

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi semakin berkembang pesat sehingga wajib bagi kita untuk terus mengikuti kemajuan teknologi tersebut. Perkembangan tersebut telah membawa manusia menentukan tingkat efisien yang tinggi dalam segala bidang. Salah satu bidang yang membutuhkan aplikasi yang dapat menunjang keputusan adalah bidang pendidikan yaitu untuk penentuan peminatan siswa di SMA NEGERI 2 PERBAUNGAN, karena bidang peminatan ini memerlukan pendukung keputusan yang tepat dan cepat dalam penentuan kemampuan akademik siswa.

SMA NEGERI 2 PERBAUNGAN memiliki dua jurusan yaitu IPA dan IPS, dalam penentuan jurusan biasanya dilakukan analisis data nilai yang dilakukan oleh sekelompok guru beserta pembagian kelas di kelompokkan berdasarkan nilai akademik yang di peroleh oleh siswa pada saat ujian masuk SMA, namun cara manual tersebut dinilai kurang efektif karena membutuhkan waktu yang lama juga rentan dengan kesalahan dalam menganalisis data nilai serta pemeriksaan nilai akibat *human error*. SMAN 2 Perbaungan ini memiliki sekitar 200 lebih siswa/i sehingga masalah yang sering muncul ialah pemilihan jurusan yang berlangsung lama karena dilakukan secara manual serta beberapa masalah seperti kesalahan dalam melakukan penghitungan nilai hasil akibat banyaknya siswa/i, oleh karena itu diperlukan sistem penilaian baru yang dapat memutuskan jurusan yang sesuai

dengan kriteria siswa/i dengan memanfaatkan data nilai Ujian Akhir Sekolah siswa di SMP.

Nilai ujian akhir sekolah dijabarkan untuk menentukan jurusan yang sesuai dengan potensi siswa, sehingga tidak timbul permasalahan seperti terhambatnya perkembangan potensi siswa akibat asal-asalan dalam memilih jurusan di SMA. Dengan menggabungkan semua nilai siswa dan di proses melalui sistem ini maka dapat membantu guru dalam menentukan kriteria siswa yang cocok untuk masuk ke jurusan IPA atau IPS.

Berdasarkan pemaparan diatas penulis mengangkat judul penelitian dengan judul **“Penerapan Algoritma CART Dalam Penentuan Jurusan Siswa Di SMA (Studi Kasus : SMA NEGERI 2 PERBAUNGAN)”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang mendasari penulis melakukan penelitian ini, penulis merumuskan beberapa rumusan masalah antara lain sebagai berikut :

1. Menganalisis nilai secara manual seperti melakukan penghitungan nilai serta pemeriksaan data nilai hasil merupakan cara yang kurang efektif dan membutuhkan waktu lama.
2. Beberapa kesalahan yang dilakukan oleh tim penilai dikarenakan *human error* sangat berdampak besar karena menyangkut jurusan yang akan siswa dialami untuk 3 tahun masa belajar di SMA tersebut.

1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini tidak melebar dan memudahkan dalam proses penelitian maupun proses perancangan, maka diperlukan batasan masalah sebagai berikut :

1. Penelitian ini terbatas hanya terpusat dan difokuskan pada siswa/i baru di SMAN 2 PERBAUNGAN
2. Perancangan aplikasi ini menerapkan algoritma CART dengan menggunakan metode *Decision Tree* untuk memutuskan jurusan yang cocok untuk siswa/i.
3. Aplikasi ini hanya bisa dikelola oleh admin, dan siswa/i menggunakan aplikasi tersebut untuk mengecek pengumuman seputar jurusan yang mereka dapatkan setelah mengikuti beberapa pengklasifikasian yang ada pada aplikasi tersebut.

1.4 Tujuan Penelitian

Tujuan dari penelitian yang dilakukan oleh penulis antara lain sebagai berikut :

1. Menerapkan aplikasi penentuan jurusan yang dapat mempermudah pihak sekolah dalam menentukan jurusan yang sesuai dengan siswa.
2. Mengaplikasikan bahasa pemrograman PHP, dan *database* MySQL untuk menghasilkan aplikasi penentuan jurusan.

1.5 Manfaat Penelitian

Penulis mengharapkan dari penelitian yang dilakukan dapat memberikan efek yang positif dan memberikan manfaat. Berikut manfaat yang ingin dicapai dalam penelitian ini :

1. Memudahkan dalam melakukan penentuan jurusan.
2. Menambah wawasan penulis tentang teknologi pemrograman PHP dan basis data MySQL yang dapat diterapkan langsung dengan mengembangkan aplikasi berbasis *web*.
3. Menciptakan sebuah aplikasi berbasis web yang diharapkan dapat mengurangi kelemahan sistem penentuan jurusan di SMA NEGERI 2 PERBAUNGAN yang masih menggunakan sistem manual.
4. Sebagai syarat untuk memperoleh gelar sarjana pada program studi Teknik Informatika Universitas Islam Sumatera Utara.

1.6 Sistematika Penulisan

Untuk mempermudah dalam penyusunan dan memahami skripsi maka penulis menyajikan sistematika penulisan sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini akan dijelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini memuat tentang materi-materi pendukung dalam penyusunan skripsi, mulai dari teori-teori yang digunakan, konsep-konsep yang akan diterapkan dalam menyelesaikan permasalahan yang penulis teliti dalam penelitian ini.

BAB III : METODE PENELITIAN

Pada bab ini memuat mengenai metode yang penulis gunakan dalam menyelesaikan rumusan masalah, tahap-tahap mengenai teknik pengolahan data, perancangan aplikasi, dan pembuatan aplikasi.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini memuat hasil-hasil yang didapat dari penelitian serta melakukan pembahasan atas hasil yang diperoleh. Kesulitan yang ditemukan saat perancangan dan pembuatan aplikasi.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini memuat kesimpulan dan saran penulis atas penelitian yang dilakukan.

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Sistem adalah suatu kumpulan objek atau unsur-unsur atau bagian-bagian yang memiliki arti berbeda-beda yang saling memiliki hubungan, saling bekerjasama dan saling mempengaruhi satu sama lain serta memiliki ketertarikan pada rencana atau plane yang sama dalam mencapai suatu tujuan tertentu pada lingkungan yang kompleks. (Ariyana : 2018)

Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan. (Mulyadi : 2016)

Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar. (Romney dan Steinbert, 2015 : 3)

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai dalam melaksanakan suatu kegiatan pokok perusahaan.

2.2 Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut. (Hasan dan Asep : 2014)

Pengertian aplikasi menurut beberapa ahli adalah sebagai berikut :

- a. Menurut Jogiyanto (1999 : 12) adalah penggunaan dalam suatu komputer, intruksi atau pernyataan yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.
- b. Menurut Kamus Besar Bahasa Indonesia (1998 : 52) adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.
- c. Menurut Rachmad Hakim S, adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur *windows* dan permainan dan sebagainya.
- d. Menurut Harip Santoso, adalah suatu kelompok file (*form, class, report*) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi *payroll*, aplikasi *fixed asset*, dan lain-lain.

2.3 Pengertian Penentuan Jurusan

Penentuan jurusan adalah proses yang dilakukan sebagai tahapan agar mendapatkan jurusan yang sesuai dengan minat dan kemampuan dari masing-masing siswa. (Haris & Kurniati : 2022)

Penjurusan siswa baru merupakan proses yang dilakukan untuk mengarahkan siswa baru ke jurusan yang lebih rinci untuk meningkatkan kemampuan dan sebagai penunjang untuk mengembangkan diri. (Nugroho : 2015)

2.4 Website

Website merupakan kumpulan dari halaman-halaman *web* yang berhubungan dengan *file-file* lain yang saling terkait. Dalam sebuah *website* terdapat suatu halaman yang dikenal dengan sebutan *home page*. *Home page* adalah sebuah halaman yang pertama kali ketika seseorang mengunjungi sebuah *website*. Dari *home page*, pengunjung dapat mengklik *hyperlink* untuk pindah ke halaman lain yang terdapat dalam *website* tersebut. Sebuah *home page* biasanya merupakan sebuah *file* dengan nama *index.htm* atau *index.html*. (Jhonsen : 2002)

2.5 Pengertian Database

Database adalah sekumpulan *file* data yang saling berhubungan dan berorganisasi sedemikian rupa sehingga memudahkan untuk mendapat dan memproses data. Lingkungan sistem *database* menekankan data yang tidak tergantung (*independent data*) pada aplikasi yang akan menggunakan data. (Andi : 2006)

2.6 Teknologi Yang Digunakan

2.6.1 HTML (*Hypertext Markup Language*)

HTML (*Hypertext Markup Language*) merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan dihalaman web. Oleh karena itu agar dapat membuat program aplikasi diatas halaman web anda terlebih dahulu harus mengenal dan menguasai HTML. (Bimo Sunarfrihantono : 2002)

2.6.2 PHP (*Hypertext Preprocessor*)

PHP adalah bahasa pelengkap HTML yang dapat digunakan untuk membuat aplikasi dinamis yang memungkinkan manipulasi dan pemrosesan data. *Syntax* apa pun yang diberikan sepenuhnya dijalankan di *server* sementara hanya hasilnya yang dikirim ke *browser*. Kemudian itu adalah bahasa *scripting* yang ditempatkan di *server* dan diproses oleh *server*. Hasilnya dikirim ke klien tempat pengguna menggunakan *browser*. PHP merupakan sebuah perangkat lunak *Open Source*. (Tim EMS 2012 : 61)

2.6.3 XAMPP

XAMPP merupakan *software server apache* dimana dalam XAMPP yang telah tersedia *database server* seperti MySQL dan PHP programming. XAMPP memiliki keunggulan yaitu cukup mudah dioperasikan, tidak memerlukan biaya serta mendukung instalasi pada *Windows* dan *Linux*. Keuntungan lain yang didapatkan adalah hanya dengan melakukan instalasi cukup satu kali kemudian didalamnya tersedia MySQL, *apache web server*, *database server* PHP. (Mawaddah & Fauzi, 2018)

2.6.4 MySQL

MySQL merupakan suatu program *database server* dimana perangkat tersebut mampu untuk digunakan sebagai transaksi menerima dan mengirim dengan waktu yang singkat pengguna dengan jumlah yang banyak sesuai standar SQL (*Structured Query Language*) yaitu bahasa pemrograman *database*. MySQL dapat diakses berdasarkan *privillage* (hak *user*) secara bersamaan. (Parulian, 2017)

2.7 Unified Modelling Language (UML)

UML (Unified Modelling Language) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (*modelling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. (Nugroho 2010 : 6)

UML (Unified Modelling Language) adalah bahasa pemodelan standar yang memiliki sintak dan semantik. (Widodo 2011 : 6)

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa “*Unified Modelling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis objek (*Object Oriented Programming*)”.

2.7.1 Langkah-Langkah Pembuatan UML




Dalam membuat *UML* tidak terdapat aturan yang baku dalam penggunaannya, akan tetapi para pengembang di seluruh dunia sering menggunakan langkah-langkah sebagai berikut dalam membuat *UML*. Berikut adalah langkah-langkah yang paling umum untuk membuat *UML* :

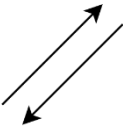
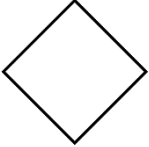
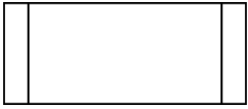
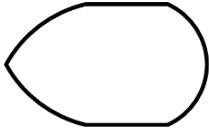
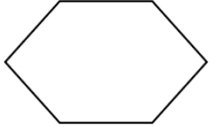
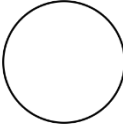
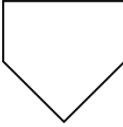
1. Membuat *Flowchart* Diagram

Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan

dihubungkan dengan garis atau arah panah. *Flowchart* berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain itu dengan menggunakan bagan alur proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan *flowchart* dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis. Adapun simbol-simbol yang digunakan dalam pembuatan *flowchart* diagram dapat dilihat pada tabel berikut :

Tabel 2.1 Simbol Flowchart Diagram

Simbol	Nama Simbol	Penjelasan Simbol
	<i>Input/Output</i>	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
	<i>Terminator</i>	Simbol untuk permulaan (<i>start</i>) atau akhir (<i>stop</i>) dari suatu kegiatan.
	<i>Process</i>	Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
	<i>Flow</i>	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.

		<p>Simbol ini disebut juga <i>connecting line</i>.</p>
	<p><i>Decision</i></p>	<p>Simbol pemilihan proses berdasarkan kondisi yang ada.</p>
	<p><i>Predifine Process</i></p>	<p>Simbol untuk pelaksanaan suatu bagian (<i>sub-program</i>)/prosedure.</p>
	<p><i>Display</i></p>	<p>Simbol yang menyatakan peralatan <i>output</i> yang digunakan.</p>
	<p><i>Preparation</i></p>	<p>Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.</p>
	<p><i>On-page Reference</i></p>	<p>Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang sama.</p>
	<p><i>Off-page Reference</i></p>	<p>Simbol untuk keluar-masuk atau penyambungan proses pada lembar kerja yang berbeda.</p>
	<p><i>Document</i></p>	<p>Simbol yang menyatakan <i>input</i> bersal dari dokumen dalam bentuk</p>

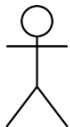
		kertas atau <i>output</i> dicetak dalam bentuk kertas.
	<i>Database</i>	Tempat penyimpanan data.

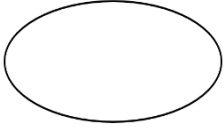

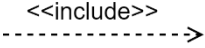
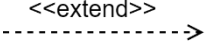
Sumber : <https://www.dicoding.com/blog/flowchart-adalah/>

2. Membuat *Use Case* Diagram

Pada tahap ini membuat aktor atau *user* tipe apa saja yang terlibat dalam sistem dan menentukan hal yang bisa dilakukan *user* terhadap sistem. Adapun simbol-simbol yang digunakan dalam pembuatan *use case* diagram dapat dilihat pada tabel berikut :

Tabel 2.2 Simbol *Use Case* Diagram

Simbol	Nama Simbol	Penjelasan Simbol
	<i>Actor</i>	Menggambarkan tokoh atau seseorang yang berinteraksi dengan sistem. Dan dapat menerima dan memberi informasi pada sistem.


	<i>Use Case</i>	Menjelaskan fungsi dari kegunaan sistem yang di rancang.
	<i>Assiciation</i>	Menghubungkan antara <i>use case</i> dengan aktor tertentu.
	<i>Include</i>	Menunjukkan bahwa <i>use case</i> satu merupakan bagian dari <i>use case</i> lainnya.
	<i>Extend</i>	Menunjukkan arah panah secara putus-putus dari <i>use case</i> ke <i>base use case</i> .






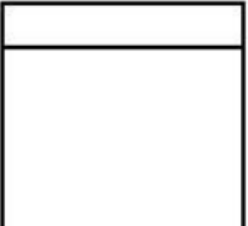
Sumber : <https://www.dicoding.com/blog/apa-itu-uml/>

3. Membuat *Activity* Diagram

Activity diagram mirip dengan *flowchart*. Pada tahap ini menggambarkan bagaimana sistem bekerja secara keseluruhan. Adapun simbol-simbol yang digunakan dalam pembuatan *use case* diagram dapat dilihat pada tabel berikut :

Tabel 2.3 Simbol *Activity* Diagram

Simbol	Nama Simbol	Penjelasan Simbol
	<i>Initial</i>	Titik awal untuk memulai suatu aktivitas.

	<i>Final</i>	Titik akhir untuk mengakhiri aktivitas.
	<i>Activity</i>	Menandakan sebuah aktivitas.
	<i>Decision</i>	Pilihan untuk mengambil keputusan.
	<i>Fork Atau Join</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Flow Final</i>	Untuk mengakhiri suatu aliran.
	<i>Swimlane</i>	Untuk mengelompokkan activity berdasarkan aktor.

Sumber : <https://www.dicoding.com/blog/apa-itu-uml/>

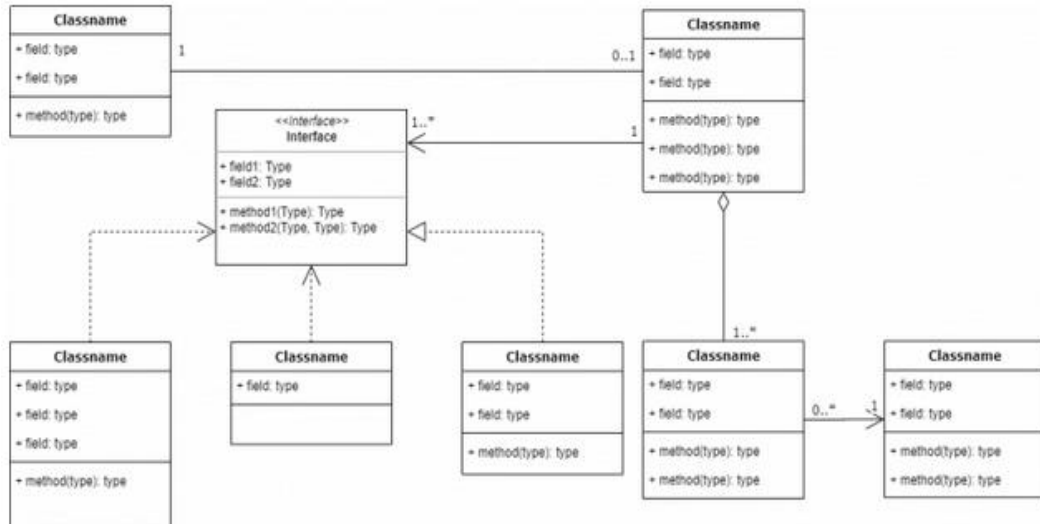
4. Membuat *Class Diagram*

Pada tahap ini menentukan *class-class* apa saja yang digunakan.

Pembuatan *class diagram* dapat mempermudah dalam

pengimplementasian sistem yang dirancang ke dalam baris-baris kode.

Adapun contoh dari *Class* diagram dapat dilihat pada gambar berikut.



Gambar 2.1 Contoh *Class* Diagram

Sumber : <https://www.dicoding.com/blog/apa-itu-uml/>

2.8 Decision Tree

Decision tree adalah metode pengambilan keputusan teknik klasifikasi dengan struktur *flowchart* yang mirip dengan pohon. Model yang dibentuk oleh metode ini sangat mudah dipahami sehingga menjadikan metode ini sangat umum dan populer. Terdiri dari beberapa algoritma dalam membangun *tree* yaitu CART, ID5, dan C4.5. Dalam penelitian ini menggunakan algoritma CART.

2.9 Algoritma CART (*Classification And Regression Tree*)

Merupakan algoritma yang umum dan banyak digunakan yang mengintegrasikan berbagai faktor dari sumber yang berbeda untuk masalah

klasifikasi dan regresi berdasarkan biner rekursif. Algoritma ini tergolong dalam model *nonparametric* yang tidak memerlukan bentuk fungsional dan telah terbukti menjadi alat ampuh untuk masalah prediksi dan klasifikasi. CART mengandalkan partisi biner rekursif dari data dasar pembangunan pohon regresi. Pohon didirikan dari dataset yang dikumpulkan di simpul pohon akar dan setiap node dibagi menjadi dua node turun menggunakan variabel pemisahan. Pemilihan variabel pemisahan mencari penurunan variabilitas dalam node dan untuk peningkatan variabilitas antara node, dengan setiap partisi yang diperoleh menghasilkan pohon dengan variabilitas yang lebih sedikit daripada pohon sebelumnya.

Berdasarkan penelitian menunjukkan bahwa berdasarkan nilai precision, nilai precision metode naive bayes lebih baik dibandingkan dengan metode yang lain dengan nilai 77,51%. Sedangkan berdasarkan nilai recall dan accuracy, decision tree lebih baik dibandingkan dengan metode yang lain dengan nilai recall sebesar 90,80% dan nilai accuracy sebesar 79,41%.

Tahapan-tahapan yang dilakukan pada algoritma CART ini sebagai berikut: Pertama, lakukan penyusunan calon cabang (candidate split) terhadap seluruh variabel prediktor dengan lengkap. Daftar yang berisi calon cabang tersebut dinamakan calon cabang mutakhir. Kedua, lakukan penilaian seluruh calon cabang mutakhir dengan melakukan perhitungan nilai besaran kesesuaian $\Phi(s|t)$. Lakukan tahapan kedua sampai tidak adalagi noktah keputusan dan akan berhenti jika tidak ada lagi noktah keputusan. Untuk mencari kesesuaian $\Phi(s|t)$ dari calon cabang s pada noktah keputusan t menggunakan persamaan sebagai berikut :

$$\Phi(s|t) = 2P_L P_R Q(s|t) \dots \dots \dots (1)$$

$$\text{Dimana } Q(s|t) = \sum_{i=1}^{\text{jumlah}} |P(j|tL) - P(j|tR)| \dots \dots \dots (2)$$

t_L = cabang kiri dari noktah keputusan t

t_R = cabang kanan dari noktah keputusan t

$$\text{dimana } P_L = \frac{\text{jumlah catatan pada calon kiri } tL}{\text{jumlah catatan pada data latihan}} \dots \dots \dots (3)$$

$$P(j|t_L) = \frac{j \text{ calon cabang kiri } tL}{\text{noktah keputusan } t} \dots \dots \dots (4)$$

$$P_R = \frac{\text{jumlah catatan pada calon kiri } tR}{\text{jumlah catatan pada data latihan}} \dots \dots \dots (5)$$

$$P(j|t_R) = \frac{j \text{ calon cabang kiri } tR}{\text{noktah keputusan } t} \dots \dots \dots (6)$$

2.9.1 Kelebihan Algoritma CART

Adapun kelebihanannya yaitu :

1. Metode klasifikasi ini lebih mudah di interpretasikan.
2. Mempunyai tingkat akurasi yang tinggi.
3. Dapat melakukan handling variabel dalam jumlah banyak dengan skala variabel campuran melalui prosedur pemilihan biner.

2.9.2 Kekurangan Algoritma CART

Adapun kekurangannya yaitu :

1. Hasil akhir tidak didasarkan pada model probabilistik.
2. Tidak ada tingkat probabilitas atau selang kepercayaan yang berhubungan dengan dugaan yang didapat dari pohon CART untuk pengelompokkan data baru.

2.10 Penelitian Terdahulu

Ada beberapa penelitian yang terkait dengan penelitian yang di angkat oleh penulis adalah :

1. Penelitian Faqih Abdul Haris dan Kurniati pada tahun 2020 yang berjudul “PENERAPAN ALGORITMA *CLASSIFICATION AND REGRESSION TREE* (CART) UNTUK KLASIFIKASI JURUSAN SISWA BARU MAN 1 OKU TIMUR”.

Kesimpulan : Algoritma *CART* sangat direkomendasikan untuk mengklasifikasikann penjurusan, dengan catatan jika atribut yang dimiliki dalam penentuan jurusan mayoritas kedalam bentuk numerik maka Algoritma *CART* lebih dianjurkan diterapkan dalam melakukann klasifikasi penjurusan siswa. Atribut pilihan jurusan menjadi *node* pertama dalam mengklasifikasikan jurusan, Algoritma *CART* dapat dikatakan baik karena masuk kategori *Excellent Classification*.

2. Penelitian Ade, Lince, dan Taronisokhi pada tahun 2018 yang berjudul “IMPLEMENTASI ALGORITMA CART UNTUK MENGLASIFIKASIKAN BUKU YANG PALING SERING DIBACA (STUDI KASUS : YAYASAN CINTA BACA)”.

Kesimpulan : Pengujian klasifikasi buku menggunakan teknik *data mining* dengan Algoritma *Cart* dapat dimulai dengan *menginput* data pembaca dengan *database Microsoft Excel* dengan ekstensi CSV. Mengklasifikasikan data dengan Algoritma *Cart* dengan cara menentukan atribut klasifikasi dari hasil transformasi sehingga dapat memperoleh buku yang paling sering dibaca.

3. Penelitian Eti Yonika pada tahun 2019 yang berjudul “Implementasi Algoritma Classification and aregression Tree (Cart) Dalam Klasifikasi Ekonomi Keluarga Pada Desa Dagang Kelambir Tg.Morawa”.

Kesimpulan : Data warga dapat diklasifikasi dengan menggunakan Algoritma Classification and Regresion Tree (Cart) dengan melihat data yang memenuhi nilai kesesuaian terbesar berdasarkan data warga sebelumnya. Namun dalam penyusunan candidate split dan perhitungan keseluruhan calon cabangnya sangat sulit jika data yang diolah dalam jumlah yang besar.