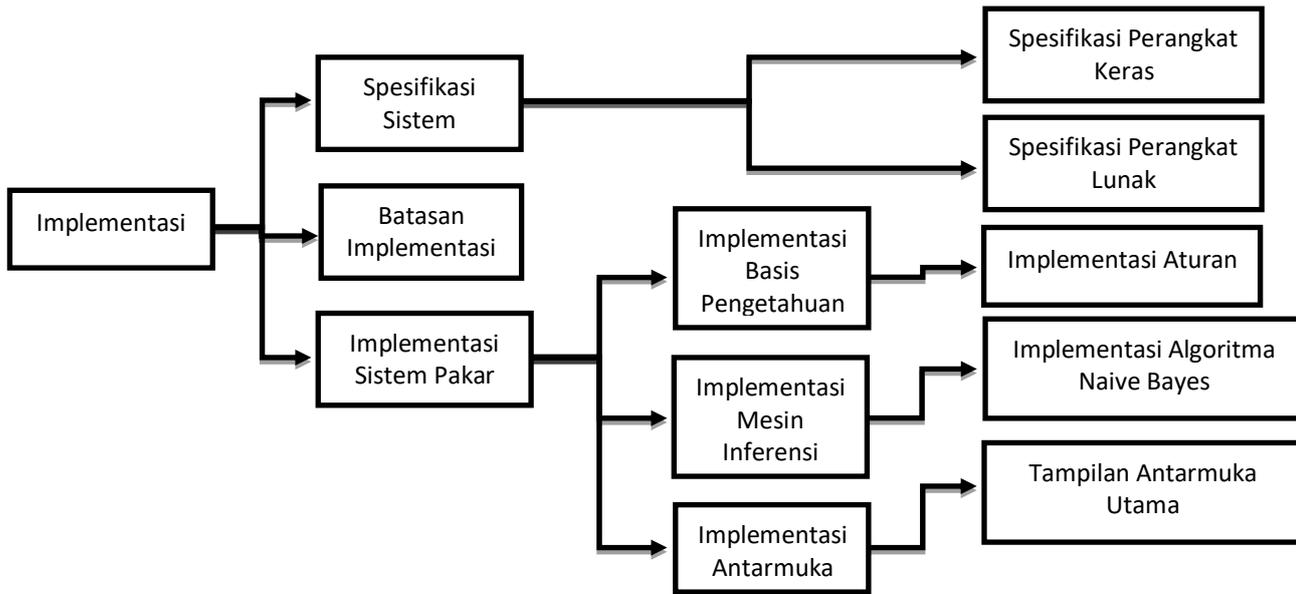


BAB 5 IMPLEMENTASI

Bab ini membahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari analisis kebutuhan dan proses perancangan perangkat lunak yang telah dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan - batasan dalam implementasi, implementasi algoritma pada program dan implementasi antarmuka, pohon implementasi sistem pakar dapat dilihat pada Gambar 5.1.



Gambar 0.1 Pohon Implementasi

1.1 Spesifikasi Sistem

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

1.1.1 Spesifikasi Perangkat Keras

Pengembangan Sistem Pakar Diagnosis Penyakit pada tanaman cengkeh menggunakan sebuah personal computer dengan spesifikasi perangkat keras yang ditunjukkan pada Tabel 5.1.

Tabel 0.1 Spesifikasi Perangkat Keras

| Nama Komponen | Spesifikasi |
|---------------|---------------------------------------|
| Prosesor | Intel(R) Corei3(R) CPU N2830 @2.16GHz |
| Memori(RAM) | 4,00 GB |

| | |
|--------------|--|
| Jenis Sistem | Sistem Operasi 64-bit, prosesor berbasis x64 |
| Harddisk | 500 GB |

1.1.2 Spesifikasi Perangkat Lunak

Pengembangan Sistem Pakar Diagnosis Penyakit pada tanaman cengkeh menggunakan sebuah personal computer dengan spesifikasi perangkat lunak yang ditunjukkan pada Tabel 5.2.

Tabel 0.2 Spesifikasi Perangkat Lunak

| | |
|--------------------|----------------------------------|
| Sistem Operasi | Microsoft Windows 10 Home 64-bit |
| Bahasa Pemrograman | Java |
| Tools Pemrograman | Android Studio |
| Emulator | HAXM |

1.2 Batasan Implementasi

Batasan dalam implementasi Sistem Pakar diagnosis Penyakit pada tanaman cengkeh adalah sebagai berikut :

1. Sistem yang dibangun berdasarkan ruang lingkup aplikasi android dengan menggunakan bahasa pemrograman java.
2. Data yang digunakan dalam sistem pakar disimpan dalam bentuk array.
3. Data yang digunakan berupa data-data gejala fisik pada tanaman cengkeh, data penyebab dari setiap gejala dan data penyakit yang menyerang tanaman cengkeh,
4. Masukan yang dilakukan oleh pengguna ke sistem berupa gejala fisik yang dialami tanaman cengkeh.
5. Keluaran dari sistem adalah salah satu dari 3 penyakit pada tanaman cengkeh.
6. Metode yang digunakan adalah naive bayes.
7. Semua pengguna dapat mengakses sistem tanpa harus melakukan login.
8. Semua pengguna memiliki hak akses yang sama.
9. Semua pengguna berhak mengakses menu yang ada pada sistem.
10. Data yang digunakan pada sistem permanen dan tidak dapat diubah lagi.

1.3 Implementasi Sistem Pakar

Hasil perancangan sistem yang telah diuraikan pada bab perancangan dijadikan sebagai acuan untuk melakukan implementasi Sistem Pakar diagnosis Penyakit pada tanaman cengkeh. Bagian-bagian yang akan diimplementasikan pada sistem pakar ini yaitu implementasi basis pengetahuan, implementasi mesin inferensi dan implementasi antarmuka.

1.3.1 Implementasi Basis Pengetahuan

Pada implementasi basis pengetahuan ini akan membahas tentang implementasi basis aturan yang digunakan dalam sistem pakar ini.

1.3.1.1 Implementasi Basis Aturan

Implementasi basis aturan ini mengacu pada sub bab basis pengetahuan. Basis aturan ini berdasarkan pada data yang diberikan oleh pakar yang digunakan untuk mendapatkan nilai prior dan likelihood.

1.3.2 Implementasi Mesin Inferensi

Sistem Pakar diagnosis Penyakit pada tanaman cengkeh ini memiliki tiga tahap proses perhitungan yaitu yang pertama adalah proses perhitungan nilai probabilitas prior yang didapatkan dari peluang suatu penyakit, tahap yang kedua yaitu proses perhitungan nilai probabilitas likelihood yang didapatkan dari peluang masing-masing penyakit sesuai masukan gejala dari pengguna dan yang terakhir adalah tahap perhitungan nilai probabilitas posterior yang didapatkan dari perkalian nilai probabilitas prior dan likelihood masing-masing penyakit.

1.3.2.1 Implementasi Algoritma Naive Bayes

Implementasi algoritma proses diagnosis penyakit tanaman cengkeh mengacu pada proses perhitungan naive bayes. Proses diagnosis pertama kali dilakukan dengan melihat kondisi tanaman cengkeh, kemudian pengguna memasukkan kondisi gejala fisik yang ada pada tanaman cengkeh untuk dilakukan proses diagnosis.

Implementasi Algoritma proses diagnosis dengan proses perhitungan naive bayes dapat dilihat pada Source Code 5.1 hingga 5.5.

```
double gejala1[] = {8,0,0};
double gejala2[] = {8,0,0};
double gejala3[] = {8,0,0};
double gejala4[] = {8,0,0};
double gejala5[] = {15,0,4};
double gejala6[] = {0,4,0};
double gejala7[] = {0,4,0};
double gejala8[] = {0,4,0};
double gejala9[] = {0,7,4};
double gejala10[] = {0,0,4};
double gejala11[] = {0,0,8};
double penyakit[] = {15,7,8};
double totaldata[] = {30};
```

Source Code 0.1 Inisialisasi Variabel

Implementasi algoritma yang pertama adalah inisialisasi variabel yang digunakan untuk menginisialisasikan data latih, data disimpan dalam bentuk array dimana variabel gejala1 hingga variabel gejala11 menyimpan nilai peluang munculnya gejala pada setiap penyakit, kemudian variabel penyakit berfungsi untuk menyimpan jumlah setiap penyakit pada data latih dan variabel total data berfungsi untuk menyimpan jumlah data latih yang digunakan.

```
double priorpenyakit[] = {0,0,0};
    for (int i = 0; i < 3; i++) {
        priorpenyakit[i] = penyakit[i] / totaldata[0];
    }
```

Source Code 0.2 Penghitungan Nilai Prior Setiap Penyakit

Implementasi algoritma yang kedua adalah perhitungan nilai prior setiap gejala dimana nilai prior ini memanggil nilai yang tersimpan pada array penyakit dan totaldata.

```
double like1[]={1,1,1};
    if (gejalaa1.isChecked()) {
        for (int i = 0; i < 3; i++) {
            like1[i] = gejala1[i] / penyakit[i];
        }
    }
double like2[]={1,1,1};
    if (gejalaa2.isChecked()) {
        for (int i = 0; i < 3; i++) {
            like2[i] = gejala2[i] / penyakit[i];
        }
    }
double like3[]={1,1,1};
    if (gejalaa3.isChecked()) {
        for (int i = 0; i < 3; i++) {
            like3[i] = gejala3[i] / penyakit[i];
        }
    }
double like4[]={1,1,1};
    if (gejalaa4.isChecked()) {
        for (int i = 0; i < 3; i++) {
            like4[i] = gejala4[i] / penyakit[i];
        }
    }
```

```

    }
}
double like5[]={1,1,1};
if (gejalaa5.isChecked()) {
    for (int i = 0; i < 3; i++) {
        like5[i] = gejala5[i] / penyakit[i];
    }
}
double like6[]={1,1,1};
if (gejalaa6.isChecked()) {
    for (int i = 0; i < 3; i++) {
        like6[i] = gejala6[i] / penyakit[i];
    }
}
double like7[]={1,1,1};
if (gejalaa7.isChecked()) {
    for (int i = 0; i < 3; i++) {
        like7[i] = gejala7[i] / penyakit[i];
    }
}
double like8[]={1,1,1};
if (gejalaa8.isChecked()) {
    for (int i = 0; i < 3; i++) {
        like8[i] = gejala8[i] / penyakit[i];
    }
}
double like9[]={1,1,1};
if (gejalaa9.isChecked()) {
    for (int i = 0; i < 3; i++) {
        like9[i] = gejala9[i] / penyakit[i];
    }
}
double like10[]={1,1,1};
if (gejalaa10.isChecked()) {
    for (int i = 0; i < 3; i++) {

```

```

        like10[i] = gejala10[i] / penyakit[i];
    }
}
double like11[]={1,1,1};
if (gejalaa11.isChecked()) {
    for (int i = 0; i < 3; i++) {
        like11[i] = gejala11[i] / penyakit[i];
    }
}

double liketotal[] = {0,0,0};
for (int i = 0; i < 3; i++) {
    liketotal[i] = like1[i] * like2[i] * like3[i] * like4[i] * like5[i] * like6[i] * like7[i]
* like8[i] * like9[i] * like10[i] *
        like11[i];
}

```

Source Code 0.3 Penghitungan Nilai Likelihood Masing-Masing Gejala dan Penyakit

Implementasi Algoritma selanjutnya adalah perhitungan nilai probabilitas likelihood, perhitungan probabilitas likelihood dibagi menjadi dua tahap yaitu perhitungan probabilitas likelihood untuk setiap gejala dan perhitungan likelihood untuk setiap penyakit.

```

double posterirorpenyakit[]={0,0,0};
for (int i = 0; i < 3; i++) {
    posterirorpenyakit[i] = priorpenyakit[i] * liketotal[i];
}

```

Source Code 0.4 Penghitungan Nilai Posterior Masing-Masing Penyakit

Implementasi Algoritma selanjutnya adalah implementasi algoritma untuk menghitung probabilitas posterior setiap penyakit dimana setelah didapatkan probabilitas prior dan likelihood untuk setiap penyakit kemudian kedua probabilitas dikalikan untuk mendapatkan nilai probabilitas posterior.

```

if (posterirorpenyakit[0] > posterirorpenyakit[1] && posterirorpenyakit[0] >
posterirorpenyakit[2] ) {
    setContentView(R.layout.bakteri);
} else if (posterirorpenyakit[1] > posterirorpenyakit[0] &&
posterirorpenyakit[1] > posterirorpenyakit[2]) {
    setContentView(R.layout.cacar);
}

```

```

    } else if (posteriorpenyakit[2] > posteriorpenyakit[1] &&
posteriorpenyakit[2] > posteriorpenyakit[0] ) {
        setContentView(R.layout.akar);
    } else {
        setContentView(R.layout.nothing);
    }
}

```

Source Code 0.5 Pencarian Nilai Posterior Terbesar

Implementasi algoritma yang terakhir adalah implementasi algoritma untuk mencari nilai probabilitas posterior terbesar dimana penyakit yang memiliki probabilitas posterior yang terbesar yang dijadikan hasil diagnosis.

1.3.3 Implementasi Antarmuka

Antarmuka sistem pakar diagnosis penyakit tanaman cengkeh ini digunakan oleh pengguna untuk berinteraksi dengan perangkat lunak. Antarmuka halaman terbagi menjadi 5 halaman yaitu antarmuka halaman utama, antarmuka halaman daftar penyakit, antarmuka halaman cara penggunaan, antarmuka halaman diagnosis dan antarmuka halaman hasil diagnosis.

1.3.3.1 Tampilan Antarmuka Halaman Utama

Tampilan antarmuka halaman utama yang ditunjukkan pada Gambar 5.2 merupakan halaman awal ketika pengguna membuka sistem, pada halaman ini terdapat tiga tombol atau menu yaitu diagnosa, daftar penyakit dan cara penggunaan.



Gambar 0.2 Tampilan Antarmuka Halaman Utama

1.3.3.2 Tampilan Antarmuka Daftar Penyakit

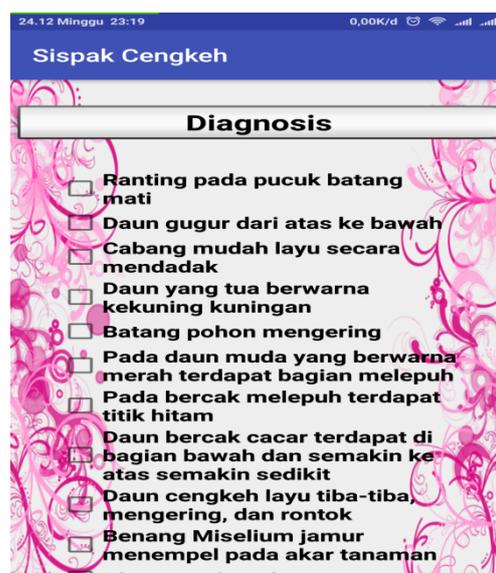
Tampilan antarmuka tentang penyakit yang ditunjukkan pada Gambar 5.3 merupakan tampilan halaman yang muncul ketika pengguna mengklik tombol daftar penyakit pada halaman utama, pada halaman ini pengguna akan mendapatkan daftar penyakit yang menyerang tanaman cengkeh dan gejala-gejalanya.



Gambar 0.3 Tampilan Antarmuka Halaman Daftar Gejala dan penyakit

1.3.3.3 Tampilan Antarmuka Diagnosis

Tampilan antarmuka diagnosis yang ditunjukkan pada Gambar 5.4 merupakan tampilan halaman yang akan muncul ketika pengguna memilih menu Diagnosis pada halaman utama dimana pada halaman ini akan ditampilkan list checkbox gejala penyakit tanaman cengkeh dan juga sebuah tombol diagnosis.



1.3.3.4 Tampilan Antarmuka Hasil Diagnosis

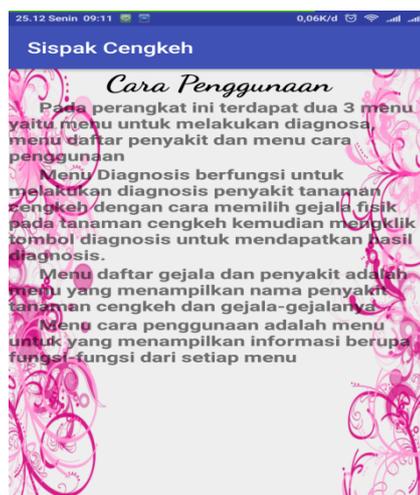
Tampilan antarmuka hasil diagnosis yang ditunjukkan pada Gambar 5.5 merupakan tampilan halaman yang muncul ketika pengguna mengklik tombol diagnosis pada halaman diagnosis, pada halaman ini pengguna akan mendapatkan informasi hasil perhitungan dan diagnosis penyakit tanaman cengkeh.



Gambar 0.5 Tampilan Antarmuka Halaman Hasil Diagnosis

1.3.3.5 Tampilan Antarmuka Halaman Cara Penggunaan

Tampilan antarmuka cara penggunaan yang ditunjukkan pada Gambar 5.6 merupakan tampilan halaman yang muncul ketika pengguna mengklik tombol cara penggunaan pada halaman utama, pada halaman ini pengguna akan mendapatkan informasi fungsi setiap menu pada perangkat lunak..



Gambar 0.6 Tampilan Antarmuka Halaman Cara penggunaan