

BAB 4
ANALISIS DATA DAN PEMBAHASAN

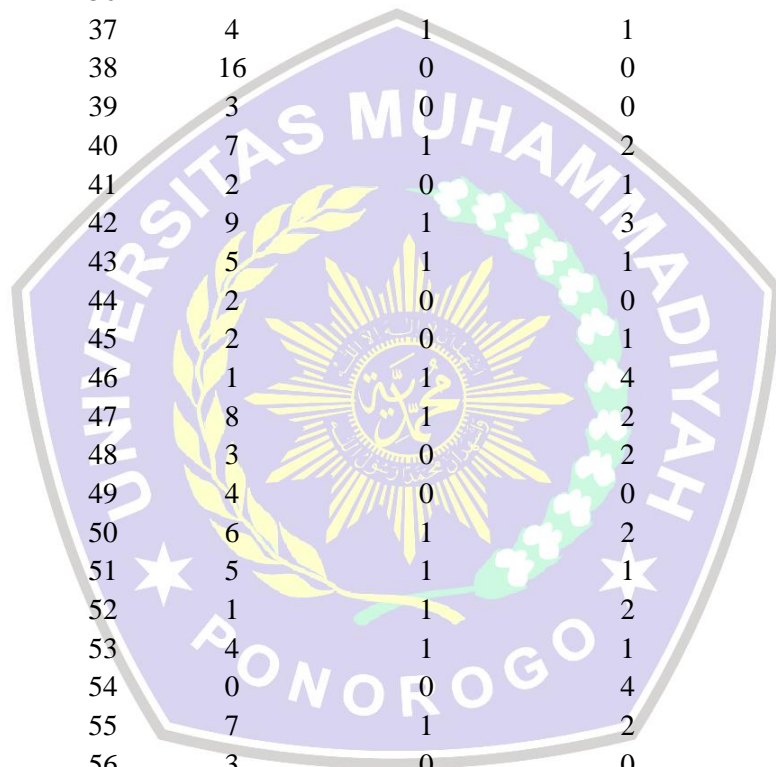
4.1. Analisis Data

Dataset yang akan di gunakan pada penelitian ini merupakan data kasus *covid-19* per kelurahan Kabupaten Ponorogo. Data tersebut di peroleh dari tiap UPT Puskesmas dari 14 kecamatan yang ada di ponorogo dalam bentuk peta sebaran *covid-19* di setiap kecamatan. Sumber data di peroleh dari petugas Surveilans yang di update pada halaman media sosial Instagram UPT Puskesmas sekabupaten Ponorogo.

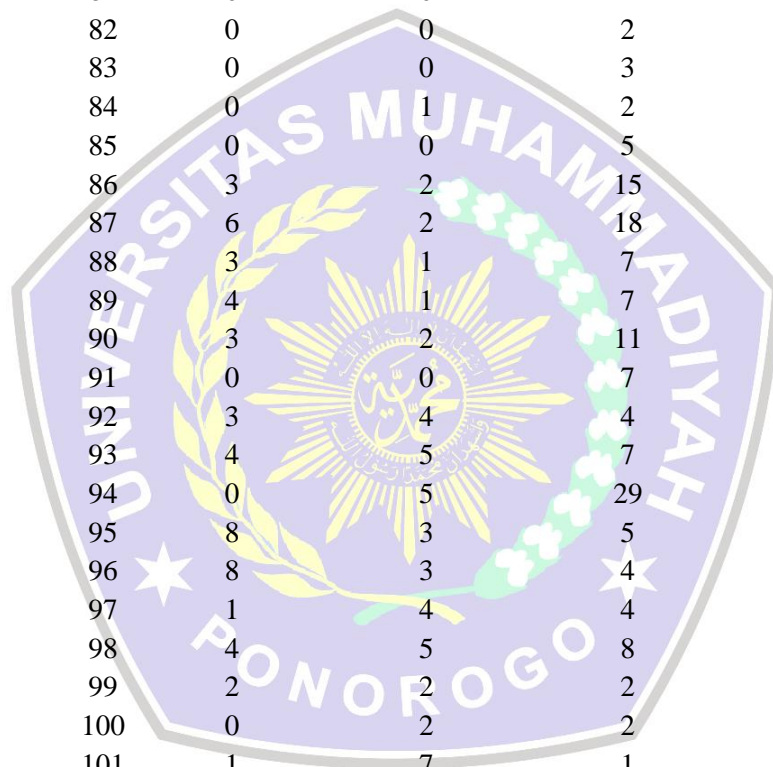
Tabel 4.1 Data persebaran *covid-19* Kabupaten ponorogo per kelurahan

No	Positif	Pasien Dalam Pengawasan (PDP)	Orang Dalam Pengawasan (ODP)
1	2	0	0
2	4	8	0
3	2	6	12
4	3	4	0
5	5	0	1
6	3	0	8
7	4	4	1
8	2	3	2
9	1	9	1
10	1	4	12
11	6	0	0
12	6	0	0
13	0	12	0
14	0	0	24
15	5	0	8
16	5	0	2
17	5	0	3
18	4	2	2
19	5	1	1
20	5	1	1
21	5	1	1
22	4	2	8
23	5	0	10

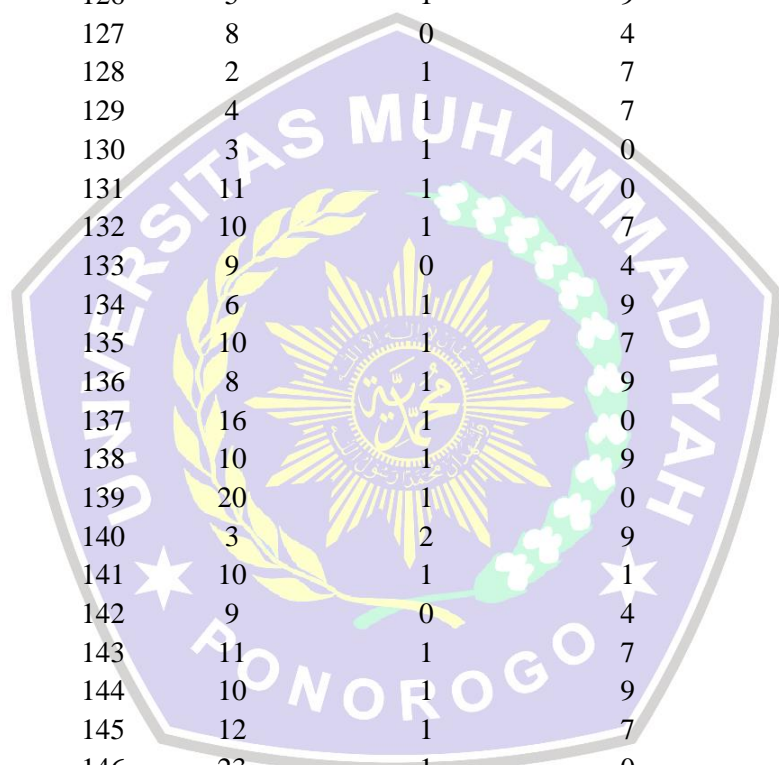
24	4	4	1
25	4	0	4
26	3	4	4
27	2	6	6
28	2	6	2
29	2	4	8
30	2	4	6
31	0	1	1
32	0	0	1
33	0	0	2
34	0	1	0
35	0	0	4
36	1	1	2
37	4	1	1
38	16	0	0
39	3	0	0
40	7	1	2
41	2	0	1
42	9	1	3
43	5	1	1
44	2	0	0
45	2	0	1
46	1	1	4
47	8	1	2
48	3	0	2
49	4	0	0
50	6	1	2
51	5	1	1
52	1	1	2
53	4	1	1
54	0	0	4
55	7	1	2
56	3	0	0
57	9	1	3
58	5	1	1
59	1	1	4
60	8	1	2
61	3	0	2
62	6	1	2
63	5	1	1
64	1	1	4
65	2	0	1
66	8	1	1
67	3	0	1
68	4	0	0



69	0	0	7
70	4	0	2
71	4	0	1
72	2	0	5
73	2	1	3
74	1	0	3
75	2	0	1
76	5	0	2
77	1	0	7
78	4	0	3
79	2	2	3
80	3	2	3
81	0	0	1
82	0	0	2
83	0	0	3
84	0	1	2
85	0	0	5
86	3	2	15
87	6	2	18
88	3	1	7
89	4	1	7
90	3	2	11
91	0	0	7
92	3	4	4
93	4	5	7
94	0	5	29
95	8	3	5
96	8	3	4
97	1	4	4
98	4	5	8
99	2	2	2
100	0	2	2
101	1	7	1
102	0	2	1
103	0	0	2
104	0	0	10
105	0	0	6
106	0	0	5
107	1	0	0
108	2	0	11
109	0	0	36
110	0	1	10
111	2	1	10
112	3	1	10
113	0	1	5



114	3	1	11
115	4	1	11
116	4	0	2
117	4	0	3
118	0	1	9
119	4	0	4
120	3	0	5
121	0	1	7
122	3	0	4
123	3	1	9
124	5	0	5
125	6	0	4
126	5	1	9
127	8	0	4
128	2	1	7
129	4	1	7
130	3	1	0
131	11	1	0
132	10	1	7
133	9	0	4
134	6	1	9
135	10	1	7
136	8	1	9
137	16	1	0
138	10	1	9
139	20	1	0
140	3	2	9
141	10	1	1
142	9	0	4
143	11	1	7
144	10	1	9
145	12	1	7
146	23	1	0
147	1	1	0
148	2	2	0
149	2	1	0
150	1	1	2
151	7	0	2
152	1	0	7
153	1	1	5
154	1	2	6
155	2	6	2
156	1	7	4
157	1	0	2
158	5	0	0



159	3	0	1
160	9	0	2
161	7	0	2
162	6	0	3
163	0	1	2
164	0	2	3
165	1	2	3
166	1	0	5
167	1	3	5
168	1	5	6
169	0	0	4
170	2	6	12

Sumber Data:

1. UPT Puskesmas Badegan Ponorogo
<https://www.instagram.com/puskesmasbadegan/?hl=en>
2. UPT Puskesmas Jambon
<https://www.instagram.com/pkmjambon.po/?hl=en>
3. UPT Puskesmas Jenangan
<https://www.instagram.com/puskesmasjenanganofficial/?hl=en>
4. UPT Puskesmas Jetis
https://www.instagram.com/puskesmas_jetis_ponorogo/?hl=en
5. UPT Puskesmas Mlarak
https://www.instagram.com/puskesmasmlarak_ponorogo/?hl=en
6. UPT Puskesmas Ngrayun
<https://www.instagram.com/puskesmasngrayun/?hl=en>
7. UPT Puskesmas Ponorogo
https://www.instagram.com/puskesmas_ponorogo_selatan/?hl=en
https://www.instagram.com/puskesmas_ponorogoutara/?hl=en
8. UPT Puskesmas Pudak
<https://www.instagram.com/puskesmaspudak/?hl=en>

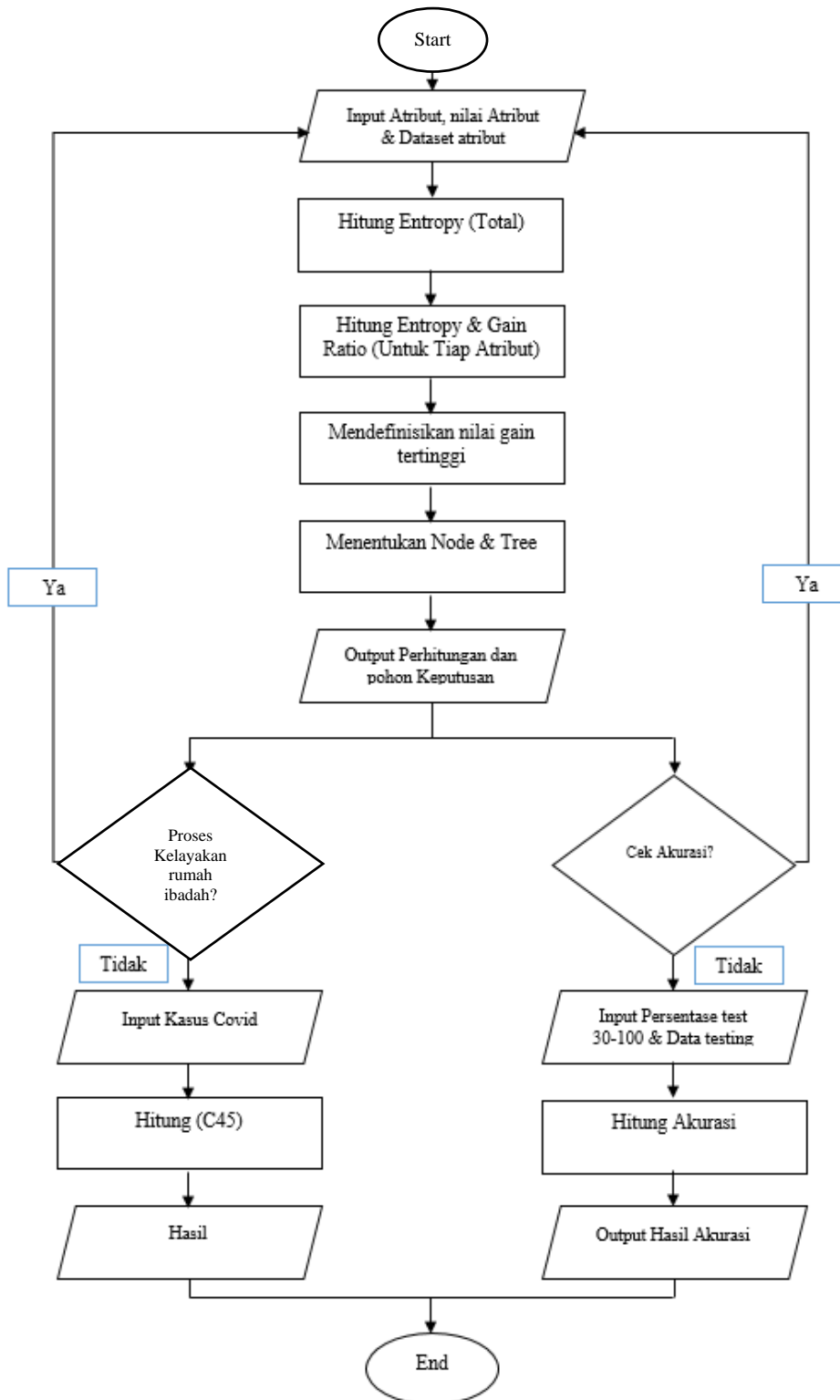
9. UPT Puskesmas Pulung
<https://www.instagram.com/puskesmaspulung/?hl=en>
10. UPT Puskesmas Sambit
<https://www.instagram.com/puskesmassambit.png/?hl=en>
11. UPT Puskesmas Siman
https://www.instagram.com/puskesmas_siman/?hl=en
12. UPT Puskesmas Slahung
<https://www.instagram.com/pkmslahung/?hl=en>
13. UPT Puskesmas Sooko
<https://www.instagram.com/puskesmas.sooko/?hl=en>
14. UPT Puskesmas Sukorejo
<https://www.instagram.com/puskesmassukorejo.ponorogo/?hl=en>

Dataset tersebut yang nantinya akan digunakan sebagai data training dalam melakukan pemrosesan algoritma C4.5 pada sistem.

4.2. Implementasi Algoritma C4.5

Implementasi algoritma C4.5 dilakukan dengan perhitungan secara manual terhadap dataset. Dalam melakukan pengujian, penulis melakukan proses perhitungan manual terhadap 20% dari dataset dengan menggunakan algoritma C4.5. Tujuan dari perhitungan secara manual tersebut ialah untuk menyesuaikan apakah proses perhitungan yang dilakukan oleh sistem telah sesuai dan memiliki hasil yang sama dengan perhitungan manual. Jika perhitungan dari sistem telah sesuai dengan perhitungan manual maka dapat disimpulkan bahwa program dan algoritma C4.5 yang telah di implementasikan kedalam source code sistem telah berjalan dengan baik.

A. Flowcart diagram proses Algoritma C4.5



Gambar : 4.1 Flowcart diagram algoritma C4.5

1. Input Dataset

Tabel 4.2 Data training

No	Positif	PDP	ODP	Kelayakan rumah ibadah
1	2	0	0	Ya (Maksimal 50% Kapasitas)
2	4	8	0	Tidak
3	2	6	12	Tidak
4	3	4	0	Ya (Maksimal 50% Kapasitas)
5	5	0	1	Ya (Maksimal 50% Kapasitas)
6	3	0	8	Ya (Maksimal 50% Kapasitas)
7	4	7	1	Tidak
8	2	3	2	Ya (Maksimal 50% Kapasitas)
9	1	9	1	Tidak
10	2	24	12	Tidak
11	6	0	0	Tidak
12	6	0	0	Tidak
13	0	12	0	Tidak
14	0	0	36	Tidak
15	5	0	8	Ya (Maksimal 50% Kapasitas)
16	5	0	2	Ya (Maksimal 50% Kapasitas)
17	5	0	3	Ya (Maksimal 50% Kapasitas)
18	4	2	2	Ya (Maksimal 50% Kapasitas)
19	5	1	1	Ya (Maksimal 50% Kapasitas)
20	5	1	1	Ya (Maksimal 50% Kapasitas)

2. Hitung entropi total

Dataset tersebut memiliki 20 kasus yang terdiri dari 11 “Ya (Maksimal 50% Kapasitas)” dan 9 “Tidak”

pada kolom Kelayakan rumah ibadah(lihat table 4.2). Untuk menghitung entropy total dengan rumus:

$$Entropy(S) = \sum_{i=1}^n (-p_i) * \log_2(p_i)$$

Keterangan

S : Himpunan Kasus

n : Jumlah partisi atribut S

pi : Proporsi dari Si terhadap S

Jadi Entropi total:

$$(S) = \left(-\left(\frac{11}{20}\right) \times \text{Log}_2\left(\frac{11}{20}\right)\right) + \left(-\left(\frac{9}{20}\right) \times \text{Log}_2\left(\frac{9}{20}\right)\right) = 0.9927745$$

Tabel 4.3 Hasil perhitungan pada dataset

Atribut	Nilai	Jumlah Kasus	Yes	No	Entropy	Gain
Total		20	11	9	0.9927745	

3. Hitung entropi dan gain ratio (untuk tiap atribut)

Setelah mendapatkan entropy total dari keseluruhan kasus, Lakukan analisis pada tiap atribut dan nilai nilainya kemudian hitung entropinya seperti yang di tampilkan pada table berikut:

Tabel 4.4 Analisis atribut, nilai, entropi dan gain

Atribut	Nilai	Jumlah		Entropy	gain
		Kasus	Yes		
Total		20	11	9	0.9927745
Positif	0	2	0	2	0
	1	1	0	1	0
	2	4	2	2	1
	3	2	2	0	0
	4	3	1	2	0.9182958
	5	6	6	0	0
	6	2	0	2	0
PDP	0	9	6	3	0.9182958
	1	2	2	0	0
	2	1	1	0	0
	3	1	1	0	0
	4	1	1	1	0
	6	1	0	1	0
	7	1	0	1	0
	8	1	0	1	0
	9	1	0	1	0
	12	1	0	1	0
	24	1	0	1	0
	ODP	0	6	2	4
1		5	3	2	0.9709505
2		3	3	0	0

3	1	1	0	0
8	2	2	0	0
12	2	0	2	0
36	1	0	1	0

Untuk menghitung gain setiap atribut menggunakan rumus:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i)$$

Keterangan

S : Himpunan Kasus

A : Atribut

n : Jumlah partisi atribut A

|S_i| : Jumlah Kasus pada partisi ke-i

|S| : Jumlah Kasus dalam S

Jadi :

$$\begin{aligned} \text{Gain (Positif)} &= 0.9927745 - \left(\left(\frac{2}{20} \right) \times 0 \right) - \left(\left(\frac{1}{20} \right) \times 0 \right) - \left(\left(\frac{4}{20} \right) \times 1 \right) \\ &- \left(\left(\frac{2}{20} \right) \times 0 \right) - \left(\left(\frac{3}{20} \right) \times 0.91829583 \right) - \left(\left(\frac{6}{20} \right) \times 0 \right) - \left(\left(\frac{2}{20} \right) \times 0 \right) = 0.65503 \end{aligned}$$

Hitung gain (Positif), gain (PDP) dan gain (ODP). Hasilnya dapat dilihat pada table 4.4.

4. Mendefinisikan nilai gain tertinggi

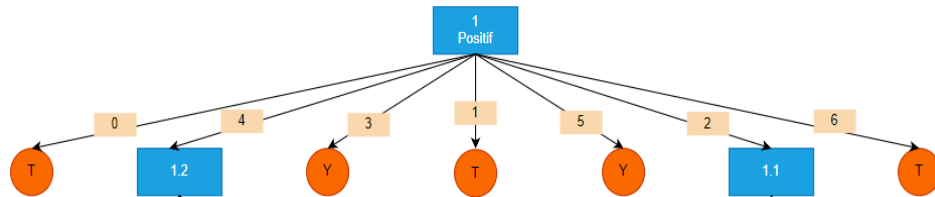
Dari table 4.4 nilai gain terbesar yaitu gain (Positif). Maka atribut Positif merupakan atribut terbaik untuk di jadikan node akar (root node).

5. Menentukan Node dan tree

Pada nilai Positif 0 memiliki 2 kasus dan semuanya memiliki jawaban Tidak ($\text{Sum}(\text{total}) / \text{Sum}(\text{Tidak}) = 2/2=1$). Pada nilai Positif 1 memiliki 1 kasus dan semuanya memiliki jawaban Tidak ($\text{Sum}(\text{total}) / \text{Sum}(\text{Tidak}) = 1/1=1$). Pada nilai Positif 3 memiliki 2 kasus dan semuanya

memiliki jawaban Ya ($\text{Sum}(\text{total}) / \text{Sum}(\text{Ya}) = 2/2=1$). Pada nilai Positif 5 memiliki 6 kasus dan semuanya memiliki jawaban Ya ($\text{Sum}(\text{total}) / \text{Sum}(\text{Ya}) = 6/6=1$). Pada nilai Positif 6 memiliki 2 kasus dan semuanya memiliki jawaban Tidak ($\text{Sum}(\text{total}) / \text{Sum}(\text{Tidak}) = 2/2=1$).

Dengan demikian nilai Positif 0, 1, 3, 5 dan 6 menjadi daun atau *leaf*.



Gambar : 4.2 Pohon keputusan node 1 (*root node*)

Berdasarkan pembentukan pohon keputusan node 1 (*root node*) akan di analisis lebih lanjut. Untuk mempermudah dapat mengambil data yang memiliki nilai Positif = 2 sehingga terbentuk seperti table berikut:

Tabel 4.5 Data yang memiliki nilai Positif = 2

Positif	PDP	ODP	Kelayakan rumah ibadah
2	0	0	Ya (Maksimal 50% Kapasitas)
2	6	12	Tidak
2	3	2	Ya (Maksimal 50% Kapasitas)
2	24	12	Tidak

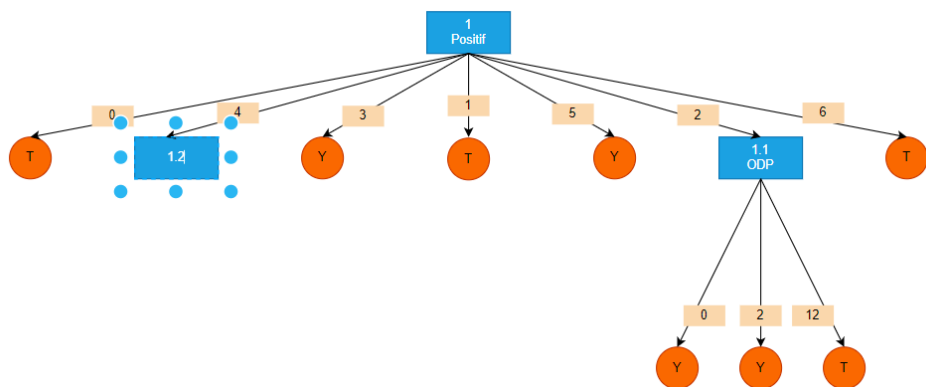
Selanjutnya hitung lagi nilai entropi atribut Positif = 2 dan entropi setiap atribut dan gainnya. Sehingga hasilnya seperti pada table. Setelah pilih atribut yang memiliki gain tertinggi untuk di buatkan node berikutnya.

Tabel 4.6 Tabel hasil analisis node 1.2

Atribut	Nilai	Jumlah		Entropy	gain
		Kasus	Yes		
Total		4	2	2	
PDP					1
	0	1	1	0	
	3	1	1	0	
	6	1	0	1	
	24	1	0	1	
ODP					1
	0	1	1	0	

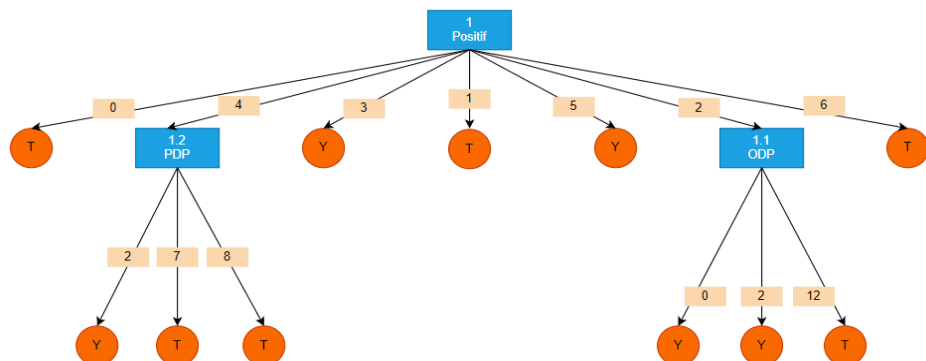
2	1	1	0	0
12	2	0	2	0

Dari table yang terbentuk di atas, Terdapat nilai gain yang sama antara atribut PDP dan ODP. Sehingga perlu dicari gain ratio tertingginya, dari perhitungan di peroleh gain ratio tertinggi pada atribut ODP, dan nilai yang di jadikan daun atau *leaf* adalah nilai ODP 0, 2, 12. Sehingga dapat di bentuk pohon keputusan:



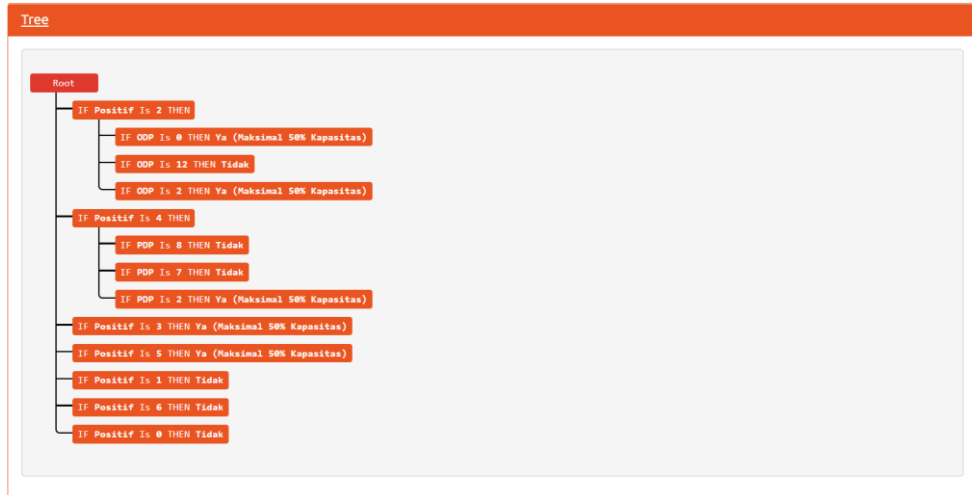
Gambar : 4.3 Pohon keputusan analisis node 1.2

Selanjutnya Dilakukan analisis node 1.3 dan 1.1 seperti menentukan daun atau *leaf* pada cabang ODP = 1 dan ODP = 12. Sehingga pohon keputusan yang dapat terbentuk akan seperti pada gambar berikut ini.



Gambar : 4.4 Pohon keputusan akhir

Dari pohon keputusan akhir yang di peroleh dari perhitungan manual dilakukan pencocokan dengan pohon keputusan yang di proses oleh sistem.



Gambar : 4.5 Pohon keputusan yang terbentuk dari proses perhitungan C4.5 pada sistem

Dari pohon keputusan akhir yang di dihasilkan oleh aplikasi dengan hasil perhitungan manual Membentuk aturan dan tree yang sama, sehingga dapat di Tarik kesimpulan bahwa algoritma yang di implementasikan pada source code program telah berjalan dengan baik.

6. Cek Akurasi

Setelah rule dan treenya terbentuk maka dapat di cek tingkat akurasi dari dataset yang digunakan kemudian dapat dilakukan perhitungan untuk data uji yang baru. Tujuan dilakukan pengecekan akurasi yaitu untuk menyimpulkan apakah dataset yang di gunakan Memperoleh hasil keakuratan yang tinggi dengan algoritma yang digunakan. Jika angka presentasi tinggi, menandakan bahwa algoritma tersebut cocok di gunakan dalam melakukan perhitungan terhadap dataset yang ada.

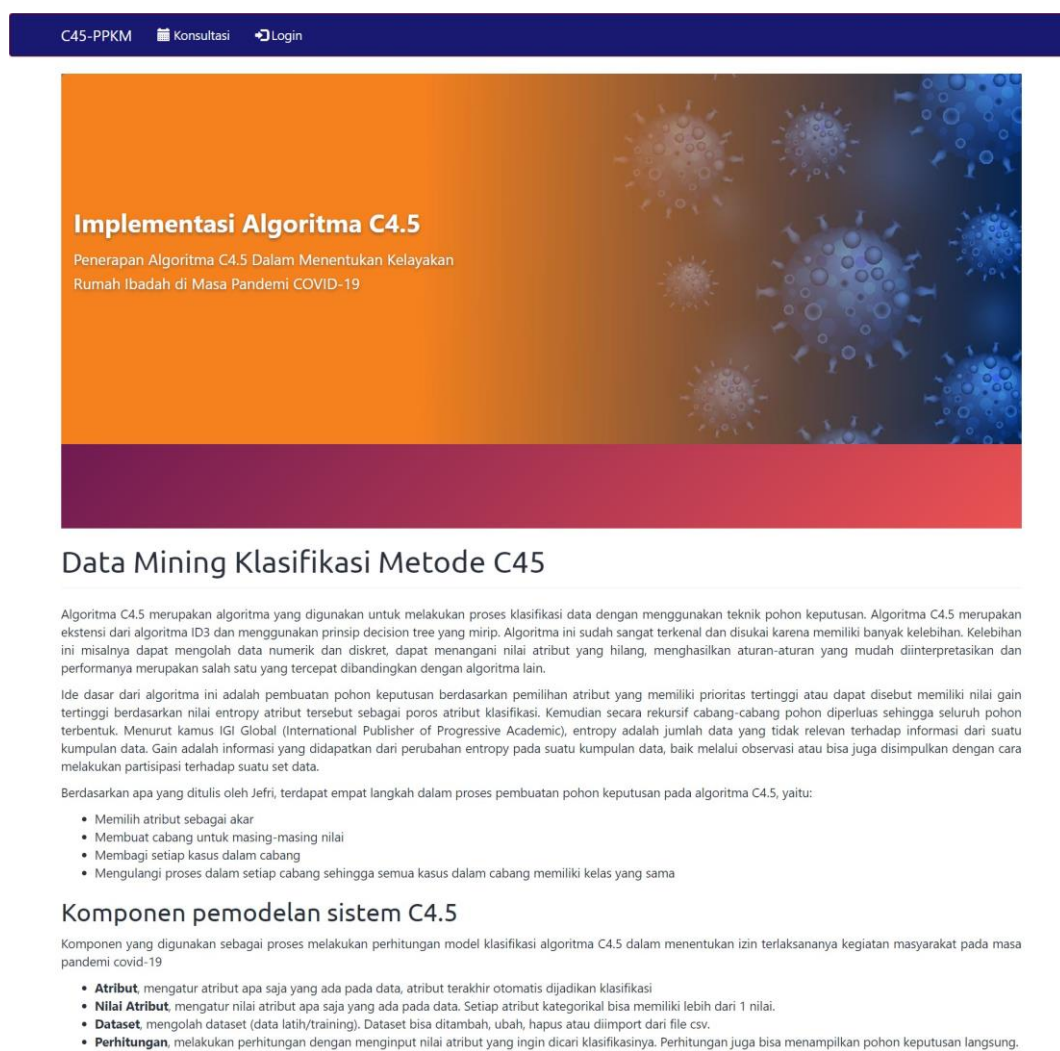
7. Proses Kelayakan rumah ibadah

Berdasarkan aturan-aturan dan pohon keputusan yang terbentuk inilah yang nantinya akan di jadikan sebagai acuan dalam menentukan Kelayakan rumah ibadah.

4.3. Pembahasan Interface

A. Halaman utama

Pada halaman utama hak akses di berikan kepada seluruh pengguna. Terdapat menu konsultasi dan login pada halaman ini. Sebagai dokumentasi dari sistem dijelaskan pengertian algoritma C45, proses perhitungan, implementasi serta aturan PPKM yang sedang di berlakukan pemerintah di masa pandemi *covid-19*. Merupakan sedikit gambaran terkait aplikasi guna memberi pemahaman kepada user akan fungsi dan kinerja dari sistem yang di gunakan.



Gambar : 4.6 Halaman utama user

B. Menu Konsultasi

Pada menu ini dapat dimanfaatkan oleh pengguna ketika ingin melakukan perhitungan atau pengecekan terkait pemberlakuan Kelayakan rumah ibadah di lingkungannya berdasarkan kasus covid yang terjadi dalam kurun waktu 7 hari terakhir.

C45-PPKM Konsultasi Login

Perhitungan

Data yang diketahui

Positif *

PDP *

ODP *

Hitung

Copyright © 2021 Universitas Muhammadiyah Ponorogo Tugas Akhir Kuliah

Gambar : 4.7 Halaman konsultasi

C. Menu login

Menu login di peruntukkan kepada admin yang memiliki hak akses untuk melakukan perubahan data di dalam sistem dengan memasukkan username dan password.

C45-PPKM Konsultasi Login

Login

Username

Password

Masuk

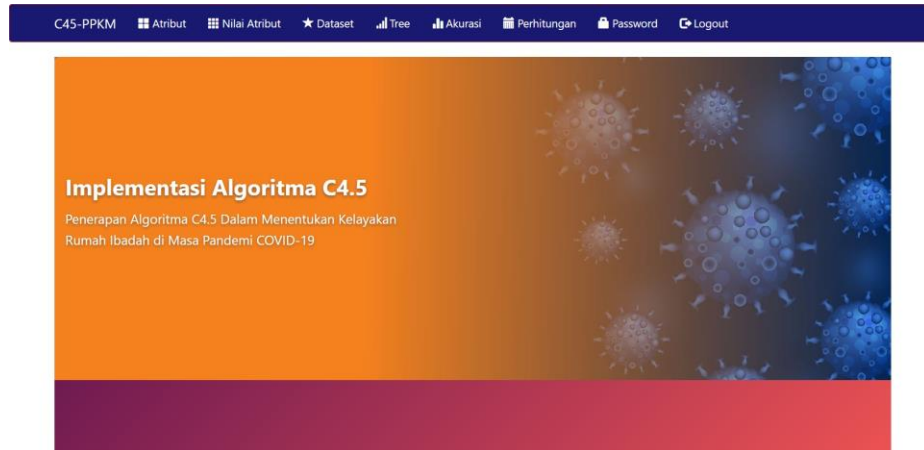
User: admin, password: admin

Copyright © 2021 Universitas Muhammadiyah Ponorogo Tugas Akhir Kuliah

Gambar : 4.8 Halaman login

D. Halaman utama admin

Setelah login maka akan di arahkan pada halaman utama admin. Pada halaman utama memiliki tampilan yang sama dengan halaman utama dari user. Pada halaman admin terdapat menu atribut, nilai atribut, dataset, tree, akurasi, perhitungan, password dan menu logout.



Data Mining Klasifikasi Metode C45

Algoritma C4.5 merupakan algoritma yang digunakan untuk melakukan proses klasifikasi data dengan menggunakan teknik pohon keputusan. Algoritma C4.5 merupakan ekstensi dari algoritma ID3 dan menggunakan prinsip decision tree yang mirip. Algoritma ini sudah sangat terkenal dan disukai karena memiliki banyak kelebihan. Kelebihan ini misalnya dapat mengolah data numerik dan diskret, dapat menangani nilai atribut yang hilang, menghasilkan aturan-aturan yang mudah diinterpretasikan dan performanya merupakan salah satu yang tercepat dibandingkan dengan algoritma lain.

Ide dasar dari algoritma ini adalah pembuatan pohon keputusan berdasarkan pemilihan atribut yang memiliki prioritas tertinggi atau dapat disebut memiliki nilai gain tertinggi berdasarkan nilai entropy atribut tersebut sebagai poros atribut klasifikasi. Kemudian secara rekursif cabang-cabang pohon diperluas sehingga seluruh pohon terbentuk. Menurut kamus IGI Global (International Publisher of Progressive Academic), entropy adalah jumlah data yang tidak relevan terhadap informasi dari suatu kumpulan data. Gain adalah informasi yang didapatkan dari perubahan entropy pada suatu kumpulan data, baik melalui observasi atau bisa juga disimpulkan dengan cara melakukan partisipasi terhadap suatu set data.

Berdasarkan apa yang ditulis oleh Jefri, terdapat empat langkah dalam proses pembuatan pohon keputusan pada algoritma C4.5, yaitu:

- Memilih atribut sebagai akar
- Membuat cabang untuk masing-masing nilai
- Membagi setiap kasus dalam cabang
- Mengulangi proses dalam setiap cabang sehingga semua kasus dalam cabang memiliki kelas yang sama

Komponen pemodelan sistem C4.5

Komponen yang digunakan sebagai proses melakukan perhitungan model klasifikasi algoritma C4.5 dalam menentukan izin terlaksananya kegiatan masyarakat pada masa pandemi covid-19

- **Atribut**, mengatur atribut apa saja yang ada pada data, atribut terakhir otomatis dijadikan klasifikasi
- **Nilai Atribut**, mengatur nilai atribut apa saja yang ada pada data. Setiap atribut kategorikal bisa memiliki lebih dari 1 nilai.
- **Dataset**, mengolah dataset (data latih/training). Dataset bisa ditambah, ubah, hapus atau diimport dari file csv.
- **Perhitungan**, melakukan perhitungan dengan menginput nilai atribut yang ingin dicari klasifikasinya. Perhitungan juga bisa menampilkan pohon keputusan langsung.

Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM) mikro

Mulai tanggal 3/7/2021, pemerintah kembali menerapkan Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM) Darurat di sejumlah daerah. Kebijakan ini diberlakukan sebagai salah satu upaya memutus rantai penyebaran Covi-19 yang terus meningkat.

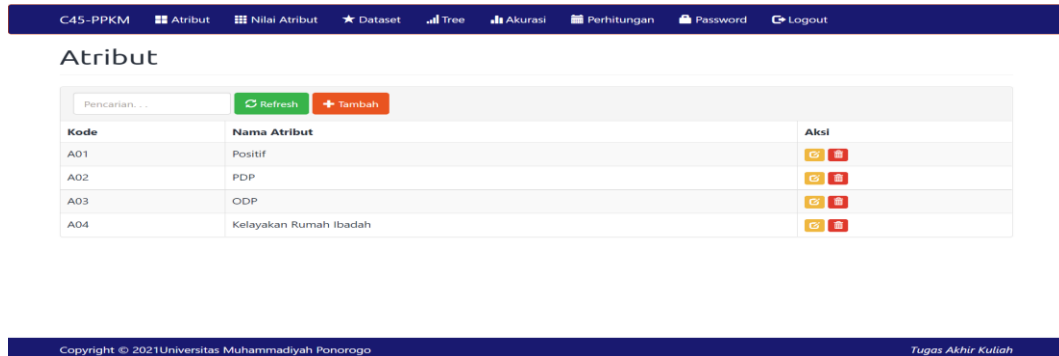
Aturan PPKM mikro:



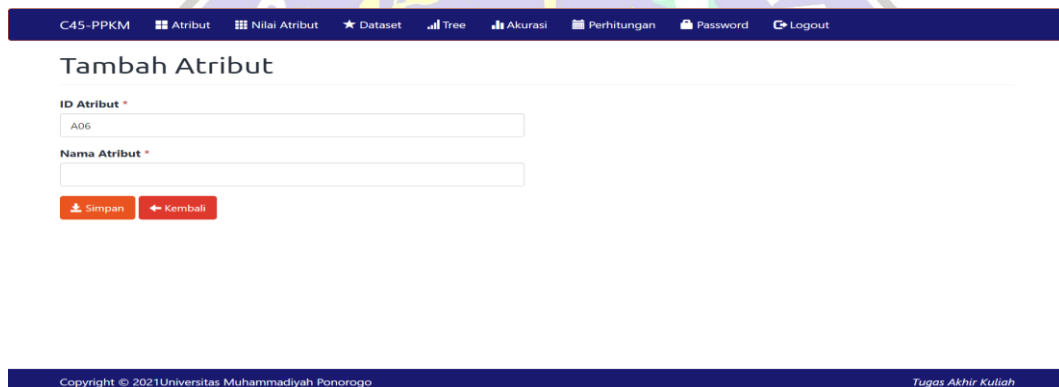
Gambar : 4.9 Halaman utama admin

E. Menu Atribut

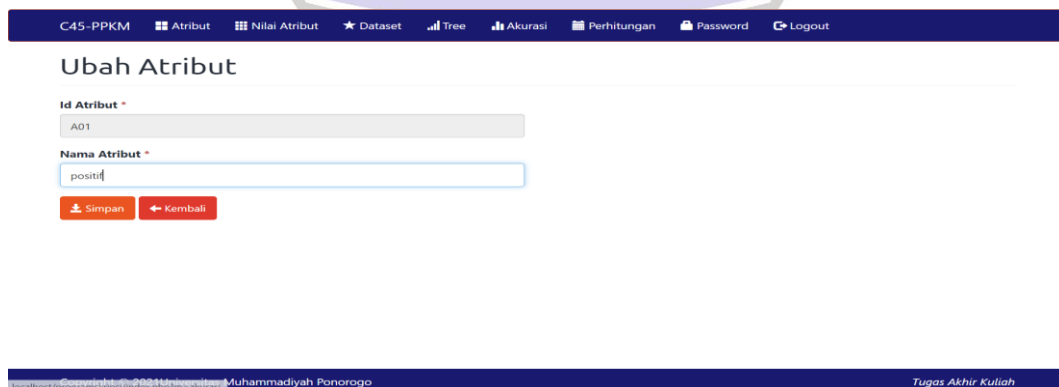
Pada menu ini admin dapat melakukan update, edit dan hapus atribut yang digunakan sebagai indikator dalam proses perhitungan menentukan Kelayakan rumah ibadah. Untuk atribut yang digunakan yaitu kasus Positif, OTG, PDP dan ODP.



Gambar : 4.10 Halaman Atribut



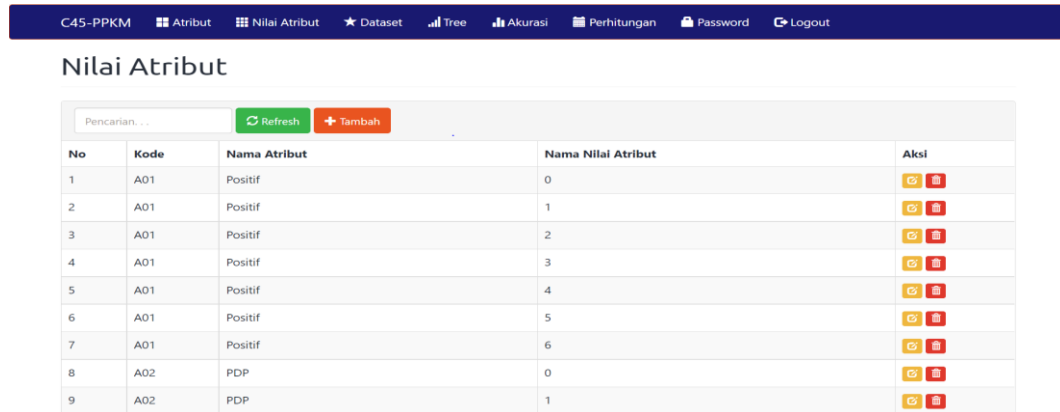
Gambar : 4.11 Halaman Tambah Atribut





















Gambar : 4.12 Halaman Ubah Atribut

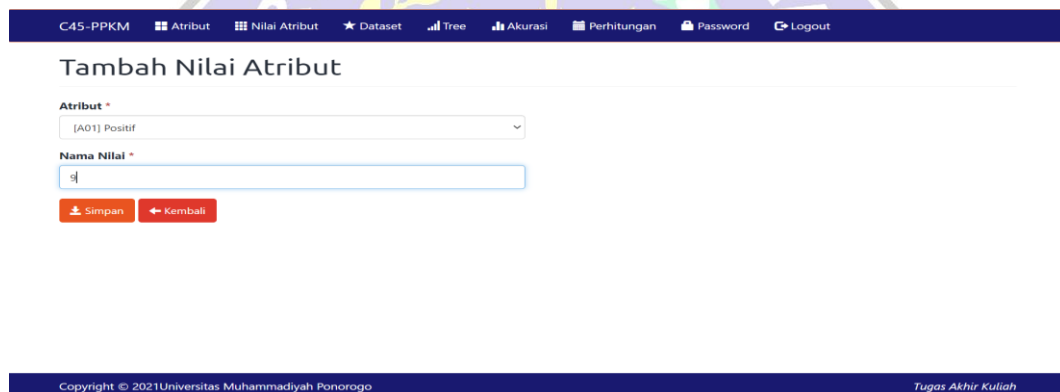
F. Menu nilai atribut

Input nilai untuk setiap atribut di gunakan untuk mempermudah proses saat melakukan input maupun perubahan data pada dataset. Nilai atribut yang digunakan merupakan nilai dari kasus terendah hingga kasus tertinggi untuk tiap atribut.



No	Kode	Nama Atribut	Nama Nilai Atribut	Aksi
1	A01	Positif	0	 
2	A01	Positif	1	 
3	A01	Positif	2	 
4	A01	Positif	3	 
5	A01	Positif	4	 
6	A01	Positif	5	 
7	A01	Positif	6	 
8	A02	PDP	0	 
9	A02	PDP	1	 



Gambar : 4.13 Halaman Nilai Atribut



Tambah Nilai Atribut

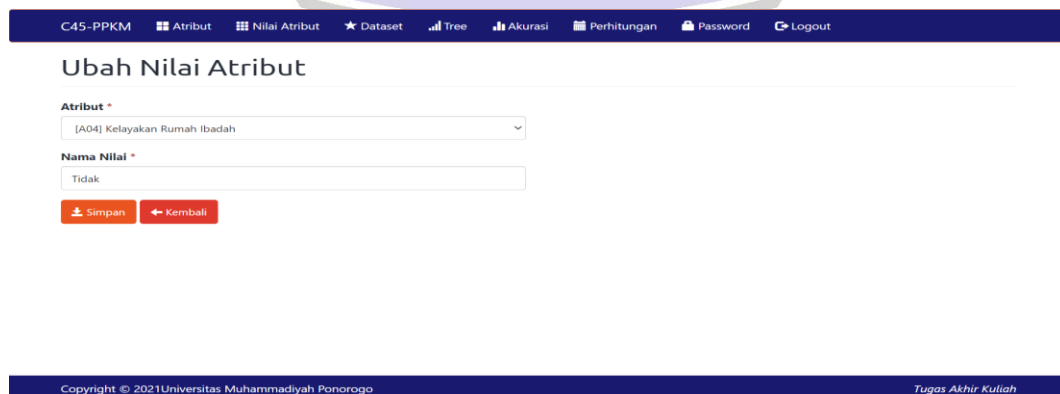
Atribut *
[A01] Positif

Nama Nilai *
0

Copyright © 2021 Universitas Muhammadiyah Ponorogo Tugas Akhir Kuliah



Gambar : 4.14 Halaman Tambah Nilai Atribut



Ubah Nilai Atribut

Atribut *
[A04] Kelayakan Rumah Ibadah

Nama Nilai *
Tidak

Copyright © 2021 Universitas Muhammadiyah Ponorogo Tugas Akhir Kuliah

Gambar : 4.15 Halaman Ubah Nilai Atribut

G. Menu Dataset

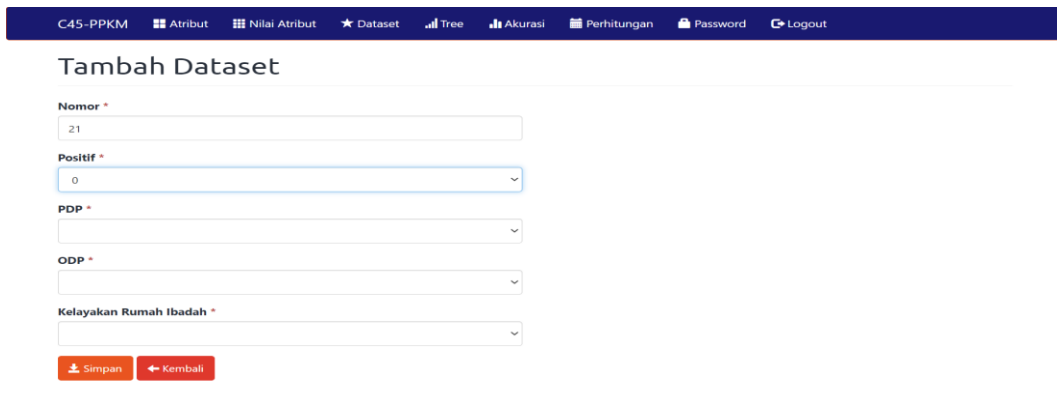
Menu ini memberikan akses untuk melakukan tambah dataset, import dataset dan melakukan perubahan pada dataset sebelumnya. Dataset inilah yang digunakan sebagai data latih untuk membentuk rule dan pohon keputusan dalam menentukan Kelayakan rumah ibadah.

The image shows two screenshots of a web application interface. The top screenshot displays the 'Dataset' page, which includes a navigation bar with options like 'C45-PPKM', 'Atribut', 'Nilai Atribut', 'Dataset', 'Tree', 'Akurasi', 'Perhitungan', 'Password', and 'Logout'. Below the navigation bar, there are three buttons: 'Refresh', '+ Tambah', and 'Import'. The main content is a table with the following data:

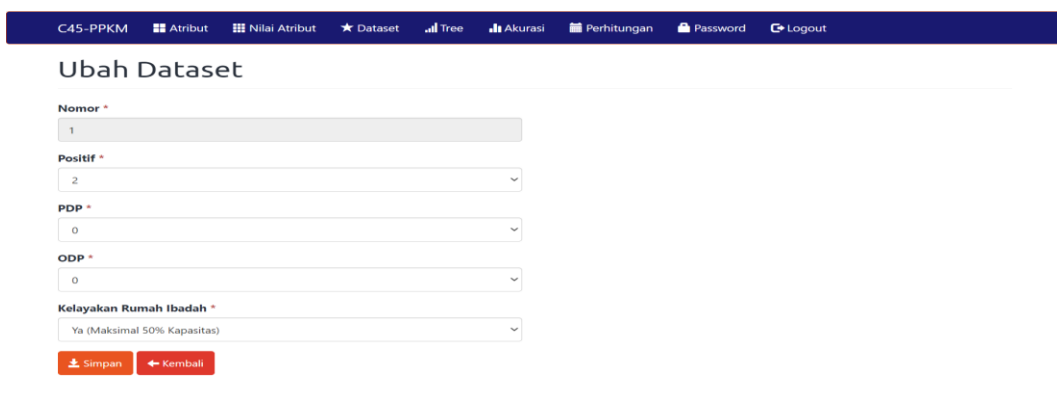
Nomor	Positif	PDP	ODP	Kelayakan Rumah Ibadah	Aksi
1	2	0	0	Ya (Maksimal 50% Kapasitas)	[Edit] [Delete]
2	4	8	0	Tidak	[Edit] [Delete]
3	2	6	12	Tidak	[Edit] [Delete]
4	3	4	0	Ya (Maksimal 50% Kapasitas)	[Edit] [Delete]
5	5	0	1	Ya (Maksimal 50% Kapasitas)	[Edit] [Delete]
6	3	0	8	Ya (Maksimal 50% Kapasitas)	[Edit] [Delete]
7	4	7	1	Tidak	[Edit] [Delete]
8	2	3	2	Ya (Maksimal 50% Kapasitas)	[Edit] [Delete]
9	1	9	1	Tidak	[Edit] [Delete]

The bottom screenshot shows the 'Import Dataset' page, which has the same navigation bar. It features a 'Pilih file' section with a 'Choose File' button and the text 'No file chosen'. Below this are two buttons: 'Import' and 'Kembali'. At the bottom of the page, there is a footer with the text 'Copyright © 2021 Universitas Muhammadiyah Ponorogo' and 'Tugas Akhir Kuliah'.

Gambar : 4.17 Halaman Import Dataset



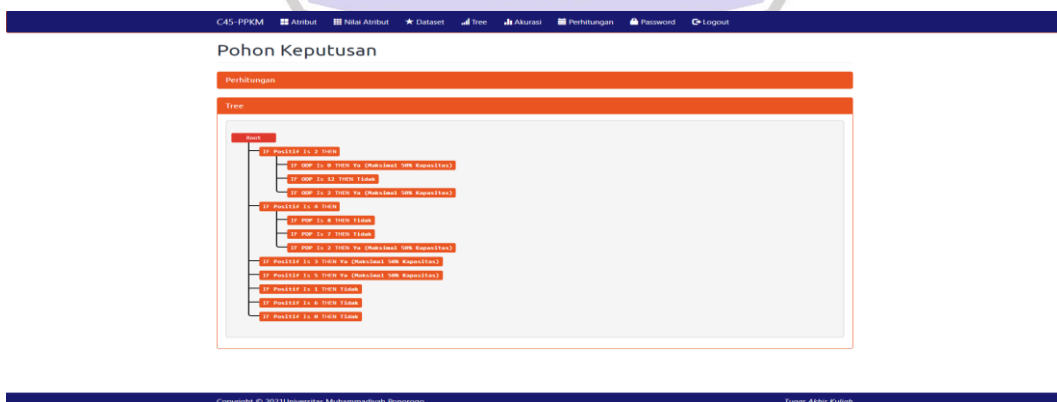
Gambar : 4.18 Halaman Tambah Dataset



Gambar : 4.19 Halaman Ubah Dataset

H. Menu Tree

Pada Menu ini sistem akan menampilkan hasil dari proses perhitungan algoritma C4.5 terhadap dataset yang telah di inputkan. Outputnya berupa proses perhitungan dan hasil dari pohon keputusan yang di bentuk.



Gambar : 4.20 Halaman Tree

I. Menu Akurasi

Pada menu akurasi merupakan menu yang di gunakan untuk melakukan pengecekan tingkat akurasi perhitungan algoritma C4.5 terhadap dataset yang di inputkan. Jika angka presentasi tinggi, menandakan bahwa algoritma tersebut cocok di gunakan dalam melakukan perhitungan terhadap dataset yang ada.

Nomor	Positif	PDP	ODP	Kelayakan Rumah Ibadah	C45	Benar
1	2	0	0	Ya (Maksimal 50% Kapasitas)	Ya (Maksimal 50% Kapasitas)	✓
2	4	8	0	Tidak	Tidak	✓
3	2	6	12	Tidak	Tidak	✓
4	3	4	0	Ya (Maksimal 50% Kapasitas)	Ya (Maksimal 50% Kapasitas)	✓
5	5	0	1	Ya (Maksimal 50% Kapasitas)	Ya (Maksimal 50% Kapasitas)	✓
6	3	0	8	Ya (Maksimal 50% Kapasitas)	Ya (Maksimal 50% Kapasitas)	✓
7	4	7	1	Tidak	Tidak	✓
8	2	3	2	Ya (Maksimal 50% Kapasitas)	Ya (Maksimal 50% Kapasitas)	✓
9	1	9	1	Tidak	Tidak	✓
10	2	24	12	Tidak	Tidak	✓
11	6	0	0	Tidak	Tidak	✓
12	6	0	0	Tidak	Tidak	✓
13	0	12	0	Tidak	Tidak	✓
14	0	0	36	Tidak	Tidak	✓
15	5	0	8	Ya (Maksimal 50% Kapasitas)	Ya (Maksimal 50% Kapasitas)	✓
16	5	0	2	Ya (Maksimal 50% Kapasitas)	Ya (Maksimal 50% Kapasitas)	✓
17	5	0	3	Ya (Maksimal 50% Kapasitas)	Ya (Maksimal 50% Kapasitas)	✓
18	4	2	2	Ya (Maksimal 50% Kapasitas)	Ya (Maksimal 50% Kapasitas)	✓
19	5	1	1	Ya (Maksimal 50% Kapasitas)	Ya (Maksimal 50% Kapasitas)	✓

Gambar : 4.21 Halaman akurasi

J. Menu perhitungan

Pada halaman inilah yang akan di jadikan sebagai pengambilan keputusan terkait Kelayakan rumah ibadah. Pada halaman ini pengguna melakukan input data covid-19 di suatu lingkungan selama kurun waktu 7 hari terakhir. Dari data tersebut sistem akan melakukan pencocokan terhadap rule dan pohon keputusan yang terbentuk dari dataset. Dari rule dan pohon keputusan tersebut di cari aturan yang sama ataupun mendekati dengan data yang ingin di uji. Jika tidak terdapat aturan yang sama dan tidak ada pula aturan yang mendekati dengan data yang ingin di uji maka sistem tidak dapat mendefinisikan data uji. Sehingga data uji di tentukan dengan cara perhitungan manual dengan dasar aturan PPKM mikro dan indikator

epidemiologi. Untuk selanjutnya data tersebut dapat tambahkan ke dalam dataset untuk di bentuk rule dan pohon keputusannya.

Data yang diketahui

Positif * 1

PDP * 8

ODP * 12

Hitung

Perhitungan

Tree

Hasil

Jika Positif = 1 dan PDP = 8 dan ODP = 12 maka = Tidak

Copyright © 2021 Universitas Muhammadiyah Ponorogo Tugas Akhir Kuliah

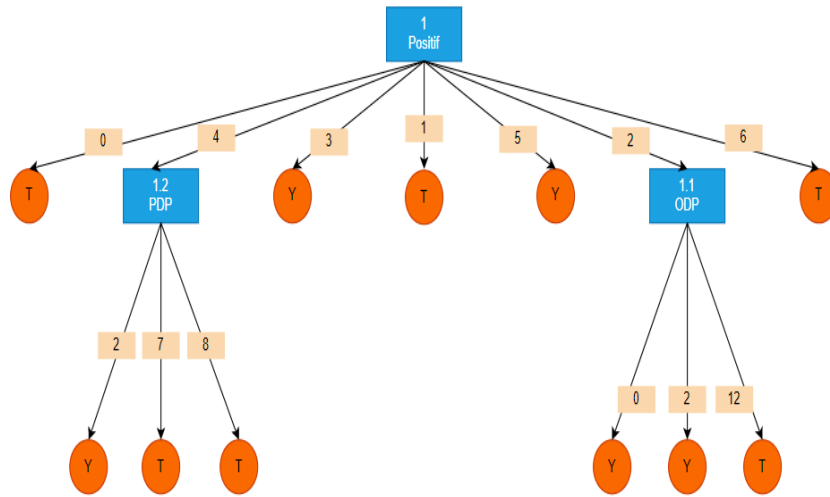
Gambar : 4.22 Halaman Perhitungan

- K. Menu Password
Merupakan menu yang dapat digunakan untuk melakukan perubahan password bagi admin.
- L. Menu logout
Merupakan menu yang di peruntukkan bagi admin jika ingin keluar dari sistem.

4.4. Pengujian Sistem

Pengujian sistem merupakan suatu proses yang dilakukan untuk menguji apakah algoritma yang di jalankan oleh sistem telah berjalan sesuai yang diharapkan. Dalam menguji kesesuaian perhitungan sistem yang di bangun, penulis melakukan perhitungan manual pada dataset yang sama menggunakan aplikasi ms excel, untuk selanjutnya dari perhitungan manual dan perhitungan yang dilakukan oleh sistem di lihat apakah memiliki aturan dan membentuk pohon keputusan yang sama. Setelah membandingkan antara perhitungan manual dan perhitungan yang dilakukan oleh sistem, diperoleh hasil dan terbentuk pohon keputusan yang sama

persis. Sehingga dapat ditarik kesimpulan bahwa perhitungan yang dilakukan oleh sistem telah berjalan baik sesuai dengan algoritma yang digunakan (Algoritma C4.5).



Gambar : 4.23 Tree hasil perhitungan manual



Gambar : 4.24 Tree hasil perhitungan sistem

Tahap pengujian selanjutnya yaitu dengan adanya penentuan indikator Pasien Dalam Pemantauan (PDP) dan Orang Dalam Pengawasan (ODP) juga proses pengelompokan zonasi untuk kedua indikator tersebut, Maka perhitungan untuk indikator PDP dan ODP di terapkan ke dalam sistem sebagai atribut tambahan,

Sehingga hasil pemodelan C4.5 pada sistem mampu melakukan perincian proses klasifikasi lebih baik dan detail dengan penambahan atribut PDP dan ODP.

The screenshot shows a web interface with the following sections:

- Data yang diketahui:** Three dropdown menus for 'Positif *' (value 1), 'PDP *' (value 8), and 'ODP *' (value 12). Below them is a red 'Hitung' button.
- Perhitungan:** A section header.
- Tree:** A section header.
- Hasil:** A section containing the text: 'Jika Positif = 1 dan PDP = 8 dan ODP = 12 maka = Tidak'.

At the bottom of the interface, there is a footer with the text: 'Copyright © 2021 Universitas Muhammadiyah Ponorogo' and 'Tugas Akhir Kuliah'.

Gambar : 4.25 Klasifikasi sistem

4.5. Source code program (Algoritma C4.5)

func_c45.php

Merupakan syntax program yang digunakan untuk melakukan proses perhitungan algoritma C4.5 pada program yang di bangun.

```
<?php
function convert_testing($testing, $cols = array(), $dataset)
{
    if(!$cols)
        return $testing;
    $arr = array();
    foreach ($dataset as $key => $val) {
        foreach ($cols as $k) {
            $arr[$k][] = $val[$k];
        }
    }
    $arr2 = array();
    $count = count($dataset);
```



```

$pebagi = $count / 4;

foreach ($arr as $key => $val) {
    asort($val);
    $i = 1;
    foreach ($val as $k => $v) {
        $arr2[$key][ceil($i / $pebagi)][] = $v;
        $i++;
    }
}
$arr3 = array();
foreach ($arr2 as $key => $val) {
    foreach ($val as $k => $v) {
        $arr3[$key][$k] = max($v);
    }
}
$arr4 = array();
foreach ($testing as $key => $val) {
    foreach ($cols as $k) {
        foreach ($arr3[$k] as $a => $b) {
            if ($val[$k] <= $b) {
                $testing[$key][$k] = '<=' . number_format($b, 2);
                break 1;
            }
        }
    }
}
//echo '<pre>' . print_r($dataset, 1) . '</pre>';
return $testing;
}

/**

```

```

* Class C45
* Melakukan proses pembentukan pohon keputusan berdasarkan dataset
* Melakukan prediksi berdasarkan pohon keputusan yang dihasilkan
*/
class c45
{
    protected $data; //dataset
    protected $target; //atribut target
    protected $target_values; //label semua atribut target
    protected $atribut; //data atribut

    protected $tree; //pohon keputusan
    protected $atribut_values; //nilai dari masing-masing atribut

    protected $counter; //menghitung berapa total cabang yang sudah dicari
    protected $is_debug; //apakah ingin menampilkan proses perhitungan

    function __construct($data, $atribut, $target, $is_debug = false)
    {
        $this->data = $data;
        $this->atribut = $atribut;
        //menghapus atribut terakhir (target)
        array_pop($this->atribut);
        $this->target = $target;
        //menghitung nilai yang ada untuk atribut target
        $this->target_values = $this->possible_values($data, $target);
        $this->is_debug = $is_debug;
        //memanggil fungsi hitung
        $this->hitung();
    }
}
/**
 * menampilkan pohon keputusan

```

```

*/
public function display()
{
    echo "<ul class='c45_tree'><li><a href='javascript:void(0)' class='btn
btn-xs btn-danger'>Root</a></li>";
    $this->_display($this->tree);
    echo "</ul>";
}
/**
 * fungsi recursif untuk menampilkan pohon keputusan
 * @param array $tree Pohon keputusan induk
 */
public function _display($tree)
{
    echo "<ul>";
    foreach ($tree['next'] as $key => $val) {
        echo "<li><a href='javascript:void(0)' class='btn btn-xs btn-
primary'> IF <b>$tree[value]</b> Is <b>$key</b> THEN";
        if (!$val['next'])
            echo " <b>$val[value]</b>";
        echo "</a>";
        //jika masih ada anak, maka panggil _display lagi
        $this->_display($val);
        echo '</li>';
    }
    echo "</ul>";
}
/**
 * memprediksi hasil berdasarkan inputan user
 * @param array $values Inputan user
 */
function predict($values)

```

```

{
    return $this->_predict($this->tree, $values);
}
/**
 * fungsi rekursif memprediksi hasil berdasarkan inputan user
 * @param array $tree Pohon keputusan
 * @param array $values Inputan user
 */
function _predict($tree, $values)
{
    if (!$tree['next'])
        return $tree['value'];

    $value = $values[$tree['value']];

    if (isset($tree['next'][$value])) {
        //jika masih ada anak, maka panggil _predict lagi
        return $this->_predict($tree['next'][$value], $values);
    }

    // jika tidak ditemukan cabang sesuai inputan
    return 'Undefined';
}
/**
 * melakukan proses perhitungan untuk membuat pohon keputusan
 */
function hitung()
{
    $this->counter = 1;
    $this->_hitung($this->tree, $this->data, $this->atribut, 'Root');
}
/**

```

* fungsi recursif melakukan proses perhitungan untuk membuat pohon keputusan

```
* @param array $tree Pohon keputusan
* @param array $data Dataset
* @param array $atribut Atribut
* @param string|null $attr_value Nilai atribut
*/
function _hitung(&$tree, $data, $atribut, $attr_value = null)
{

    // echo '<pre>' . print_r($atribut, 1) . '</pre>';

    $this->counter++;
    //jika perulangan lebih dari 10000 maka akan distop untuk mencegah
error jika data terlalu kompleks
    if ($this->counter > 10000)
        return;

    //jumlah label
    $target_count = $this->possible_values($data, $this->target);

    if (count($target_count) == 1) { // jika hanya 1 kemungkinan, maka h
asil sudah ditemukan
        $this-
>dd("\n===Hasil Cabang <b>$attr_value</b>:" . key($target_count) . "===
=");
        $tree['value'] = key($target_count);
        $tree['next'] = array();
        $this->dd("\n");
        return;
    } else { //jika lebih dari 1 lanjut ke bawah
        $this->dd("\n===Perhitungan Cabang <b>$attr_value</b>====");
```

```

}
//gain terbaik
$best_gain = -1;
//atribut terbaik
$best_atribut = 'None';

//jika atribut sudah habis, maka berhenti
if (!$atribut) {
    arsort($target_count);
    $this-
>dd("\n===Hasil Cabang <b>$attr_value</b>:" . key($target_count) . "===
=");
    $tree['value'] = key($target_count);
    $tree['next'] = array();
    $this->dd("\n");
    return;
}
//lakukan perhitungan untuk semua atribut
foreach ($atribut as $attr) {
    $this->dd("\n<span class='text-primary'>$attr</span>:");
    //menghitung gain
    $gain = $this->gain($data, $attr);
    //menghitung split_info
    $split_info = $this->split_info($data, $attr);
    //mengitung gain ratio
    $gain_ratio = $split_info == 0 ? $gain : $gain / $split_info;
    $this->dd("\n\t<b class='text-
info'>GAIN</b>: " . round($gain, 3) . "");
    $this->dd("\n\t<b class='text-
info'>SPLIT INFO</b>: " . round($split_info, 3) . "");
    $this->dd("\n\t<b class='text-
info'>GAIN RATIO</b>: " . round($gain_ratio, 3) . "");
}

```

```

//memilih gain dan atribut terbaik
if ($gain_ratio > $best_gain) {
    $best_gain = $gain_ratio;
    $best_atribut = $attr;
}
}

//menampilkan atribut terbaik (gain terbesar)
$this->dd("\n<b class='text-
success'>Atribut terbaik</b>: $best_atribut (" . round($best_gain, 3) . ")");

//menghitung nilai untuk atribut terbaik
$p = $this->possible_values($data, $best_atribut);
//menghapus atribut terbaik agar tidak dihitung lagi
unset($atribut[$best_atribut]);

$this->dd("\n");

foreach ($p as $val => $count) {
    //memfilter data dengan cara menghilangkan data yang berisi atribu
t terbaik
    $new_data = $this->filter_data($data, $best_atribut, $val);
    $tree['value'] = $best_atribut;
    //menghitung lagi untuk semua nilai atribut terbaik
    $this-
>_hitung($tree['next'][$val], $new_data, $atribut, "$best_atribut($val)");
}
}
/**
 * memfilter data dengan menghilangkan nilai atribut terbaik
 * @param array $data Dataset
 * @param string $best_attr Nama atribut terbaik

```

```

* @param string $value nilai atribut terbaik
*/
function filter_data($data, $best_attr, $value)
{
    $arr = array();
    foreach ($data as $val) {
        if ($val[$best_attr] == $value) {
            unset($val[$best_attr]);
            $arr[] = $val;
        }
    }
    return $arr;
}
/**
* menghitung split info
* @param array $data Dataset
* @param string $attr Atribut
*/
function split_info($data, $attr)
{
    $values = $this->possible_values($data, $attr);
    $split_info = 0.0;
    $pembagi = array_sum($values);
    foreach ($values as $value => $count) {
        $split_info += $count == 0 ? 0 : $count / $pembagi * log($count /
$split_info, 2);
    }
    return -$split_info;
}
/**
* menghitung nilai gain
* @param array $data Dataset

```



```

* @param string $attr Atribut
*/
function gain($data, $attr)
{
    $values = $this->possible_values($data, $attr);
    $total = count($data);
    $gain = 0.0;
    foreach ($values as $value => $count) {
        $e = $this->entropy($data, $attr, $value);
        $this->dd("\n\t<span class='text-
danger'>$value</span>($count/$total): " . round($e, 3));
        $gain += $e * $count / $total;
    }
    $e = $this->entropy($data);
    return $e - $gain;
}
/**
* menghitung nilai entropy
* @param array $data Dataset
* @param string|null $attr Nama atribut
* @param string|null $value Nilai atribut
*/
function entropy($data, $attr = null, $value = null)
{
    $p = $this->calculate_p($data, $attr, $value);
    $entropy = 0.0;
    foreach ($p as $key => $val) {
        $entropy -= $val == 0 ? 0 : $val * log($val, 2);
    }
    return $entropy;
}
/**

```

```

* menghitung nilai probabilitas (p)
* @param array $data Dataset
* @param string|null $attr Nama atribut
* @param string|null $vakue Nilai atribut
*/
function calculate_p($data, $attr, $attr_value)
{
    $p = array();

    foreach ($this->target_values as $key => $val) {
        $p[$key] = 0;
    }
    foreach ($data as $val) {
        if ($attr == null) {
            $p[$val[$this->target]]++;
        } else if ($val[$attr] == $attr_value) {
            $p[$val[$this->target]]++;
        }
    }
    $p_total = array_sum($p);
    foreach ($p as $key => &$val) {
        $val /= $p_total;
    }
    return $p;
}
/**
* menghitung pilihan nilai yang ada untuk atribut tertentu
* @param array $data Dataset
* @param string|null $attr Nama atribut
*/
function possible_values($data, $attr)
{

```

```

    $arr = array();
    foreach ($data as $val) {
        $arr[$val[$attr]] = array_key_exists($val[$attr], $arr) ? $arr[$val[$
attr]] + 1 : 1;
    }
    // echo '<pre>' . print_r($arr, 1) . '</pre>';
    return $arr;
}
/**
 * menampilkan proses perhitungan jika is_debug bernilai true
 * @param string $str teks yang ditampilkan
 */
function dd($str)
{
    if ($this->is_debug)
        echo "$str";
}
}

```

