

BAB I

PENDAHULUAN

Proses data dalam bidang Data Mining membutuhkan jumlah data yang sangat besar. Banyak kendala pemrosesan data besar untuk diolah menjadi informasi dan menemukan pola data menarik. Hal ini dikaitkan dengan kekuatan perangkat keras dan juga kemampuan perangkat lunak dalam pemrosesan data yang besar.

Salah satu teknik atau metode dalam data mining adalah algoritma C-45. Algoritma ini menggunakan struktur pohon keputusan dengan cara melakukan pemrosesan data dalam jumlah besar, kemudian menyaring informasi berharga dan ditempatkan dalam bentuk struktur pohon. Tujuan dari model analisis C-45 adalah menemukan pola data data besar, dimana pola data dapat digunakan sebagai model pendukung keputusan terhadap data-data baru yang akan dianalisis.

Salah satu persoalan dalam proses C-45 adalah proses iterasi dalam membentuk struktur pohon. Proses iterasi yang dilakukan sangat bergantung pada kemampuan hardware dan juga teknik pemrograman yang dilakukan. Persoalan menjadi semakin kompleks ketika pendekatan pemrograman yang dilakukan tidak optimal dalam proses analisis.

Penelitian ini adalah penelitian yang bersifat implementatif, algoritma C-45 diimplementasikan secara sistem dalam bentuk prosedur yang sudah dioptimalkan. Tujuan akhir adalah produk berupa prosedur yang secara umum dapat diterapkan pada berbagai sistem database untuk analisis data mining.

BAB II LANDASAN TEORI

Algoritma C4.5 adalah algoritma yang didasarkan pada algoritma Tree atau pohon keputusan. Algoritma ini adalah algoritma turunan dari algoritma ID3 yang dikembangkan oleh J. Ross Quinlan. Dalam prosedur algoritma ID3, input yang digunakan adalah contoh data untuk training/pembelajaran sistem, label training dan atribut dari data. Algoritma C4.5 digunakan untuk mengubah bentuk data tabel menjadi model pohon, mengubah model pohon menjadi aturan/rule kemudian menyederhanakan rule hasil akhir.

Beberapa keuntungan dari penerapan algoritma C4.5 adalah sebagai berikut:

1. Keputusan yang sebelumnya kompleks dan global, diubah menjadi simpel dan spesifik.
2. Meminimalkan perhitungan yang tidak diperlukan, karena data yang diuji adalah data berdasarkan kriteria atau kelas tertentu.
3. Dinamis dalam pemilihan syarat analisis data sehingga pada akhirnya dapat meningkatkan kualitas keputusan.
4. Menghindari munculnya permasalahan karena algoritma C-45 selalu menggunakan kriteria yang lebih sedikit.

Dengan kelebihan pada poin 4, sekaligus sebenarnya menjadikan algoritma C4.5 memiliki keterbatasan dalam hal kualitas keputusan. Keputusan yang dihasilkan oleh algoritma C4.5 adalah keputusan yang mengacu pada bagaimana desain dari hasil pohon keputusan.

Algoritma C4.5 menggunakan dasar pengukuran *entropy* atau lebih dikenal sebagai *information gain*. Pengukuran ini memanfaatkan berbagai atribut yang secara heuristic dapat digunakan sebagai contoh *class* atau kategori yang akan dianalisis. Untuk konsep rumus pengukuran menggunakan dua rumusan utama yaitu perhitungan *entropy* dan perhitungan nilai *gain*.

Rumus entropy adalah sebagai berikut:

$$E(A) = \sum_{j=1}^v \frac{S_{1j} + \dots + S_{mj}}{S} (S_{1j}, \dots, S_{mj}),$$
 dimana A adalah attribute dengan nilai 1 sampai dengan v dan dapat dibagi menjadi beberapa partisi S dengan subset (1 ... m).

Oleh karena $\frac{S_{1j+\dots+S_{mj}}}{S}$ adalah partisi dari contoh class dibagi dengan jumlah keseluruhan data yang dihitung, maka nilai terkecil dari *entropy* adalah subset terbesar dari partisi data yang digunakan.

Rumusannya dapat dihitung menggunakan rumus $I(S_{1j}, S_{2j}, \dots, S_{mj}) = -\sum_{i=1}^m p_{ij} \log_2(p_{ij})$ dimana P_{ij} adalah nilai probabilitas dari sampel S yang ada di class C .

Sedangkan rumus gain dari atribut A , dapat dihitung dengan konsep matematis sebagai berikut: $Gain(A) = I(S_1, S_2, \dots, S_m) - E(A)$ atau dengan kata lain Gain adalah reduksi nilai dari entropy yang disebabkan oleh nilai dari atribut A .

Secara sederhana, algoritma C4.5 dapat dituliskan dalam beberapa langkah berikut:

1. Pilih atribut yang akan dijadikan node/titik simpul dalam struktur pohon. Caranya adalah dengan menentukan *gain* atau informasi yang ada di dalam atribut tersebut, ambil nilai tertinggi dari atribut-atribut yang ada.
2. Buat node cabang untuk setiap nilai yang dihitung
3. Bagi class atau kategori dalam setiap cabang
4. Ulangi proses untuk setiap cabang sampai semua kategori terhitung

Di bawah ini adalah contoh yang selalu di bahas dalam berbagai buku konsep tentang data mining, khususnya algoritma C4.5. Contoh pengambilan keputusan bermain badminton atau tidak adalah contoh yang paling baik yang dapat menggambarkan algoritma C4.5 secara sempurna.

Data yang dijadikan parameter untuk memutuskan bermain badminton atau tidak pada ruang terbuka, data mempertimbangkan 4 kriteria utama sebagai atribut yaitu

- outlook (keadaan cuaca),
- temperature (temperatur),
- humidity (kondisi kelembapan udara) dan
- windy (keadaan angin)

Tabel 1: Data Konsep C.45

no	outlook	temperature	humidity	windy	play
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Cloudy	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cold	Normal	False	Yes
6	Rainy	Cold	Normal	True	Yes
7	Cloudy	Cold	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cold	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Cloudy	Mild	High	True	Yes
13	Cloudy	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

Berdasarkan 14 data yang tersedia, terdapat dua class yang akan dianalisis yaitu C1 play badminton = yes dan C2 play badminton = no. C1 berjumlah 10 data dan C2 berjumlah 4 data. Untuk menghitung *information gain* dari kedua class tersebut, maka digunakan rumus sebagai berikut:

$$I(S_1, S_2) = I(10, 4) = \frac{-10}{14} \log_2 \frac{10}{14} - \frac{4}{14} \log_2 \frac{4}{14} = 0.863$$

Selanjutnya, perlu dilakukan perhitungan nilai entropy untuk setiap atribut; dimulai dari atribut outlook. Atribut outlook memiliki 3 nilai yaitu cloudy, rainy dan sunny

$$I(\text{cloudy} : \text{yes} = 4, \text{no} = 0) = I(4, 0) = \frac{-4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} = 0$$

$$I(\text{rainy} : \text{yes} = 4, \text{no} = 1) = I(4, 1) = \frac{-4}{5} \log_2 \frac{4}{5} - \frac{1}{5} \log_2 \frac{1}{5} = 0.722$$

$$I(\text{sunny} : \text{yes} = 2, \text{no} = 3) = I(2, 3) = \frac{-2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.970$$

Berdasarkan hasil hitungan di atas, maka nilai tersebut dapat disubstitusikan ke rumus entropy $E(A) = \sum_{j=1}^v \frac{S_{1j} + \dots + S_{mj}}{s} (S_{1j}, \dots, S_{mj})$ dan didapatkan hasilnya sebagai berikut

$$E(\text{outlook}) = \frac{4}{14} 0 + \frac{5}{14} 0.722 + \frac{5}{14} 0.970 = 0.604.$$

Berdasarkan nilai tersebut, maka dapat dilakukan perhitungan untuk nilai Gain dari atribut outlook yaitu

$$Gain(outlook) = I(S_1 - S_2) - E(outlook) = 0.863 - 0.604 = 0.258$$

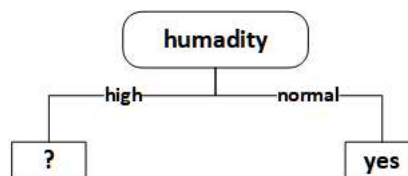
Dengan cara yang sama, maka dapat dilakukan perhitungan untuk atribut lain yaitu temperatur, humadity dan windy. Untuk memudahkan perhitungan data, maka di bawah ini disajikan perhitungan dalam bentuk tabel untuk keseluruhan atribut yang dimaksud.

Tabel 2: Hitungan Gain dan Entropy Iterasi 1

ATRIBUT	INFORMASI	JUMLAH DATA	NO	YES	NILAI I	GAIN
TOTAL DATA		14	4	10	0.863	
OUTLOOK	CLOUDY	4	0	4	0	0.258
	RAINY	5	1	4	0.721	
	SUNNY	5	3	2	0.970	
TEMPERATURE	COLD	4	0	4	0	0.183
	HOT	4	2	2	1	
	MILD	6	2	4	0.918	
HUMADITY	HIGH	7	4	3	0.985	0.370
	NORMAL	7	0	7	0	
WINDY	FALSE	8	2	6	0.811	0.006
	TRUE	6	2	4	0.918	

Berdasarkan hasil hitungan keseluruhan dalam tabel di atas, maka terlihat bahwa nilai Gain terbesar adalah atribut humadity bila dibandingkan dengan atribut lainnya. Nilai dari humadity memiliki informasi high dan normal.

Berdasarkan information gain, maka nilai output dari normal adalah yes (terdapat 7 data dengan output yes dan 0 data dengan output no) sehingga tidak diperlukan perhitungan lebih lanjut. Sedangkan informasi high perlu dilakukan perhitungan lebih lanjut pada iterasi kedua. Dalam algoritma C4.5, atribut ini dapat digambarkan dalam bentuk tree sebagai node awal.



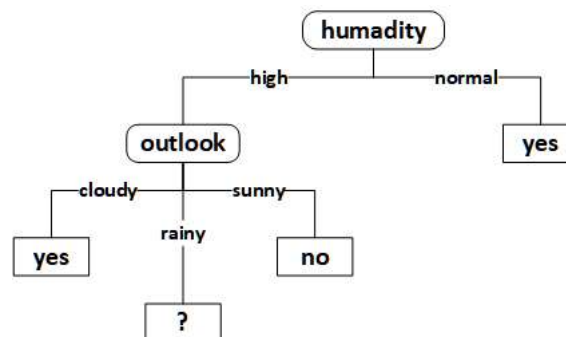
Gambar 1: Gambar Tree Iterasi 1

Untuk menghitung iterasi 2, maka data yang dianalisis adalah data dengan atribut humadity dengan informasi high saja. Hasil perhitungan di *information gain* dan entropy disajikan dalam bentuk tabel di bawah ini.

Tabel 3: Hitungan Gain dan Entropy Iterasi 2

ATRIBUT	INFORMASI	JUMLAH DATA	NO	YES	NILAI I	GAIN
HUMADITY HIGH		7	4	3	0.985	
OUTLOOK	CLOUDY	2	0	2	0	0.7
	RAINY	2	1	1	1	
	SUNNY	3	3	0	0	
TEMPERATURE	COOL	0	0	0	0	0.02
	HOT	3	2	1	0.918	
	MILD	4	2	2	1	
WINDY	FALSE	4	2	2	1	0.02
	TRUE	3	2	1	0.918	

Berdasarkan pada hasil perhitungan iterasi 2, maka dapat dilakukan penggambaran tree lebih lanjut dengan memperhatikan nilai gain tertinggi yaitu 0.7, keputusan yes untuk cloudy dan keputusan no untuk sunny. Penggambaran tree sebagai berikut:



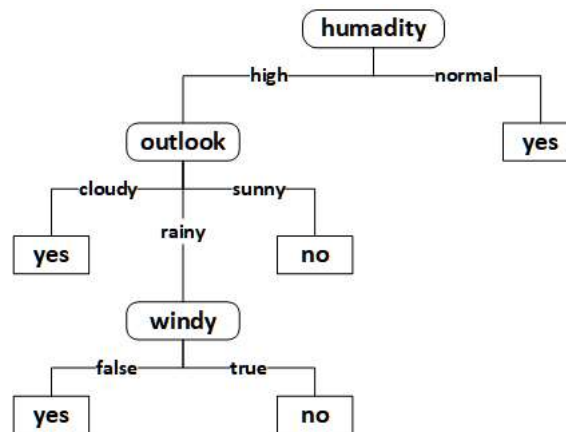
Gambar 2: Gambar Tree Iterasi 2

Proses iterasi 3, mengambil atribut humidity dengan informasi bernilai high dan outlook dengan informasi bernilai rainy. Tabel hasil hitungan untuk kedua atribut tersebut adalah sebagai berikut:

Tabel 4: Hitungan Gain dan Entropy Iterasi 3

ATRIBUT	INFORMASI	JUMLAH DATA	NO	YES	NILAI I	GAIN
HUMADITY HIGH, OUTLOOK RAINY		2	1	1	1	
TEMPERATURE	COOL	2	1	1	1	0
	HOT	2	1	1	1	
	MILD	2	1	1	1	
WINDY	FALSE	1	0	1	0	1
	TRUE	1	1	0	0	

Dengan demikian, proses penggambaran tree dapat dilakukan secara lengkap dengan berpedoman pada tabel hasil hitungan pada iterasi 3.



Gambar 3: Gambar Tree Iterasi 3

Dengan memperhatikan hasil akhir dari tree yang digambarkan di atas, maka semua informasi telah terproses sehingga dapat dijadikan pedoman dalam proses menentukan keputusan play badminton = yes atau play badminton = 0 dengan memperhatikan atribut dan informasi yang digunakan dalam tree tersebut.

BAB III

METODE PENELITIAN

Penelitian ini menggunakan standar metode data mining¹ yaitu:

1. **Data Source:** mencari data sumber untuk penerapan penelitian ini. Data yang digunakan adalah data dari konsep yang dijelaskan pada bab sebelumnya. Dan uji coba beberapa studi kasus sebagai data uji coba dalam implementasi algoritma C.45
2. **Data Cleaning:** sumber data yang digunakan dilakukan pembersihan data untuk membuang berbagai data yang tidak konsisten. Tujuan adalah mendapatkan data yang berkualitas yang merepresentasikan pola Data Mining.
3. **Data Integration:** oleh karena dalam penelitian ini data yang digunakan bisa bersumber dari database yang berbeda, maka proses integrasi menggunakan bahasa Structure Query Language (SQL) yang secara umum diterapkan pada berbagai sistem basis data.
4. **Data Selection;** data yang digunakan dalam penelitian ini bisa dalam bentuk data yang kompleks dengan banyak atribut. Perlu dilakukan seleksi terdapat atribut-atribut apa saja yang digunakan dalam proses analisis data mining.
5. **Data Transformation.** Data dari berbagai sumber, ditransformasikan ke dalam struktur tabel yang baku untuk diproses. Transformasi data menjadi pola baku penting karena prosedur yang digunakan diharapkan menjadi prosedur yang universal sehingga bisa diterapkan pada berbagai penyelesaian kasus data mining dengan algoritma C-45.
6. **Data Mining:** menerapkan algoritma C-45 ke dalam proses analisis data mining untuk menemukan pola data menarik.
7. **Pattern Evaluation.** Data dengan pola pohon keputusan dilakukan ujicoba sebagai bahan evaluasi terkait efektivitas implementasi prosedur perhitungan dan efisiensi waktu perhitungan algoritma C-45.
8. **Knowledge Presentation.** Representasi pengetahuan dan pola data mining ke dalam bentuk struktur keputusan (struktur pohon)

¹ David Olson – Dursun Delen, *Advanced Data Mining Techniques* (Springer: Verlag Berlin Heidelberg, 2008), 9

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Dataset yang digunakan dalam penelitian ini mengacu pada dataset yang dijelaskan pada bab sebelumnya pada bagian konsep. Data tersebut digunakan dalam implementasi program. Jika diimplementasikan secara sederhana dan mengikuti proses kerja algoritma, maka penyelesaian dari soal konsep pada bab sebelumnya bisa diselesaikan dengan langkah-langkah berikut:

1. Langkah Pertama adalah mempersiapkan data dan tabel untuk menampung semua nilai yang akan dianalisis.

```
drop database if exists c45;
create database c45;
use c45;

create table tblBadminton
(
nourut int primary key,
outlook varchar(10),
temperature varchar(10),
humadity varchar(10),
windy varchar(10),
play varchar(10)
);

LOAD DATA LOCAL INFILE 'dbC45.csv'
INTO TABLE tblBadminton
FIELDS TERMINATED BY ';'
ENCLOSED BY ''
IGNORE 1 LINES;

select * from tblBadminton;
```

2. Menghitung nilai entropy dan nilai nilai gain

```
create table tblHitung
(
keterangan varchar(20),
atribut varchar(20),
jumlahkasus int,
tidak int,
ya int,
entropy double,
gain double
);

select @jumkasus := count(*) from tblBadminton;
```

```

select @ya := count(*) from tblBadminton where play = 'Yes';
select @tidak := count(*) from tblBadminton where play =
'No';
select @entropy :=
(-(@tidak/@jumkasus)*log2(@tidak/@jumkasus))+(-(@ya/@jumkasu
s)*log2(@ya/@jumkasus));
select @jumkasus, @tidak, @ya, @entropy;

insert into tblHitung
(keterangan, jumlahkasus, tidak, ya, entropy)
values ('Total', @jumkasus, @tidak, @ya, @entropy);

insert into tblHitung (atribut, jumlahkasus, tidak, ya)
select distinct(A.outlook), count(*) as jumkasus,
(select count(*) from tblBadminton as b
where b.play = 'No' and b.outlook = A.outlook) as
'Tidak',
(select count(*) from tblBadminton as c
where c.play = 'Yes' and c.outlook = A.outlook) as
'Ya'
from tblBadminton As A
group by A.outlook;
update tblHitung set keterangan = 'Outlook' where keterangan
IS NULL;

```

3. Proses menghitung dilakukan per atribut mengikuti cara yang sama dengan langkah sebelumnya

```

insert into tblHitung (atribut, jumlahkasus, tidak, ya)
select distinct(A.temperature), count(*) as jumkasus,
(select count(*) from tblBadminton as b
where b.play = 'No' and b.temperature = A.temperature)
as 'Tidak',
(select count(*) from tblBadminton as c
where c.play = 'Yes' and c.temperature =
A.temperature) as 'Ya'
from tblBadminton As A
group by A.temperature;
update tblHitung set keterangan = 'Temperature' where
keterangan IS NULL;

insert into tblHitung (atribut, jumlahkasus, tidak, ya)
select distinct(A.humadity), count(*) as jumkasus,
(select count(*) from tblBadminton as b
where b.play = 'No' and b.humadity = A.humadity) as
'Tidak',
(select count(*) from tblBadminton as c
where c.play = 'Yes' and c.humadity = A.humadity) as
'Ya'
from tblBadminton As A
group by A.humadity;

```

```

update tblHitung set keterangan = 'Humadity' where
keterangan IS NULL;

insert into tblHitung (atribut, jumlahkasus, tidak, ya)
select distinct(A.windy), count(*) as jumkasus,
      (select count(*) from tblBadminton as b
       where b.play = 'No' and b.windy = A.windy) as 'Tidak',
      (select count(*) from tblBadminton as c
       where c.play = 'Yes' and c.windy = A.windy) as 'Ya'
from tblBadminton As A
group by A.windy;
update tblHitung set keterangan = 'Windy' where keterangan
IS NULL;

update tblHitung set atribut = ' ' where atribut is NULL;

update tblHitung set entropy =
(- (tidak/jumlahkasus)*log2(tidak/jumlahkasus)) + (- (ya/jumlahk
asus)*log2(ya/jumlahkasus));

update tblHitung set entropy = 0 where entropy is null;

drop temporary table if exists tblTampung;

create temporary table tblTampung
(
keterangan varchar(20),
gain double
);

insert into tblTampung (keterangan, gain)
select keterangan, @entropy - sum((jumlahkasus/@jumkasus) *
entropy) as HitungGain
from tblHitung
group by keterangan;

update tblHitung set gain =
(select tblTampung.gain
from tblTampung
where tblTampung.keterangan = tblHitung.keterangan);

select * from tblBadminton;

```

4. Menampilkan hasil akhir perhitungan algoritma

```

select ucase(keterangan) as KETERANGAN,
ucase(atribut) as ATRIBUT,
jumlahkasus as JUMLAH,
tidak as NO,
ya as YES,
round(entropy, 5) as ENTROPY,
round(gain, 5) as GAIN
from tblHitung;

```

Dalam proses menuliskan kode program seperti pembahasan di atas, maka seluruh proses dikerjakan secara sekuensial. Khususnya pada langkah kedua, jika terdapat banyak atribut yang harus dihitung, maka penulisan kode program menyesuaikan dengan jumlah atribut yang dihitung. Tentunya kode program semacam ini menjadi tidak efisien jika diterapkan pada kasus dengan jumlah atribut yang sangat banyak.

Kelemahan ini dapat diselesaikan dengan menggunakan generator untuk algoritma C.45. Ada beberapa syarat dalam proses mengembangkan generator algoritma C.45, yaitu:

1. Tentukan identitas tabel yang menjadi sumber data bagi algoritma dalam melakukan analisis
2. Atribut yang menjadi parameter perhitungan variabel yang menentukan besarnya nilai gain dan nilai entropy
3. Tentukan jumlah variabel hasil analisis algoritma
4. Lakukan proses generator dengan memanfaatkan proses rekursif sampai seluruh parameter selesai dihitung dan mendapatkan output hasil hitungan setiap iterasi.

Dalam proses implementasi, tentunya penelitian ini menggunakan software Database Management System yang sudah dilengkapi dengan fasilitas Stored Procedure. Dengan memanfaatkan teknik generator, maka proses dapat dikerjakan secara iteratif.

1. Menentukan proses pembacaan seluruh atribut dari tabel dengan nama tblData

```
SELECT COLUMN_NAME
      FROM INFORMATION_SCHEMA.COLUMNS
      WHERE TABLE_NAME = N'tblData'
      AND COLUMN_NAME NOT IN(select
distinct(tblHasil.atribut) from tblHasil);
```

2. Setiap proses iterasi, disimpan hasilnya berdasarkan proses iteratif yang terjadi. Oleh karena setiap proses harus membaca atribut, maka generator juga menyertakan pembacaan atribut yang dihitung.

```
SELECT COLUMN_NAME
      FROM INFORMATION_SCHEMA.COLUMNS
      WHERE TABLE_NAME = N'tblData'
      AND COLUMN_NAME NOT IN(select distinct(tblHasil.atribut)
from tblHasil);
```

```

select @iterasike = concat('tblIterasi', @loop);

select @SQL = 'CREATE TABLE dbo.' + QUOTENAME(@iterasike) +
              '(iterasi int,urut INT NOT NULL
IDENTITY PRIMARY KEY, kolom varchar(100), pilihan
varchar(100), ' +
              'jumdata int, diag1 int, diag2 int,
' +
              'entropy float, gain float)';

EXEC sp_executesql @SQL;

select @SQL = 'SET IDENTITY_INSERT dbo.' +
QUOTENAME(@iterasike) + ' ON'
EXEC sp_executesql @SQL;

```

3. Hasil iterasi perlu disimpan dan direkam dalam sebuah tabel sehingga memudahkan pemanggilan data tersimpan untuk dibuatkan visual gambar tree.

```

select @iterasike = concat('tblIterasi', @loop);

select @SQL = 'CREATE TABLE dbo.' + QUOTENAME(@iterasike) +
              '(iterasi int,urut INT NOT NULL IDENTITY
PRIMARY KEY, kolom varchar(100), pilihan varchar(100), ' +
              'jumdata int, HASIL1 int, HASIL2 int, ' +
              'entropy float, gain float)';
EXEC sp_executesql @SQL;

select @SQL = 'SET IDENTITY_INSERT dbo.' +
QUOTENAME(@iterasike) + ' ON'
EXEC sp_executesql @SQL;

```

4. Proses rekursif dapat dijalankan untuk membaca hasil perhitungan setiap iterasi atribut yang dihitung

```

open cAmbilKolom
fetch next from cAmbilKolom into @namakolom;

while @@fetch_status = 0
begin
  select @norec = 0;
  if @namakolom = 'pasien'
  begin
    select @totaldata = count(*) from tblData;
    select @jumdata= count(*) from tblData;
    select @jum1 = count(*) from tblData where
diagnosa =HASIL1;
    select @jum2 = count(*) from tblData where
diagnosa ='Stroke Hemorrhage Subarachnoid';
    if @jumdata=0
    begin

```

```

        set @ent=0;
    end
else
    begin
        set @ent= (-1 * (@jum1/@jumdata) *
log((@jum1/@jumdata), 2))
        +(-1 * (@jum2/@jumdata) *
log((@jum2/@jumdata), 2))
    end

    set @entropytotal = @ent;

    SELECT @SQL = 'INSERT INTO dbo.' +
QUOTENAME(@iterasike) +
        + '
(iterasi, kolom, pilihan, jumdata, diag1, diag2, entropy,
gain) ' +
        ' VALUES(' +
cast(@loop as varchar) + ', 'total' +
@namakolom + ''', ''-' + ', ' + cast(@jumdata as varchar)
+ ', ' + cast(@jum1 as varchar) + ', ' + cast(@jum2 as
varchar) + cast(@ent as varchar) + ', 0)';
EXEC sp_executesql @SQL;
    end
else
    if @namakolom <> HASIL and @namakolom <>
KETERANGAN
    begin
        select @SQL = 'INSERT INTO dbo.' +
QUOTENAME(@iterasike) +
        '(pilihan, jumdata, atribut1, atribut2) ' +
        'select distinct(A.' + @namakolom + '), ' +
        'count(A.' + @namakolom + ') as Jumlah, ' +
        '(' +
        'select count(*) ' +
        'from tblData ' +
        'where hasil = ''HASIL1'' ' +
        'and ' + @namakolom + ' = A.' + @namakolom + ' '
+ ') as diag1, ' + '(' + 'select count(*) ' + 'from tblData
' + 'where atribut = ''HASIL1'' ' + 'and ' + @namakolom + '
= A.' + @namakolom + ' ' + ') as HASIL2 ' + 'from tblData as
A ' + 'group by A.' + @namakolom + ''';
        EXEC sp_executesql @SQL;
        select @SQL = 'update dbo.' + QUOTENAME(@iterasike) +
        ' set iterasi = ' + cast(@loop as varchar) + ', kolom
= '' ' + @namakolom + '' where iterasi IS NULL';
        EXEC sp_executesql @SQL;

        select @SQL = 'declare cHitung cursor for ' +
        'selecturut, jumdata, atribut1, atribut2 from dbo.' +
QUOTENAME(@iterasike) + ' whereurut <> 1 and iterasi = ' +
cast(@loop as varchar);
        EXEC sp_executesql @SQL;

```

5. Nilai gain dan nilai entropy dapat dihitung setelah proses rekursif selesai dilakukan

```

open cHitung
select @gain=0;
fetch next from cHitung into
@nourut, @jumdata, @jum1, @jum2;
while @@fetch_status = 0
begin
    if (@jum1 = 0) or
    (@jum2 = 0)
        begin
            set
            @ent=0;
        end
    else
        begin
            set
            @ent= (-1 * (@jum1/@jumdata) * log((@jum1/@jumdata), 2))
            +(-1 *
            (@jum2/@jumdata) * log((@jum2/@jumdata), 2));
        end
    select @gain=@gain
    + ((@jumdata/@totaldata) * @ent)
    select @SQL =
    'update dbo.' + QUOTENAME(@iterasike) +
    ' set entropy = '
    + cast(@ent as varchar) +
    ' where urut = ' +
    cast(@nourut as varchar);
    EXEC sp_executesql
    @SQL;
    fetch next from
    cHitung into @nourut, @jumdata, @jum1, @jum2;
end
select
@gain=@entropytotal-@gain;
select @SQL = 'update dbo.' +
QUOTENAME(@iterasike) +
' set gain = ' +
cast(@gain as varchar) +
' where kolom = ''' +
(cast(@namakolom as varchar)) + ''''
EXEC sp_executesql @SQL;
select @namakolom;
close cHitung
deallocate cHitung
end

```



```
        fetch next from cAmbilKolom into @namakolom;
    end

    close cAmbilKolom
    deallocate cAmbilKolom
    --select * from tblHasil
    select @sql = 'insert into tblHasil(iterasi, atribut,
pilihan) ' +
        'select iterasi, kolom, pilihan from dbo.' + cast(@iterasike
as varchar) +
        ' where kolom <> ''totalpasien'' AND gain = (select
max(gain) from dbo.' +
        cast(@iterasike as varchar) + ')'
    EXEC sp_executesql @SQL;
end
```

Dengan memanfaatkan generator algoritma C.45, maka perlu dilakukan percobaan

Tabel 5: Uji Coba Data

UJI	STUDI KASUS	JUMLAH ATRIBUT	HASIL KEPUTUSAN	WAKTU	KETERANGAN
1	Play Badminton	4 atribut: 1. OUTLOOK 2. TEMPERATURE 3. HUMADITY 4. WINDY	2 keputusan: 1. Play YES 2. Play NO	0,2 detik	14 data, 3 iterasi
2	Kelulusan Mahasiswa	4 atribut: 1. Kognitif 2. Psikomotorik 3. Afektif 4. Nilai Pra Ujian	2 keputusan: 1. Lulus 2. Tidak lulus	0,5 detik	125 data, 5 iterasi
3	Penyakit Stroke	32 atribut 1. Usia 2. Jenis Kelamin 3. Tekanan Darah (mmHg) 4. Temperatur °C 5. Pernafasan 6. Denyut Nadi x/menit 7. Kesadaran 8. Nyeri Kepala 9. Afasia reseptif 10. pusing 11. Mual 12. Muntah 13. Lemas Bicara 14. Sulit bicara 15. Sesak Nafas 16. Penurunan 17. Kesadaran 18. Waktu Serangan 19. Paresis 20. Gangguan 21. Penglihatan 22. Gangguan Memori 23. Gerakkan kaki dan tangan 24. Kejang 25. Kaku pada area wajah	2 keputusan: 1. Stroke hemorrhage Intracerebral 2. Stroke Hemorrhage Subarachnoid	0,3 detik 0,7 detik 0,4 detik	100 data, 17 iterasi 200 data, 13 iterasi 300 data, 10 iterasi

26. Kaku pada leher
27. Riwayat Hipertensi
28. Riwayat DM
29. Riwayat Stroke
30. Riwayat Stroke Keluarga
31. Riwayat Hipertensi
32. Riwayat Merokok

Berdasarkan hasil uji coba di atas, maka dapat dilihat beberapa hal:

1. Hasil iterasi dari generator C.45 tidak dipengaruhi oleh jumlah data maupun atribut yang diproses. Hal ini lebih pada faktor kualitas data yang dianalisis.
2. Demikian juga kecepatan waktu proses dari generator C.45 tidak dipengaruhi oleh jumlah data maupun atribut data.

BAB V

PENUTUP

5.1 KESIMPULAN

Berdasarkan pembahasan bab 1 sampai dengan bab 4, maka dapat disimpulkan bahwa algoritma C.45 dapat dibuat generator algoritma untuk menghasilkan perhitungan detil dengan jumlah atribut yang besar. Proses generator memanfaatkan rekursif iteratif dari perhitungan nilai gain dan nilai entropy. Kedua nilai tersebut didasarkan pada atribut yang dihitung dan keputusan yang dihasilkan dari atribut yang menjadi faktor penentu hasil.

Dalam penelitian ini juga menyertakan waktu analisis dari generator algoritma C.45. Waktu analisis tidak menggambarkan secara konstan dan linear terkait jumlah data, atribut dan hasil analisis. Waktu dipengaruhi oleh kualitas data yang digunakan dalam uji coba, termasuk jumlah iterasi yang dihasilkan selalu berdasarkan atribut yang diolah.

5.2 SARAN

Dalam penelitian ini menggunakan 3 sample data yang dianalisis yaitu data konsep, data penentuan kelulusan mahasiswa berdasarkan 3 atribut yaitu aspek kognitif, psikomotorik dan afektif, dan data tentang keputusan penyakit store berdasarkan 31 atribut yang mempengaruhinya.

Perlu dilakukan uji coba untuk data lain sehingga dapat dilihat kecepatan analisis dari sisi waktu dan juga ketepatan dan akurasi hasil. Hasil yang akurat juga dapat disimulasikan langsung menggunakan aplikasi lain untuk pembuatan struktur pohon keputusan dari algoritma ini.