BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Penelitian yang dilakukan oleh Zakarias Situmorang dkk pada 2022, membahas tentang algoritma C-45 dalam memprediksi minat calon mahasiswa. Dalam penelitian ini, penulis mengevaluasi bahwa setelah penelitian yang dilakukan maka diperoleh hasil 4 aturan baru dengan menggunakan kriteria jenis kelamin, minat, jurusan asal sekolah dan hobi. Tujuan penelitian adalah untuk memprediksi minat calon mahasiswa dengan algoritma C45 melalui data preprocessing yang meliputi data selection, data preprocessing/ data cleaning, data transformation dan data reduction. Masalah klasifikasi berakhir dengan dihasilkan sebuah pengetahuan yang direpresentasikan dalam bentuk diagram yang biasa disebut pohon keputusan (*decision tree*).

Penelitian yang dilakukan oleh Amalia Iftitah dkk pada tahun 2022, membahas Penerapan Algoritma C.45 Untuk Analisis Pengadaan Peralatan dan Mesin Kantor. Berawal dari Kartu Inventaris Barang (KIB) yang masih belum dikelola dengan baik yaitu KIB peralatan dan mesin kantor dan menyulitkan pada kegiatan pengadaan peralatan kantor, sehingga diperlukan sistem pendukung keputusan untuk memprediksi pengusulan pengadaan

peralatan dan mesin kantor. Prediksi ini dilakukan dengan tujuan untuk mempermudah khususnya sub bidang aset dalam melaksanakan tugas. Prediksi tersebut memiliki manfaat dalam pelaksanaan penggantian peralatan dan mesin kantor dapat dikelola dengan baik sesuai dengan kebutuhan karyawan agar kinerja karyawan lebih maksimal. Metode pengumpulan data menggunakan metode survey, literatur, dan wawancara dengan anggota subbidang aset. Metode pengolahan data dalam laporan ini

menggunakan metode kuantitatif dengan algoritma C.45. Analisis mengenai prediksi pengusulan peralatan dan mesin kantor menggunakan algoritma C.45 dengan bantuan Aplikasi Rapidminer mampu memberikan hasil yang berupa decision tree yang dapat digunakan oleh anggota subbidang aset dalam mengambil keputusan.

Menurut Embun Fajar Wati dkk, pada tahun 2022, penelitian tentang penerapan algoritma KNN, Naive Bayes Dan C4.5 dalam memprediksi kelulusan mahasiswa. Bertujuan untuk meneliti tingkat kelulusan dan jumlah siswa yang akan mempengaruhi proses akreditasi bagi perguruan tinggi. Seleksi penerimaan mahasiswa baru berfungsi untuk mendapatkan mahasiswa yang berkualitas. Kualitas mahasiswa dapat diukur dengan masa pendidikan di perguruan tinggi. Lulusan yang tepat waktu yang mempunyai predikat mahasiswa berkualitas. Jumlah kelulusan mahasiswa sangat mempengaruhi penilaian akreditasi bagi suatu perguruan tinggi. Banyak faktor yang sangat berpengaruh terhadap ketepatan waktu kelulusan bagi mahasiswa seperti, jenis kelamin, umur, status pernikahan, IPK, dan status pekerjaan. Variabel-variabel tersebut akan diolah dengan algoritma KNN (K-Nearest Neighbor), Naive Baye, C4.5. Data preprocessing menggunakan data mahasiswa yang terdiri dari jenis kelamin, umur, status pernikahan, dan status pekerjaan juga IPK. Berdasarkan data kelulusan, 203 siswa lulus tepat waktu dan 97 siswa lulus terlambat.Setelah dilakukan transformasi, keseluruhan data dapat digunakan karena tidak ada nilai yang kosong. Data yang diubah adalah umur (muda : 19 - 24, tua : 25 - 50) dan IPK (besar : 3 - 4, kecil : 1 - 2.9). Hasil confusion matrix, menunjukkan bahwa Naive Bayes mempunyai accuracy 100.00% dan AUC 1.000 lebih tinggi dibandingkan dengan C4.5 dan KNN. Sehingga algoritma Naive Bayes mempunyai kinerja yang lebih baik dibandingkan dengan KNN dan C4.5.

Menurut Saufika Sukmawati dkk, pada tahun 2022, melakukan penelitian tentang perbandingan algoritma C-45 dan Naïve Bayes untuk klasifikasi buruh migran Indonesia. Upaya pemerintah untuk pelindungan

pekerja migran Indonesia dikembangkan sistem komputerisasi tenaga kerja luar negeri oleh Badan Nasional Penempatan dan Perlindungan TKI. Permasalahannya adalah adanya pekerja migran Indonesia (PMI) yang dipulangkan karena permasalahan ketenagakerjaan selama di luar negeri, sehingga dibutuhkan interpretasi pada pola data penempatan PMI yang dapat digunakan memprediksi negara tujuan. Penelitian ini membandingkan dua algoritma klasifikasi yaitu algoritma C 4.5 dan algoritma Naïve Bayes untuk mengetahui pola penempatan PMI dengan menggunakan data penempatan PMI wilayah BP3TKI Semarang. Algoritma Naïve Bayes digunakan untuk mengklasifikasikan data PMI dengan menghitung probabilitas dari data training dan data testing. Algoritma C4.5 digunakan untuk memprediksi dengan mengolah variabel usia, gender, pendidikan, staus_perkawinan, pendidikan, negara_tujuan, status_PMI, sektor_pekerjaan. Percobaan dilakukan dengan data training 1802 dan data testing 772, menghasilkan nilai akurasi paling tinggi bagi kedua algoritma. Algoritma C 4.5 mampu memprediksi lebih baik dengan tingkat akurasi sebesar 84.84% sedangkan Algoritma Naïve Bayes menghasilkan nilai akurasi sebesar 58.29%.

Penelitian selanjutnya dilakukan oleh Jordan Nata Permana, dkk pada tahun 2022 bertujuan untuk mengetahui hasil klasifikasi C4.5 dan Naïve Bayes serta untuk mengetahui keakuratan klasifikasi kedua metode. Variabel yang digunakan dalam penelitian ini adalah status kelulusan (Y), masuk (X_1), jenis kelamin (X_2), asal daerah (X_3), IPK (X_4), dan Kelompok UKT (X_5). Setelah dilakukan analisis, diperoleh hasil rata-rata tingkat akurasi algoritma C4.5 adalah 61,99% dan tingkat akurasi Naïve Bayes sebesar 69,67% . Sehingga dapat dikatakan bahwa Naïve Bayes merupakan metode yang lebih baik dalam mengklasifikasikan ketepatan waktu studi mahasiswa statistika FMIPA Universitas Mulawarman dibandingkan dengan Algoritma C4.5.

Tabel 2. 1 Studi Sebelumnya

NO	Penulis	Tahun	Tujuan	Metode	Hasil
1.	Zakarias Situmorang dkk	2022	Untuk memprediksi minat calon mahasiswa dengan algoritma C45 melalui data preprocessing yang meliputi data selection, data preprocessing/ data cleaning, data transformation dan data reduction	Algoritma C-45	Penulis mengevaluasi bahwa setelah penelitian yang dilakukan maka diperoleh hasil 4 aturan baru dengan menggunakan kriteria jenis kelamin, minat, jurusan asal sekolah dan hobi
2.	Amalia Iftitah dkk	2022	Prediksi ini dilakukan dengan tujuan untuk mempermudah khususnya sub bidang aset dalam melaksanakan tugas. Prediksi tersebut memiliki manfaat dalam pelaksanaan penggantian peralatan dan mesin kantor dapat dikelola dengan baik sesuai dengan kebutuhan karyawan agar kinerja karyawan lebih maksimal	Algoritma C-45	Analisis mengenai prediksi pengusulan peralatan dan mesin kantor menggunakan algoritma C.45 dengan bantuan Aplikasi Rapidminer mampu memberikan hasil yang berupa decision tree yang dapat digunakan oleh anggota subbidang aset dalam mengambil keputusan.
3.	Embun Fajar Wati dkk	2022	Untuk meneliti tingkat kelulusan dan jumlah siswa yang akan mempengaruhi proses	Algoritma C-45, KNN dan Naïve Bayes	Hasil confusion matrix, menunjukkan bahwa Naive Bayes mempunyai accuracy 100.00% dan AUC 1.000 lebih tinggi

			akreditasi bagi perguruan tinggi		dibandingkan dengan C4.5 dan KNN. Sehingga algoritma Naive Bayes mempunyai kinerja yang lebih baik dibandingkan dengan KNN dan C4.5.
4.	Saufika Sukmawati dkk	2022	Penelitian ini bertujuan untuk membandingkan dua algoritma klasifikasi yaitu algoritma C 4.5 dan algoritma Naïve Bayes untuk mengetahui pola penempatan PMI dengan menggunakan data penempatan PMI wilayah BP3TKI Semarang. Algoritma Naïve Bayes digunakan untuk mengklasifikasikan data PMI dengan menghitung probabilitas dari data training dan data testing.	Algoritma C-45 dan Naïve Bayes	Percobaan dilakukan dengan data training 1802 dan data testing 772, menghasilkan nilai akurasi paling tinggi bagi kedua algoritma. Algoritma C 4.5 mampu memprediksi lebih baik dengan tingkat akurasi sebesar 84.84% sedangkan Algoritma Naïve Bayes menghasilkan nilai akurasi sebesar 58.29%.
5.	Jordan Nata Permana, dkk	2022	Bertujuan untuk mengetahui hasil klasifikasi C4.5 dan Naïve Bayes serta untuk mengetahui keakuratan klasifikasi kedua metode dalam memprediksi ketepatan waktu studi mahasiswa.	Algoritma C.45 dan Naïve Bayes	Setelah dilakukan analisis, diperoleh hasil rata-rata tingkat akurasi algoritma C4.5 adalah 61,99% dan tingkat akurasi Naïve Bayes sebesar 69,67% . Sehingga dapat dikatakan bahwa Naïve Bayes merupakan metode yang lebih baik

					dalam mengklasifikasikan ketepatan waktu studi mahasiswa statistika FMIPA Universitas Mulawarman dibandingkan dengan Algoritma C4.5.

2.2. Landasan Teori

2.2.1 Definisi *Machine Learning*

Machine learning dapat didefinisikan sebagai aplikasi komputer dan algoritma matematika yang diadopsi dengan cara pembelajaran yang berasal dari data dan menghasilkan prediksi di masa yang akan datang (Goldberg & Holland, 1988). Adapun proses pembelajaran yang dimaksud adalah suatu usaha dalam memperoleh kecerdasan yang melalui dua tahap antara lain latihan (*training*) dan pengujian (*testing*) (Huang, Zhu, & Siew, 2006).

Pengertian *Machine Learning* menurut Putra (2019) adalah teknik untuk melakukan inferensi (menitikberatkan ranah hubungan variabel) terhadap data dengan pendekatan matematis. Inti *machine learning* adalah untuk membuat model (matematis) yang merefleksikan pola-pola data. *Machine learning* memungkinkan komputer atau suatu program dapat menemukan pengetahuan tanpa diprogram secara eksplisit.

2.2.2 Proses *Machine Learning*

Berikut adalah penjelasan tentang proses *Machine Learning*:

1. Identifikasi Masalah

   Tahap pertama dalam proses *Machine Learning* melibatkan identifikasi masalah yang ingin diselesaikan. Ini melibatkan pemahaman yang mendalam tentang tujuan bisnis atau penelitian serta pengidentifikasian

jenis masalah yang ingin diselesaikan, seperti klasifikasi, *regresi*, *clustering*, atau tugas lainnya.

2. Pengumpulan Data

   Setelah masalah diidentifikasi, langkah selanjutnya adalah mengumpulkan data yang diperlukan untuk memecahkan masalah tersebut. Data dapat diperoleh dari berbagai sumber, termasuk basis data, file CSV, API, atau sumber data lainnya.

3. Pra-Pemrosesan Data

   Data yang dikumpulkan seringkali memerlukan pra-pemrosesan sebelum dapat digunakan secara efektif. Pra-pemrosesan data melibatkan langkah-langkah seperti pembersihan data (misalnya, menghapus entri yang hilang atau tidak valid), normalisasi data (menyamakan skala data), dan pemrosesan lanjutan seperti pengkodean variabel kategorikal.

4. Pemilihan Model

   Setelah data diproses, langkah berikutnya adalah memilih model *Machine Learning* yang paling sesuai dengan masalah yang dihadapi. Pemilihan model bergantung pada jenis masalah yang dihadapi (*klasifikasi, regresi*, dan lain-lain), serta karakteristik data yang tersedia.

5. Pelatihan Model

   Model *Machine Learning* dipelajari dari data yang tersedia melalui proses pelatihan. Selama pelatihan, model disesuaikan dengan data latih untuk menyesuaikan parameter internalnya sehingga dapat membuat prediksi yang akurat.

6. Evaluasi Model

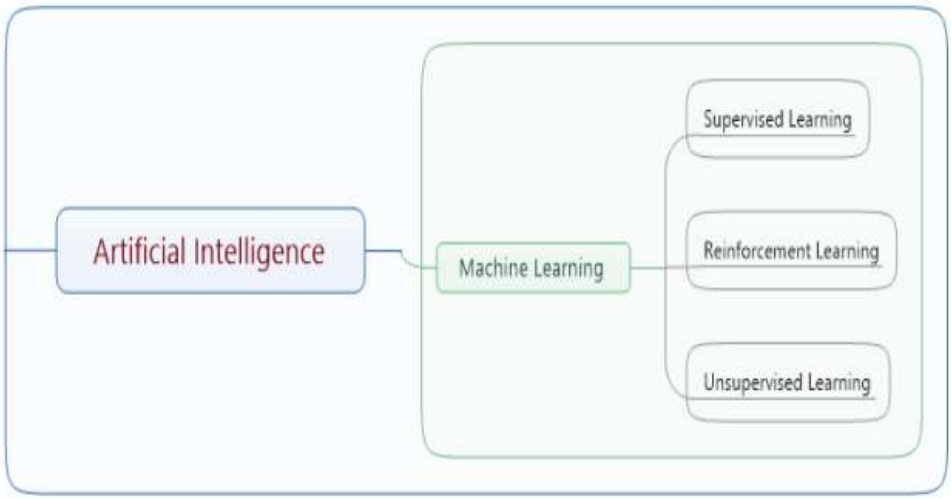
   Setelah pelatihan, model dievaluasi menggunakan data yang terpisah yang tidak digunakan selama pelatihan (data uji). Evaluasi model memungkinkan kita untuk mengukur kinerja model, seperti akurasi, *presisi, recall*, atau metrik lainnya yang sesuai dengan masalah yang dihadapi.

7. Penyetelan Model

   Jika kinerja model tidak memenuhi ekspektasi, langkah terakhir adalah menyetel model dengan melakukan perubahan pada parameter model atau menggunakan teknik yang berbeda untuk meningkatkan kinerja model.

2.2.3 Teknik *Machine Learning*

Penelitian terkini mengungkapkan bahwa *machine learning* terbagi menjadi tiga kategori: *Supervised Learning, Unsupervised Learning, Reinforcement Learning* (Somvanshi & Chavan, 2016). Skema keterkaitan *artificial intelligence* dan *machine learning* dapat dijelaskan dalam Gambar 2.1 berikut ini :



Gambar 2.1. *Skema Artificial Intelligence dan Machine Learning*

Teknik yang digunakan oleh *Supervised Learning* adalah metode klasifikasi di mana kumpulan data sepenuhnya diberikan label untuk mengklasifikasikan kelas yang tidak dikenal.

Selanjutnya Teknik *Unsupervised Learning* sering disebut *cluster* dikarenakan tidak ada kebutuhan untuk pemberian label dalam kumpulan

data dan hasilnya tidak mengidentifikasi contoh di kelas yang telah ditentukan (Thupae, Isong, Gasela, & AbuMahfouz, 2018).

Sedangkan *Reinforcement Learning* biasanya berada antara *Supervised Learning dan Unsupervised Learning* (Board, 2017), teknik ini bekerja dalam lingkungan yang dinamis di mana konsepnya harus menyelesaikan tujuan tanpa adanya pemberitahuan dari komputer secara eksplisit jika tujuan tersebut telah tercapai (Das & Nene, 2017).

Metode *supervised learning* didasarkan pada kumpulan sampel data yang memiliki label. Kumpulan sampel digunakan untuk meringkas karakteristik distribusi ukuran perilaku dalam setiap jenis aplikasi sehingga membentuk model perilaku dari data (Amei, Huailin, Qingfeng, & Ling, 2011). *Supervised learning* dikelompokkan lebih lanjut dalam masalah klasifikasi dan regresi. Masalah klasifikasi adalah ketika variabel output berbentuk kategori, seperti merah atau biru atau penyakit dan tidak ada penyakit. Sedangkan masalah regresi adalah ketika variabel output adalah nilai riil, seperti *dollar* atau berat (Brownlee, 2016).

Supervised *learning* memiliki beberapa algoritma populer seperti *Back-propagation (Negnevitsky, 2005), Linear regression, Random Forest, Support Vector Machines (Brownlee, 2016), Naive Bayesian, Metode Rocchio, Decision Tree, k-Nearest Neighbor, Neural Network (Darujati & Gumelar, 2012), Logistic Regression*, dan *Neural Network* (Lakshmi & Sheshasaayee, 2015). Kemudian beberapa algoritma untuk klasifikasi pun disebutkan dalam seperti *Support Vector Machines (SVM), Normal Bayesian Classifier (NBC), K-Nearest Neighbor (KNN), Trees Gradient Boosted (GBT), Random Trees (RT),* dan *Artificial Neural Networks (ANN)* (Židek, Pitel', & Hošovsk\`y, 2017). Algoritma lainnya pun dibahas dalam (Athmaja, Hanumanthappa, & Kavitha, 2017) seperti *Gaussian Mixture models, Hidden Markov Models, logistic regression, Kernel Regression, Deep neural networks, Deep belief networks, PCA, Kernel Perceptron*.

Beberapa masalah dalam kategori ini berkisar pada klasifikasi misalnya dalam bidang lalu lintas seperti pengembangan *Automatic Plate Plate Recognition (ALPR)* yang dapat digunakan di banyak aplikasi, seperti pemantauan lalu lintas jalan, pembayaran tol otomatis dan lain-lain.

Dalam jenis pembelajaran *Unsupervised Learning*, sistem disediakan dengan beberapa input sampel tetapi tidak ada output yang hadir. Karena tidak ada output yang diinginkan di sini kategorisasi dilakukan sehingga algoritma membedakan dengan benar antara kumpulan data. Ini adalah tugas mendefinisikan fungsi untuk menggambarkan struktur yang tersembunyi dari data yang tidak berlabel (Somvanshi & Chavan, 2016). *Unsupervised learning* dikelompokkan lebih lanjut dalam masalah clustering dan asosiasi.

Masalah pengelompokan (clustering) adalah tempat untuk menemukan pengelompokan yang melekat dalam data, seperti mengelompokkan pelanggan berdasarkan pada perilaku pembelian. Sedangkan masalah asosiasi adalah aturan yang menggambarkan sebagian besar data yang ada, seperti orang yang membeli A juga cenderung membeli B (Brownlee, 2016).

Unsupervised *learning* memiliki beberapa algoritma populer seperti *k-means, Apriori (Brownlee, 2016), Independent Subspace Analysis (ISA) (G. Wu et al., 2013), DBSCAN* (Ester, Kriegel, Sander, Xu, & others, 1996). Beberapa masalah misalnya dalam bidang finansial untuk meninjau sejumlah besar data maka unsupervised learning biasanya dapat digunakan. Dalam bidang industri misalnya dalam, kemudian dalam bidang kedokteran *unsupervised learning* digunakan dalam proses segmentasi pembuluh darah (Dharmawan, Ng, & Rahardja, 2018), dan bidang teknologi seperti jaringan komputer maupun pencegahan serangan keamanannya.

Reinforcement *learning* berasal dari teori belajar hewan. Pembelajaran ini tidak memerlukan pengetahuan sebelumnya, dapat secara mandiri mendapatkan kebijakan opsional dengan pengetahuan yang

diperoleh melalui coba-coba dan terus berinteraksi dengan lingkungan yang dinamis (Qiang & Zhongli, 2011). Masalah *reinforcement learning* diselesaikan dengan mempelajari pengalaman baru melalui *trial-and-error* (Mahmud, Kaiser, Hussain, & Vassanelli, 2018). *Algoritma reinforcement learning* terkait dengan algoritma pemrograman dinamis yang sering digunakan untuk menyelesaikan masalah optimisasi (Mitchell, 1997).

Banyak masalah *machine learning* dalam dunia nyata termasuk dalam kategori ini. Hal ini dikarenakan bisa mencapai harga yang mahal bahkan memakan waktu untuk pemberian label pada data terkait kemungkinan untuk memerlukan akses ke bagian pakar. Padahal data yang tidak memiliki label itu didapatkan dengan harga yang murah dan mudah untuk dikumpulkan dan disimpan (Brownlee, 2016).

2.2.4 Decision Tree

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*.

Decision tree merupakan metode klasifikasi yang paling popular digunakan. Selain karena pembangunannya relatif cepat, hasil dari model yang dibangun mudah untuk dipahami.

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*.

2.2.5 *Algoritma* C4.5

Algoritma C 4.5 adalah salah satu metode untuk membuat *decision tree* berdasarkan *training data* yang telah disediakan. Algoritma C 4.5 merupakan pengembangan dari ID3. Beberapa pengembangan yang dilakukan pada C 4.5 adalah bisa mengatasi *missing value*, bisa mengatasi c*ontinue data*, dan *pruning*.

Pohon keputusan merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami. Dan mereka juga dapat diekspresikan dalam bentuk bahasa basis data seperti *Structured Query Language* untuk mencari *record* pada kategori tertentu. Pohon keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel *input* dengan sebuah variabel target.

Karena pohon keputusan memadukan antara eksplorasi data dan pemodelan, pohon keputusan sangat bagus sebagai langkah awal dalam proses pemodelan bahkan ketika dijadikan sebagai model akhir dari beberapa teknik lain. Sebuah pohon keputusan adalah sebuah struktur yang dapat digunakan untuk membagi kumpulan data yang besar menjadi himpunan-himpunan *record* yang lebih kecil dengan menerapkan serangkaian aturan keputusan. Dengan masingmasing rangkaian pembagian, anggota himpunan hasil menjadi mirip satu dengan yang lain *(Berry dan Linoff, 2004).*

Sebuah model pohon keputusan terdiri dari sekumpulan aturan untuk membagi sejumlah populasi yang heterogen menjadi lebih kecil, lebih homogen dengan memperhatikan pada variabel tujuannya. Sebuah pohon keputusan mungkin dibangun dengan seksama secara manual atau dapat tumbuh secara otomatis dengan menerapkan salah satu atau beberapa algoritma pohon keputusan untuk memodelkan himpunan data yang belum terklasifikasi.

Variabel tujuan biasanya dikelompokkan dengan pasti dan model pohon keputusan lebih mengarah pada perhitungan *probability* dari tiap-tiap *record* terhadap kategori-kategori tersebut atau untuk mengklasifikasi *record* dengan mengelompokkannya dalam satu kelas. Pohon keputusan juga dapat digunakan untuk mengestimasi nilai dari variabel *continue* meskipun ada beberapa teknik yang lebih sesuai untuk kasus ini.

Banyak algoritma yang dapat dipakai dalam pembentukan pohon keputusan,antara lain ID3, CART, dan C4.5 *(Larose, 2006).*

Data dalam pohon keputusan biasanya dinyatakan dalam bentuk tabel dengan atribut dan *record.* Atribut menyatakan suatu parameter yang dibuat sebagai kriteria dalam pembentukan pohon. Misalkan untuk menentukan main tenis, kriteria yang diperhatikan adalah cuaca, angin, dan temperatur.

Salah satu atribut merupakan atribut yang menyatakan data solusi per *item* data yang disebut target atribut. Atribut memiliki nilai-nilai yang dinamakan dengan *instance.* Misalkan atribut cuaca mempunyai *instance* berupa cerah, berawan, dan hujan *(Basuki dan Syarif, 2003).*

Proses pada pohon keputusan adalah mengubah bentuk data (tabel) menjadi model pohon, mengubah model pohon menjadi *rule*, dan menyederhanakan *rule (Basuki dan Syarif, 2003)*. Berikut ini algoritma dasar dari C4.5:

Input : sampel *training*, label *training*, atribut

1. Membuat simpul akar untuk pohon yang dibuat
2. Jika semua sampel positif, berhenti dengan suatu pohon dengan satu simpul akar, beri tanda (+)
3. Jika semua sampel negatif, berhenti dengan suatu pohon dengan satu simpul akar, beri tanda (-)
4. Jika atribut kosong, berhenti dengan suatu pohon dengan suatu simpul akar, dengan label sesuai nilai yang terbanyak yang ada pada label training
5. Untuk yang lain, Mulai

a. A ------ atribut yang mengklasifikasikan sampel dengan hasil terbaik (berdasarkan *Gain* rasio)
b. Atribut keputusan untuk simpul akar ----- A
c. Untuk setiap nilai, *vi*, yang mungkin untuk A
    1) Tambahkan cabang di bawah akar yang berhubungan dengan A= *vi*
    2) Tentukan sampel Svi sebagai subset dari sampel yang mempunyai nilai vi untuk atrribut A
    3) Jika sampel Svi kosong
        i. Di bawah cabang tambahkan simpul daun dengan label = nilai yang terbanyak yang ada pada label training
        ii. Yang lain tambah cabang baru di bawah cabang yang sekarang C4.5 (sampel *training*, label *training*, atribut-[A])
d. Berhenti

Mengubah tree yang dihasilkan dalam beberapa *rule*. Jumlah *rule* sama dengan jumlah *path* yang mungkin dapat dibangun dari *root* sampai *leaf node*.

Tree Pruning dilakukan untuk menyederhanakan *tree* sehingga akurasi dapat bertambah. *Pruning* ada dua pendekatan, yaitu:

a. *Pre-pruning*, yaitu menghentikan pembangunan suatu *subtree* lebih awal (yaitu dengan memutuskan untuk tidak lebih jauh mempartisi data training). Saat seketika berhenti, maka *node* berubah menjadi *leaf* (*node* akhir). *Node* akhir ini menjadi kelas yang paling sering muncul di antara *subset* sampel.
b. *Post-pruning*, yaitu menyederhanakan *tree* dengan cara membuang beberapa cabang *subtree* setelah *tree* selesai dibangun. *Node* yang jarang dipotong akan menjadi *leaf* (*node* akhir) dengan kelas yang paling sering muncul.

Untuk memudahkan penjelasan mengenai algoritma C 4.5 berikut ini disertakan contoh kasus yang dituangkan dalam Tabel 2.2 berikut ini :

Tabel 2.2. Keputusan Bermain Tenis

No	CUACA	TEMPERATUR	KELEMBABAN	ANGIN	BERMAIN
1	Cerah	Panas	Tinggi	Tidak	Tidak
2	Cerah	Panas	Tinggi	Ya	Tidak
3	Mendung	Panas	Tinggi	Tidak	Ya
4	Hujan	Sedang	Tinggi	Tidak	Ya
5	Hujan	Dingin	Normal	Tidak	Ya
6	Hujan	Dingin	Normal	Ya	Ya
7	Mendung	Dingin	Normal	Ya	Ya
8	Cerah	Sedang	Tinggi	Tidak	Ya
9	Cerah	Dingin	Normal	Tidak	Tidak
10	Hujan	Sedang	Normal	Tidak	Ya
11	Cerah	Sedang	Normal	Ya	Ya
12	Mendung	Sedang	Tinggi	Ya	Ya
13	Mendung	Panas	Normal	Tidak	Ya
14	Hujan	Sedang	Tinggi	Ya	Tidak

Dalam kasus yang tertera pada Tabel 2.2 akan dibuat pohon keputusan untuk menentukan main tenis atau tidak dengan melihat keadaan cuaca, temperatur, kelembaban dan keadaan angin.

Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut:

1. Pilih atribut sebagai akar
2. Buat cabang untuk masing-masing nilai
3. Bagi kasus dalam cabang
4. Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *Gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *Gain* digunakan rumus seperti tertera dalam Rumus 1 *(Craw, 2005)*.

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

Keterangan:
S = himpunan kasus
A = fitur
n = jumlah partisi atribut A
|Si| = proporsi Si terhadap S
|S| = jumlah kasus dalam S

Sedangkan perhitungan nilai *Entropy* dapat dilihat pada rumus 2 berikut *(Craw, 2005)*:

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

Dengan :

S = himpunan kasus

n = jumlah partisi S

pi = proporsi dari *Si* terhadap S

Berikut ini adalah penjelasan lebih rinci mengenai masing-masing langkah dalam pembentukan pohon keputusan dengan menggunakan algoritma C4.5 untuk menyelesaikan permasalahan pada Tabel 2.2

1. Menghitung jumlah kasus, jumlah kasus untuk keputusan Ya, jumlah kasus untuk keputusan Tidak, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut cuaca, temperatur, kelembaban dan angin. Setelah itu lakukan penghitungan *Gain* untuk masing-masing atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.3 berikut ini :

Tabel 2. 3 ***Perhitungan Node 1***

Node			Jumlah Kasus (S)	Tidak (S1)	Ya (S2)	*Entropy*	*Gain*
1	TOTAL		14	4	10	0.863120569	
	CUACA						0.258521037
		MENDUNG	4	0	4		
		HUJAN	5	1	4	0.721928095	
		CERAH	5	3	2	0.970950594	
	TEMPERATUR						0.183850925
		DINGIN	4	0	4	0	
		PANAS	4	2	2	1	
		SEDANG	6	2	4	0.918295834	
	KELEMBABAN						0.370506501
		TINGGI	7	4	3	0.985228136	
		NORMAL	7	0	7	0	
	ANGIN						0.005977711
		TIDAK	8	2	6	0.811278124	
		YA	6	4	2	0.918295834	

Baris total kolom *Entropy* pada Tabel 2.3 dihitung dengan rumus 2, sebagai berikut:

$$Entropy(Total) = (-\frac{4}{14} * Log2(\frac{4}{14})) + (-\frac{10}{14} * Log2(\frac{10}{14}))$$

$$Entropy(Total) = 0.863120569$$

Sementara itu nilai *Gain* pada baris cuaca dihitung dengan menggunakan rumus 1, sebagai berikut :

$$Gain(Total, Cuaca)$$

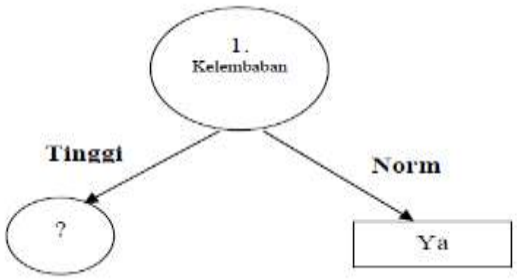
$$= Entropy(Total) - \sum_{i=1}^{n} \left(\frac{|Cuaca|}{|Total|} \right) * Entropy\ Cuaca$$

$$Gain(Total, Cuaca) = Entropy(Total) - ((\frac{4}{14*0}) + ((\frac{5}{14*0,723}) + ((\frac{5}{14*0,97}))$$

$$Gain(Total, Cuaca) = 0,23$$

Dari hasil pada Tabel 2.3 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah kelembaban yaitu sebesar 0.37. Dengan demikian kelembaban dapat menjadi *node* akar. Ada 2 nilai atribut dari kelembaban yaitu tinggi dan normal. Dari kedua nilai atribut tersebut, nilai atribut normal sudah mengklasifikasikan kasus menjadi 1 yaitu keputusannya Ya, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut tinggi masih perlu dilakukan perhitungan lagi.

Dari hasil tersebut dapat digambarkan pohon keputusan sementara, tampak seperti Gambar 2.2 berikut ini :



Gambar 2. 2 Pohon Keputusan Hasil Perhitungan Node 1

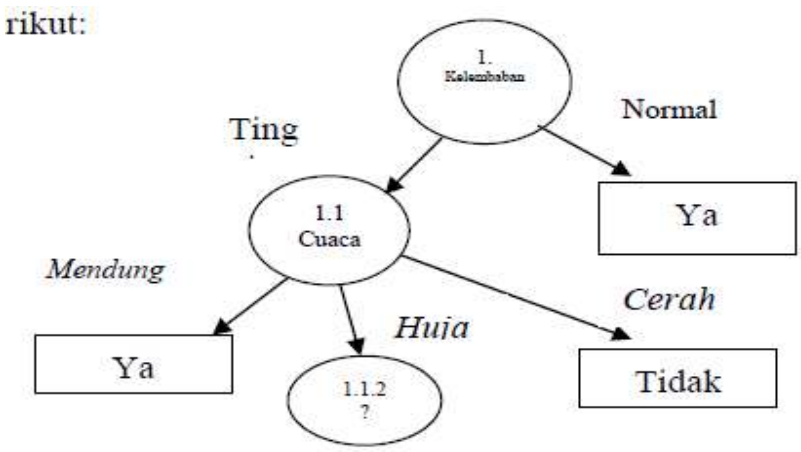
2. Menghitung jumlah kasus, jumlah kasus untuk keputusan Ya, jumlah kasus untuk keputusan Tidak, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut cuaca, temperatur dan angin yang dapat menjadi node akar dari nilai atribut tinggi. Setelah itu lakukan penghitungan *Gain* untuk masingmasing atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.4 berikut ini :

Tabel 2. 4 *Perhitungan Node 1.1*

Node			Jumlah Kasus (S)	Tidak (S1)	Ya (S2)	Entropy	Gain
1.1	KELEMBABAN-TINGGI		7	4	3	0.985228136	
	CUACA						0.69951385
		MENDUNG	2	0	2	0	
		HUJAN	2	1	1	1	
		CERAH	2	3	0	0	
	TEMPERATUR						0.020244207
		DINGIN	0	0	0	0	
		PANAS	3	2	1	0.918295834	
		SEDANG	4	2	2	1	
	ANGIN						0.020244207
		TIDAK	4	2	2	1	
		YA	3	4	1	0.918295834	

Dari hasil pada Tabel 2.4 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah cuaca yaitu sebesar 0.699. Dengan demikian cuaca dapat menjadi *node* cabang dari nilai atribut tinggi. Ada 3 nilai atribut dari cuaca yaitu mendung, hujan dan cerah. dari ketiga nilai atribut tersebut, nilai atribut mendung sudah mengklasifikasikan kasus menjadi 1 yaitu keputusannya Ya dan nilai atribut cerah sudah mengklasifikasikan kasus menjadi satu dengan keputusan Tidak, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut hujan masih perlu dilakukan perhitungan lagi.

Pohon keputusan yang terbentuk sampai tahap ini ditunjukkan pada Gambar 2.3 berikut:



Gambar 2. 3 Pohon Keputusan Hasil Perhitungan Node 1.1

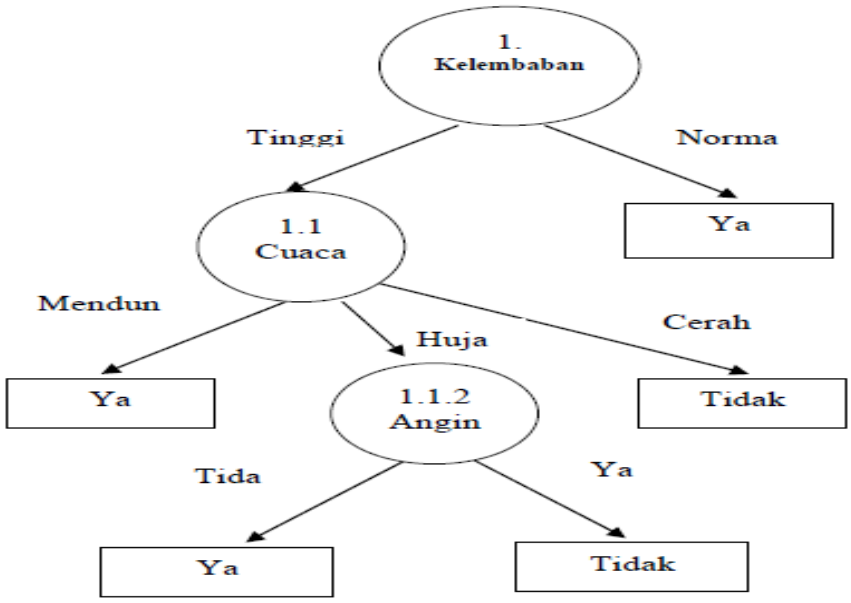
3. Menghitung jumlah kasus, jumlah kasus untuk keputusan Ya, jumlah kasus untuk keputusan Tidak, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut temperatur dan angin yang dapat menjadi *node* cabang dari nilai atribut hujan. Setelah itu lakukan penghitungan *Gain* untuk masingmasing atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.5 berikut ini :

Tabel 2. 5 *Perhitungan Node 1.1.2*

Node			Jumlah Kasus (S)	Tidak (S1)	Ya (S2)	*Entropy*	*Gain*
1.1	KELEMBABAN-TINGGI dan CUACA – HUJAN		2	1	1	1	
	TEMPERATUR						0
		DINGIN	0	0	0	0	
		PANAS	0	0	0	0	
		SEDANG	2	1	1	1	
	ANGIN						1
		TIDAK	1	0	1	0	
		YA	1	1	0	0	

Dari hasil pada Tabel 2.5 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah angin yaitu sebesar 1. Dengan demikian angin dapat menjadi *node* cabang dari nilai atribut hujan. Ada 2 nilai atribut dari angin yaitu Tidak dan Ya. Dari kedua nilai atribut tersebut, nilai atribut Tidak sudah mengklasifikasikan kasus menjadi 1 yaitu keputusannya Ya dan nilai atribut Ya sudah mengklasifikasikan kasus

menjadi satu dengan keputusan Tidak, sehingga tidak perlu dilakukan perhitungan lebih lanjut untuk nilai atribut ini. Pohon keputusan yang terbentuk sampai tahap ini ditunjukkan pada Gambar 2.3 berikut ini :



Gambar 2. 3 Pohon Keputusan Hasil Perhitungan Node 1.1.2

Dengan memperhatikan pohon keputusan pada Gambar 2.5 diketahui bahwa semua kasus sudah masuk dalam kelas. Dengan demikian, pohon keputusan pada Gambar 2.5 merupakan pohon keputusan terakhir yang terbentuk.

2.2.5.1 Entropy

Entropy adalah ukuran dari teori informasi yang dapat mengetahui karakteristik dari *impury*, dan *homogenity* dari kumpulan data. Sebuah obyek yang diklasifikasikan dalam pohon harus dites nilai entropinya. Dari nilai *entropy* tersebut kemudian dihitung nilai *information gain* (IG) masing-masing atribut. Pemillihan atribut pada ID3 dilakukan dengan properti statistik, yang disebut dengan *information gain*. Dengan tujuan untuk mendefinisikan *gain*, pertama-tama digunakanlah ide dari teori informasi yang disebut entropi. Entropi mengukur jumlah dari informasi yang ada pada atribut. Rumus menghitung entropi informasi adalah:

$$\text{Entropy}(S) = -p_+ \log p_+ - p_- \log_2 p_-$$

Dimana:

S = ruang (data) *sample* yang digunakan untuk *training*.

P+ = adalah jumlah yang bersolusi positif (mendukung) pada data *sample* untuk kriteria tertentu.

P- = adalah jumlah yang bersolusi negatif (tidak mendukung) pada data *sample* untuk kriteria tertentu.

2.2.5.2 Information Gain

Information Gain adalah ukuran efektivitas suatu atribut dalam mengklasifikasikan data. *Gain* digunakan untuk mengukur seberapa baik suatu atribut memisahkan *training example* ke dalam kelas target. Atribut dengan informasi tertinggi akan dipilih.

2.2.6 Algoritma Naïve Bayes Classifier (NBC)

Algoritma Naïve Bayes akan mengevaluasi setiap atribut yang mengkontribusi prediksi pada atribut target. *Naïve Bayes* tidak memperhitungkan relasi antar atribut-atribut kontributor prediksi, tidak seperti *Decision Tree* yang memperhitungkan relasi antara atribut. Bentuk tugas dasar yang dilakukan oleh algoritma *Naïve Bayes* adalah hanyalah klasifikasi (*ZhaoHui & MacLennan, 2005, p.132*). Naïve Bayes merupakan teknik *data mining* dengan pendekatan teori probabilitas untuk membangun sebuah model klasifikasi berdasarkan pada kejadian masa lalu yang mempunyai potensi membentuk sebuah objek baru yang dikategorikan sebagai kelas yang memiliki probabilitas terbaik (*Turban et all, 2011, p.220*).

Naïve Bayes memiliki kemampuan yang cepat dalam membuat model, mempunyai kemampuan memprediksi dan juga menyediakan metode baru dalam mengeksplor dan memahami data. Algoritma *Naïve Bayes* hanya mendukung pada atribut yang bertipe data *discrete* atau

discretized, atau tidak mendukung atribut yang bernilai *continuous* (numerik) dan semua atribut dapat menjadi independen, menjadi atribut yang memberi kontribusi kepada atribut yang diprediksi.

Bayesian classification adalah pengklasifikasian statistik yang dapat *digunakan* untuk memprediski probabilitas keanggotaan suatu *class*. *Bayesian classification* didasarkan pada teorema Bayes yang memiliki kemampuan klasifikasi serupa dengan *decision tree* dan *neural network*. *Bayesian classification* terbukti memiliki akurasai dan kecepatan yang tinggi saat diaplikasikan ke dalam *database* dengan data yang besar (Kusrini,2009).

Teorema Bayes memiliki bentuk umum sebagai berikut:

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

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 prob.)

P(H) = Probabilitas hipotesis H (prior prob.)

P(X|H) = Probabilitas X berdasarkan kondisi tersebut

P(X) = Probabilitas dari X

Berdasarkan rumus di atas kejadian H merepresentasikan sebuah kelas dan X merepresentasikan sebuah atribut. P(H) disebut *prior probability* H, contoh dalam kasus ini adalah probabilitas kelas yang mendeklarasikan normal. P(X) merupakan prior probability X, contoh untuk probabilitas sebuah atribut protocol_type. P(H|X) adalah *posterior probability* yang merefleksikan probabilitas munculnya kelas normal terhadap data atribut protocol_type. P(X|H) menunjukkan kemungkinan munculnya prediktor X (protocol_type) pada kelas normal. Dan begitu

juga seterusnya untuk proses menghitung probabilitas keempat kelas lainnya.

Sebagai contoh kasus *Naïve Bayes* seperti pada Tabel 2.6. Bertujuan menemukan pola yang digunakan dalam mendeteksi permohonan kredit yang beresiko tinggi.

Tabel 2. 6 *Contoh Data Set Naïve Bayes*

Name	*Debt*	*Income*	*Married?*	*Risk*
Joe	*High*	*High*	*Yes*	*Good*
Sue	*Low*	*High*	*Yes*	*Good*
John	*Low*	*High*	*No*	*Poor*
Mary	*High*	*Low*	*Yes*	*Poor*
Fred	*Low*	*Low*	*Yes*	*Poor*

1. Membuat model berdasarkan pada kasus

Tabel 2. 7 *Tabel Perhitungan Contoh Kasus Naïve Bayes*

		Counts	*Counts*	*Probabilities*	*Probabilities*
Independent Variables	*Value*	*Good Risk*	*Poor Risk*	*Good Risk*	*Poor Risk*
Debt	*High*	1	1	0.50	0.33
Debt	*Low*	1	2	0.50	0.67
Income	*High*	2	1	1.00	0.33
Income	*Low*	0	2	0.00	0.67
Married	*Yes*	2	2	1.00	0.67
Married	*No*	0	1	0.00	0.33
Total by Risk		2	3		

- Hitunglah *Counts* berdasarkan jumlah data
- Hitunglah pula *Total by Risk* berdasarkan *data set*
- Hitung *Probabilities* = *Counts* / *Total by Risk*

Cara membaca: Peluang *Good Risk Customer* jika diketahui *Debt*-nya *High* adalah 0.5=50% 2.

2. Prediksi resiko berdasarkan model yang telah dibuat

Tabel 2. 8 *Tabel Hasil Contoh Kasus Naïve Bayes*

Name	*Debt*	*Income*	*Married?*	*Risk Actual*	*Good Risk Score*	*Poor Risk Score*	*Risk Predicted*
Joe	*High*	*High*	*Yes*	*Good*	0.200	0.044	*Good*
Sue	*Low*	*High*	*Yes*	*Good*	0.077	0.034	*Good*
John	*Low*	*High*	*No*	*Poor*	0	0.086	*Poor*
Mary	*High*	*Low*	*Yes*	*Poor*	0	0.096	*Poor*
Fred	*Low*	*Low*	*Yes*	*Poor*	0	0.137	*Poor*

- *Score* = (*Total by Risk* / *Total Record*) * *Probabilities of Debt* * *Probabilities of Income* * *Probabilities of Married*
- Jika *Good Risk Score* > *Poor Risk Score* maka *Risk Predicted* = *Good*, dan sebaliknya jika *Good Risk Score* < *Poor Risk Score maka Risk Predicted* = *Bad*

2.2.7 RapidMiner

2.2.7.1 Sejarah *RapidMiner*

RapidMiner sebelumnya dikenal sebagai YALE (*Yet Another Learning Environment*), dikembangkan mulai tahun 2001 oleh *Ralf Klinkenberg*, *Ingo Mierswa*, dan *Simon Fischer* di Unit *Artificial Intelligence* dari *Technical University of Dortmund*. Mulai tahun 2006, perkembangannya didorong oleh *I-Fast*, sebuah perusahaan yang didirikan oleh *Ingo Mierswa* dan *Ralf Klinkeberg* pada tahun yang sama. Pada tahun 2007, nama software YALE diubah menjadi RapidMiner dan mendirikan perusahaan *I-Fast GmbH*. Pada akhir Mei, *free open-source suite data mining* YALE berganti nama menjadi *RapidMiner*. Dalam versi tersebut, terdapat semua fungsi yang ada pada YALE dan menambahkan banyak fungsi baru termasuk antar muka pengguna pun sepenuhnya direvisi. Perbaikan dari YALE ke *Rapid Miner* dilakukan agar lebih berguna untuk analisis pekerjaan sehari-hari.

RapidMiner dan *plugin* yang sekarang menyediakan lebih dari 400 *learning* dan *preprocessing operator* serta kombinasi yang tak terhitung jumlahnya. Oleh karena itu, *RapidMiner* adalah pelengkap pengetahuan penemuan suite yang dapat digunakan untuk semua tugas *machine learning*. Di antara fitur baru tersebut yaitu adanya ruang kerja untuk proyek yang berbeda dengan meningkatkan visualisasi dari kriteria kinerja seperti *ROC curve* atau plot 3D dari matriks.

2.2.7.2 Pengertian *RapidMiner*

RapidMiner adalah aplikasi *machine learning open-source* yang terkemuka dan ternama di dunia. Dirancang sebagai aplikasi yang berdiri sendiri untuk analisis data dan sebagai mesin pengolah *machine learning* untuk diintegrasikan ke dalam produk sendiri. Ribuan aplikasi RapidMiner di lebih dari 40 negara memberikan banyak manfaat bagi penggunanya, antara lain : Integrasi data, Analitis ETL, Data Analisis, dan Pelaporan dalam suatu suite tunggal.

RapidMiner merupakan sebuah lingkungan untuk *machine learning*, *data mining*, *text mining* dan *predictive analytics*.

- *Machine learning* : Algoritma di mana perilaku komputer ber-evolusi berdasarkan data empiris, seperti sensor atau database.
- *Data mining* : Proses mengekstrak pola-pola dari data set yang besar dengan mengombinasikan metoda statistika, kecerdasan buatan dan *database*.
- *Text mining* : Mirip dengan *text analytics*, yaitu proses untuk mendapatkan informasi bermutu tinggi dari teks.
- *Predictive analytics* : Teknik-teknik statistika yang menganalisa fakta masa kini dan masa lalu untuk memprediksi kejadian di masa depan.

RapidMiner merupakan aplikasi *open source* berlisensi AGPL (*GNU Affero General Public License*) versi 3. RapidMiner pernah meraih peringkat satu sebagai *tool data mining* untuk proyek nyata pada

poll oleh *KDnuggets*, sebuah koran data-mining, pada 2010-2011. RapidMiner menyediakan prosedur *data mining* dan *machine learning* termasuk: ETL (*extraction*, *transformation*, *loading*), data *preprocessing*, visualisasi, *modelling* dan evaluasi. Proses *data mining* tersusun atas operator-operator yang *nestable*, dideskripsikan dengan XML, dan dibuat dengan GUI. RapidMiner ditulis dalam bahasa pemrograman *Java* yang mengintegrasikan proyek *data mining* Weka dan statistika R.

2.2.7.3 Terminologi dasar *RapidMiner*

1. Atribut dan atribut target
    a. Atribut: karakteristik atau fitur dari data yang menggambarkan sebuah proses atau situasi. Terdapat dua macam atribut yaitu ID dan atribut biasa.
    b. Atribut target: atribut yang menjadi tujuan untuk diisi oleh proses *data mining*. Terdapat 3 atribut target yaitu *label*, *cluster*, *weight*.
2. Peran atribut (*attribute role*), yang termasuk dalam peran atribut yaitu *label*, *cluster*, *weight*, ID, biasa.
3. Tipe nilai (*value type*), berikut ini merupakan beberapa tipe nilai :
    a. *Nominal* : nilai secara kategori
    b. *Numeric* : nilai numerik secara umum
    c. *Integer* : bilangan bulat
    d. *Real* : bilangan nyata
    e. *Text* : teks bebas tanpa struktur
    f. *Binominal* : nominal dua nilai
    g. *Polynominal* : nominal lebih dari dua nilai
    h. *date_time* : tanggal dan waktu
    i. *date* : hanya tanggal
    j. *time* : hanya waktu

4. *Data* dan *metadata*
    a. *Data* menyebutkan obyek-obyek dari sebuah konsep yang ditunjukkan sebagai baris dari tabel.
    b. *Metadata* menggambarkan karakteristik dari konsep tersebut yang ditunjukkan sebagai kolom dari tabel.
5. *Modelling*
    a. Penggunaan metode *data mining* terhadap data.
    b. Hasilnya disebut model.
    c.

2.2.7.4 Desain proses analisa dalam *RapidMiner*

1. Fleksibilitas dan fungsionalitas
    a. Sangat fleksibel untuk mendefinisikan proses analisa secara visual dengan GUI.
    b. Meliputi lebih dari 500 fungsionalitas *data mining* dalam bentuk operator-operator.
2. Skalabilitas
    a. Mulai versi 4.6 ~ .. fokus utama pada skalabilitas untuk data ukuran besar.
    b. Konsep *view* untuk data mirip seperti *database*.
    c. Transformasi data *on-the-fly* tanpa *copy*.
    d. 100 juta data set bukanlah data yang besar.
3. Format data
    a. Terhubung sangat baik dengan berbagai sumber data: *Oracle, IBM DB2, Microsoft SQL Server, MySQL, PostgreSQL, Ingres, Excel, Access, SPSS, CSV files* dan berbagai format lain.
    b. Bersama-sama dengan operator-operator untuk data *preprocessing*, bisa digunakan juga sebagai *tool* ETL (*extraction*, *transformation*, *loading*) dengan hasil yang menakjubkan.

2.2.7.5 Repositori Pertama

Pada saat menjalankan *RapidMiner* untuk pertama kali, akan menanyakan pembuatan repositori baru. Repositori ini berfungsi sebagai lokasi penyimpanan terpusat untuk data dan proses analisa kita.

2.2.7.6 Perspektif dan *View*

1. Sebuah perspektif berisi pilihan elemen-elemen GUI, yang disebut *view*, yang dapat dikonfigurasi secara bebas. Elemen-elemen ini dapat diatur bagaimanapun juga sesuka kita.
2. Tiga perspektif:
   a. Perspektif selamat datang (*welcome perspective*).
   b. Perspektif desain (*design perspective*).
   c. Perspektif hasil (*result perspective*).

2.2.7.7 Perspektif *Desain*

1. Perspektif pusat di mana semua proses analisa dibuat dan dimanage.
2. Pindah ke perspektif *desain* dengan cara klik tombol paling kiri atau gunakan menu *View* → *Perspectives* → *Design*.
3. *View*, terdapat beberapa *view* di dalam RapidMiner yaitu *Operators*, *Repositories*, *Process*, *Parameters*, *Help*, *Comment*, *Overview*, *Problems*, *Log*
   a. *View Operator*. Semua tahapan kerja (operator) ditampilkan di sini secara berkelompok, dan bisa diikutsertakan di dalam proses analisa.
      - *Process control* : untuk mengontrol aliran proses, seperti *loop* atau *conditional branch*.
      - *Utility* : untuk mengelompokkan *subprocess*, juga *macro* dan *logger*.
      - *Repository Access* : untuk membaca dan menulis repositori.

- *Import* : untuk membaca data dari berbagai format eksternal.
- *Export* : untuk menulis data ke berbagai format eksternal.
- *Data Transformation* : untuk transformasi data dan metadata.
- *Modelling* : untuk proses *data mining* yang sesungguhnya seperti klasifikasi, regresi, clustering, aturan asosiasi dll.
- *Evaluation* : untuk menghitung kualitas dari modelling.

b. *View Repositori*. Komponen pusat yang menyediakan servis untuk manajemen dan pen-strukturan proses analisa, baik data, metadata, proses maupun hasil.

c. *View Proses*. Untuk menampilkan tahap-tahap individual operator di dalam proses analisa dan juga interkoneksi di antara mereka.

d. *View Parameter*. Operator-operator mungkin memerlukan parameter untuk bisa berfungsi. Setelah sebuah operator dipilih di *view proses*, parameternya ditampilkan di *view* ini.

e. *View Help* dan *Comment*.
   - *View Help* menampilkan deskripsi dari operator.
   - *View Comment* menampilkan komentar yang dapat diedit terhadap operator.

f. *View Overview*. Menampilkan seluruh area kerja dan menyorot seksi yang ditampilkan saat ini dengan sebuah kotak kecil.

g. *View Problem*. Menampilkan setiap pesan *warning* dan *error*.

h. *View Log*. Menampilkan pesan log selama melakukan desain dan eksekusi proses.

2.2.7.8 Operator dan Proses

Proses *machine learning* pada dasarnya adalah mendefinisikan proses analisa dengan menyatakan urutan tahap kerja individual. Komponen dari proses ini disebut operator yang didefinisikan dengan:

1. Deskripsi *input*.
2. Deskripsi *output*.
3. Aksi yang dilakukan.
4. Parameter yang diperlukan.

Sebuah operator bisa disambungkan melalui *port* masukan (kiri) dan *port* keluaran (kanan). Indikator status dari operator:

1. Lampu status : merah (tak tersambung), kuning (lengkap tetapi belum dijalankan), hijau (sudah behasil dijalankan).
2. Segitiga *warning* : bila ada pesan status.
3. *Breakpoint* : bila ada *breakpoint* sebelum/sesudahnya.
4. *Comment* : bila ada komentar.
5. *Subprocess* : bila mempunyai *subprocess*.

2.2.8 Status Gizi

2.2.8.1 Pengertian Status Gizi

Status gizi adalah suatu ukuran mengenai kondisi tubuh seseorang yang dapat dilihat dari makanan yang dikonsumsi dan penggunaan zat-zat gizi di dalam tubuh. Status gizi dibagi menjadi tiga kategori, yaitu status gizi kurang, gizi normal, dan gizi lebih *(Almatsier, 2005)*.

Status gizi normal merupakan suatu ukuran status gizi dimana terdapat keseimbangan antara jumlah energi yang masuk ke dalam tubuh dan energi yang dikeluarkan dari luar tubuh sesuai dengan kebutuhan individu. Energi yang masuk ke dalam tubuh dapat berasal dari karbohidrat, protein, lemak dan zat gizi lainnya *(Nix, 2005)*. Status gizi normal merupakan keadaan yang sangat diinginkan oleh semua orang *(Apriadji, 1986)*.

Status gizi kurang atau yang lebih sering disebut *undernutrition* merupakan keadaan gizi seseorang dimana jumlah energi yang masuk lebih sedikit dari energi yang dikeluarkan. Hal ini dapat terjadi karena

jumlah energi yang masuk lebih sedikit dari anjuran kebutuhan individu *(Wardlaw, 2007)*.

Status gizi lebih (*overnutrition*) merupakan keadaan gizi seseorang dimana jumlah energi yang masuk ke dalam tubuh lebih besar dari jumlah energi yang dikeluarkan *(Nix, 2005)*. Hal ini terjadi karena jumlah energi yang masuk melebihi kecukupan energi yang dianjurkan untuk seseorang, akhirnya kelebihan zat gizi disimpan dalam bentuk lemak yang dapat mengakibatkan seseorang menjadi gemuk *(Apriadji, 1986)*.

2.2.8.2 Penilaian Status Gizi

Penilaian status gizi merupakan penjelasan yang berasal dari data yang diperoleh dengan menggunakan berbagai macam cara untuk menemukan suatu populasi atau individu yang memiliki risiko status gizi kurang maupun gizi lebih *(Hartriyanti dan Triyanti, 2007)*. Penilaian status gizi terdiri dari dua jenis, yaitu:

1. Penilaian Langsung
   a. Antropometri

      Antropometri merupakan salah satu cara penilaian status gizi yang berhubungan dengan ukuran tubuh yang disesuaikan dengan umur dan tingkat gizi seseorang. Pada umumnya antropometri mengukur dimensi dan komposisi tubuh seseorang *(Supariasa, 2001)*. Metode antropometri sangat berguna untuk melihat ketidakseimbangan energi dan protein. Akan tetapi, antropometri tidak dapat digunakan untuk mengidentifikasi zat-zat gizi yang spesifik *(Gibson, 2005)*.

   b. Klinis

      Pemeriksaan klinis merupakan cara penilaian status gizi berdasarkan perubahan yang terjadi yang berhubungan erat dengan kekurangan maupun kelebihan asupan zat gizi. Pemeriksaan klinis dapat dilihat pada jaringan epitel yang

terdapat di mata, kulit, rambut, mukosa mulut, dan organ yang dekat dengan permukaan tubuh (kelenjar tiroid) *(Hartriyanti dan Triyanti, 2007)*.

c. Biokimia

Pemeriksaan biokimia disebut juga cara laboratorium. Pemeriksaan biokimia pemeriksaan yang digunakan untuk mendeteksi adanya defisiensi zat gizi pada kasus yang lebih parah lagi, dimana dilakukan pemeriksaan dalam suatu bahan biopsi sehingga dapat diketahui kadar zat gizi atau adanya simpanan di jaringan yang paling sensitif terhadap deplesi, uji ini disebut uji biokimia statis. Cara lain adalah dengan menggunakan uji gangguan fungsional yang berfungsi untuk mengukur besarnya konsekuensi fungsional daru suatu zat gizi yang spesifik Untuk pemeriksaan biokimia sebaiknya digunakan perpaduan antara uji biokimia statis dan uji gangguan fungsional *(Baliwati, 2004)*.

d. Biofisik

Pemeriksaan biofisik merupakan salah satu penilaian status gizi dengan melihat kemampuan fungsi jaringan dan melihat perubahan struktur jaringan yang dapat digunakan dalam keadaan tertentu, seperti kejadian buta senja *(Supariasa, 2001)*.

2. Penilaian Tidak Langsung

a. Survei Konsumsi Makanan

Survei konsumsi makanan merupakan salah satu penilaian status gizi dengan melihat jumlah dan jenis makanan yang dikonsumsi oleh individu maupun keluarga. Data yang didapat dapat berupa data kuantitatif maupun kualitatif. Data kuantitatif dapat mengetahui jumlah dan jenis pangan yang dikonsumsi, sedangkan data kualitatif dapat diketahui frekuensi makan dan cara seseorang maupun keluarga dalam

memperoleh pangan sesuai dengan kebutuhan gizi *(Baliwati, 2004)*.

b. Statistik Vital

Statistik vital merupakan salah satu metode penilaian status gizi melalui data-data mengenai statistik kesehatan yang berhubungan dengan gizi, seperti angka kematian menurut umur tertentu, angka penyebab kesakitan dan kematian, statistik pelayanan kesehatan, dan angka penyakit infeksi yang berkaitan dengan kekurangan gizi *(Hartriyanti dan Triyanti, 2007)*.

c. Faktor Ekologi

Penilaian status gizi dengan menggunakan faktor ekologi karena masalah gizi dapat terjadi karena interaksi beberapa faktor ekologi, seperti faktor biologis, faktor fisik, dan lingkungan budaya. Penilaian berdasarkan faktor ekologi digunakan untuk mengetahui penyebab kejadian gizi salah (*malnutrition*) di suatu masyarakat yang nantinya akan sangat berguna untuk melakukan intervensi gizi *(Supariasa, 2001)*.

2.2.8.3 Indeks Antropometri

Indeks antropometri adalah pengukuran dari beberapa parameter. Indeks antropometri bisa merupakan rasio dari satu pengukuran terhadap satu atau lebih pengukuran atau yang dihubungkan dengan umur dan tingkat gizi. Salah satu contoh dari indeks antropometri adalah Indeks Massa Tubuh (IMT) atau yang disebut dengan *Body Mass Index* (Supariasa, 2001).

Standar Antropometri Anak didasarkan pada parameter berat badan dan panjang/tinggi badan yang terdiri atas 4 (empat) indeks, meliputi:

1. Indeks Berat Badan menurut Umur (BB/U).

   Indeks BB/U ini menggambarkan berat badan relatif dibandingkan dengan umur anak. Indeks ini digunakan untuk menilai anak dengan berat badan kurang (*underweight*) atau sangat kurang (*severely nderweight*), tetapi tidak dapat digunakan untuk mengklasifikasikan anak gemuk atau sangat gemuk. Penting diketahui bahwa seorang anak dengan BB/U rendah, kemungkinan mengalami masalah pertumbuhan, sehingga perlu dikonfirmasi dengan indeks BB/PB atau BB/TB atau IMT/U sebelum diintervensi.

2. Indeks Panjang Badan menurut Umur atau Tinggi Badan menurut Umur (PB/U atau TB/U).

   Indeks PB/U atau TB/U menggambarkan pertumbuhan panjang atau tinggi badan anak berdasarkan umurnya. Indeks ini dapat mengidentifikasi anak-anak yang pendek (*stunted*) atau sangat pendek (*severely stunted*), yang disebabkan oleh gizi kurang dalam waktu lama atau sering sakit. Anak-anak yang tergolong tinggi menurut umurnya juga dapat

   diidentifikasi. Anak-anak dengan tinggi badan di atas normal (tinggi sekali) biasanya disebabkan oleh gangguan endokrin, namun hal ini jarang terjadi di Indonesia.

3. Indeks Berat Badan menurut Panjang Badan/Tinggi Badan (BB/PB atau BB/TB).

   Indeks BB/PB atau BB/TB ini menggambarkan apakah berat badan anak sesuai terhadap pertumbuhan panjang/tinggi badannya. Indeks ini dapat digunakan untuk mengidentifikasi anak gizi kurang (*wasted*), gizi buruk (*severely wasted*) serta anak yang memiliki risiko gizi lebih (*possible risk of overweight*). Kondisi gizi buruk biasanya disebabkan oleh

penyakit dan kekurangan asupan gizi yang baru saja terjadi (akut) maupun yang telah lama terjadi (kronis).

4. Indeks Masa Tubuh menurut Umur (IMT/U) Indeks IMT/U digunakan untuk menentukan kategori gizi buruk, gizi kurang, gizi baik, berisiko gizi lebih, gizi lebih dan obesitas. Grafik IMT/U dan grafik BB/PB atau BB/TB cenderung menunjukkan hasil yang sama. Namun indeks IMT/U lebih sensitif untuk penapisan anak gizi lebih dan obesitas. Anak dengan ambang batas IMT/U >+1SD berisiko gizi lebih sehingga perlu ditangani lebih lanjut untuk mencegah terjadinya gizi lebih dan obesitas.

Kategori dan ambang batas status gizi anak sebagai berikut :

Indeks	**Kategori Status Gizi**	**Ambang Batas (Z-Score)**
Berat Badan menurut Umur **(BB/U) anak usia 0 - 60 bulan**	Berat badan sangat kurang (*severely underweight*)	<-3 SD
	Berat badan kurang (*underweight*)	- 3 SD sd <- 2 SD
	Berat badan normal	-2 SD sd +1 SD
	Risiko Berat badan lebih[1]	> +1 SD
Panjang Badan atau Tinggi Badan menurut Umur **(PB/U atau TB/U) anak usia 0 - 60 bulan**	Sangat pendek (*severely stunted*)	<-3 SD
	Pendek (*stunted*)	- 3 SD sd <- 2 SD
	Normal	-2 SD sd +3 SD
	Tinggi[2]	> +3 SD

Berat Badan menurut Panjang Badan atau Tinggi Badan **(BB/PB atauBB/TB) anak usia 0 - 60 bulan**	Gizi buruk (*severely wasted*)	<-3 SD
	Gizi kurang (*wasted*)	- 3 SD sd <- 2 SD
	Gizi baik (normal)	-2 SD sd +1 SD
	Berisiko gizi lebih (*possible risk of overweight*)	> + 1 SD sd + 2 SD
	Gizi lebih (*overweight*)	> + 2 SD sd + 3 SD
	Obesitas (*obese*)	> + 3 SD
Indeks Massa Tubuh menurut Umur **(IMT/U) anak usia 0 - 60 bulan**	Gizi buruk (*severely wasted*)[3]	<-3 SD
	Gizi kurang (*wasted*)[3]	- 3 SD sd <- 2 SD
	Gizi baik (normal)	-2 SD sd +1 SD
	Berisiko gizi lebih (*possible risk of overweight*)	> + 1 SD sd + 2 SD
	Gizi lebih (*overweight*)	> + 2 SD sd +3 SD
	Obesitas (*obese*)	> + 3 SD
Indeks Massa Tubuh menurut	Gizi buruk (*severely thinness*)	<-3 SD
Indeks	**Kategori Status Gizi**	**Ambang Batas (Z-Score)**
Umur **(IMT/U) anak usia 5 - 18tahun**	Gizi kurang (*thinness*)	- 3 SD sd <- 2 SD
	Gizi baik (normal)	-2 SD sd +1 SD
	Gizi lebih (*overweight*)	+ 1 SD sd +2 SD
	Obesitas (*obese*)	> + 2 SD

Sumber : Permenkes No. 2 Tahun 2020