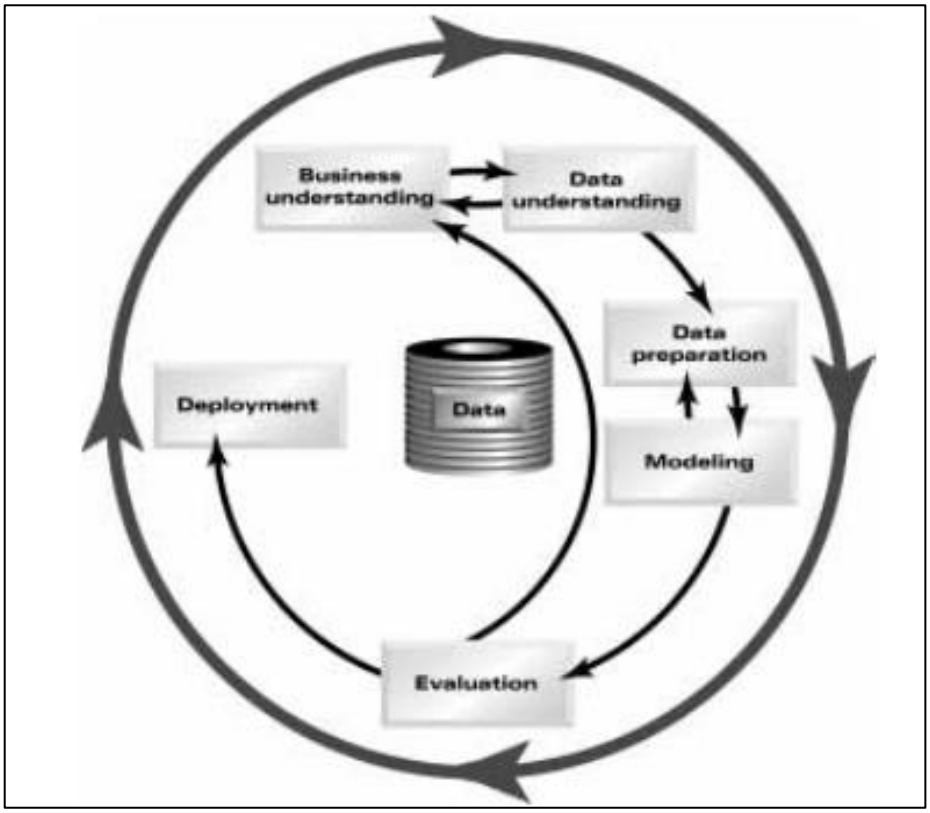
BAB II
STUDI LITERATUR

2.1 Data *Mining*

Data *mining* adalah suatu konsep yang digunakan untuk menmukan pola dan pengetahuan yang menarik dari basis data [4]. Data *mining* merupakan proses semi otomatik yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi pengetahuan potensial dan bermanfaat yang tersimpan di dalam basis data yang besar[8].



Gambar 2.1. CRISP-DM : Enam Fase

Sumber : Buku The CRISP-DM Model: The New Blueprint for Data *Mining* [9]

Data *mining* adalah bagian dari proses *Knowledge Discovery in Databases* (KDD) yang terdiri dari beberapa tahapan seperti pemilihan data, pra-pengolahan,

transformasi, data *mining*, dan evaluasi hasil [10]. Merujuk pada *Cross-industry standard process for data mining* (CRISP-DM) yang dapat dilihat pada **Gambar 2.1**, tahapan tahapan yang dilakukan adalah sebagai berikut [9] :

1. *Business understanding*

   Pada tahap pertama bisa disebut juga tahap pemahaman penelitian, menentukan tujuan proyek penelitian dalam perumusan mendefinisikan masalah data *mining*.

2. *Data Understanding*

   Dilakukan pengumpulan data, kemudian menganalisa data serta evaluasi kualitas data.

3. *Data Preparation*

   Persiapkan data mentah kemudian di seting untuk data akhir yang akan digunakan untuk fase selanjutnya, pilih kasus dan variabel yang diinginkan yang digunakan untuk menganalisa sesuai analisa masalah, lakukan transformasi pada variabel tertentu jika diinginkan, bersihkan data untuk alat pemodelan. Ada 4 tugas dalam data preparation yaitu data integration dengan tujuan untuk mengkombonasikan beberapa sumber data, data cleaning bertujuan untuk menghapus noise dan inconsistent data, data reduction yaitu memilih fitur yang akan digunakan berdasarkan beberapa pendekatan dan data transformation yaitu menyesuaikan bentuk data dengan wadahnya. Ada 6 cara untuk menangani noise data atau data kosong. Diantaranya sebagai berikut.

   - Mengabaikan tuple/*record*/baris
   - Mengisi secara manual data yang kosong
   - Memakai konstanta global semisal *Unkown* atau tak hingga.
   - Memakai perhitungan nilai tengah/rataan dari suatu fitur.
   - Memakai nilai tengah atau rataan dengan kelas yang sama dengan *record*
   - Menggunakan metode dengan mengisi nilai yang paling mungkin dengan metode regression, formula bayesian, atau decision tree.

   Outlier sering membuat data menjadi bias. Sehingga perlu pendeteksian outlier dengan cara :

   - Hitung Q1(kuartil bawah),Q2(median),Q3(kuartil atas.
   - Kemudian hitung intequartil range (IQR) yang merupakan rentang nilai

yang tidak termasuk outlier. IQR=Q3-Q1

- Maka nilai yang 1,5 * IQR lebih kecil dari Q1 atau 1,5 * IQR lebih besar dari Q3 dimaksud sebagai outlier.

4. *Modelling*

   Pada tahap ini, pilih dan terapkan teknik pemodelan yang tepat, lakukan pengaturan model untuk mengoptimalkan hasil, jika diperlukan lakukan ulang ke tahap persiapan sesuai dengan persyaratan spesifikasi dari teknik data *mining* tertentu .

5. *Evaluation*

   Melakukan evaluasi satu atau lebih model, tentukan apakah model sudah mencapai tujuan yang diterapkan dalam tahap pertama, mengambil keputusan mengenai penggunaan hasil data *mining*

6. *Deployment*

   Memanfaatkan model yang telah dibuat, *deployment* yang sederhana adalah sampai menghasilkan laporan sedangkan *deployment* yang kompleks adalah melaksanakan model untuk proses data *mining* paralel pada departemen lain.

Secara umum ada 6 tugas dalam data *mining*. Adapun tugas yang umum dilakukan dengan data *mining* yaitu *Description, Estimation, Prediction, Classification, Clustering* dan *Association.* [7]

1. *Description*

   Merupakan pengolahan data *mining* dengan melakukan prediksi/ peramalan. Tujuan metode ini untuk membangun model prediksi suatu nilai yang mempunyai ciri-ciri tertentu.

2. *Estimation*

   Estimasi merupakan teknik dalam data *mining* untuk memperkirakan nilai variabel target numerik menggunakan sekumpulan variabel prediktor numerik dan / atau kategori. Model dibuat menggunakan *record* "*complete*", yang memberikan nilai variabel target, serta prediktor. Kemudian untuk observasi baru dibuat estimasi nilai variabel target berdasarkan nilai prediktornya

3. *Prediction*

   Teknik prediksi dalam data *mining* mirip dengan klasifikasi dan estimasi, hanya saja untuk prediksi, hasilnya untuk di masa depan.

4. *Clustering*

   Teknik untuk mengelompokkan data ke dalam suatu kelompok tertentu.

5. *Classification*

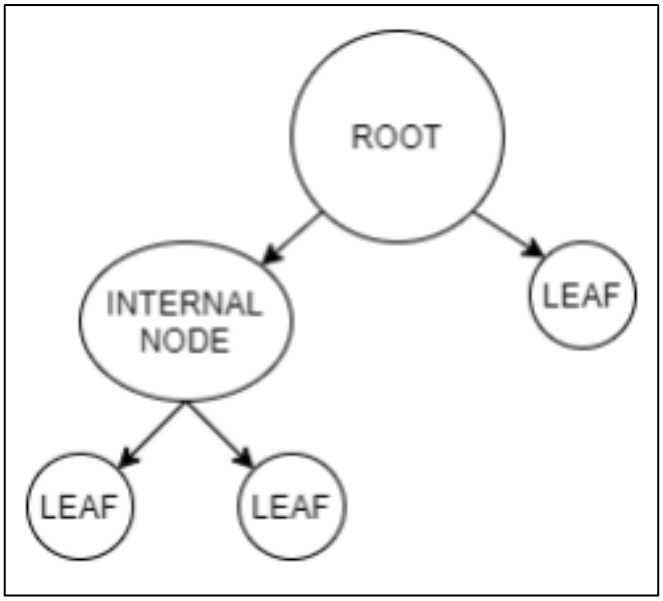
   Classification merupakan Teknik untuk mengklasifikasikan data. Perbedaannya dengan metode clustering terletak pada data, dimana pada clustering variabel dependen tidak ada, sedangkan pada kelasification diharuskan ada variabel dependen.

6. *Association*

   Teknik dalam data *mining* yang mempelajari hubungan antar data.

2.2 *Decision Tree*

Algorima *decision tree* merupakan serangkaian proses untuk membuat model berbentuk mirip pohon terbalik yang digunakan untuk menentukan target kelas yang telah diberikan nilai pada tiap fiturnya [11]. Setiap pohon memiliki *node*, *node* mewakili suatu atribut yang harus dipenuhi untuk menuju *node* selanjutnya hingga berakhir di *leaf* (tidak ada *node* lagi). Konsep data dalam *decision tree* adalah data dinyatakan dalam bentuk tabel yang terdiri dari fitur dan *record*. Fitur digunakan sebagai parameter yang dibuat sebagai kriteria dalam pembuatan pohon.



Gambar 2.2. *Decision Tree*

Adapun ilustrasi dari *decision tree* dapat dilihat pada **Gambar 2.2.** Adapun istilah istilah dalam *decision tree* adalah sebagai berikut :

1. *Root* merupakan *node* yang tidak memiliki input dari *node* sebelumnya. Dalam artian root merupakan derajat tertinggi dari atribut yang akan dijadikan rule. Adapun Root dapat memiliki beberapa output *node*.
2. *Internal Node* merupakan *node* yang memiliki input dari *node* yang derajatnya lebih tinggi. Internal *node* juga dapat memiliki beberapat output berupa *node*.
3. *Leaf* hanya memiliki input dari *internal node*/*root* dan merupakan luaran dari *decision tree*. Adapun luarannya berupa kelas yang akan menjadi label dari suatu *record*.

Dalam suatu penelitian tentang beasiswa menunjukkan bahwa algortima *decision tree* C4.5 lebih baik dibandingkan dengan *naïve bayes*[12]. perbandingan beberapa variasi dari algoritma *decision tree* yaitu ID3, C4.5, CART dan SLIQ didapatkan hasil bahwasannya C4.5 memiliki akurasi tertinggi [5]. Maka dalam penelitian ini penulis akan menggunakan algoritma *decision tree* C4.5.

Pada fitur *selection* pembuatan *decision tree* menggunakan perhitungan *node impurity*. Adapun secara umum perhitungan *node impurity* (I) dapat menggunakan formula pada **Persamaan 2.1**. dimana s adalah atribut yang akan di split, D adalah *dataset* yang akan dipecah dan s(D) = (D1, ..., Dk). Berdasarkan indeks i, seleksi terbaik dari kumpulan data D adalah yang nilai maksimal ΔI (s, D).

$$\Delta I(s,D) = I(D) - \sum_{i=1}^{k} \frac{|\mathrm{Di}|}{|\mathrm{D}|} SI(D_i) \quad \text{.....(2.1)}$$

Keterangan :

s : Himpunan Kasus

D : Atribut

I : *Node* Impurity/Gain

SI(Di): *Split Information* atribut D pada partisi ke-i

$|D_i|$: Jumlah kasus pada partisi ke-i

|D| : Jumlah kasus dalam D

2.3 Algoritma *decision tree* C4.5

Pada tahun 1970-an hingga awal 1980-an, J. Ross Quinlan, seorang peneliti yang berfokus pada machine learning, mengembangkan algoritma *decision tree* yang dikenal sebagai ID3 (*Iterative Dichotomiser*). Pekerjaan ini diperluas pada pekerjaan sebelumnya pada konsep pembelajaran, dijelaskan oleh E. B. Hunt, J. Marin, dan P. T. Stone. Quinlan kemudian mempresentasikan C4.5 (penyempurnaan ID3), yang menjadi tolak ukur yang sering dibandingkan dengan algoritma pembelajaran yang lebih baru. Pada tahun 1984, sekelompok ahli statistik (L. Breiman, J. Friedman, R. Olshen, dan C. Stone) menerbitkan buku *Classification* and Regression Trees (CART), yang menggambarkan generasi *binary tree*. ID3 dan CART diciptakan secara independen satu sama lain pada waktu yang hampir bersamaan, namun mengikuti pendekatan serupa untuk mempelajari pohon keputusan dari tuple pelatihan. Kedua algoritma landasan ini menghasilkan kesibukan pada induksi pohon keputusan. [4] Adapun perbedaan utama pada C4.5 dibandingkan dengan ID3 [13] yaitu :

1. Pengukuran *node impurity* yang dimodifikasi

   Dalam ID3 dikenal *Information Gain*. Adapun tujuan dalam modifikasi pengukuran *node impurity* adalah menghilangkan bias ditahap split fitur selection. Dalam C4.5 dikenal dengan *Information Gain Ratio* (IGR) yang didefinisikan pada **Persamaan 2.2**.

$$\Delta I(s,D) = \frac{\Delta I_E(s,D)}{SI(s,D)} \text{ ……..……...(2.2)}$$

Keterangan :

s : Himpunan Kasus

D : Atribut

I_E : Gain pada atribut E

Dimana split information SI(s,D) adalah entropy dari s(D) = (D1,..., Dn).Adapun perhitungan SI dapat menggunakan formula pada **Persamaan 2.3**.

$$SI(s,D) = -\sum p_i \log_2 p_i \ , \left(p_i = \frac{|\mathrm{Di}|}{|\mathrm{D}|}\right) \text{ ……..……....(2.3)}$$

Keterangan :

s : Himpunan Kasus

D : Atribut

p_i : Proporsi dari $|D_i|$ terhadap $|D|$

$|D_i|$: Jumlah kasus pada partisi ke-i

$|D|$: Jumlah kasus dalam D

2. Mendukung untuk menangani atribut kontinu (tidak perlu membedakannya)
3. Pengenalan metode pruning
4. Metode yang tepat untuk menangani *missing value*

Dalam pengembangannya C4.5 membutuhkan inputan berupa *training data* dan *test data*. *Training data* berupa data sampel yang akan digunakan untuk membangun sebuah *decision tree* yang telah tervalidasi kebenarannya. Sedangkan *test data* merupakan sekumpulan data yang nantinya akan digunakan dalam melakukan pengujian pada *decision tree*. Pada tahap latihan atau pembuatan *decision tree*, algoritma *decision tree C4.5* memiliki 2 target kerja, yaitu:

1. Pembuatan *decision tree*

   Tujuan dari implementasi algoritma *decision tree* adalah untuk Menyusun struktur data pohon yang dapat digunakan untuk menentukan kelas/target dari sebuah *record* baru yang belum memiliki kelas. Algoritma ini memilih pemecahan kasus yang terbaik dengan menghitung IGR, kemudian melakukan hal yang sama yaitu menghitung IGR pada beberapa *node* yang terbentuk di level berikutnya hingga terbentuk *leaf* yang merupakan kelas dari data.
2. Pembuatan aturan aturan (*decision rule*)

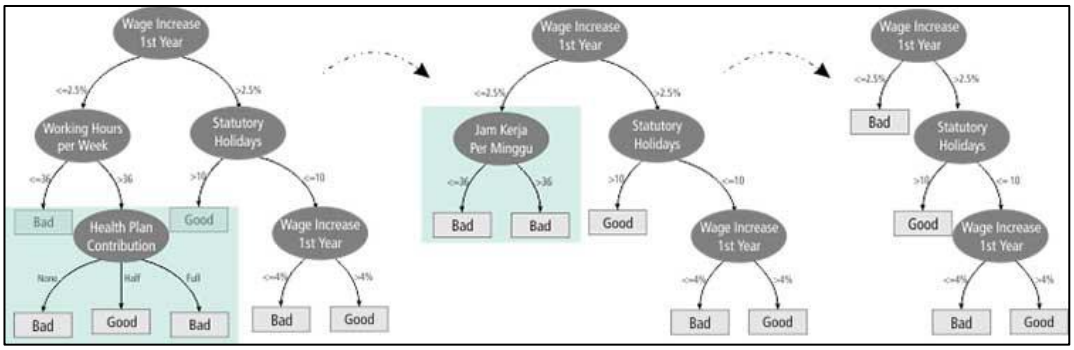
   Setelah *decision tree* terbentuk, selanjutnya akan di terjemahkan menjadi serangkaian aturan (*Decision rule*) membentuk suatu kondisi dalam bentuk *if-then*. Setiap *node* akan membentuk suatu percabangan dengan kondisi atau suatu *if*, sedangkan untuk leaf akan membentuk hasil atau suatu *then.* [14]

   Selain

2.4 *Reduced Error Pruning* (REP)

Reduced Error Pruning (REP) merupakan salah metode dalam *machine learning* yang berfungsi untuk memangkan *decision tree*, dimana metode ini akan menghilangkan bagian dari *decision tree* yang memiliki kemampuan prediksi yang rendah [15]. REP adalah salah satu algoritma *post pruning*. *Post pruning* adalah

salah satu metode pemangkasan *decision tree*, dimana proses pemangkasan pohon dilakukan setelah *decision tree* selesai dibangun atau dibuat [16]. *Post pruning* juga dikenal sebagai pemangkasan mundur, karena metode ini memangkas pohon keputusan dari *internal node* paling bawah menuju *internal node* paling atas [17].



Gambar 2.3. Reduce error *pruning* model health plan contribution

sumber: https://informatikalogi.com/algoritma-c4-5/

Metode REP merupakan salah satu metode *pruning* yang paling sederhana [18]. REP diusulkan oleh quinlan yang juga merupakan penemu dari algoritma C4.5. metode ini mempertimbangkan setiap *internal node* untuk dipangkas. Metode REP akan memangkas *internal node* dengan tingkat *error rate* yang tinggi dan menggantikannya dengan *leaf node*. Banyak peneliti yang menemukan bahwa REP memiliki performa yang sama baiknya dengan metode *pruning* yang lain dalam hal akurasi dan pembentukan ukuran daun [19].

Adapun contoh REP pada sebuah model diilustrsikan pada **Gambar 2.6.** Pada contoh tersebut dapat diketahui bahwa atribut "*Health Plan Contribution*" banyak memiliki kelas Bad, maka dari itu dipangkas menjadi *leaf node* Bad untuk keseluruhan atribut tersebut, dan seterusnya.

REP akan membagi data menjadi dua, yaitu training data dan test data. Training data adalah data yang digunakan untuk membentuk pohon keputusan, sedangkan test data digunakan untuk menghitung nilai *error rate* pada pohon setelah dipangkas. Karena REP merupakan bagian dari metode post *pruning* maka pemangkasan dimulai dari *internal node* paling bawah ke atas. Langkah - langkah pemangkasan pohon keputusan dengan metode REP adalah sebagai berikut :

1. Mengganti *internal node* paling bawah dengan *leaf node*, dan melabeli *leaf*

node dengan atribut yang memiliki kelas yang dominan muncul.

2. Setelah itu test data diproses menggunakan rule hasil pemangkasan, kemudian dihitung nilai *error rate*nya.
3. Kemudian test data juga diproses dengan rule awal, yaitu rule yang terbentuk sebelum pohon dipangkas
4. Kemudian dihitung nilai *error rate* dari test data yang telah diproses dengan rule awal.
5. Apabila nilai *error rate* yang dihasilkan dari pemangkasan pohon lebih kecil, maka pemangkasan dilakukan dan apabila *error rate* yang dihasilkan lebih besar maka tidak dilakukan pemangkasan.

2.5 *Feature Importance*

Feature Importance adalah tingkat pengaruh suatu fitur dalam model yang dibuat. Dalam decision tree dengan C4.5 menggunakan entrophy sehingga perhitungan *Feature Importance* dipengaruhi oleh entrophy itu sendiri. Adapun bobot entrophy suatu node dapat dihitung dengan persamaan 2.4.

$$weighted\ impurity \quad = \quad \frac{N_m}{N}\left(E_m - \left(\frac{N_l}{N_m} * E_l\right) - \left(\frac{N_r}{N_m} * E_r\right)\right) \qquad (2.4)$$

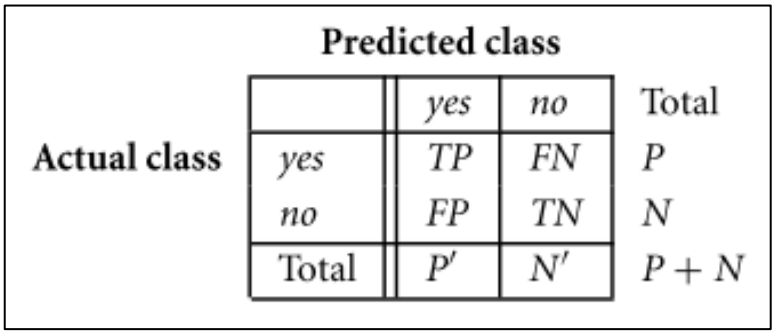
Keterangan:

N : Banyaknya sampel
Nm : Banyaknya sampel pada node m
Nl : Banyaknya sampe pada child node left
Nr : Banyaknya sampe pada child node right
Em : Entrophy node m
El : Entrophy child node left
Er : Entrophy child node right

2.6 *Confusion Matrix*

Confusion matrix adalah metode yang berguna untuk menganalisis seberapa baik model pengklasifikasi dapat mengenali tupel dari berbagai kelas. Dalam

perhitungannya dikenal 4 istilah yang akan dipakai diilustrasikan pada **Gambar 2.4**. 4 Istilah yang dimaksud yaitu TP, TN, FP, dan FN.



Gambar 2.4. *Confusion Matrix*

Tabel 2.1. Perhitungan Pada *Confusion Matrix*

Perhitungan	Persamaan
Accuracy, Recognition rate	$\frac{TP + TN}{P + N}$ (2.5)
Error Rate, Misclasification rate	$\frac{FP + FN}{P + N}$ (2.6)
Sensitivity, True Positive rate, Recall	$\frac{TP}{P}$ (2.7)
Specificity, True Negaative rate	$\frac{TN}{N}$ (2.8)
Precision	$\frac{TP}{TP + FP}$ (2.9)
F,F_1,F-score Rata rata Precision dan Recall	$\frac{2\ x\ Precision\ x\ Recall}{Precision +\ Recall}$ (2.10)
F_β, Yang mana β non-negative bilangan asli	$\frac{(1 + \beta^2)\ x\ Precision\ x\ Recall}{\beta^2\ x\ Precision + Recall}$ (2.11)

Nilai True Positive (TP) merupakan nilai yang didapat apabila nilai aktual terprediksi dengan nilai benar positif sedangkan True Negative (TN) merupakan nilai yang didapat apabila nilai aktual terprediksi dengan nilai benar negatif. Nilai False Positive (FP) didapatkan apabila nilai aktual bernilai negatif namun terprediksi dengan nilai positif. Sedangkan False Negative (FN) didapatkan apabila nilai aktual bernilai positif terprediksi dengan nilai negatif.

TP dan TN menjelaskan kapan model melakukan klasifikasi dengan benar. Semakin tinggi nilai TP dan TN, maka akan semakin baik pula nilai ketepatan klasifikasi yang didapatkan. FP dan FN memberi tahu saat pengklasifikasi melakukan kesalahan [4]. Terdapat beberapa perhitungan nilai pada confusion matrix, seperti nilai akurasi, presisi dan recall/sensitivity. Akurasi merupakan nilai yang dihasilkan dari tingkat kesamaan antara nilai prediksi dengan nilai aktual. Adapun perhitungan yang dimaksud dapat dilihat pada **Tabel 2.1.**

2.7 Penelitian Terkait

Pada tahap ini penulis mencari jurnal jurnal terpercaya yang berkaitan dengan topik yang diteliti. Setelah mencari, membaca serta menelaah dari beberapa sumber terpercaya. Penulis mendapatkan beberapa penelitian yang topik bahasannya sama dengan penelitian ini yaitu tentang beasiswa . Sehingga penulis mencoba merangkum dan menjabarkan hasil dari penelitian yang terkait dari **Tabel 2.2**. Selanjutnya dari penelitian terkait, penulis menjadikannya sebagai referensi dalam konfigurasi model dan rujukan dalam mempersiapkan data.

Bantuan biaya pendidikan (beasiswa) diatur dan ditetapkan dalam sebuah aturan perundang undangan. Dari peraturan tersebut [20] menandakan bahwasannya beasiswa menjadi sebuah tugas yang penting bagi pemerintah. Selain tugas dari pemerintah beasiswa juga merupakan salah satu komponen dalam meningkatkan motivasi mahasiswa. Hal ini didukung dengan penelitian [21] yang mana melakukan survey kepada mahasiswa UIN Alaudin Makasar. Hasil analisis deskriptif tes menunjukkan bahwa terdapat pengaruh pemberian beasiswa terhadap motivasi belajar mahasiswa. Hal ini ditandai dengan jawaban responden yang berada pada kategori tinggi.

Klasifikasi merupakan salah satu task pada data *mining*. Lebih dari 2 algoritma yang dapat digunakan dalam mengklasifikasi. Tahun 2020 Meida Cahyo Untoro dan rekan rekan melakukan penelitian tentang perbandingan kinerja beberapa algoritma klasifikasi yaitu *Decision Tree,* K-NN, Naive Bayes and SVM pada sekumpulan data UCI. Penelitian tersebut menyimpulkan bahwa *decision tree* merupakan model terbaik dengan akurasi tertinggi dibanding algoritma yang lain [22]. Hasil tersebut juga diperkuat oleh sebuah paper yang ditulis oleh Choirul Anan

dan Harry Budi Santoso pada tahun 2018. Penelitian ini menjelaskan perbandingan kinerja algoritma *decision tree* C4.5 dan Naïve Bayes. Dari hasil penelitian tersebut menjelaskan bahwasannya Akurasi dari model C4.5 96,4%. *Naïve Bayes* 95.11%. Dari penelitian ini kita dapat mengetahui bahwa algoritma *decision tree* C4.5 lebih baik dibandingkan Naïve Bayes[12].

Pada algoritma *decision tree* tidak hanya C4.5 namun ada beberapa versi. Tahun 2017 Mochammad Yusa dan rekan melakukan penelitian tentang evaluasi performa pada beberapa algoritma *decision tree* yaitu ID3, C4.5 dan CART. Kesimpulan dari penelitian tersebut menghasilkan akurasi C4.5 adalah yang terbaik dengan akurasi tertinggi yaitu 54%. Anuja Priyam Hal tersebut juga pernah melakukan penelitian yang membandingkan algoritma-algoritma *decision tree*. Algoritma yang dibandingkan yaitu ID3, C4.5 dan CART. Didapatkan hasil bahwasannya C4.5 merupakan algoritma terbaik dibandingkan kedua algoritma yang lain dengan akurasi tertinggi yaitu 68% [5].

Quinlan (1987) mengusulkan REP pada pembangunan *decision tree* dengan mengurangi kesalahan pada *decision tree* dalam mengklasifikasi. Metode ini telah dibuktikan oleh Binh Thai Pham pada tahun 2020. REP mampu meningkatkan kinerja pelatihan dan kemampuan memprediksi yang masing masing hingga 12% dan 7%. Selain *pruning* juga ada beberapa metode *pruning* yaitu Cost-Complexity Pruning (CCP), Pessimistic Error Pruning (PEP), Minimum Error Pruning (MEP) and Critical Value Pruning (CVP). REP terbukti memiliki kemampuan terbaik pada sekumpulan data yang kontinyu.

Dalam menyeleksi calon penerima beasiswa dapat memanfaatkan Algoritma C4.5. Penelitian yang dilakukan oleh Nadiya Hijriana dan Muhammad Rasyidan pada Tahun 2017 [23] menghasilkan sebuah model klasifikasi berupa decision tree. Pengembangan itu memanfaatkan algoritma *decision tree* C4.5. Serta setelah dievaluasi serta dianalisa atribut yang paling berpengaruh hingga tidak terlalu berpengaruh secara berturut turut yaitu : indeks prestasi semester, penghasilan dan tanggungan. Ditahun yang sama yaitu 2017 Maulana Miftakhul Faizin melakukan penelitian dengan algoritma *decision tree* C4.5 dengan suatu metode post-*pruning* yaitu *Reduced Error Pruning* (REP) [6] .

Tabel 2.2. Penelitian Terkait

No	Peneliti	Judul	Metode	Hasil
1	Meida Cahyo Untoro, dkk (2020)	Evaluation of *Decision Tree*, K-NN, Naive Bayes and SVM with MWMOTE on UCI *Dataset*	*Decision Tree*, K-NN, Naive Bayes dan SVM	*decision tree* 96.30%, K-NN 92.95%, Support Vector Machine 82.00%, and Naïve Bayes 78.74%
2	Abdurraghib Segaf Suweleh, dkk (2020)	Aplikasi Penentuan Penerima Beasiswa Menggunakan Algoritma C4.5	*decision tree* C4.5 dengan missing value Listwise Deletion , *decision tree* C4.5 dengan missing value Mean Substitution	Akurasi, Specificity dan sensitivitas dari *decision tree* C4.5 dengan *Listwise Deletion* yaitu 92%, 92.3%, dan 91.6%. Sedangkan Akurasi, Specificity dan sensitivitas *decision tree* C4.5 dengan *Mean Substitution* masing masing yaitu, 88%, 75%, dan 100%
3	Fiqih Satria, dkk (2020)	Prediksi Ketepatan Waktu Lulus Mahasiswa Menggunakan Algoritma C4.5 Pada Fakultas Dakwah Dan Ilmu Komunikasi UIN Raden Intan Lampung	*decision tree* C4.5	C4.5 memiliki nilai akurasi, presisi dan recall masing-masing 58,2%, 86,8% dan 60,4%. Dari perhitungan F-Measure didapatkan nilai 71 %

No	Peneliti	Judul	Metode	Hasil
4	Łukasz Gadomer dan Zenon A. Sosnowski (2020)	Pruning trees in C-fuzzy random forest	*Reduced Error Pruning* (REP), Cost-Complexity Pruning (CCP), Pessimistic Error Pruning (PEP), Minimum Error Pruning (MEP) and Critical Value Pruning (CVP).	Setiap *pruning* memiliki kemampuan pemangkasannya masing masing. REP adalah metode pemangkasan terbaik pada sekumpulan data yang kontinyu. Sedangkap CVP merupakan metode pemangkasan terbaik pada sekumpulan data diskrit.
5	Binh Thai Pham, dkk (2020)	Ensemble *machine learning* models based on *Reduced Error Pruning* Tree for prediction of rainfall-induced landslides	*Reduced Error Pruning* (REP)	REP meningkatkan kinerja pelatihan dan kemampuan prediksi mencapai 12% dan 7%
6	Choirul Anan dan Harry Budi Santoso (2018)	Perbandingan Kinerja Algoritma C4.5 dan Naive Bayes untuk Klasifikasi Penerima Beasiswa	*decision tree* C4.5 dan Naïve Bayes	Akurasi dari model C4.5 96,4%. Naïve Bayes 95.11%. Waktu proses keduanya 0s

No	Peneliti	Judul	Metode	Hasil
7	Mochammad Yusa, dkk (2017)	Evaluasi Performa Algoritma Klasifikasi *decision tree* Id3, C4.5, Dan Cart Pada *Dataset* Readmisi Pasien Diabetes	*decision tree* ID3, C4.5, dan CART	Model terbaik C4.5 dengan akurasi 54,13% *time execution* sebesar 4 detik. Dibandingkan kedua metode lainnya yaitu ID3 dan CART yang masing masing akurasinya 47,82% dan 47,84%. Serta time execution masing masing 9 detik dan 6 detik.
8	Maulana Miftakhul Faizin (2017)	Penerapan Metode Reduced Error (Rep) Pruning Pada Sistem Diagnosis Dehidrasi Pada Anak Berbasis Metode *decision tree* Dan Algoritma C4.5	*decision tree* C4.5 dengan *Reduced Error Pruning* (REP)	Nilai *Error Rate* dan Akurasi dengan REP : 0.142857143 dan 85% Sedangkan jika menggunakan REP Nilai *Error Rate* dan Akurasi : 0.085714286 dan 91%
9	Anuja Priyam (2013)	Comparative analysis *decision tree classification*	ID3, C4.5, CART	C4.5 merupakan algoritma terbaik dibandingkan kedua algoritma yang lain
10	Asmirawati (2016)	Pengaruh pemberian beasiswa terhadap motivasi belajar	Kuantitatif	terdapat pengaruh pemberian beasiswa terhadap motivasi belajar mahasiswa