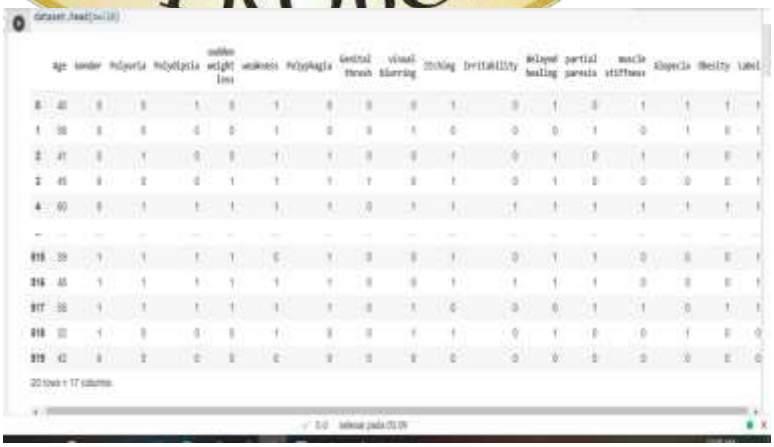
BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil Penyajian Data dan Implementasi

Pada bab ini akan di jelaskan hasil penelitian tentang mengklasifikasi penyakit diabetes menggunakan metode *decision tree c4.5* dengan menerapkan implementasi algoritma. Tahapan penelitian ini telah di jelaskan pada bab sebelumnya akan mengimplementasikan pada bab ini. Antara lain meliputi hasil dari pengumpulan dataset, hasil *preprocessing*, hasil implementasi *algoritma c4.5* dan hasil uji coba dari metode tersebut.

4.1.1 Pengumpulan data set

Pada penelitian ini pengambilan data di lakukan untuk mempermudah peneliti dalam melakukan penelitian. Peneliti mengambill Dataset dari Metropolitan University, Sylhet, Bangladeshrahmansadik004 '@' gmail.com(https://archive.ics.uci.edu/ml/machine-learning databases/00529/diabetes_data_upload.csv) dari semua dataset di banglades terdiri dari 17 variabel yang Jumblah keseluruhan dataset tersebut sebanyak 520 data, dengan jumlah pasien positive sebanyak 320 dan jumblah pasien yang negative sebanyak 200. Dataset awal berupa data nominal Yes dan No kemudian di ubah kedalam bentuk biner 0 atau 1 yang artinya jika 0 maka Negatif dan jika 1 maka positif (Fajri). Berikut ini data yang di gunakan untuk implementasi algoritma decision tree c4.5 pada **gambar 4.1**



Gambar 4. 1 Dataset Banglades

Keterangan :

Age = usia

Gender = Jenis kelamin

Polyuria = sering buang air kecil

Polydipsia = sering haus

Sudden weight los = Penurunan Berat Badan

Weaknes = Kelemahan

Polyphagia = banyak makan

Genital thursh = Sariawan Genita

Visual blurring = nyeri pada lutut

Itching = Gatal

Irrabilitas = lekas marah

Delayed healing = penyembuhan Tertunda

Paresis Parsial = lampuh karena gangguan saraf

Mucles stiffness = ketidakseimbangan Otot

Alopecia = kerontokan rambut

Obesitas = kegemukan

Label = tingkatan

Dari data banglades berupa data numeric dan di ganti dengan data nominal menggunakan manual keterangan :

Tabel 4. 1 Nominal menjadi numerik

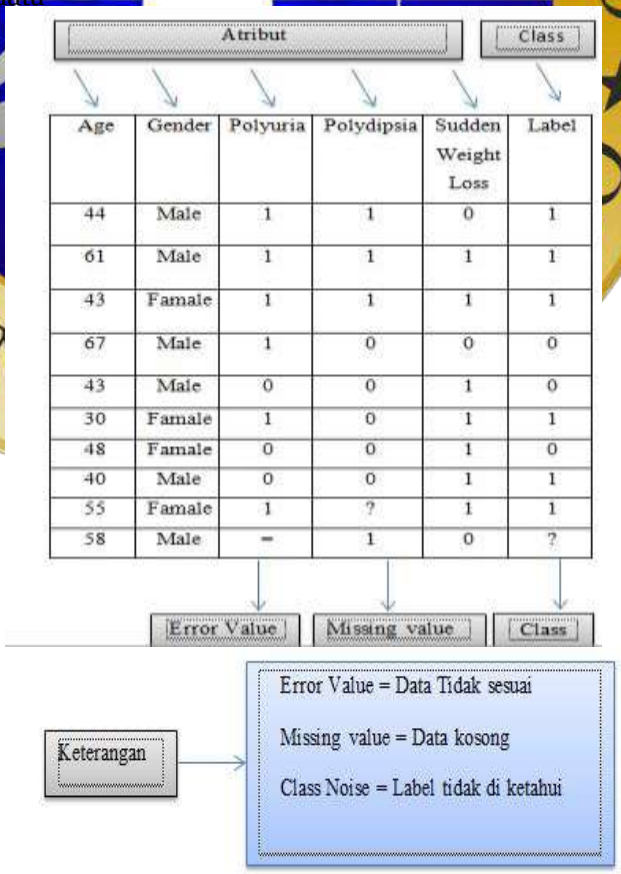
Data Awal	Nominal	Artinya
YES	1	Positive
NO	0	Negative
Gender	Pria 0	Wanita 1

4.1.2 Prosesing Data

Preprocessing yaitu pembersihan data agar mendapatkan data yang baik dan data yang berkualitas untuk di lakukan proses data mining, proses preprocessing di bagi menjadi 4 yaitu :

- Pembersian data di gunakan untuk menghilangkan atau melengkapi data yang tidak lengkap, menghaluskan data noisy, serta memperbaiki data yang kurang konsisten

Contoh data



Gambar 4. 2 Contoh data Noise

- Cara menangani data yang tidak lengkap atau kurang konsisten yaitu :
  a. Menghilangkan atau mengabaikan
  b. Mengisi nilai dengan manual.

  Contoh data tidak lengkap

Age	Gender	Polyuria	Polydipsia	Sudden Weight Loss	Label
44	Male	1	1	0	1
61	Male	?	1	1	1
43	Famale	1	1	1	1
67	Male	1	0	0	0
43	Male	0	0	1	0
30	Famale	1	?	1	1
48	Famale	0	0	1	0
40	Male	0	0	?	1
55	Famale	1	0	1	1
58	Male	1	1	0	1

↓

Age	Gender	Polyuria	Polydipsia	Sudden Weight Loss	Label
44	Male	1	1	0	1
61	Male	1	1	1	1
43	Famale	1	1	1	1
67	Male	1	0	0	0
43	Male	0	0	1	0
30	Famale	1	0	1	1
48	Famale	0	0	1	0
40	Male	0	0	1	1
55	Famale	1	0	1	1
58	Male	1	1	0	1

Gambar 4. 3 Data tidak lengkap

c. Integrasi data

   Integrasi banyak database,banyak kubus data, atau banyak file.

d. Transformasi data Normalisasi dan agregasi

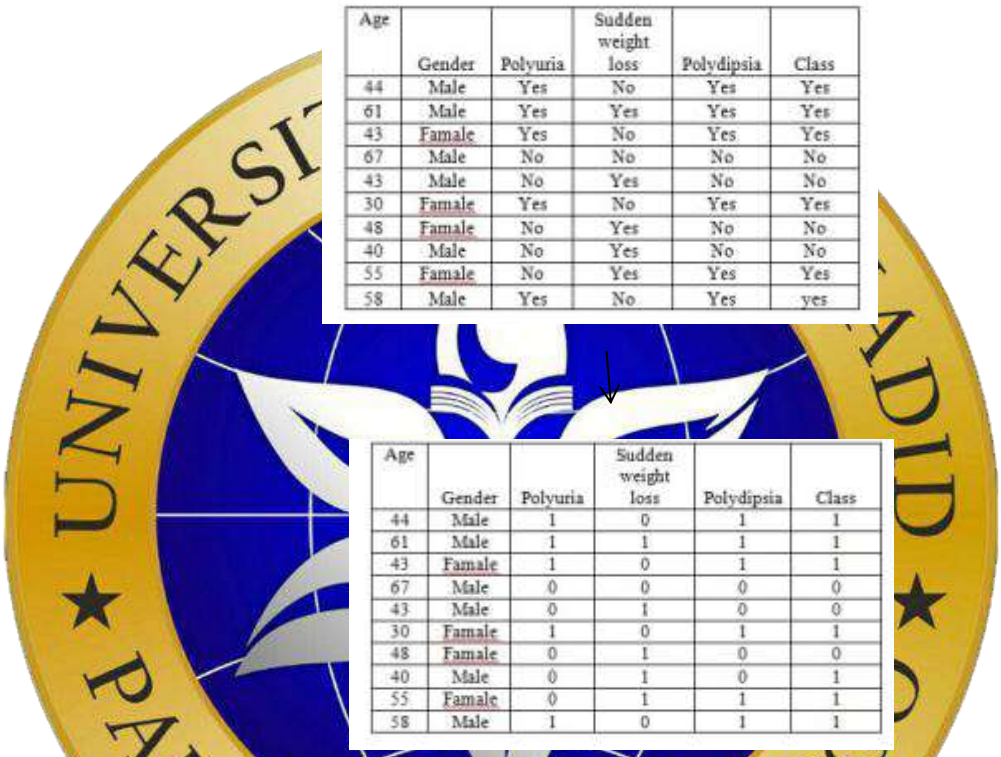
e. Reduksi data

Mendapatkan representasi yang di reduksi dalam volume tetapi menghasilkan analitikal yang sama atau mirip.

f. Diskritasi data

Diskritasi data adalah mengubah tipe data nominal menjadi numeric.

- Contoh data nominal menjadi data numeric



Age	Gender	Polyuria	Sudden weight loss	Polydipsia	Class
44	Male	Yes	No	Yes	Yes
61	Male	Yes	Yes	Yes	Yes
43	Famale	Yes	No	Yes	Yes
67	Male	No	No	No	No
43	Male	No	Yes	No	No
30	Famale	Yes	No	Yes	Yes
48	Famale	No	Yes	No	No
40	Male	No	Yes	No	No
55	Famale	No	Yes	Yes	Yes
58	Male	Yes	No	Yes	yes

Age	Gender	Polyuria	Sudden weight loss	Polydipsia	Class
44	Male	1	0	1	1
61	Male	1	1	1	1
43	Famale	1	0	1	1
67	Male	0	0	0	0
43	Male	0	1	0	0
30	Famale	1	0	1	1
48	Famale	0	1	0	0
40	Male	0	1	0	1
55	Famale	0	1	1	1
58	Male	1	0	1	1

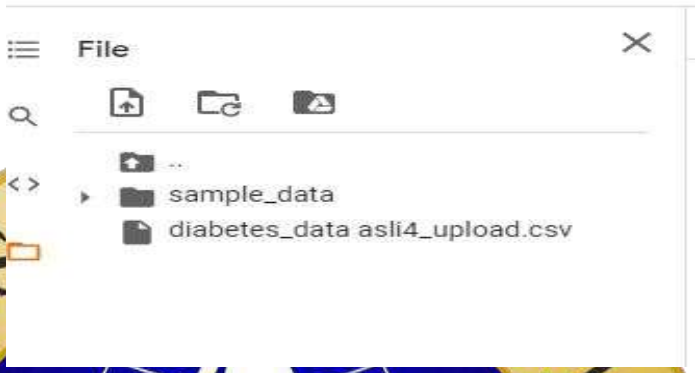
Gambar 4. 4 Nominal menjadi numeric

4.1.4 Implementasi Algoritma Decision tree C4.5

Pada tahap ini di jelaskan pemprosesan dataset kemudian akan di implementasikan menggunakan *algoritma decision tree c4*.5 adalah sebuah pemprosesan sala satu program dalam melakukan klasifikasi penyakit diabetes. Di mana *decision tree* adalah sebuah setruktur yang akan di gunakan untuk membagi data mentah yaitu untuk data training 80% dan data testing 20% dengan menerapkan sebuah pohon keputusan. Pada metode *decision tree* dengan menggunakan python

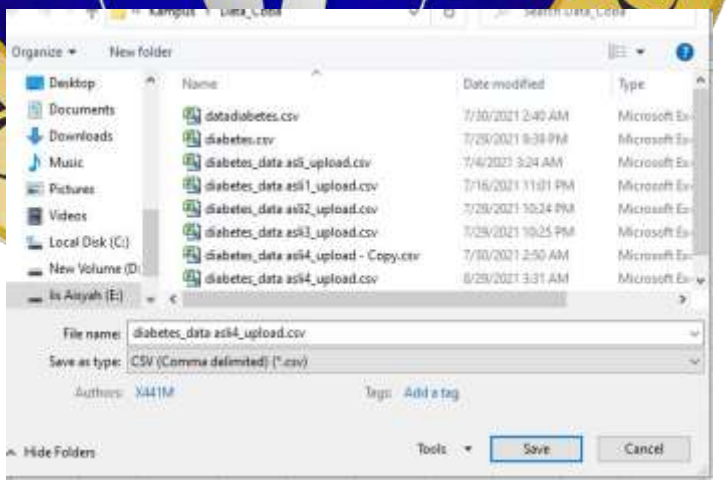
Berikut langkah- langkah untuk mengklasifikasikan penyakit diabetes dengan menggunakan algoritma decision tree c4.5 sebagai berikut :

1. Tahap pertama yang perlu di lakukan adalah uplot file Dataset untuk di jadikan sampel data.



Gambar 4. 5 Uploat file

2. Klik file_Uploat to session storage.
   Pilih file yang akan di uplout dengan menggunakan format csv atau comma separated Value. Lebih jelas liat pada **gambar 4.6**



Gambar 4. 6 pilih file yang akan di uplout

3. Mengimport library yang di butuhkan serta memanggil model yang telah di simpan.

```
import numpy as np
import pandas as pd
from sklearn.tree import plot_
tree
```

Segmen Program 4. 1 Proses library

Import numpy as np di gunakan untuk komputasi metric, import pandas as pd di gunakan untuk managemen data dari sumber luar sala satunya csv, from sklearn.tree import plot_tree di gunakan untuk memanggil decision tree.

4. Membaca dataset dari file ke pandas dataFrame.

```
dataset=pd.read_csv('/content/diabetes
_data asli4_upload.csv')
dataset.head(n=520)
```

Segmen Program 4. 2 Membaca Dataset

Dataset = pd.read_csv(‘/content/diabetes_data asli4_upload.csv’) yang di gunakan untuk memanggil dataset yang sudah di uplout).

5. Tampilan Dataset

Dataset yang tampil adalah 520 dataset dengan jumblah 16 variabel.

Gambar 4. 7 Tampilan Datase Banglades

6. Memisahkan antara kriteria (X) dengan output (Y)

```
X=dataset.drop('Label',axis=1)
Y=dataset.Label
```

Segmen Program 4. 3 Memisahkan output

X=dataset.drop(‘Label’ ,axsis=1)

Y =dataset.Label

Di gunakan untuk memisahkan kreteria (X) dengan output (Y) untuk kreteria ada 16 yaitu (Age, Gender, polyuria, polydipsia, sudden weight loss, weakness ,polyphagia ,genital thrush, visual blurring, itcting, irritability ,delayed healing , partial paresis, muscle stiffness, Alopecia, obesity,) dan untuk outpout ada 1 yaitu label. Untuk keterangan outpout yaitu 1 untuk positive dan 0 untuk negative.

7. Mengimport model decision tree

```
from sklearn.model_selection imp
ort train_test_split
from sklearn.tree import Decisio
nTreeClassifier
import matplotlib.pyplot as plt
```

Segmen Program 4. 4 Mengimport model decision tree

From sklear.model_selection import train_test_split berguna untuk model decision tree, from skearn.tree import decisionTreeClassifier untuk memanggil fangsen decision tree, dan import matplotlib pyplot as plt untuk gambar.

8. Mengubah dataframer ke array numpy

```
dataset = dataset.to_numpy()
```

Segmen Program 4. 5 Mengubah data

Semula datanya seperti gambar **4.13**

	Age	Gender	Polyuria	...	Alopecia	Obesity	Label
0	40	0	0	...	1	1	1
1	58	0	0	...	1	0	1
2	41	0	1	...	1	0	1
3	45	0	0	...	0	0	1
4	60	0	1	...	1	1	1
..	...	...	...	...	...	...	...
515	39	1	1	...	0	0	1
516	48	1	1	...	0	0	1
517	58	1	1	...	0	1	1
518	32	1	0	...	1	0	0
519	42	0	0	...	0	0	0

[520 rows x 17 columns]

Gambar 4. 8 Data awal

Kemudian di ubah dalam bentuk numpy dan hasilnya akan berubah pada gambar di bawah ini

```
[[40  0  0 ...  1  1  1]
 [58  0  0 ...  1  0  1]
 [41  0  1 ...  1  0  1]
 ...
 [58  1  1 ...  0  1  1]
 [32  1  0 ...  1  0  0]
 [42  0  0 ...  0  0  0]]
```

Gambar 4. 9 Data di rubah menjadi Array

9. Membagi dataset yaitu data training dan data testing Untuk data training (80%) dan data Testing (20%)

```
dataTraining = np.concatenate((dataset[0:208,
:], dataset[260:468, :]),
                              axis=0)
dataTesting = np.concatenate((dataset[208:260,
:], dataset[468:520, :]),
                              axis=0)
```

Segmen Program 4. 6 Memisahkan data training dan data testing

10. Tampilan untuk data Training

```
print(dataTraining)
len(dataTraining)
```

```
[[40  0  0 ...  1  1  1]
 [58  0  0 ...  1  0  1]
 [41  0  1 ...  1  0  1]
 ...
 [27  0  0 ...  0  0  0]
 [38  0  0 ...  0  0  0]
 [43  0  0 ...  1  0  0]]
416
```

Gambar 4. 10 Tampilan untuk data training

Dari seluruh dataset terdapat 520 data dan yang akan di jadikan data training yaitu 80% maka akan menghasilkan 416 data training.

11. Tampilan data Testing

```
print(dataTesting)
len(dataTesting)
```

```
[[54  0  0 ...  1  0  0]
 [43  0  0 ...  1  0  0]
 [39  0  0 ...  0  0  0]
 ...
 [58  1  1 ...  0  1  1]
 [32  1  0 ...  1  0  0]
 [42  0  0 ...  0  0  0]]
104
```

Gambar 4. 41 Menampilkan data testing

Dari seluruh dataset terdapat 520 data dan yang akan di jadikan data testing yaitu 20% maka akan menghasilkan 104 data testing

12. Melakukan prediksi

```
X_train,X_test,Y_train,Y_test=train_test_spl
it(X,Y,test_size=0.2)

dt.fit(X_train,Y_train)

Y_pred=dt.predict(X_test)

from sklearn.metrics import accuracy_score,cla
ssification_report,confusion_matrix

print('Akurasi:',accuracy_score(Y_test,Y_pred)
)
```

Segmen Program 4. 7 Menampilkan prediksi

a) X_tain, X_test, Y_train, y_test_split(X,Y,test_size=0.2) untuk memisahkan antara data testing dan training.

b) dt.fit(X_train, Y_Train) untuk memodelkan decision tree

c) y_pred=dt.predict(X_test) untuk menyiman nama prediksi
d) From sklearn.matrics import untuk mengimport matrics dan yang akan di import accuracy_score, classification_report,confusion_matrix.
e) print (‘akurasi’accuracy_score(Y_test,Y_pred) berguna untuk menjalankan hasil.

13. Hasil keseluruhan

Setelah melakukan semuanya maka akan menghasilkan nilai akhir.

```
print('Hasil:',classification_report(Y_test,Y_pred))
```

```
Hasil:                precision    recall  f1-score   support

           0       0.95      0.97      0.96        38
           1       0.98      0.97      0.98        66

    accuracy                           0.97       104
   macro avg       0.97      0.97      0.97       104
weighted avg       0.97      0.97      0.97       104
```

Gambar 4. 12 hasil keseluruhan

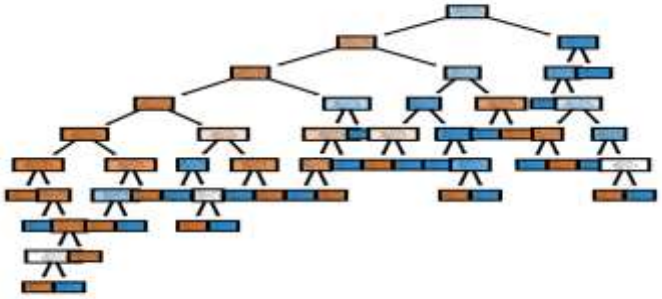
14. Algoritma decision tree

Pada tahap ini akan menghasilkan suatu pohon keputuan di mana pohon keputusan lebih mempermudah dalam pengambilan keputusan.

```
fn=['Age','Gender','polyuria','polydipsia','sudde
n weight loss','weakness','Polyphagia','Genital t
hrush','visual blurring','Itching','Irritability'
,'delayed healing','partial paresis','muscle stif
fness','Alopecia','Obesity']
cn=['1', '0']
fig, axs = plt.subplots(nrows = 1,ncols = 1,figsi
ze = (3,3), dpi=1000)
plot_tree(dt,
      feature_names = fn,
      class_names=cn,
      filled = True);fig.savefig('imagename.png')
```

Segmen Program 4. 8 menampilkan pohon keputusan

15. Hasil pohon keputusan



Gambar 4. 13 hasil pohon keputusan

Keterangan warna pada pohon keputusan :

Jika nilai 0 maka keputusannya Negative

Jika nilai 1 maka keputusannya Positive

= Negatif dengan samples lebih besar

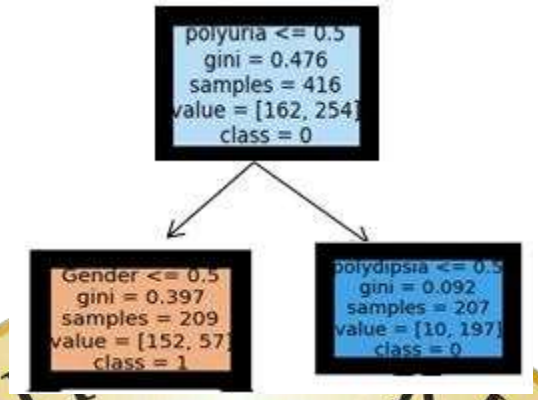
= Positive dengan samples lebih besar

= Negatif dengan samples lebih kecil

= Positive dengan samples lebih kecil

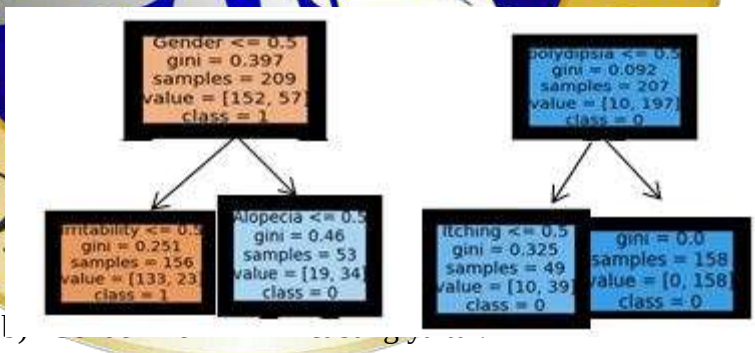
= Positive dengan samples paling kecil

Keterangan untuk pohon keputusan pada **gambar 4.20** sebagai berikut :



a) Polyuria sebagai not awal karena memilik nilai gain tertinggi dengan nilai 0.476 dan jumlah samples data 416 dengan class 0 (Negative). Kemudian membentuk 2 cabang yaitu :

- Gender memiliki nilai gain 0.397 dan jumlah samples 209 dengan class 1(Positive) .
- polydipsia memiliki nilai gain 0.092 dengan jumlah samples 207 dengan class 0 (Negative).



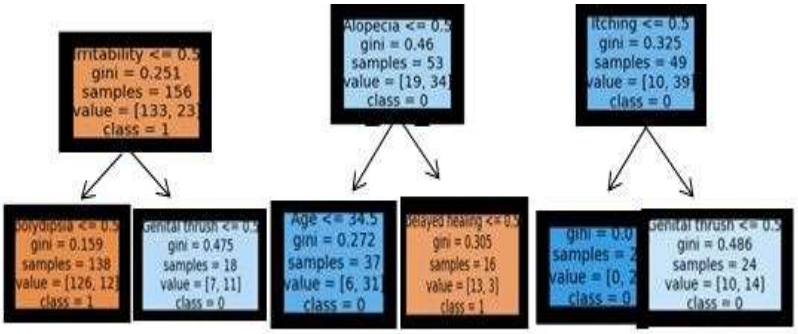
b)

- Irritability memiliki nilai gain 0.251 dengan jumlah samples 156 dengan class 1 (positive).
- Alopecia memiliki nilai gain 0.46 dengan jumlah sample 53 dengan class 0 (negative) .

c) Polydipsia memiliki 2 cabang yaitu :

- Itching memiliki nilai gain 0.325 dengan jumlah samples 49 dengan class 0 (Negstive).

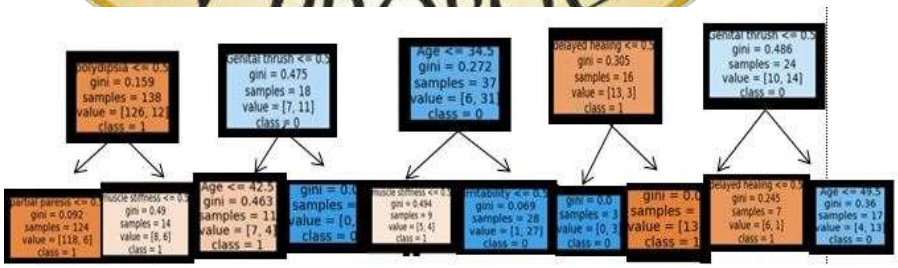
- Gain dengan nilai 0 dengan jumblah samples 158.



d) Irritability memiliki 2 cabang yaitu :
   - Polydipsia memiliki nilai gain 0.159 dengan jumlah sample 138 dengan class 1 (Positive).
   - Genetical thrush memiliki nilai gain 0.475 samples 18 dengan class 0 (Negative).

e) Alopecia memiliki 2 cabang yaitu :
   - Age memiliki nilai gain 0.272 dengan jumlah samples 37 dengan class 0 (Negative).
   - Delayed healing memiliki nilai gain 0.305 dengan jumlah samples 16 dengan class 1 (Positive).

f) Itcting memiliki 2 cabang yaitu :
   - Gain memiliki nilai 0.0 dengan jumlah samples 2 dengan class 0 (Negative).
   - Genetical thrush memiliki nilai gain 0.486 dengan jumlah samples 24 dengan class 0 (Negative).



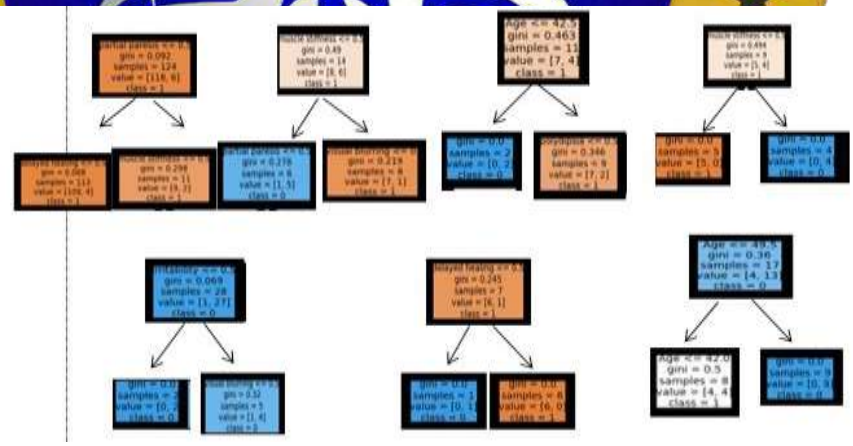
g) Polydipsia memiliki 2 cabang yaitu :
   - Partial paresis memiliki nilai gain 0.092 dengan samples 124 dengan class 1 (Positive).

- Muscle stiffness memiliki nilai gain 0.49 dengan jumlah samples 14 dengan class 1 (positive).

h) Genetical memiliki 2 cabang yaitu :
   - Age memiliki nilai gain 0.463 dengan samples 11 dengan class 1 (Positive).
   - Gain memiliki nilai 0.0 dengan jumlah samples 7 dengan class 0 (Negative).

i) Age memiliki 2 cabang yaitu
   - Muscle stifitness memiliki nilai gain 0.494 dengan jumlah samles 9 dengan class 1 (Positive).
   - Irritability memiliki nilai gain 0.069 dengan jumlah samples 28 dengan class 0 (Negative).

j) Delayed healing memiliki 2 cabang yaitu :
   - Gain memiliki nilai 0.0 dengan sample 13 dengan class 1 (Positive).
   - Age memiliki nilai gain 0.36 dengan jumlah samples 17 dengan class 0 (Negative).



k) Partial paresis memiliki 2 cabang yaitu :
   - Delayed healing dengan nilai gain 0.068 dengan samples 113 dengan class 1 (Positive)

- Muscles stiffness dengan nilai gain 0.298 dengan jumlah samples 11 dengan class 1 (Positive).

l) Muscles stiffens memiliki 2 cabang yaitu :
   - Partial paresis memiliki nilai gain 0.278 dengan jumlah samples 6 dengan class 0 (negative).
   - Visual blurring memiliki nilai gain 0.219 dengan jumlah samples 8 dengan class 1 (Positive).

m) Age memiliki 2 cabang yaitu :
   - Gain dengan nilai 0.0 dengan jumlah samples 2 dengan class 0 (Negative).
   - Polydipsia memiliki nilai gain 0.346 dengan jumlah samples 9 dengan class 1 (Positive).

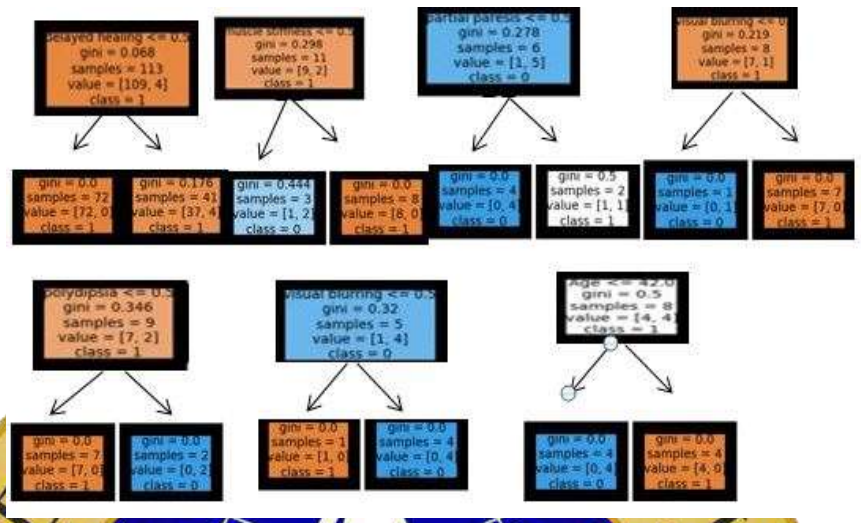
n) Muscle stiffness memiliki 2 cabang yaitu
   - Gain dengan nilai 0.0 dengan jumlah samples 5 dengan class 1 (Positive).
   - Gain dengan nilai 0.0 dengan jumlah samples 4 dengan class 0 (Negative).

o) Irritability memiliki 2 cabang yaitu :
   - Gain dengan nilai 0.0 dengan jumlah samples 23 dengan class 0 (Negative).
   - Visual blurring memiliki nilai gain 0.32 dengan jumlah samples 5 dengan class 0 (Negative).

p) Delayed healing memiliki 2 cabang yaitu :
   - Gain dengan nilai 0.0 dengan jumlah sampel 1 dengan class 0 (Negative).
   - Gain dengan nilai 0.0 dengan jumlah samples 6 dengan class 1 (Positive).

q) Age memiliki 2 cabang yaitu :
   - Age memiliki nilai gain 0.5 dengan jumlah samples 8 dengan class 1 (Positive).

- Gain dengan nilai 0.0 dengan jumlah samples 9 dengan class 0 (Negative).



r) Delayed healing memiliki 2 cabang yaitu :
   - Gain dengan nilai 0.0 dengan jumlah samples 72 dengan class 1 (Positive).
   - Gain dengan nilai 0.176 dengan jumlah samples 41 dengan class 1 (Positive).

s) Muscles stiffness memiliki 2 cabang yaitu :
   - Gain dengan nilai 0.298 dengan jumlah samples 3 dengan class 0 (Negative).
   - Gain dengan nilai 0.0 dengan jumlah samples 8 dengan class 1 (positive).

t) Partial paresis memiliki 2 cabang yaitu :
   - Gain dengan nilai 0.0 dengan samples 4 dengan class 0 (Negative).
   - Gain dengan nilai 0.2 dengan jumlah samples 2 dengan class 1 (Positive).

u) Visual blurring memiliki 2 cabang yaitu :
   - Gain dengan nilai 0.0 dengan jumlah samples 1 dengan class 0 (Positive).
   - Gain dengan nilai 0.0 dengan jumlah samples 7 dengan class 1 (Positive).

v) Polydipsia memiliki 2 cabang yaitu :

- Gain dengan nilai 0.0 dengan jumlah sampels 7 dengan class 1 (Positive)
- Gain dengan nilai 0.0 dengan jumlah samples 2 dengan class 0 (Negative).

w) Visual blurring memiliki 2 cabang yaitu :

- Gain dengan nilai 0.0 dengan jumlah samples 1 dengan class 1(Positive).
- Gain dengan nilai 0.0 dengan jumlah samples 4 dengan class 0 (Negative).

x) Age memili 2 cabang yaitu :

- Gain memiliki nilai 0.0 dengan samples 4 dengan class 0 (Negative).
- Gain dengan nilai 0.0 dengan jumlah samples 4 dengan class 1 (Positive).

4.2 Analisis Data

Hasil dari mengklasifikasikan penyakit diabetes dengan menggunakan decision tree c4.5 pada tahap analisis data akan di bahas pada setiap percobaan. Analisis di lakukan dengan menggunakan data testing pada dataset hasil uji coba yang di lakukan terhadap data training terdapat 80% dengan jumlah data 520 sehingga di dapatkan untuk data training 416 data. Dan data testing 104 data berhasil terdeteksi sesuai data dengan akurasi 0.97%