMBAR PENGESAHAN**

**KLASIFIKASI *INCOMPLETE DATA* PENYAKIT LIVER PADA  
MANUSIA DENGAN MENGGUNAKAN ALGORITMA VOTING  
*FEATURE INTERVAL-5 (VFI5)***

**SKRIPSI  
KONSENTRASI KOMPUTER CERDAS DAN VISUALISASI**

Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
**REVIANGGA DIKA SATYATAMA**  
**NIM. 0910683077**

Skripsi ini diuji dan dinyatakan lulus pada  
tanggal 5 Juli 2013

Penguji I

Penguji II

**Dian Eka Ratnawati, S.Si., M.Kom**  
**NIP. 19730619 200212 2 001**

**Rekyan Regasari MP, ST., MT.**  
**NIK. 770414 06 1 2 0253**

Penguji III

**Imam Cholissodin, S.Si., M.Kom**  
**NIK. 850719 16 1 1 0422**

Mengetahui,  
Ketua Program Studi Informatika/Ilmu Komputer

**Drs. Marji., M.T.**  
**NIP. 19670801 199203 1 001**

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena hanya dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan proposal skripsi dengan judul *“Klasifikasi Incomplete Data Penyakit Liver Pada Manusia Dengan Menggunakan Algoritma Voting Feature Interval-5 (VIF5)”*.

Dalam pelaksanaan dan penulisan skripsi ini penulis mendapatkan banyak bantuan dari berbagai pihak baik secara moril dan materiil. Dalam kesempatan ini penulis ingin mengucapkan terimakasih sebesar - besarnya kepada:

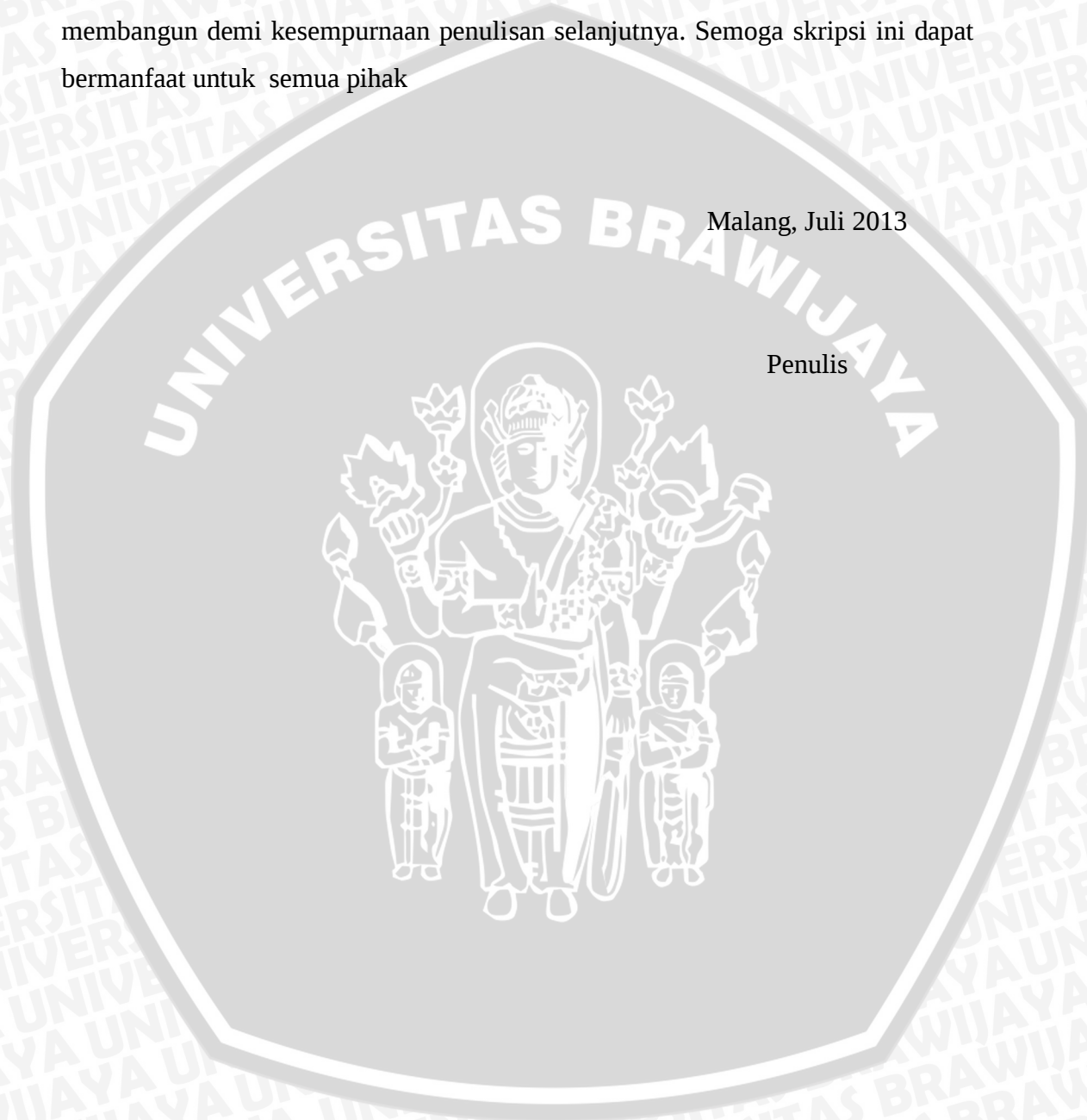
1. Indriati, ST., M.Kom. selaku pembimbing utama yang telah meluangkan waktu memberikan pengarahan dan bimbingan kepada penulis.
2. Budi Dharma Setiawan, S.Kom, M.Cs. selaku pembimbing pendamping yang telah meluangkan waktu memberikan pengarahan dan bimbingan kepada penulis
3. Drs. Marji, MT., selaku Ketua Program Studi Teknik Informatika, Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
5. Segenap staf, karyawan dan civitas di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah banyak membantu penulis dalam pelaksanaan penyusunan skripsi ini.
6. Secara khusus penulis ingin mengucapkan terima kasih kepada kedua orang tua yang penulis cintai, yang telah memberikan dukungan baik materi maupun non-materi selama penulis menyelesaikan skripsi ini.
7. Leli Intan Andari, yang senantiasa menemani, mendengarkan keluhan, memberikan motivasi, doa dan semangat kepada penulis.
8. Seluruh mahasiswa Program Teknologi Informasi & Ilmu Komputer angkatan 2009, khususnya teman – teman TIF C yang telah membantu terealisasinya skripsi ini.

9. Semua pihak yang telah membantu terselesaikannya penyusunan skripsi ini yang tidak dapat disebutkan satu per satu.

Penulis sadar bahwa skripsi ini masih jauh dari kesempurnaan, untuk itu dengan segala kerendahan hati penulis mengharapkan saran dan kritik yang membangun demi kesempurnaan penulisan selanjutnya. Semoga skripsi ini dapat bermanfaat untuk semua pihak

Malang, Juli 2013

Penulis



## ABSTRAK

Reviangga Dika Satyatama. 2013. : Klasifikasi *Incomplete Data* Penyakit Liver Pada Manusia Dengan Menggunakan Algoritma *Voting Feature Interval-5* (VFI5). Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing: Indriati, ST., M.Kom dan Budi Darma Setiawan , ST., M.Cs

Penyakit liver di dunia umumnya masih tergolong tinggi. Masalah yang ditimbulkan oleh penyakit liver adalah susahnya mengenali penyakit liver sejak dini, bahkan ketika penyakit tersebut sudah menyebar. Metode klasifikasi dengan Algoritma *Voting Feature Interval-5* (VFI-5) dapat mendiagnosa penyakit tersebut. Pengklasifikasian *instance* baru berdasarkan *voting* pada klasifikasi yang dibuat oleh nilai tiap-tiap *feature* secara terpisah. Selain masalah tersebut, dalam dunia medis sering ditemui kasus data tidak lengkap (*incomplete data*) yang dapat menyebabkan menurunnya tingkat akurasi dalam proses diagnosa. Metode yang digunakan untuk mengatasi data hilang pada nilai nominal adalah mengganti data hilang dengan modus sedangkan untuk nilai numerik adalah mengganti data hilang dengan *mean*. Tujuan dari penelitian ini adalah untuk membangun aplikasi klasifikasi data penyakit liver pada manusia agar dapat memudahkan pengguna serta meningkatkan akurasi hasil klasifikasi *incomplete data*. Aplikasi ini dapat digunakan sebagai sistem klasifikasi data penyakit liver serta dapat meningkatkan akurasi hasil klasifikasi *incomplete data* penyakit liver. Hal ini berdasarkan hasil pengujian akurasi, nilai rata-rata akurasi untuk perlakuan terhadap *incomplete data* dengan cara mengganti nilai data dengan *mean* atau modus mendapatkan nilai tertinggi sebesar 0,88 untuk *incomplete data* pelatihan dan sebesar 0,87 untuk *incomplete data* uji.

**Kata Kunci :** Klasifikasi, *Incomplete Data*, Algoritma *Voting Feature Interval-5*

## ABSTRACT

Reviangga Dika Satyatama. 2013. : *Klasifikasi Incomplete Data Penyakit Liver Pada Manusia Dengan Menggunakan Algoritma Voting Feature Interval-5 (VFI5)*. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing: Indriati, ST., M.Kom dan Budi Darma Setiawan , ST., M.Cs

*Liver disease in the world in general is still relatively high. Problems caused by liver disease is difficult to recognize early on, even when the disease has spread. Classification method with Algorithm Voting Feature Interval-5 (VFI-5) can diagnose the disease. The classification of a new instance based on voting on the classification made by the value of each feature separately. In addition to these problems, in the medical world often encountered cases of incomplete data which can lead to reduced levels of accuracy in the diagnosis process. The method used to overcome the missing data at nominal value is to replace the missing data by modus while the numerical value is to replace missing data with the mean. The purpose of this research is to build a data classification application of liver disease in humans in order to facilitate users and improve the accuracy of classification results of incomplete data. This application can be used as a data classification system of liver disease and can improve the accuracy of classification results incomplete liver disease data. It is based on the accuracy of test results, the average accuracy for the treatment of incomplete data by replacing the data with a mean value or mode get the highest score by 0.88 for the incomplete of training data and 0.87 for the incomplete of test data.*

**Keywords:** *Classification, Incomplete Data, Algorithms Voting Feature Interval-5*

## DAFTAR ISI

|                                              |      |
|----------------------------------------------|------|
| KATA PENGANTAR .....                         | i    |
| ABSTRAK .....                                | vi   |
| ABSTRACT .....                               | vii  |
| DAFTAR ISI .....                             | viii |
| DAFTAR GAMBAR .....                          | xi   |
| DAFTAR TABEL .....                           | xiv  |
| BAB I PENDAHULUAN .....                      | 1    |
| 1.1 Latar Belakang .....                     | 1    |
| 1.2 Rumusan Masalah .....                    | 3    |
| 1.3 Batasan Masalah .....                    | 3    |
| 1.4 Tujuan .....                             | 3    |
| 1.5 Manfaat .....                            | 4    |
| 1.6 Sistematika Penulisan .....              | 5    |
| BAB II TINJAUAN PUSTAKA .....                | 6    |
| 2.1 Kajian Pustaka .....                     | 6    |
| 2.2 Liver .....                              | 7    |
| 2.2.1 Penyebab Penyakit Liver .....          | 7    |
| 2.2.2 Gejala Penyakit Liver .....            | 10   |
| 2.2.3 Diagnosis Penyakit Liver .....         | 11   |
| 2.3 <i>Data Mining</i> .....                 | 13   |
| 2.3.1 Tahap-Tahap <i>Data Mining</i> .....   | 16   |
| 2.3.2 Pengelompokan <i>Data Mining</i> ..... | 21   |

|                                                    |                                               |    |
|----------------------------------------------------|-----------------------------------------------|----|
| 2.4                                                | Klasifikasi.....                              | 24 |
| 2.5                                                | <i>Voting Feature Interval-5 (VFI5)</i> ..... | 25 |
| 2.5.1                                              | Pelatihan.....                                | 26 |
| 2.5.2                                              | Prediksi.....                                 | 28 |
| 2.6                                                | <i>Incomplete Data</i> .....                  | 28 |
| 2.7                                                | Evaluasi .....                                | 29 |
| 2.7.1                                              | Akurasi .....                                 | 29 |
| BAB III METODOLOGI PENELITIAN DAN PERANCANGAN..... |                                               | 30 |
| 3.1                                                | Metodologi Penelitian .....                   | 30 |
| 3.1.1                                              | Studi Literatur .....                         | 31 |
| 3.1.2                                              | Penyusunan Dasar Teori .....                  | 31 |
| 3.1.3                                              | Analisa dan Perancangan .....                 | 32 |
| 3.1.4                                              | Implementasi.....                             | 35 |
| 3.1.5                                              | Pengujian.....                                | 35 |
| 3.1.6                                              | Penulisan Laporan.....                        | 38 |
| 3.2                                                | Perancangan.....                              | 39 |
| 3.2.1                                              | Analisis Kebutuhan .....                      | 39 |
| 3.2.2                                              | Perancangan Perangkat Lunak .....             | 42 |
| 3.2.3                                              | Perhitungan Manual .....                      | 62 |
| BAB IV IMPLEMENTASI.....                           |                                               | 73 |
| 4.1                                                | Spesifikasi Sistem.....                       | 74 |
| 4.1.1                                              | Spesifikasi Perangkat Keras .....             | 74 |
| 4.1.2                                              | Spesifikasi Perangkat Lunak .....             | 74 |
| 4.2                                                | Batasan-Batasan Implementasi.....             | 75 |
| 4.3                                                | Implementasi Basis Data .....                 | 75 |

|                                   |                                                                         |     |
|-----------------------------------|-------------------------------------------------------------------------|-----|
| 4.4                               | Implementasi Algoritma.....                                             | 76  |
| 4.4.1                             | Implementasi Algoritma Memasukkan Data Pelatihan.....                   | 76  |
| 4.4.2                             | Implementasi Algoritma <i>Preprocessing</i> Pelatihan Klasifikasi ..... | 78  |
| 4.4.3                             | Implementasi Algoritma Pelatihan Klasifikasi .....                      | 79  |
| 4.4.4                             | Implementasi Algoritma Memasukkan Data Uji .....                        | 82  |
| 4.4.5                             | Implementasi Algoritma <i>Preprocessing</i> Prediksi Klasifikasi .....  | 84  |
| 4.4.6                             | Implementasi Algoritma Prediksi Klasifikasi .....                       | 86  |
| 4.5                               | Implementasi Antar Muka Aplikasi .....                                  | 88  |
| 4.5.1                             | Halaman Utama.....                                                      | 89  |
| 4.5.2                             | Halaman Memasukkan Data Pelatihan .....                                 | 89  |
| 4.5.3                             | Halaman Data Pelatihan Klasifikasi.....                                 | 90  |
| 4.5.4                             | Halaman Memasukkan Data Uji.....                                        | 91  |
| 4.5.5                             | Halaman Hasil Klasifikasi.....                                          | 92  |
| BAB V PENGUJIAN DAN ANALISIS..... |                                                                         | 94  |
| 5.1                               | Pengujian.....                                                          | 94  |
| 5.1.1                             | Pengujian Akurasi .....                                                 | 94  |
| 5.2                               | Analisis.....                                                           | 121 |
| 5.2.1                             | Analisis Hasil Pengujian Akurasi.....                                   | 121 |
| BAB VI PENUTUP .....              |                                                                         | 124 |
| 6.1                               | Kesimpulan.....                                                         | 124 |
| 6.2                               | Saran.....                                                              | 124 |
| DAFTAR PUSTAKA .....              |                                                                         | 126 |

## DAFTAR GAMBAR

|                                                                                          |    |
|------------------------------------------------------------------------------------------|----|
| Gambar 2.1 Bidang Ilmu <i>Data Mining</i> .....                                          | 16 |
| Gambar 2.2 Tahap-Tahap <i>Data Mining</i> .....                                          | 17 |
| Gambar 2.3 Proses <i>Knowledge Discovery In Database</i> (KDD) .....                     | 21 |
| Gambar 2.4 Proses Pembentukan <i>Interval</i> Pada Data Nominal.....                     | 25 |
| Gambar 2.5 Proses Pembentukan <i>Interval</i> Pada Data Linier .....                     | 26 |
| Gambar 3.1 Desain Penelitian.....                                                        | 30 |
| Gambar 3.2 Diagram Blok Klasifikasi.....                                                 | 33 |
| Gambar 3.3 Diagram Blok Perancangan.....                                                 | 39 |
| Gambar 3.4 Diagram Alir Sistem Pelatihan Klasifikasi .....                               | 43 |
| Gambar 3.5 Perancangan Algoritma Memasukkan Data Pelatihan .....                         | 44 |
| Gambar 3.6 Diagram Alir Proses <i>Preprocessing</i> .....                                | 45 |
| Gambar 3.7 Perancangan Algoritma Proses <i>Preprocessing</i> Pelatihan Klasifikasi       | 46 |
| Gambar 3.8 Diagram Alir Proses Pelatihan Klasifikasi.....                                | 48 |
| Gambar 3.9 Perancangan Algoritma Proses Pelatihan Klasifikasi .....                      | 50 |
| Gambar 3.10 Diagram Alir Sistem Prediksi Klasifikasi .....                               | 50 |
| Gambar 3.11 Perancangan Algoritma Memasukkan Data Uji.....                               | 52 |
| Gambar 3.12 Perancangan Algoritma Proses <i>Preprocessing</i> Prediksi Klasifikasi ..... | 53 |
| Gambar 3.13 Diagram Alir Proses Prediksi Klasifikasi.....                                | 54 |
| Gambar 3.14 Perancangan Algoritma Proses Prediksi Klasifikasi .....                      | 55 |
| Gambar 3.15 Perancangan Basis Data .....                                                 | 57 |
| Gambar 3.16 <i>Site Map</i> Sistem.....                                                  | 58 |
| Gambar 3.17 Perancangan Antarmuka Halaman Utama .....                                    | 58 |
| Gambar 3.18 Perancangan Antarmuka Halaman Memasukkan Data Pelatihan...                   | 59 |
| Gambar 3.19 Perancangan Antarmuka Halaman Data Pelatihan Klasifikasi .....               | 60 |
| Gambar 3.20 Perancangan Antarmuka Halaman Memasukkan Data Uji.....                       | 61 |
| Gambar 3.21 Perancangan Antarmuka Halaman Hasil Klasifikasi .....                        | 62 |

|                                                                                    |     |
|------------------------------------------------------------------------------------|-----|
| Gambar 4.1 Pohon Implementasi.....                                                 | 73  |
| Gambar 4.2 Implementasi Basis Data.....                                            | 76  |
| Gambar 4.3 Implementasi Algoritma Memasukkan Data Pelatihan.....                   | 77  |
| Gambar 4.4 Implementasi Algoritma <i>Preprocessing</i> Pelatihan Klasifikasi ..... | 79  |
| Gambar 4.5 Implementasi Algoritma Pelatihan Klasifikasi .....                      | 81  |
| Gambar 4.6 Implementasi Algoritma Memasukkan Data Uji .....                        | 84  |
| Gambar 4.7 Implementasi Algoritma <i>Preprocessing</i> Prediksi Klasifikasi .....  | 85  |
| Gambar 4.8 Implementasi Algoritma Prediksi Klasifikasi .....                       | 88  |
| Gambar 4.9 Tampilan Antar Muka Halaman Utama .....                                 | 89  |
| Gambar 4.10 Tampilan Antar Muka Halaman Memasukkan Data Pelatihan.....             | 90  |
| Gambar 4.11 Tampilan Antar Muka Halaman Data Pelatihan Klasifikasi.....            | 91  |
| Gambar 4.12 Tampilan Antar Muka Halaman Memasukkan Data Uji .....                  | 92  |
| Gambar 4.13 Tampilan Antar Muka Halaman Hasil Klasifikasi .....                    | 93  |
| Gambar 5.1 Pohon Pengujian Dan Analisis .....                                      | 94  |
| Gambar 5.2 Diagram Alir Proses Pengujian Akurasi .....                             | 95  |
| Gambar 5.3 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 70% ..... | 97  |
| Gambar 5.4 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 50% ..... | 98  |
| Gambar 5.5 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 25% ..... | 99  |
| Gambar 5.6 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 20% ....  | 100 |
| Gambar 5.7 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 15% ....  | 102 |
| Gambar 5.8 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 10% ....  | 103 |
| Gambar 5.9 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 5% .....  | 105 |
| Gambar 5.10 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 2% ....  | 106 |
| Gambar 5.11 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 70% .....      | 107 |
| Gambar 5.12 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 50% .....      | 109 |
| Gambar 5.13 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 25% .....      | 110 |
| Gambar 5.14 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 20% .....      | 111 |
| Gambar 5.15 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 15% .....      | 112 |
| Gambar 5.16 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 10% .....      | 114 |
| Gambar 5.17 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 5% .....       | 115 |
| Gambar 5.18 Nilai Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 2% .....       | 116 |

|                                                                                                            |     |
|------------------------------------------------------------------------------------------------------------|-----|
| Gambar 5.19 Grafik Nilai Rata-Rata Akurasi Setiap Perlakuan Terhadap <i>Incomplete Data</i> Pelatihan..... | 118 |
| Gambar 5.20 Grafik Nilai Rata-Rata Akurasi Setiap Perlakuan Terhadap <i>Incomplete Data</i> Uji.....       | 120 |



## DAFTAR TABEL

|                                                                                                     |     |
|-----------------------------------------------------------------------------------------------------|-----|
| Tabel 3.1 Contoh Tabel Pengujian.....                                                               | 37  |
| Tabel 3.2 Identifikasi Aktor .....                                                                  | 40  |
| Tabel 3.3 Daftar Kebutuhan Fungsional .....                                                         | 41  |
| Tabel 3.4 Daftar Kebutuhan Non-fungsional.....                                                      | 42  |
| Tabel 3.5 Ilustrasi Data Pelatihan .....                                                            | 63  |
| Tabel 3.6 Ilustrasi Hasil <i>Preprocessing</i> Klasifikasi.....                                     | 64  |
| Tabel 3.7 Ilustrasi Hasil Pelatihan Klasifikasi .....                                               | 67  |
| Tabel 4.1 Spesifikasi Perangkat Keras Komputer.....                                                 | 74  |
| Tabel 4.2 Spesifikasi Perangkat Lunak Komputer.....                                                 | 74  |
| Tabel 5.1 Pengujian Akurasi Dengan Data Yang Lengkap .....                                          | 96  |
| Tabel 5.2 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 70%..                   | 96  |
| Tabel 5.3 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 50%..                   | 97  |
| Tabel 5.4 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 25%..                   | 99  |
| Tabel 5.5 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 20%                     | 100 |
| Tabel 5.6 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 15%                     | 101 |
| Tabel 5.7 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 10%                     | 103 |
| Tabel 5.8 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 5%..                    | 104 |
| Tabel 5.9 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Pelatihan 2%..                    | 106 |
| Tabel 5.10 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 70%.....                     | 107 |
| Tabel 5.11 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 50%.....                     | 108 |
| Tabel 5.12 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 25%.....                     | 109 |
| Tabel 5.13 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 20%.....                     | 111 |
| Tabel 5.14 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 15%.....                     | 112 |
| Tabel 5.15 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 10%.....                     | 113 |
| Tabel 5.16 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 5%.....                      | 115 |
| Tabel 5.17 Pengujian Akurasi Dengan Tingkat <i>Incomplete Data</i> Uji 2%.....                      | 116 |
| Tabel 5.18 Nilai Rata-Rata Akurasi Menurut Perlakuan Terhadap <i>Incomplete Data</i> Pelatihan..... | 117 |

|                                                                                                |     |
|------------------------------------------------------------------------------------------------|-----|
| Tabel 5.19 Nilai Rata-Rata Akurasi Menurut Perlakuan Terhadap <i>Incomplete Data Uji</i> ..... | 119 |
|------------------------------------------------------------------------------------------------|-----|

|                                                    |     |
|----------------------------------------------------|-----|
| Tabel 5.20 Jumlah Data Pelatihan Klasifikasi ..... | 122 |
|----------------------------------------------------|-----|



## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Penyakit liver di dunia umumnya masih tergolong tinggi. Penyakit liver menempati urutan keempat belas dengan angka kematian mencapai 848.636 per tahun, dengan prosentase 0,012% [WLE-10]. Sedangkan di Indonesia, penyakit liver merupakan penyebab kematian keempat belas dengan angka kematian mencapai 23.024 pertahun, dengan prosentase 0,01% [WLE-10]. Pada umumnya penyakit liver lebih dikenal sebagai penyakit yang berhubungan dengan organ liver. Penyakit liver dengan komplikasinya merupakan masalah kesehatan yang masih sulit diatasi di Indonesia. Secara umum diperkirakan angka insiden penyakit liver di rumah sakit seluruh Indonesia berkisar antara 0,6-14,5% [MAL-12].

Masalah yang ditimbulkan oleh penyakit liver adalah susahnya mengenali penyakit liver sejak dini, bahkan ketika penyakit tersebut sudah menyebar. Diagnosa penyakit liver yang lebih awal dapat meningkatkan tingkat kelangsungan hidup pasien. [LIN-09]. Dengan perkembangan teknologi saat ini diagnosa penyakit dapat menggunakan metode *data mining* [NOV-12]. Salah satu pengembangan dari *data mining* adalah klasifikasi [ROM-09]. Klasifikasi merupakan sebuah model yang terbentuk untuk memprediksi suatu kategori [JUN-09]. Salah satu contoh dari metode klasifikasi adalah Algoritma *Voting Feature Interval-5 (VFI-5)*.

Algoritma klasifikasi VFI5 merupakan suatu algoritma yang merepresentasikan deskripsi sebuah konsep oleh sekumpulan *interval* nilai-nilai *feature* atau atribut. Pengklasifikasian *instance* baru berdasarkan *voting* pada klasifikasi yang dibuat oleh nilai tiap-tiap *feature* secara terpisah. VFI 5 merupakan algoritma klasifikasi yang bersifat *non-incremental* di mana semua

*instance* pelatihan diproses secara bersamaan [GUV-00]. Keunggulan algoritma VFI 5 adalah algoritma ini cukup kokoh (*robust*) terhadap *feature* yang tidak relevan dengan mekanisme *voting*-nya [GUV-98]. Waktu rata-rata yang digunakan algoritma VFI untuk melakukan klasifikasi adalah 12,68 *msec* [DEM-97]. Waktu rata-rata yang digunakan algoritma *Nearest Neighbor* untuk melakukan klasifikasi adalah 4.664,65 *msec* [DEM-97]. Waktu rata-rata yang digunakan algoritma *Naive Bayes* untuk melakukan klasifikasi adalah 64,18 *msec* [DEM-97]. Algoritma *Decision Tree* lebih lambat daripada algoritma *Naive Bayes* [KON-93]. Algoritma klasifikasi VFI5 mampu melakukan klasifikasi lebih cepat dibandingkan dengan algoritma *Nearest Neighbor* dan *Decision Tree* [DEM-97]. Selain masalah tersebut, dalam dunia medis sering ditemui kasus data tidak lengkap (*incomplete data*) yang dapat menyebabkan menurunnya tingkat akurasi dalam proses diagnosa.

Dalam penanganan masalah *incomplete data* dapat diatasi dengan beberapa metode. Cara yang paling mudah untuk mengatasi data tidak lengkap adalah dengan menghapus satu baris data yang tidak lengkap. Teknik ini terkadang menyebabkan hilangnya informasi yang potensial. Pendekatan yang kedua adalah dengan mengganti semua data hilang dengan rataannya [ENN-01]. Salah satu teknik untuk mengatasi data hilang pada nilai nominal adalah mengganti data hilang dengan modus sedangkan untuk nilai numerik adalah mengganti data hilang dengan *mean* [MEI-05].

Berdasarkan penelitian sebelumnya yang berjudul '*Diagnosis Gangguan Sistem Urinari Pada Anjing Dan Kucing Menggunakan VFI 5*', penelitian ini membahas tentang klasifikasi data pasien penyakit liver pada manusia. Pada penelitian sebelumnya diperoleh hasil akurasi sebesar 86,31% [BUO-09]. Metode yang digunakan pada penelitian ini adalah Algoritma *Voting Features Interval-5* (VFI5) sebagai metode klasifikasi. Dalam proses *preprocessing* ditambahkan sistem untuk mengganti nilai *incomplete data* yang belum ada dari penelitian sebelumnya. Dengan sistem ini diharapkan dapat memperoleh hasil klasifikasi

yang lebih cepat dan lebih akurat untuk klasifikasi *incomplete data* penyakit liver pada manusia.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan permasalahan pada skripsi ini yaitu sebagai berikut:

1. Membangun suatu aplikasi dengan implementasi algoritma *Voting Feature Interval-5* (VFI5) sebagai sistem klasifikasi yang tepat bagi data penyakit liver.
2. Merancang suatu mekanisme pengganti data tidak lengkap (*incomplete data*) untuk meningkatkan akurasi klasifikasi penyakit data liver.

## 1.3 Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka pada penelitian ini dibatasi oleh hal-hal sebagai berikut:

1. Metode yang digunakan adalah klasifikasi dengan *Voting Feature Interval-5* (VIF5) sebagai penentuan bobot parameter penyakit liver.
2. Data yang digunakan pada penelitian ini adalah *Indian Liver Patient Dataset* yang diperoleh dari url: [http://archive.ics.uci.edu/ml/datasets/ILPD+\(Indian+Liver+Patient+Datase+t\)](http://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Datase+t))
3. Aplikasi hanya akan memproses data dalam bentuk format xls dan sql.
4. Pembahasan diterapkan pada aplikasi berbasis *web*

## 1.4 Tujuan

Tujuan yang ingin dicapai dalam pembuatan skripsi ini adalah:

1. Membangun aplikasi klasifikasi data penyakit liver pada manusia agar dapat memudahkan pengguna untuk mendapatkan hasil klasifikasi dari parameter penyakit tersebut.
2. Menerapkan metode *Voting Feature Interval-5 (VIF5)* untuk diterapkan dalam pengklasifikasian data penyakit liver.
3. Membangun sistem pengganti data tidak lengkap (*incomplete data*) untuk meningkatkan akurasi hasil klasifikasi dengan cara mengganti nilai data tidak lengkap dengan *mean* atau modus.

### 1.5 Manfaat

Penulisan skripsi ini diharapkan mempunyai manfaat yang baik dan berguna bagi pembaca dan penulis. Adapun manfaat yang diharapkan adalah sebagai berikut:

- Bagi Penulis

Sebagai sarana untuk memperluas wawasan ilmu yang sudah didapat di masa perkuliahan, khususnya *data mining*.

- Bagi Pengembang

Aplikasi ini diharapkan bisa menjadi acuan dalam melakukan pengembangan aplikasi lain yang masih berkaitan dalam topik yang digunakan pada aplikasi ini. Sehingga, akan terciptanya aplikasi yang lebih kompleks seiring berkembangnya informasi dan kebutuhan dari pengguna.

- Bagi Pengguna

Sebagai alat pembantu dalam proses diagnosa penyakit liver dari pasien meskipun data yang tersedia tidak lengkap (*incomplete data*), sehingga nantinya dapat memberikan penanganan medis yang tepat bagi pasien.

## 1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

### **BAB I      Pendahuluan**

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, metodologi pembahasan, dan sistematika penulisan.

### **BAB II     Kajian Pustaka dan Dasar teori**

Menguraikan tentang dasar teori dan referensi yang mendasari pembuatan klasifikasi data pada penyakit liver yaitu kajian pustaka, liver, *data mining*, klasifikasi, algoritma *Voting Feature Interval-5*, *incomplete data*, dan evaluasi.

### **BAB III    Metode Penelitian dan Perancangan**

Membahas metode yang digunakan dalam penelitian yang terdiri dari studi literatur, perancangan sistem perangkat lunak, implementasi sistem perangkat lunak, pengujian dan analisis, penulisan laporan serta analisis kebutuhan dan perancangan Klasifikasi *Incomplete data* Penyakit Liver Pada Manusia Dengan Menggunakan Algoritma *Voting Feature Interval-5* (VIF5).

### **BAB IV    Implementasi**

Membahas implementasi dari klasifikasi *incomplete data* penyakit liver yang sesuai dengan perancangan sistem yang telah dibuat.

### **BAB V     Pengujian dan Analisis**

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

### **BAB VI    Penutup**

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran – saran untuk pengembangan lebih lanjut.

## BAB II

### TINJAUAN PUSTAKA

Bab ini berisi tinjauan pustaka yang meliputi kajian pustaka dan dasar teori yang diperlukan untuk penelitian. Kajian pustaka adalah membahas penelitian yang telah ada dan yang diusulkan. Dasar teori adalah membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan.

Kajian pustaka pada penelitian ini adalah membahas penelitian sebelumnya yang berjudul '*Diagnosis Gangguan Sistem Urinari Pada Anjing Dan Kucing Menggunakan VFI 5*'. Teori dasar yang akan dibahas pada bab ini yaitu konsep dasar *Data mining*, Klasifikasi, Algoritma *Voting Feature Interval-5 (VFI-5)*, *Incomplete data* dan Akurasi.

#### 2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini adalah membahas penelitian sebelumnya yang berjudul '*Diagnosis Gangguan Sistem Urinari Pada Anjing Dan Kucing Menggunakan VFI 5*'. Penelitian sebelumnya melakukan pengujian klasifikasi dengan *feature* pemeriksaan laboratorium. Penelitian ini juga melakukan pengujian klasifikasi tanpa *feature* pemeriksaan laboratorium. Hasil yang diperoleh dari penelitian ini adalah tingkat akurasi yang diperoleh pada proses klasifikasi tanpa *feature* pemeriksaan laboratorium lebih kecil daripada tingkat akurasi dari proses klasifikasi dengan *feature* pemeriksaan laboratorium. Tingkat rata-rata akurasi pada proses klasifikasi tanpa *feature* pemeriksaan laboratorium yaitu sebesar 77,38%, sedangkan pada proses klasifikasi dengan *feature* pemeriksaan laboratorium yaitu sebesar 86,31% [BUO-09].

Penelitian ini membahas tentang klasifikasi data pasien penyakit liver pada manusia. Metode yang digunakan pada penelitian ini adalah Algoritma *Voting Features Interval-5 (VFI-5)*. Perbedaan yang dibuat penulis pada penelitian ini adalah pada penggunaan algoritma *Voting Feature Interval-5* di khususnya untuk

data pasien penyakit liver pada manusia. Selain itu perbedaan juga terdapat pada proses *preprocessing*, yaitu proses melengkapi *incomplete data*. Hal ini mengakibatkan setiap data yang tidak lengkap (*incomplete data*) akan diganti nilainya sesuai dengan jenis nilai datanya, sehingga dapat meningkatkan akurasi dari aplikasi ini.

## 2.2 Liver

Liver atau hati adalah organ terbesar dalam tubuh padat dan juga dianggap kelenjar. Liver ini terletak di bagian kanan atas perut dilindungi oleh tulang rusuk. Penyakit liver adalah setiap gangguan fungsi liver yang menyebabkan penyakit. Liver bertanggung jawab untuk fungsi-fungsi kritis dalam tubuh, hilangnya fungsi-fungsi dapat menyebabkan kerusakan yang signifikan pada tubuh. Liver adalah satu-satunya organ dalam tubuh yang dapat dengan mudah mengganti sel yang rusak, tetapi jika sel-sel cukup hilang, liver tidak mungkin dapat memenuhi kebutuhan tubuh. Kerusakan pada liver dapat terjadi dengan berbagai cara, antara lain [SAL-13]:

1. Terjadi peradangan pada sel
2. Aliran empedu terhambat
3. Kolesterol atau *triglycerida* menumpuk
4. Jaringan liver rusak oleh bahan kimia dan mineral, atau disusupi oleh sel-sel abnormal

### 2.2.1 Penyebab Penyakit Liver

Penyakit liver dapat disebabkan oleh beberapa hal, antara lain [SAL-13]:

1. Penyalahgunaan alkohol

Alkohol, secara langsung merupakan racun bagi sel hati dan dapat menyebabkan peradangan hati yang biasa disebut sebagai hepatitis alkoholik [SAL-13].

## 2. Sirosis

Sirosis adalah tahap akhir dari penyakit hati. Jaringan parut pada hati dan hilangnya fungsi sel hati dapat menyebabkan penyakit liver [SAL-13].

## 3. Jamur beracun

Mengonsumsi jamur tak dikenal di alam liar dapat menyebabkan timbulnya penyakit liver karena jamur yang dikonsumsi merupakan jamur beracun [SAL-13].

## 4. Infeksi hepatitis

- Hepatitis A adalah infeksi virus yang menyebar terutama melalui rute *fecal-oral* ketika sejumlah kecil dari kotoran yang terinfeksi secara tidak sengaja tertelan.
- Hepatitis B ditularkan oleh paparan cairan tubuh (jarum dari pecandu obat, darah yang terkontaminasi, dan kontak seksual) dan dapat menyebabkan infeksi akut, tetapi juga dapat berkembang menjadi kronis dan dapat menyebabkan peradangan (hepatitis kronis) yang menyebabkan sirosis dan kanker liver.
- Hepatitis C menyebabkan hepatitis kronis.
- Hepatitis D adalah virus yang membutuhkan infeksi bersamaan dengan hepatitis B untuk bertahan hidup, dan menyebar melalui paparan cairan tubuh (jarum dari pecandu obat, darah yang terkontaminasi, dan kontak seksual).
- Hepatitis E adalah virus yang menyebar melalui paparan makanan dan air yang terkontaminasi [SAL-13].

#### 5. Virus Lain

Virus lain juga dapat menyebabkan peradangan liver atau hepatitis sebagai bagian dari *cluster* gejala. Infeksi virus dengan infeksi *mononucleosis* (virus *Epstein Barr*), *adenovirus*, dan virus *sitomegalo* dapat menyebabkan liver. Non-virus infeksi seperti toksoplasmosis dan *Rocky Mountain spotted fever* adalah penyebab kurang umum dari penyakit liver [SAL-13].

#### 6. Akumulasi Lemak

NASH atau *non-alkohol steatohepatitis* (juga disebut sebagai "*fatty liver*") menggambarkan akumulasi lemak dalam liver yang dapat menyebabkan peradangan liver dan penurunan bertahap dalam fungsi liver [SAL-13].

#### 7. Hemochromatosis

*Hemachromatosis* (kelebihan zat besi) adalah gangguan metabolisme yang mengarah ke kadar besi abnormal dalam tubuh. Kelebihan zat besi dapat terakumulasi dalam jaringan dari pankreas jantung, liver, dan dapat menyebabkan peradangan, sirosis, kanker liver, dan gagal liver. *Hemachromatosis* adalah penyakit yang diwariskan [SAL-13].

#### 8. Penyakit Wilson

Penyakit Wilson merupakan penyakit bawaan yang mempengaruhi kemampuan tubuh untuk memetabolisme tembaga. Penyakit Wilson dapat menyebabkan sirosis dan gagal liver [SAL-13].

#### 9. Penyakit Gilbert

Penyakit Gilbert, adalah kelainan dalam metabolisme *bilirubin* dalam liver [SAL-13].

#### 10. Kanker

Kanker primer liver timbul dari struktur liver dan sel. Contohnya adalah *karsinoma hepatoseluler* dan *cholangiocarcinoma* [SAL-13].

#### 11. Kelainan Aliran Darah

*Budd Chiari syndrome* adalah penyakit dimana gumpalan darah terbentuk di vena hepatic dan mencegah darah meninggalkan liver [SAL-13].

#### 12. Sirosis bilier primer dan *primary sclerosing cholangitis*

Sirosis bilier primer dan *primary sclerosing cholangitis* dapat menyebabkan jaringan parut progresif dari saluran-saluran empedu, menyebabkan saluran empedu menjadi sempit. Sempitnya saluran empedu dapat menyebabkan terhambatnya aliran empedu yang melalui liver [SAL-13].

### 2.2.2 Gejala Penyakit Liver

Gejala penyakit liver secara klasik dapat dijelaskan sebagai berikut [SAL-13]:

1. Mual
2. Muntah
3. Nyeri pada kuadran kanan atas perut
4. *Jaundice* (perubahan warna kuning pada kulit karena konsentrasi *bilirubin* tinggi dalam aliran darah)
5. Timbulnya gejala kelelahan, lemah, dan penurunan berat badan yang mungkin terjadi.

### 2.2.3 Diagnosis Penyakit Liver

Untuk mendiagnosa penyakit liver dapat dilakukan dengan berbagai cara, antara lain [SAL-13]:

#### 1. Pemeriksaan Fisik

Penyakit liver dapat memiliki temuan fisik yang mempengaruhi hampir semua sistem tubuh termasuk jantung, paru-paru, perut, kulit, otak dan fungsi kognitif, dan bagian lain dari sistem saraf. Pemeriksaan fisik sering membutuhkan evaluasi seluruh tubuh.

#### 2. Tes darah

Tes darah sangat membantu dalam menilai peradangan liver dan fungsi, Tes darah untuk fungsi liver khusus meliputi:

- AST dan ALT (bahan kimia *transaminase* dirilis dengan peradangan sel liver);
- GGT dan alkaline fosfatase (bahan kimia yang dikeluarkan oleh sel yang melapisi saluran empedu);
- *Bilirubin*
- Protein dan kadar *albumin*.

#### 3. Pencitraan dapat digunakan untuk memvisualisasikan, tidak hanya liver, tetapi organ terdekat lain yang mungkin sakit. Contoh pencitraan meliputi:

- CT scan.
- MRI (*magnetic resonance imaging*)
- USG (gelombang pencitraan suara, yang sangat membantu dalam menilai kandung empedu dan saluran empedu).

Dalam penelitian ini terdapat beberapa parameter yang digunakan untuk mendiagnosa penyakit liver, antara lain:

### 1. *Total Bilirubin*

*Total bilirubin* merupakan jumlah pigmen empedu yang dihasilkan dari pemecahan *heme* dan reduksi *biliverdin*. *Bilirubin* secara normal bersirkulasi di dalam plasma sebagai suatu kompleks dengan *albumin*, diambil oleh sel-sel hati dan dikonjugasikan menjadi *bilirubin diglukuronid* yang merupakan pigmen larut dalam air yang diekskresikan ke dalam empedu. Konsentrasi *bilirubin* yang tinggi dapat menyebabkan ikterus [KKD-98].

### 2. *Direct Bilirubin*

*Direct bilirubin* merupakan *bilirubin* yang telah diambil oleh sel-sel hati dan dikonjugasikan membentuk *bilirubin diglukuronid* yang larut dalam air [KKD-98].

### 3. *Alkaline Phosphotase*

*Alkaline phosphotase* merupakan *enzim* yang dibuat dalam hati, tulang, dan plasenta dan biasanya hadir dalam konsentrasi tinggi pada darah yang tumbuh dan empedu dan dalam konsentrasi rendah pada darah. Fosfatase alkali dilepaskan ke dalam darah dalam jumlah yang meningkat selama kerusakan sel-sel hati dan selama aktivitas normal seperti pertumbuhan tulang dan kehamilan. Tingkat abnormal rendah fosfatase alkali hadir dalam kondisi genetik dan hipotiroidisme. Zat ini diukur dalam tes darah rutin [ALP-13].

### 4. *Alanine Aminotransferase*

*Alanine aminotransferase* merupakan *enzim* yang secara normal dijumpai pada serum dan jaringan tubuh, terutama pada hati. *Alanine aminotranferase* dilepaskan ke dalam serum sebagai akibat dari kerusakan jaringan, oleh karena itu konsentrasinya dalam serum dapat meningkat pada pasien dengan kerusakan sel hati yang akut [KKD-98].

### 5. *Aspartate Aminotransferase*

*Aspartate aminotransferase* merupakan *enzim* yang biasanya terdapat dalam jaringan tubuh, terutama pada jantung dan hati. *Enzim* itu dilepaskan ke dalam serum sebagai akibat dari cedera jaringan, oleh

karena itu konsentrasi dalam serum dapat meningkat pada penyakit seperti infark miokard atau kerusakan akut pada sel-sel hati [KKD-98].

#### 6. *Total Proteins*

*Total proteins* merupakan jumlah dari setiap kelompok senyawa organik kompleks yang mengandung karbon, hidrogen, oksigen, nitrogen, dan sulfur. Protein, konstituen utama protoplasma semua sel, mempunyai berat molekul yang tinggi dan terdiri atas gabungan asam  $\alpha$ -amino dengan ikatan peptida. Dua puluh macam asam amino umumnya ditemukan pada protein, masing-masing protein memiliki rangkaian asam amino yang unik, secara genetik ada batasannya masing-masing serta menentukan bentuk dan fungsi protein yang khas. Protein berfungsi sebagai enzim, unsur struktural, hormon, *imunoglobulin*, dan lain-lain serta terlibat dalam kegiatan transpor oksigen, konsentrasi otot, transpor elektron dan aktivitas lainnya [KKD-98].

#### 7. *Albumin*

*Albumin* merupakan protein yang larut dalam air dan juga dalam konsentrasi larutan garam yang sedang [KKD-98].

#### 8. *Albumin and Globulin Ratio*

*Albumin and globulin ratio* merupakan perbandingan *albumin* dengan *globulin*, yang merupakan konstituen utama protein yang ditemukan dalam darah. Rasio yang abnormal terlihat ketika kadar *albumin* atau *globulin* meningkat atau menurun. Rasio abnormal terlihat pada berbagai gangguan, termasuk penyakit ginjal dan hati [AGR-13].

### 2.3 *Data Mining*

*Data mining* adalah suatu istilah yang digunakan untuk menemukan pengetahuan yang tersembunyi di dalam *database*. *Data mining* merupakan proses semi otomatis yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi pengetahuan potensial dan berguna yang bermanfaat yang tersimpan di dalam *database* besar [TUR-05]. Menurut Gartner Group *data mining* adalah

suatu proses menemukan hubungan yang berarti, pola, dan kecenderungan dengan memeriksa dalam sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika [LAR-06].

Selain definisi di atas beberapa definisi juga diberikan seperti, “*data mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual.” [PRA-06]. “*Data mining* adalah analisis otomatis dari data yang berjumlah besar atau kompleks dengan tujuan untuk menemukan pola atau kecenderungan yang penting yang biasanya tidak disadari keberadaannya.” [PRA-06].

“*Data mining* merupakan analisis dari peninjauan kumpulan data untuk menemukan hubungan yang tidak diduga dan meringkas data dengan cara yang berbeda dengan sebelumnya, yang dapat dipahami dan bermanfaat bagi pemilik data.” [LAR-06]. “*Data mining* merupakan bidang dari beberapa keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, *database*, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari *database* yang besar.” [LAR-06]. Kemajuan luar biasa yang terus berlanjut dalam bidang *data mining* didorong oleh beberapa faktor, antara lain : [LAR-06]

1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam *data warehouse*, sehingga seluruh perusahaan memiliki akses ke dalam *database* yang baik.
3. Adanya peningkatan akses data melalui navigasi *web* dan intranet.
4. Tekanan kompetisi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi perangkat lunak untuk *data mining* (ketersediaan teknologi).

6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.

Berdasarkan definisi-definisi yang telah disampaikan, hal penting yang terkait dengan *data mining* adalah :

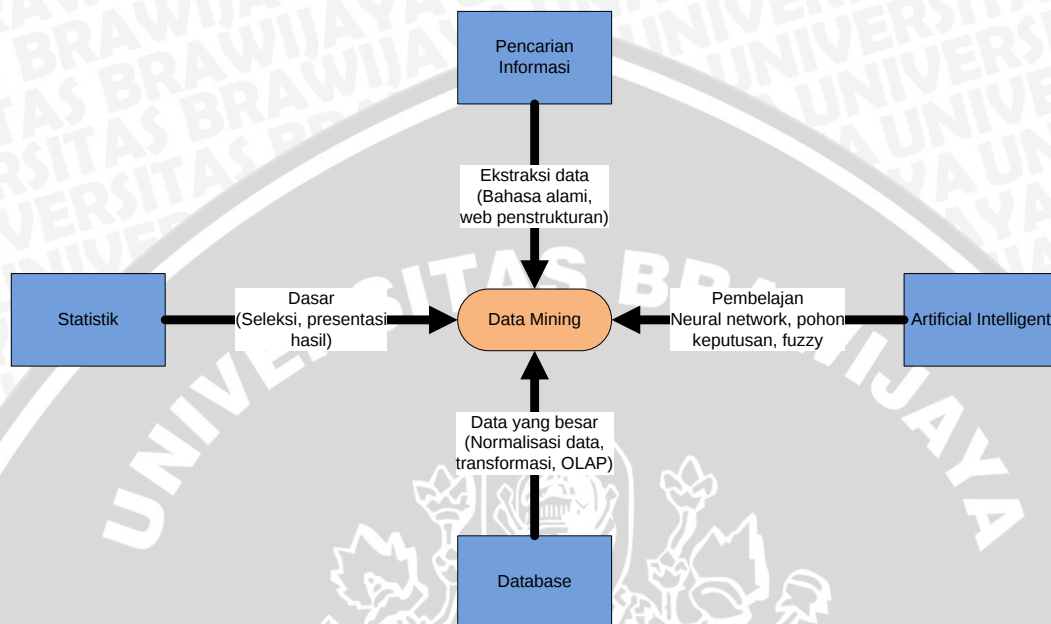
1. *Data mining* merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses berupa data yang sangat besar.
3. Tujuan *data mining* adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

Beberapa definisi awal dari *data mining* meyorakan fokus pada proses otomatisasi. Menurut Berry dan Linoff [BER-04], dalam buku *Data mining Technique for Marketing, Sales, and Customer Support* mendefinisikan *data mining* sebagai suatu proses eksplorasi dan analisis secara otomatis maupun semi otomatis terhadap data dalam jumlah besar dengan tujuan menemukan pola atau aturan yang berarti [LAR-06].

Tiga tahun kemudian, dalam buku *Mastering Data mining* mereka memberikan definisi ulang terhadap pengertian *data mining* dan memberikan pernyataan bahwa “jika ada yang kami sesalkan adalah frasa secara otomatis maupun semi otomatis, karena kami merasa hal tersebut memberikan fokus berlebih pada teknik otomatis dan kurang pada eksplorasi dan analisis”. Hal tersebut memberikan pemahaman yang salah bahwa *data mining* merupakan produk yang dapat dibeli dibandingkan keilmuan yang harus dikuasai [LAR-06].

Pernyataan tersebut menegaskan bahwa dalam *data mining* otomatisasi tidak menggantikan campur tangan manusia. Manusia harus ikut aktif dalam setiap fase dalam proses *data mining*. Kehebatan kemampuan algoritma *data mining* yang terdapat dalam perangkat lunak analisis yang terdapat saat ini memungkinkan terjadinya kesalahan penggunaan yang berakibat fatal. Pengguna mungkin menerapkan analisis yang tidak tepat terhadap kumpulan data dengan

menggunakan pendekatan yang berbeda. Oleh karenanya, dibutuhkan pemahaman tentang statistik dan struktur model matematika yang mendasari kerja perangkat lunak [LAR-06].



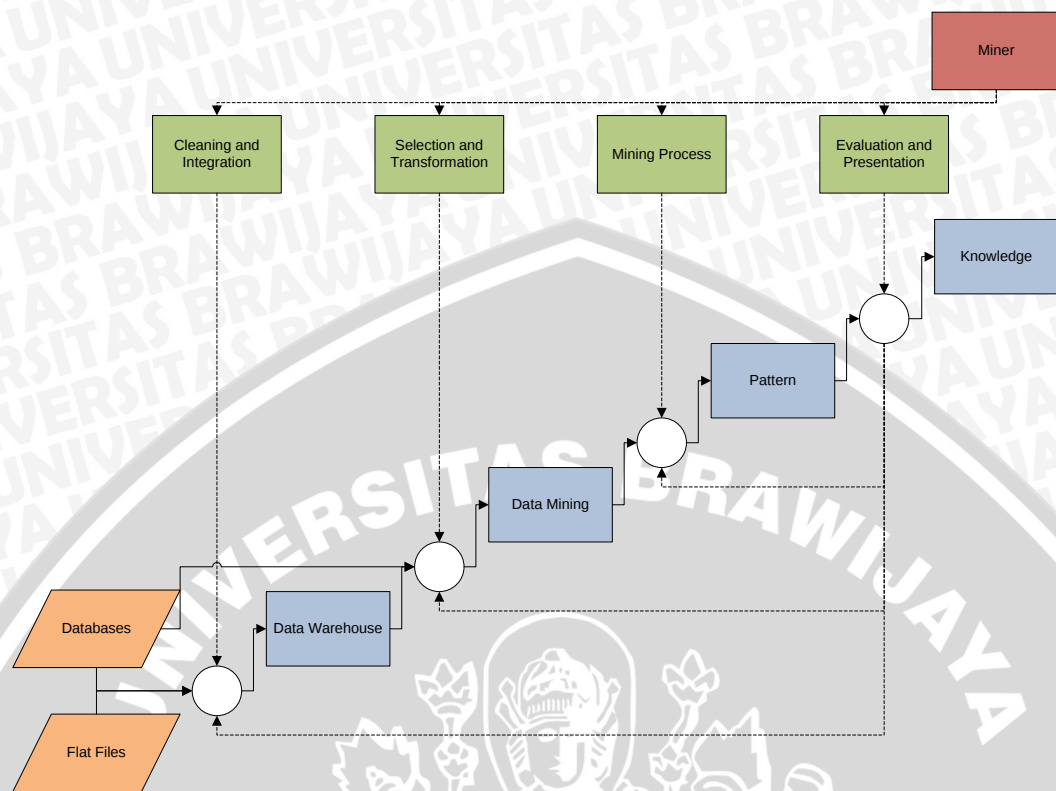
**Gambar 2.1 Bidang Ilmu Data Mining**

Sumber: [LAR-06]

*Data mining* bukanlah suatu bidang yang sama sekali baru. Salah satu kesulitan untuk mendefinisikan *data mining* adalah kenyataan bahwa *data mining* mewarisi banyak aspek dan teknik dari bidang-bidang ilmu yang sudah mapan terlebih dahulu. Gambar 2.1 menunjukkan bahwa *data mining* memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), *machine learning*, statistik, *database*, dan juga *information retrieval* [PRA-06].

### 2.3.1 Tahap-Tahap Data Mining

*Data mining* dipahami sebagai suatu proses, yang memiliki tahapan-tahapan tertentu yang bersifat interaktif dan juga ada umpan balik dari setiap tahapan sebelumnya. Adapun tahapan tersebut dapat dilihat pada Gambar 2.2.



**Gambar 2.2 Tahap-Tahap Data Mining**

Sumber: [LAR-06]

Tahap – tahap tersebut, bersifat interaktif dimana pemakai terlibat langsung atau dengan perantaraan *knowledge base*.

### 1. Pembersihan data

Pembersihan data merupakan proses menghilangkan *noise* dan data yang tidak konsisten atau data tidak relevan. Pada umumnya data yang diperoleh, baik dari *database* suatu perusahaan maupun hasil eksperimen, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak *valid* atau juga hanya sekedar salah ketik. Selain itu, ada juga atribut-atribut data yang tidak relevan dengan hipotesa *data mining* yang dimiliki. Data-data yang tidak relevan itu juga lebih baik dibuang. Pembersihan data juga akan mempengaruhi performansi dari teknik *data mining* karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

## 2. Integrasi data

Integrasi data merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru. Tidak jarang data yang diperlukan untuk *data mining* tidak hanya berasal dari satu *database* tetapi juga berasal dari beberapa *database* atau *file* teks. Integrasi data dilakukan pada atribut-atribut yang mengidentifikasi entitas-entitas yang unik seperti atribut nama, jenis produk, nomor pelanggan dan lainnya. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya. Sebagai contoh bila integrasi data berdasarkan jenis produk ternyata menggabungkan produk dari kategori yang berbeda maka akan didapatkan korelasi antar produk yang sebenarnya tidak ada.

## 3. Seleksi data

Data yang ada pada *database* sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari *database*. Sebagai contoh, sebuah kasus yang meneliti faktor kecenderungan orang membeli dalam kasus *market basket analysis*, tidak perlu mengambil nama pelanggan, cukup dengan id pelanggan saja.

## 4. Transformasi data

Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *data mining*. Beberapa metode *data mining* membutuhkan format data yang khusus sebelum bisa diaplikasikan. Sebagai contoh beberapa metode standar seperti analisis asosiasi dan *clustering* hanya bisa menerima masukan data kategorikal. Karenanya data berupa angka numerik yang berlanjut perlu dibagi-bagi menjadi beberapa *interval*. Proses ini sering disebut transformasi data.

### 5. Proses *mining*

Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.

### 6. Presentasi pengetahuan

Untuk mengidentifikasi pola-pola menarik kedalam *knowledge based* yang ditemukan. Dalam tahap ini hasil dari teknik *data mining* berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai. Bila ternyata hasil yang diperoleh tidak sesuai hipotesa ada beberapa alternatif yang dapat diambil seperti menjadikannya umpan balik untuk memperbaiki proses *data mining*, mencoba metode *data mining* lain yang lebih sesuai, atau menerima hasil ini sebagai suatu hasil yang di luar dugaan yang mungkin bermanfaat.

Istilah *data mining* dan *Knowledge Discovery in Database* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses KDD adalah *data mining*. Proses KDD secara garis besar dapat dijelaskan sebagai berikut [FAY-96].

#### 1. Data Selection

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

#### 2. *Pre-processing/Cleaning*

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten,

dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi). Juga dilakukan proses *enrichment*, yaitu proses “memperkaya” data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

### 3. *Transformation*

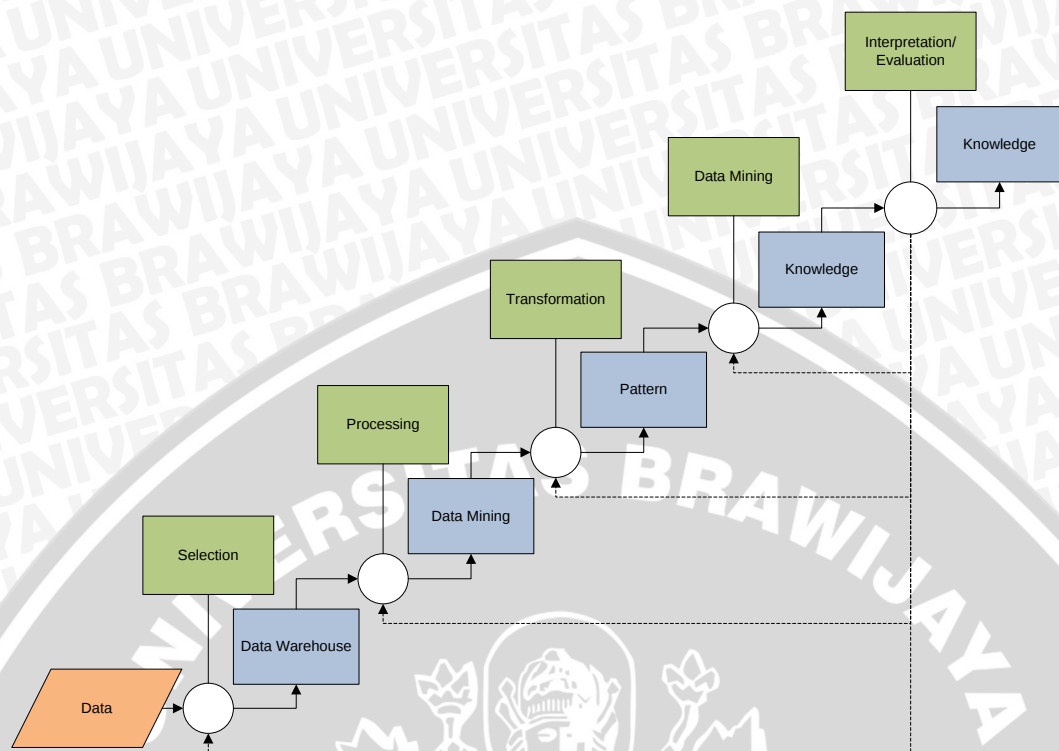
*Coding* adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

### 4. *Data mining*

*Data mining* adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode dan algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

### 5. *Interpretation/Evaluation*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya. Penjelasan diatas dapat di representasikan pada Gambar 2.3



**Gambar 2.3 Proses *Knowledge Discovery In Database* (KDD)**

Sumber: [FAY-96]

### 2.3.2 Pengelompokan *Data Mining*

*Data mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat di lakukan, yaitu [LAR-06].

#### 1. Deskripsi

Terkadang peneliti dan analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

## 2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel *target* estimasi lebih ke arah numerik dari pada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel *target* sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel *target* dibuat berdasarkan nilai variabel prediksi. Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

## 3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.

Contoh prediksi dalam bisnis dan penelitian adalah:

- a. Prediksi harga beras dalam tiga bulan yang akan datang.
- b. Prediksi presentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikan.

Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

## 4. Klasifikasi

Dalam klasifikasi, terdapat *target* variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

Contoh lain klasifikasi dalam bisnis dan penelitian adalah:

- a. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan.
- b. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
- c. Mendiagnosa penyakit seorang pasien untuk mendapatkan termasuk kategori apa.

#### 5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan.

Kluster adalah kumpulan *record* yang memiliki kemiripan suatu dengan yang lainnya dan memiliki ketidakmiripan dengan *record* dalam kluster lain.

Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel *target* dalam pengklusteran. Pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel *target*. Akan tetapi, algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

Contoh pengklusteran dalam bisnis dan penelitian adalah:

- a. Mendapatkan kelompok-kelompok konsumen untuk *target* pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
- b. Untuk tujuan audit akutansi, yaitu melakukan pemisahan terhadap perilaku finansial dalam baik dan mencurigakan.

- c. Melakukan pengklusteran terhadap ekspresi dari *gen*, dalam jumlah besar.

#### 6. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam suatu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja.

Contoh asosiasi dalam bisnis dan penelitian adalah:

- a. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respon positif terhadap penawaran *upgrade* layanan yang diberikan.
- b. Menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli bersamaan.

Untuk mendukung penelitian ini penulis menggunakan metode klasifikasi.

### 2.4 Klasifikasi

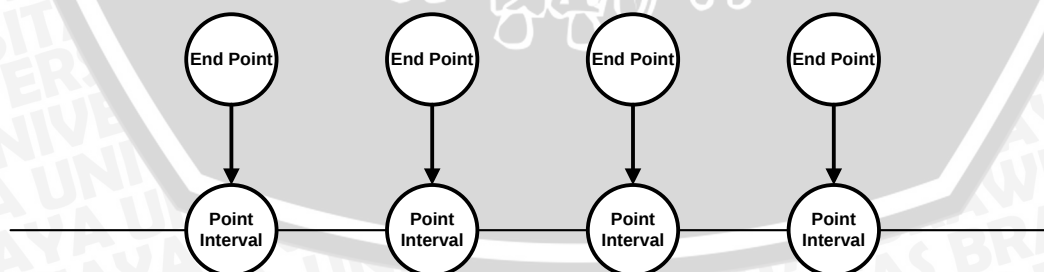
Klasifikasi merupakan proses menemukan sekumpulan model (atau fungsi) yang menggambarkan dan membedakan konsep atau kelas-kelas data. Tujuannya adalah agar model tersebut dapat digunakan untuk memprediksi kelas dari suatu objek atau data yang label kelasnya tidak diketahui [HAN-01].

Klasifikasi terdiri atas dua tahap, yaitu pelatihan dan prediksi. Pada tahap pelatihan, dibentuk sebuah model domain permasalahan dari setiap *instance* yang ada. Penentuan model tersebut berdasarkan analisis pada sekumpulan data pelatihan, yaitu data yang label kelasnya sudah diketahui. Pada tahap prediksi, dilakukan prediksi kelas dari *instance* (kasus) baru dengan menggunakan model yang telah dibuat pada tahap pelatihan [GUV-98]. Untuk mendukung penelitian ini penulis menggunakan Algoritma *Voting Features Interval-5*(VFI-5).

## 2.5 Voting Feature Interval-5 (VFI5)

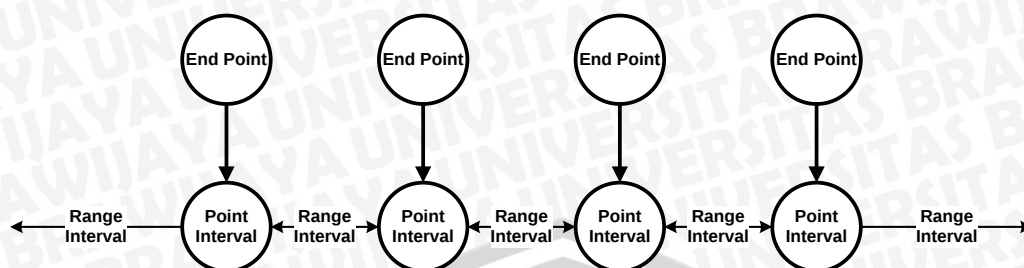
Algoritma klasifikasi *Voting Feature Interval* (VFI5) merepresentasikan deskripsi sebuah konsep oleh sekumpulan *interval* nilai-nilai *feature* atau atribut [DEM-97]. Pengklasifikasian *instance* baru berdasarkan *voting* pada klasifikasi yang dibuat oleh nilai tiap-tiap *feature* secara terpisah, VFI5 merupakan algoritma klasifikasi yang bersifat *non-incremental* di mana semua *instance* pelatihan diproses secara bersamaan [GUV-00].

Dari *instance-instance* pelatihan tersebut, Algoritma VFI5 membuat *interval* untuk setiap *feature*. *Interval-interval* yang dibuat dapat berupa *range interval* atau *point interval*. *Point interval* terdiri atas seluruh *end point* secara berturut-turut. *Range interval* terdiri atas nilai-nilai antara dua *end point* yang berdekatan tetapi tidak termasuk kedua *end point* tersebut. *End point* terbagi menjadi dua macam menurut bentuk datanya, yaitu *end point* untuk *feature linier* dan *end point* untuk *feature nominal*. *End point* untuk *feature linier* merupakan nilai minimum dan nilai maksimum *feature* tersebut, sedangkan *end point* untuk *feature nominal* merupakan semua nilai yang berbeda yang ada pada *feature* kelas yang sedang diamati. Dengan begitu, jika suatu *feature* merupakan *feature linier* maka akan dibentuk *point interval* dan *range interval*, sedangkan jika *feature* tersebut merupakan *feature nominal* maka hanya dibentuk *point interval* [DEM-97]. Untuk lebih jelasnya dalam pembentukan *interval* dapat dilihat pada Gambar 2.4 dan Gambar 2.5



**Gambar 2.4 Proses Pembentukan *Interval* Pada Data Nominal**

Sumber: [DEM-97]



**Gambar 2.5 Proses Pembentukan *Interval* Pada Data Linier**

Sumber: [DEM-97]

Untuk setiap *interval*, *vote* setiap kelas pada *interval* tersebut disimpan. Dengan demikian, sebuah *interval* dapat merepresentasikan beberapa kelas dengan menyimpan *vote* setiap kelas, sehingga algoritma VFI dapat dikatakan sebagai *Multi-Class feature projection based algorithms* [DEM-97].

Keunggulan algoritma VFI 5 adalah algoritma ini cukup kokoh (*robust*) terhadap *feature* yang tidak relevan tetapi mampu memberikan hasil yang baik pada *real-world datasets* yang ada. VFI 5 mampu menghilangkan pengaruh yang kurang menguntungkan dari *feature* yang tidak relevan dengan mekanisme *voting*-nya [GUV-98]. Algoritma Klasifikasi VFI 5 mampu melakukan klasifikasi lebih cepat dibandingkan dengan algoritma *Nearest Neighbor* dan *Decision Tree*, [DEM-97]. VFI 5 mampu menangani nilai *feature* yang tidak diketahui (hilang) dengan cara mengabaikan nilai *feature* tersebut yang ada pada data pelatihan dan data pengujian. Namun pada algoritma klasifikasi lain seperti *Nearest Neighbor* dan *Decision Tree*, nilai tersebut harus diganti [DEM-97]. Algoritma VFI5 terdiri atas dua tahap, yaitu pelatihan dan prediksi klasifikasi.

### 2.5.1 Pelatihan

Langkah pertama pada tahap pelatihan adalah menemukan *end point* setiap *feature*  $f$  dari setiap kelas  $c$ . *Feature* dibagi menjadi dua macam, yaitu *feature linier* dan *feature nominal*. *Feature linier*, yaitu *feature* yang nilainya memiliki urutan atau bisa dibandingkan tingkatannya. *End point* untuk *feature linier* merupakan nilai minimum dan nilai maksimum *feature* tersebut. Contoh nilai *linier* adalah 100, 12.5, dan 0.5. *Feature nominal*, yaitu *feature* yang nilainya

tidak memiliki urutan dan tidak bisa dibandingkan tingkatannya. *End point* untuk *feature* nominal merupakan semua nilai yang berbeda yang ada pada *feature* kelas yang sedang diamati. Contoh nilai nominal adalah *Male*, *Female*, benar, dan salah. Sebelum dibentuk *interval*, seluruh *end point* yang diperoleh untuk setiap *feature linier* diurutkan. Jika suatu *feature* merupakan *feature linier* maka akan dibentuk dua *interval* yaitu *point interval* dan *range interval*. Jika *feature* tersebut merupakan *feature* nominal maka hanya dibentuk *point interval* [DEM-97].

Batas bawah pada *range interval* (ujung paling kiri) adalah  $-\infty$  sedangkan batas atas *range interval* (ujung paling kanan) adalah  $+\infty$ . Jumlah maksimum *end point* pada *feature linier* adalah  $2k$  sedangkan jumlah maksimum *interval*-nya adalah  $4k+1$ , dengan  $k$  adalah jumlah kelas yang diamati. Setelah itu, jumlah *instance* pelatihan setiap kelas  $c$  dengan *feature*  $f$  untuk setiap *interval*  $i$  dihitung dan direpresentasikan sebagai  $interval\_class\_count[f,i,c]$ . Untuk setiap *instance* pelatihan, dicari *interval*  $i$ , yaitu *interval* nilai *feature*  $f$  dari *instance* pelatihan  $e$  ( $e_f$ ) tersebut berada. Jika *interval*  $i$  merupakan *point interval* dan  $e_f$  sama dengan *interval* tersebut, maka jumlah kelas *instance* tersebut ( $e_f$ ) pada *interval*  $i$  ditambah 1. Jika *interval*  $i$  merupakan *range interval* dan  $e_f$  jatuh pada *interval* tersebut maka jumlah kelas *instance*  $e_f$  pada *interval*  $i$  ditambah 1. Hasil proses ini merupakan *vote* kelas  $c$  pada *interval*  $i$  [DEM-97]. Perhitungan *interval\_vote* dapat dirumuskan menjadi Persamaan (2-1)

$$interval\_vote[f,i,c] = \frac{interval\_count[f,i,c]}{class\_count[c]} \quad (2-1)$$

Sumber: [DEM-97]

Untuk menghilangkan efek perbedaan distribusi setiap kelas, *vote* kelas  $c$  untuk *feature*  $f$  pada *interval*  $i$  dinormalisasi dengan cara membagi *vote* tersebut dengan jumlah *instance* kelas  $c$  yang direpresentasikan dengan  $class\_count[c]$ . Hasil normalisasi ini dinotasikan sebagai  $interval\_class\_vote[f,i,c]$ . Kemudian nilai-nilai  $interval\_class\_vote[f,i,c]$  dinormalisasi sehingga jumlah *vote* beberapa kelas pada setiap *feature* sama dengan 1. Normalisasi ini bertujuan agar setiap *feature* memiliki kekuatan *voting* yang sama pada proses klasifikasi yang tidak

dipengaruhi oleh ukurannya [DEM-97]. Perhitungan *interval\_class\_vote* dapat dirumuskan sebagai Persamaan (2-2)

$$\text{interval\_class\_vote}[f,i,c] = \frac{\text{interval\_vote}[f,i,c]}{\text{count\_interval\_vote}[f,i]} \quad (2-2)$$

Sumber: [DEM-97]

### 2.5.2 Prediksi

Proses klasifikasi diawali dengan inisialisasi *vote* setiap kelas dengan nilai nol. Untuk setiap *feature*  $f$ , dicari *interval*  $i$  di mana nilai  $e_f$  jatuh, dengan  $e_f$  merupakan nilai *feature*  $f$  dari *instance* tes  $e$ . Jika  $e_f$  tidak diketahui (hilang), *feature* tersebut tidak di ikutsertakan dalam *voting* (memberikan *vote* nol untuk setiap kelas). Oleh karena itu, *feature* yang memiliki nilai tidak diketahui diabaikan [DEM-97].

Jika  $e_f$  diketahui maka *interval* tersebut dapat ditemukan. *Interval* tersebut dapat menyimpan *instance* pelatihan dari beberapa kelas. Kelas-kelas dalam sebuah *interval* direpresentasikan oleh *vote* kelas-kelas tersebut pada *interval* itu. Untuk setiap kelas  $c$ , *feature*  $f$  memberikan *vote* yang sama dengan *interval\_class\_vote* $[f,i,c]$ . Notasi tersebut merepresentasikan *vote feature*  $f$  yang diberikan untuk kelas  $c$  [DEM-97].

Setiap *feature*  $f$  mengumpulkan *vote-vote*-nya dalam sebuah vektor  $\langle \text{feature\_vote}[f,C_1], \dots, \text{feature\_vote}[f,C_j], \dots, \text{feature\_vote}[f,C_k] \rangle$ , di mana *feature\_vote* $[f,C_j]$  merupakan *vote feature*  $f$  untuk kelas  $C_j$  dan  $k$  adalah jumlah kelas. Kemudian  $d$  vektor *vote*, di mana  $d$  merupakan jumlah *feature*, dijumlahkan untuk memperoleh total vektor *vote*  $\langle \text{vote}[C_1], \dots, \text{vote}[C_k] \rangle$ . Kelas dengan jumlah *vote* terbesar diprediksi sebagai kelas dari *instance* tes  $e$  [DEM-97].

## 2.6 Incomplete Data

Ada beberapa metode untuk mengatasi data tidak lengkap. Cara yang paling mudah untuk mengatasi data tidak lengkap adalah dengan menghapus satu baris data yang tidak lengkap. Teknik ini terkadang menyebabkan hilangnya

informasi yang potensial. Pendekatan yang kedua adalah dengan mengganti semua data hilang dengan rataannya [ENN-01].

Suatu data terdiri dari nilai nominal dan nilai numerik. Salah satu teknik untuk mengatasi data hilang pada nilai nominal adalah mengganti data hilang dengan modus sedangkan untuk nilai numerik adalah mengganti data hilang dengan *mean* [MEI-05].

## 2.7 Evaluasi

Evaluasi merupakan kegiatan yang membandingkan antara hasil implementasi dengan kriteria dan standar yang telah ditetapkan untuk melihat keberhasilannya. Dari evaluasi kemudian akan tersedia informasi mengenai sejauh mana suatu kegiatan tertentu telah dicapai sehingga bisa diketahui bila terdapat selisih antara standar yang telah ditetapkan dengan hasil yang bisa dicapai. Evaluasi yang digunakan pada penelitian ini adalah dengan menghitung akurasi dari hasil klasifikasi yang di hasilkan oleh aplikasi.

### 2.7.1 Akurasi

Sistem klasifikasi data mengembalikan sekumpulan data sebagai hasil dari klasifikasi. Ukuran umum yang digunakan untuk mengukur kualitas dari data klasifikasi adalah akurasi. Akurasi menyatakan seberapa dekat nilai hasil pengukuran dengan nilai sebenarnya (*true value*) atau nilai yang dianggap benar (*accepted value*) [LAB-11]. Jika tidak ada data sebenarnya atau nilai yang dianggap benar, maka tidak mungkin untuk menentukan berapa akurasi dari pengukuran tersebut. Perhitungan akurasi dapat dirumuskan sebagai Persamaan (2-3)

$$\text{tingkat akurasi} = \frac{\sum \text{data uji benar diklasifikasi}}{\sum \text{total data uji}} \quad (2-3)$$

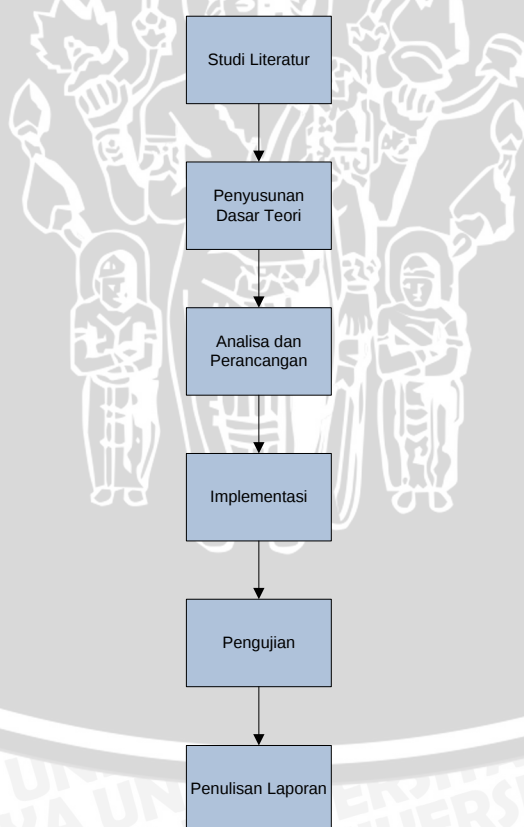
Sumber: [BUO-09]

## BAB III

### METODOLOGI PENELITIAN DAN PERANCANGAN

#### 3.1 Metodologi Penelitian

Pada sub bab ini dijelaskan langkah-langkah yang akan dilakukan dalam pengerjaan skripsi, yaitu studi literatur, penyusunan dasar teori, analisa dan perancangan, implementasi, analisis dan pengujian dari aplikasi perangkat lunak yang akan dibuat, hingga penulisan laporan. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya. Gambar 3.1 menunjukkan desain penelitian secara umum.



**Gambar 3.1 Desain Penelitian**

Sumber : Perancangan

### 3.1.1 Studi Literatur

Studi literatur mempelajari mengenai penjelasan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut diperoleh dari buku, jurnal, *e-book*, penelitian sebelumnya, dan datatasi *project*.

### 3.1.2 Penyusunan Dasar Teori

Penyusunan dasar teori dilakukan setelah mendapatkan referensi yang tepat untuk mendukung penulisan penelitian ini. Teori-teori pendukung tersebut meliputi:

#### 1. Liver

Meliputi penyebab penyakit liver, gejala penyakit liver, serta diagnosis penyakit liver.

#### 2. Data mining

Meliputi tahap-tahap dalam proses penambangan data (*data mining*) serta pengelompokan *data mining*.

#### 3. Klasifikasi

Meliputi tahap-tahap klasifikasi data sehingga mendapatkan hasil klasifikasi yang tepat

#### 4. Voting Feature Interval-5 (VFI5)

Meliputi tahap-tahap *Voting Feature Interval -5* (VFI5), yaitu tahap pelatihan dan prediksi yang digunakan penulis untuk mendapatkan hasil klasifikasi data penyakit liver pada manusia.

#### 5. Incomplete data

Meliputi tahap mengganti nilai data yang hilang sehingga meningkatkan akurasi dari klasifikasi data penyakit liver pada manusia.

#### 6. Evaluasi

Meliputi pengertian akurasi yang berfungsi pada saat pengujian sistem.

### 3.1.3 Analisa dan Perancangan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan oleh sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi kebutuhan sistem dan siapa saja yang terlibat di dalamnya. Berikut analisis kebutuhan dalam aplikasi klasifikasi data penyakit liver pada manusia ini.

#### 3.1.3.1 Kebutuhan Antar Muka

Kebutuhan-kebutuhan untuk pengembangan perangkat lunak ini sebagai berikut :

- 1) Program yang akan dibangun harus mempunyai tampilan yang familiar bagi pemakai.
- 2) Program yang di bangun harus mempunyai tampilan yang memungkinkan pengguna untuk memasukkan data parameter penyakit liver.
- 3) Program yang di bangun harus mampu menampilkan hasil atau keluaran dari proses sebelumnya.

#### 3.1.3.2 Kebutuhan Data

Data yang diolah oleh perangkat lunak ini adalah:

- 1) Data penyakit liver pada manusia digunakan untuk *data* pelatihan dan prediksi dari proses klasifikasi aplikasi ini.
- 2) Data yang diolah berbentuk nilai nominal dan numerik.
- 3) Data hasil proses klasifikasi akan di simpan di dalam *database*.

#### 3.1.3.3 Kebutuhan Fungsional

Fungsi-fungsi yang dimiliki oleh perangkat lunak ini adalah:

- 1) Program harus mampu melakukan proses *preprocessing*, yakni melengkapi data yang tidak lengkap menurut bentuk nilainya.
- 2) Perangkat lunak harus mampu melakukan proses pembobotan pada setiap parameter penyakit liver yang akan di klasifikasi.

- 3) Perangkat lunak harus mampu menghasilkan keluaran yang sesuai dengan proses klasifikasi.

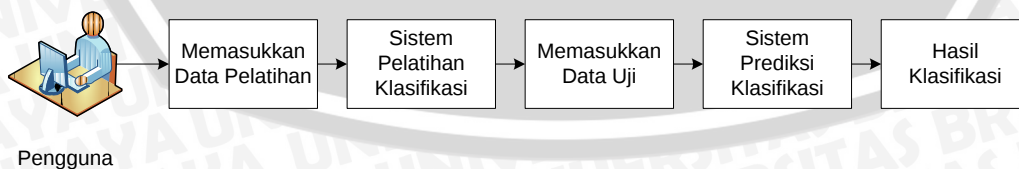
#### 3.1.3.4 Arsitektur Program

Perancangan aplikasi klasifikasi data penyakit liver pada manusia ini dapat dijelaskan sebagai berikut. Pertama, pengguna dapat memasukkan data pelatihan dalam bentuk format xls atau sql. Pengguna juga dapat memasukkan data pelatihan secara manual atau satu per satu. Setelah proses memasukkan data selesai pengguna dapat mulai melakukan proses pelatihan.

Sistem kemudian akan memproses data yang telah dimasukkan oleh pengguna melalui proses yang terdiri dari pelatihan dan proses melengkapi data hilang. Setelah proses pelatihan selesai, pengguna dapat melakukan klasifikasi data dengan cara memasukkan nilai parameter penyakit liver. Kemudian sistem akan mengklasifikasikan data masukan tersebut.

#### 3.1.3.5 Diagram Blok Sistem

Secara umum sistem ini dimulai dengan masukan berupa data pelatihan dari pengguna. Data yang dimasukkan oleh pengguna akan melalui tahapan *preprocessing*, yang dilanjutkan dengan proses pelatihan. Setelah proses pelatihan selesai maka pengguna dapat memasukkan data uji. Data uji yang dimasukkan oleh pengguna akan melalui tahap *preprocessing*, yang dilanjutkan dengan proses prediksi klasifikasi untuk mendapatkan hasil klasifikasi. Gambar 3.2 menunjukkan *diagram* blok klasifikasi dari aplikasi Klasifikasi *Incomplete data* Penyakit Liver Pada Manusia Dengan Menggunakan Algoritma *Voting Feature Interval-5* (VFI5).



**Gambar 3.2 Diagram Blok Klasifikasi**

Sumber: Perancangan

Diagram blok sistem secara keseluruhan terdiri dari tiga komponen utama yaitu proses memasukkan data pelatihan, memasukkan data uji, dan hasil klasifikasi. Dalam aplikasi ini terdapat dua sistem yang berjalan, yaitu sistem pelatihan klasifikasi dan sistem prediksi klasifikasi. Komponen utama dan sistem dari aplikasi ini dapat dijelaskan sebagai berikut:

a) Memasukkan Data Pelatihan

Bagi pengguna, proses memasukkan data pelatihan klasifikasi merupakan proses untuk memberikan masukan untuk sistem pelatihan klasifikasi. Data masukan harus sesuai dengan format, yaitu berupa file xls, sql, atau masukan pengguna secara manual. Data masukan dari pengguna bisa merupakan data yang tidak lengkap.

b) Sistem Pelatihan Klasifikasi

Bagi pengguna, sistem pelatihan klasifikasi merupakan proses untuk membentuk pola untuk proses prediksi klasifikasi dengan cara melakukan proses *preprocessing*, yaitu proses untuk mengganti data yang tidak lengkap dengan nilai *mean* atau modus. Setelah proses *preprocessing* selesai maka program akan melakukan proses pelatihan klasifikasi, yaitu proses untuk memberikan nilai *vote* dari tiap *interval* untuk tiap *feature* dari data masukan pengguna.

c) Memasukkan Data Uji

Bagi pengguna, proses memasukkan data uji klasifikasi merupakan proses untuk memberikan masukan untuk sistem prediksi klasifikasi. Data masukan harus sesuai dengan format, yaitu berupa file xls, sql, atau masukan pengguna secara manual. Data masukan dari pengguna bisa merupakan data yang tidak lengkap.

#### d) Sistem Prediksi Klasifikasi

Bagi pengguna, sistem prediksi klasifikasi merupakan proses untuk mencari hasil klasifikasi dari data uji masukan pengguna dengan cara melakukan proses *preprocessing*, yaitu proses untuk mengganti data yang tidak lengkap dengan nilai *mean* atau modus. Setelah proses *preprocessing* selesai maka program akan melakukan proses prediksi klasifikasi, yaitu proses mencari nilai terbesar dari jumlah *vote* untuk tiap *feature* data uji..

#### e) Hasil Klasifikasi

Bagi pengguna, keluaran dari sistem prediksi adalah hasil klasifikasi yang di hasilkan oleh program sesuai dengan jumlah terbesar bobot *feature* tiap kelas yang di hitung sebelumnya.

### 3.1.4 Implementasi

Implementasi aplikasi klasifikasi data ini dilakukan dengan mengacu pada perancangan sistem. Implementasi perangkat lunak dilakukan menggunakan bahasa pemrograman PHP dan *database* MySQL. Implementasi aplikasi ini meliputi:

- Pembuatan antarmuka pengguna berupa halaman-halaman *web* yang menerima masukan pengguna.
- Melakukan proses *preprocessing* pada data yang di masukkan oleh pengguna.
- Melakukan proses pembobotan setiap *feature* data dengan *Voting Feature Interval-5 (VFI-5)*.
- Menghasilkan klasifikasi dari data parameter yang telah di masukkan pengguna.

### 3.1.5 Pengujian

Pengujian perangkat lunak pada penelitian ini dilakukan agar dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang melandasinya. Pengujian yang dilakukan meliputi:

- Data pelatihan dan data uji berasal dari UCI *Machine Learning* ([url: http://archive.ics.uci.edu/ml/datasets/ILPD+\(Indian+Liver+Patient+Dataset\)](http://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset))) yang berjumlah 583 *record*. *Data set* ini mengandung 416 *record* dengan kelas terjangkit penyakit liver dan 167 *record* dengan kelas tidak terjangkit penyakit liver. *Data set* ini mengandung 441 *record* pasien dengan jenis kelamin laki-laki dan 142 *record* pasien dengan jenis kelamin perempuan. *Data set* ini diambil dari timur laut dari negara bagian Andhra Pradesh di negara India.
- *Data set* ini memiliki 11 variabel yang dapat dijelaskan sebagai berikut:
  - *Age* : Umur dari pasien
  - *Gender* : Jenis Kelamin dari pasien
  - *TB* : *Total Bilirubin*
  - *DB* : *Direct Bilirubin*
  - *Alkphos* : *Alkaline Phosphotase*
  - *Sgpt* : *Alamine Aminotransferase*
  - *Sgot* : *Aspartate Aminotransferase*
  - *TP* : *Total Protiens*
  - *ALB* : *Albumin*
  - *A/G Ratio* : *Albumin and Globulin Ratio*
  - *Selector* : *Field* yang digunakan untuk membagi data menjadi dua set (kelas terjangkit penyakit liver dan kelas tidak terjangkit penyakit liver). *Field* ini di isi oleh pakar
- Dari 583 *record data set* yang dapat digunakan hanya 579 *record*, 4 *record* tidak dapat digunakan karena ada data yang hilang, sehingga tidak dapat digunakan dalam pengujian. Dari 579 *record data set* ini mengandung 414 *record* dengan kelas terjangkit penyakit liver, 165 *record* dengan kelas tidak terjangkit penyakit liver, 439 *record* pasien dengan jenis kelamin laki-laki dan 140 *record* pasien dengan jenis kelamin perempuan.

- Pengukuran tingkat akurasi program menggunakan 70% *data set* sebagai data pelatihan atau sebanyak kurang lebih 405, sedangkan *data test* yang menggunakan 30% dari *data set* yaitu sebanyak 174 *record*.
- Pengukuran tingkat akurasi *data test* akan di hitung menurut akurasi.
- Pengukuran tingkat akurasi program juga dilakukan dengan menggunakan tingkat *incomplete data* sebesar 2%, 5%, 10%, 15%, 20%, 25%, 50% dan 70% dari data pelatihan. Data yang dihilangkan akan diletakkan secara acak pada semua *feature / field*. Perhitungan jumlah data yang hilang dihitung dari prosentase dari seluruh jumlah data. Jumlah keseluruhan data didapat dari hasil perkalian jumlah *record* dengan jumlah variabel yaitu sebesar 4455 data. Contoh untuk jumlah data dengan *incomplete data* pelatihan sebesar 2% didapatkan sejumlah data sebesar 4366 data.
- Pengukuran tingkat akurasi program juga dilakukan dengan menggunakan tingkat *incomplete data* sebesar 2%, 5%, 10%, 15%, 20%, 25%, 50% dan 70% dari data uji. Data yang dihilangkan akan diletakkan secara acak pada semua *feature / field*. Perhitungan jumlah data yang hilang dihitung dari prosentase dari seluruh jumlah data. Jumlah keseluruhan data didapat dari hasil perkalian jumlah *record* dengan jumlah variabel yaitu sebesar 1740 data. Contoh untuk jumlah data dengan *incomplete data* pelatihan sebesar 10% didapatkan sejumlah data sebesar 1567 data. Contoh tabel pengujian dapat dilihat pada Tabel 3.1.
- Pengujian hasil program dengan cara membandingkan keluaran program dengan keluaran manual.

**Tabel 3.1 Contoh Tabel Pengujian**

| Perlakuan                                   | Tingkatan<br><i>Incomplete</i><br>Data Pelatihan | Tingkatan<br><i>Incomplete</i><br>Data Uji | Akurasi |
|---------------------------------------------|--------------------------------------------------|--------------------------------------------|---------|
| Menghapus satu baris <i>incomplete</i>      | 70%                                              | 0%                                         | -       |
| <i>data</i> , mengabaikan <i>incomplete</i> | 50%                                              | 0%                                         | -       |

|                                                                      |     |     |   |
|----------------------------------------------------------------------|-----|-----|---|
| data, mengganti <i>incomplete data</i> dengan <i>mean</i> atau modus | 25% | 0%  | - |
|                                                                      | 20% | 0%  | - |
|                                                                      | 15% | 0%  | - |
|                                                                      | 10% | 0%  | - |
|                                                                      | 5%  | 0%  | - |
|                                                                      | 2%  | 0%  | - |
|                                                                      | 0%  | 70% | - |
|                                                                      | 0%  | 50% | - |
|                                                                      | 0%  | 25% | - |
|                                                                      | 0%  | 20% | - |
|                                                                      | 0%  | 15% | - |
|                                                                      | 0%  | 10% | - |
|                                                                      | 0%  | 5%  | - |
|                                                                      | 0%  | 2%  | - |
| Data lengkap                                                         | 0%  | 0%  | - |

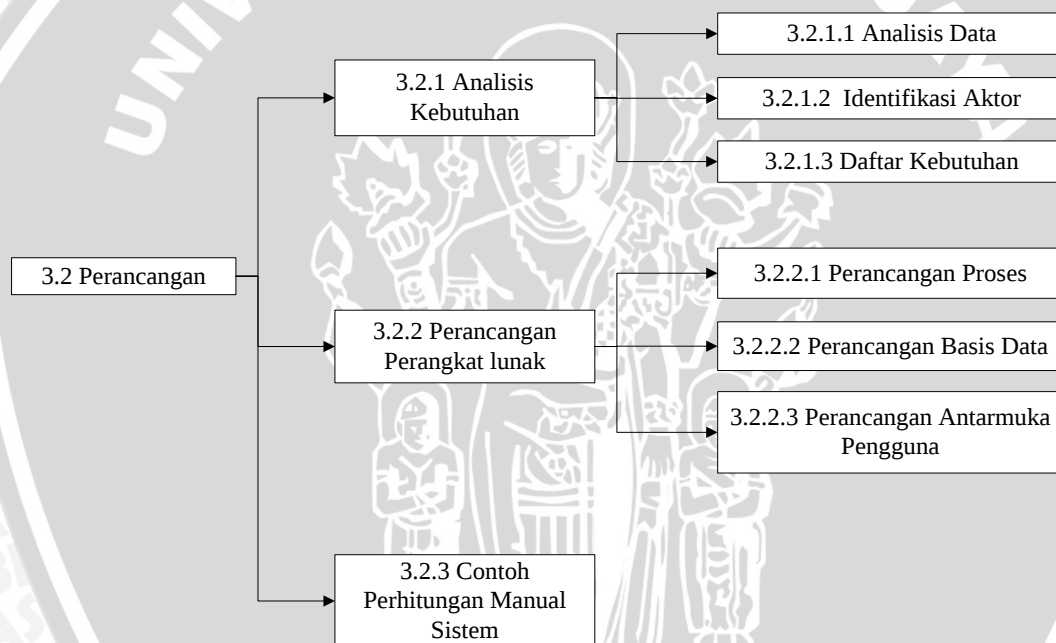
Sumber: Perancangan

### 3.1.6 Penulisan Laporan

Laporan penelitian ditulis setelah semua proses pengerjaan skripsi dilalui. Laporan berisi datatasi perancangan aplikasi yang akan berguna untuk pengembangan aplikasi selanjutnya.

## 3.2 Perancangan

Sub bab ini membahas mengenai perancangan aplikasi klasifikasi data. Perancangan yang dilakukan meliputi analisa kebutuhan, perancangan perangkat lunak, dan contoh penghitungan manual sistem yang akan dibangun. Tahap analisis kebutuhan terdiri dari tiga langkah yaitu analisis data yang diperlukan, identifikasi aktor, dan membuat daftar kebutuhan sistem. Proses perancangan perangkat lunak mempunyai tiga tahap, yaitu perancangan proses sistem klasifikasi data, perancangan *basis datae*, dan perancangan antarmuka pengguna. Tahap-tahap perancangan yang dilakukan seperti yang digambarkan pada Gambar 3.3 berikut ini.



**Gambar 3.3 Diagram Blok Perancangan**

Sumber : Perancangan

### 3.2.1 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan oleh sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi kebutuhan sistem dan siapa saja yang terlibat di dalamnya.

Berikut analisis kebutuhan dalam aplikasi klasifikasi *incomplete data* pada penyakit liver.

### 3.2.1.1 Analisis Data

Analisis data bertujuan untuk mendapatkan struktur penyimpanan data yang dibutuhkan aplikasi klasifikasi data. Struktur penyimpanan data pada aplikasi klasifikasi *incomplete data* penyakit liver disusun berdasarkan analisis data sebagai berikut :

- 1) Data pelatihan dan uji yang akan digunakan dalam aplikasi ini di sadur dari website UCI Machine Learning. (url: [http://archive.ics.uci.edu/ml/datasets/ILPD+\(Indian+Liver+Patient+Dataset\)](http://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset)) )
- 2) Data hasil proses pelatihan klasifikasi akan di simpan di *database*.

### 3.2.1.2 Identifikasi Aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor - aktor yang akan berinteraksi dengan aplikasi klasifikasi *incomplete data* penyakit liver pada manusia. Tabel 3.2 menunjukkan aktor beserta deskripsinya yang merupakan hasil dari proses identifikasi aktor.

**Tabel 3.2 Identifikasi Aktor**

| Aktor    | Deskripsi Aktor                                                                                                           |
|----------|---------------------------------------------------------------------------------------------------------------------------|
| Pengguna | Pengguna merupakan aktor pengguna yang menggunakan sistem klasifikasi <i>incomplete data</i> penyakit liver pada manusia. |

Sumber: Perancangan

### 3.2.1.3 Daftar Kebutuhan

Daftar kebutuhan terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Pada daftar kebutuhan akan dispesifikasikan menjadi dua yaitu spesifikasi kebutuhan fungsional *client* dan spesifikasi kebutuhan fungsional *administrator*. Spesifikasi kebutuhan fungsional *client* ditunjukkan pada Tabel 3.3

Tabel 3.3 Daftar Kebutuhan Fungsional

| ID      | Requirements                                                                                                                                                                                                                                                     | Aktor    | Nama Proses                                   |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-----------------------------------------------|
| SRS_001 | Sistem harus menyediakan antarmuka untuk memasukkan data pelatihan sesuai dengan format yang telah ditentukan.                                                                                                                                                   | Pengguna | Memasukkan Data Pelatihan                     |
| SRS_002 | Sistem memiliki kemampuan untuk melakukan proses <i>preprocessing</i> pada data pelatihan yang di masukkan oleh pengguna, yaitu sistem harus bisa mengganti nilai data yang tidak lengkap ( <i>incomplete data</i> ) sesuai dengan metode yang telah ditentukan. | Sistem   | <i>Preprocessing</i> Pelatihan<br>Klasifikasi |
| SRS_003 | Sistem memiliki kemampuan untuk melakukan pelatihan klasifikasi terhadap data pelatihan yang telah di masukkan oleh pengguna.                                                                                                                                    | Sistem   | Pelatihan Klasifikasi                         |
| SRS_004 | Sistem harus menyediakan antarmuka untuk memasukkan data uji sesuai dengan format yang telah ditentukan.                                                                                                                                                         | Pengguna | Memasukkan Data Uji                           |
| SRS_005 | Sistem memiliki kemampuan untuk melakukan proses <i>preprocessing</i> pada data uji yang di masukkan oleh pengguna, yaitu sistem harus bisa mengganti nilai data yang tidak lengkap ( <i>incomplete data</i> ) sesuai dengan metode yang telah ditentukan.       | Sistem   | <i>Preprocessing</i> Prediksi<br>Klasifikasi  |
| SRS_006 | Sistem memiliki kemampuan untuk melakukan prediksi klasifikasi terhadap data uji yang telah di masukkan oleh pengguna berdasarkan jumlah terbesar bobot <i>feature</i> tiap kelas yang di hitung sebelumnya.                                                     | Sistem   | Prediksi klasifikasi                          |

Sumber: Perancangan

Daftar kebutuhan non-fungsional ditunjukkan pada Tabel 3.4.

**Tabel 3.4 Daftar Kebutuhan Non-fungsional**

| Parameter               | Deskripsi Kebutuhan                                                                                                                     |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <i>Availability</i>     | Perangkat lunak harus dapat beroperasi terus-menerus selama waktu yang diinginkan (24 jam atau bahkan melebihi jam kerja).              |
| <i>Interoperability</i> | Fungsi-fungsi semua operasi perangkat lunak harus dapat berjalan dengan baik, khususnya di sistem PHP dan sistem <i>database</i> MySQL. |
| <i>Portability</i>      | Perangkat lunak harus dapat digunakan di berbagai <i>web browser</i> yang tersedia.                                                     |
| <i>Usability</i>        | Perangkat lunak harus dapat digunakan dengan mudah oleh pengguna                                                                        |

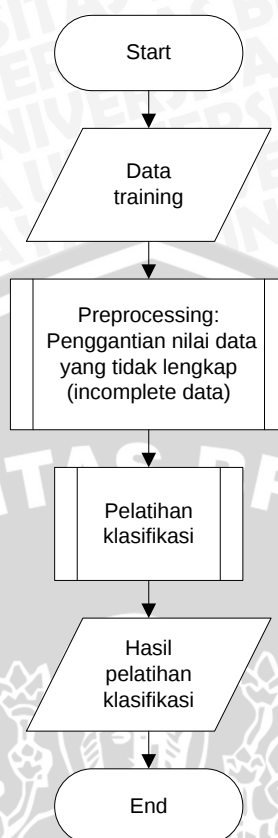
Sumber: Perancangan

### 3.2.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri dari tiga tahap, antara lain perancangan perancangan proses, perancangan *database*, dan perancangan antarmuka pengguna.

#### 3.2.2.1 Perancangan Proses

Perancangan proses merupakan perancangan tahap atau urutan sistem untuk melakukan proses klasifikasi *incomplete data* penyakit liver pada manusia. Berdasarkan diagram blok pada Gambar 3.2 dapat dilihat bahwa secara garis besar aplikasi ini memiliki dua sistem utama, yaitu sistem pelatihan dan sistem prediksi klasifikasi. Dalam sistem pelatihan terdapat dua proses yaitu proses *preprocessing* untuk data pelatihan dan proses pelatihan klasifikasi. Dalam sistem prediksi klasifikasi terdapat dua proses yaitu proses *preprocessing* untuk data uji dan prediksi klasifikasi. Diagram alir sistem untuk pelatihan klasifikasi ditunjukkan pada Gambar 3.4.



**Gambar 3.4 Diagram Alir Sistem Pelatihan Klasifikasi**

Sumber: Perancangan

Proses pertama dalam melakukan klasifikasi data penyakit liver adalah memasukkan data pelatihan. Data pelatihan yang digunakan pada aplikasi ini dapat berupa dokumen yang mempunyai ekstensi xls, sql, atau masukan secara manual dari pengguna. Data pelatihan masukan berupa beberapa *feature* yang berbentuk data nominal maupun numerik. Data pelatihan masukan dari pengguna dapat berupa data yang tidak lengkap. Perancangan algoritma memasukkan data pelatihan mengacu pada spesifikasi kebutuhan SRS\_001 dapat dilihat pada Gambar 3.5.

#### **A. PREPARATION**

Nama algoritma: Memasukkan *Data Pelatihan*

Deskripsi: Proses memasukkan *data pelatihan* ke dalam sistem

Deklarasi:

- String : Umur, Total bilirubin, Direct bilirubin, Alkaline phospotase, Alamine aminotransferase, Aspartate aminotransferase, Total protiens, Albumin, Albumin dan Globulin ratio
- Boolean : Jenis Kelamin dan Kelas Penyakit

## B. ALGORITMA

### Masukan:

- Umur
- Jenis Kelamin
- *Total bilirubin*
- *Direct bilirubin*
- *Alkaline phospotase*
- *Alamine aminotransferase*
- *Aspartate aminotransferase*
- *Total protiens*
- *Albumin*
- *Albumin dan Globulin ratio*
- Kelas Penyakit

### Proses:

1. Mengisi umur pada kolom "Umur"
2. Memilih jenis kelamin pada form "Jenis Kelamin"
3. Mengisi *total bilirubin* pada kolom "*Total bilirubin*"
4. Mengisi *direct bilirubin* pada kolom "*Direct bilirubin*"
5. Mengisi *alkaline phospotase* pada kolom "*Alkaline phospotase*"
6. Mengisi *alamine aminotransferase* pada kolom "*Alamine aminotransferase*"
7. Mengisi *aspartate aminotransferase* pada kolom "*Aspartate aminotransferase*"
8. Mengisi *total protiens* pada kolom "*Total protiens*"
9. Mengisi *albumin* pada kolom "*Albumin*"
10. Mengisi *albumin dan globulin ratio* pada kolom "*Albumin dan Globulin ratio*"
11. Memilih kelas penyakit pada form "Kelas Penyakit" atau memasukkan data dalam file di form yang sudah tersedia.
12. Data masukan dapat berupa data yang tidak lengkap
13. Menekan Tombol "*Submit*"
14. Sistem akan melakukan proses selanjutnya

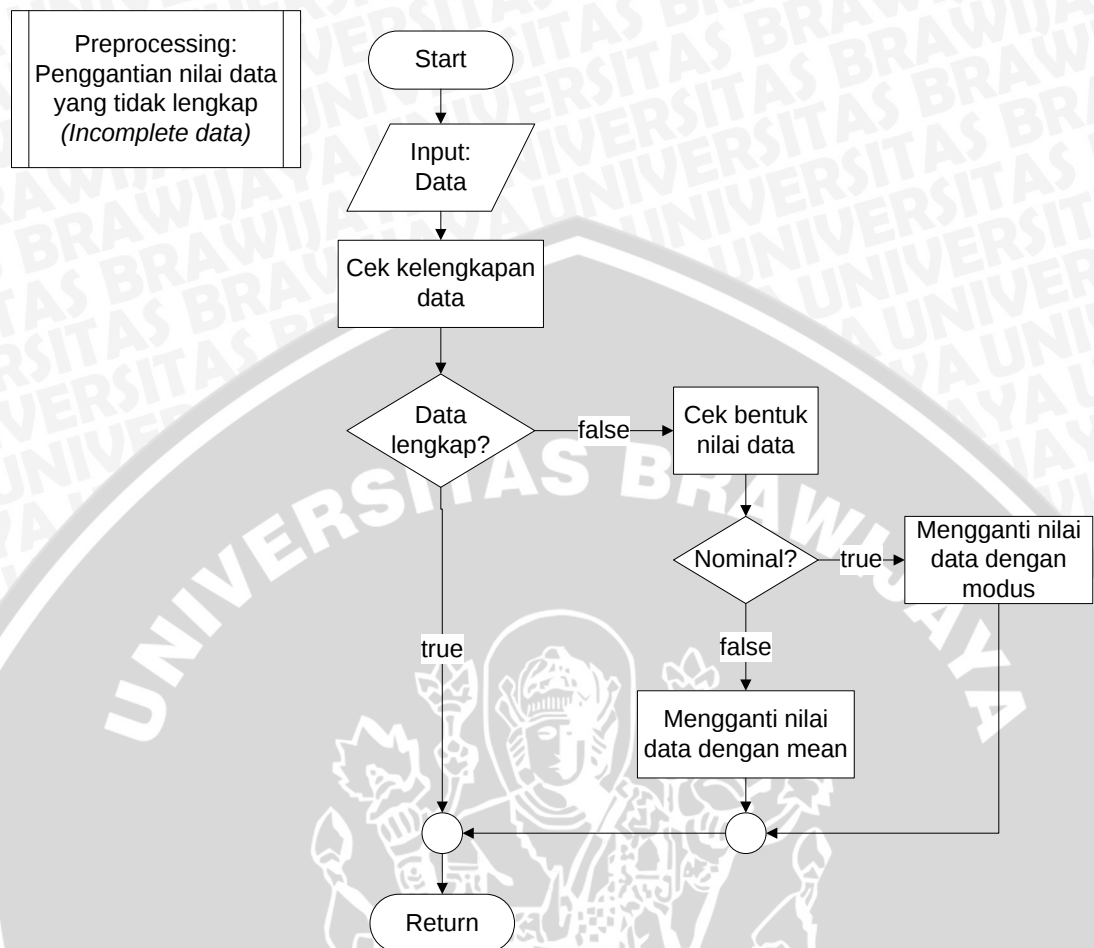
### Keluaran:

- *Data pelatihan*

## Gambar 3.5 Perancangan Algoritma Memasukkan Data Pelatihan

Sumber: Perancangan

Setelah proses memasukkan data pelatihan maka sistem akan melakukan proses *preprocessing*. Diagram alir sistem untuk proses *preprocessing* ditunjukkan pada Gambar 3.6.



**Gambar 3.6 Diagram Alir Proses *Preprocessing***

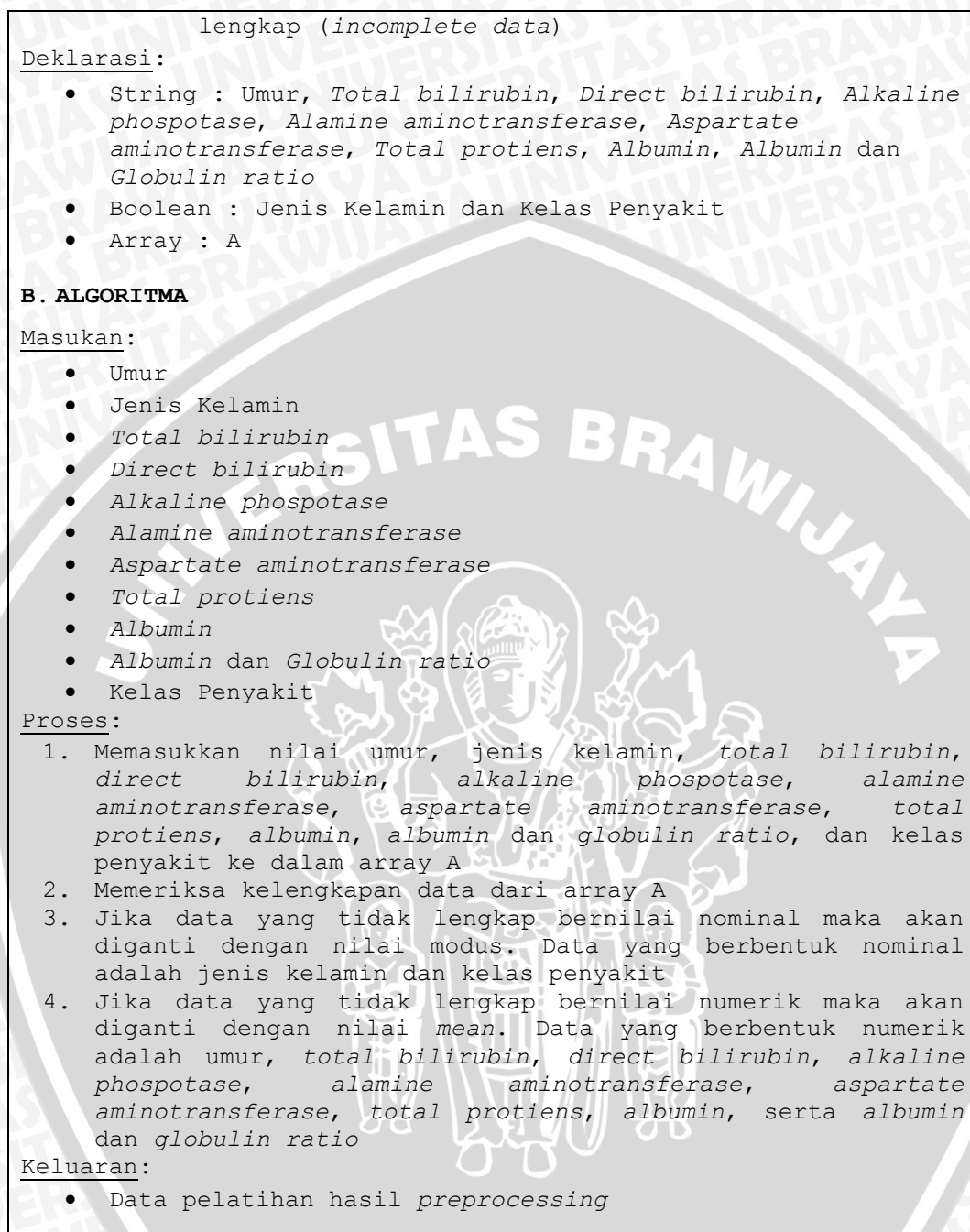
Sumber: Perancangan

Proses *preprocessing* bertujuan untuk mengganti nilai data tidak lengkap dari data masukan pengguna. Pada proses ini sistem akan memeriksa kelengkapan data. Data yang tidak lengkap akan diganti dengan *mean* atau modus. Data akan diganti dengan *mean* jika data yang tidak lengkap berbentuk nominal. Data akan diganti dengan modus jika data yang tidak lengkap berbentuk numerik. Perancangan algoritma *preprocessing* pelatihan mengacu pada spesifikasi kebutuhan SRS\_002 dapat dilihat pada Gambar 3.7.

#### **A. PREPARATION**

Nama algoritma: *Preprocessing* Pelatihan Klasifikasi

Deskripsi: Proses *preprocessing* pada data pelatihan masukan pengguna adalah penggantian nilai data yang tidak

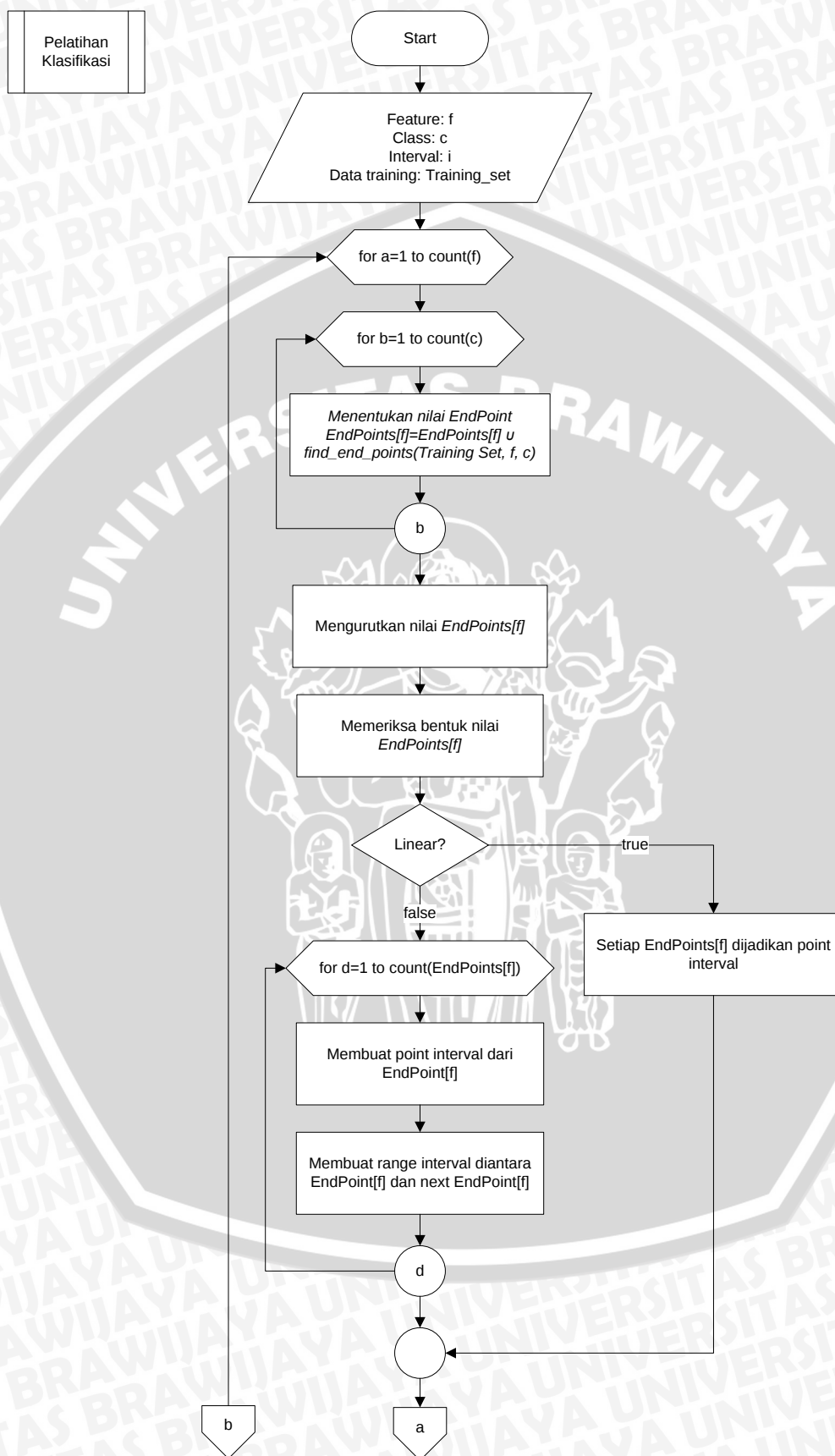


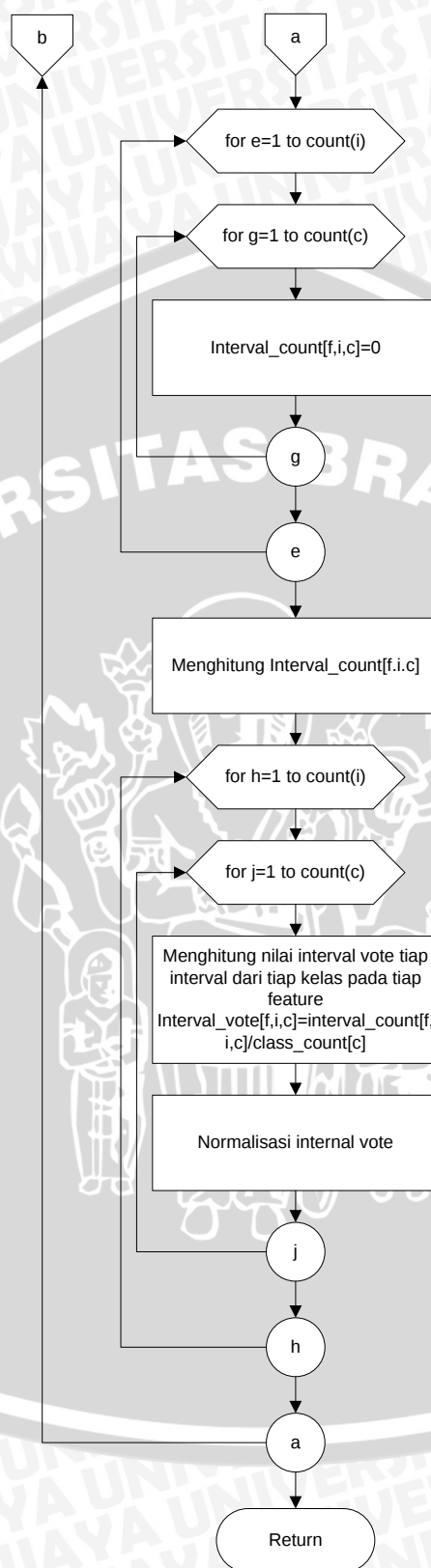
**Gambar 3.7 Perancangan Algoritma Proses *Preprocessing* Pelatihan**

### Klasifikasi

Sumber: Perancangan

Setelah proses *preprocessing* pelatihan maka sistem akan melakukan proses pelatihan klasifikasi. Diagram alir sistem untuk proses pelatihan klasifikasi ditunjukkan pada Gambar 3.8.





**Gambar 3.8 Diagram Alir Proses Pelatihan Klasifikasi**

Sumber: Perancangan

Proses pelatihan klasifikasi bertujuan untuk mendapatkan pola dari data pelatihan masukan dari pengguna yang akan ditampilkan berupa hasil pelatihan klasifikasi. Pola yang terbentuk dari pelatihan akan digunakan untuk menentukan hasil klasifikasi melalui proses prediksi klasifikasi. Pada proses ini sistem akan membentuk nilai *end point* dari tiap *feature*. *End point* yang telah terbentuk akan dijadikan *interval* dari tiap *feature* menurut bentuk datanya. Data berbentuk nominal akan membentuk *point interval* saja. Data berbentuk numerik akan membentuk *point interval* dan *range interval*. *Interval vote* akan dihitung setelah semua *interval* terbentuk, kemudian *interval vote* akan dinormalisasi. Perancangan algoritma pelatihan klasifikasi mengacu pada spesifikasi kebutuhan SRS\_003 dapat dilihat pada Gambar 3.9.

#### A. PREPARATION

Nama algoritma: Pelatihan Klasifikasi

Deskripsi: Proses pembobotan setiap *feature* hasil dari *preprocessing*

Deklarasi:

- Array : Data pelatihan hasil *preprocessing*, *endpoint*, *interval count*, *interval vote*, *interval class vote*
- Int : F, I, C

#### B. ALGORITMA

Masukan:

- Data hasil *preprocessing*

Proses:

1. Menghitung jumlah *feature* (F)
2. Menghitung jumlah kelas (C)
3. Menghitung nilai *endpoint* setiap *feature* dari setiap kelas dengan rumus  
 $EndPoints[f]=EndPoints[f] \cup find\_end\_points(Training\ Set, f, c)$
4. Memasukkan nilai *endpoint* ke dalam array *endpoint*
5. Mengurutkan *endpoint*
6. Memeriksa bentuk *endpoint*
7. Jika *endpoint* berbentuk linear maka sistem akan menghitung jumlah *endpoint* pada tiap *feature*, kemudian membuat *point interval* dari *endpoint* dan membuat *range interval* diantara *endpoint* dan *endpoint* berikutnya.
8. Jika *endpoint* tidak berbentuk linear maka sistem akan membuat *point interval* dari *endpoint* dan tidak membuat *range interval*.
9. Menghitung jumlah *interval* (I)
10. Menetapkan *interval\_count*=0
11. Menghitung jumlah *interval\_count* tiap kelas dari tiap *interval* dan dari tiap *feature*
12. Memasukkan nilai *interval count* ke dalam array *interval*

count

13. Menghitung *interval\_vote* dari tiap kelas dari tiap *interval* dan dari tiap *feature* dengan rumus  $Interval\ vote[f,i,c]=interval\_count[f,i,c]/class\_count[c]$
14. Memasukkan nilai *interval\_vote* ke dalam array *interval vote*
15. Melakukan normalisasi dari *interval vote* sebagai *interval\_class\_vote*, kemudian memasukkannya ke dalam array *interval class vote*.

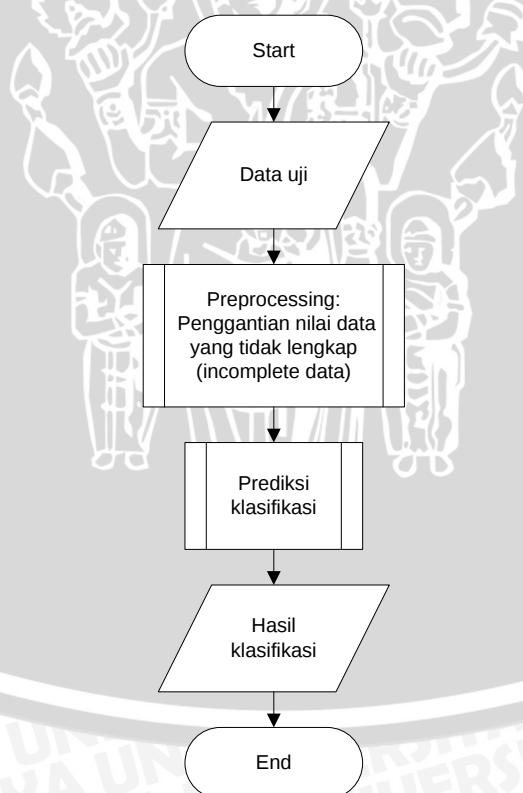
Keluaran:

- *Interval class vote* dari tiap kelas dari tiap *interval* dan dari tiap *feature*
- Tabel data pelatihan yang telah melalui proses *preprocessing* dan pelatihan klasifikasi

**Gambar 3.9 Perancangan Algoritma Proses Pelatihan Klasifikasi**

Sumber: Perancangan

Setelah proses pelatihan selesai maka pengguna dapat melakukan proses prediksi klasifikasi. Diagram alir sistem untuk prediksi klasifikasi ditunjukkan pada Gambar 3.10.



**Gambar 3.10 Diagram Alir Sistem Prediksi Klasifikasi**

Sumber: Perancangan

Proses pertama yang dilakukan untuk melakukan prediksi klasifikasi adalah memasukkan data uji. Data uji yang digunakan pada aplikasi ini dapat berupa dokumen yang mempunyai ekstensi xls, sql, atau masukan secara manual dari pengguna. Data uji masukan berupa beberapa *feature* yang berbentuk data nominal maupun numerik. Data uji masukan dari pengguna dapat berupa data yang tidak lengkap. Perancangan algoritma memasukkan data pelatihan mengacu pada spesifikasi kebutuhan SRS\_004 dapat dilihat pada Gambar 3.11.

#### A. PREPARATION

Nama algoritma: Memasukkan Data Uji

Deskripsi: Proses memasukkan data uji ke dalam sistem

Deklarasi:

- String : Umur, Total bilirubin, Direct bilirubin, Alkaline phospotase, Alamine aminotransferase, Aspartate aminotransferase, Total protiens, Albumin, Albumin dan Globulin ratio
- Boolean : Jenis Kelamin

#### B. ALGORITMA

Masukan:

- Umur
- Jenis Kelamin
- Total bilirubin
- Direct bilirubin
- Alkaline phospotase
- Alamine aminotransferase
- Aspartate aminotransferase
- Total protiens
- Albumin
- Albumin dan Globulin ratio

Proses:

1. Mengisi umur pada kolom "Umur"
2. Memilih jenis kelamin pada form "Jenis Kelamin"
3. Mengisi total bilirubin pada kolom "Total bilirubin"
4. Mengisi direct bilirubin pada kolom "Direct bilirubin"
5. Mengisi alkaline phospotase pada kolom "Alkaline phospotase"
6. Mengisi alamine aminotransferase pada kolom "Alamine aminotransferase"
7. Mengisi aspartate aminotransferase pada kolom "Aspartate aminotransferase"
8. Mengisi total protiens pada kolom "Total protiens"
9. Mengisi albumin pada kolom "Albumin"
10. Mengisi albumin dan globulin ratio pada kolom "Albumin dan Globulin ratio" atau memasukkan data dalam file di form yang sudah tersedia.
11. Data masukan dapat berupa data yang tidak lengkap
12. Menekan Tombol "Submit"

13. Sistem akan melanjutkan ke proses selanjutnya  
Keluaran:

- Data uji

### Gambar 3.11 Perancangan Algoritma Memasukkan Data Uji

Sumber: Perancangan

Setelah proses memasukkan data uji maka sistem akan melakukan proses *preprocessing*. Diagram alir sistem untuk proses *preprocessing* ditunjukkan pada Gambar 3.6. Proses *preprocessing* bertujuan untuk mengganti nilai data tidak lengkap dari data masukan pengguna. Pada proses ini sistem akan memeriksa kelengkapan data. Data yang tidak lengkap akan diganti dengan *mean* atau modus. Data akan diganti dengan *mean* jika data yang tidak lengkap berbentuk nominal. Data akan diganti dengan modus jika data yang tidak lengkap berbentuk numerik. Perancangan algoritma *preprocessing* prediksi mengacu pada spesifikasi kebutuhan SRS\_005 dapat dilihat pada Gambar 3.12.

#### A. PREPARATION

Nama algoritma: *Preprocessing* Prediksi Klasifikasi

Deskripsi: Proses *preprocessing* pada data uji masukan pengguna adalah penggantian nilai data yang tidak lengkap (*incomplete data*)

Deklarasi:

- String : Umur, Total bilirubin, Direct bilirubin, Alkaline phosphotase, Alamine aminotransferase, Aspartate aminotransferase, Total protiens, Albumin, Albumin dan Globulin ratio
- Boolean : Jenis Kelamin
- Array : A

#### B. ALGORITMA

Masukan:

- Umur
- Jenis Kelamin
- Total bilirubin
- Direct bilirubin
- Alkaline phosphotase
- Alamine aminotransferase
- Aspartate aminotransferase
- Total protiens
- Albumin
- Albumin dan Globulin ratio

Proses:

1. Memasukkan nilai umur, jenis kelamin, total bilirubin, direct bilirubin, alkaline phosphotase, alamine

aminotransferase, aspartate aminotransferase, total protiens, albumin, albumin dan globulin ratio ke dalam array A

2. Memeriksa kelengkapan data dari array A
3. Jika data yang tidak lengkap bernilai nominal maka akan diganti dengan nilai modus. Data yang berbentuk nominal adalah jenis kelamin.
4. Jika data yang tidak lengkap bernilai numerik maka akan diganti dengan nilai *mean*. Data yang berbentuk numerik adalah umur, total bilirubin, direct bilirubin, alkaline phospotase, alamine aminotransferase, aspartate aminotransferase, total protiens, albumin, serta albumin dan globulin ratio

Keluaran:

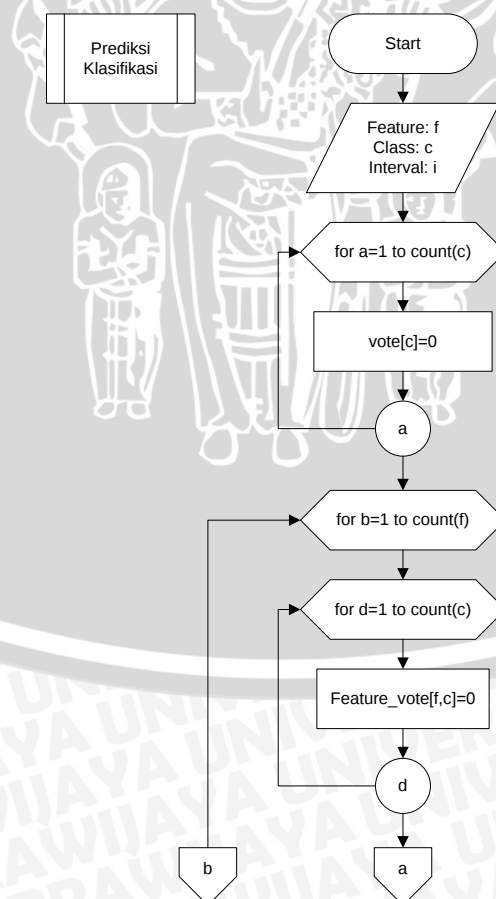
- Data uji hasil *preprocessing*

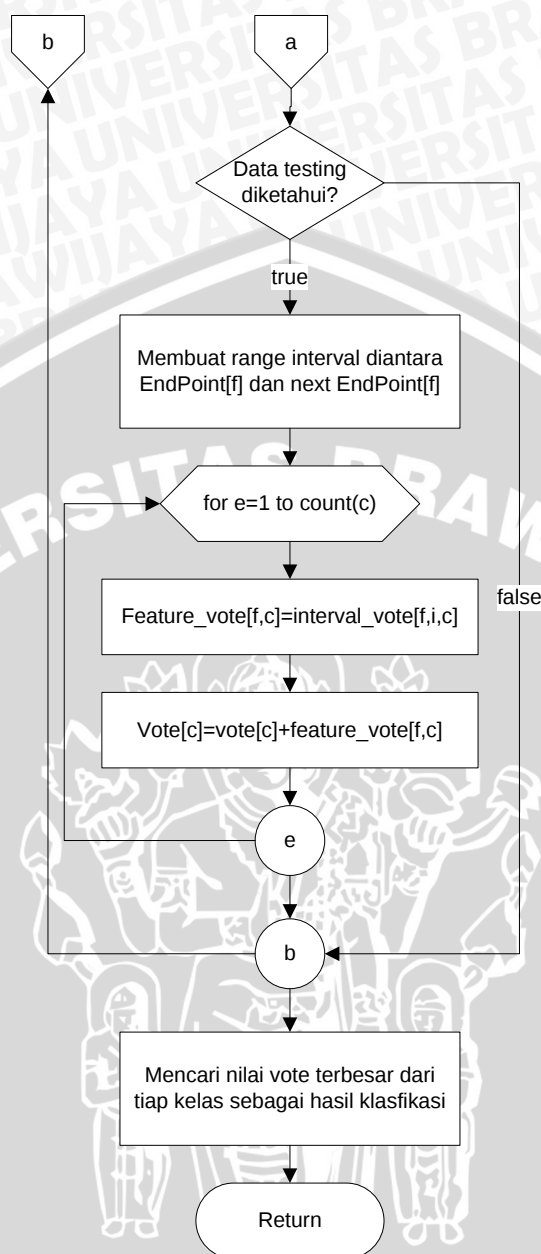
**Gambar 3.12 Perancangan Algoritma Proses *Preprocessing* Prediksi**

### Klasifikasi

Sumber: Perancangan

Setelah proses *preprocessing* prediksi selesai, maka sistem akan melakukan proses prediksi klasifikasi. Diagram alir sistem untuk proses prediksi klasifikasi ditunjukkan pada Gambar 3.13.





**Gambar 3.13 Diagram Alir Proses Prediksi Klasifikasi**

Sumber: Perancangan

Proses prediksi klasifikasi bertujuan untuk mendapatkan hasil klasifikasi dari data uji masukan dari pengguna. Pada proses ini sistem akan memeriksa nilai *feature* dari tiap data uji yang telah dimasukkan oleh pengguna. Nilai *feature* akan diperiksa kecocokannya dengan *interval* yang telah terbentuk dalam proses pelatihan klasifikasi. Dari *interval* yang telah ditemukan, kemudian sistem akan mengambil *interval vote* untuk tiap kelasnya. *Interval vote* ini akan dijumlahkan

untuk seluruh *feature*. *Vote* yang terbesar menurut kelasnya akan dijadikan hasil klasifikasi. Perancangan algoritma proses prediksi klasifikasi mengacu pada spesifikasi kebutuhan SRS\_006 dapat dilihat pada Gambar 3.14.

#### A. PREPARATION

Nama algoritma: Prediksi Klasifikasi

Deskripsi: Proses prediksi klasifikasi berdasarkan jumlah terbesar *interval class vote* dari tiap kelas

Deklarasi:

- Array : *vote*, *interval class vote*, *feature vote*, A
- Int : F, I, C
- String : Umur, Total bilirubin, Direct bilirubin, Alkaline phospotase, Alamine aminotransferase, Aspartate aminotransferase, Total protiens, Albumin, Albumin dan Globulin ratio
- Boolean : Jenis Kelamin

#### B. ALGORITMA

Masukan:

- Umur
- Jenis Kelamin
- Total bilirubin
- Direct bilirubin
- Alkaline phospotase
- Alamine aminotransferase
- Aspartate aminotransferase
- Total protiens
- Albumin
- Albumin dan Globulin ratio
- Interval class vote

Proses:

1. Hitung jumlah kelas (C)
2. Menetapkan nilai pada array *vote*=0
3. Menghitung jumlah *feature* (F)
4. Menetapkan nilai *feature\_vote* tiap kelas dari tiap *feature* sebagai 0
5. Memeriksa data uji dengan *interval*, jika cocok maka memasukkan nilai *interval class vote* dari tiap kelas ke dalam array *feature vote*
6. Menambahkan array *feature vote* dengan array *vote*, kemudian disimpan dalam array *vote*
7. Mencari nilai terbesar *vote* dari tiap kelas
8. Menetapkan kelas dari nilai *vote* terbesar sebagai hasil klasifikasi
9. Jika nilai antar *vote* sama maka kelas akan di indentifikasi sebagai kelas tidak terjangkau penyakit.

Keluaran:

- Tabel data uji beserta hasil klasifikasi yang telah melalui proses prediksi klasifikasi

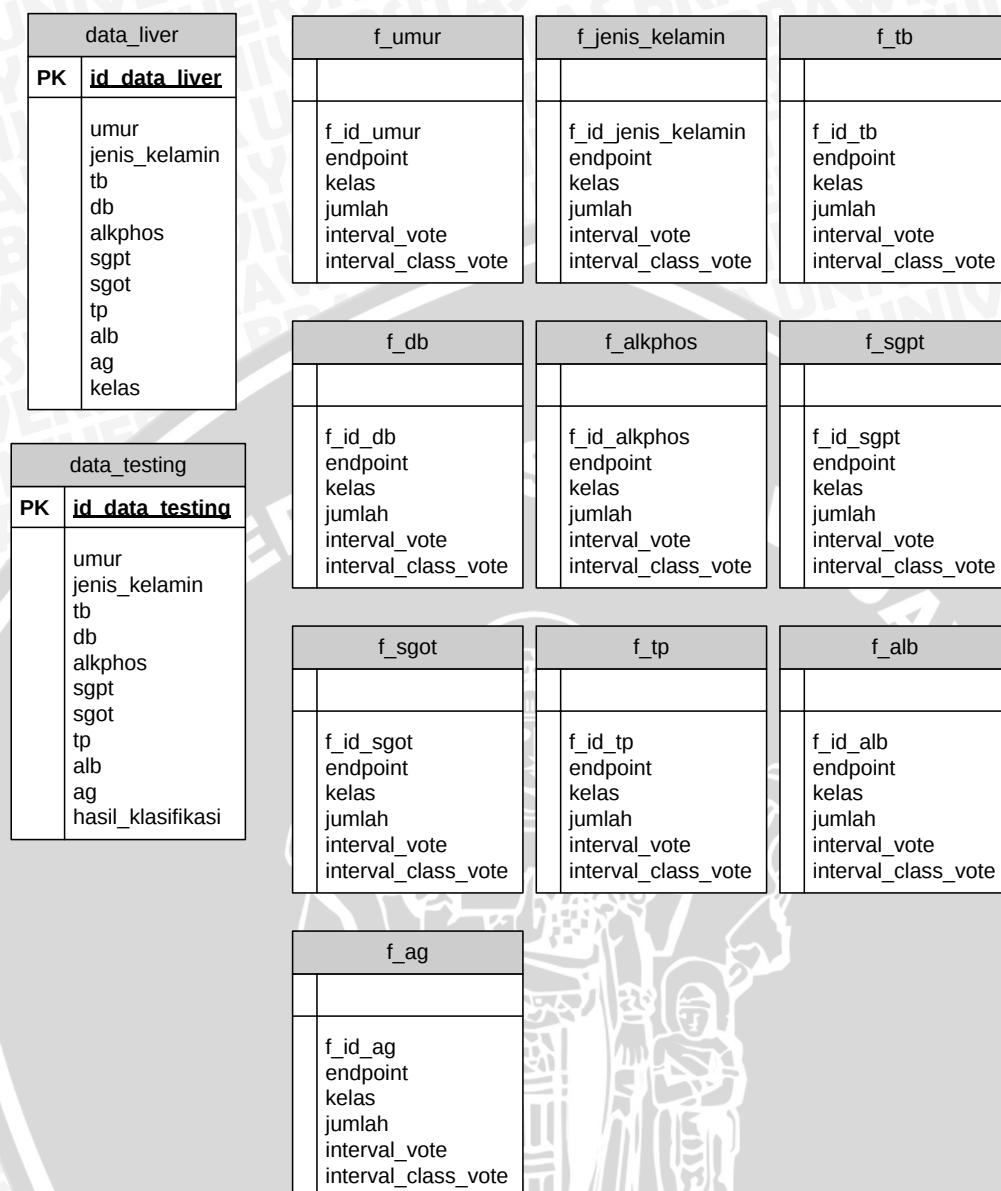
**Gambar 3.14 Perancangan Algoritma Proses Prediksi Klasifikasi**

Sumber: Perancangan

### 3.2.2.2 Perancangan Basis Data

Perancangan *database* merupakan perancangan manajemen data yang akan digunakan sistem. Manajemen data termasuk basis data, yang mengandung data relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System* (DBMS). Sistem ini menggunakan DBMS yaitu MySQL. Perancangan basis data sistem ini menggunakan dua belas tabel yaitu tabel *data\_liver*, *f\_umur*, *f\_jenis\_kelamin*, *f\_tb*, *f\_db*, *f\_alkphos*, *f\_sgpt*, *f\_sgot*, *f\_tp*, *f\_alb*, *f\_ag*, dan *data\_testing*. Adapun perancangan basis data pada sistem ini ditunjukkan pada Gambar 3.15.

Entitas *data\_liver* merepresentasikan tabel yang berisi data penyakit liver pada manusia yang di masukkan oleh pengguna sebagai *data* pelatihan. Tabel *f\_umur* berisi *point interval* dan *range interval* dari *feature* umur beserta *interval class vote*-nya. Tabel *f\_jenis\_kelamin* berisi *point interval* dari *feature* jenis kelamin beserta *interval class vote*-nya. Tabel *f\_tb* berisi *point interval* dan *range interval* dari *feature total bilirubin* beserta *interval class vote*-nya. Table *f\_db* berisi *point interval* dan *range interval* dari *feature direct bilirubin* beserta *interval class vote*-nya. Table *f\_alkphos* berisi *point interval* dan *range interval* dari *feature alkaline phospotase* beserta *interval class vote*-nya. Tabel *f\_sgpt* berisi *point interval* dan *range interval* dari *feature almaline aminotransferase* beserta *interval class vote*-nya. Tabel *f\_sgot* berisi *point interval* dan *range interval* dari *feature aspartate aminotransferase* beserta *interval class vote*-nya. Tabel *f\_tp* berisi *point interval* dan *range interval* dari *feature total protiens* beserta *interval class vote*-nya. Tabel *f\_alb* berisi *point interval* dan *range interval* dari *feature albumin* beserta *interval class vote*-nya. Tabel *f\_ag* berisi *point interval* dan *range interval* dari *feature albumin globulin ratio* beserta *interval class vote*-nya. Tabel *data\_testing* berisi data masukan pengguna untuk melakukan klasifikasi data. Selain itu, tabel *data\_testing* juga berisi hasil klasifikasi. Tidak ada relasi antar tabel, karena tabel disini berfungsi untuk membantu kinerja program dalam melakukan proses klasifikasi data.

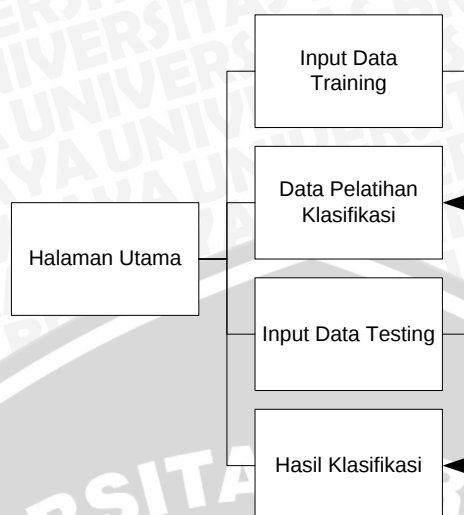


Gambar 3.15 Perancangan Basis Data

Sumber: Perancangan

### 3.2.2.3 Perancangan Antarmuka Pengguna

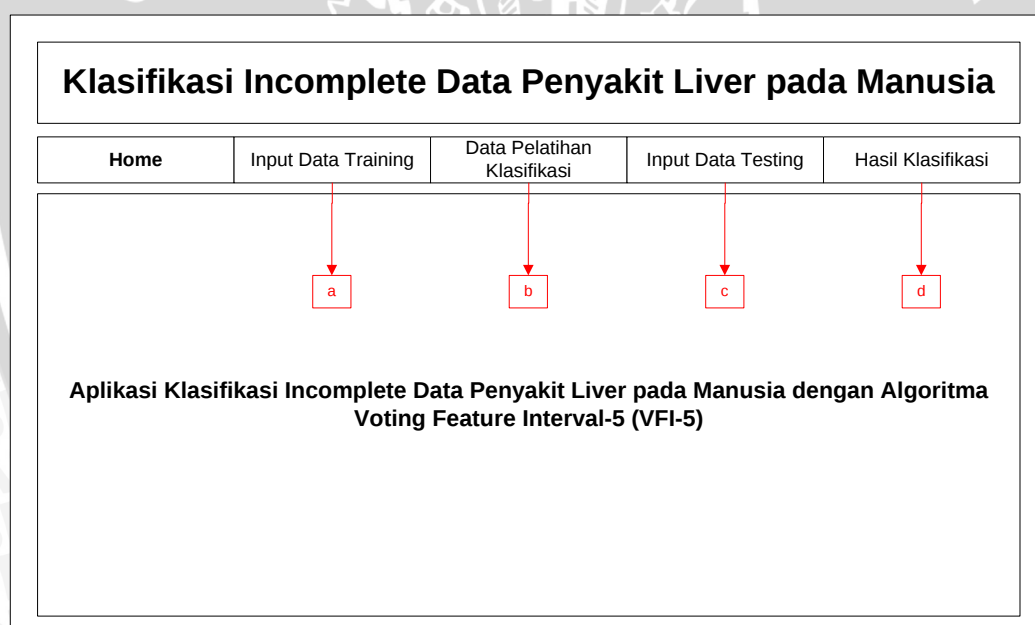
Perancangan antarmuka bertujuan untuk mewakili keadaan sebenarnya dari sistem yang akan dibangun. Sistem klasifikasi *incomplete data* penyakit liver pada manusia dibagi menjadi empat halaman, yaitu halaman utama, halaman proses, halaman hasil, dan halaman about us. *Site map* sistem klasifikasi data ditunjukkan Gambar 3.16.



**Gambar 3.16 Site Map Sistem**

Sumber: Perancangan

1. Perancangan antarmuka pengguna halaman utama

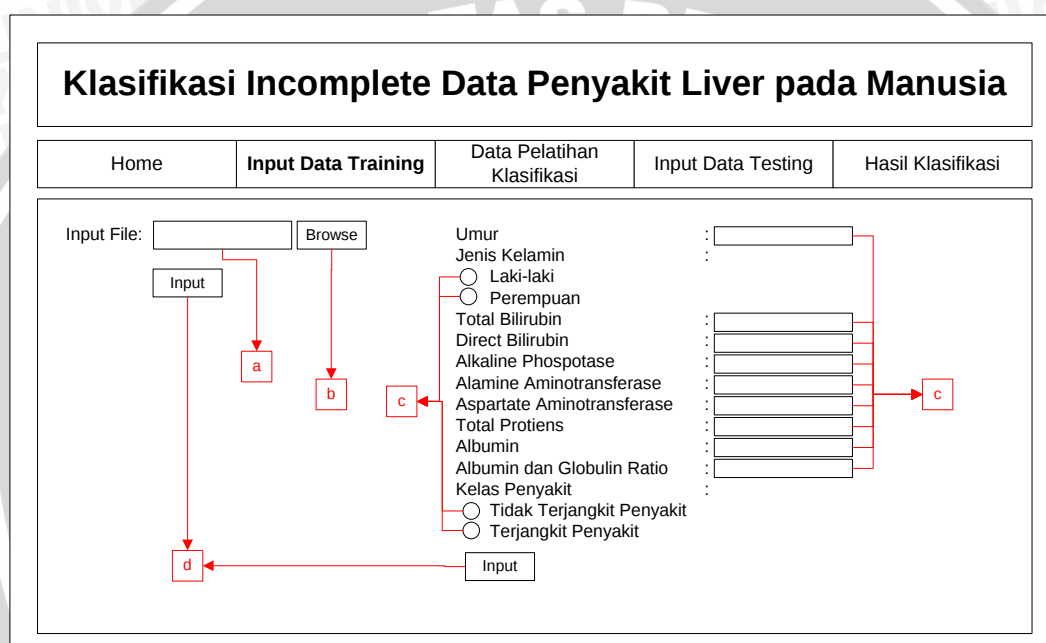


**Gambar 3.17 Perancangan Antarmuka Halaman Utama**

Sumber: Perancangan

Halaman utama menampilkan profil atau gambaran klasifikasi *incomplete data* penyakit liver pada manusia. Gambar 3.17 memiliki keterangan sebagai berikut:

- Tombol Memasukkan Data Pelatihan untuk proses memasukkan data pelatihan.
  - Tombol Data Pelatihan Klasifikasi untuk menampilkan tabel data-data yang masuk dalam pelatihan klasifikasi.
  - Tombol Memasukkan Data Uji untuk proses memasukkan data uji.
  - Tombol Hasil Klasifikasi untuk menampilkan hasil klasifikasi dari data uji.
2. Perancangan antarmuka pengguna halaman memasukkan data pelatihan



**Gambar 3.18 Perancangan Antarmuka Halaman Memasukkan Data Pelatihan**

Sumber: Perancangan

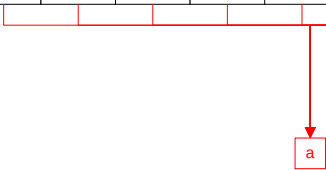
Halaman memasukkan data pelatihan digunakan untuk memasukkan data pelatihan baik secara manual atau berbentuk *file*. Format dokumen yang dapat dimasukkan antara lain xls dan sql. Gambar 3.18 memiliki keterangan sebagai berikut :

- Field* untuk memasukkan data berupa *file*.
- Tombol *browse* untuk memilih *file* data yang akan dimasukkan.

- c. Field untuk memasukkan data secara manual
  - d. Tombol proses untuk menjalankan sistem.
3. Perancangan antarmuka pengguna halaman data pelatihan klasifikasi

### Klasifikasi Incomplete Data Penyakit Liver pada Manusia

| Home |               |                 | Input Data Training |                      |                          | Data Pelatihan Klasifikasi |                |         | Input Data Testing         |                |      | Hasil Klasifikasi |  |
|------|---------------|-----------------|---------------------|----------------------|--------------------------|----------------------------|----------------|---------|----------------------------|----------------|------|-------------------|--|
| Umur | Jenis Kelamin | Total Bilirubin | Direct Bilirubin    | Alkaline Phosphatase | Alamine Aminotransferase | Aspartate Aminotransferase | Total Proteins | Albumin | Albumin dan Globulin Ratio | Kelas Penyakit | Edit | Delete            |  |
|      |               |                 |                     |                      |                          |                            |                |         |                            |                | Edit | Delete            |  |



a



b



c

**Gambar 3.19 Perancangan Antarmuka Halaman Data Pelatihan Klasifikasi**

Sumber: Perancangan

Halaman data pelatihan klasifikasi digunakan untuk menampilkan data pelatihan masukan pengguna yang telah melalui proses *preprocessing*. Gambar 3.19 memiliki keterangan sebagai berikut :

- a. Kolom data pelatihan klasifikasi
- b. Kolom untuk *edit* data pelatihan klasifikasi.
- c. Kolom untuk menghapus data pelatihan klasifikasi.

#### 4. Perancangan antarmuka pengguna memasukkan data uji

### Klasifikasi Incomplete Data Penyakit Liver pada Manusia

|      |                     |                               |                           |                   |
|------|---------------------|-------------------------------|---------------------------|-------------------|
| Home | Input Data Training | Data Pelatihan<br>Klasifikasi | <b>Input Data Testing</b> | Hasil Klasifikasi |
|------|---------------------|-------------------------------|---------------------------|-------------------|

Input File:

a

Umur\*

Jenis Kelamin\*  
☐ Laki-laki  
☐ Perempuan

Total Bilirubin\*

Direct Bilirubin\*

Alkaline Phosphotase\*

Alamine Aminotransferase\*

Aspartate Aminotransferase\*

Total Protiens\*

Albumin\*

Albumin dan Globulin Ratio\*

(\*) Harus di isi

c

d

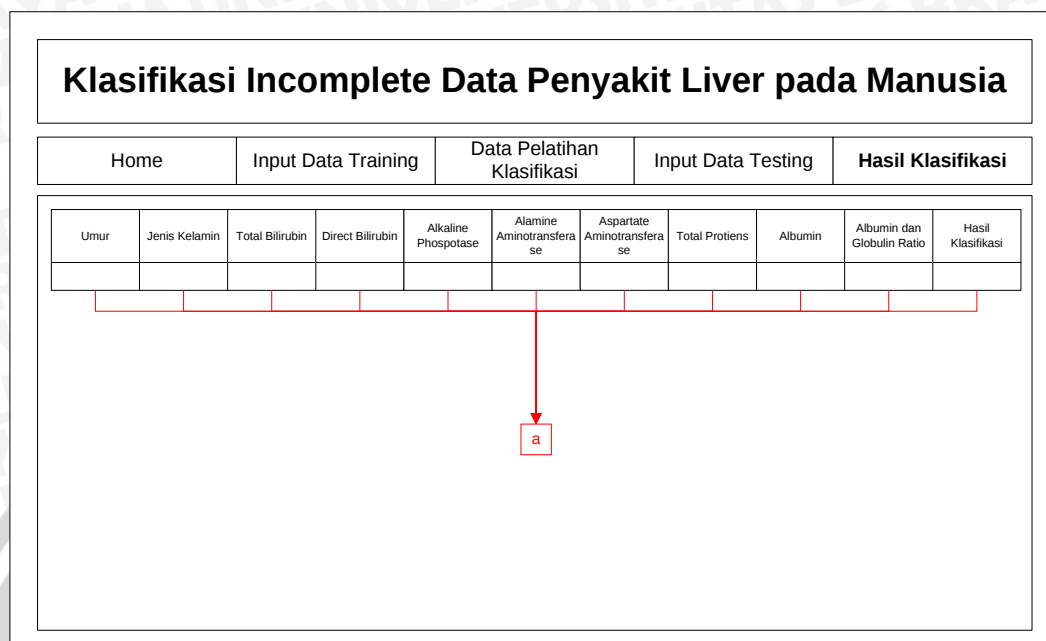
**Gambar 3.20 Perancangan Antarmuka Halaman Memasukkan Data Uji**

Sumber: Perancangan

Halaman memasukkan data uji digunakan untuk memasukkan data uji baik secara manual atau berbentuk *file*. Format dokumen yang dapat di masukkan antara lain xls dan sql. Gambar 3.20 memiliki keterangan sebagai berikut :

- a. *Field* untuk memasukkan data berupa *file*.
- b. Tombol *browse* untuk memilih *file* data yang akan dimasukkan.
- c. Field untuk memasukkan data secara manual
- d. Tombol proses untuk menjalankan sistem.

### 5. Perancangan Antarmuka pengguna halaman hasil klasifikasi



**Gambar 3.21 Perancangan Antarmuka Halaman Hasil Klasifikasi**

Sumber: Perancangan

Halaman hasil diakses setelah pengguna memasukkan data uji. Halaman ini berfungsi untuk menampilkan hasil klasifikasi. Gambar 3.21 memiliki keterangan sebagai berikut :

- a. Kolom hasil klasifikasi dari data uji

### 3.2.3 Perhitungan Manual

Penghitungan manual berfungsi untuk memberikan gambaran umum perancangan sistem yang akan dibangun. Sistem klaifikasi data yang dibangun dalam penelitian ini menggunakan metode algoritma *Voting Feature Interval-5* (VFI-5). Contoh data yang akan digunakan dalam perhitungan manual dapat dilihat pada Tabel 3.5:

Tabel 3.5 Ilustrasi Data Pelatihan

| Umur | Jk            | Tb   | Db  | Alkphos | Sgpt | Sgot | Tp  | Alb | Ag   | Kelas |
|------|---------------|------|-----|---------|------|------|-----|-----|------|-------|
| 84   | <i>Female</i> | 0,7  | 0,2 | 188     | 13   | 21   | 6   | 3,2 | 11   | 0     |
| 57   | <i>Male</i>   | 4    | 1,9 | 190     |      | 111  | 5,2 | 1,5 | 0,4  | 1     |
| 52   | <i>Male</i>   | 0,9  | 0,2 | 156     | 35   |      | 4,9 | 2,9 | 1,4  | 1     |
| 57   | <i>Male</i>   | 1    | 0,3 | 187     | 19   | 23   | 5,2 | 2,9 | 1,2  | 0     |
| 38   |               | 2,6  | 1,2 | 410     | 59   | 57   | 5,6 | 3   | 0,8  | 0     |
| 38   | <i>Female</i> | 2,6  | 1,2 | 410     | 59   | 57   | 5,6 | 3   | 0,8  | 0     |
| 30   | <i>Male</i>   |      | 0,4 | 482     | 102  | 80   | 6,9 | 3,3 | 0,9  | 1     |
| 17   | <i>Female</i> | 0,7  | 0,2 | 145     | 18   |      | 7,2 | 3,9 | 1,18 | 0     |
| 46   | <i>Female</i> | 14,2 | 7,8 | 374     | 38   | 77   | 4,3 | 2   | 0,8  | 1     |
| 48   | <i>Male</i>   |      | 0,6 | 263     | 38   | 66   | 5,8 | 2,2 | 0,61 | 1     |

Sumber: Perancangan

Keterangan:

- Jk : Jenis Kelamin
- Tb : Total bilirubin
- Db : Direct bilirubin
- Alkphos : Alkaline phosphatase
- Sgpt : Alanine Aminotransferase
- Sgot : Aspartate Aminotransferase
- Tp : Total proteins
- Alb : Albumin
- Ag : Albumin dan Globulin ratio
- Kelas : Kelas Penyakit, 1 = Terjangkit Penyakit, 0 = Tidak Terjangkit Penyakit

Contoh manualisasi VFI-5 pada kasus klasifikasi data pada beberapa data adalah sebagai berikut:

### 3.2.3.1 Tahap Preprocessing

Tahap pertama yang dilakukan penggantian nilai data yang tidak lengkap. Pada proses penggantian data yang hilang (*incomplete data*) data numerik diganti dengan *mean* dari data, sedangkan data nominal diganti dengan modus. Tabel 3.6 merupakan contoh hasil dari penggantian nilai data yang tidak lengkap.

Salah satu contoh pengantian nilai data numerik pada “*Alamine aminotransferase*” (sgpt). Data *Alamine aminotransferase* yang hilang diganti dengan *mean* dari seluruh data *Alamine aminotransferase*. Berikut perhitungannya:

$$\text{mean} = \frac{13 + 35 + 19 + 59 + 59 + 102 + 18 + 38 + 38}{9} = 42,33 = 42$$

Jadi data *Alamine aminotransferase* akan diganti dengan nilai 42 sebagai hasil dari *mean* seluruh data *Alamine aminotransferase*. Sedangkan, salah satu contoh pengantian nilai data nominal pada “Jenis Kelamin”. Data Jenis Kelamin yang hilang diganti dengan modus dari seluruh data Jenis Kelamin. Berikut perhitungannya:

Jumlah data jenis kelamin yang bernilai “*Male*” sebanyak 5, sedangkan jumlah data jenis kelamin yang bernilai “*Female*” sebanyak 4. Jadi nilai data yang diganti dalam Jenis Kelamin adalah “*Male*”.

**Tabel 3.6 Ilustrasi Hasil Preprocessing Klasifikasi**

| Umur | Jk     | Tb   | Db  | Alkphos | Sgpt | Sgot | Tp  | Alb | Ag  | Kelas |
|------|--------|------|-----|---------|------|------|-----|-----|-----|-------|
| 84   | Female | 0,7  | 0,2 | 188     | 13   | 21   | 6   | 3,2 | 11  | 0     |
| 57   | Male   | 4    | 1,9 | 190     | 42   | 111  | 5,2 | 1,5 | 0,4 | 1     |
| 52   | Male   | 0,9  | 0,2 | 156     | 35   | 62   | 4,9 | 2,9 | 1,4 | 1     |
| 57   | Male   | 1    | 0,3 | 187     | 19   | 23   | 5,2 | 2,9 | 1,2 | 0     |
| 38   | Male   | 2,6  | 1,2 | 410     | 59   | 57   | 5,6 | 3   | 0,8 | 0     |
| 38   | Female | 2,6  | 1,2 | 410     | 59   | 57   | 5,6 | 3   | 0,8 | 0     |
| 30   | Male   | 3,34 | 0,4 | 482     | 102  | 80   | 6,9 | 3,3 | 0,9 | 1     |

|    |        |      |     |     |    |    |     |     |      |   |
|----|--------|------|-----|-----|----|----|-----|-----|------|---|
| 17 | Female | 0,7  | 0,2 | 145 | 18 | 62 | 7,2 | 3,9 | 1,18 | 0 |
| 46 | Female | 14,2 | 7,8 | 374 | 38 | 77 | 4,3 | 2   | 0,8  | 1 |
| 48 | Male   | 3,34 | 0,6 | 263 | 38 | 66 | 5,8 | 2,2 | 0,61 | 1 |

Sumber: Perancangan

Keterangan:

- Jk : Jenis Kelamin
- Tb : Total bilirubin
- Db : Direct bilirubin
- Alkphos : Alkaline phosphotase
- Sgpt : Alamine Transferase
- Sgot : Aspartate Transferase
- Tp : Total protiens
- Alb : Albumin
- Ag : Albumin dan Globulin ratio
- Kelas :Kelas Penyakit, 1 = Terjangkit Penyakit, 0 = Tidak Terjangkit Penyakit

### 3.2.3.2 Pelatihan Klasifikasi

Pelatihan klasifikasi dilakukan setelah proses praprocess selesai dilakukan. Proses ini adalah proses pemberian *interval class vote* pada setiap *feature* yang telah melewati proses *preprocessing*. Berikut adalah contoh pemberian *interval class vote* pada *feature Direct bilirubin (db)*:

- Tentukan *endpoint* dari *Direct bilirubin (db)* dan mengurutkannya.

Min *direct bilirubin* untuk kelas tidak terjangkit penyakit = 0,2

Min *direct bilirubin* untuk kelas terjangkit penyakit = 0,2

Max *direct bilirubin* untuk kelas tidak terjangkit penyakit = 1,2

Max *direct bilirubin* untuk kelas terjangkit penyakit = 7,8

*Endpoint* yang terbentuk adalah 0,2; 1,2; dan 7,8

- Tentukan *point interval* dan *range interval*

*Point interval* didapatkan dari *endpoint* sedangkan *range interval* didapatkan dari antara *endpoint* dengan *endpoint* berikutnya. *Interval* yang terbentuk adalah sebagai berikut:

- <0,2
- 0,2
- Antara 0,2 dengan 1,2
- 1,2
- Antara 1,2 dengan 7,8
- 7,8
- >7,8

- Hitung *interval vote* dari *feature Direct bilirubin* (db), contoh untuk *point interval* 0,2. Berdasarkan Persamaan (2-1), maka perhitungan internal vote dapat dilihat sebagai berikut:

$$interval\_vote[f, i, c] = \frac{interval\_count[f, i, c]}{class\_count[c]} \quad (2-1)$$

$$interval\_vote[direct\ bilirubin, 0.2, 0] = \frac{interval\_count[direct\ bilirubin, 0.2, 0]}{class\_count[0]}$$

$$interval\_vote[direct\ bilirubin, 0.2, 0] = \frac{2}{5} = 0.4$$

$$interval\_vote[direct\ bilirubin, 0.2, 1] = \frac{interval\_count[direct\ bilirubin, 0.2, 1]}{class\_count[1]}$$

$$interval\_vote[direct\ bilirubin, 0.2, 1] = \frac{1}{5} = 0.2$$

- Normalisasi *interval vote*. Berdasarkan Persamaan(2-2) normalisasi dapat dihitung sebagai berikut:

$$\text{interval\_class\_vote}[f, i, c] = \frac{\text{interval\_vote}[f, i, c]}{\text{count\_interval\_vote}[f, i]} \quad (2-2)$$

$$\begin{aligned} &\text{interval\_class\_vote}[\text{direct bilirubin}, 0.2, 0] \\ &= \frac{\text{interval\_vote}[\text{direct bilirubin}, 0.2, 0]}{\text{count\_interval\_vote}[\text{direct bilirubin}, 0.2]} \end{aligned}$$

$$\text{interval\_class\_vote}[\text{direct bilirubin}, 0.2, 0] = \frac{0.4}{0.6} = 0.6667$$

$$\begin{aligned} &\text{interval\_class\_vote}[\text{direct bilirubin}, 0.2, 1] \\ &= \frac{\text{interval\_vote}[\text{direct bilirubin}, 0.2, 1]}{\text{count\_interval\_vote}[\text{direct bilirubin}, 0.2]} \end{aligned}$$

$$\text{interval\_class\_vote}[\text{direct bilirubin}, 0.2, 1] = \frac{0.2}{0.6} = 0.3333$$

Perhitungan pemberian bobot seperti diatas dilakukan pada setiap *interval* pada setiap *feature*. Tabel 3.7 merupakan ilustrasi hasil dari pelatihan klasifikasi.

**Tabel 3.7 Ilustrasi Hasil Pelatihan Klasifikasi**

| Feature | Interval         | Kelas | Interval_class_vote |
|---------|------------------|-------|---------------------|
| Umur    | <17              | 0     | 0                   |
|         |                  | 1     | 0                   |
|         | 17               | 0     | 1                   |
|         |                  | 1     | 0                   |
|         | Antara 17 dan 30 | 0     | 0                   |
|         |                  | 1     | 0                   |
|         | 30               | 0     | 0                   |
|         |                  | 1     | 1                   |
|         | Antara 30 dan 57 | 0     | 0.4                 |
|         |                  | 1     | 0.6                 |
|         | 57               | 0     | 0.5                 |
|         |                  | 1     | 0.5                 |
|         | Antara 57 dan 84 | 0     | 0                   |
|         |                  | 1     | 0                   |
|         | 84               | 0     | 1                   |
|         |                  | 1     | 0                   |
|         | >84              | 0     | 0                   |
|         |                  | 1     | 0                   |

|                      |                     |   |        |
|----------------------|---------------------|---|--------|
| Jenis Kelamin        | 0                   | 0 | 0.75   |
|                      |                     | 1 | 0.25   |
|                      | 1                   | 0 | 0.3333 |
|                      |                     | 1 | 0.6667 |
| Total bilirubin      | <0.7                | 0 | 0      |
|                      |                     | 1 | 0      |
|                      | 0.7                 | 0 | 1      |
|                      |                     | 1 | 0      |
|                      | Antara 0.7 dan 0.9  | 0 | 0      |
|                      |                     | 1 | 0      |
|                      | 0.9                 | 0 | 0      |
|                      |                     | 1 | 1      |
|                      | Antara 0.9 dan 2.6  | 0 | 1      |
|                      |                     | 1 | 0      |
|                      | 2.6                 | 0 | 1      |
|                      |                     | 1 | 0      |
|                      | Antara 2.6 dan 14.2 | 0 | 0      |
|                      |                     | 1 | 1      |
|                      | 14.2                | 0 | 0      |
|                      |                     | 1 | 1      |
|                      | >14.2               | 0 | 0      |
|                      |                     | 1 | 0      |
| Direct bilirubin     | <0.2                | 0 | 0      |
|                      |                     | 1 | 0      |
|                      | 0.2                 | 0 | 0.6667 |
|                      |                     | 1 | 0.3333 |
|                      | Antara 0.2 dan 1.2  | 0 | 0.3333 |
|                      |                     | 1 | 0.6667 |
|                      | 1.2                 | 0 | 1      |
|                      |                     | 1 | 0      |
|                      | Antara 1.2 dan 7.8  | 0 | 0      |
|                      |                     | 1 | 1      |
|                      | 7.8                 | 0 | 0      |
|                      |                     | 1 | 1      |
|                      | >7.8                | 0 | 0      |
|                      |                     | 1 | 0      |
| Alkaline phosphotase | <145                | 0 | 0      |
|                      |                     | 1 | 0      |
|                      | 145                 | 0 | 1      |
|                      |                     | 1 | 0      |
|                      | Antara 145 dan 156  | 0 | 0      |
|                      |                     | 1 | 0      |
|                      | 156                 | 0 | 0      |
|                      |                     | 1 | 1      |
|                      | Antara 156 dan 410  | 0 | 0.4    |
|                      |                     | 1 | 0.6    |
|                      | 410                 | 0 | 1      |
|                      |                     | 1 | 0      |
|                      | Antara 410 dan 482  | 0 | 0      |
|                      |                     | 0 | 0      |

|                                   |                    |   |      |
|-----------------------------------|--------------------|---|------|
|                                   |                    | 1 | 0    |
|                                   |                    | 0 | 0    |
| 482                               |                    | 1 | 1    |
|                                   |                    | 0 | 0    |
| >482                              |                    | 1 | 0    |
|                                   |                    | 0 | 0    |
| <i>Alamine aminotransferase</i>   | <13                | 0 | 0    |
|                                   |                    | 1 | 0    |
|                                   | 13                 | 0 | 1    |
|                                   |                    | 1 | 0    |
|                                   | Antara 13 dan 35   | 0 | 1    |
|                                   |                    | 1 | 0    |
|                                   | 35                 | 0 | 0    |
|                                   |                    | 1 | 1    |
|                                   | Antara 35 dan 59   | 0 | 0    |
|                                   |                    | 1 | 1    |
|                                   | 59                 | 0 | 1    |
|                                   |                    | 1 | 0    |
|                                   | Antara 59 dan 102  | 0 | 0    |
|                                   |                    | 1 | 0    |
|                                   | 102                | 0 | 0    |
|                                   |                    | 1 | 1    |
|                                   | >102               | 0 | 0    |
|                                   |                    | 1 | 0    |
| <i>Aspartate aminotransferase</i> | <21                | 0 | 0    |
|                                   |                    | 1 | 0    |
|                                   | 21                 | 0 | 1    |
|                                   |                    | 1 | 0    |
|                                   | Antara 21 dan 62   | 0 | 1    |
|                                   |                    | 1 | 0    |
|                                   | 62                 | 0 | 0.5  |
|                                   |                    | 1 | 0.5  |
|                                   | Antara 62 dan 111  | 0 | 0    |
|                                   |                    | 1 | 1    |
|                                   | 111                | 0 | 0    |
|                                   |                    | 1 | 1    |
|                                   | >111               | 0 | 0    |
|                                   |                    | 1 | 0    |
| <i>Total protiens</i>             | <4.3               | 0 | 0    |
|                                   |                    | 1 | 0    |
|                                   | 4.3                | 0 | 0    |
|                                   |                    | 1 | 1    |
|                                   | Antara 4.3 dan 5.2 | 0 | 0    |
|                                   |                    | 1 | 1    |
|                                   | 5.2                | 0 | 0.5  |
|                                   |                    | 1 | 0.5  |
|                                   | Antara 5.2 dan 6.9 | 0 | 0.75 |
|                                   |                    | 1 | 0.25 |
|                                   | 6.9                | 0 | 0    |
|                                   |                    | 1 | 1    |

|                            |                    |   |        |
|----------------------------|--------------------|---|--------|
| Albumin                    | Antara 6.9 dan 7.2 | 0 | 0      |
|                            |                    | 1 | 0      |
|                            | 7.2                | 0 | 1      |
|                            |                    | 1 | 0      |
|                            | >7.2               | 0 | 0      |
|                            |                    | 1 | 0      |
|                            | <1.5               | 0 | 0      |
|                            |                    | 1 | 0      |
|                            | 1.5                | 0 | 0      |
|                            |                    | 1 | 1      |
|                            | Antara 1.5 dan 2.9 | 0 | 0      |
|                            |                    | 1 | 1      |
|                            | 2.9                | 0 | 0.5    |
|                            |                    | 1 | 0.5    |
|                            | Antara 2.9 dan 3.3 | 0 | 0      |
|                            |                    | 1 | 1      |
|                            | 3.3                | 0 | 0      |
|                            |                    | 1 | 1      |
| Albumin dan Globulin ratio | Antara 3.3 dan 3.9 | 0 | 0      |
|                            |                    | 1 | 0      |
|                            | 3.9                | 0 | 1      |
|                            |                    | 1 | 0      |
|                            | >3.9               | 0 | 0      |
|                            |                    | 1 | 0      |
|                            | <0.4               | 0 | 0      |
|                            |                    | 1 | 0      |
|                            | 0.4                | 0 | 0      |
|                            |                    | 1 | 1      |
|                            | Antara 0.4 dan 0.8 | 0 | 0      |
|                            |                    | 1 | 1      |
|                            | 0.8                | 0 | 0.6667 |
|                            |                    | 1 | 0.3333 |
|                            | Antara 0.8 dan 1.2 | 0 | 0.3333 |
|                            |                    | 1 | 0.6667 |
|                            | 1.2                | 0 | 1      |
|                            |                    | 1 | 0      |
|                            | Antara 1.2 dan 1.4 | 0 | 0      |
|                            |                    | 1 | 0      |
|                            | 1.4                | 0 | 0      |
|                            |                    | 1 | 1      |
|                            | >1.4               | 0 | 0      |
|                            |                    | 1 | 0      |

Sumber: Perancangan

### 3.2.3.3 Prediksi Klasifikasi

Proses ini merupakan proses penentuan hasil klasifikasi dengan cara menentukan jumlah terbesar dari *interval class vote* dari tiap kelas dari tiap *feature*. Berikut adalah contoh perhitungan dari prediksi data:

- Menjumlahkan seluruh *interval class vote* dari tiap kelas dengan cara mencocokkan *interval* dengan nilai data tiap *feature*.

Contoh :

Diketahui:

- Umur = 75
- Jenis kelamin = *Male*
- *Total bilirubin* = 0.9
- *Direct bilirubin* = 0.2
- *Alkaline phosphotase* = 162
- *Almaline aminotransferase* = 25
- *Aspartate aminotransferase* = 20
- *Total protiens* = 6.9
- *Albumin* = 3.7
- *Albumin dan globulin ratio* = 1.1

Maka dengan mencocokkan nilai *feature* di atas dengan *interval* yang sudah ada, akan didapatkan *interval class vote* yang nantinya akan ditambahkan seperti perhitungan berikut:

*vote kelas0*

$$= 0 + 0.3333 + 0 + 0.6667 + 0.4 + 1 + 0 + 0 + 0 + 0.6667 = 3.0667$$

*vote kelas1*

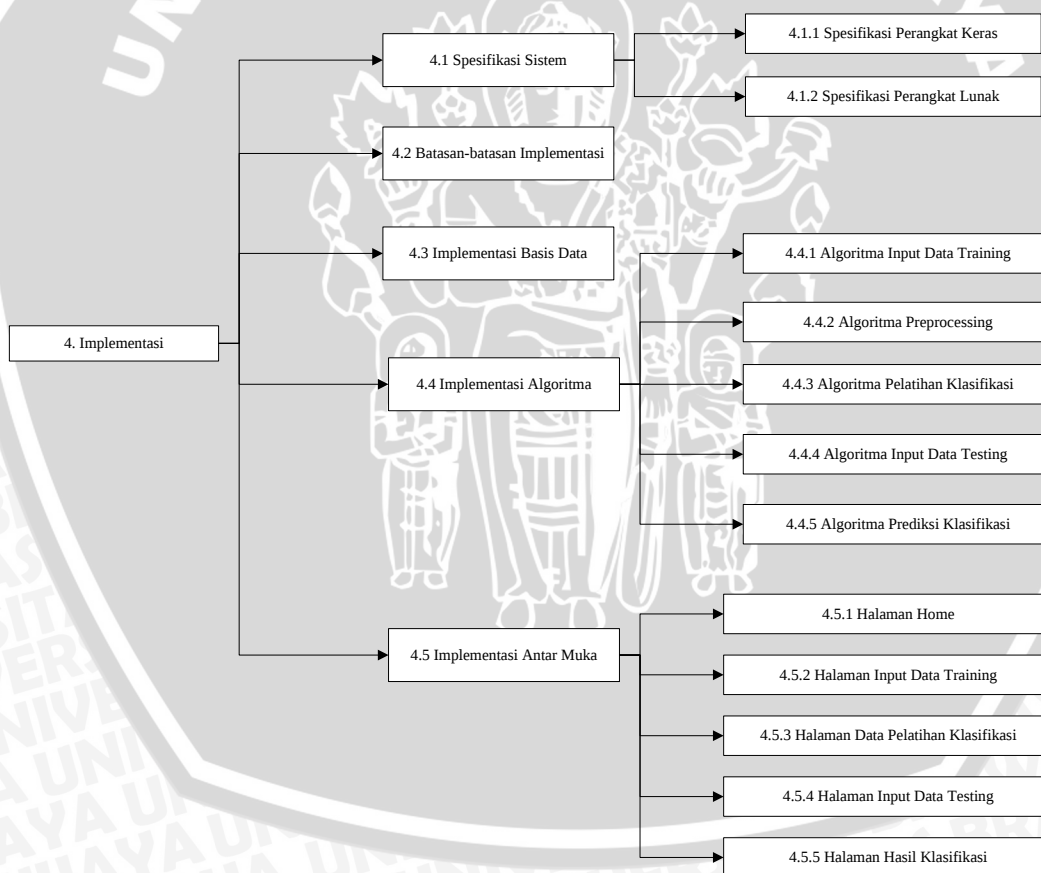
$$= 0 + 0.6667 + 1 + 0.3333 + 0.6 + 0 + 0 + 1 + 0 \\ + 0.3333 = 3.933$$

Jadi hasil klasifikasi yang didapat adalah kelas 1 yaitu terjangkit penyakit.



## BAB IV IMPLEMENTASI

Bab ini membahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak yang dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi algoritma, dan implementasi antar muka. Tahap-tahap pembahasan implementasi yang dikerjakan digambarkan pada Gambar 4.1 berikut ini.



**Gambar 4.1 Pohon Implementasi**

Sumber: Implementasi

#### 4.1 Spesifikasi Sistem

Hasil analisis kebutuhan yang telah diuraikan pada bab 3 menjadi acuan untuk melakukan implementasi menjadi sebuah sistem yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak

##### 4.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras meliputi spesifikasi prosesor, memori, hardisk, *mother board*, dan kartu grafis dijelaskan pada Tabel 4.1.

**Tabel 4.1 Spesifikasi Perangkat Keras Komputer**

| Nama Komponen       | Spesifikasi                                            |
|---------------------|--------------------------------------------------------|
| Prosesor            | Intel(R) Atom(TM) CPU N550 @ 1.50GHz (4 CPUs), ~1.5GHz |
| Memori (RAM)        | 1024MB RAM                                             |
| Hardisk             | WD kapasitas 298 GB                                    |
| <i>Mother Board</i> | Lenovo Intel Motherboard                               |
| Kartu Grafis        | Intel(R) Graphics Media Accelerator 3150               |

Sumber: Implementasi

##### 4.1.2 Spesifikasi Perangkat Lunak

Pengembangan subsistem menggunakan perangkat lunak dengan spesifikasi yang dijelaskan pada Tabel 4.2

**Tabel 4.2 Spesifikasi Perangkat Lunak Komputer**

| Nama Komponen                 | Spesifikasi                                                          |
|-------------------------------|----------------------------------------------------------------------|
| Sistem operasi                | Windows 7 Starter 32 bit.                                            |
| <i>Tools</i> pemrograman      | Notepad++, XAMPP 3.1.0, Adobe Dreamweaver CS3, dan Internet Explorer |
| <i>Database</i> dan antarmuka | MySQL dan PHPmyAdmin                                                 |

Sumber: Implementasi

## 4.2 Batasan-Batasan Implementasi

Batasan-batasan dalam mengimplementasikan sistem adalah sebagai berikut:

1. Pembuatan aplikasi sistem klasifikasi *incomplete data* penyakit liver pada manusia ini dikerjakan dengan bahasa pemrograman *PHP* dan *database MySQL*.
2. Data yang dapat diproses oleh sistem adalah data penyakit liver pada manusia.
3. Metode yang digunakan pada sistem adalah *Voting Feature Interval-5*.
4. Aplikasi hanya akan memproses data dalam bentuk format *xls* atau *sql*
5. Data yang akan di klasifikasi disadur dari data *Indian Liver Patient Dataset* dari *UCI Machine Learning*. (url: [http://archive.ics.uci.edu/ml/datasets/ILPD+\(Indian+Liver+Patient+Datase+t\)](http://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Datase+t)))

## 4.3 Implementasi Basis Data

Implementasi penyimpanan data dilakukan dengan *database management system MySQL*. Hasil implementasi penyimpanan data ini berupa *script – script SQL*. Gambar 4.2 menggambarkan implementasi perancangan basis data dari Perangkat lunak Klasifikasi *Incomplete Data* Penyakit Liver pada Manusia

|                                                                                                                                                                                                             |                                                                                                                                                                                                          |                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>skripsi_f_alkphos</b><br>id : int(10)<br>f_id : varchar(20)<br>endpoint : varchar(20)<br>kelas : varchar(20)<br>jumlah : varchar(20)<br>interval_vote : varchar(20)<br>interval_class_vote : varchar(20) | <b>skripsi_f_umur</b><br>id : int(10)<br>f_id : varchar(20)<br>endpoint : varchar(20)<br>kelas : varchar(20)<br>jumlah : varchar(20)<br>interval_vote : varchar(20)<br>interval_class_vote : varchar(20) | <b>skripsi_f_db</b><br>id : int(10)<br>f_id : varchar(20)<br>endpoint : varchar(20)<br>kelas : varchar(20)<br>jumlah : varchar(20)<br>interval_vote : varchar(20)<br>interval_class_vote : varchar(20)            | <b>skripsi_f_sgpt</b><br>id : int(10)<br>f_id : varchar(20)<br>endpoint : varchar(20)<br>kelas : varchar(20)<br>jumlah : varchar(20)<br>interval_vote : varchar(20)<br>interval_class_vote : varchar(20)                                                                                                                    |
| <b>skripsi_f_tp</b><br>id : int(10)<br>f_id : varchar(20)<br>endpoint : varchar(20)<br>kelas : varchar(20)<br>jumlah : varchar(20)<br>interval_vote : varchar(20)<br>interval_class_vote : varchar(20)      | <b>skripsi_f_alb</b><br>id : int(10)<br>f_id : varchar(20)<br>endpoint : varchar(20)<br>kelas : varchar(20)<br>jumlah : varchar(20)<br>interval_vote : varchar(20)<br>interval_class_vote : varchar(20)  | <b>skripsi_f_sgpt</b><br>id : int(10)<br>f_id : varchar(20)<br>endpoint : varchar(20)<br>kelas : varchar(20)<br>jumlah : varchar(20)<br>interval_vote : varchar(20)<br>interval_class_vote : varchar(20)          | <b>skripsi_data_testing</b><br>id_data : int(5)<br>umur : int(5)<br>jenis_kelamin : varchar(11)<br>tb : decimal(10,2)<br>db : decimal(10,2)<br>alkphos : int(5)<br>sgpt : int(5)<br>tp : int(5)<br>tb : decimal(10,2)<br>tp : decimal(10,2)<br>alb : decimal(10,2)<br>ag : decimal(10,2)<br>hasil_klasifikasi : varchar(50) |
| <b>skripsi_f_ag</b><br>id : int(10)<br>f_id : varchar(20)<br>endpoint : varchar(20)<br>kelas : varchar(20)<br>jumlah : varchar(20)<br>interval_vote : varchar(20)<br>interval_class_vote : varchar(20)      | <b>skripsi_f_tb</b><br>id : int(10)<br>f_id : varchar(20)<br>endpoint : varchar(20)<br>kelas : varchar(20)<br>jumlah : varchar(20)<br>interval_vote : varchar(20)<br>interval_class_vote : varchar(20)   | <b>skripsi_f_jenis_kelamin</b><br>id : int(10)<br>f_id : varchar(20)<br>endpoint : varchar(20)<br>kelas : varchar(20)<br>jumlah : varchar(20)<br>interval_vote : varchar(20)<br>interval_class_vote : varchar(20) | <b>skripsi_data_liver</b><br>id_data : int(5)<br>umur : int(5)<br>jenis_kelamin : varchar(11)<br>tb : decimal(10,2)<br>db : decimal(10,2)<br>alkphos : int(5)<br>sgpt : int(5)<br>tp : int(5)<br>tb : decimal(10,2)<br>tp : decimal(10,2)<br>alb : decimal(10,2)<br>ag : decimal(10,2)<br>kelas : varchar(50)               |

Gambar 4.2 Implementasi Basis Data

Sumber: Implementasi

#### 4.4 Implementasi Algoritma

Aplikasi klasifikasi *incomplete data* mempunyai beberapa proses utama yang terbagi dalam beberapa fungsi. Pada penulisan skripsi ini hanya dicantumkan algoritma dari beberapa proses saja sehingga tidak semua algoritma akan dicantumkan. Algoritma proses yang dicantumkan antara lain adalah proses memasukkan data pelatihan, *preprocessing* pelatihan, pelatihan klasifikasi, memasukkan data uji, *preprocessing* prediksi dan prediksi klasifikasi. Implementasi algoritma ini akan direpresentasikan dalam bentuk *code* dengan bahasa pemrograman PHP.

##### 4.4.1 Implementasi Algoritma Memasukkan Data Pelatihan

Operasi pada algoritma memasukkan data pelatihan bertujuan untuk menangkap masukan data pelatihan pengguna. Pengguna dapat memasukkan data dengan cara memasukkan data dalam satu *file* atau memasukkan data secara manual pada *form* yang telah disediakan. Untuk proses memasukkan *file*, data masukan pengguna akan diseleksi oleh sistem, sehingga hanya format *file* tertentu saja yang dapat diproses. Gambar 4.3 merupakan cuplikan *code* untuk algoritma

memasukkan *file* yang mengimplementasikan perancangan algoritma memasukkan pada sub Bab 3.2.2.4

```

1.  <?php
2.  include "excel_reader2.php";
3.  if ($_FILES['userfile']['type']=="application/vnd.ms-excel"){
4.      include("koneksi_db.php");
5.      $data = new Spreadsheet_Excel_Reader($_FILES
6.  ['userfile']['tmp_name'], false);
7.      $baris = $data->rowcount($sheet_index=0);
8.      for ($i=1; $i <= $baris; $i++){
9.          $umur = $data->val($i, 1);
10.         $umur=check_value($umur);
11.         $jenis_kelamin = $data->val($i, 2);
12.         $jenis_kelamin=check_value($jenis_kelamin);
13.         $tb = $data->val($i, 3);
14.         $tb=check_value($tb);
15.         $db = $data->val($i, 4);
16.         $db=check_value($db);
17.         $alkphos = $data->val($i, 5);
18.         $alkphos=check_value($alkphos);
19.         $sgpt = $data->val($i, 6);
20.         $sgpt=check_value($sgpt);
21.         $sgot = $data->val($i, 7);
22.         $sgot=check_value($sgot);
23.         $stp = $data->val($i, 8);
24.         $stp=check_value($stp);
25.         $alb = $data->val($i, 9);
26.         $alb=check_value($alb);
27.         $ag = $data->val($i, 10);
28.         $ag=check_value($ag);
29.         $kelas = $data->val($i, 11);
30.         $kelas=check_value($kelas);
31.         $hasil = mysql_query("INSERT INTO data_liver
32.         (umur,jenis_kelamin,tb,db,alkphos,sgpt,sgot,tp,alb,ag,kelas)
33.         VALUES ($umur,$jenis_kelamin,$tb,$db,$alkphos,$sgpt,$sgot,$stp,$
34.         alb,$ag,$kelas)");
35.     }
36.     else if($_FILES['userfile']['type']=="text/plain"){
37.         $dbconn = mysql_connect('localhost','root','');
38.         mysql_select_db('skripsi',$dbconn);
39.         $file = $_FILES['userfile']['name'];
40.         if($fp = file_get_contents($file)) {
41.             $var_array = explode(';', $fp);
42.             foreach($var_array as $value) {
43.                 mysql_query($value.';', $dbconn);
44.             }
45.         }
46.     }
47.     else{
48.         header ("Location: form_input_data_training.php");
49.     }
50. }
51. >

```

**Gambar 4.3 Implementasi Algoritma Memasukkan Data Pelatihan**

Sumber: Implementasi

Penjelasan implementasi algoritma memasukkan data pelatihan pada Gambar 4.3 yaitu:

1. Baris 2 berfungsi untuk memanggil library upload untuk *file* xls.
2. Baris 3 berfungsi untuk memeriksa tipe *file* dari dokumen, jika *file* berbentuk xls maka data akan dimasukkan ke dalam *database* sesuai dengan baris 4-32.
3. Baris 33 berfungsi untuk memeriksa tipe *file* dari dokumen, jika *file* berbentuk sql maka data akan dimasukkan ke dalam *database* sesuai dengan baris 34-43.
4. Baris 44-46 berfungsi untuk mengembalikan ke halaman memasukkan data pelatihan, karena jenis dokumen yang di masukkan tidak sesuai dengan format yang di tentukan.

#### 4.4.2 Implementasi Algoritma *Preprocessing* Pelatihan Klasifikasi

Operasi pada algoritma *preprocessing* pelatihan klasifikasi bertujuan untuk melengkapi data yang tidak lengkap sesuai dengan bentuknya. Data nominal akan diganti dengan modus, sedangkan data numerik akan diganti dengan *mean*. Gambar 4.4 merupakan cuplikan *code* untuk algoritma *preprocessing* yang mengimplementasikan perancangan algoritma *preprocessing* pelatihan klasifikasi pada sub bab 3.2.2.4

```

1. <?php
2.     include('koneksi_db.php');
3.     $array = array("umur", "jenis_kelamin", "tb", "db", "alkphos",
4.     "sgpt", "sgot", "tp", "alb", "ag", "kelas");
5.     for ($i=0; $i <= 10; $i++){
6.         $result= mysql_query("SELECT id_data FROM `data_liver`
7.         WHERE {$array[$i]} IS NULL");
8.         if($result){
9.             $a=0;
10.            while ($row = mysql_fetch_array($result)){
11.                if ($i==1 || $i==10){
12.                    $result1=mysql_query("SELECT
13.                    {$array[$i]}, count({$array[$i]}) as jumlah FROM data_liver GROUP
14.                    BY kelas ORDER BY jumlah DESC LIMIT 1");
15.                }
16.            }
17.        }
18.    }

```

```

12.         else{
13.             $result1= mysql_query("SELECT
avg({$array[$i]}) from data_liver");
14.         }
15.         while ($row1 = mysql_fetch_array($result1)){
16.             $result2=mysql_query("update data_liver
set {$array[$i]}={$row1[0]} where id_data={$row[$a]}");
17.         }
18.         unset ($row1);
19.     }
20.     $a++;
21.     unset ($row);
22. }
23. }
24. ?>

```

**Gambar 4.4 Implementasi Algoritma *Preprocessing* Pelatihan Klasifikasi**

Sumber: Implementasi

Penjelasan implementasi algoritma *preprocessing* pelatihan klasifikasi pada Gambar 4.4 yaitu:

1. Baris 2 berfungsi untuk melakukan koneksi ke *database*.
2. Baris 3 berfungsi untuk inisialisasi *feature*.
3. Baris 4-5 berfungsi untuk memeriksa kelengkapan tiap *feature*.
4. Baris 9 berfungsi untuk memeriksa bentuk data yang hilang, jika data berbentuk nominal maka akan dicari modus datanya sesuai dengan baris10, sedangkan jika data berbetuk numerik maka akan dicari *mean* datanya sesuai dengan baris13.
5. Baris 16 berfungsi untuk memperbarui nilai data yang tidak lengkap dengan *mean* atau modus yang sudah di cari sebelumnya.

#### 4.4.3 Implementasi Algoritma Pelatihan Klasifikasi

Operasi pada algoritma pelatihan klasifikasi bertujuan untuk mendapatkan *interval class vote* dari *interval-interval* pada setiap *feature*. Proses pelatihan klasifikasi menggunakan metode *Voting Feature Interval-5*. Gambar 4.5 merupakan cuplikan *code* untuk algoritma pelatihan klasifikasi yang mengimplementasikan perancangan algoritma pelatihan klasifikasi pada sub bab 3.2.2.4

```

1.  <?php
2.  include ("koneksi_db.php");
3.  $array = array("umur", "jenis_kelamin", "tb", "db", "alkphos",
4.  "sgpt", "sgot", "tp", "alb", "ag");
5.  $a=0;
6.  $c=0;
7.  for ($i=0; $i <= 9; $i++){
8.      $result= mysql_query("select distinct max({$array[$i]}) as
9.      endpoint from data_liver where kelas=0 union distinct select
10.     distinct max({$array[$i]}) from data_liver where kelas=1 union
11.     distinct select distinct min({$array[$i]}) from data_liver where
12.     kelas=0 union distinct select distinct min({$array[$i]}) from
13.     data_liver where kelas=1");
14.     while( $row = mysql_fetch_array($result)) {
15.         $EndPoint[] = $row[ "endpoint"];
16.     }
17.     sort($EndPoint);
18.     foreach ($EndPoint as $key => $val) {
19.         $endpoint[] = $val;
20.     }
21.     unset($EndPoint);
22.     $b=count($endpoint);
23.     $a=$b-$c;
24.     $c=$c+$a;
25.     $jumlahendpoint[]=$a;
26. }
27. $b=count($endpoint);
28. $d=0;
29. $result=mysql_query("select count(*) as jumlah from data_liver
30. where kelas=0");
31. while( $row=mysql_fetch_array($result) ) {
32.     $jumlah_kelas[0]=$row["jumlah"];
33. }
34. $result=mysql_query("select count(*) as jumlah from data_liver
35. where kelas=1");
36. while( $row=mysql_fetch_array($result) ) {
37.     $jumlah_kelas[1]=$row["jumlah"];
38. }
39. for ($a=0;$a<$b;$a=$a){
40.     for($i=0; $i<=9; $i++){
41.         $result=mysql_query("truncate f_{$array[$i]}");
42.         for ($c=0;$c<((2*$jumlahendpoint[$i])+1);$c++){
43.             if ($c==0){
44.                 $result=mysql_query("select count(*) as
45.                 jumlah from data_liver where kelas=0 and
46.                 {$array[$i]}<{$endpoint[$a]} union all select count(*) as jumlah
47.                 from data_liver where kelas=1 and {$array[$i]}<{$endpoint[$a]}");
48.                 $qendpoint="< ". $endpoint[$a];
49.             }
50.             else if($c%2==1){
51.                 $result=mysql_query("select count(*) as
52.                 jumlah from data_liver where kelas=0 and
53.                 {$array[$i]}={$endpoint[$a]} union all select count(*) as jumlah
54.                 from data_liver where kelas=1 and {$array[$i]}={$endpoint[$a]}");
55.                 $qendpoint=$endpoint[$a];
56.             }
57.             else if($c%2==0 and $c!=0 and
58.             $c!=(2*$jumlahendpoint[$i])){
59.                 $a++;
60.                 $result=mysql_query("select count(*) as
61.                 jumlah from data_liver where kelas=0 and
62.                 {$array[$i]}>{$endpoint[($a-1)]} and {$array[$i]}<{$endpoint[$a]}
63.                 union all select count(*) as jumlah from data_liver where kelas=1

```

```

45. and {$array[$i]}>{$endpoint[($a-1)]} and
    {$array[$i]}<{$endpoint[$a]}");
46.                                     $qendpoint="between ".$endpoint[($a-
47.                                     )].". and ".$endpoint[$a];
48.                                     }
49.                                     else if($c==(2*$jumlahendpoint[$i])){
50.                                         $a++;
51.                                         $result=mysql_query("select count(*) as
52.                                         jumlah from data_liver where kelas=0 and
53.                                         {$array[$i]}>{$endpoint[$a-1]} union all select count(*) as
54.                                         jumlah from data_liver where kelas=1 and
55.                                         {$array[$i]}>{$endpoint[$a-1]}");
56.                                         $qendpoint="> ".$endpoint[$a-1];
57.                                     }
58.                                     while( $row=mysql_fetch_array($result) ) {
59.                                         $finalendpoint[$d][]=$row["jumlah"];
60.                                     }
61.                                     $interval_vote0=($finalendpoint[$d][0]/$jumlah_kelas[0]);
62.                                     $interval_votel=($finalendpoint[$d][1]/$jumlah_kelas[1]);
63.                                     $jumlah_interval_vote=$interval_vote0+$interval_votel;
64.                                     if ($jumlah_interval_vote!=0){
65.                                         $interval_class_vote0=$interval_vote0/$jumlah_interval_vote
66.                                     ;
67.                                         $interval_class_votel=$interval_votel/$jumlah_interval_vote
68.                                     ;
69.                                     }
70.                                     else {
71.                                         $interval_class_vote0=0;
72.                                         $interval_class_votel=0;
73.                                     }
74.                                     $result=mysql_query("insert into
75.                                     f_{$array[$i]}
76.                                     values ('f'.'.$c.', '$qendpoint', '0', '({$finalendpoint[$d][0]}', '$int
77.                                     erval_vote0', '$interval_class_vote0'), ('f'.'.$c.', '$qendpoint', '1'
78.                                     ,
79.                                     '{$finalendpoint[$d][1]}', '$interval_votel', '$interval_class_vote
80.                                     1')");
81.                                     $d++;
82.                                     }
83.                                 }
84.                            }
85.                    }
86.                }
87.            }
88.        }
89.    }
90.    }
91.    }
92.    }
93.    }
94.    }
95.    }
96.    }
97.    }
98.    }
99.    }
100.    }
101.    }
102.    }
103.    }
104.    }
105.    }
106.    }
107.    }
108.    }
109.    }
110.    }
111.    }
112.    }
113.    }
114.    }
115.    }
116.    }
117.    }
118.    }
119.    }
120.    }
121.    }
122.    }
123.    }
124.    }
125.    }
126.    }
127.    }
128.    }
129.    }
130.    }
131.    }
132.    }
133.    }
134.    }
135.    }
136.    }
137.    }
138.    }
139.    }
140.    }
141.    }
142.    }
143.    }
144.    }
145.    }
146.    }
147.    }
148.    }
149.    }
150.    }
151.    }
152.    }
153.    }
154.    }
155.    }
156.    }
157.    }
158.    }
159.    }
160.    }
161.    }
162.    }
163.    }
164.    }
165.    }
166.    }
167.    }
168.    }
169.    }
170.    }
171.    }
172.    }
173.    }
174.    }
175.    }
176.    }
177.    }
178.    }
179.    }
180.    }
181.    }
182.    }
183.    }
184.    }
185.    }
186.    }
187.    }
188.    }
189.    }
190.    }
191.    }
192.    }
193.    }
194.    }
195.    }
196.    }
197.    }
198.    }
199.    }
200.    }
201.    }
202.    }
203.    }
204.    }
205.    }
206.    }
207.    }
208.    }
209.    }
210.    }
211.    }
212.    }
213.    }
214.    }
215.    }
216.    }
217.    }
218.    }
219.    }
220.    }
221.    }
222.    }
223.    }
224.    }
225.    }
226.    }
227.    }
228.    }
229.    }
230.    }
231.    }
232.    }
233.    }
234.    }
235.    }
236.    }
237.    }
238.    }
239.    }
240.    }
241.    }
242.    }
243.    }
244.    }
245.    }
246.    }
247.    }
248.    }
249.    }
250.    }
251.    }
252.    }
253.    }
254.    }
255.    }
256.    }
257.    }
258.    }
259.    }
260.    }
261.    }
262.    }
263.    }
264.    }
265.    }
266.    }
267.    }
268.    }
269.    }
270.    }
271.    }
272.    }
273.    }
274.    }
275.    }
276.    }
277.    }
278.    }
279.    }
280.    }
281.    }
282.    }
283.    }
284.    }
285.    }
286.    }
287.    }
288.    }
289.    }
290.    }
291.    }
292.    }
293.    }
294.    }
295.    }
296.    }
297.    }
298.    }
299.    }
300.    }
301.    }
302.    }
303.    }
304.    }
305.    }
306.    }
307.    }
308.    }
309.    }
310.    }
311.    }
312.    }
313.    }
314.    }
315.    }
316.    }
317.    }
318.    }
319.    }
320.    }
321.    }
322.    }
323.    }
324.    }
325.    }
326.    }
327.    }
328.    }
329.    }
330.    }
331.    }
332.    }
333.    }
334.    }
335.    }
336.    }
337.    }
338.    }
339.    }
340.    }
341.    }
342.    }
343.    }
344.    }
345.    }
346.    }
347.    }
348.    }
349.    }
350.    }
351.    }
352.    }
353.    }
354.    }
355.    }
356.    }
357.    }
358.    }
359.    }
360.    }
361.    }
362.    }
363.    }
364.    }
365.    }
366.    }
367.    }
368.    }
369.    }
370.    }
371.    }
372.    }
373.    }
374.    }
375.    }
376.    }
377.    }
378.    }
379.    }
380.    }
381.    }
382.    }
383.    }
384.    }
385.    }
386.    }
387.    }
388.    }
389.    }
390.    }
391.    }
392.    }
393.    }
394.    }
395.    }
396.    }
397.    }
398.    }
399.    }
400.    }
401.    }
402.    }
403.    }
404.    }
405.    }
406.    }
407.    }
408.    }
409.    }
410.    }
411.    }
412.    }
413.    }
414.    }
415.    }
416.    }
417.    }
418.    }
419.    }
420.    }
421.    }
422.    }
423.    }
424.    }
425.    }
426.    }
427.    }
428.    }
429.    }
430.    }
431.    }
432.    }
433.    }
434.    }
435.    }
436.    }
437.    }
438.    }
439.    }
440.    }
441.    }
442.    }
443.    }
444.    }
445.    }
446.    }
447.    }
448.    }
449.    }
450.    }
451.    }
452.    }
453.    }
454.    }
455.    }
456.    }
457.    }
458.    }
459.    }
460.    }
461.    }
462.    }
463.    }
464.    }
465.    }
466.    }
467.    }
468.    }
469.    }
470.    }
471.    }
472.    }
473.    }
474.    }
475.    }
476.    }
477.    }
478.    }
479.    }
480.    }
481.    }
482.    }
483.    }
484.    }
485.    }
486.    }
487.    }
488.    }
489.    }
490.    }
491.    }
492.    }
493.    }
494.    }
495.    }
496.    }
497.    }
498.    }
499.    }
500.    }
501.    }
502.    }
503.    }
504.    }
505.    }
506.    }
507.    }
508.    }
509.    }
510.    }
511.    }
512.    }
513.    }
514.    }
515.    }
516.    }
517.    }
518.    }
519.    }
520.    }
521.    }
522.    }
523.    }
524.    }
525.    }
526.    }
527.    }
528.    }
529.    }
530.    }
531.    }
532.    }
533.    }
534.    }
535.    }
536.    }
537.    }
538.    }
539.    }
540.    }
541.    }
542.    }
543.    }
544.    }
545.    }
546.    }
547.    }
548.    }
549.    }
550.    }
551.    }
552.    }
553.    }
554.    }
555.    }
556.    }
557.    }
558.    }
559.    }
560.    }
561.    }
562.    }
563.    }
564.    }
565.    }
566.    }
567.    }
568.    }
569.    }
570.    }
571.    }
572.    }
573.    }
574.    }
575.    }
576.    }
577.    }
578.    }
579.    }
580.    }
581.    }
582.    }
583.    }
584.    }
585.    }
586.    }
587.    }
588.    }
589.    }
590.    }
591.    }
592.    }
593.    }
594.    }
595.    }
596.    }
597.    }
598.    }
599.    }
600.    }
601.    }
602.    }
603.    }
604.    }
605.    }
606.    }
607.    }
608.    }
609.    }
610.    }
611.    }
612.    }
613.    }
614.    }
615.    }
616.    }
617.    }
618.    }
619.    }
620.    }
621.    }
622.    }
623.    }
624.    }
625.    }
626.    }
627.    }
628.    }
629.    }
630.    }
631.    }
632.    }
633.    }
634.    }
635.    }
636.    }
637.    }
638.    }
639.    }
640.    }
641.    }
642.    }
643.    }
644.    }
645.    }
646.    }
647.    }
648.    }
649.    }
650.    }
651.    }
652.    }
653.    }
654.    }
655.    }
656.    }
657.    }
658.    }
659.    }
660.    }
661.    }
662.    }
663.    }
664.    }
665.    }
666.    }
667.    }
668.    }
669.    }
670.    }
671.    }
672.    }
673.    }
674.    }
675.    }
676.    }
677.    }
678.    }
679.    }
680.    }
681.    }
682.    }
683.    }
684.    }
685.    }
686.    }
687.    }
688.    }
689.    }
690.    }
691.    }
692.    }
693.    }
694.    }
695.    }
696.    }
697.    }
698.    }
699.    }
700.    }
701.    }
702.    }
703.    }
704.    }
705.    }
706.    }
707.    }
708.    }
709.    }
710.    }
711.    }
712.    }
713.    }
714.    }
715.    }
716.    }
717.    }
718.    }
719.    }
720.    }
721.    }
722.    }
723.    }
724.    }
725.    }
726.    }
727.    }
728.    }
729.    }
730.    }
731.    }
732.    }
733.    }
734.    }
735.    }
736.    }
737.    }
738.    }
739.    }
740.    }
741.    }
742.    }
743.    }
744.    }
745.    }
746.    }
747.    }
748.    }
749.    }
750.    }
751.    }
752.    }
753.    }
754.    }
755.    }
756.    }
757.    }
758.    }
759.    }
760.    }
761.    }
762.    }
763.    }
764.    }
765.    }
766.    }
767.    }
768.    }
769.    }
770.    }
771.    }
772.    }
773.    }
774.    }
775.    }
776.    }
777.    }
778.    }
779.    }
780.    }
781.    }
782.    }
783.    }
784.    }
785.    }
786.    }
787.    }
788.    }
789.    }
790.    }
791.    }
792.    }
793.    }
794.    }
795.    }
796.    }
797.    }
798.    }
799.    }
800.    }
801.    }
802.    }
803.    }
804.    }
805.    }
806.    }
807.    }
808.    }
809.    }
810.    }
811.    }
812.    }
813.    }
814.    }
815.    }
816.    }
817.    }
818.    }
819.    }
820.    }
821.    }
822.    }
823.    }
824.    }
825.    }
826.    }
827.    }
828.    }
829.    }
830.    }
831.    }
832.    }
833.    }
834.    }
835.    }
836.    }
837.    }
838.    }
839.    }
840.    }
841.    }
842.    }
843.    }
844.    }
845.    }
846.    }
847.    }
848.    }
849.    }
850.    }
851.    }
852.    }
853.    }
854.    }
855.    }
856.    }
857.    }
858.    }
859.    }
860.    }
861.    }
862.    }
863.    }
864.    }
865.    }
866.    }
867.    }
868.    }
869.    }
870.    }
871.    }
872.    }
873.    }
874.    }
875.    }
876.    }
877.    }
878.    }
879.    }
880.    }
881.    }
882.    }
883.    }
884.    }
885.    }
886.    }
887.    }
888.    }
889.    }
890.    }
891.    }
892.    }
893.    }
894.    }
895.    }
896.    }
897.    }
898.    }
899.    }
900.    }
901.    }
902.    }
903.    }
904.    }
905.    }
906.    }
907.    }
908.    }
909.    }
910.    }
911.    }
912.    }
913.    }
914.    }
915.    }
916.    }
917.    }
918.    }
919.    }
920.    }
921.    }
922.    }
923.    }
924.    }
925.    }
926.    }
927.    }
928.    }
929.    }
930.    }
931.    }
932.    }
933.    }
934.    }
935.    }
936.    }
937.    }
938.    }
939.    }
940.    }
941.    }
942.    }
943.    }
944.    }
945.    }
946.    }
947.    }
948.    }
949.    }
950.    }
951.    }
952.    }
953.    }
954.    }
955.    }
956.    }
957.    }
958.    }
959.    }
960.    }
961.    }
962.    }
963.    }
964.    }
965.    }
966.    }
967.    }
968.    }
969.    }
970.    }
971.    }
972.    }
973.    }
974.    }
975.    }
976.    }
977.    }
978.    }
979.    }
980.    }
981.    }
982.    }
983.    }
984.    }
985.    }
986.    }
987.    }
988.    }
989.    }
990.    }
991.    }
992.    }
993.    }
994.    }
995.    }
996.    }
997.    }
998.    }
999.    }
1000.    }

```

**Gambar 4.5 Implementasi Algoritma Pelatihan Klasifikasi**

Sumber: Implementasi

Penjelasan implementasi algoritma pelatihan klasifikasi pada Gambar 4.5 yaitu:

1. Baris 2 berfungsi untuk melakukan koneksi ke *database*.
2. Baris 3 berfungsi untuk inisialisasi *feature*.
3. Baris 7 berfungsi untuk menentukan nilai *endpoint*.
4. Baris 11 berfungsi untuk mengurutkan nilai *endpoint*.

5. Baris 35-52 berfungsi untuk menentukan *point interval* dan *range interval*.
6. Baris 56-57 berfungsi untuk menentukan *interval vote* tiap kelas.
7. Baris 59-66 berfungsi untuk menentukan *interval class vote* tiap kelas.
8. Baris 67 berfungsi untuk memasukkan nilai *interval*, *interval vote* dan *interval class vote* dari tiap *feature* ke dalam *database*.

#### 4.4.4 Implementasi Algoritma Memasukkan Data Uji

Operasi pada algoritma memasukkan data uji bertujuan untuk menangkap masukan data uji dari pengguna yang selanjutnya akan di prediksi hasil klasifikasinya. Pengguna dapat memasukkan data dengan cara memasukkan data dalam satu *file* atau memasukkan data secara manual pada *form* yang telah disediakan. Untuk proses memasukkan *file*, data masukan pengguna akan diseleksi oleh sistem, sehingga hanya format dokumen tertentu saja yang dapat diproses. Data yang masuk harus lengkap. Gambar 4.6 merupakan cuplikan *code* untuk algoritma memasukkan data uji yang mengimplementasikan perancangan algoritma memasukkan pada sub bab 3.2.2.4

```

1. <?php
2. include "excel_reader2.php";
3. include("koneksi_db.php");
4. $result=mysql_query("SELECT SUM(TABLE_ROWS) AS jumlah FROM
   information_schema.Tables WHERE TABLE_SCHEMA = 'skripsi' AND
   TABLE_NAME <> 'data_liver' AND TABLE_NAME <> 'data_testing'");
5. while( $row = mysql_fetch_array($result)) {
6.     $jumlah[] = $row["jumlah"];
7. }
8. $a=0;
9. if ($jumlah[0]<=0){
10.     header ("Location: form_input_data_testing.php");
11. }
12. else if($_FILES['userfile']['type']=="application/vnd.ms-
   excel"){
13.     $data = new
   Spreadsheet_Excel_Reader($_FILES['userfile']['tmp_name'],
   false);
14.     $baris = $data->rowcount($sheet_index=0);
15.     for ($i=1; $i <= $baris; $i++){
16.         $umur = $data->val($i, 1);
17.         if ($umur==""){
18.             continue;
19.         }

```

```

20.      $jenis_kelamin = $data->val($i, 2);
21.      if ($jenis_kelamin==""){
22.          continue;
23.      }
24.      $tb = $data->val($i, 3);
25.      if ($tb==""){
26.          continue;
27.      }
28.      $db = $data->val($i, 4);
29.      if ($db==""){
30.          continue;
31.      }
32.      $alkphos = $data->val($i, 5);
33.      if ($alkphos==""){
34.          continue;
35.      }
36.      $sgpt = $data->val($i, 6);
37.      if ($sgpt==""){
38.          continue;
39.      }
40.      $sgot = $data->val($i, 7);
41.      if ($sgot==""){
42.          continue;
43.      }
44.      $tp = $data->val($i, 8);
45.      if ($tp==""){
46.          continue;
47.      }
48.      $alb = $data->val($i, 9);
49.      if ($alb==""){
50.          continue;
51.      }
52.      $ag = $data->val($i, 10);
53.      if ($ag==""){
54.          continue;
55.      }
56.      $hasil = mysql_query("INSERT INTO
data_testing(umur,jenis_kelamin,tb,db,alkphos,sgpt,sgot,tp,alb,a
g)
VALUES($umur,$jenis_kelamin,$tb,$db,$alkphos,$sgpt,$sgot,$tp,$al
b,$ag)");
57.      $a++;
58.  }
59.  }
60.  else if($_FILES['userfile']['type']=="text/plain"){
61.      $dbconn = mysql_connect('localhost','root','');
62.      mysql_select_db('skripsi',$dbconn);
63.      $file = $_FILES['userfile']['name'];
64.      if($fp = file_get_contents($file)) {
65.          $var_array = explode(';', $fp);
66.          foreach($var_array as $value) {
67.              mysql_query($value.';', $dbconn);
68.              $a++;
69.          }
70.      }
71.  }
72.  else{
73.      header("Location: form_input_data_testing");
74.  }
75.  if ($a>0){
76.      include ("prediksi.php");
77.      header("Location: view_hasil_klasifikasi.php");
78.  }

```

|     |                                                  |
|-----|--------------------------------------------------|
| 79. | else {                                           |
| 80. | header("Location: form_input_data_testing.php"); |
| 81. | }                                                |
| 82. | ?>                                               |

**Gambar 4.6 Implementasi Algoritma Memasukkan Data Uji**

Sumber: Implementasi

Penjelasan implementasi algoritma memasukkan data uji pada Gambar 4.6 yaitu:

1. Baris 2 berfungsi untuk memanggil library upload untuk *file* xls.
2. Baris 4 berfungsi untuk memeriksa *interval class vote* dari setiap *feature*, jika *interval class vote* ada maka data uji akan di proses.
3. Baris 12 berfungsi untuk memeriksa tipe *file* dari dokumen, jika *file* berbentuk xls maka data akan dimasukkan ke dalam *database* sesuai dengan baris 13-59, jika data tidak lengkap maka data tidak akan dimasukkan ke dalam *database*.
4. Baris 60 berfungsi untuk memeriksa tipe *file* dari dokumen, jika *file* berbentuk sql maka data akan dimasukkan ke dalam *database* sesuai dengan baris 61-71.
5. Baris 73 berfungsi untuk mengembalikan ke halaman memasukkan data uji, karena jenis dokumen yang di masukkan tidak sesuai dengan format yang di tentukan.

#### 4.4.5 Implementasi Algoritma *Preprocessing* Prediksi Klasifikasi

Operasi pada algoritma *preprocessing* prediksi klasifikasi bertujuan untuk melengkapi data yang tidak lengkap sesuai dengan bentuknya. Data nominal akan diganti dengan modus, sedangkan data numerik akan diganti dengan *mean*. Gambar 4.4 merupakan cuplikan *code* untuk algoritma *preprocessing* yang mengimplementasikan perancangan algoritma *preprocessing* prediksi klasifikasi pada sub bab 3.2.2.4

```

1. <?php
2.     include('koneksi_db.php');
3.     $array = array("umur", "jenis_kelamin", "tb", "db","alkphos",
4.     "sgpt", "sgot", "tp", "alb", "ag");
5.     for ($i=0; $i <= 9; $i++){
6.         $result= mysql_query("SELECT id_data FROM `data_testing`
7.         WHERE {$array[$i]} IS NULL");
8.         if($result){
9.             $a=0;
10.            while ($row = mysql_fetch_array($result)){
11.                if ($i==1){
12.                    $result1=mysql_query("SELECT
13.                    {$array[$i]}, count({$array[$i]}) as jumlah FROM data_testing
14.                    GROUP BY kelas ORDER BY jumlah DESC LIMIT 1");
15.                }
16.                else{
17.                    $result1= mysql_query("SELECT
18.                    avg({$array[$i]}) from data_testing");
19.                }
20.                while ($row1 = mysql_fetch_array($result1)){
21.                    $result2=mysql_query("update
22.                    data_testing set {$array[$i]}={$row1[0]} where
23.                    id_data={$row[$a]}");
24.                }
25.                unset ($row1);
26.                $a++;
27.                unset ($row);
28.            }
29.        }
30.    }
31.    ?>

```

**Gambar 4.7 Implementasi Algoritma *Preprocessing* Prediksi Klasifikasi**

Sumber: Implementasi

Penjelasan implementasi algoritma *preprocessing* prediksi klasifikasi pada Gambar 4.7 yaitu:

1. Baris 2 berfungsi untuk melakukan koneksi ke *database*.
2. Baris 3 berfungsi untuk inisialisasi *feature*.
3. Baris 4-5 berfungsi untuk memeriksa kelengkapan tiap *feature*.
4. Baris 9 berfungsi untuk memeriksa bentuk data yang hilang, jika data berbentuk nominal maka akan dicari modus datanya sesuai dengan baris 10, sedangkan jika data berbetuk numerik maka akan dicari *mean* datanya sesuai dengan baris13.
5. Baris 16 berfungsi untuk memperbarui nilai data yang tidak lengkap dengan *mean* atau modus yang sudah di cari sebelumnya.

#### 4.4.6 Implementasi Algoritma Prediksi Klasifikasi

Operasi pada algoritma prediksi klasifikasi bertujuan untuk mendapatkan hasil klasifikasi dari masukan pengguna berdasarkan jumlah terbesar dari *interval class vote* dari tiap kelas. Gambar 4.7 merupakan cuplikan *code* untuk algoritma prediksi klasifikasi yang mengimplementasikan perancangan algoritma prediksi klasifikasi pada sub bab 3.2.2.4.

```

1. <?php
2. include("koneksi_db.php");
3. $array = array("id_data", "umur", "jenis_kelamin", "tb", "db",
4. "alkphos", "sgpt", "sgot", "tp", "alb", "ag");
5. $a=0;
6. $c=0;
7. $vote=array();
8. for ($i=1; $i<=10; $i++){
9.     $result=mysql_query("SELECT count(distinct f_id) as
10.     jumlah_feature from f_{$array[$i]}")or die(mysql_error());
11.     while( $row = mysql_fetch_array($result)) {
12.         $jumlah_feature[$i] = $row["jumlah_feature"];
13.     }
14. $result=mysql_query("select id_data, umur, jenis_kelamin, tb,
15. db, alkphos, sgpt, sgot, tp, alb, ag from data_testing")or
16. die(mysql_error());
17. while( $row = mysql_fetch_array($result)) {
18.     $array[$array[0]][1] = $row["id_data"];
19.     $array[$array[1]][1] = $row["umur"];
20.     $array[$array[2]][1] = $row["jenis_kelamin"];
21.     $array[$array[3]][1] = $row["tb"];
22.     $array[$array[4]][1] = $row["db"];
23.     $array[$array[5]][1] = $row["alkphos"];
24.     $array[$array[6]][1] = $row["sgpt"];
25.     $array[$array[7]][1] = $row["sgot"];
26.     $array[$array[8]][1] = $row["tp"];
27.     $array[$array[9]][1] = $row["alb"];
28.     $array[$array[10]][1] = $row["ag"];
29. }
30. unset($endpoint);
31. for ($b=1; $b<=10; $b++){
32.     $x=0;
33.     $x=intval($jumlah_feature[$b]/2);
34.     $result=mysql_query ("select distinct endpoint from
35. f_{$array[$b]} ORDER BY char_length(endpoint), endpoint * 1
36. limit ".$x."")or die(mysql_error());
37.     while( $row = mysql_fetch_array($result)) {
38.         $endpoint[]=$row["endpoint"];
39.     }
40. }
41. for ($a=0; $a<count($array[$array[0]]); $a++){
42.     for ($d=0; $d<count($endpoint); $d++){
43.         $vote_kelas0=0;
44.         $vote_kelas1=0;
45.         for ($b=1; $b<=10; $b++){
46.             $vote[0]=0;
47.             $vote[1]=0;
48.             for ($c=0; $c<$jumlah_feature[$b]; $c++){
49.                 if ($c==0){

```

```

44.         if
($array[$array[$b]][$a]<$endpoint[$d]){
45.             unset($vote);
46.             $result=mysql_query("SELECT interval_class_vote FROM
f_{$array[$b]} WHERE f_id='f'.".$c.'" ORDER BY kelas ASC") or
die(mysql_error());
47.             while( $row =
mysql_fetch_array($result)) {
48.                 $vote[]=$row["interval_class_vote"];
49.             }
50.             }
51.             else{
52.                 $vote[0]=0;
53.                 $vote[1]=0;
54.             }
55.             }
56.             else if($c%2==1){
57.                 if
($array[$array[$b]][$a]==$endpoint[$d]){
58.                     unset($vote);
59.                     $result=mysql_query("SELECT interval_class_vote FROM
f_{$array[$b]} WHERE f_id='f'.".$c.'" ORDER BY kelas ASC") or
die(mysql_error()) or die(mysql_error());
60.                     while( $row =
mysql_fetch_array($result)) {
61.                         $vote[]=$row["interval_class_vote"];
62.                     }
63.                     }
64.                     else{
65.                         $vote[0]=0;
66.                         $vote[1]=0;
67.                     }
68.                     }
69.                     else if($c%2==0 and $c!=0 and
$c!=( $jumlah_feature[$b]-1)){
70.                         $d++;
71.                         if
($array[$array[$b]][$a]>$endpoint[$d-1] and
$array[$array[$b]][$a]<$endpoint[$d]){
72.                             unset($vote);
73.                             $result=mysql_query("SELECT interval_class_vote FROM
f_{$array[$b]} WHERE f_id='f'.".$c.'" ORDER BY kelas ASC") or
die(mysql_error());
74.                             while( $row =
mysql_fetch_array($result)) {
75.                                 $vote[]=$row["interval_class_vote"];
76.                             }
77.                             }
78.                             else{
79.                                 $vote[0]=0;
80.                                 $vote[1]=0;
81.                             }
82.                             }
83.                             else if($c=( $jumlah_feature[$b]-1)){
84.                                 $d++;
85.                                 if
($array[$array[$b]][$a]>$endpoint[$d-1]){
86.                                     unset($vote);
87.                                     $result=mysql_query("SELECT interval_class_vote FROM
f_{$array[$b]} WHERE f_id='f'.".$c.'" ORDER BY kelas ASC") or
die(mysql_error());
88.                                     while( $row =
mysql_fetch_array($result)) {

```

```

90.         $vote[]=$row["interval_class_vote"];
91.     }
92.     }
93.     else{
94.         $vote[0]=0;
95.         $vote[1]=0;
96.     }
97. }
98. $vote_kelas0=$vote_kelas0+$vote[0];
99. $vote_kelas1=$vote_kelas1+$vote[1];
100. }
101. }
102. if ($vote_kelas0>$vote_kelas1){
103.     $hasil='Tidak Terjangkit Penyakit';
104. }
105. else{
106.     $hasil='Terjangkit Penyakit';
107. }
108. $result=mysql_query("update data_testing set
109. hasil_klasifikasi='$hasil' where
    id_data=".$sarray[$sarray[0]][$a].""") or die(mysql_error());
110. }
111. }
112. ?>

```

**Gambar 4.8 Implementasi Algoritma Prediksi Klasifikasi**

Sumber: Implementasi

Penjelasan implementasi algoritma prediksi klasifikasi pada Gambar 4.8 yaitu:

1. Baris 2 berfungsi untuk melakukan koneksi ke *database*.
2. Baris 3 berfungsi untuk inisialisasi *feature*.
3. Baris 44-98 berfungsi untuk mencari *interval class vote* tiap kelas dari tiap *interval* dari tiap *feature* yang dimasukkan ke dalam variabel *vote*.
4. Baris 99-100 berfungsi untuk menjumlahkan semua *vote* dari tiap kelas.
5. Baris 103-108 berfungsi untuk mencari nilai *vote* terbesar yang nantinya akan menjadi hasil klasifikasi dari data uji
6. Baris 109 berfungsi untuk memasukkan nilai hasil klasifikasi ke dalam *database*.

#### 4.5 Implementasi Antar Muka Aplikasi

Antarmuka Aplikasi klasifikasi *incomplete data* penyakit liver pada manusia digunakan oleh pengguna untuk berinteraksi dengan sistem perangkat lunak.

Antarmuka perangkat lunak ini dibagi menjadi lima, yaitu antarmuka halaman utama, halaman memasukkan data pelatihan, halaman data pelatihan klasifikasi, halaman memasukkan data uji, dan halaman hasil klasifikasi.

#### 4.5.1 Halaman Utama

Halaman utama merupakan halaman pertama yang akan dibuka ketika sistem dijalankan. Dari halaman home kita dapat memilih tombol memasukkan data pelatihan, data pelatihan klasifikasi, memasukkan data uji atau hasil klasifikasi untuk memasukkan data dan melihat hasil klasifikasi. Gambar 4.9 menunjukkan implementasi tampilan antarmuka dari halaman utama yang mengacu pada perancangan antarmuka halaman utama dalam sub bab 3.2.2.5.



**Gambar 4.9 Tampilan Antar Muka Halaman Utama**

Sumber: Implementasi

#### 4.5.2 Halaman Memasukkan Data Pelatihan

Halaman memasukkan data pelatihan berfungsi untuk memasukkan data pelatihan baik berbentuk *file* maupun secara manual. Gambar 4.10 menunjukkan implementasi tampilan antarmuka dari halaman memasukkan data pelatihan yang

mengacu pada perancangan antarmuka memasukkan data pelatihan dalam sub bab 3.2.2.5.

**Gambar 4.10 Tampilan Antar Muka Halaman Memasukkan Data Pelatihan**  
Sumber: Implementasi

#### 4.5.3 Halaman Data Pelatihan Klasifikasi

Halaman data pelatihan klasifikasi berfungsi untuk menampilkan data pelatihan yang telah dimasukkan oleh pengguna. Pada halaman ini pengguna dapat melakukan *edit*, tambah dan hapus data pelatihan. Gambar 4.11 menunjukkan implementasi tampilan antarmuka dari halaman data pelatihan klasifikasi yang mengacu pada perancangan antarmuka halaman data pelatihan klasifikasi dalam sub bab 3.2.2.5.

## Klasifikasi Incomplete Data Penyakit Liver pada Manusia

[HOMEPAGE](#)
[INPUT DATA TRAINING](#)
[DATA PELATIHAN KLASIFIKASI](#)
[INPUT DATA TESTING](#)
[HASIL KLASIFIKASI](#)

Tabel Data Pelatihan Klasifikasi

[View All](#) | [View Paginated](#)

| Umur | Jenis Kelamin | Total Bilirubin | Direct Bilirubin | Alkaline Phosphatase | Alamine Aminotransferase | Aspartate Aminotransferase | Total Protiens | Albumin | Albumin and Globulin Ratio | Kelas Penyakit                  | Edit                 | Delete                 |
|------|---------------|-----------------|------------------|----------------------|--------------------------|----------------------------|----------------|---------|----------------------------|---------------------------------|----------------------|------------------------|
| 84   | Female        | 0.70            | 0.20             | 188                  | 13                       | 21                         | 6.00           | 3.20    | 1.10                       | Tidak Terjangkit Penyakit Liver | <a href="#">Edit</a> | <a href="#">Delete</a> |
| 57   | Male          | 4.00            | 1.90             | 190                  | 42                       | 111                        | 5.20           | 1.50    | 0.40                       | Terjangkit Penyakit Liver       | <a href="#">Edit</a> | <a href="#">Delete</a> |
| 52   | Male          | 0.90            | 0.20             | 156                  | 35                       | 82                         | 4.90           | 2.90    | 1.40                       | Terjangkit Penyakit Liver       | <a href="#">Edit</a> | <a href="#">Delete</a> |
|      |               |                 |                  |                      |                          |                            |                |         |                            | Tidak Terjangkit Penyakit Liver |                      |                        |

**Gambar 4.11 Tampilan Antar Muka Halaman Data Pelatihan Klasifikasi**

Sumber: Implementasi

### 4.5.4 Halaman Memasukkan Data Uji

Halaman memasukkan data uji berfungsi untuk memasukkan data uji baik berbentuk *file* maupun secara manual. Gambar 4.12 menunjukkan implementasi tampilan antarmuka dari halaman memasukkan data uji yang mengacu pada perancangan antarmuka memasukkan data uji dalam sub bab 3.2.2.5.

**Klasifikasi Incomplete Data Penyakit Liver pada Manusia**

HOMEPAGE INPUT DATA TRAINING DATA PELATIHAN KLASIFIKASI INPUT DATA TESTING HASIL KLASIFIKASI

**Input File Data Testing**

Pilih File Excel:  Browse...

**Input Manual Data Testing**

Umur \*:  
Jenis Kelamin \*:  
Male: ☐  
Female: ☐  
Total Bilirubin \*:  
Direct Bilirubin \*:  
Alkaline Phosphatase \*:  
Alamine Aminotransferase \*:  
Aspartate Aminotransferase \*:  
Total Protien \*:

**Gambar 4.12 Tampilan Antar Muka Halaman Memasukkan Data Uji**

Sumber: Implementasi

#### **4.5.5 Halaman Hasil Klasifikasi**

Halaman hasil klasifikasi berfungsi untuk menampilkan hasil klasifikasi yang dibuat oleh sistem. Gambar 4.13 menunjukkan implementasi tampilan antarmuka dari halaman hasil klasifikasi yang mengacu pada perancangan antarmuka halaman hasil klasifikasi dalam sub bab 3.2.2.5.

## Klasifikasi Incomplete Data Penyakit Liver pada Manusia

HOME PAGE

INPUT DATA TRAINING

DATA PELATIHAN KLASIFIKASI

INPUT DATA TESTING

HASIL KLASIFIKASI

Tabel Hasil Klasifikasi

View All | View Paginated

| Umur | Jenis Kelamin | Total Bilirubin | Direct Bilirubin | Alkaline Phosphatase | Alanine Aminotransferase | Aspartate Aminotransferase | Total Proteins | Albumin | Albumin and Globulin Ratio | Hasil Klasifikasi         | Edit                 | Delete                 |
|------|---------------|-----------------|------------------|----------------------|--------------------------|----------------------------|----------------|---------|----------------------------|---------------------------|----------------------|------------------------|
| 84   | Female        | 0.70            | 0.20             | 188                  | 13                       | 21                         | 6.00           | 3.20    | 1.10                       | Tidak Terjangkit Penyakit | <a href="#">Edit</a> | <a href="#">Delete</a> |
| 57   | Male          | 1.00            | 0.30             | 187                  | 19                       | 23                         | 5.20           | 2.90    | 1.20                       | Tidak Terjangkit Penyakit | <a href="#">Edit</a> | <a href="#">Delete</a> |
| 38   | Female        | 2.80            | 1.20             | 410                  | 59                       | 57                         | 5.80           | 3.00    | 0.80                       | Tidak Terjangkit Penyakit | <a href="#">Edit</a> | <a href="#">Delete</a> |
| 46   | Female        | 14.20           | 7.80             | 374                  | 38                       | 77                         | 4.30           | 2.00    | 0.80                       | Terjangkit Penyakit       | <a href="#">Edit</a> | <a href="#">Delete</a> |

Gambar 4.13 Tampilan Antar Muka Halaman Hasil Klasifikasi

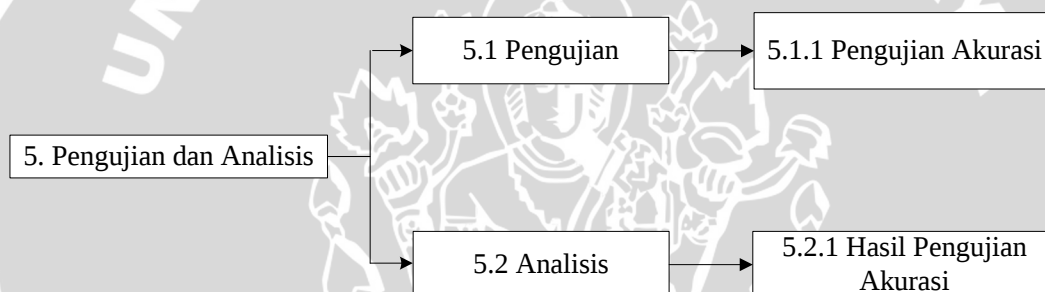
Sumber: Implementasi



## BAB V

### PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian dan analisis Sistem klasifikasi *incomplete data* penyakit liver pada manusia dengan algoritma *Voting Feature Interval-5*. Proses pengujian dilakukan dengan cara pengujian akurasi. Pengujian akurasi dilakukan dengan cara menghitung akurasi pada setiap tingkatan jumlah data yang hilang dan perlakuan untuk data yang hilang. Gambar 5.1 menunjukkan pohon pengujian dan analisis.



**Gambar 5.1 Pohon Pengujian Dan Analisis**

Sumber: Pengujian dan Analisis

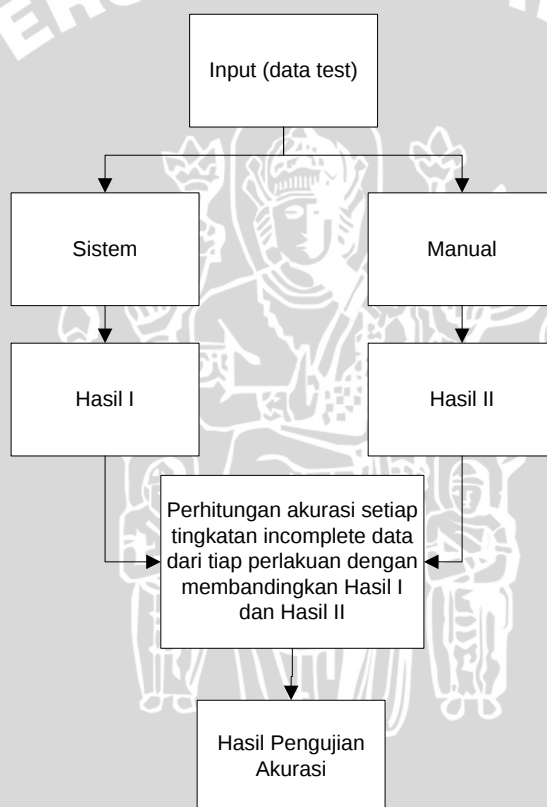
#### 5.1 Pengujian

Proses pengujian dilakukan dengan cara pengujian akurasi. Pengujian akurasi dilakukan untuk mengetahui performa dari sistem yang dibangun.

##### 5.1.1 Pengujian Akurasi

Pengujian akurasi digunakan untuk mengetahui performa dari sistem klasifikasi *incomplete data* dalam memberikan hasil klasifikasi kepada pengguna dengan menggunakan metode *Voting Feature Interval-5*. Pengujian ini dilakukan dengan menghitung akurasi pada setiap tingkatan *incomplete data* pelatihan maupun *incomplete data* uji serta perlakuan untuk *incomplete data*. Perhitungan akurasi dilakukan dengan cara membandingkan antara hasil klasifikasi sistem

dengan hasil klasifikasi yang sudah ditentukan dari data uji. Pengujian dilakukan menggunakan 174 data uji dan 405 data pelatihan yang disadur dari *UCI Machine Learning*. Data pelatihan akan dibuat menjadi 2 bentuk yaitu data yang tidak lengkap dan data yang lengkap. Data pelatihan dan data uji akan dibuat *incomplete* dengan tingkatan 70%, 50%, 25%, 20%, 15%, 10%, 5%, dan 2%. Data yang tidak lengkap akan mengalami tiga perlakuan yaitu menghapus satu baris *incomplete data*, mengabaikan *incomplete data*, dan mengganti *incomplete data* dengan *mean* atau modus. Gambar 5.2 merupakan alur proses pada pengujian akurasi yang dilakukan.



**Gambar 5.2 Diagram Alir Proses Pengujian Akurasi**

Sumber: Pengujian dan Analisis

Rincian hasil pengujian akurasi dengan data yang lengkap ditunjukkan pada Tabel 5.1

**Tabel 5.1 Pengujian Akurasi Dengan Data Yang Lengkap**

| Akurasi |
|---------|
| 0,85    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan data yang lengkap menghasilkan nilai akurasi yang cukup besar yaitu sebesar sebesar 0,85. Dari hasil ini dapat disimpulkan bahwa algoritma *Voting Feature Interval-5* dapat melakukan klasifikasi dengan baik.

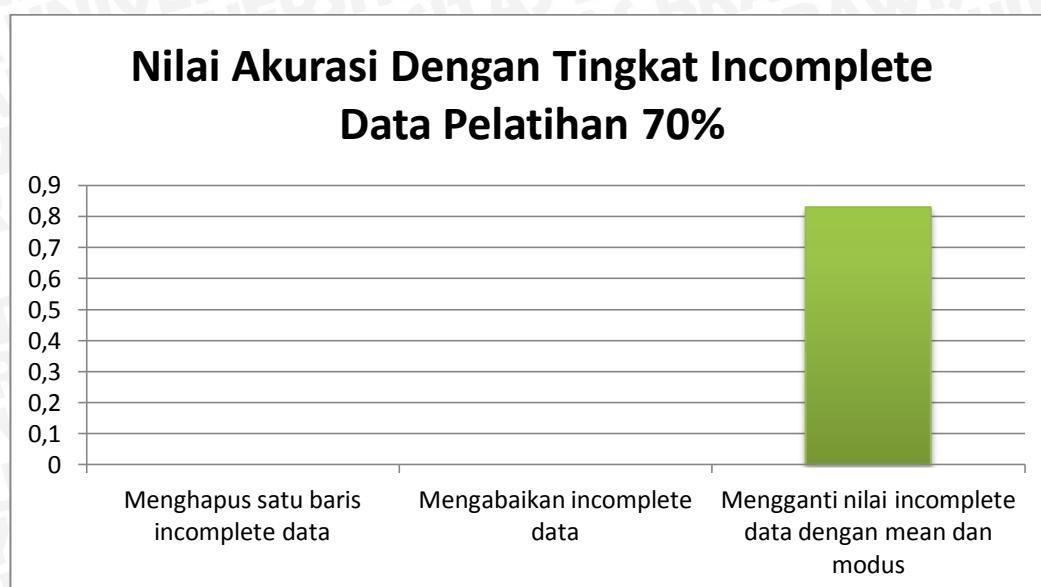
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data* maka dilakukan pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 70%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 70% ditunjukkan pada Tabel 5.2.

**Tabel 5.2 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 70%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0       |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0       |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,83    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* pelatihan sebesar 70% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,83.



**Gambar 5.3 Nilai Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 70%**

Sumber: Pengujian dan Analisis

Gambar 5.3 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* pelatihan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* pelatihan sebesar 70% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

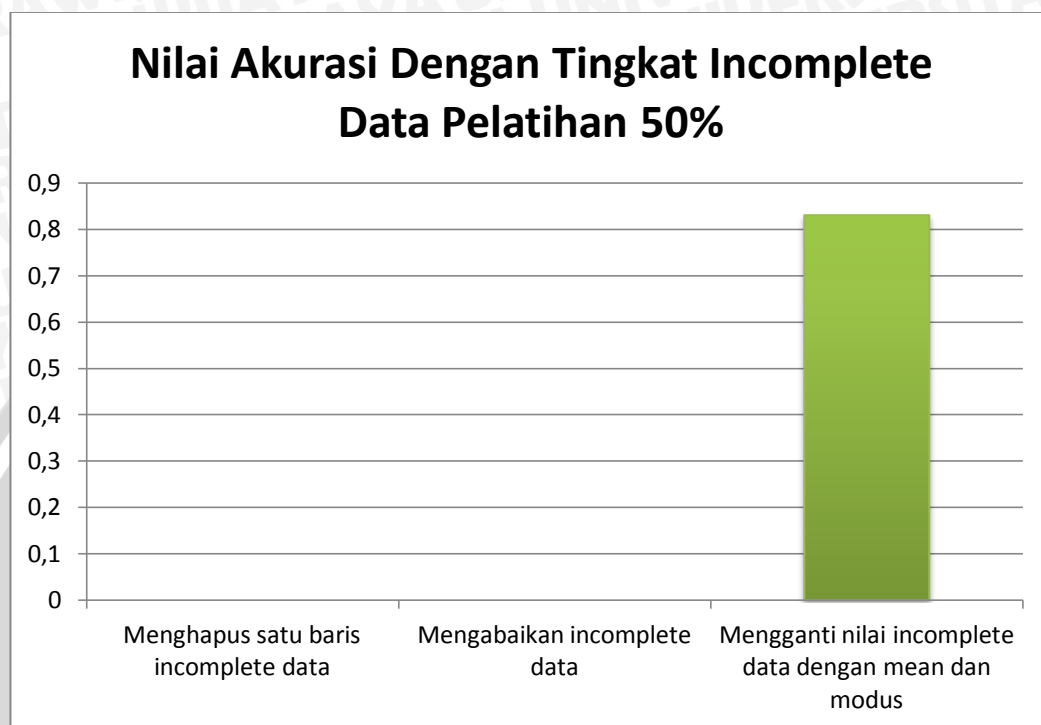
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data* maka dilakukan pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 50%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 50% ditunjukkan pada Tabel 5.3.

**Tabel 5.3 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 50%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0       |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0       |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,83    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* pelatihan sebesar 50% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,83.



**Gambar 5.4 Nilai Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 50%**

Sumber: Pengujian dan Analisis

Gambar 5.4 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* pelatihan yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* pelatihan sebesar 50% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

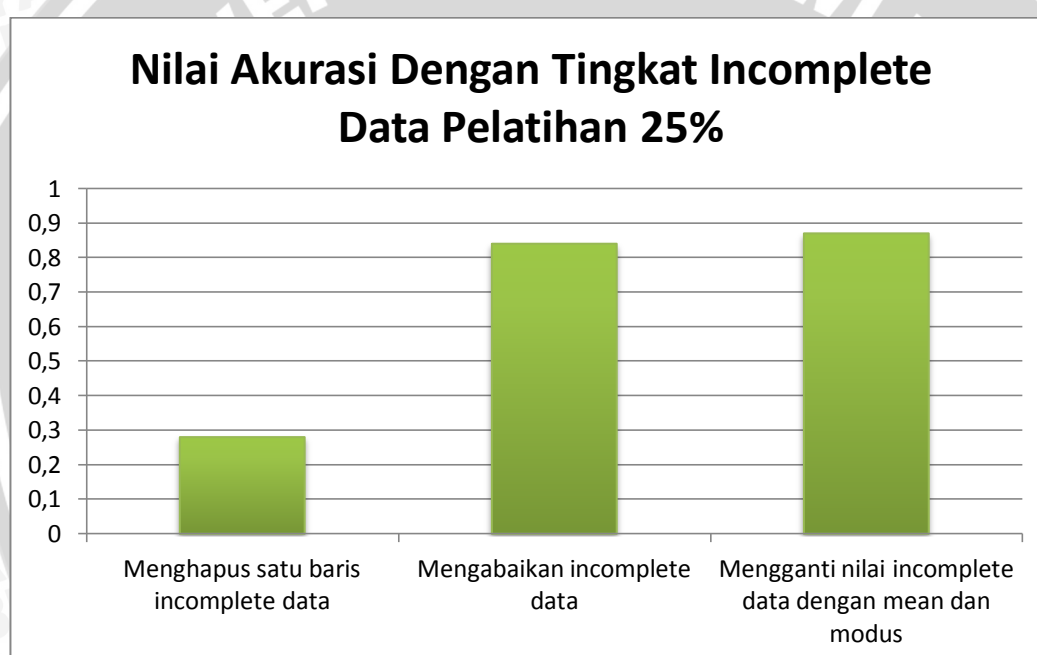
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 25%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 25% ditunjukkan pada Tabel 5.4.

**Tabel 5.4 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 25%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,28    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,84    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,87    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* pelatihan sebesar 25% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,87.



**Gambar 5.5 Nilai Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 25%**

Sumber: Pengujian dan Analisis

Gambar 5.5 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* pelatihan yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* pelatihan sebesar 25% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

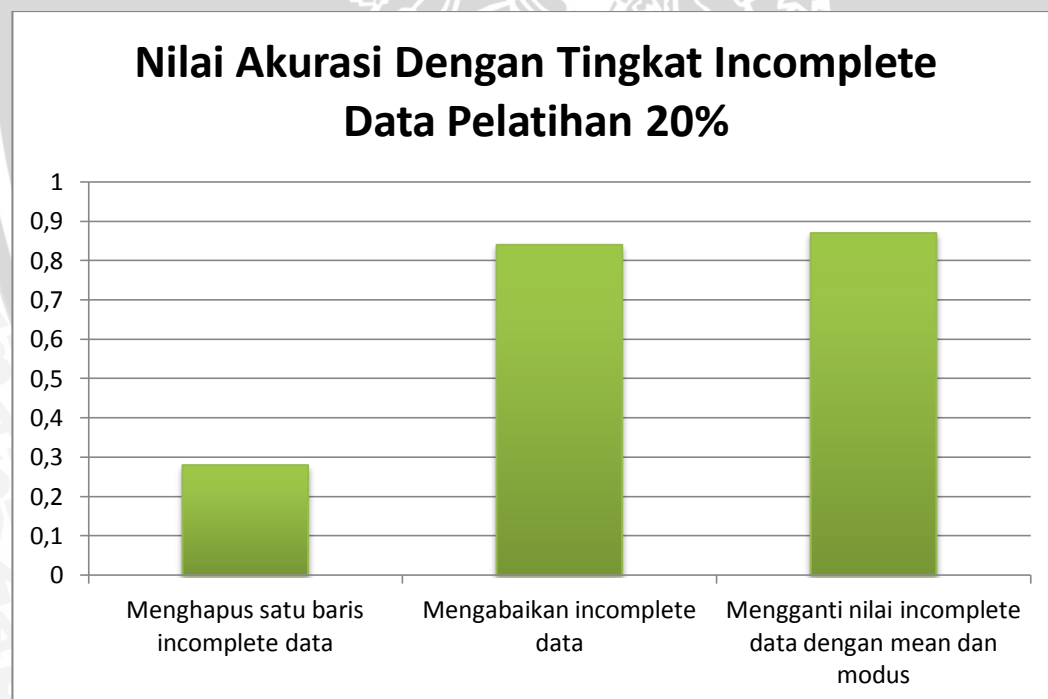
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 20%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 20% ditunjukkan pada Tabel 5.5.

**Tabel 5.5 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 20%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,28    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,84    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,87    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* pelatihan sebesar 20% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,87.



**Gambar 5.6 Nilai Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 20%**

Sumber: Pengujian dan Analisis

Gambar 5.6 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* pelatihan yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* pelatihan sebesar 20% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

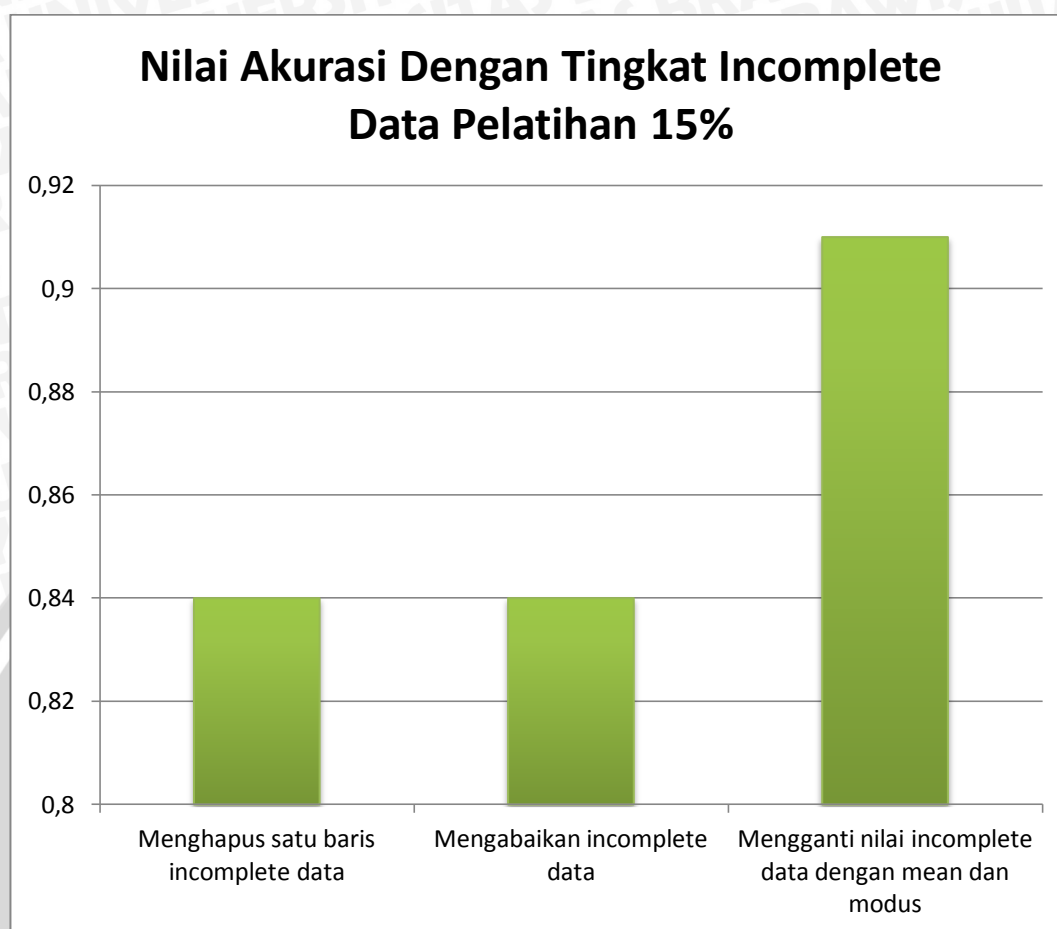
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 15%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 15% ditunjukkan pada Tabel 5.6.

**Tabel 5.6 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 15%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,84    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,84    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,91    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* pelatihan sebesar 15% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,91.



**Gambar 5.7 Nilai Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 15%**

Sumber: Pengujian dan Analisis

Gambar 5.7 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* pelatihan yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* pelatihan sebesar 15% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

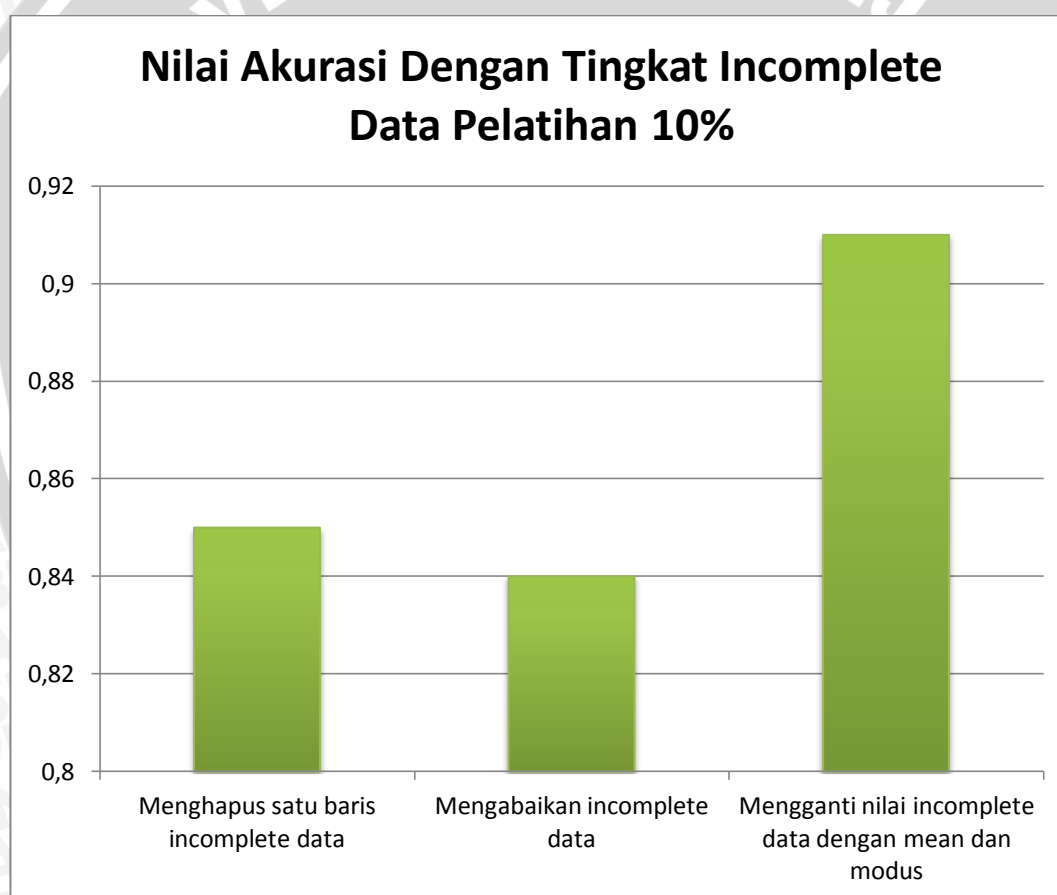
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 10%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 10% ditunjukkan pada Tabel 5.7.

**Tabel 5.7 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 10%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,85    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,84    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,91    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* pelatihan sebesar 10% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,91.



**Gambar 5.8 Nilai Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 10%**

Sumber: Pengujian dan Analisis

Gambar 5.8 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* pelatihan yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* pelatihan sebesar 10% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

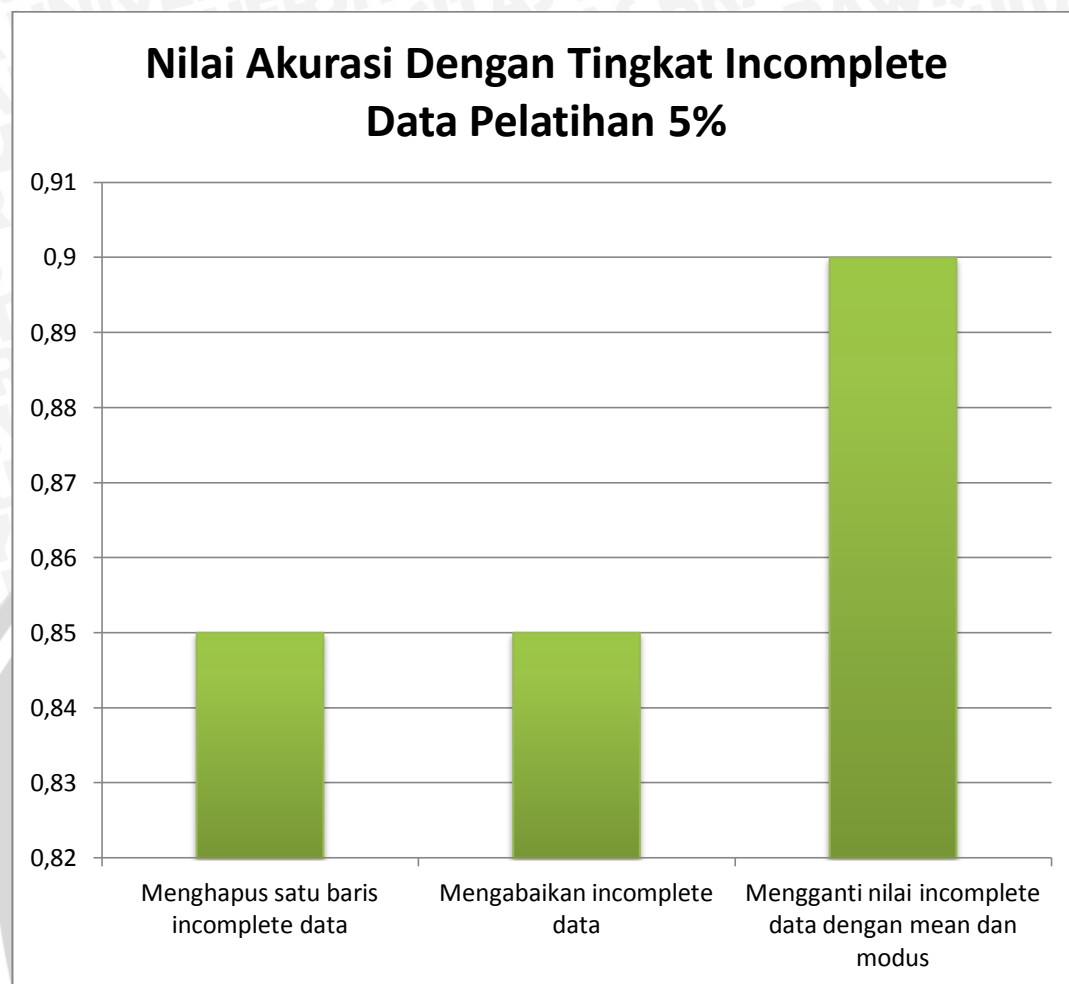
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* sebesar pelatihan 5%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 5% ditunjukkan pada Tabel 5.8.

**Tabel 5.8 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 5%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,85    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,85    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,9     |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* pelatihan sebesar 5% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,9.



**Gambar 5.9 Nilai Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 5%**

Sumber: Pengujian dan Analisis

Gambar 5.9 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* pelatihan yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* pelatihan sebesar 5% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

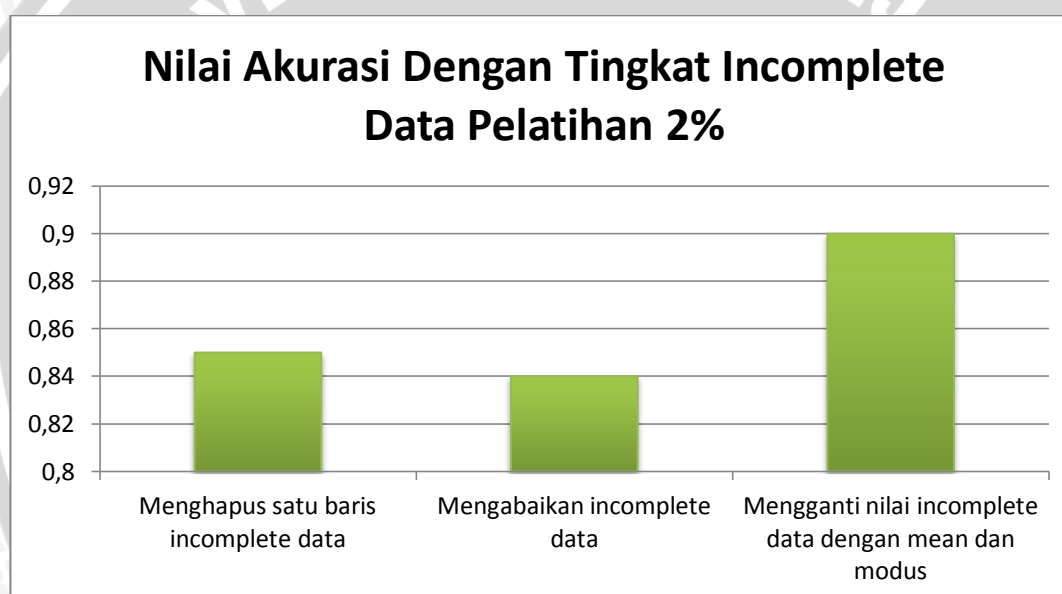
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 2%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* pelatihan sebesar 2% ditunjukkan pada Tabel 5.9.

**Tabel 5.9 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 2%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,85    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,84    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,9     |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* pelatihan sebesar 2% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,9.



**Gambar 5.10 Nilai Akurasi Dengan Tingkat *Incomplete Data* Pelatihan 2%**

Sumber: Pengujian dan Analisis

Gambar 5.10 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* pelatihan yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* pelatihan sebesar 2% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

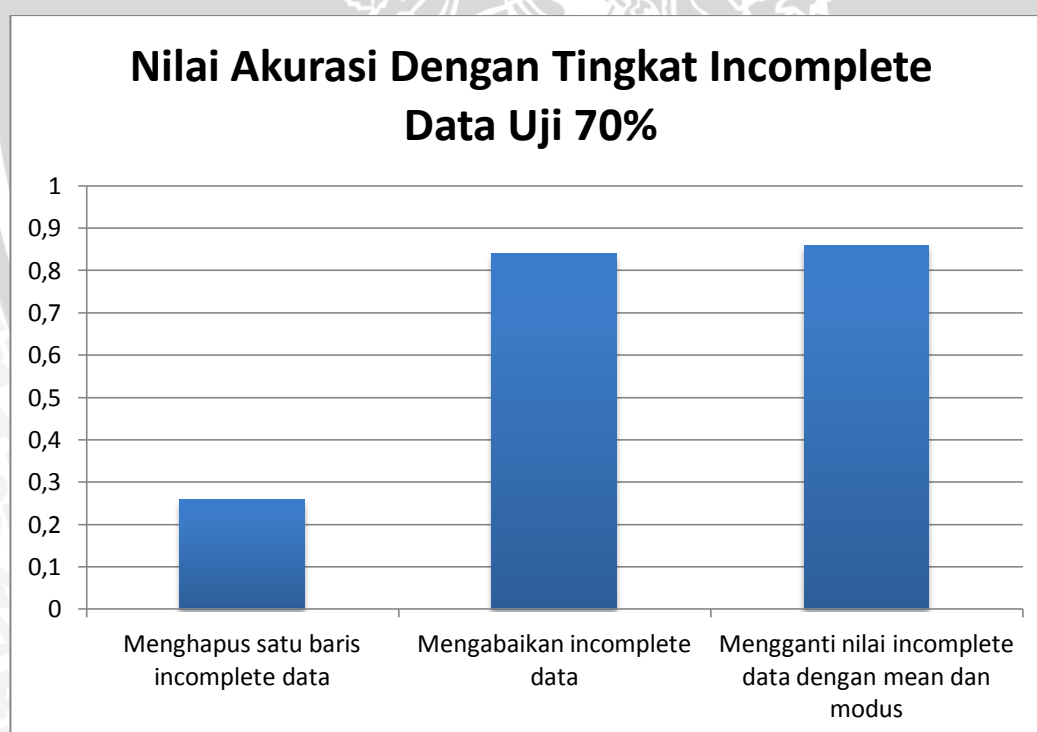
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data* maka dilakukan pengujian akurasi dengan tingkat *incomplete data* uji sebesar 70%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* uji sebesar 70% ditunjukkan pada Tabel 5.10.

**Tabel 5.10 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Uji 70%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,26    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,84    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,86    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* uji sebesar 70% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,86.



**Gambar 5.11 Nilai Akurasi Dengan Tingkat *Incomplete Data* Uji 70%**

Sumber: Pengujian dan Analisis

Gambar 5.11 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* uji. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* uji sebesar 70% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

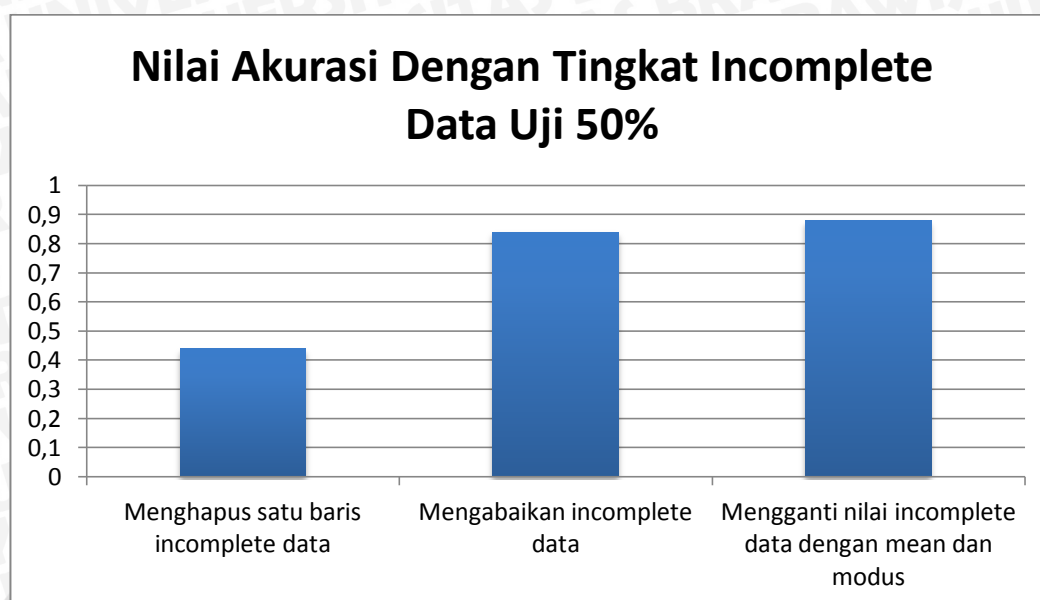
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data* maka dilakukan pengujian akurasi dengan tingkat *incomplete data* uji sebesar 50%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* uji sebesar 50% ditunjukkan pada Tabel 5.11.

**Tabel 5.11 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Uji 50%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,44    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,84    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,86    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* uji sebesar 50% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,86.



**Gambar 5.12 Nilai Akurasi Dengan Tingkat *Incomplete Data* Uji 50%**

Sumber: Pengujian dan Analisis

Gambar 5.12 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* uji yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* uji sebesar 50% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

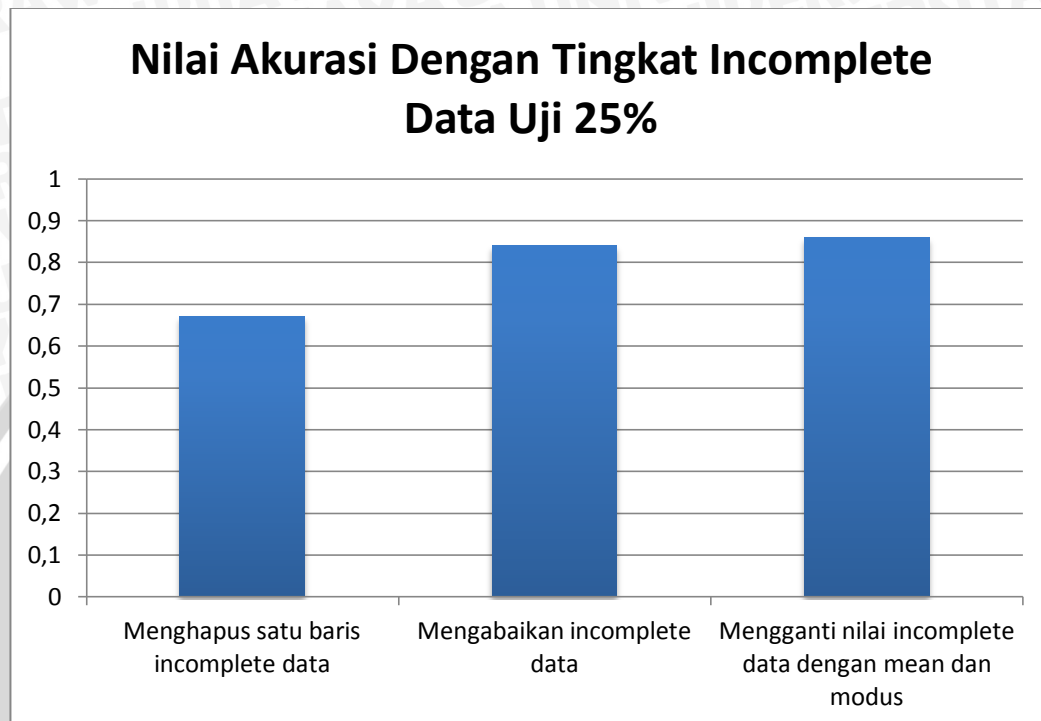
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* uji sebesar 25%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* uji sebesar 25% ditunjukkan pada Tabel 5.12.

**Tabel 5.12 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Uji 25%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,67    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,84    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,86    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* uji sebesar 25% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau *modus* yaitu sebesar 0,86.



**Gambar 5.13 Nilai Akurasi Dengan Tingkat *Incomplete Data* Uji 25%**

Sumber: Pengujian dan Analisis

Gambar 5.13 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* uji yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau *modus*. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* uji sebesar 25% adalah mengganti nilai *incomplete data* dengan *mean* atau *modus*.

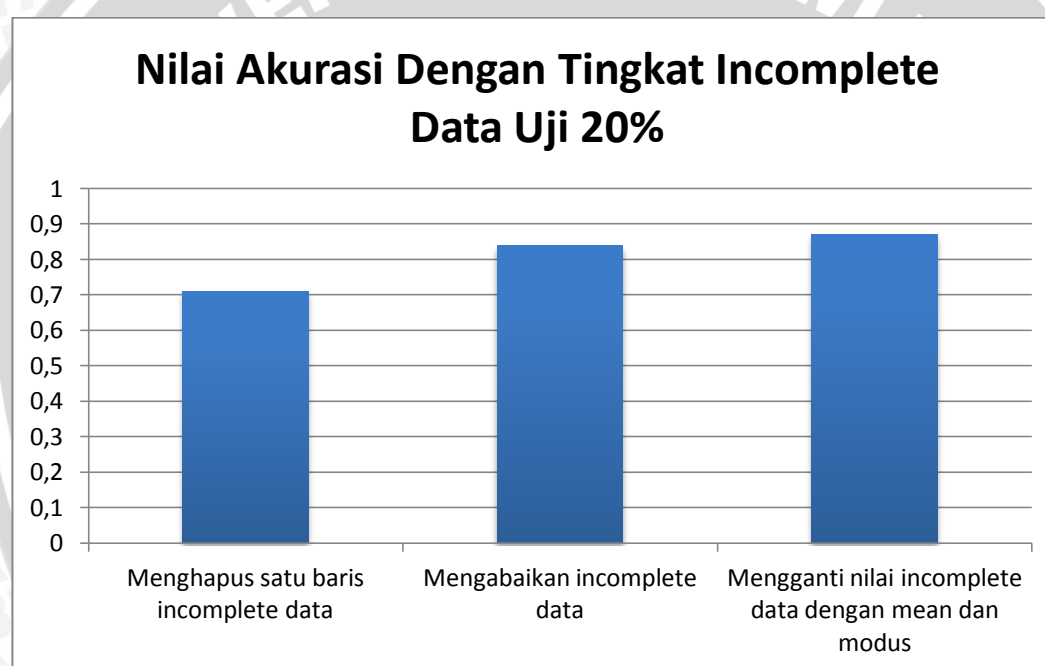
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* uji sebesar 20%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* uji sebesar 20% ditunjukkan pada Tabel 5.13.

**Tabel 5.13 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Uji 20%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,71    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,84    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,87    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* uji sebesar 20% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,87.



**Gambar 5.14 Nilai Akurasi Dengan Tingkat *Incomplete Data* Uji 20%**

Sumber: Pengujian dan Analisis

Gambar 5.14 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* uji yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* uji sebesar 20% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

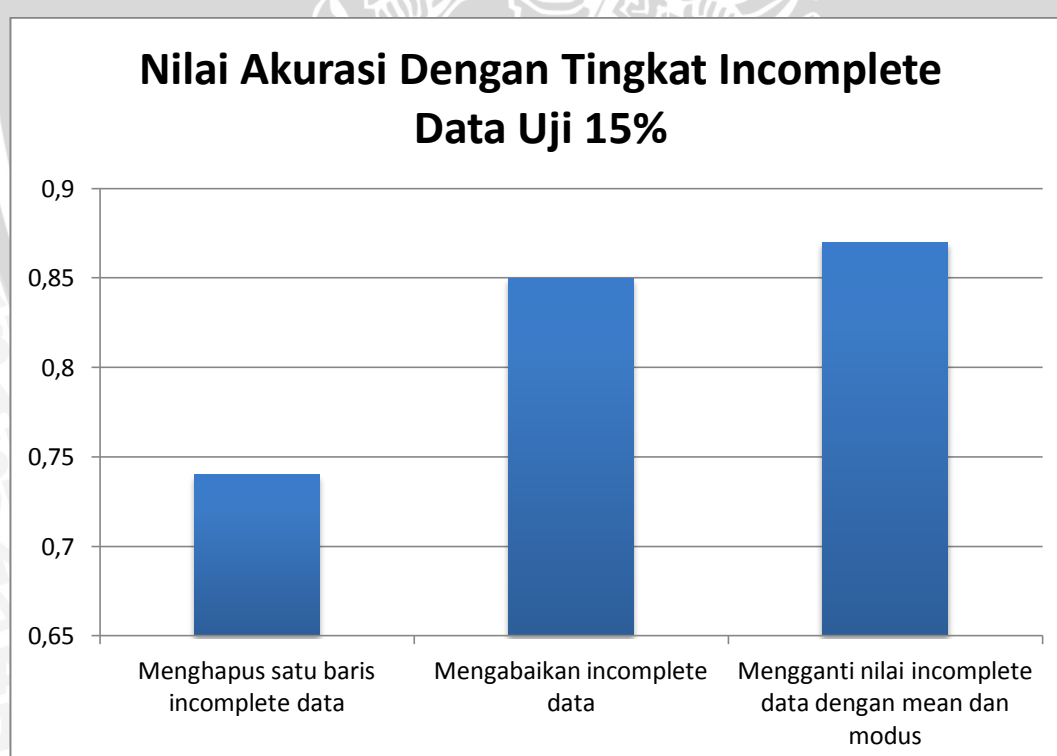
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* uji sebesar 15%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* uji sebesar 15% ditunjukkan pada Tabel 5.14.

**Tabel 5.14 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Uji 15%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,74    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,85    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,87    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* uji sebesar 15% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,87.



**Gambar 5.15 Nilai Akurasi Dengan Tingkat *Incomplete Data* Uji 15%**

Sumber: Pengujian dan Analisis

Gambar 5.15 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* uji yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* uji sebesar 15% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

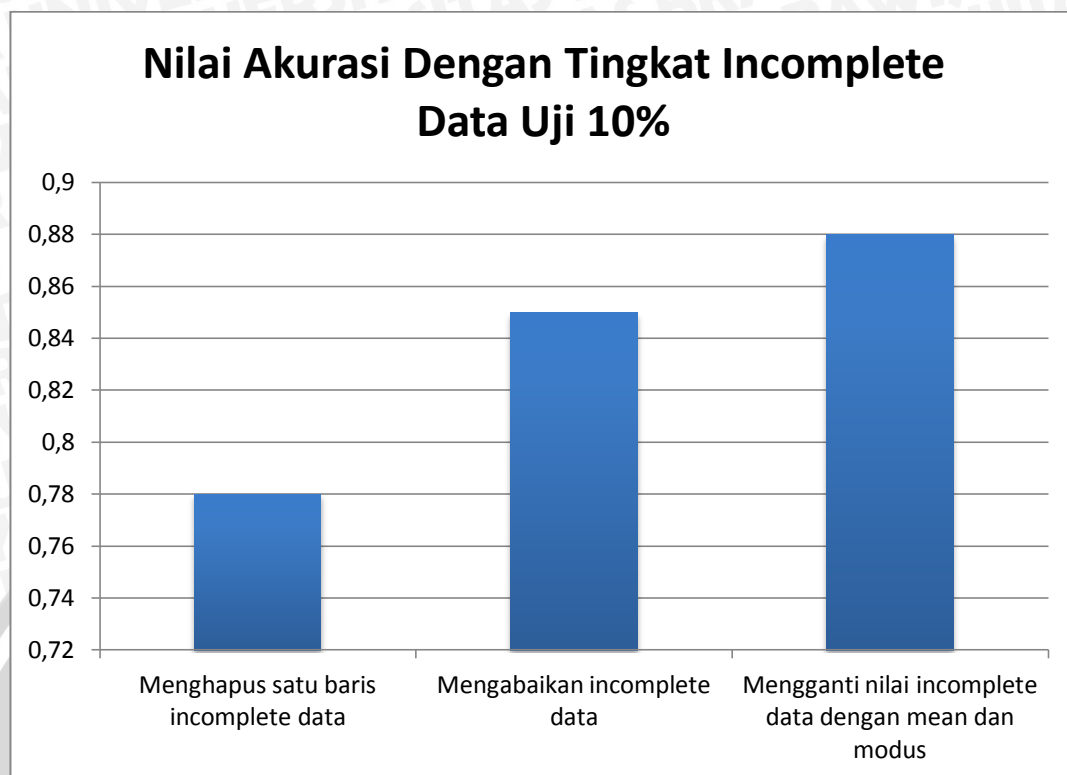
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* uji sebesar 10%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* uji sebesar 10% ditunjukkan pada Tabel 5.15.

**Tabel 5.15 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Uji 10%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,78    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,85    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,88    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* uji sebesar 10% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,88.



**Gambar 5.16 Nilai Akurasi Dengan Tingkat *Incomplete Data* Uji 10%**

Sumber: Pengujian dan Analisis

Gambar 5.16 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* uji yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* uji sebesar 10% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

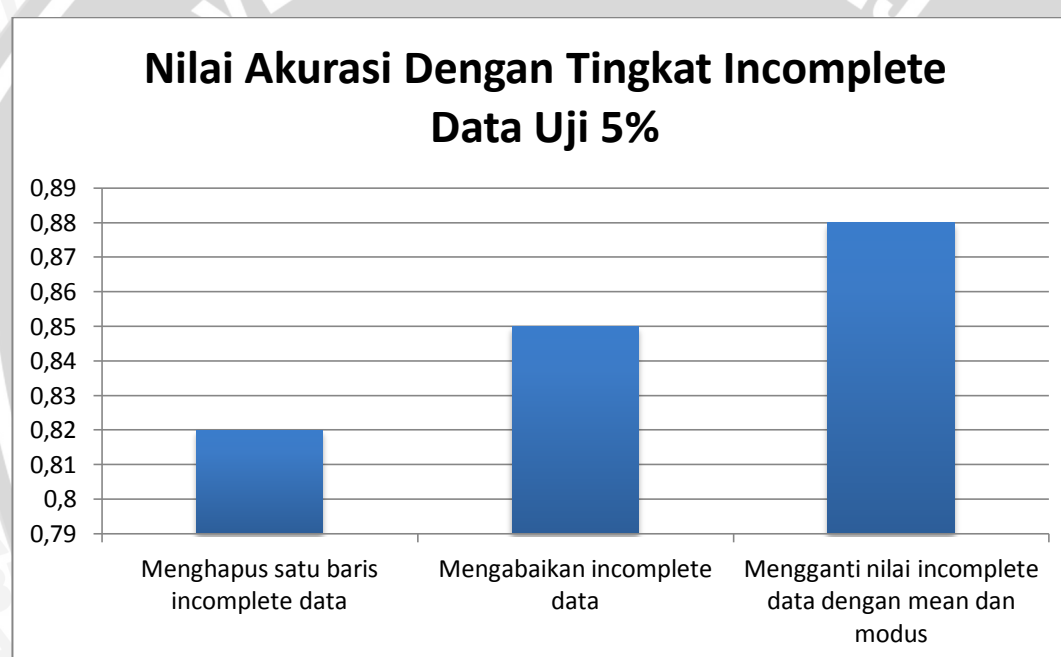
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* sebesar uji 5%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* uji sebesar 5% ditunjukkan pada Tabel 5.16.

**Tabel 5.16 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Uji 5%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,82    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,85    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,88    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* uji sebesar 5% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,88.



**Gambar 5.17 Nilai Akurasi Dengan Tingkat *Incomplete Data* Uji 5%**

Sumber: Pengujian dan Analisis

Gambar 5.17 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* uji yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* uji sebesar 5% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

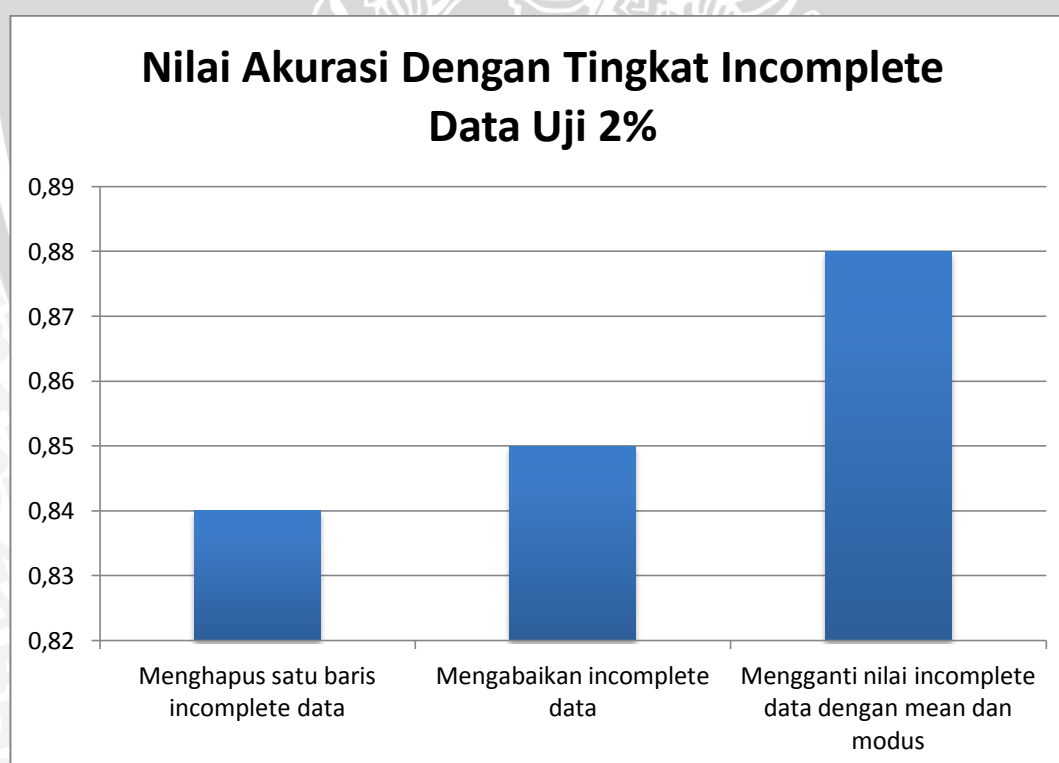
Untuk mengetahui nilai akurasi masing-masing perlakuan terhadap tingkat *incomplete data*, maka dilakukan pengujian akurasi dengan tingkat *incomplete data* uji sebesar 2%. Rincian hasil pengujian akurasi dengan tingkat *incomplete data* uji sebesar 2% ditunjukkan pada Tabel 5.17.

**Tabel 5.17 Pengujian Akurasi Dengan Tingkat *Incomplete Data* Uji 2%**

| NO | Perlakuan Terhadap <i>Incomplete Data</i>                           | Akurasi |
|----|---------------------------------------------------------------------|---------|
| 1  | Menghapus satu baris <i>incomplete data</i>                         | 0,84    |
| 2  | Mengabaikan <i>incomplete data</i>                                  | 0,85    |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,88    |

Sumber: Pengujian dan Analisis

Hasil pengujian akurasi 174 data uji dengan tingkat *incomplete data* uji sebesar 2% menghasilkan nilai akurasi terbesar pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus yaitu sebesar 0,88.



**Gambar 5.18 Nilai Akurasi Dengan Tingkat *Incomplete Data* Uji 2%**

Sumber: Pengujian dan Analisis

Gambar 5.18 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan tingkatan *incomplete data* uji yang dapat dihasilkan. Dari grafik dapat dilihat bahwa nilai akurasi terbesar ditunjukkan pada perlakuan mengganti nilai *incomplete data* dengan *mean* atau modus. Dapat disimpulkan bahwa perlakuan terbaik terhadap *incomplete data* uji sebesar 2% adalah mengganti nilai *incomplete data* dengan *mean* atau modus.

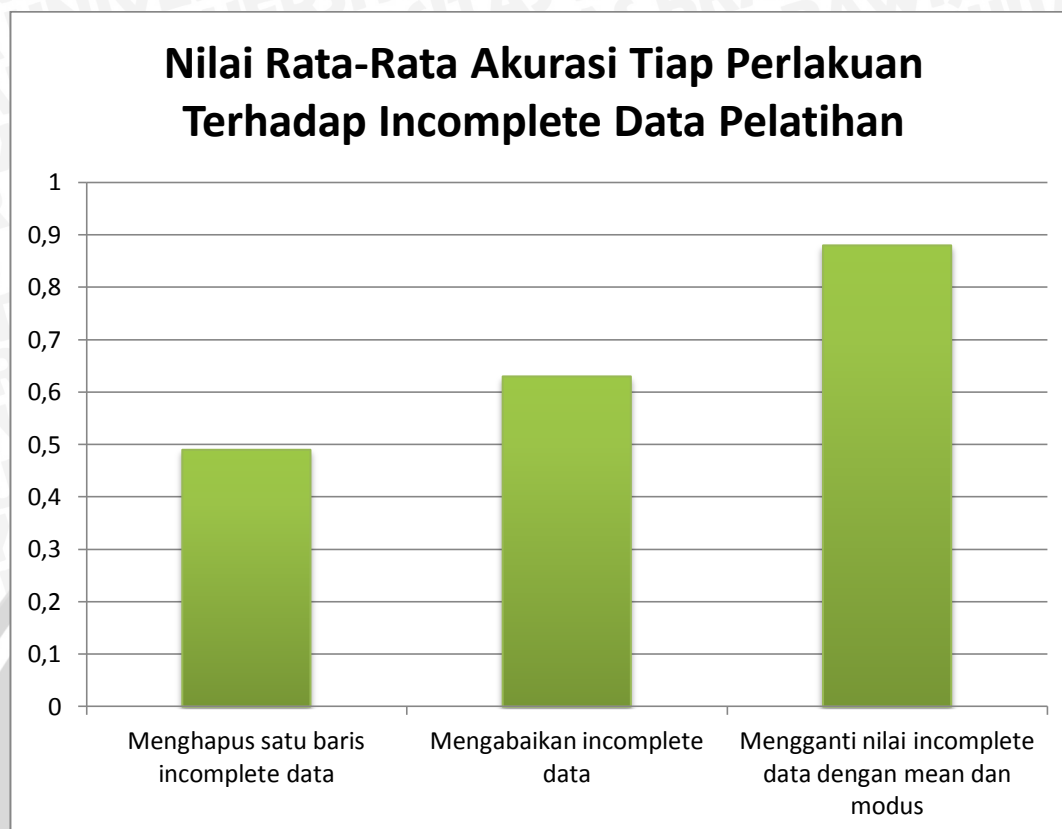
Nilai rata-rata akurasi dari setiap tingkat perlakuan terhadap *incomplete data* pelatihan yang dihasilkan oleh sistem dapat dilihat pada Tabel 5.18.

**Tabel 5.18 Nilai Rata-Rata Akurasi Menurut Perlakuan Terhadap *Incomplete Data* Pelatihan**

| Perlakuan Terhadap <i>Incomplete Data</i>                           | Tingkatan <i>Incomplete Data</i> |      |      |      |      |      |      |      | Nilai Rata-Rata Akurasi |
|---------------------------------------------------------------------|----------------------------------|------|------|------|------|------|------|------|-------------------------|
|                                                                     | 70%                              | 50%  | 25%  | 20%  | 15%  | 10%  | 5%   | 2%   |                         |
| Menghapus satu baris <i>incomplete data</i>                         | 0                                | 0    | 0,28 | 0,28 | 0,84 | 0,85 | 0,85 | 0,85 | 0,49                    |
| Mengabaikan <i>incomplete data</i>                                  | 0                                | 0    | 0,84 | 0,84 | 0,84 | 0,84 | 0,85 | 0,84 | 0,63                    |
| Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan modus | 0,83                             | 0,83 | 0,87 | 0,87 | 0,91 | 0,91 | 0,9  | 0,9  | 0,88                    |

Sumber: Pengujian dan Analisis

Dari Tabel 5.18 dapat disimpulkan bahwa nilai rata - rata akurasi dari setiap perlakuan terhadap *incomplete data* pelatihan berturut-turut adalah 0.49, 0.63, dan 0,88 untuk tingkatan perlakuan terhadap *incomplete data* berturut-turut menghapus satu baris *incomplete data*, mengabaikan *incomplete data*, dan mengganti nilai *incomplete data* dengan *mean* atau modus.



**Gambar 5.19 Grafik Nilai Rata-Rata Akurasi Setiap Perlakuan Terhadap *Incomplete Data* Pelatihan**

Sumber: Pengujian dan Analisis

Gambar 5.19 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan perlakuan terhadap *incomplete data* pelatihan. Dari grafik dapat dilihat bahwa nilai rata - rata akurasi terbaik adalah ketika dilakukan perlakuan terhadap *incomplete data* pelatihan dengan mengganti nilai *incomplete data*. Hal ini dikarenakan ketika dilakukan pelatihan klasifikasi pada 405 data latih tidak ada nilai data yang hilang. Sedangkan ketika dilakukan perlakuan terhadap *incomplete data* dengan menghapus satu baris dan perlakuan terhadap *incomplete data* dengan mengabaikannya terjadi penurunan tingkat akurasi, ketika dilakukan pelatihan klasifikasi pada 405 data latih ada nilai data yang hilang. Dapat disimpulkan bahwa sistem dapat mendapatkan hasil terbaik ketika nilai *incomplete data* diganti dengan *mean* atau *modus*.

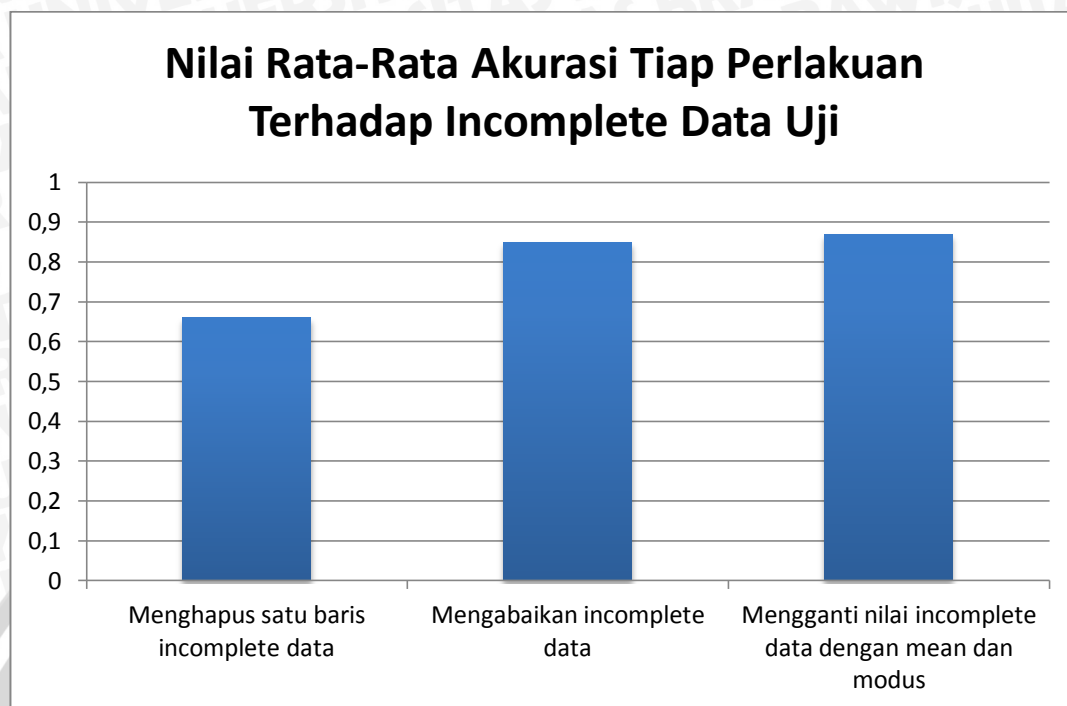
Nilai rata-rata akurasi dari setiap tingkat perlakuan terhadap *incomplete data* uji yang dihasilkan oleh sistem dapat dilihat pada Tabel 5.19.

**Tabel 5.19 Nilai Rata-Rata Akurasi Menurut Perlakuan Terhadap *Incomplete Data* Uji**

| Perlakuan Terhadap <i>Incomplete Data</i>                                  | Tingkatan <i>Incomplete Data</i> |      |      |      |      |      |      |      | Nilai Rata-Rata Akurasi |
|----------------------------------------------------------------------------|----------------------------------|------|------|------|------|------|------|------|-------------------------|
|                                                                            | 70%                              | 50%  | 25%  | 20%  | 15%  | 10%  | 5%   | 2%   |                         |
| Menghapus satu baris <i>incomplete data</i>                                | 0,26                             | 0,44 | 0,67 | 0,71 | 0,74 | 0,78 | 0,82 | 0,84 | 0,66                    |
| Mengabaikan <i>incomplete data</i>                                         | 0,84                             | 0,84 | 0,84 | 0,84 | 0,85 | 0,85 | 0,85 | 0,85 | 0,85                    |
| Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan <i>modus</i> | 0,86                             | 0,86 | 0,86 | 0,87 | 0,87 | 0,88 | 0,88 | 0,88 | 0,87                    |

Sumber: Pengujian dan Analisis

Dari Tabel 5.19 dapat disimpulkan bahwa nilai rata - rata akurasi dari setiap perlakuan terhadap *incomplete data* uji berturut-turut adalah 0.66, 0.85, dan 0,87 untuk tingkatan perlakuan terhadap *incomplete data* berturut-turut menghapus satu baris *incomplete data*, mengabaikan *incomplete data*, dan mengganti nilai *incomplete data* dengan *mean* atau *modus*.



**Gambar 5.20 Grafik Nilai Rata-Rata Akurasi Setiap Perlakuan Terhadap *Incomplete Data Uji***

Sumber: Pengujian dan Analisis

Gambar 5.20 adalah grafik yang menjelaskan tentang hasil pengujian akurasi sistem berdasarkan perlakuan terhadap *incomplete data* uji. Dari grafik dapat dilihat bahwa nilai rata - rata akurasi terbaik adalah ketika dilakukan perlakuan terhadap *incomplete data* uji dengan mengganti nilai *incomplete data*. Hal ini dikarenakan ketika dilakukan prediksi klasifikasi pada 174 data uji tidak ada nilai data yang hilang. Sedangkan ketika dilakukan perlakuan terhadap *incomplete data* dengan menghapus satu baris dan perlakuan terhadap *incomplete data* dengan mengabaikannya terjadi penurunan tingkat akurasi, ketika dilakukan prediksi klasifikasi pada 174 data uji ada nilai data yang hilang. Dapat disimpulkan bahwa sistem dapat mendapatkan hasil terbaik ketika nilai *incomplete data* diganti dengan *mean* atau *modus*.

## 5.2 Analisis

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian sistem klasifikasi *incomplete data* yang telah dilakukan. Proses analisis mengacu pada dasar teori sesuai dengan hasil pengujian yang didapatkan. Analisis dilakukan terhadap hasil pengujian di setiap tahap pengujian. Proses analisis yang dilakukan meliputi analisis hasil pengujian akurasi.

### 5.2.1 Analisis Hasil Pengujian Akurasi

Proses analisis terhadap hasil pengujian akurasi dilakukan dengan menghitung nilai akurasi dari klasifikasi data dengan algoritma *Voting Feature Interval-5*. Berdasarkan hasil klasifikasi data didapatkan nilai akurasi yang cukup tinggi yaitu sebesar 0,85. Sehingga dapat disimpulkan bahwa algoritma *Voting Feature Interval-5* dapat melakukan klasifikasi dengan baik terhadap data penyakit liver pada manusia.

Selain klasifikasi data yang lengkap, proses analisis juga dilakukan terhadap hasil pengujian akurasi dengan menghitung nilai rata-rata akurasi setiap perlakuan dari tiap tingkatan *incomplete data*. Dari tiga perlakuan *incomplete data* yang dihitung terdapat penurunan hasil pengujian pada perlakuan menghapus satu baris data yang hilang dan mengabaikan data yang hilang. Untuk mengetahui penyebab penurunan hasil pengujian maka penulis menghitung data yang diterima sistem saat pelatihan klasifikasi setelah melewati proses *preprocessing*. Data yang dihitung adalah data pelatihan yang *incomplete* dengan tingkatan sebesar 50%. Tabel 5.20 merupakan tabel jumlah data pelatihan klasifikasi yang diterima sistem.

**Tabel 5.20 Jumlah Data Pelatihan Klasifikasi**

| NO | Perlakuan <i>Incomplete Data</i>                                           | Jumlah Data Pelatihan Klasifikasi |
|----|----------------------------------------------------------------------------|-----------------------------------|
| 1  | Menghapus satu baris <i>incomplete data</i>                                | 1                                 |
| 2  | Mengabaikan <i>incomplete data</i>                                         | 405                               |
| 3  | Mengganti nilai <i>incomplete data</i> dengan <i>mean</i> dan <i>modus</i> | 405                               |

Sumber: Pengujian dan Analisis

Pada tabel 5.11 dilakukan perhitungan jumlah data pelatihan klasifikasi yang diterima oleh sistem. Hasil dari tabel di atas menunjukkan bahwa perlakuan 2 dan 3 mengembalikan jumlah seluruh baris data pelatihan yaitu sebesar 405 data, sedangkan perlakuan 1 hanya mengembalikan 1 data pelatihan.

Pada perlakuan 1 terjadi penurunan akurasi karena dengan menghapus satu baris, maka jumlah baris data akan semakin berkurang sehingga *interval* yang dibuat juga berbeda dengan data aslinya. Selain itu jumlah data yang terdapat dalam suatu *interval* tergantung dengan data yang ada. Hal ini mengakibatkan nilai *interval vote* dapat berbeda dari data aslinya. Pada perlakuan 1 jika letak data yang hilang minimal terdapat pada tiap baris maka tidak ada data yang dikembalikan untuk pelatihan klasifikasi karena sistem akan menghapus semua baris data. Sistem tidak dapat melakukan pelatihan klasifikasi jika data yang dikembalikan hanya sekelas saja, karena *endpoint* tidak akan terbentuk. Perlakuan 1 kurang baik dalam menangani masalah *incomplete data*, karena banyaknya baris data yang hilang dapat mengurangi informasi yang potensial untuk klasifikasi data. Hal ini dibuktikan dengan tingkat rata-rata akurasi sebesar 0,49 untuk klasifikasi *incomplete data* pelatihan dan tingkat rata-rata akurasi sebesar 0,66 untuk klasifikasi *incomplete data* uji.

Pada perlakuan 2 terjadi peningkatan dan penurunan akurasi karena adanya data yang hilang tetapi tidak ada baris yang dihapus sehingga *interval* yang dibuat berbeda dengan data aslinya. Selain itu jumlah data yang terdapat dalam suatu *interval* tergantung dengan data yang ada. Hal ini mengakibatkan

nilai *interval vote* dapat berbeda. Pada perlakuan 2 jika tidak terdapat *endpoint* dari salah satu *feature* untuk salah satu kelas tidak ada, maka data tidak dapat diproses untuk pelatihan klasifikasi. Perlakuan 2 cukup baik dalam menghadapi masalah *incomplete data*, karena banyaknya data yang hilang tetapi tidak hilang satu baris data, sehingga dapat mengurangi informasi yang potensial untuk klasifikasi data. Hal ini dibuktikan dengan tingkat rata - rata akurasi yang cukup tinggi yaitu sebesar 0,63 untuk klasifikasi *incomplete data* pelatihan dan tingkat rata-rata akurasi sebesar 0,85 untuk klasifikasi *incomplete data* uji.

Pada perlakuan 3 terjadi peningkatan akurasi karena data yang hilang telah diganti dengan mean atau modus, sehingga *interval* yang terbentuk tidak jauh beda dengan data yang asli. Perlakuan 3 merupakan perlakuan terbaik dalam menghadapi *incomplete data*, karena dapat mengganti nilai data yang hilang sehingga data yang penting dapat dipertahankan untuk klasifikasi dan *interval* yang terbentuk tidak jauh beda dengan data aslinya. Hal ini dibuktikan dengan tingkat rata - rata akurasi yang tinggi yaitu sebesar 0,88 untuk klasifikasi *incomplete data* pelatihan dan tingkat rata-rata akurasi sebesar 0,87 untuk klasifikasi *incomplete data* uji.

Sehingga dapat disimpulkan bahwa semakin banyak nilai data yang hilang maka semakin kecil juga tingkat akurasinya. Dalam menyelesaikan masalah *incomplete data* tersebut, dapat disimpulkan bahwa mengganti nilai data dengan *mean* atau modus dapat meningkatkan akurasi dari klasifikasi *incomplete data*, karena dapat mengganti nilai data yang hilang sehingga data yang penting dapat dipertahankan untuk klasifikasi dan *interval* yang terbentuk tidak jauh beda dengan data aslinya.

## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Aplikasi sistem Klasifikasi *Incomplete Data* Penyakit Liver pada Manusia dengan Menggunakan Algoritma *Voting Feature Interval-5* (VFI5) telah dibuat sesuai perancangan dan dapat digunakan sebagai sistem klasifikasi data penyakit liver.
2. Aplikasi sistem Klasifikasi *Incomplete Data* Penyakit Liver pada Manusia dengan Menggunakan Algoritma *Voting Feature Interval-5* (VFI5) telah dibuat sesuai perancangan dan dapat digunakan untuk meningkatkan akurasi hasil klasifikasi *incomplete data* penyakit liver dengan cara mengganti nilai *incomplete data* dengan mean atau modus. Hal ini berdasarkan hasil pengujian akurasi, nilai rata-rata akurasi untuk perlakuan terhadap *incomplete data* dengan cara mengganti nilai data dengan *mean* atau modus mendapatkan nilai sebesar 0,88 untuk klasifikasi *incomplete data* pelatihan dan nilai sebesar 0,87 untuk klasifikasi *incomplete data* uji.

#### 6.2 Saran

Saran yang diberikan untuk pengembangan penelitian selanjutnya, antara lain:

1. Dapat dilakukan pengembangan sistem klasifikasi data dengan mengkombinasikan algoritma *Voting Feature Interval-5* (VFI5) dengan algoritma atau metode yang lain.
2. Pada sisi aplikasi, sistem dapat dikembangkan menjadi aplikasi klasifikasi *incomplete data* untuk multi kelas.

3. Untuk pengembangan lebih lanjut, disarankan menggunakan bahasa pemrograman lain.



## DAFTAR PUSTAKA

- [AGR-13] Kamus Kesehatan. 2013. *Rasio Albumin-Globulin* | Kamus Kesehatan. <http://kamuskesehatan.com/arti/rasio-albumin-globulin/>. Diakses 6 Juli 2013 pukul 15.00 WIB
- [ALP-13] Kamus Kesehatan. 2013. *Fosfatase Alkali* | Kamus Kesehatan. <http://kamuskesehatan.com/arti/fosfatase-alkali/>. Diakses 6 Juli 2013 pukul 15.00 WIB
- [BER-04] Berry, M. J. A dan Linoff G. S. 2004. *Data Mining Techniques For Marketing, sales, Customer Relationship Management*. Second Edition. Wiley Publishing, Inc.
- [BUO-09] Buono, A., Ramdhany, D N., Kustiyo A., & Handharyani, E. 2009. *Diagnosis Gangguan Sistem Urinari Pada Anjing Dan Kucing Menggunakan VFI 5*. Jurnal Ilmu Komputer dan Informasi 2 (2): 86-94
- [DEM-97] Demiros, Gulsen dkk. 1997. *Non-Incremental Classification Learning Algorithms Based On Voting Feature Intervals*.
- [ENN-01] Ennett CM, Frize M, Walker CR. 2001. *Influence of Missing Values on Artificial Neural Network Performance*.
- [FAY-96] Fayyad, U. M. 1996. *Advances in Knowledge Discovery and Data mining*. Cambridge, MA: The MIT Press.
- [GUV-00] Güvenir HA dan Emeksiz N. 2000. *An Expert System for the Differential Diagnosis of Erythematous-Squamous Diseases*. Expert Systems with Applications 18(1): 43-49.
- [GUV-98] Güvenir HA, Demiröz G, dan İlter N. 1998. *Learning Differential Diagnosis of Erythematous-Squamous Diseases using Voting Feature*

- Intervals. Artificial Intelligence in Medicine* 13 (3): 147-165.
- [HAN-01] Han J dan Kamber M. 2001. *Data mining Concepts & Techniques*.
- [JUN-09] Juniawan, Indra. 2009. *Klasifikasi Dokumen Teks Berbahasa Indonesia Menggunakan Minor Component Analysis*.
- [KKD-98] Kumala, P., Komala, S., Santoso, A. H., Sulaiman, J. R., Rienita, Y. Dan Nuswantasari, D. 1998. *Kamus Saku Kedokteran Dorland*. Edisi ke-25. Jakarta: EGC.
- [KON-93] Kononenko, I. 1993. *Inductive And Bayesian Learning In Medical Diagnosis*. *Applied Artificial Intelligence* (7): 317-337
- [LAB-11] Labatut, Vincent dan Cherifi, Hocine. 2011. *Accuracy Measures for the Comparison of Classifiers*.
- [LAR-06] Larose D, T.. 2006. *Data mining Methods and Models*. Jhon Wiley & Sons, Inc. Hoboken New Jersey
- [LIN-09] Lin, Rong-Ho. 2009. *An intelligent model for liver disease diagnosis. Artificial Intelligence in Medicine*.
- [MAL-12] Malau, Arda Sariani. 2012. *Karakteristik Penderita Sirosis Hati yang Dirawat Inap di Rumah Sakit Friska Medan Tahun 2006-2010*.
- [MEI-05] Mei-Ling, Shyu. 2005. *Handling Missing Values Via Decomposition of the Conditioned Set*.
- [NOV-12] Novianti, Fourina Ayu, dkk. 2012. *Analisis Diagnosis Pasien Kanker Payudara Menggunakan Regresi Logistik dan Support Vector Machine (SVM) Berdasarkan Hasil Mamografi*. *Jurnal Sains Dan Seni ITS* 1 (1): 147-152
- [PRA-06] Pramudiono,I. 2006. *Apa itu Data mining?*.  
<http://datamining.japati.net/cgi-bin/indodm.cgi?bacaarsip%11555276>

[1&artikel](#). Diakses 12 Mei 2013 pukul 15.00 WIB

- [ROM-09] F. Romansyah, I.S. Sitanggang, S. Nurdianti. 2009. *Fuzzy Decision Tree Dengan Algoritma ID3 Pada Data Diabetes*. Internet Working Indonesia Journal 1 (2): 45-52
- [SAL-13] Salindeho, E. 2013. *Penyakit Liver*. [http://www.bmodtcenter.com/artiket\\_kesehatan.html](http://www.bmodtcenter.com/artiket_kesehatan.html). Diakses 6 Juli 2013 pukul 15.00 WIB
- [TUR-05] Turban, E., Aronson, J. E. & Liang, T. 2005. *Decision Support Systems and Intelligent Systems (Sistem Pendukung Keputusan dan Sistem Cerdas)*
- [WLE-10] World Life Expentacy. 2010. *World Rangking Total Deaths*. <http://www.worldlifeexpectancy.com/world-rankings-total-deaths>. Diakses 12 Mei 2013 pukul 15.00 WIB

