BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Menurut [6] penelitian tentang Decision Tree untuk diagnosis Diabetes Type II dengan algoritma *C4.5* menghasilkan akurasi 78, 176% dengan melakukan teknik *handling missing value*. Disini variabel yang digunakan ada 8 plus kelas yaitu Number of times pregnant (banyaknya kehamilan), Plasma glucose concentration a 2 hours in an oral glucose tolerance test (kadar glukosa dua jam setelah makan), Diastolic blood pressure (tekanan darah), Triceps skin fold thickness (ketebalan kulit), 2-Hour serum insulin (insulin), Body mass index (berat tubuh), Diabetes pedigree function (riwayat diabetes dalam keluarga), Age (umur), dan Class variable (positive diabetes (1) dan negative diabetes (0). Penanganan *missing value* yang hanya melakukan *deletion* pada atribut *preagnant.*

Menurut [9], Tujuan dari peneltian ini adalah untuk meprediksi dampak penyakit diabetes terhadap penyakit jantung. Penelitian ini menggunakan teknik *naïve bayes classifier* untuk menghasilkan prediksi optimal dengan menggunakan *minimum training set.* Dataset yang digunakan adalah data set klinik yang dikumpulkan dari *diabetec research institute* di Chennai dan record terdiri dari 500 pasien. Hasil rata-rata research menggunakan Weka adalah *precision* sebesar 71%, recall sebesar 74%, dan F-measure sebesar 71.2%. Dari hasil penelitian ini menunjukkan bahwa perbandingan dengan metode yang umum yang terdapat pada literature penelitian ini dapat dinyatakan bahwa atribut yang dianalisa tidak secara langsung sebagai indikator penyakit jantung. Atribut yang digunakan untuk pengujian sangat sedikit, maka perlu dicoba untuk menambah atribut untuk penelitian selanjutnya, penggunaan atribut dari diagnosa diabetes, dapat digunakan untuk melanjutkan type prediksi yang lain dari penyakit yang diangkat dari diabetes.

Menurut [10], Penelitian ini khusus membahas pentingnya penanganan *missing value.* Karena banyak penelitian yang menggunakan *preprocessing* dengan *handling missing value,* tetapi sedikit yang menguraikan beberapa teknik yang ada pada *handling missing value. Handling missing value* dibagi menjadi 2 yaitu *sequential*

methods atau lebih dikenal dengan *preprocessing,* sedangkan *parralel methods* bukan pada *preprocessing* melainkan diperoleh langsung dari dataset asli yaitu dimana *missing data* diperoleh selama proses *knowledge.*

Menurut [11], Penelitian ini fokus pada aspek *medical diagnosis* dengan mengumpulkan data diabetes, hepatitis, dan penyakit jantung dan mengembangkan *intelligent medical decision support system* untuk membantu dunia medis. Penelitian ini mengusulkan penggunaan *decision tree C4.5, ID3* dan *CART algorithm* untuk mengklasifikasi penyakit dan membandingkan keefektifan terhadap ketiga algoritma tersebut. *Decision support system* yang dapat mempelajari hubungan antara *symptoms,* pathologi dari penyakit, *family history* dan hasil tes akan berguna untuk tenaga medis dan rumah sakit. Adapun hasil akurasi dari ketiga *decision tree* tersebut dapat dilihat pada tabel di bawah ini.

Tabel 2.1 Tabel Prediksi Akurasi

No.	Nama algoritma	Akurasi %
1	*CART Algorithm*	83.2
2	*ID3 Algorithm*	64.8
3	*C4.5 Algorithm*	71.4

Dari tabel di atas bahwa *CART Algorithm* menunjukkan performa yang lebih baik untuk akurasi *Decision Support System* dibandingkan dengan *ID3 Algorithm* dan *C4.5 Algorithm.*Untuk meningkatkan *decision support system*, interaksi harus dipertimbangkan antara obat pada pasien yang berbeda.

Menurut [12], Penelitian ini fokus pada pengolahan data dengan *rapidminer* untuk *preprocessing* termasuk *attribute identification and selection, outlier removal, data normalization and numerical discretization, visual data analysis, hidden relationships discovery* dan prediksi diabetes. Data set yang digunakan dari *Pima Indian Diabetes Data (PIDD) Uci Machine Learning Repository.* Tidak dibandingkan dengan *data mining tools* yang lain, misalnya *Weka* dan *Tanagra.*

Menurut [13], Menganalisa mekanisme penanganan *missing value* dan cara-cara penanganannya (*treatment rules).* Klasifikasi dari metode penanganan *treatment meods)* pada *missing data,* pada umunya dibagi menjadi 3 yaitu:

a. *Case deletion:* menghapus *record* yang hilang pada atribut.

b. *Parameter estimation*: algoritma *Expectation-Maximization* digunakan untuk menentukan nilai maksimum sebagai prosedur untuk memperkirakan parameter pada *missing value.*
c. *Imputation techniques*: mengisi (*replace)* data yang hilang dengan memperhatikan keterkaitan dataset.

Belum dikomparasikan dengan teknik penanganan *missing value yang lain degan memperhatikan* keterkaitan dengan atribut yang lain.\

Menurut [7], Besarnya *missing data* dikelompokkan menjadi 3 kelompok yaitu: pertama, apabila data yang hilang kurang dari 1% dari jumlah dataset yang ada, maka *missing data* ini dikategorikan *considered trivial* (dianggap sepele) karena tidak akan membuat masalah pada proses *Knowledge Discovery in Database (KDD);* kedua apabila data yang hilang antara 1-5% dari total jumlah dataset, maka dikategorikan *manageable;* ketiga: apabila data yang hilang antara 5-15%, maka memerlukan metode khusus untuk menanganinya; keempat apabila data yang hilang lebih dari 15%, maka sangat besar dampaknya pada penafsiran atau interprestasi sekaligus pada hasil akurasinya. Belum diketahui teknik *handling missing value* yang tetap memperhatikan keterkaitan dengan atribut yang lain.

Dari beberapa penelitian terkait di atas bisa diringkas dengan tabel dibawah ini.

Tabel 2.2 Ringkasan Penelitian Terkait

No.	Penulis	Judul	Masalah	Metode
1	[6]	*Decision tree Discovery for the Diabetes of Type II Diabetes Information Sciences*, 303-307	Penanganan *missing value* hanya dilakukan membiarkan dan menghapus *record* pada atribut yang yang mempunyai nilai nol (*case deletion)*. Penanganan *missing value* bisa ditangani dengan metode yang lain misalnya *imputation* yang lain.	*C4.5*
2	[9]	*Diagnosis of Heart Disease for Diabeteic*	Atribut yang digunakan untuk pengujian sangat sedikit, maka perlu dicoba untuk menambah	Naïve Bayes

		Patient using Naïve Bayes Method	atribut untuk penelitian selanjutnya, penggunaan atribut dari diagnosa diabetes, dapat digunakan untuk melanjutkan type prediksi yang lain dari penyakit yang diangkat dari diabetes.	
3	[11]	*Decision Support System for Medical Diagnosis Using Data Mining.*	Untuk meningkatkan *decision support system*, interaksi harus dipertimbangkan antara obat pada pasien yang berbeda.	*CART* *ID3* *C4.5*
	[10]	*A Comparison of Several Approaches to Missing Attribute Values in Data Mining*	Perlu penanganan *missing value* dengan memperhatikan keterkaitan dengan atribut yang lain, penanganan *missing value* dengan *sequential methods* atau yang lazim disebut *preprocessing* akan menghasilkan akurasi yang lebih siknifikan dengan tidak menghilangkan informasi penting yang ada pada dataset.	*C4.5* *LERS*
4	[14]	*Diabetes Data Analysis and Prediction Model Discovery Using Rapidminer*	Untuk meningkatkan *decision support system*, interaksi harus dipertimbangkan antara obat pada pasien yang berbeda.	*ID3*
5	[15]	*A Review of Missing Data Treatment Methods*," pp.	Belum dikomparasikan dengan teknik penanganan *missing value yang lain degan memperhatikan* keterkaitan dengan atribut yang	MMI BII *C4.5*

		1–8, 2005	lain.	
6	[7]	*E. Acu, "The treatment of missing values and its effect in the classifier accuracy*," no. 1995, pp. 1–9	Belum diketahui teknik *handling missing value* yang tetap memperhatikan keterkaitan dengan atribut yang lain.	LDA KNN

Dari beberapa penelitian di atas bisa ditarik kesimpulan penting yaitu perlunya penanganan *missing attribute value* dengan *sequential methods* atau yang biasa disebut *preprocessing* karena pada *sequential methods* di dalamnya terdapat beberapa teknik penanganan *missing value,* sehingga teknik-teknik tersebut bisa dikomparasi dan diterapkankan pada algoritma *C4.5* dan *Naïve Bayes* sehingga pada akhirnya akan diperoleh akurasi yang lebih akurat dan signifikan karena selain mencerminkan kondisi yang sebenarnya juga tetap memperhitungkan keterkaitan dengan atribut yang lain dan tidak akan kehilangan informasi penting dari data set yang ada.

2.2 Landasan Teori

2.2.1 Missing Attribute Value

Missing value hal biasa yang terjadi pada dataset dan kalau tidak ditangani secara maksimal akan mengakibatkan kurang signifikan pada akurasi hasil pengujian dan bahkan akan mengakibatkan hilangnya informasi berharga dari dataset yang ada[6]. Klasifikasi dari metode penanganan atau *treatment methods* pada *missing data,* pada umunya dibagi menjadi 3 [15] yaitu:

a. *Case deletion:* menghapus *record* yang hilang pada atribut.
b. *Parameter estimation*: algoritma *Expectation-Maximization* digunakan untuk menentukan nilai maksimum sebagai prosedur untuk memperkirakan parameter pada *missing value.*
c. *Imputation techniques*: mengisi (*replace)* data yang hilang dengan nilai yang baru.

Beberapa alasan terjadinya *missing data* menurut [10] karena:

1. Nilai pada atribut tidak tercatat karena *irrelevant,* misalnya seorang dokter mampu menganalisa pasien tanpa tes medis atau seorang pemilik rumah diminta mengevaluasi AC sedangkan di rumah tersebut tidak ada peralatan untuk mengecek AC. Kasus seperti ini disebut "*do not care".*
2. Data lupa tidak dimasukkan ke tabel atau keliru terhapus.

Ada beberapa cara atau teknik *handling missing value,* diantaranya adalah *sequential methods* yang lazim disebut preprocessing [8], yaitu:

1. *Deleting Cases with Missing Attribute Value*, cara ini sebuah cara yang paling sederhana yaitu menghapus semua record yang mempunyai *missing value.*
2. *The Most Common Value of an Attribute,* cara ini mengisi atribut yang hilang atau *missing value* dengan data yang sering muncul atau *modus.*
3. *The Most Common Value of an Attribute Restricted to a Concept ,* cara ini mengisi atribut yang hilang atau *missing value* dengan data yang sering muncul atau *modus* tetapi khusus pada *instance* atau *record* yang mempunyai *class atau label* yang sama.
4. *Assigning All Possible Attribute Values to a Missing Attribute Value,* yaitu mengisi atribut yang ada nilai yang hilang dengan cara mengisi semua jenis *record* yang ada.
5. *Assigning All Possible Attribute Values Restricted to a Concept,* yaitu mengisi atribut yang ada nilai yang hilang dengan cara mengisi semua jenis *record* yang ada tetapi terbatas pada *class* atau *variabel* yang sama.
6. *Replacing Missing Attribute Values by the Attribute Mean,* ini untuk atribut *numeric* dengan cara mengisi nilai yang hilang dengan nilai rata-rata atau *means.*
7. *Replacing Missing Attribute Values by the Attribute Mean Restricted to a Concep,* ini untuk atribut *numeric* dengan cara mengisi nilai yang hilang dengan nilai rata-rata atau *means* tetapi rata-rata dari *class* yang sama.

Contoh penerapan teknik *sequential methods* untuk menangani *missing value.*

Tabel 2.3 Dataset dengan missing value (data kategorikal)

Case	*Attribute*			*Decision*
	Temperature	*Headache*	*Nausea*	*Flu*
1	High	**?**	No	Yes

2	Very_high	Yes	Yes	Yes
3	**?**	No	No	No
4	High	Yes	Yes	Yes
5	High	**?**	Yes	No
6	Normal	Yes	No	No
7	Normal	No	Yes	No
8	**?**	Yes	**?**	Yes

Contoh perhitungan teknik *handling missing attribute value* dengan *sequential methods,* diantaranya seperti tabel berikut ini:

Tabel 2.4 Deleting Cases with Missing Attribute Value

CASE	ATTRIBUTE			DECISION
	TEMPERATURE	HEADACHE	NAUSEA	FLU
1	very_high	Yes	yes	yes
2	high	Yes	yes	yes
3	normal	Yes	no	no
4	normal	No	yes	no

Tabel 2.5 The Most Common Value of an Attribute

CASE	ATTRIBUTE			DECISION
	TEMPERATURE	HEADACHE	NAUSEA	FLU
1	high	**Yes**	no	yes
2	very_high	Yes	yes	yes
3	**high**	No	no	no
4	high	Yes	yes	yes
5	high	**Yes**	yes	no
6	normal	Yes	no	no
7	normal	No	yes	no
8	**high**	Yes	**yes**	yes

Tabel 2.6 The Most Common Value of an Attribute Restricted to a Concept

CASE	ATTRIBUTE			DECISION
	TEMPERATURE	HEADACHE	NAUSEA	FLU
1	high	**Yes**	no	yes
2	very_high	Yes	yes	yes
3	**normal**	No	no	no
4	high	Yes	yes	yes

5	high	**No**	yes	no
6	normal	Yes	no	no
7	normal	No	yes	no
8	**high**	Yes	**yes**	yes

Tabel 2.7 Assigning All Possible Attribute Values to a Missing Attribute Value

CASE	ATTRIBUTE			DECISION
	TEMPERATURE	HEADACHE	NAUSEA	FLU
1^{i}	high	**Yes**	no	Yes
1^{ii}	high	**No**	no	Yes
2	very_high	Yes	yes	Yes
3^{i}	**high**	No	no	No
3^{ii}	**very_high**	No	no	No
3^{iii}	**normal**	No	no	No
4	high	Yes	yes	Yes
5^{i}	high	**Yes**	yes	No
5^{ii}	high	**No**	yes	No
6	normal	Yes	no	No
7	normal	No	yes	No
8^{i}	**high**	Yes	**yes**	Yes
8^{ii}	**high**	Yes	**No**	Yes
8^{iii}	**very_high**	Yes	**yes**	Yes
8^{iv}	**very_high**	Yes	**No**	Yes
8^{v}	**normal**	Yes	**yes**	Yes
8^{vi}	**normal**	Yes	**yes**	Yes

Tabel 2.8 Assigning All Possible Attribute Values Restricted to a Concept

CASE	ATTRIBUTE			DECISION
	TEMPERATURE	HEADACHE	NAUSEA	FLU
1	high	**Yes**	no	Yes
2	very_high	Yes	yes	yes
3^{i}	**normal**	No	no	no
3^{ii}	**high**	No	no	no
4	high	Yes	yes	yes
5^{i}	high	**Yes**	yes	no
5^{ii}	high	**No**	yes	no
6	normal	Yes	no	no
7	normal	No	yes	no

8^{i}	**high**	Yes	**yes**	yes
8^{ii}	**high**	Yes	**no**	yes
8^{iii}	**very_high**	Yes	**yes**	yes
8^{iv}	**very_high**	Yes	**no**	yes

Tabel 2.9 Replacing Missing Attribute Values by the Attribute Mean

CASE	ATTRIBUTE			DECISION
	TEMPERATURE	HEADACHE	NAUSEA	FLU
1	100.2	**Yes**	no	yes
2	102.6	Yes	yes	yes
3	**99.2**	No	no	no
4	99.6	Yes	yes	yes
5	99.8	**Yes**	yes	no
6	96.4	Yes	no	no
7	96.9	No	yes	no
8	**99.2**	*Yes*	**yes**	*yes*

Tabel 2.10 Replacing Missing Attribute Values by the Attribute Mean Restricted to a Concep

CASE	ATTRIBUTE			DECISION
	TEMPERATURE	HEADACHE	NAUSEA	FLU
1	100.2	**Yes**	no	yes
2	102.6	Yes	yes	yes
3	**97.6**	No	no	no
4	99.6	Yes	yes	yes
5	99.8	**No**	yes	no
6	96.4	Yes	no	no
7	96.9	No	yes	no
8	**100.8**	Yes	**yes**	yes

2.2.2 Algoritma C4.5

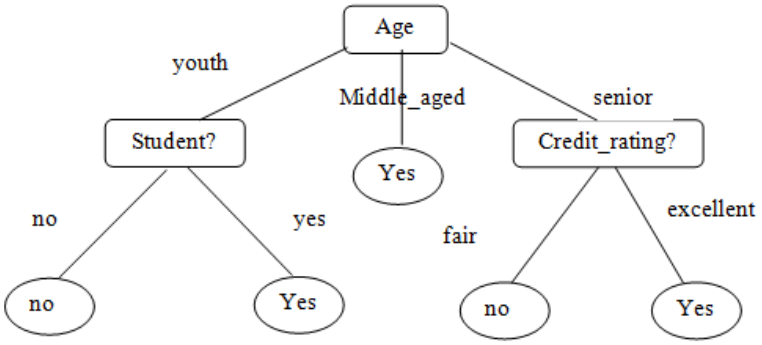
Algoritma *C4.5* adalah hasil dari pengembangan algoritma ID3 (*Iterative Dichotomiser*) yang dikembangkan oleh Quinlan [16]. Algoritma ini digunakan untuk membangun sebuah pohon keputusan yang mudah dimengerti, fleksibel, dan menarik karena dapat divisualisasikan dalam bentuk gambar [17]. Sebelumnya diakhir tahun 1970 sampai awal tahun 1980 J. Ross Quinlan, seorang peneliti dibidang *machine learning*, membuat sebuah algoritma *decision tree* yang dikenal dengan ID3 (*Iterative*

Dichotomiser). Kalau *ID3,* pengukuran seleksi atribut ditentukan oleh *Information Gain,* sedangkan *C4.5* pengukuran seleksi atribut ditentukan oleh *GainRatio* [16].

Algoritma *C4.5* atau pohon keputusan mirip sebuah pohon dimana terdapat node internal (bukan daun) yang mendeskripsikan atribut-atribut, setiap cabang menggambarkan hasil dari atribut yang diuji, dan setiap daun menggambarkan kelas. Pohon keputusan dengan mudah dapat dikonversi ke aturan klasifikasi. Secara umum keputusan pengklasifikasi pohon memiliki akurasi yang baik, namun keberhasilan penggunaan tergantung pada data yang akan diolah. *Decision tree* merupakan salah satu metode klasifikasi yang menggunakan representasi struktur pohon (*tree*) di mana setiap *node* merepresentasikan atribut, cabangnya merepresentasikan nilai dari atribut, dan daun merepresentasikan kelas. *Node* yang paling atas dari *decision tree* disebut sebagai *root*. Pada *decision tree* terdapat 3 jenis *node*, yaitu:

a. *Root Node*, merupakan *node* paling atas, pada *node* ini tidak ada *input* dan bisa tidak mempunyai *output* atau mempunyai *output* lebih dari satu.
b. *Internal Node* , merupakan *node* percabangan, pada *node* ini hanya terdapat satu *input* dan mempunyai *output* minimal dua.
c. *Leaf node atau terminal node* , merupakan *node* akhir, pada *node* ini hanya terdapat satu *input* dan tidak mempunyai *output*.

Pohon keputusan bekerja mulai dari akar paling atas, jika diberikan sejumlah data uji, misalnya X dimana kelas dari data X belum diketahui, maka pohon keputusan akan menelusuri mulai dari akar sampai node dan setiap nilai dari atribut sesuai data X diuji apakah sesuai dengan aturan pohon keputusan, kemudian pohon keputusan akan memprediksi kelas dari tupel X.



Gambar 2.1 Contoh konsep pohon keputusan untuk menentukan pembelian komputer berdasarkan atribut age, student dan credit rating[16]

Gambar 2.1 menggambarkan pohon keputusan untuk memprediksi apakah seseorang membeli komputer. Node internal disimbolkan dengan persegi, cabang disimbolkan dengan garis, dan daun disimbolkan dengan oval. Telah dijelaskan diatas bahwa *C4.5* adalah *successor* atau penerus dari *ID3,* oleh karena itu rumus *C4.5* untuk mengukur pemilihan atribut juga menyempurnakan dari *ID3.*

Langkah-langkah untuk membuat sebuah pohon keputusan algoritma *C4.5* yang merupakan pengembangan dari *ID3* tersebut, seperti di bawah ini [16]:

1. Mempersiapkan data training, data training yaitu data yang diambil dari data histori yang pernah terjadi sebelumnya atau disebut data masa lalu dan sudah dikelompokan dalam kelas-kelas tertentu.
2. Menentukan akar pohon. Akar pohon ditentukan dengan cara menghitung *GainRatio* tertinggi dari masing-masing atribut. Sebelum menghitung *GainRatio*, terlebih dahulu menghitung *Total Entropy* sebelum dicari masing-masing *Entropy class,* adapun rumus mencari Entropy seperti di bawah ini:

$$\boldsymbol{Entropy\ (S) = \sum_{i=1}^{n} -pi * log_{\,2}\ pi} \qquad (1)$$

Keterangan:

S = Himpunan kasus

n = jumlah partisi *S*

pi = proporsi dari S_i terhadap *S*

Dimana $log_2 pi$ dapat dihitung dengan cara:

$$log(X) = \frac{ln(X)}{ln(2)} \qquad (2)$$

3. Menghitung nilai *GainRatio* sebagai akar pohon, tetapi sebelumnya menghitung *Gain* dan *SplitEntropy (SplitInfo),* rumus untuk menghitung *Gain* seperti dibawah ini:

$$\boldsymbol{Gain(S,A) = Entropy\ (S) - \sum_{i+1}^{n} \frac{|\ S_i\ |}{|\ S\ |} * Entropy\ (S_i)} \qquad (3)$$

Rumus untuk menghitung *SplitEntropy, seperti di bawah ini:*

$$\boldsymbol{SplitEntropy_A(S) = -\sum_{i+1}^{n} \frac{|S_i|}{|S|} * log_2 \left(\frac{|S_i|}{|S|}\right)} \quad (4)$$

Rumus untuk menghitung *GainRatio,* dibawah ini:

$$\boldsymbol{GainRatio\ (A) = \frac{Gain\ (A)}{SplitEntropy\ (A)}} \quad (5)$$

Keterangan:

S = Himpunan Kasus

A = Atribut

n = jumlah partisi atribut *A*

$|S_i|$ = jumlah kasus pada partisi ke-*i*

$|S|$ = jumlah kasus dalam *S*

4. Ulangi langkah ke-2 dan ke-3 hingga semua tupel terpartisi
5. Proses partisi pohon keputusan akan berhenti disaat:
   a. Semua tupel dalam node *N* mendapatkan kelas yang sama
   b. Tidak ada atribut didalam tupel yang dipartisi lagi
   c. Tidak ada tupel didalam cabang yang kosong

2.2.3 Contoh Penerapan Agoritma C4.5

- Menentukan data training yang sudah ditentukan kelasnya.

Tabel 2.11 Pelanggan AllElektronik

No	Age	Income	Student	Credit Rating	Class_buys computer
1	youth	high	No	Fair	no
2	youth	high	No	Exellent	no
3	middle aged	high	No	Fair	yes
4	senior	medium	No	Fair	yes
5	senior	low	yes	Fair	yes
6	senior	low	yes	Exellent	no
7	middle aged	low	yes	Exellent	yes
8	youth	medium	No	Fair	no
9	youth	low	yes	Fair	yes
10	senior	medium	yes	Fair	yes

11	youth	medium	yes	Exellent	yes
12	middle aged	medium	No	Exellent	yes
13	middle aged	high	yes	Fair	yes
14	senior	medium	No	Exellent	no

- Menghitung akar pohon dengan cara menghitung *GainRatio,* sebelumnya menghitung Total *Entropy, Entropy Class, Gain, SplitEntropy*
  - Menghitung Total Entropy

    = (-9/14*(ln(9/14)/ln(2)))+(-5/14*(n(5/14)/ln(2))) = 0.94
  - Menghitung *Entropy Class*

    Entropy (age – youth)

    =(-2/5(ln(2/5)/ln(2)))+(-3/5*(ln(3/5)/ln(2))) = 0.25*

    Entropy (age –middle aged)

    = (-4/4(ln(4/4)/ln(2)))+(-0/4*(ln(0/4)/ln(2))) = 0*

    Entropy (age –middle aged)

    =(-3/5(ln(3/5)/ln(2)))+(-2/5*(ln(2/5)/ln(2))) = 0.97*

    Entropy (income –high)

    =(-2/4(ln(2/4)/ln(2)))+(-2/4*(ln(2/4)/ln(2))) = 1*

    Entropy (income –medium)

    =(-4/6(ln(4/6)/ln(2)))+(-2/6*(ln(2/6)/ln(2))) = 0.92*

    Entropy (income –low)

    =(-3/4(ln(3/4)/ln(2)))+(-1/4*lnN(1/4)/ln(2))) = 0.81*

    Entropy (student –yes)

    =(-6/7(LN(6/7)/LN(2)))+(-1/7*(LN(1/7)/LN(2))) = 0.59*

    Entropy (student –no)

    =(-3/7(LN(3/7)/LN(2)))+(-4/7*(LN(4/7)/LN(2))) = 0.99*

    Entropy (credit rating –fair)

    =(-6/8(LN(6/8)/LN(2)))+(-2/8*(LN(2/8)/LN(2))) = 0.81*

    Entropy (credit rating - excelent)

    =(-3/6(LN(3/6)/LN(2)))+(-3/6*(LN(3/6)/LN(2))) = 1*
    - Kunci pencarian entrophy
    - Jika diantara kolom “yes” atau “no” ada yang bernilai 0 (nol) maka entropy-nya di pastikan juga bernilai 0 (nol).

- Jika kolom “yes” dan “no” mempunyai nilai yang sama maka entropy-nya di pastikan juga bernilai 1 (satu).

- Menghitung *Gain*

  Gain (age)

  = 0.94 - ((5/14)* 0.97)+((4/14)*0)+ ((5/14)* 0.97) = 0.25

  Gain (income)

  = 0.94 - ((4/14)* 1)+((6/14)*0.92)+ ((4/14)* 0.81) = 0.03

  Gain (student)

  = 0.94 - ((7/14)* 0.59)+((7/14)*0.99) = 0.15

  Gain (credit rating)

  = 0.94 - ((8/14)* 0.81)+((6/14)*1) = 0.05

- Menghitung *SplitEntropy*

  SplitEntropy (age)

  =(-5/14*(LN(5/14)/LN(2)))+(-4/14*(LN(4/14)/LN(2)))+(-5/14*(LN(5/14)/LN(2))) = 1.58

  SplitEntropy (income)

  =(-4/14*(LN(4/14)/LN(2)))+(-6/14*(LN(6/14)/LN(2)))+(-4/14*(LN(4/14)/LN(2))) = 1.56

  SplitEntropy (student)

  =(-7/14*(LN(7/14)/LN(2)))+(-7/14*(LN(7/14)/LN(2))) = 1

  SplitEntropy (credit rating)

  =(-8/14*(LN(8/14)/LN(2)))+(-6/14*(LN(6/14)/LN(2))) = 0.99

- Menghitung *GainRatio*

  GainRatio (age)

  = 0.25 / 1.58 = 0.16

  GainRatio (income)

  = 0.03 / 1.56 = 0.02

  GainRatio (student)

  = 0.15 / 1 = 0.15

Ringkasan tabel di bawah ini merupakan hasil perhitungan di atas.

Tabel 2.12 Hasil perhitungan Node 1

NODE			Jumlah	yes	no	Entropy	Gain	SplitEntropy	Gain Ratio
1	Total		14	9	5	0.94			
	age						0.25	1.58	**0.16**
		youth	5	2	3	0.97			
		middle aged	4	4	0	0			
		senior	5	3	2	0.97			
	income						0.03	1.56	0.02
		high	4	2	2	1			
		medium	6	4	2	0.92			
		low	4	3	1	0.81			
	student						0.15	1	0.15
		yes	7	6	1	0.59			
		no	7	3	4	0.99			
	Credit rating						0.05	0.99	0.05
		fair	8	6	2	0.81			
		excellent	6	3	3	1			

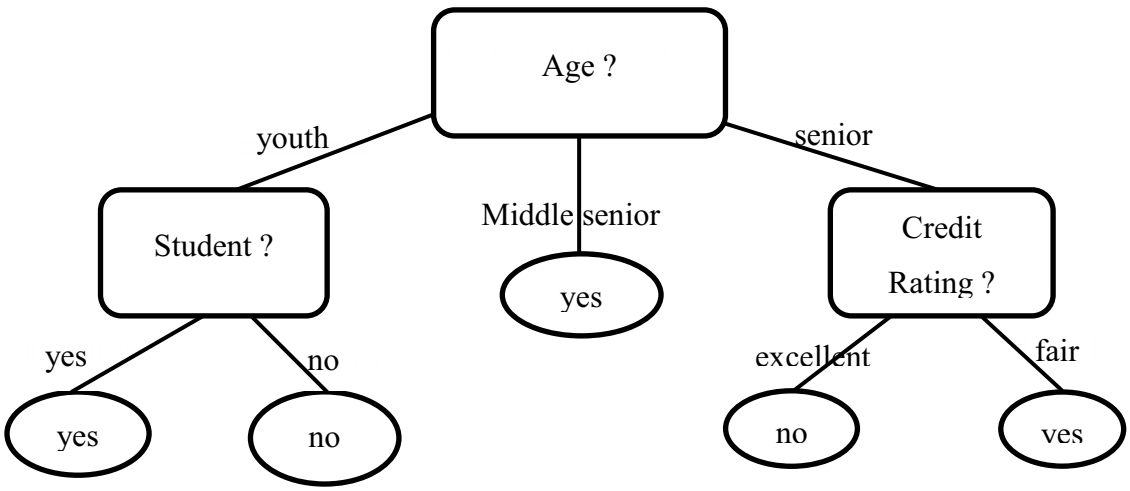
Tabel 2.13 Hasil perhitungan Node 1.1

NODE			jumlah	yes	no	Entropy	Gain	SplitInfo	Gain Ratio
1.1	total								
	Age - youth		5	2	3	0.971			
	income						0.57	1.52	0.38
		high	2	0	2	0			
		medium	2	1	1	1			
		low	1	1	0	0			
	student						0.97	0.97	1
		yes	2	2	0	0			
		no	3	0	3	0			
	credit_ rating						0.02	0.97	0.02
		fair	3	1	2	0.92			
		excellent	2	1	1	1			

Tabel 2.14 Hasil perhitungan Node 1.2

NODE 1.2			jumlah	yes	no	Entropy	Gain	SplitInfo	Gain Ratio
1	total								
	Age - senior		5	3	2	0.97			
	income						0.02	0.97	0.02
		medium	3	2	1	0.92			
		low	2	1	1	1			
	student						0.02	0.97	0.02
		yes	3	2	1	0.92			
		no	2	1	1	1			
	credit_ rating						0.97	0.97	0.97
		fair	3	3	0	0			
		excellent	2	0	2	0			

Dari perhitungan di atas diketahui bahwa nilai *GainRatio (age)* merupakan *GainRatio* tertinggi, oleh karena itu atribut *age* menjadi akar pohon, dan akan dihasilkan pohon keputusan seperti dibawah ini.



Gambar 2.2 Pohon keputusan [16]

2.2.4 *Algoritma* Naïve Bayes

Naïve Bayes merupakan pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class [16]. *Bayes* memiliki akurasi dan kecepatan yang sangat tinggi saat diaplikasi ke dalam database dengan data yang besar. *Naive Bayes* merupakan algoritma yang dapat meminimalkan tingkat dibandingkan dengan semua pengklasifikasi lainnya. Namun, dalam praktek ini tidak selalu terjadi, karena untuk ketidak akuratan dalam asumsi yang dibuat untuk penggunaannya class yang tidak utuh dan kurangnya data probabilitas yang tersedia. Pengklasifikasi Bayesian juga berguna dalam pembenaran teoritis untuk pengklasifikasi lain yang tidak secara eksplisit menggunakan teorema [16].

Teorema *Bayes* adalah suatu pendekatan pada ketidaktentuan yang diukur dengan probabilitas. Teorema Bayes memiliki rumusan umum sebagai berikut:

$$P(X|H) = \frac{P(H|X)P(X)}{P(H)} \tag{6}$$

Naive Bayes adalah penyederhanaan dari *teorema bayes*. Berikut rumus *Naive Bayes* menurut [16]:

$$P(X|H) = P(H|X)P(X) \tag{7}$$

Keterangan:

X : data dengan class yang belum diketahui

H : hipotesis data *X*, merupakan suatu class spesifik

P(H|X) : probabilitas hipotesis *H* berdasarkan kondisi *X* (*posteriori probability*)

P(H) : probabilitas hipotesis *H* (*prior probability*)

P(X|H) : probabilitas *X* berdasarkan kondisi pada hipotesis *H*

P(X) : probabilitas dari *X*

Beberapa keuntungan dari algoritma klasifikasi *Naive Bayes* [17], adalah:

1. Kuat terhadap pengisolasi gangguan pada data
2. Jika terjadi kasus missing value ketika proses komputasi sedang berlangsung, maka objek tersebut akan diabaikan

3. Dapat digunakan untuk data yang tidak relevan

Beberapa kelemahan dari algoritma klasifikasi *Naive Bayes* [18]:

1. Harus mengasumsi bahwa antar fitur tidak terkait (*independent*) Dalam realita, keterkaitan itu ada.
2. Keterkaitan tersebut tidak dapat dimodelkan oleh Naïve Bayesian Classifier.

2.2.5 Contoh Penerapan Algoritma Naïve Bayes

Contoh penerapan algoritma *Naïve Bayes,* data set diambil dari tabel 2.11 dengan data testing di bawah ini, dimana data testing di bawah ini belum diketahui kelasnya.

Tabel 2.15 Contoh Data testing untuk menentukan class pada *Naive Bayes*

No	Age	Income	Student	Credit Rating	Class_buys computer
1	youth	medium	yes	fair	?

JAWAB

$$P(X|H) = P(H|X)P(X)$$

- Tahap 1 menghitung *prior probability* dari setiap kelas/label

  $P(buys\ computer = yes) = 9/14 = 0.643$

  $P(buys\ computer = no) = 5/14 = 0.357$

- Tahap 2 menghitung probabilitas hipotesis (*posteriori probability*)

  $P(age = youth \mid buys\ computer = yes) = 2/9 = 0.222$

  $P(age = youth \mid buys\ computer = no) = 3/5 = 0.600$

  $P(income = medium \mid buys\ computer = yes) = 4/9 = 0.444$

  $P(income = medium \mid buys\ computer = no) = 2/5 = 0.400$

  $P(student = yes \mid buys\ computer = yes) = 6/9 = 0.667$

  $P(student = yes \mid buys\ computer = no) = 1/5 = 0.200$

  $P(credit\ rating = fair \mid buys\ computer = yes) = 6/9 = 0.667$

  $P(credit\ rating = fair \mid buys\ computer = no) = 2/5 = 0.400$

- Dari *probability* di atas, diperoleh:

  $P(H \mid buys\ computer = yes) = P(age = youth \mid buys\ computer = yes) *$

  $P(income = medium \mid buys\ computer = yes) *$

  $P(student = yes \mid buys\ computer = yes) *$

$P(credit\ rating = fair \mid buys\ computer = yes)$
$= 0.222 * 0.444 * 0.667 * 0.667 = 0.044$

$P(H \mid buys\ computer = no) = P(age = youth \mid buys\ computer = no) *$
$P(income = medium \mid buys\ computer = no) *$
$P(student = yes \mid buys\ computer = no) *$
$P(credit\ rating = fair \mid buys\ computer = yes)$
$= 0.600 * 0.400 * 0.200 * 0.400 = 0.019$

- Menentukan kelas dapat dihitung:

  $P(H \mid buys\ computer = yes) * P(buys\ computer = yes) = 0.044 * 0.643 = 0.028$

  $P(H \mid buys\ computer = no) * P(buys\ computer = no) = 0.019 * 0.357 = 0.007$

Sehingga diketahui untuk kelas/label *buys computer = yes* karena *yes* lebih besar daripada *no.*

Dari uraian di atas dapat dibuat ringkasan antara kelebihan dan kekurangan antara *C4.5* dengan Naïve Bayes. Adapun kelebihan dari dan kelemahannya, seperti tabel berikut:

Tabel 2.16 Kelebihan dan kelemahan algoritma *C4.5* dan *Naive Bayes*

NO.	METODE	KELEBIHAN	KELEMAHAN
1	*C4.5* [19]	1. Daerah pengambilan keputusan yang sebelumnya kompleks dan sangat global, dapat diubah menjadi lebih simpel dan spesifik. 2. Eliminasi perhitungan-perhitungan yang tidak diperlukan, karena ketika menggunakan metode pohon keputusan maka sample diuji hanya berdasarkan kriteria atau kelas	1. Terjadi overlap terutama ketika kelas-kelas dan kriteria yang digunakan jumlahnya sangat banyak. Hal tersebut juga dapat menyebabkan meningkatnya waktu pengambilan keputusan dan jumlah memori yang diperlukan. 2. Pengakumulasian

		tertentu. 3. Fleksibel untuk memilih fitur dari internal node yang berbeda, fitur yang terpilih akan membedakan suatu kriteria dibandingkan kriteria yang lain dalam node yang sama. Kefleksibelan metode pohon keputusan ini meningkatkan kualitas keputusan yang dihasilkan jika dibandingkan ketika menggunakan metode penghitungan satu tahap yang lebih konvensional.	jumlah eror dari setiap tingkat dalam sebuah pohon keputusan yang besar. 3. Kesulitan dalam mendesain pohon keputusan yang optimal.
2	*NAÏVE BAYES* [17] [18]	1. Kuat terhadap pengisolasi gangguan pada data 2. Jika terjadi kasus missing value ketika proses komputasi sedang berlangsung, maka objek tersebut akan diabaikan. 3. Dapat digunakan untuk data yang tidak relevan.	1. Harus mengasumsi bahwa antar fitur tidak terkait (*independent*) Dalam realita, keterkaitan itu ada. 2. Keterkaitan tersebut tidak dapat dimodelkan oleh Naïve Bayesian Classifier. 3. Tidak membutuhkan

			skema estimasi parameter perulangan yang rumit, ini berarti bisa diaplikasikan untuk *data set* berukuran besar [20]

Dari tabel diatas, setelah menganalisa kelebihan dan kekurangan dari kedua algoritma tersebut, maka yang sesuai dengan dataset peneliti yaitu dari *Pima Indian Diabetes Data (PIDD)* dengan menerapkan *sequential methods* untuk penanganan *missing value*-nya maka teknik *Assigning All Possible Attribute Values to a Missing Attribute Value* dan *Assigning All Possible Attribute Values Restricted to a Concept* yang diimplementasikan pada algoritma *Naïve Bayes* menghasilkan akurasi yang lebih baik.

2.2.6 Evaluasi Algoritma Klasifikasi Data Mining

1. Evaluasi *Confusion Matrix*

Untuk melakukan evaluasi terhadap model klasifikasi berdasarkan perhitungan objek testing mana yang diprediksi benar dan tidak benar. *Confusion Matrix* berisi informasi tentang aktual (*actual*) dan prediksi (*predicted*) pada sistem klasifikasi. Kinerja sistem seperti ini biasanya dievaluasi dengan menggunakan data pada matriks. Perhitungan ini ditabulasikan kedalam tabel yang disebut *confusion matrix,* [17]. Bentuk *confusion matrix* dapat dilihat pada Tabel berikut ini:

Tabel 2.17 Confusion matrix[17]

CLASSIFICATION		PREDICTED CLASS	
		CLASS:YES	CLASS:NO
OBSERVED CLASS	CLASS:YES	a (True Positive-TP)	b (False Negative-FN)
	CLASS:NO	c (False Possitive-FP)	d (True negative-TN)

Pada Tabel 2.17 untuk *True positive* merupakan tupel positif didata set yang diklasifikasikan positif, *True negatives* merupakan tupel negatif di data set yang diklasifikasikan negatif. *False positives* adalah tupel positif di data set yang diklasifikasikan negatif *False negatives* merupakan jumlah tupel negatif yang diklasfikasikan positif.

Setelah dilakukan *confusion matrix* berikutnya akan dihitung *accuracy*, *sensitivity*, *specificity, PPV, NPV*. *Sensitivity* digunakan untuk membandingkan jumlah *true positives* terhadap jumlah tupel yang *positives* sedangkan *specificity* adalah perbandingan jumlah *true negatives* terhadap jumlah tupel yang *negatives.* Sedangkan untuk PPV(*Positives Predictive Value* atau nilai prediktif positif) adalah proporsi kasus dengan hasil diagnosa positif, NPV(*Negatives Predictive Value* atau nilai prediktif negatif) adalah proporsi kasus dengan hasil diagnosa negatif. Berikut perhitungannya:

a. Keakuratan (*Accuracy)* adalah proporsi jumlah prediksi yang benar. Hal ini ditentukan dengan menggunakan rumus *accuracy berikut:* :

$$Accuracy = \frac{a+b}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

b. *Sensitivity* juga dapat dikatakan *true positive rate* (TP *rate*) atau *recall*. Sebuah *sensitivity* 100% berarti bahwa pengklasifikasian mengakui sebuah kasus yang diamati positif. Misalnya semua orang memiliki kanker ganas diakui sebagai sakit.

$$Sensitivity = \frac{number\ of\ True\ Possitive}{number\ of\ True\ Possitive + number\ of\ False\ Negative} \quad (9)$$

c. *Specificity* adalah mengukur atau mengamati kasus bahwa yang diamati teridentifasi negatif.

$$Specificity = \frac{number\ of\ true\ negative}{number\ of\ true\ negative + number\ of\ false\ negative} \quad (10)$$

d. *PPV (*nilai prediktif positif) adalah tingkat positif salah (FP) adalah proporsi kasus negatif yang salah diklasifikasikan sebagai positif, yang dihitung dengan menggunakan persamaan:

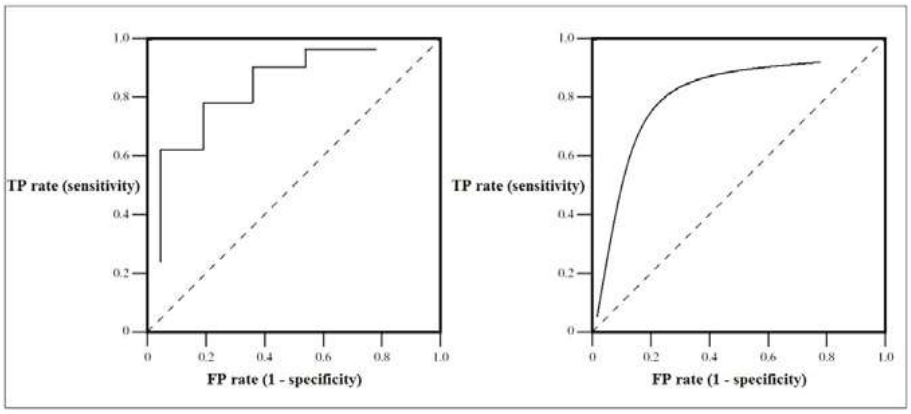
$$PPV = \frac{number\ of\ true\ possitive}{number\ of\ true\ possitive + number\ of\ false\ possitive} \quad (11)$$

e. *NPV (*Nilai prediktif Negatif) adalah tingkat negatif sejati (TN) didefinisikan sebagai proporsi kasus negatif yang diklasifikasikan dengan benar, yang dihitung dengan menggunakan persamaan:

$$NPV = \frac{number\ of\ true\ negative}{number\ of\ true\ negative + number\ of\ false\ negative} \quad (12)$$

2. Evaluasi *ROC Curve*

Kurva *ROC* (*Receiver Operating Characteristic*) adalah ilustrasi grafis dari kemampuan diskriminan dan biasanya diterapkan untuk masalah klasifikasi biner [17]. Secara teknik, kurva ROC juga disebut grafik ROC, dua dimensi grafik yaitu TP rate diletakan pada sumbu Y, sedangkan FP rate diletakan pada sumbu X. Grafik ROC menggambarkan trade-off antara manfaat (‘*true positives’*) dan biaya (‘*false positives*’). Berikut tampilan dua jenis kurva ROC (*discrete* dan *continuous*).



Gambar 2.3 Grafik ROC (discrete dan continuous)[17]

Dari Gambar II.2 ada beberapa yang penting untuk dicatat. Titik kiri bawah (0,0) yaitu diantara nilai TP dan FP, titik (1,1) merupakan klasifikasi positif. Titik (0,1) merupakan klasifikasi sempurna (yaitu tidak ada FN dan tidak ada FP). yang benar-benar acak akan memberikan titik sepanjang garis diagonal dari kiri bawah ke sudut kanan atas. Garis ini membagi ruang ROC sebagai berikut:

(a) poin diatas garis diagonal merupakan hasil klasifikasi yang baik

(b) poin dibawah garis diagonal merupakan hasil klasifikasi yang buruk dapat disimpulkan satu poin untuk ROC adalah lebih baik dari pada yang lainnya jika arah garis melintang dari kiri bawah ke kanan atas didalam grafik tersebut.

Dapat disimpulkan bahwa, satu point pada kurva ROC adalah lebih baik dari pada yang lainnya jika arah garis melintang dari kiri bawah ke kanan atas didalam grafik. Tingkat *AUC* dapat di diagnosa sebagai berikut [17]:

- 0.90 – 1.00 = *Excellent classification* (paling baik)
- 0.80 – 0.90 = *Good classification* (Baik)
- 0.70 – 0.80 = *Fair classification* (Adil atau sama)
- 0.60 – 0.70 = *Poor classification* (Rendah)
- 0.50 – 0.60 = *Failure* (Gagal)

2.3 Penerapan sequential methods untuk handling missing value pada algoritma *C4.5* dan Naïve Bayes untuk memprediksi penyakit *Diabetes Mellitus*

Diabetes melitus merupakan suatu kelompok penyakit metabolik dengan karakteristik hiperglikemia yang terjadi karena kelainan sekresi insulin, kerja insulin atau kedua-duanya [3]. Hiperglikemia kronik pada diabetes berhubungan dengan kerusakan jangka panjang, disfungsi dan kegagalan beberapa organ tubuh, terutama mata, ginjal, syaraf, jantung dan pembuluh darah. Selain itu diabetes melitus juga merupakan suatu penyakit yang tidak dapat dituangkan dalam suatu jawaban yang jelas dan singkat tetapi secara umum dapat dikatakan sebagai suatu kumpulan problema anatomik dan kimiawi yang merupakan akibat dari sejumlah faktor dimana didapat defisiensi insulin absolut atau relatif dan gangguan fungsdi insulin [3]. Penyebab meningkatnya jumlah penderita diabetes di Indonesia ditentukan oleh jumlah penduduk yang meningkat, penduduk berumur diatas 40 tahun meningkat, urbanisasi, pendapatan perkapita yang tingi, restoran cepat saji, hidup santai dan berkurangnya penyakit infeksi dan kurang gizi.

Gejala diabetes ditandai dengan rasa haus yang berlebihan, sering kencing terutama malam hari, banyak makan serta berat badan yang turun dengan cepat. Disamping itu kadang-kadang ada keluhan lemah, kesemutan pada jari tangan dan kaki, cepat lapar, gatal-gatal, penglihatan jadi kabur, gairah seks menurun, luka sukar untuk sembuh dan pada ibu-ibu sering melahirkan bayi diatas empat kilogram. Berbagai faktor genetik, lingkungan dan cara hidup berperan dalam perjalanan penyakit diabetes. Ada kecenderungan penyakit ini timbul dalam keluarga. Di samping

itu juga ditemukan perbedaan kekerapan dan komplikasi diantara ras, negara dan kebudayaan.

Secara garis besar diabetes melitus dibagi menjadi tiga tipe sebagai berikut:

a. Diabetes melitus tipe 1

Biasanya diabetes melitus tipe satu diderita oleh orang-orang dinegara subtropik dan kekerapan tertingggi ditemukan di Eropa Utara. Gambaran klinisnya biasanya timbul pada masa kanak-kanak dan puncaknya pada masa akil balig, tetapi ada juga yang timbul pada masa dewasa.

b. Diabetes melitus tipe 2

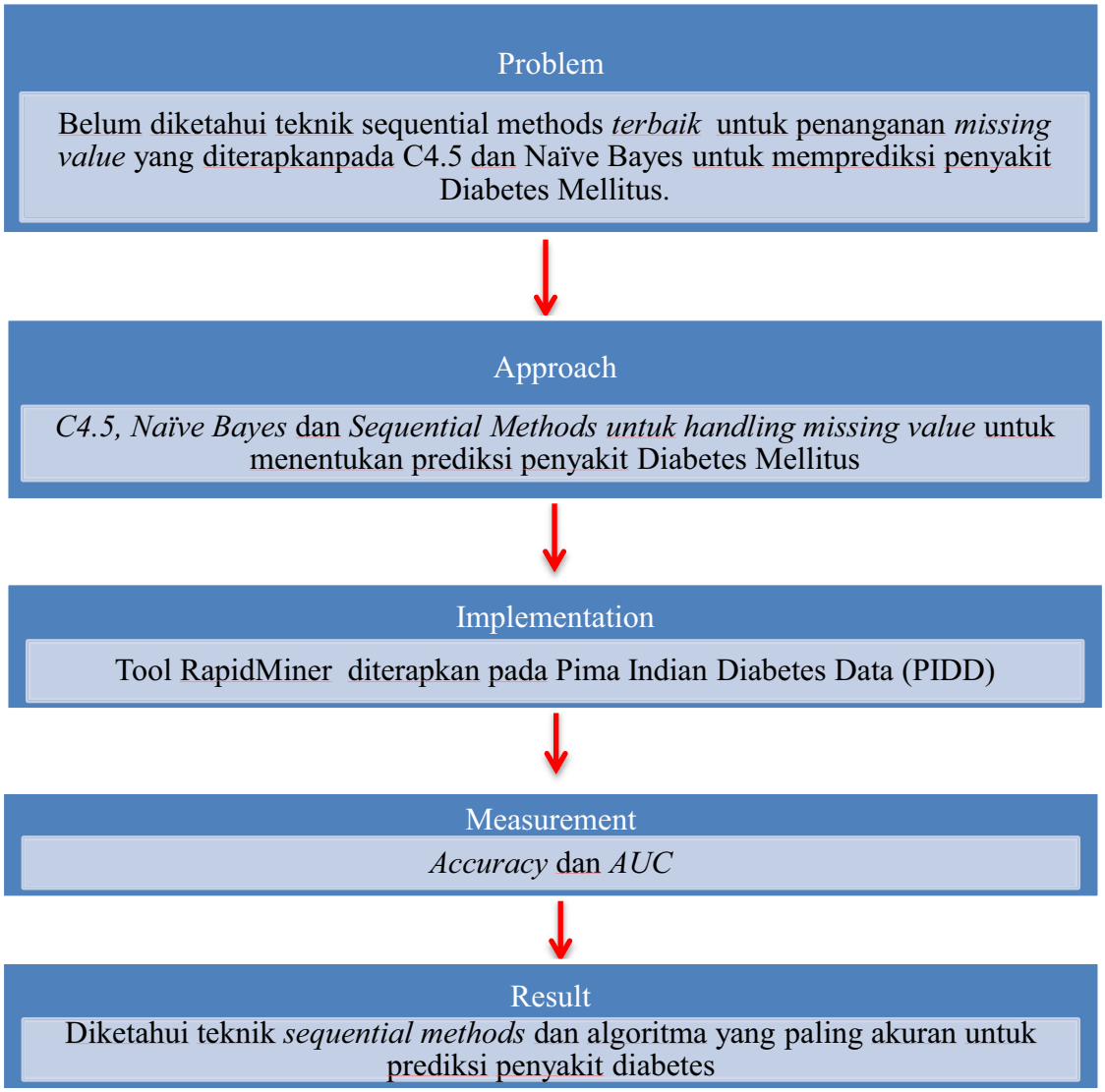
Diabetes melitus tipe dua adalah jenis yang paling banyak ditemukan (lebih dari 90%) dan timbulnya semakin sering ditemukan setelah umur 40 tahun. Pada keadaan kadar glukosa darah tidak terlalu tinggi atau belum ada komplikasi, biasanya pasien tidak berobat ke rumah sakit atau dokter. Ada juga yang sudah didiagnosis sebagai diabetes tetapi karena kekurangan biaya biasanya pasien tidak berobat lagi. Hal ini menyebabkan jumlah pasien diabetes yang tidak terdiagnosis lebih banyak daripada yang terdiagnosis.

c. Diabetes melitus gestasional

Diabetes melitus gestasional adalah diabetes yang timbul selama kehamilan. Ini meliputi 2-5% dari seluruh diabetes. Jenis ini sangat penting diketahui karena dampak pada janin kurang baik bila tidak ditangani dengan benar. Dataset penyakit diabetes tersebut di atas diambil dari *Pima Indian Diabetes Data (PIDD)* dari *UCI Machine Learning Repository.* Dari dataset tersebut banyak atribut yang mempunyai *missing data.* Oleh karena itu diperlukan penanganan terhadap data yang hilang tersebut. *Salah* satu teknik *handling missing value* yang dilakukan pada *preprocessing data* adalah *sequential methods.* dari metode ini beberapa cara dilakukan baik *deletion cases* maupun *imputation.*

Dari penerapan *sequential methods* untuk *handling missing* value pada algoritma *C4.5* dan *Naïve Bayes* untuk memprediksi penyakit *Diabetes Mellitus*, dilakukan pendalaman teori dari dataset yang mempunyai banyak *missing data* tersebut.

Adapun skema kerangka pemikiran dari penerapan di atas dapat ditampilkan seperti pada diagram di bawah ini.



Gambar 2.4 Kearangka Pemikiran

Dari diagram di atas akan dijelaskan seperti di bawah ini:

1. *Problem.*

   Dari kumpulan beberapa literatur tentang diabetes dan *handling missing value* serta menganalisa dataset yang ada, maka diperoleh permasalahan yaitu belum diketahui teknik *sequential methods* yang paling akurat untuk penanganan *missing value* dan algoritma yang akurat untuk memprediksi penyakit *Diabetes Mellitus*.

2. *Approach*

   Pendekatan ini dipilih berdasarkan studi literature tentang diabetes, *C4.5, Naïve Bayes* dan *handling missing value* untuk menentukan prediksi penyakit *Diabetes Mellitus* dengan menerapkan *sequential methods* untuk menangani *missing data.*

3. *Implementation*

Tool RapidMiner diterapkan pada data *Pima Indian Diabetes Data (PIDD)* yang sudah dilakukan *handling missing value* dengan beberapa teknik yang ada pada *sequential methods*

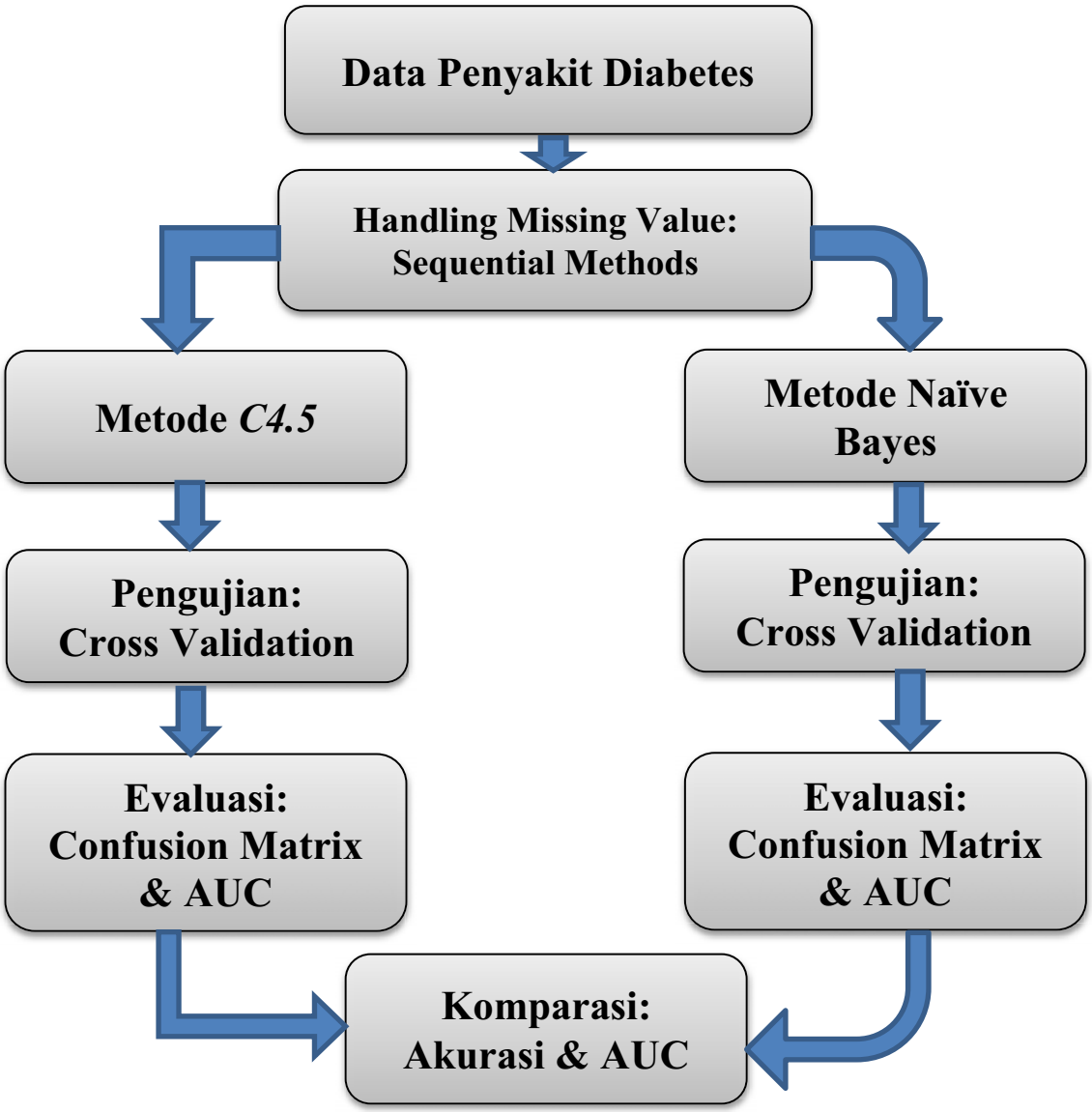
4. *Measurement*

Untuk mengetahui teknik yang paling akurat pada *sequential methods* yang diterapkan pada algoritma *C4.5* dan *Naïve Bayes,* maka dilakukan evaluasi dengan *cross validation,* dimana dari evaluasi *cross validation ini* akan menghasilkan *Confusion Matrix* yang berisi *accuracy* dan *AUC* yang merupakan *ROC Curve.*

5. Result

Dari tahapan teori di atas pada akhirnya akan diketahui teknik *sequential methods* dan algoritma klasifikasi yang paling akurat untuk memprediksi penyakit *Diabetes Mellitus*.

Dari penerapan *sequential methods* untuk *handling missing* value pada algoritma *C4.5* dan *Naïve Bayes* untuk memprediksi penyakit *Diabetes Mellitus* tersebut di atas dapat dibuatkan gambar seperti berikut ini.



Gambar 2.5 Skema Penerapan sequential methods untuk handling missing value pada algoritma *C4.5* dan Naïve Bayes untuk memprediksi penyakit *Diabetes Mellitus*