BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Beasiswa PPA

Beasiswa PPA (Peningkatan Prestasi Akademik) adalah salah satu dari jenis beasiswa yang diberikan oleh pemerintah melalui STMIK Amikom Purwokerto pada mahasiswa. Beasiswa PPA adalah beasiswa yang diberikan pemerintah pada mahasiswa untuk peningkatan pemerataan dan kesempatan belajar bagi mahasiswa yang kurang mampu dan mengalami kesulitan dalam hal ekonomi, terutama bagi mahasiswa yang mempunyai prestasi dibidang akademik. Tujuan dari diadakannya beasiswa PPA yaitu:

a. Meningkatkan pemerataan dan kesempatan belajar bagi mahasiswa yang kurang mampu dan mengalami kesulitan dalam hal ekonomi.
b. Mendorong dan mempertahankan semangat dan minat belajar mahasiswa agar dapat menyelesaikan pendidikannya dengan tepat waktu.
c. Sebagai pendorong bagi mahasiswa untuk mempertahankan prestasi akademik yang dimilikinya, sehingga dapat membantu dan memotivasi peningkatan kualitas pendidikannya.

2. *Data Mining*

a. Pengertian *Data mining*

Data Mining adalah suatu teknik yang cukup cepat dan mudah untuk menemukan suatu pengetahuan, pola atau hubungan antar data secara otomatis. Pengetahuan dalam *data mining* dapat ditemukan pada lima proses berurutan yaitu seleksi, prapemrosesan, transformasi, *data mining* dan interpretasi atau evaluasi (Fayyad et al.1996).

Menurut Lorena dalam Jamhur (2014) *data mining* adalah suatu proses yang meliputi pengumpulan, pemakaian data historis yang menentukan keteraturan, pola, dan hubungannya dalam set data berukuran besar. *Data mining* juga dapat diartikan sebagai proses yang memperkerjakan satu atau lebih teknik pembelajaran komputer untuk menganalisis dan menghasilkan pengetahuan secara otomatis atau kumpulan proses untuk menggali nilai tambah dari beberapa kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual (Sijabat, 2015).

Menurut Afriyudi dan Widyanto dalam Syahputra dan Safitri (2018) *data mining* adalah proses yang menggunakan teknik statistik, perhitungan, kecerdasan buatan, dan *machine learning* untuk menghasilkan dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang berhubungan dengan basis data yang besar. Hal yang terpenting yang berkaitan dengan data mining yaitu :

1. *Data mining* yaitu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses yaitu data yang sangat banyak.
3. Tujuan dari *data mining* adalah untuk mendapatkan suatu relasi atau pola yang mungkin memberikan petunjuk yang bermanfaat.

b. Pengelompokan *Data Mining*

Data mining dapat dikelompokan menjadi beberapa kelompok diantaranya yaitu :

1. Deskripsi

   Deskripsi adalah penggambaran pola dan kecenderungan yang terdapat dalam data yang sudah ada.

2. Estimasi

   Estimasi hampir sama dengan klasifikasi, kecuali pada *variable* target estimasi lebih ke arah numerik daripada kategori. Model dari estimasi dibangun menggunakan *record* yang lengkap serta menyediakan nilai dari variabel target sebagai nilai prediksi. Berikutnya pada estimasi nilai dari *variable* target dibuat dari nilai *variable* prediksi.

3. Prediksi

Prediksi yaitu memperkirakan sebuah nilai yang belum diketahui dimasa mendatang. Estimasi dan klasifikasi hampir mirip dengan prediksi.

4. Klasifikasi

Dalam klasifikasi terdapat target variabel kategori, misalnya pada kategori penggolongan pendapatan dapat dibagi menjadi tiga kategori, yaitu kategori tinggi, sedang dan rendah.

5. Pengklasteran (*Clustering*)

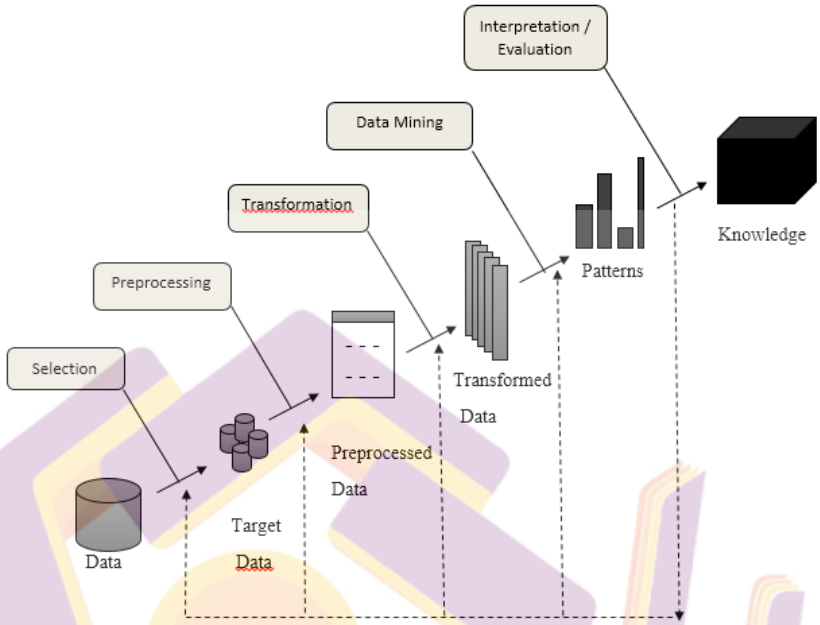
Pengelompokan dari beberapa *record*, pengamatan dan pembentukan kelas dari beberapa objek yang memiliki kemiripan disebut dengan pengklasteran. Kumpulan dari beberapa *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record* dalam klaster lain disebut dengan Klaster. Berbeda dengan klasifikasi, pengklasteran tidak ada variable target didalamnya. Pengklasteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dan variable target, tetapi mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), dimana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan dalam kelompok lain akan bernilai minimum.

6. Asosiasi

Asosiasi memiliki tugas untuk menemukan atribut yang muncul dalam waktu yang bersamaan. Dalam dunia bisnis asosiasi lebih umum disebut dengan analisis keranjang belanja.

c. *Knowledge Discovery In Database* (KDD)

Menurut Fayyad dalam buku (Kusrini, 2009). K*nowledge discovery in database* (KDD) dan *data mining* digunakan secara bergantian untuk menjelaskan proses penggalian informasi yang tersembunyi dalam basis data yang besar. K*nowledge discovery in database* (KDD) dan *data mining* memiliki konsep yang berbeda tetapi masih saling berkaitan antara satu dengan yang lainnya. *Data mining* merupakan salah satu tahapan dalam KDD. Proses KDD dapat dijelaskan sebagai berikut (Narwati, 2010) :



Gambar 2.1 Proses *knowledge discovery in database* (Nasari, 2015)

Knowledge Discovery in Database (KDD) memiliki beberapa proses, diantaranya :

1. *Data selection*

   Proses pemiihan atau penyeleksian data dari sekumpulan data operasional perlu dilakukan sebelum menuju pada tahapan penggalian informasi dalam KDD disebut dengan seleksi data. Data dari hasil penyeleksian digunakan untuk proses *data mining* yang disimpan dalam suatu berkas secara terpisah dari data operasional.

2. *Pre-processing* atau *Cleaning*

Sebelum melanjutkan pada tahapan *data mining*, dilakukan proses *cleaning* pada data yang menjadi fokus KDD terlebih dahulu. *Cleaning* memiliki beberapa proses antara lain yaitu pembuangan data yang terduplikasi, pemeriksaan data yang tidak konsisten serta memperbaiki kesalahan pada data, seperti kesalahan cetak (*tipografi*) serta dilakukan proses *enrichement* yaitu proses untuk memperbanyak data yang sudah ada dengan data ataupun informasi lain yang saling berkaitan dan diperlukan untuk KDD, contohnya seperti data atau informasi eksternal.

3. *Transformasi*

Transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining* disebut juga dengan *coding*. Tahap *coding* pada KDD merupakan proses kreatif yang sangat bergantung pada pola informasi yang akan dicari dalam suatu basis data.

4. *Data mining*

Suatu proses untuk mencari pola atau informasi yang menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu disebut dengan *data mining*. *Data mining* memiliki teknik atau metode atau algoritme yang sangat

bervariasi. Tujuan dan tahapan pada proses KDD bergantung pada pemilihan metode atau algoritme yang tepat.

5. *Interpretation* atau *Evaluation*

   Interpretation adalah bagian dari tahapan proses KDD. Proses dalam *data mining* menghasilkan pola informasi yang mudah dimengerti ketika ditampilkan. Tahapan *interpretation* memuat pemeriksaan pada pola atau informasi yang ditemukan apakah bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

d. Missing Value

Missing value adalah keadaan dimana beberapa nilai atribut dalam dataset kosong atau tidak ada nilainya. *Missing value* dapat terjadi karena informasi tentang objek yang diberikan sulit dicari atau informasi tersebut tidak ada, sehingga menyebabkan turunnya keakuratan dan kualitas data pada saat diolah. Selain itu, *missing values* juga dapat terjadi karena responden tidak memberikan jawaban pada alternatif jawaban yang ada, ataupun terjadinya kesalahan pada saat pengumpulan data, misalnya seperti pertanyaan yang terlewat sehingga tidak memberikan jawaban.

1) Macam-macam *missing values* menurut Little dan Rubin, yaitu
   a. *Missing Completely at Random* (MCAR), *missing* data yang terjadi tidak berkaitan dengan nilai semua *variable*, baik itu pada *variable* dengan *missing* data atau dengan *variable* pengamatan. Kesimpulannya berarti *missing* data dapat terjadi secara acak.
   b. *Missing at Random* (MAR) *missing* data yang terjadi hanya berkaitan dengan *variable* respon atau pengamatan.
   c. *Not Missing at Random* (NMAR) *missing* data yang terjadi pada suatu *variable* berkaitan dengan *variable* itu sendiri, sehingga tidak bisa diprediksi dari *variable* lain pada suatu *dataset.*
2) M*issing values* dapat ditangani dengan cara melakukan penghapusan data yang tidak lengkap. Jika data yang tidak lengkap jumlahnya relatif kecil, dibandingkan dengan keseluruhan data maka menghapus data yang tidak lengkap merupakan salah satu pendekatan yang masuk akal. Penghapusan data hilang ini tentunya memberikan hasil pendugaan yang kurang baik dikarenakan beberapa alasan berikut ini, yaitu:
   a. Jika dilakukan penghapusan data hilang maka contoh yang diambil akan berkurang, sehingga menyebabkan ketepatan dalam pendugaan berkurang.

b. Jika individu yang dikeluarkan ternyata hasilnya sangat berbeda dari data yang lain maka akan menghasilkan pendugaan yang bias.

Solusi yang tepat agar ukuran sampel tidak berkurang yaitu dengan melakukan *imputasi missing values* pada data. *Imputasi* yaitu tahap pengisian atau penggantian *missing values* pada *dataset* dengan nilai-nilai yang mungkin dari informasi yang didapatkan dari *dataset* tersebut.

3. Pohon Keputusan (*Decision Tree*)

Pohon keputusan menggunakan struktur pohon (*tree*) dalam penggambarannya, dimana pada setiap *node* menggambarkan atribut, cabangnya menggambarkan nilai dari atribut dan daun menggambarkan kelas. *Node* yang paling atas dari pohon keputusan disebut dengan *root*. Pohon keputusan merupakan metode klasifikasi yang paling banyak digunakan. Pohon keputusan memiliki fungsi untuk mengubah fakta menjadi pohon keputusan yang menggambarkan suatu aturan yang mudah dimengerti.

Manfaat dari pohon keputusan yaitu kemampuannya dalam mem-*breakdown* proses pengambilan keputusan yang rumit menjadi lebih simpel sehingga proses pengambilan keputusan mudah serta lebih menjelaskan solusi dari permasalahan yang ada. Banyak algoritme yang

dapat dipakai dalam pembentukan pohon keputusan, antara lain *ID3, C4.5* dan *CART*.

Arsitektur pohon keputusan dibuat dengan sedemikian rupa agar menyerupai pohon asli, dalam pohon keputusan terdapat beberapa bagian, yaitu:

1. *Root Node* yaitu terletak pada bagian paling atas dari pohon keputusan.
2. *Internal Node* adalah suatu percabangan dimana membutuhkan satu *input* dan mengeluarkan maksimal dua *output*.
3. *Leaf Node*, node ini terletak pada ujung pohon. *Leaf Node* hanya memiliki satu *input* dan tidak memiliki *output*.

Pohon keputusan juga mempunyai kelebihan dan kekurangan, diantaranya yaitu:

1. Kelebihan pohon keputusan
   a. Pengambilan keputusan pada daerah yang kompleks dapat diubah menjadi lebih sederhana.
   b. Proses pengujian hanya berdasarkan kriteria yang diperlukan saja sehingga dapat menghilangkan perhitungan yang tidak penting.
   c. Fitur yang dipilih dari *internal node* yang berbeda lebih mudah menyesuaikan. Fitur yang telah dipilih akan menjadi pembeda antara kriteria yang satu dengan kriteria yang lainnya.

d. Atribut dengan jumlah yang tidak banyak dapat mengurangi masalah yang dihasilkan pada *node internal* tanpa menurunkan kualitas keputusan yang akan diperoleh.

2. Kekurangan pohon keputusan
   a. Terjadinya *overlap* apabila hasil keputusan dan kriteria digunakan dalam jumlah yang sangat banyak. Sehingga berakibat bertambahnya waktu yang digunakan untuk pengambilan keputusan serta jumlah memori yang dibutuhkan semakin tinggi.
   b. Kumpulan jumlah *error* setiap tingkat pohon keputusan besar.
   c. Sulitnya mendesain pohon keputusan yang optimal.
   d. Kualitas dari keputusan yang didapatkan tergantung dengan bagaimana pohon tersebut didesain.

4. Klasifikasi

Suatu proses dalam menemukan model atau fungsi yang menerangkan serta menjadi pembeda antara konsep dengan kelas data, tujuan dari klasifikasi untuk memperkirakan kelas dari suatu objek yang tidak diketahui labelnya. Cara yang dilakukan untuk mencapai tujuan tersebut yaitu dengan membentuk suatu model yang membedakan data kedalam kelas-kelas yang berbeda berdasarkan aturan atau fungsi tertentu. Modelnya bisa berupa

aturan “jika-maka” berupa pohon keputusan atau formula matematis (Bustami, 2014).

5. Algoritme *C4.5*

Algoritme *C4.5* merupakan salah satu algoritme yang digunakan untuk membentuk *decision tree* (pohon keputusan). Algoritme ini merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. *Decision tree* adalah model prediksi menggunakan struktur *tree*. *Decision tree* memiliki konsep yaitu dapat mengubah data menjadi *decision tree* dan aturan-aturan keputusan (*decision rules*). Algoritme *C4.5* merupakan pengembangan dari algoritme *ID3*. Pengembangan yang dilakukan algoritme *ID3* antara lain yaitu bisa mengatasi *missing value*, bisa mengatasi *continu* data, dan *pruning*.

Cara algoritme *C4.5* dalam membangun pohon keputusan yaitu :

a. Memilih atribut yang nantinya akan digunakan sebagai *node* akar.
b. Buatlah sebuah cabang untuk setiap nilai.
c. Bagilah kasus dalam sebuah cabang.
d. Kemudian lakukan pengulangan proses untuk setiap cabang sampai semua kasus memiliki kelas yang sama pada cabang tersebut.

Algoritme *C4.5* menggunakan kriteria *split* yang telah dirubah yang dinamakan dengan *Gain Ratio* dalam proses pemilihan *split* atribut. Atribut *split* merupakan tahapan untuk membentukan suatu pohon keputusan pada algoritme *C4.5*.

Tahapan dari algoritme C4.5 adalah sebagai berikut :

a. Menyiapkan suatu data *training* yaitu diambil dari data yang ada sebelumnya dan sudah dikelompokan kedalam kelas tertentu.

b. Menghitung nilai *Entropy* dengan rumus:

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

Keterangan :

S = Himpunan kasus

n = Jumlah partisi S

pi = Proporsi S_i terhadap S

c. Setelah mendapatkan nilai *Entropy* maka akan digunakan untuk mencari nilai *gain* dengan rumus:

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

Keterangan :

S = Himpunan kasus

A = Atribut

N = Jumlah partisi atribut A

$|S_i|$ = Jumlah kasus pada partisi ke-i

|S| = Jumlah kasus dalam S

d. Kemudian mencari nilai *Split Info* dengan rumus:

$$\textit{Split Information (S,A)} = \sum_{i=1}^{0} \frac{Si}{S} log_2 \frac{Si}{S}$$

Keterangan :

S = Himpunan kasus

A = Atribut

Si = Jumlah sampel untuk atribut i

e. Setelah mendapatkan nilai *Gain* dan *Split Info*, lalu mencari nilai *Gain Ratio* dengan rumus sebagai berikut:

$$\textit{Gain Ratio (S,A)} = \frac{Gain(S,A)}{Split\ Information(S,A)}$$

Keterangan :

S = Himpunan kasus

A = Atribut

Gain(S,A) = Info gain pada atribut A

SplitInfo = split info pada atribut A

f. Nilai *Gain Ratio* tertinggi akan digunakan sebagai atribut akar, maka terbentuk pohon keputusan sebagai *node* awal.

g. Ulangi proses kedua sampai semua cabang memiliki kelas yang sama.

h. Maka akan terbentuk pohon keputusan.

i. Dari pohon keputusan yang terbentuk maka dapat menentukan *rule-rule*.

Berikut adalah tahapan dari algoritme *C4.5* yang dijelaskan dan diterapkan pada data tabel dibawah ini:

Tabel 2.1 Seleksi Penerimaan Siswa Baru

No.	Baca	Tulis	Hitung	Wawancara	Asal sekolah	Status
1	2	3	3	2	0	Diterima
2	2	3	3	1	0	Tidak
3	3	3	3	2	0	Diterima
4	2	1	2	2	0	Tidak
5	3	3	3	2	0	Diterima
6	2	3	3	1	1	Diterima
7	1	1	2	2	0	Tidak
8	3	3	3	2	0	Diterima
9	1	1	3	2	0	Tidak
10	2	1	2	1	1	Diterima

(Sumber: Fitriani, 2018)

Pada tabel 2.1 menjelaskan data seleksi penerimaan beasiswa, yang terdiri dari 10 data. Pada kriteria baca memiliki 3 kategori yaitu 1,2 dan 3. Kriteria tulis memiliki 2 kategori yaitu 1 dan 3. Kriteria hitung memiliki 2 kategori yaitu 2 dan 3. Kriteria wawancara memiliki 2 kategori yaitu 1 dan 2 serta kriteria asal sekolah memiliki 2 kategori yaitu 0 dan 1. Atribut status yang merupakan atribut kelas tujuan memiliki dua kategori keputusan yaitu diterima dan tidak.

Berikut tahapan dalam algoritme *C4.5* seperti dibawah ini:

a. Menyiapkan data yang sudah dikelompokan pada kelas tertentu. Data sudah ada pada tabel 2.1

b. Menghitung nilai *Entropy*

Atribut = Baca, Tulis, Hitung, Wawancara, Asal Sekolah dan Status

Atribut keputusan status dibagi menjadi 2 kelas yaitu diterima dan tidak. Pada kelas diterima berjumlah 6 dan pada kelas Tidak berjumlah 4.

Entropy [Total] = (-(6/10) * (log2 (6/10)) + (-(4/10) * (log2 (4/10)) = 0,759442099

Entropy [Baca-1] = (-(0/2) * (log2 (0/2)) + (-(2/2) * (log2 (2/2)) = 0

Entropy [Baca-2] = (-(3/5) * (log2 (3/5)) + (-(2/5) * (log2 (2/5)) = 0,970951

Entropy [Baca-3] = (-(3/3) * (log2 (3/3)) + (-(0/3) * (log2 (0/3)) = 0

Entropy [Tulis-1] = (-(1/4) * (log2 (1/4)) + (-(3/4) * (log2 (3/4)) = 0,811278

Entropy [Tulis-3] = (-(5/6) * (log2 (5/6)) + (-(1/6) * (log2 (1/6)) = 0,650022

Entropy [Hitung-2] = (-(1/3) * (log2 (1/3)) + (-(2/3) * (log2 (2/3)) = 0,918296

Entropy [Hitung-3] = (-(5/7) * (log2 (5/7)) + (-(2/7) * (log2 (2/7)) = 0,863121

Entropy [Wawancara-1] = (-(2/3) * (log2 (2/3)) + (-(1/3) * (log2 (1/3))

= 0,918296

Entropy [Wawancara-2] = (-(4/7) * (log2 (4/7)) + (-(3/7) * (log2 (3/7))

= 0,985228

Entropy [Asal Sekolah-0] = (-(4/8) * (log2 (4/8)) + (-(4/8) * (log2 (4/8)) = 1

Entropy [Asal Sekolah-1] = (-(2/2) * (log2 (2/2)) + (-(0/2) * (log2 (0/2)) = 0

c. Mencari nilai *Gain*

Gain [Baca] = 0,759442099 – ((2/10) * 0) + ((5/10) * 0,970951) +((3/10) * 0) = 0,273967

Gain [Tulis] = 0,759442099 – ((4/10) * 0,811278) + ((6/10) * 0,650022) = 0,044917

Gain [Hitung] = 0,759442099 – ((3/10) * 0,918296) + ((7/10) * 0,863121) = -0,12023

Gain [Wawancara] = 0,759442099 – ((3/10) * 0,918296) + ((7/10) * 0,985228) = -0,20571

Gain [Asal Sekolah] = 0,759442099 – ((8/10) * 1) + ((2/10) * 0) = -0,04056

d. Mencari nilai *Split Info*

Split Info [Baca] = -((2/10) * (log2 (2/10) + (5/10) * (log2 (5/10) + (3/10) * (log2 (3/10)) = 1,485475

Split Info [Tulis] = -((4/10) * (log2 (4/10) + (6/10) * (log2 (6/10)) = 0,970951

Split Info [Hitung] = -((3/10) * (log2 (3/10) + (7/10) * (log2 (7/10)) = 0,881291

Split Info [Wawancara] = -((3/10) * (log2 (3/10) + (7/10) * (log2 (7/10)) = 0,881291

Split Info [Asal Sekolah] = -((8/10) * (log2 (8/10) + (2/10) * (log2 (2/10)) = 0,721928

e. Mencari nilai *Gain Ratio*

Gain Ratio [Baca] = *Gain* / *Split Info* = 0,273967 / 1,485475 = 0,18443

Gain Ratio [Tulis] = 0,044917 / 0,970951 = 0,046261

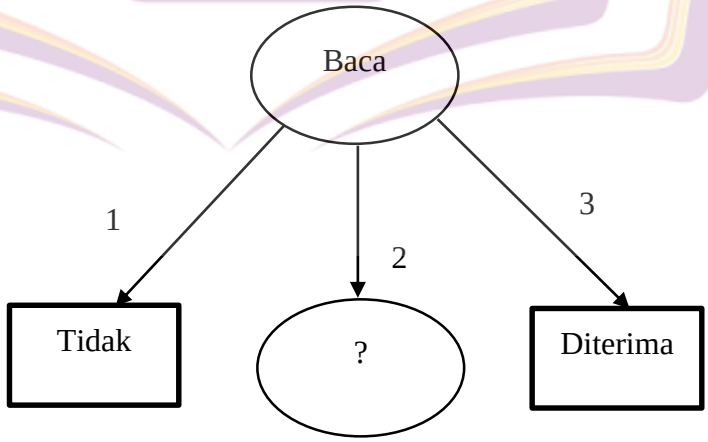
Gain Ratio [Hitung] = -0,12023 / 0,881291 = -0,13643

Gain Ratio [Wawancara] = -0,20571 / 0,881291 = -0,23341

Gain Ratio [Asal Sekolah] = -0,04056 / 0,721928 = -005618

f. Nilai *Gain Ratio* tertinggi digunakan sebagai *node* akar pertama pohon keputusan. Nilai *gain ratio* tertinggi yaitu 0,18443 dengan atribut Baca, maka atribut baca dijadikan *node* akar pertama.

g. Dibawah ini adalah pohon keputusan dengan atribut baca sebagai *node* akar pertama



Gambar 2.2 Pembentukan Pohon Keputusan

Gambar 2.2 menggambarkan suatu pohon keputusan, atribut baca memiliki 3 nilai yaiu 1,2 dan 3. Nilai atribut 1 sudah mengklasifikasikan kasus menjadi satu yaitu "Tidak" dan nilai atribut 3 juga sudah mengklasifikasikan kasus menjadi satu yaitu "Diterima". Sehingga nilai atribut 1 dan 3 tidak perlu lagi dilakukan perhitungan lebih lanjut. Tetapi nilai atribut 2 masih perlu diperhitungkan lagi karena masih terdapat "diterima" dan "tidak"

h. Ulangi proses diatas sehingga membentuk suatu pohon keputusan yang memiliki *rule-rule*

6. *Confusion Matrix*

Evaluasi kinerja dari model klasifikasi pada objek dengan perkiraan benar atau salah disebut dengan *confusion matrix*.

Tabel 2.2 *Confusion Matrix*

	Kelas Hasil Prediksi		
		Ya	Tidak
Kelas Aktual	Ya	*True Positive* (TP)	*False Negative* (FN)
	Tidak	*False Positive* (FP)	*True Negative* (TN)

Tabel 2.1 merupakan tabel *confusion matrix* dengan keterangan seperti dibawah ini:

- *True Positive* (TP) adalah proporsi positif yang terdapat dalam data set yang diklasifikasikan positif.
- *False Negative* (FN) adalah proporsi positif yang terdapat dalam data set yang diklasifikasikan negatif.
- *False Positif* (FP) adalah proporsi negatif yang terdapat dalam data set yang diklasifikasikan positif.
- *True Negatif* (TN) adalah proporsi negatif yang terdapat dalam data set yang diklasifikasikan negatif.

Selain evaluasi kinerja dari model klasifikasi berdasarkan objek dengan memperkirakan yang benar atau salah. Terdapat sejumlah ukuran yang digunakan untuk menilai atau mengevaluasi model klasifikasi, diantaranya adalah *accuracy* atau tingkat pengenalan, *error rate* atau tingkat kesalahan atau kekeliruan klasifikasi, *recall* atau *sensitivity* atau *true positive rate*, *specificity* atau *true negative rate*, *precision*, F-*Measure*, atau F_1 atau F-*score* atau rata-rata harmonik dari *precision* dan *recall*, serta F_β (J.Han et al.2012).

1) *Accuracy*

Akurasi atau tingkat pengenalan adalah proporsi kasus yang diidentifikasi benar terhadap jumlah semua kasus. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasi data secara benar.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \qquad (2.1)$$

Persamaan 2.1 diatas menjelaskan rumus perhitungan *accuracy* dimana TP merupakan proporsi positif yang terdapat dalam data set yang diklasifikasikan positif, TN proporsi negatif yang terdapat dalam data set yang diklasifikasikan negatif, FP proporsi negatif yang terdapat dalam data set yang diklasifikasikan positif dan FN proporsi positif yang terdapat dalam data set yang diklasifikasikan negatif.

2) *Error rate*

Error rate atau tingkat kesalahan atau disebut juga kekeliruan klasifikasi adalah kasus yang diidentifikasi salah dari semua kasus.

$$\textit{Error rate} = \frac{FP+FN}{TP+TN+FP+FN} \qquad (2.2)$$

Persamaan 2.2 diatas menjelaskan rumus perhitungan *error rate* dimana FP adalah jumlah kasus negatif yang diklasifikasikan sebagai positif, FN adalah jumlah kasus positif yang diklasifikasikan sebagai

negatif, TP adalah jumlah kasus positif yang diklasifikasikan sebagai positif, serta TN adalah jumlah kasus negatif yang diklasifikasikan sebagai negatif.

3) Recall

Recall atau *sensitivity* atau disebut juga *true positive rate* adalah proporsi kasus yang diidentifikasi dengan benar.

$$Recall = \frac{TP}{TP+FN} \quad (2.3)$$

Persamaan 2.3 diatas menjelaskan rumus perhitungan *recall* dimana TP adalah jumlah kasus positif yang diklasifikasikan sebagai positif dan FN adalah jumlah kasus positif yang diklasifikasikan sebagai negative

4) Specificity atau *true negative rate*

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \quad (2.4)$$

Persamaan 2.4 diatas menjelaskan rumus perhitungan *specificity* dimana TN adalah jumlah kasus negatif yang diklasifikasikan sebagai negatif dan FP adalah jumlah kasus negatif yang diklasifikasikan sebagai positif.

5) *Precision*

Precision adalah mengukur proporsi jumlah kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (2.5)$$

Persamaan 2.5 diatas menjelaskan rumus perhitungan *precision* dimana TP adalah jumlah kasus positif yang diklasifikasikan sebagai positif dan dan FP adalah jumlah kasus negatif yang diklasifikasikan sebagai positif.

6) F atau F_1 atau F-*score*

F atau F_1 atau F-*score* adalah salah satu perhitungan evaluasi yang mengkombinasikan *precision* dan *recall*.

$$\text{F-}score = \frac{2 \text{ x } precision \text{ x } recall}{precison+recall} \qquad (2.6)$$

Persamaan 2.6 diatas menjelaskan rumus perhitungan F-*score* rata-rata harmonik dari *precision* dan *recall*.

7) F_β

$$F_\beta = \frac{1+\beta^2 \text{x } precision \text{ x } recall}{\beta^2 \text{x } precision+ recall} \qquad (2.7)$$

Pada penelitian ini yang digunakan untuk evaluasi perhitungan hanya akurasi saja, karena penulis hanya ingin mengetahui tingkat ke akuratan dari algoritme *C4.5* dalam penerimaan beasiswa. Menganalisis kualitas *classifier* untuk mengenali *tuple-tuple* dari kelas yang ada merupakan tujuan dari *confusion matrix*. C*lassifier* mengenali *tuple* dengan benar, artinya *tuple* positif dikenali sebagai positif dan *tuple* negatif dikenali sebagai negatif atau dikenal dengan istilah TP dan TN. Sebaliknya, *Classifier* salah dalam mengenali *tuple*, *tuple* negatif dikenali sebagai positif dan *tuple* negatif dikenali sebagai positif atau dikenal dengan istilah FP dan FN (Suyanto, 2017).

B. Penelitian Sebelumnya

a. Penelitian yang dilakukan oleh Rismayanti (2016) yang bertujuan untuk mendapatkan sebuah *rule* dan pohon keputusan menggunakan algoritme C4.5 pada data penerimaan beasiswa. Atribut yang digunakan terdiri dari 4 atribut penilaian dan atribut kelas. Atribut penilaian yaitu IPK, Semester, Penghasilan Orang Tua (PO) dan Jumlah Tanggungan Orang Tua (JTO) serta atribut kelas yaitu status. Pengujian dilakukan dengan menggunakan aplikasi *Rapid Miner*. Hasil pengujian menghasilkan pohon keputusan berupa kategori mahasiswa yang diterima dan ditolak berdasarkan *node* akar dari sebuah *decision tree* yaitu IPK. Mahasiswa yang diterima adalah mahasiswa yang memiliki IPK bernilai Cumlaude dan Very Good dengan Jumlah Tanggungan Orang Tua terbanyak dan mahasiswa yang ditolak

adalah mahasiswa yang memilki IPK bernilai Good dan Very Good dengan Jumlah Tanggungan Orang Tua bernilai cukup.

Sedangkan pada penelitian ini, peneliti menggunakan data penerimaan beasiswa PPA di STMIK Amikom Purwokerto pada tahun 2015 sampai dengan 2018. Terdiri dari 279 *dataset*, dengan 9 atribut penilaian dan 1 atribut kelas. Pengujian dilakukan dengan menggunakan *Weka 3.9.*

b. Penelitian yang dilakukan oleh Hendrian (2018) yang bertujuan untuk mendapatkan nilai performa pada data penerimaan bantuan dana pendidikan dengan menggunakan algoritme *C4.5*. Pengujian dilakukan dengan menggunakan 254 dataset dengan menggunakan aplikasi *Rapid Miner*. Hasil pengujian menghasilkan pohon keputusan berupa atribut Penghasilan Ayah yang menjadi *node* akar dan juga tingkat akurasi yang diperoleh dari pengujian menggunakan *confusion matrix* berupa nilai *accuracy* sebesar 98,80%, nilai untuk *precision* sebesar 98,02% dan nilai untuk *sensitivity* atau *recall* sebesar 99,00% serta Kurva ROC.

Sedangkan pada penelitian ini, peneliti menggunakan data penerimaan beasiswa PPA di STMIK Amikom Purwokerto pada tahun 2015 sampai dengan 2018. Terdiri dari 279 *dataset*, dengan 9 atribut penilain dan 1 atribut kelas. Pengujian dilakukan dengan menggunakan *Weka 3.9* serta tidak menggunakan uji ROC *Curve.*

c. Penelitian yang dilakukan oleh Adi (2018) yang bertujuan untuk mengetahui nilai performa dari algoritme *naïve bayes* pada penerimaan beasiswa PPA. Pengujian dilakukan dengan menggunakan 386 *dataset* yang terdiri dari 321 data yang termasuk dalam kelas "Ya" dan 65 data dalam kelas "Tidak" dengan 13 variabel antara lain: nama lengkap, jenis kelamin, tempat lahir, tanggal lahir, fakultas, program studi, semester, IPK, NIM, nama orang tua, pekerjaan, tanggungan orang tua dan penghasilan orang tua. Setelah dilakukan *preprocessing* data, variabel yang digunakan menjadi 4 variabel karena variabel tersebut dianggap sebagai faktor penentu dalam pemberian beasiswa. Variabel atau atribut tersebut antara lain: semester, IPK, tanggungan orang tua dan penghasilan orang tua. Hasil dari pengujian menggunakan Algoritme *Naïve Bayes* menghasilkan akurasi berupa nilai akurasi terkecil sebesar 64% pada proses pengujian dengan menggunakan sampel sebanyak 100 sampel. Nilai akurasi tertinggi sebesar 97,66% pada proses pengujian dengan menggunakan sampel sebanyak 386 sampel. Hal ini menunjukan bahwa akurasi model semakin meningkat dengan bertambahnya data.

Sedangkan pada penelitian ini, peneliti menggunakan data penerimaan beasiswa PPA di STMIK Amikom Purwokerto pada tahun 2015 sampai dengan 2018. Terdiri dari 279 *dataset*, dengan 9 atribut penilaian dan 1 atribut kelas. Pengujian dilakukan dengan menggunakan *Weka 3.9.*

d. Penelitian yang dilakukan oleh Zerlinda (2019) yang bertujuan untuk mengklasifikasikan calon penerima bidikmisi menggunakan algoritme *KNN*. Dataset yang digunakan pada penelitian ini berjumlah 2039 data pendaftar bidikmisi dengan 7 atribut penilaian dan 1 atribut kelas. 7 atribut penilaian antara lain: penghasilan orang tua, jumlah tanggungan, pekerjaan ayah, pekerjaan ibu, luas tanah, sumber air, kepemilikan rumah serta 1 atribut kelas yaitu status penerimaan bidikmisi. Hasil pengujian dengan menggunakan perhitungan *confusion matrix* dengan menggunakan data *training* dan data *testing*, dimana data *training* yang digunakan sebanyak 1.539 data dan data testing sebanyak 500 data. Nilai akurasi yang dihasilkan dari pengujian sebesar 84,4% dengan nilai k=5 nilai dengan akurasi yang tertinggi.

   Sedangkan pada penelitian ini, peneliti menggunakan data penerimaan beasiswa PPA di STMIK Amikom Purwokerto pada tahun 2015 sampai dengan 2018. Terdiri dari 279 *dataset*, dengan 9 atribut penilaian dan 1 atribut kelas. Pengujian dilakukan dengan menggunakan *Weka 3.9.*

e. Penelitian yang dilakukan oleh Rahman (2015) yang bertujuan untuk mendapatkan sebuah *rule* dan pohon keputusan menggunakan Algoritme C4.5 serta untuk menyeleksi mahasiswa penerima beasiswa. Pengujian dilakukan dengan mengambil sampel sebanyak 40 mahasiswa calon penerima beasiswa. Atribut yang digunakan dalam penelitian antara lain:

IPK, pekerjaan, dan masa kerja. Hasil dari pengujian 40 sampel mahasiswa calon penerima beasiswa didapatkan sebanyak 18 mahasiswa yang tidak layak menjadi penerima beasiswa karena mempunyai IPK <3,00 dan sebanyak 8 mahasiswa yang tidak layak menjadi penerima beasiswa karena mempunyai masa kerja <5 tahun dan pekerja Non PNS, Sehingga dihasilkan sebanyak 14 mahasiswa yang layak menjadi penerima beasiswa karena telah memenuhi kriteria penerima beasiswa dari segi IPK, pekerjaan dan masa kerja yang telah ditentukan sebelumnya.

Sedangkan pada penelitian ini, peneliti menggunakan data penerimaan beasiswa PPA di STMIK Amikom Purwokerto pada tahun 2015 sampai dengan 2018. Terdiri dari 279 *dataset*, dengan 9 atribut penilaian dan 1 atribut kelas. Pengujian dilakukan dengan menggunakan *Weka 3.9.*

Tabel 2.3 Perbandingan Penelitian Sebelumnya

No	Peneliti	Dataset	Metode
1.	Rismayanti, 2016	*Dataset* penerimaan beasiswa sebanyak 20 sampel data dengan 5 atribut yang digunakan.	Penelitian ini menggunakan metode *data mining* dengan algoritme *C4.5*, tetapi tidak menggunakan *accuracy*.
2.	Hendrian, 2018	*Dataset* yang digunakan sebanyak 254 data bantuan dana pendidikan yang digunakan.	Penelitian ini menggunakan metode *data mining* dengan algoritme *C4.5* dan menggunakan *accuracy* serta kurva ROC.
3.	Adi, 2018	*Dataset* yang digunakan sebanyak 386 data dan 13 variabel yang digunakan.	Penelitian ini menggunakan metode *data mining* dengan algoritme *naïve bayes* dan menggunakan *accuracy*.
4.	Zerlinda, Slamet & Zukhronah, 2019	*Dataset* yang digunakan sebanyak 2039 data pendaftar bidikmisi dan 7 atribut yang digunakan.	Penelitian ini menggunakan metode *data mining* dengan algoritme *k-nearest neighbor* dan menggunakan *accuracy*.
5.	Rahman, 2015	*Dataset* penerimaan beasiswa 40 sampel data dengan 3 atribut yang digunakan.	Penelitian ini menggunakan metode *data mining* dengan algoritme *C4.5*, tetapi tidak menggunakan *accuracy*.

Tabel 2.4 Penelitian Sekarang

No	Peneliti	Dataset	Metode
1.	Ashari & Astuti, 2020	*Dataset* yang digunakan sebanyak 279 data penerimaan beasiswa dengan 10 atribut.	Penelitian ini menggunakan metode *data mining* dengan algoritme *C4.5* dan menggunakan *accuracy*.