

BAB IV

IMPLEMENTASI

4.1 Lingkungan Implementasi

Pada bab ini dibahas mengenai implementasi sistem sesuai dengan analisis kebutuhan dan perancangan sistem sebelumnya. Beberapa hal yang perlu diperhatikan untuk memenuhi kebutuhan sistem, yaitu lingkungan perangkat keras dan perangkat lunak dari sistem.

4.1.1 Lingkungan Perangkat Keras

Pada penelitian ini pengembangan sistem menggunakan perangkat keras sebuah laptop Asus dengan spesifikasi sebagai berikut :

1. Processor Intel® Core™ i3-2370M CPU @ 2.40GHz 2.40GHz
2. RAM 4GB
3. Harddisk kapasitas 500GB

4.1.2 Lingkungan Perangkat Lunak

Pengembangan sistem diagnosa penyakit asma dalam penelitian ini menggunakan perangkat lunak sebagai berikut :

1. Sistem Operasi Windows 7 Ultimate
2. Bahasa Pemrograman Java
3. Netbeans IDE 7.0.1 dalam mengembangkan aplikasi
4. JDK 7.0 .1development kit untuk menulis program java
5. Ms. Excel 2007

4.2 Batasan – batasan Implementasi

Beberapa batasan – batasan dalam mengimplementasikan sistem adalah sebagai berikut :

1. Input yang diterima sistem adalah berupa check dari opsi 6 gejala yang dikeluarkan sistem dan dikodekan ke dalam nilai 0-1 pada program

2. Output yang diterima user yaitu berupa hasil dari klasifikasi data ke dalam 2 kategori diagnosa yaitu Asma Bronkial dan Bronkitis Akut.
3. Untuk output yang diterima user dengan dua hasil diagnosa sekaligus yaitu Asma Bronkial dan Bronkitis Akut, maka disarankan untuk *rongten* dada.
4. Pengembangan aplikasi menggunakan Netbeans IDE 7.0.1
5. Metode yang digunakan yaitu metode ID3 dan CBR
6. Atribut yang digunakan ada 6 yaitu :
 - Batuk
 - Dahak
 - Darah
 - Sesak
 - Nyeri dada
 - Demam

4.3 Implementasi Program

Pada sub bab ini akan dijelaskan implementasi proses untuk menghasilkan klasifikasi dengan menggunakan algoritma ID3 pada sistem CBR.

4.3.1 Implementasi Proses Retrieve dan Reuse

Pada proses *Retrieve* dapat terjadi jika pengetahuan ID3 sudah terbentuk. Sedangkan proses *Reuse* adalah menggunakan pohon keputusan yang telah dibentuk untuk mengklasifikasikan data pasien baru yang akan digunakan dengan menentukan kelas dari masing – masing data pasien yang ada. Pada *source code* 4.1 akan ditampilkan kode program proses Penentuan Kelas.

```
public String getSugestion(Pasien uji
    if (root==null) {
        return "Tree belum Terbentuk";
    }
    String hasil = "";
    Node current = root;
    while(true) {
        if (current==null)
```

```

        return "-";
    if(current.isLeaf)
        return current.kelas;
    if(uji.getKriteriaValue(current.id)==1) {
        current = current.kkan;
    } else if (uji.getKriteriaValue(current.id) == -
1) {
        current = current.kkir;
    } else {
        current = current.kten;
    }
}
}
}

```

Source Code 4.1 Proses Penentuan Kelas

4.3.2 Implementasi Proses Revise

Proses *Revise* adalah proses evaluasi dan konfirmasi dari pakar yang dimaksud untuk menentukan apakah pohon aturan sebelumnya telah dapat diterapkan untuk menyelesaikan kasus yang baru. Pada *source code* 4.2 akan ditampilkan kode program proses Revise.

```

public String getSugestion(Pasien uji){
    if(root==null){
        return "Tree belum Terbentuk";
    }
    String hasil = "";
    Node current = root;
    while(true){
        if(current==null)
            return "-";
        if(current.isLeaf)
            return current.kelas;
        if(uji.getKriteriaValue(current.id)==1) {
            current = current.kkan;
        }else if (uji.getKriteriaValue(current.id) == -

```

```

1) {
    current = current.kkir;
} else {
    current = current.kten;
}
}
}
}
}

```

Source Code 4.2 Proses Revise

4.3.3 Implementasi Proses Pemilihan Data Latih

Pada proses pemilihan data latih dilakukan untuk memilih data yang akan digunakan untuk proses pembentukan *tree* dengan format *.xls*. Pada *source code* 4.1 akan ditampilkan kode program proses pemilihan data pasien sebagai data latih untuk membentuk pohon aturan.

```

public static ArrayList<Pasien> bacaFileXLS (String
namafilename){
    File file = new File(namafilename);
    try {
        Workbook workbook = Workbook.getWorkbook(file);
        Sheet sheet = workbook.getSheet(0);
        ArrayList<Pasien> data = new
ArrayList<Pasien>();
        for(int i=0;i<sheet.getColumn(0).length;i++){
            Cell temp[] = sheet.getRow(i);
            ArrayList<Integer> kriteria = new
ArrayList<Integer>();
            for(int j=2;j<temp.length-1;j++)
kriteria.add((temp[j].getContents().equalsIgnoreCase("ya"))?
1:((temp[j].getContents().equalsIgnoreCase("-"))?-1:0));
            data.add(new
Pasien(temp[0].getContents(),temp[1].getContents(),temp[temp
.length-1].getContents(),kriteria));

```

```
    }  
    workbook.close();  
    return data;  
} catch (FileNotFoundException ex) {  
  
Logger.getLogger(FileProcessing.class.getName()).log(Level.S  
EVERE, null, ex);  
    } catch (IOException ex) {  
  
Logger.getLogger(FileProcessing.class.getName()).log(Level.S  
EVERE, null, ex);  
    } catch (BiffException ex) {  
  
Logger.getLogger(FileProcessing.class.getName()).log(Level.S  
EVERE, null, ex);  
    }  
    return null;  
}  
}
```

Source Code 4.3 Proses Pemilihan Data Pasien

4.3.4 Implementasi Proses Pembentukan Tree

Proses pembentukan tree pada sistem CBR terjadi pada proses Retain dimana hasil yang telah valid dari proses revise akan diambil sebagai pengetahuan baru dan disimpan sehingga pohon aturan akan berubah untuk selanjutnya digunakan menyelesaikan kasus baru yang memiliki kemiripan di masa mendatang. Tree baru yang terbentuk merupakan penggabungan dari data pasien baru dengan data pasien yang telah ada. Pada *source code* 4.4 akan ditampilkan kode program proses pembentukan Tree pada proses Retain di sistem CBR.

```
public void prosesID3(ArrayList<Pasien> DataPasien){  
    if(DataPasien.size()==0)  
        return ;  
}
```

```
jmlNode = 0;
Stack<Node>jalanCrita = new Stack<Node>();
int [] newCek = new int [banyak_kriteria];
double IGain [] = getIGain(DataPasien);
int idRoot = getRoot(IGain,newCek);
newCek[idRoot]=1;
jmlNode++;
Node root = new Node(idRoot, DataPasien,IGain);
root.setCeks(newCek);
this.root = root;
jalanCrita.push(root);
while(!jalanCrita.isEmpty()){
    Node temp = jalanCrita.pop();
    ArrayList<Pasien> data = temp.data;
    if(temp.isLeaf)continue;
    for(int i=-1;i<2;i++){
        ArrayList<Pasien> dataPasien = new
ArrayList<Pasien>();
        for(Pasien d : data){
            if(d.kriteria.get(temp.id)==i){
                dataPasien.add(d.clone());
            }
        }
        if(dataPasien.size()!=0){
            Node anak;
            Set<String> sekKelas =
jumlahKelas(dataPasien);
            jmlNode++;
            if(sekKelas.size()<=1){
                anak = new Node(dataPasien);
                for(String f : sekKelas)
                    anak.setLeaf(f);
            }else{
                double IGainTemp [] =
getIGain(dataPasien);
```

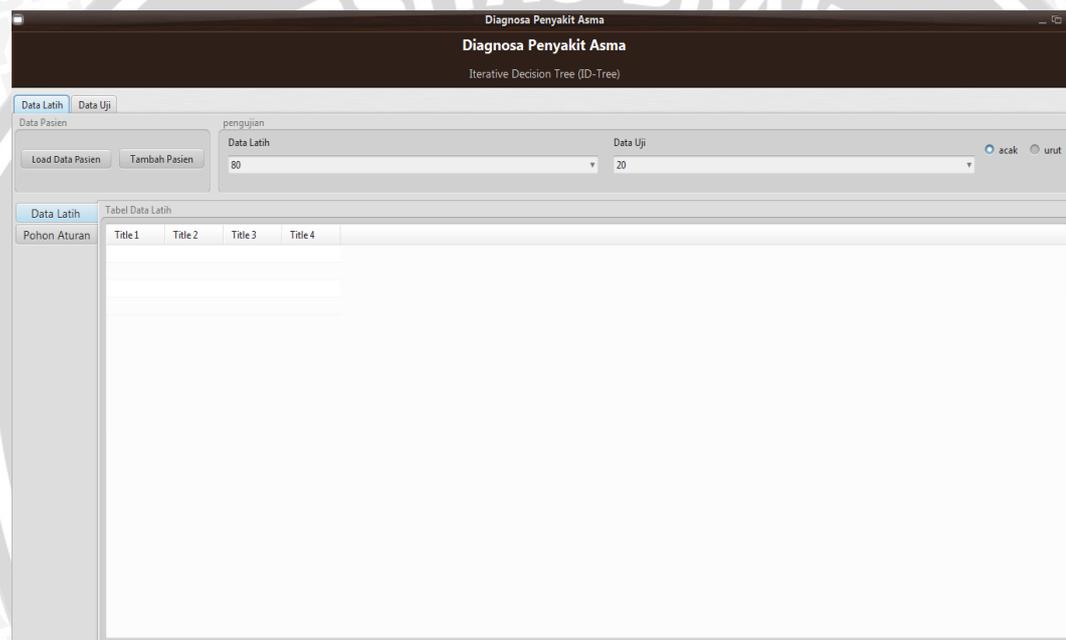
```
int a = getRoot(IGainTemp,temp.cek);
if(a!=-1){
    anak = new Node(a,
dataPasien,IGainTemp);
    anak.setCeks(temp.cek);
    anak.setCek(a);
    jalanCrita.push(anak);
}else{
    anak = new Node(dataPasien);
    String kelas = "";
    for(String f : sekKelas)
        kelas = kelas + f+", ";
    kelas=kelas.substring(0,
kelas.length()-2);
    anak.setLeaf(kelas);
}
}
if(i==1){
    temp.kkan = anak;
}else if(i==--1){
    temp.kkir = anak;
}else{
    temp.kten = anak;
}
}
}
}
```

Source Code 4.4 Proses Pembentukan *Tree* pada proses Retain di Sistem CBR

4.4 Implementasi Antarmuka

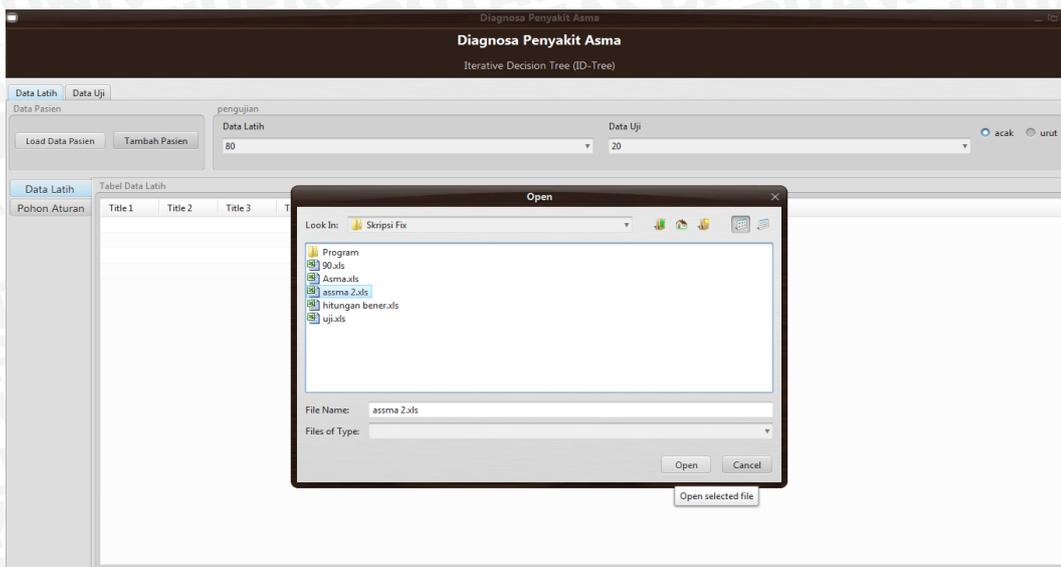
Implementasi sistem ini terdiri dari 3 bagian input yaitu input data pasien, input data latih, dan input data uji. Selain itu terdapat 3 bagian output yaitu pohon aturan yang terbentuk, hasil pengujian akurasi dan hasil klasifikasi diagnosa.

Pada Gambar 4.1 merupakan halaman pertama dari dari aplikasi diagnosa penyakit asma yang terdiri dari tombol data latih untuk proses pelatihan dan pembentukan *tree*, data uji untuk menentukan prediksi kelas pasien dan mengukur akurasi metode.

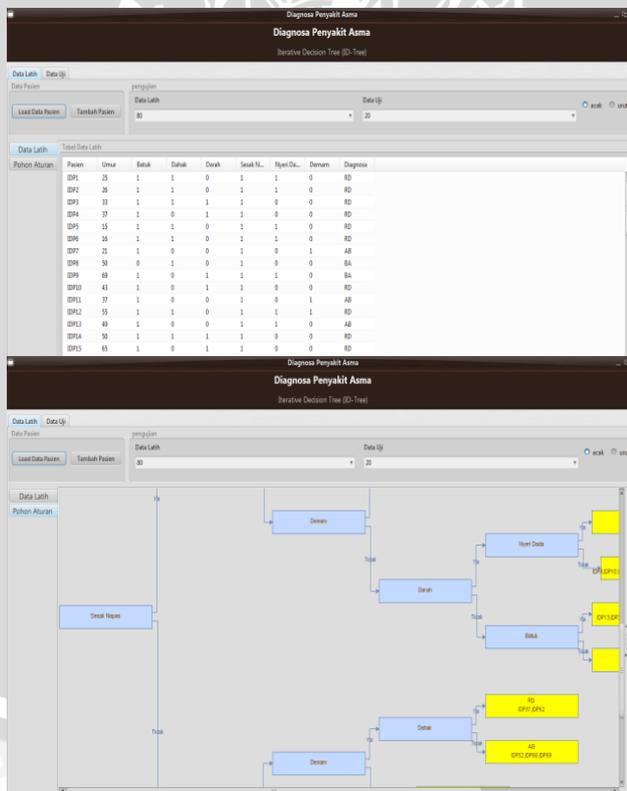


Gambar 4.1 Halaman Pertama

Proses awal yang dilakukan oleh sistem adalah membangun sebuah pohon aturan dari data yang ada dengan memilih data format .xls pada pilihan tombol “Load Data Pasien”. Kemudian Tabel Data Latih yang telah dipilih akan ditampilkan dengan menekan tombol “Data Latih” sehingga pohon aturan yang terbentuk dapat dilihat dengan menekan tombol “Pohon Aturan”. Gambar 4.2 menampilkan proses pemilihan data latih, Pohon aturan yang terbentuk akan ditampilkan pada gambar 4.3



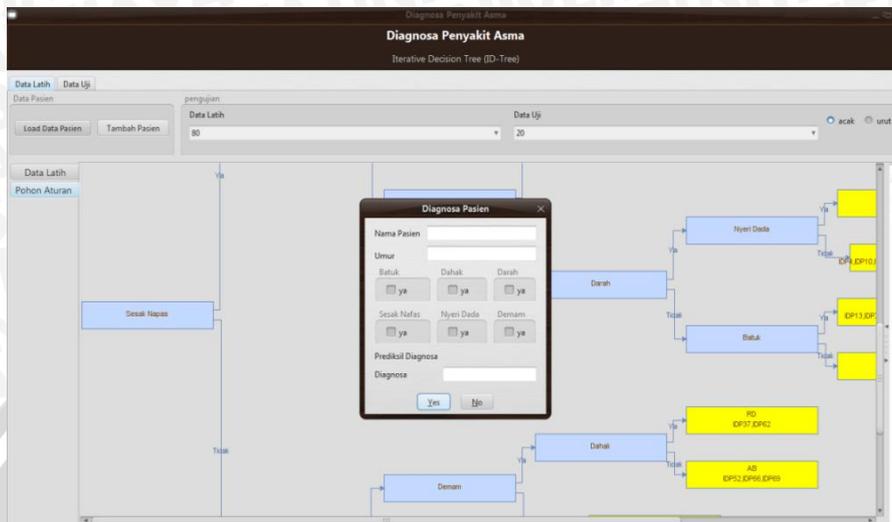
Gambar 4.2 Proses Pemilihan Data Latih



Gambar 4.3 Tampilan Pohon Aturan

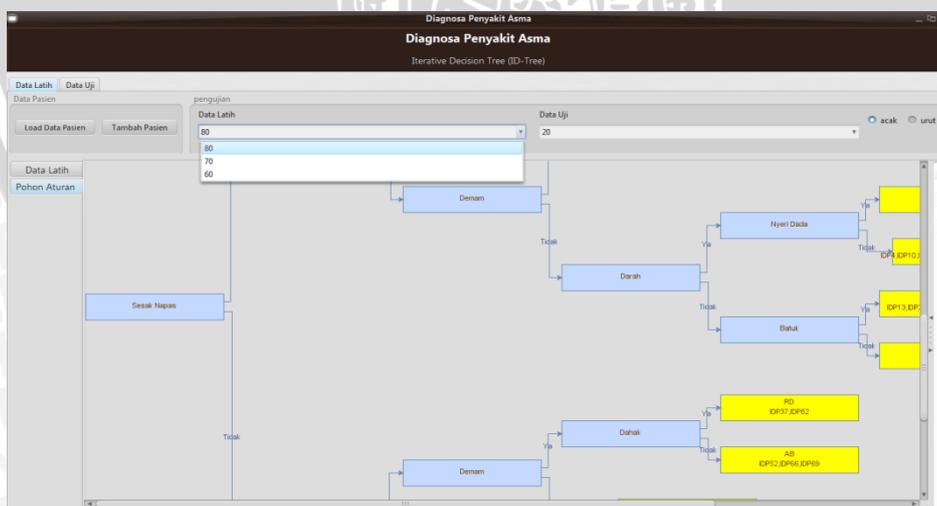
Data pasien baru dapat ditambahkan pada tombol “Tambah Pasien” dan dengan mengisi atribut gejala, nama pasien, umur dan prediksi diagnosa. Setelah itu

maka table data latih pasien juga akan bertambah dan akan ditampilkan hasil klasifikasi diagnosa dari sistem. Pada gambar 4.4 akan ditampilkan input data pasien baru.



Gambar 4.4 Tampilan Input Data Pasien Baru

Gambar 4.5 akan menampilkan pengujian dari sistem sesuai skenario pengujian yang telah dibuat. Pada proses ini dapat dipilih jumlah data uji sesuai dengan skenario masing – masing dan untuk tiap skenario pengujian akan ditampilkan beserta hasil akurasi.



Gambar 4.5 Tampilan Skenario Pengujian dan Akurasi