# BAB 5 IMPLEMENTASI

Pada bab ini akan membahas mengenai implementasi perangkat lunak berdasarkan hasil analisis kebutuhan dan perancangan perangkat lunak yang telah dibuat. Pembahasan mengenai implementasi perangkat lunak terdiri dari lingkungan implementasi, implementasi algoritme dan implementasi antarmuka pengguna.

## 5.1 Lingkungan Implementasi

Pada Bab 4 telah diuraikan hasil analisis kebutuhan dan perancangan sistem pakar yang menjadi acuan dalam melakukan implementasi menjadi sebuah sistem yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat lunak dan perangkat keras.

### 5.1.1 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan aplikasi pada penelitian ini dapat dilihat pada Tabel 5.1

**Tabel 5.1 Spesifikasi Perangkat Lunak**

| **Perangkat Lunak** | **Keterangan** |
|---|---|
| Sistem Operasi | Microsoft Windows 10 Pro |
| IDE | Android Studio |
| Bahasa Pemrograman | Java |

### 5.1.2 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pengembangan aplikasi pada penelitian ini dapat dilihat pada Tabel 5.2

**Tabel 5.2 Spesifikasi Perangkat Keras**

| **Perangkat Keras** | **Keterangan** |
|---|---|
| Processor | Intel Core i3-4005U CPU @ 1.70GHz  1.70GHz |
| Memory (RAM) | 4.00 GB |
| Laptop | ASUS X540L |
| Smartphone | Samsung Galaxy J7 Prime |

## 5.2 Implementasi Algoritme

Pada sub-bab ini akan dijelaskan mengenai proses implementasi algoritme sistem pakar diagnosis penyakit sapi berbasis android. Implementasi algoritme ini

dibangun berdasarkan perancangan yang sudah dirancang pada sub bab sebelumnya menggunakan bahasa pemrograman Java.

### 5.2.1 Implementasi Perhitungan Jumlah Kemunculan Penyakit

Proses ini merupakan proses perhitungan berapa banyak muncul masing-masing penyakit pada data latih. Proses ini menghitung secara berurutan mulai dari penyakit 1 (Anthrax) sampai dengan penyakit 5 (Thymphani). Jumlah masing-masing kemunculan penyakit akan disimpan kedalam variable yang nanti akan digunakan dalam proses selanjutnya. Kode program dari perhitungan jumlah kemunculan penyakit dapat dilihat pada Tabel 5.3.

**Tabel 5.3 Kode Program Perhitungan Jumlah Kemunculan Penyakit**

| No. | Kode Program |
|---|---|
| 1 | `//Set Jumlah Penyakit` |
| 2 | `private  void setDataJumlahPenyakit(){` |
| 3 | `this.dataJumlahPenyakit.put("P1",30);` |
| 4 | `this.dataJumlahPenyakit.put("P2",27);` |
| 5 | `this.dataJumlahPenyakit.put("P3",30);` |
| 6 | `this.dataJumlahPenyakit.put("P4",22);` |
| 7 | `this.dataJumlahPenyakit.put("P5",15);` |
| 8 | `this.dataJumlahPenyakit.put("P6",39);` |
| 9 | `this.dataJumlahPenyakit.put("P7",17);` |
| 10 | `this.dataJumlahPenyakit.put("P8",20);` |
| 11 | `this.dataJumlahPenyakit.put("P9",26);` |
| 12 | `this.dataJumlahPenyakit.put("P10",5);` |
| 13 | `this.dataJumlahPenyakit.put("P11",20);` |
| 14 | `this.dataJumlahPenyakit.put("P12",5);` |
| 15 | `this.dataJumlahPenyakit.put("P13",3);` |
| 16 | `this.dataJumlahPenyakit.put("P14",21);` |
| 17 | `this.dataJumlahPenyakit.put("P15",16);` |
| 18 | `this.dataJumlahPenyakit.put("P16",8);` |
| 19 | `}` |

Penjelasan kode program proses perhitungan jumlah penyakit:
Baris 2 – 19 : menginisiasi jumlah penyakit yang ada

### 5.2.2 Implementasi Perhitungan Jumlah Kemunculan Gejala Masukan

Proses ini merupakan proses perhitungan berapa banyak gejala masukan muncul pada masing-masing penyakit pada data uji. Proses ini menghitung jumlah kemunculan gejala masukan pada masing-masing penyakit pada data latih. Proses perhitungan pertama pada sistem akan memeriksa gejala mana yang dimasukkan oleh pengguna, setelah gejala masukan selesai diperiksa akan masuk ke proses perhitungan gejala masukan pada masing-masing penyakit. Jumlah kemunculan gejala masukan yang nanti akan digunakan dalam proses selanjutnya. Kode program dari perhitungan jumlah gejala masukan penyakit dapat dilihat pada Tabel 5.4.

**Tabel 5.4 Kode Program Perhitungan Jumlah Kemunculan Gejala Masukan**

| No. | Kode Program |
|---|---|

```
1  //Set Data Gejala Masukkan
2  if (cbxPertanyaan1.isChecked()){
3                      inputan.put("G1",1);
4                  }else{
5                      inputan.put("G1",0);
6                  }
7
8                  if (cbxPertanyaan2.isChecked()){
9                      inputan.put("G2",1);
10                 }else{
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28                     inputan.put("G2",0);
29                 }
30
31                 if (cbxPertanyaan3.isChecked()){
32                     inputan.put("G3",1);
33                 }else{
34                     inputan.put("G3",0);
35                 }
36
37                 if (cbxPertanyaan4.isChecked()){
38                     inputan.put("G4",1);
39                 }else{
40                     inputan.put("G4",0);
41                 }
42
43                 if (cbxPertanyaan5.isChecked()){
44                     inputan.put("G5",1);
45                 }else{
46                     inputan.put("G5",0);
47                 }
48
49                 if (cbxPertanyaan6.isChecked()){
50                     inputan.put("G6",1);
51                 }else{
52                     inputan.put("G6",0);
53                 }
54
55                 if (cbxPertanyaan7.isChecked()){
56                     inputan.put("G7",1);
57                 }else{
58                     inputan.put("G7",0);
59                 }
```

```
60
61                  if (cbxPertanyaan8.isChecked()){
62                      inputan.put("G8",1);
63                  }else{
64                      inputan.put("G8",0);
65                  }
66
67                  if (cbxPertanyaan9.isChecked()){
68                      inputan.put("G9",1);
69                  }else{
70                      inputan.put("G9",0);
71                  }
72
73                  if (cbxPertanyaan10.isChecked()){
74                      inputan.put("G10",1);
75                  }else{
76                      inputan.put("G10",0);
77                  }
78
79                  if (cbxPertanyaan11.isChecked()){
80                      inputan.put("G11",1);
81                  }else{
82                      inputan.put("G11",0);
83                  }
84
85                  if (cbxPertanyaan12.isChecked()){
86                      inputan.put("G12",1);
87                  }else{
88                      inputan.put("G12",0);
89                  }
90
91                  if (cbxPertanyaan13.isChecked()){
92                      inputan.put("G13",1);
93                  }else{
94                      inputan.put("G13",0);
95                  }
96
97                  if (cbxPertanyaan14.isChecked()){
98                      inputan.put("G14",1);
99                  }else{
100                     inputan.put("G14",0);
101                 }
102
103                 if (cbxPertanyaan15.isChecked()){
104                     inputan.put("G15",1);
105                 }else{
106                     inputan.put("G15",0);
107                 }
108
109                 if (cbxPertanyaan16.isChecked()){
110                     inputan.put("G16",1);
111                 }else{
112                     inputan.put("G16",0);
113                 }
114
115                 if (cbxPertanyaan17.isChecked()){
116                     inputan.put("G17",1);
117                 }else{
118                     inputan.put("G17",0);
```

```
119                     }
120
121                     if (cbxPertanyaan18.isChecked()){
123                         inputan.put("G18",1);
124                     }else{
125                         inputan.put("G18",0);
126                     }
127
128                     if (cbxPertanyaan19.isChecked()){
129                         inputan.put("G19",1);
130                     }else{
131                         inputan.put("G19",0);
132                     }
133
134                     if (cbxPertanyaan20.isChecked()){
135                         inputan.put("G20",1);
136                     }else{
137                         inputan.put("G20",0);
138                     }
139
140                     if (cbxPertanyaan21.isChecked()){
141                         inputan.put("G21",1);
142                     }else{
143                         inputan.put("G21",0);
144                     }
145
146                     if (cbxPertanyaan22.isChecked()){
147                         inputan.put("G22",1);
148                     }else{
149                         inputan.put("G22",0);
150                     }
151
152                     if (cbxPertanyaan23.isChecked()){
153                         inputan.put("G23",1);
154                     }else{
155                         inputan.put("G23",0);
156                     }
157
158                     if (cbxPertanyaan24.isChecked()){
159                         inputan.put("G24",1);
160                     }else{
161                         inputan.put("G24",0);
162                     }
163
164                     if (cbxPertanyaan25.isChecked()){
165                         inputan.put("G25",1);
166                     }else{
167                         inputan.put("G25",0);
168                     }
169
170                     if (cbxPertanyaan26.isChecked()){
171                         inputan.put("G26",1);
172                     }else{
173                         inputan.put("G26",0);
174                     }
175
176                     if (cbxPertanyaan27.isChecked()){
178                         inputan.put("G27",1);
179                     }else{
```

```
180                     inputan.put("G27",0);
181                 }
182 
183                 if (cbxPertanyaan28.isChecked()){
184                     inputan.put("G28",1);
185                 }else{
186                     inputan.put("G28",0);
187                 }
189 
190                 if (cbxPertanyaan29.isChecked()){
191                     inputan.put("G29",1);
192                 }else{
193                     inputan.put("G29",0);
194                 }
195 
196                 if (cbxPertanyaan30.isChecked()){
197                     inputan.put("G30",1);
198                 }else{
199                     inputan.put("G30",0);
200                 }
201 
202                 if (cbxPertanyaan31.isChecked()){
203                     inputan.put("G31",1);
204                 }else{
205                     inputan.put("G31",0);
206                 }
207 
208                 if (cbxPertanyaan32.isChecked()){
209                     inputan.put("G32",1);
210                 }else{
211                     inputan.put("G32",0);
212                 }
213 
214                 if (cbxPertanyaan33.isChecked()){
215                     inputan.put("G33",1);
216                 }else{
217                     inputan.put("G33",0);
218                 }
219 
220                 if (cbxPertanyaan34.isChecked()){
221                     inputan.put("G34",1);
222                 }else{
223                     inputan.put("G34",0);
224                 }
225 
226                 if (cbxPertanyaan35.isChecked()){
227                     inputan.put("G35",1);
228                 }else{
229                     inputan.put("G35",0);
230                 }
231 
232                 if (cbxPertanyaan36.isChecked()){
233                     inputan.put("G36",1);
234                 }else{
235                     inputan.put("G36",0);
236                 }
237 
238                 if (cbxPertanyaan37.isChecked()){
239                     inputan.put("G37",1);
```

```
240                 }else{
241                     inputan.put("G37",0);
242                 }
243 
244                 if (cbxPertanyaan38.isChecked()){
245                     inputan.put("G38",1);
246                 }else{
247                     inputan.put("G38",0);
248                 }
249 
250                 if (cbxPertanyaan39.isChecked()){
251                     inputan.put("G39",1);
252                 }else{
253                     inputan.put("G39",0);
254                 }
255 
256                 if (cbxPertanyaan40.isChecked()){
257                     inputan.put("G40",1);
258                 }else{
259                     inputan.put("G40",0);
260                 }
261 
262                 if (cbxPertanyaan41.isChecked()){
263                     inputan.put("G41",1);
264                 }else{
265                     inputan.put("G41",0);
266                 }
267 
268                 if (cbxPertanyaan42.isChecked()){
269                     inputan.put("G42",1);
270                 }else{
271                     inputan.put("G42",0);
272                 }
273 
274                 if (cbxPertanyaan43.isChecked()){
275                     inputan.put("G43",1);
276                 }else{
277                     inputan.put("G43",0);
278                 }
279 
280                 if (cbxPertanyaan44.isChecked()){
281                     inputan.put("G44",1);
282                 }else{
283                     inputan.put("G44",0);
284                 }
285 
286                 if (cbxPertanyaan45.isChecked()){
287                     inputan.put("G45",1);
288                 }else{
289                     inputan.put("G45",0);
290                 }
291 
292                 if (cbxPertanyaan46.isChecked()){
293                     inputan.put("G46",1);
294                 }else{
295                     inputan.put("G46",0);
296                 }
297 
298                 if (cbxPertanyaan47.isChecked()){
```

```
299                     inputan.put("G47",1);
300                 }else{
301                     inputan.put("G47",0);
302                 }
303 
304                 if (cbxPertanyaan48.isChecked()){
305                     inputan.put("G48",1);
306                 }else{
307                     inputan.put("G48",0);
308                 }
309 
310                 if (cbxPertanyaan49.isChecked()){
311                     inputan.put("G49",1);
312                 }else{
313                     inputan.put("G49",0);
314                 }
315 
316                 if (cbxPertanyaan50.isChecked()){
317                     inputan.put("G50",1);
318                 }else{
319                     inputan.put("G50",0);
320                 }
321                 if (cbxPertanyaan51.isChecked()){
322                     inputan.put("G51",1);
324                 }else{
325                     inputan.put("G51",0);
326                 }
327 
328                 if (cbxPertanyaan52.isChecked()){
329                     inputan.put("G52",1);
330                 }else{
331                     inputan.put("G52",0);
332                 }
333 
334                 if (cbxPertanyaan53.isChecked()){
335                     inputan.put("G53",1);
336                 }else{
337                     inputan.put("G53",0);
338                 }
339 
340                 if (cbxPertanyaan54.isChecked()){
341                     inputan.put("G54",1);
342                 }else{
343                     inputan.put("G54",0);
344                 }
345 
346                 if (cbxPertanyaan55.isChecked()){
347                     inputan.put("G55",1);
348                 }else{
349                     inputan.put("G55",0);
350                 }
351 
                    if (cbxPertanyaan56.isChecked()){
352                     inputan.put("G56",1);
353                 }else{
354                     inputan.put("G56",0);
355                 }
356 
357                 if (cbxPertanyaan57.isChecked()){
```

```
358                     inputan.put("G57",1);
359                 }else{
360                     inputan.put("G57",0);
361                 }
362 
363                 if (cbxPertanyaan58.isChecked()){
364                     inputan.put("G58",1);
365                 }else{
366                     inputan.put("G58",0);
367                 }
368 
369                 if (cbxPertanyaan59.isChecked()){
370                     inputan.put("G59",1);
371                 }else{
372                     inputan.put("G59",0);
373                 }
374 
375                 if (cbxPertanyaan60.isChecked()){
376                     inputan.put("G60",1);
377                 }else{
378                     inputan.put("G60",0);
379                 }
380 
381                 if (cbxPertanyaan61.isChecked()){
382                     inputan.put("G61",1);
383                 }else{
384                     inputan.put("G61",0);
385                 }
386 
387                 if (cbxPertanyaan62.isChecked()){
388                     inputan.put("G62",1);
389                 }else{
390                     inputan.put("G62",0);
391                 }
392 
393                 if (cbxPertanyaan63.isChecked()){
394                     inputan.put("G63",1);
395                 }else{
396                     inputan.put("G63",0);
397                 }
398 
399                 if (cbxPertanyaan64.isChecked()){
400                     inputan.put("G64",1);
401                 }else{
402                     inputan.put("G64",0);
403                 }
404 
405                 if (cbxPertanyaan65.isChecked()){
406                     inputan.put("G65",1);
407                 }else{
408                     inputan.put("G65",0);
409                 }
```

Penjelasan Kode program proses perhitungan jumlah kemunculan gejala masukan:

Baris 2 – 392 : proses inputan gejala yang dimasukkan oleh pengguna yang nantinya akan dihitung ke proses selanjutnya

### 5.2.3 Implementasi Perhitungan Nilai *Prior*

Proses ini merupakan proses perhitungan nilai probabilitas *prior*. Proses menghitung jumlah kemunculan gejala masukan pada masing-masing opt pada data latih. Proses perhitungan nilai probabilitas prior dapat diluhat pada baris 2-7 Tabel 5.5. Nilai probabilitas didapat dari hasil pembagian jumlah kemunculan masing-masing penyakit pada data latih dengan jumlah keseluruhan data pada data uji. Hasil dari proses perhitungan nilai probabilitas *prior* akan disimpan kedalam variabel **dataPrior**. Kode program dari perhitungan nilai *prior* dapat dilihat pada Tabel 5.5.

**Tabel 5.5 Kode Program Perhitungan Nilai *Prior***

| No. | Kode Program |
|---|---|
| 1 | `//Hitung Nilai Prior` |
| 2 | `private void prior() {` |
| 3 | |
| 4 | `int total=0;` |
| 5 | `for (int value : this.dataJumlahPenyakit.values()){` |
| 6 | `total +=value;` |
| 7 | `}` |
| 8 | `for (Map.Entry<String, Integer> entry:` |
| 9 | `this.dataJumlahPenyakit.entrySet()){` |
| 10 | `Double nilaiPrior=` |
| 11 | `entry.getValue().doubleValue()/total;` |
| 12 | |
| 13 | `this.dataPrior.put(entry.getKey().toString(),nilaiPrior);` |
| 14 | `}` |
| 15 | `}` |

Penjelasan kode program proses perhitungan nilai *prior*:
Baris 2 – 7 : menghitung nilai *prior* masing-masing penyakit yang muncul pada data latih.

Baris 8 – 14 : memproses nilai *prior* dan sudah terhitung untuk dilakukan perhitungan berikutnya yang masing-masong penyakitnya muncul pada data latih.

### 5.2.4 Implementasi Perhitungan Nilai *Likelihood*

Proses ini merupakan proses perhitungan nilai probabilitas *likelihood*. Proses ini menghitung berapa banyak jumlah setiap gejala yang dimasukkan pada masing-masing jumlah data penyakit pada data latih. Kemudian jumlah masing-masing gejala masukan pada masing-masing opt dilakukan perhitungan nilai probabilitas *likelihood* dengan membagi jumlah masing-maisng gejala masukan. Proses perhitungan nilai probabilitas *likelihood* dapat dilihat pada baris 2-20 di Tabel 5.6. Setelah didapatkan nilai masing-masing gejala masukan pada masing-masing opt akan dilakukan perhitungan nilai total *likelihood* yaitu mengalikan semua nilai probabilitas *likelihood* gejala masukan pada masing-masing jumlah data penyakit.

Proses perhitungan nilai total *likelihood* dapat didlihat pada baris 21-26 di Tabel 5.6. Kode program dari perhitungan nilai *likelihood* dapat dilihat pada Tabel 5.6.

**Tabel 5.6 Kode Program Perhitungan Nilai *Likelihood***

| No. | Kode Program |
|---|---|
| 1 | `//Hitung Nilai Likelihood & Posterior` |
| 2 | `private void likelihood(){` |
| 3 | `for (Map.Entry<String, Integer> entry:` |
| 4 | `this.dataJumlahPenyakit.entrySet()){` |
| 5 | `double` |
| 6 | `tPosterior=this.dataPrior.get(entry.getKey());` |
| 7 | `for (Map.Entry<String, Integer> entryInput:` |
| 8 | `this.inputan.entrySet()){` |
| 9 | `if (entryInput.getValue()==1){` |
| 10 | `double tLikelihood` |
| 11 | `=getPembilang(entry.getKey(),entryInput.getKey())/` |
| 12 | `entry.getValue().doubleValue();` |
| 13 | `if (tLikelihood==0.0){` |
| 14 | `tLikelihood=0.001;` |
| 15 | `}` |
| 16 | `if (tPosterior==0){` |
| 17 | `tPosterior=tLikelihood;` |
| 18 | `}else{` |
| 19 | `tPosterior=tPosterior*tLikelihood;` |
| 20 | `}` |
| 21 | `Log.d("pos", "likelihood:` |
| 22 | `"+entry.getKey()+", "+entryInput.getKey()+"="+tLikelihood+"` |
| 23 | `deg pem:"+getPembilang(entry.getKey(),entryInput.getKey())+"` |
| 24 | `penyb"+entry.getValue() +" dg posterior = "+tPosterior);` |
| 25 | `}` |
| 26 | `}` |

Penjelasan kode program proses perhitungan nilai *likelihood*:

Baris 2 – 20 : menghitung nilai *likelihood* masing-masing gejala masukan pada setiap penyakit.

Baris 21 – 26 : menghitung total nilai *likelihood* gejala masukan pada setiap penyakit.

### 5.2.5 Implementasi Perhitungan Nilai *Posterior*

Proses ini merupakan proses perhitungan nilai probabilitas *posterior*. Proses ini menghitung nilai probabilitas *posterior* masing-masing penyakit dengan mengalikan nilai probabilitas *prior* masing-masing penyakit dengan nilai total *likelihood* dari gejala masukan pada masing-masing penyakit. Proses perthitungan nilai probabilitas *posterior* dapat dilihat pada baris 3-11 di Tabel 5.7. Setelah nilai probabilitas *posterior* masing-masing opt selesai dihitung akan dipilih opt yang memiliki nilai probabilitas *posterior* tertinggi, dipilih sebagai hasil proses diagnosis. Kode program dari perhitungan nilai *posterior* dapat dilihat pada Tabel 5.7.

**Tabel 5.7 Kode Program Perhitungan Nilai *Posterior***

| No. | Kode Program |
|---|---|
| 1 | `//Hitung Nilai getMAx Posterior` |

```
2   void getMax(){
3           String key = null;
4           double val=0.0;
5           for       (Map.Entry<String,       Double>       entry:
6   this.posterior.entrySet()){
7               if (entry.getValue()>val){
8                   val=entry.getValue();
9                   key=entry.getKey();
10              }
11          }
12          //String                                              max=
13  Collections.max(this.posterior.keySet());
14          Log.d("jj", "getMax: "+key);
15          this.keyPenyakitPosteriorMax=key;
16          getCFPakar();
17      }
```

Penjelasan kode program proses perhitungan nilai *posterior*:
Baris 2 – 12 : menghitung nilai *posterior* masing-masing penyakit.
Baris 14 – 17 : memilih nilai maksimum pada nilai *posterior* masing-masing penyakit dan nilai *posterior* tertinggi akan dipilih sebagai hasil diagnosis.

### 5.2.6 Implementasi Perhitungan Nilai CF

Proses ini merupakan proses perhitungan nilai CF proses diagnosis metode *naïve bayes*. Setelah sistem menghasilkan sebuah diagnosis melalui proses perhitungan menggunakan metode *naïve bayes*, selanjutnya hasil diagnosis tersebut akan dihitung nilai keyakinan (CF). gejala akan memiliki nilai bobot CF pakar sesuai dengan nama penyakit hasil proses diagnosis metode *naïve bayes*. Untuk mendapatkan nilai CF maksimum (CF *combine*), dilakukan pengalian nilai bobot gejala CF pakar dengan nilai bobot CF pengguna, dimana nilai bobot CF pengguna bernilai 1 sesuai dengan gejala yang dimasukkan. Setelah mendapatkan hasil perhitungan CF *combine*, hasil tersebut merupakan nilai keyakinan dan persentase terhadap hasil diagnosis yang dilakukan menggunakan *naïve bayes*. Kode program dari perhitungan nilai *posterior* dapat dilihat pada Tabel 5.8.

**Tabel 5.8 Kode Program Perhitungan Nilai CF**

| No. | Kode Program |
|---|---|
| 1 | `//Cari Pembilang` |
| 2 | `private void ProsesCF() {` |
| 3 | `int x=1;` |
| 4 | `for (CF cf : this.listCFPakar){` |
| 5 | `for (Map.Entry<String, Double> entry:` |
| 6 | `cf.getPenyakit().entrySet()){` |
| 7 | `if` |
| 8 | `(entry.getKey().matches(this.keyPenyakitPosteriorMax)){` |
| 9 | `double` |
| 10 | `temp=cf.getPenyakit().get(this.keyPenyakitPosteriorMax)*this.` |
| 11 | `inputan.get("G"+x);` |
| 12 | `Log.d("Proses Cf dari pakar", "ProsesCF:` |
| 13 | `"+entry.getKey()+", "+cf.getGejala()+",` |
| 14 | `"+cf.getPenyakit().get(this.keyPenyakitPosteriorMax));` |

```
                    this.Cf_X_InputUser[x-1]=temp;
                    //Log.d("CF       usr",       "ProsesCF:
"+this.Cf_X_InputUser[x-1]);
                }
            }
            x++;
        }
        double combine[]= new double[64];
        for (int i=0; i<this.Cf_X_InputUser.length-1;i++){
            double temp;
            if (i==0){
               temp=
this.Cf_X_InputUser[0]+this.Cf_X_InputUser[1]*(1-
this.Cf_X_InputUser[0]);
            }else{
                temp=     this.Cf_X_InputUser[i+1]+combine[i-
1]*(1-this.Cf_X_InputUser[i+1]);
            }
            combine[i]=temp;
            Log.d("combine", "ProsesCF: "+combine[i]);
        }

        double presentase= combine[combine.length-1]*100;
        this.txHasil.setText("Kesimpulan:\n" +
                "Berdasarkan  perhitungan  menggunakan  naive
bayes,           sapi           mengalami           penyakit
"+this.keyPenyakitPosteriorMax+"   dengan   nilai   keyakinan
(certainty factor) sebesar "+combine[combine.length-1]+" atau
"+presentase+"%");
    }
    public void getDataTraining() {
        RequestQueue              requestQueue              =
Volley.newRequestQueue(this);
        String url = "http://wildanafif.id/demo/sispak/";
        final    ProgressDialog    progressdialog    =    new
ProgressDialog(this);
        progressdialog.setMessage("Please Wait....");
        progressdialog.setTitle("Loading");
        progressdialog.show();
        JsonArrayRequest     jsonObjectRequest     =     new
JsonArrayRequest(Request.Method.GET,url,                    new
Response.Listener<JSONArray>() {
            @Override
            public void onResponse(JSONArray response) {

                //Toast.makeText(MapsActivity.this,
""+response, Toast.LENGTH_SHORT).show();
                for(int i=0;i<response.length();i++){

                    JSONObject json_data = null;
                    try {
                        json_data                                =
response.getJSONObject(i);
                        Map<String,   Integer>   temp   =   new
HashMap<String,Integer>();

                        for (int x=1;x<66 ; x++){
                            temp.put("G"+x,
json_data.getInt("G"+x));
```

```
74
75                          }
76                          Penyakit        penyakit=         new
77  Penyakit(json_data.getString("penyakit"),temp);
78                          //Log.d("",              "onResponse:
79  "+penyakit.getGejala().toString());
80                          listPenyakit.add(penyakit);
81                      } catch (JSONException e) {
82                          e.printStackTrace();
83                      }
84                  }
85                  likelihood();
86                  progressdialog.dismiss();
87              }
89          }, new Response.ErrorListener() {
90              @Override
91              public void onErrorResponse(VolleyError error) {
92                  Log.e("LOG", error.toString());
93              }
94          });
95          requestQueue.add(jsonObjectRequest);
96      }
97      public void getCFPakar() {
98          RequestQueue                requestQueue                =
99  Volley.newRequestQueue(this);
101         String                       url                         =
102 "http://wildanafif.id/demo/sispak/index.php/welcome/cfpakar";
103         final    ProgressDialog    progressdialog    =    new
104 ProgressDialog(this);
105         progressdialog.setMessage("Please Wait....");
106         progressdialog.setTitle("Loading");
107         progressdialog.show();
108         JsonArrayRequest     jsonObjectRequest      =      new
109 JsonArrayRequest(Request.Method.GET,url,                    new
110 Response.Listener<JSONArray>() {
111             @Override
112             public void onResponse(JSONArray response) {
113
114                 //Toast.makeText(MapsActivity.this,
115 ""+response, Toast.LENGTH_SHORT).show();
116                 for(int i=0;i<response.length();i++){
117
118                     JSONObject json_data = null;
119                     try {
120                         json_data                           =
121 response.getJSONObject(i);
122                         Map<String,   Double>   temp   =   new
123 HashMap<String,Double>();
124                         for (int x=1;x<17 ; x++){
125                             temp.put("P"+x,
126 json_data.getDouble("P"+x));
127                         }
128                         CF           penyakit=            new
129 CF(json_data.getString("gejala"),temp);
130                         //Log.d("CF    PAKAR",    "onResponse:
131 "+penyakit.getPenyakit().toString());
132                         listCFPakar.add(penyakit);
133                     } catch (JSONException e) {
134                         e.printStackTrace();
```

```
135                         }
136                     }
137                     ProsesCF();
138                     progressdialog.dismiss();
139                 }
140             }, new Response.ErrorListener() {
141                 @Override
142                 public void onErrorResponse(VolleyError error) {
143                     Log.e("LOG CF", error.toString());
144                 }
145             });
146             requestQueue.add(jsonObjectRequest);
147         }
148     }
149
```

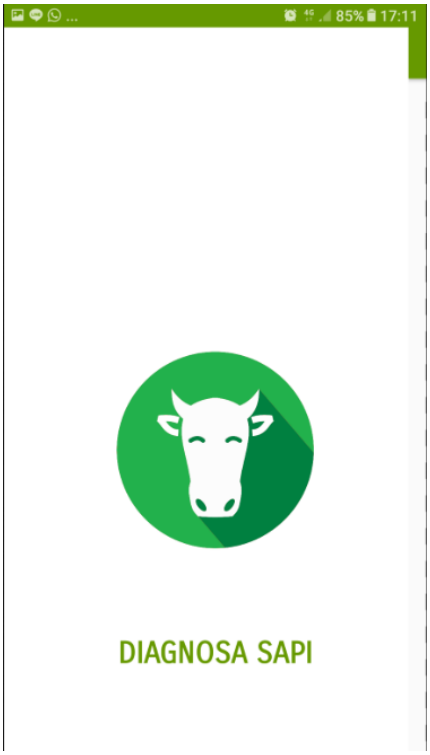
Penjelasan kode program proses perhitungan nilai CF:
Baris 2 – 19 : menghitung nilai CF hasil diagnosis
Baris 22 – 148 : menghitung nilai CF *combine* hasil diagnosis

## 5.3 Implementasi Antarmuka Pengguna

Pada sub-bab ini akan dijelaskan implementasi antarmuka pengguna sistem pakar diagnosis penyakit sapi.

### 5.3.1 Implementasi Halaman Awal

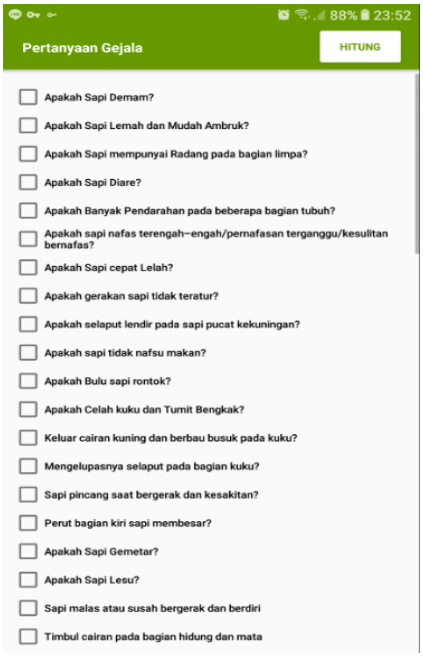
Gambar 5.1 menunjukkan hasil implementasi dari perancangan halaman awal pada Gambar 4.7.




**Gambar 5.1 Implementasi Antarmuka Pengguna Halaman Awal**

### 5.3.2 Implementasi Antarmuka Pengguna Halaman Diagnosis

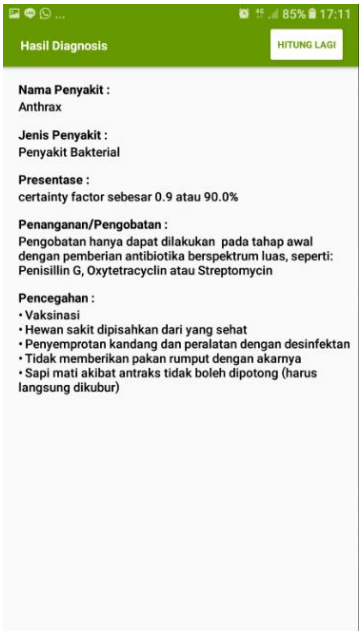
Gambar 5.2 menunjukkan hasil implementasi dari perancangan antarmuka pengguna halaman diagnosis pada Gambar 4.8.




**Gambar 5.2 Implementasi Antarmuka Pengguna Halaman Diagnosis**

### 5.3.3 Implementasi Antarmuka Pengguna Halaman Hasil Diagnosis

Gambar 5.3 menunjukkan hasil implementasi dari perancangan antarmuka pengguna halaman diagnosis pada Gambar 4.9.




**Gambar 5.3 Implementasi Antarmuka Pengguna Halaman Hasil Diagnosis**