BAB 2
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1.Tinjauan Pustaka

Penelitian yang dilakukan oleh Hendri dkk pada 2021, membahas penerapan algoritma C-45 dalam mengukur kepuasan pengunjung terhadap fasilitas di Taman Margasatwa Jakarta. Penelitian bertujuan untuk memberi rekomendasi kepuasan pengunjung taman-taman margasatwa dengan menggunakan teknik datamining penerapan datamining C4.5. Sumber data diperoleh dari penyebaran kuesioner kepada pengunjung yang pernah berkunjung ke taman margasatwa sejumlah 35 orang. Atribut yang digunakan sebagai parameter penilaian kepuasan pengunjung antara lain : Paling Diminati, Fasilitas Semua, Tempat Parkir, Kemanan dan Kebersihan. Hasil pengolahan algoritma C4.5 dengan menggunakan pohon keputusan (*decision tree*) adalah model prediksi menggunakan struktur pohon atau struktur berhirarki.

Penelitian yang dilakukan oleh Rizki Haqmanullah Pambudi dkk pada tahun 2018, tentang Penerapan Algoritma C4.5 untuk Memprediksi Nilai Kelulusan Siswa Sekolah Menengah Berdasarkan Faktor Eksternal. Faktor eksternal berpengaruh pada kegagalan siswa dalam menyelesaikan bidang studi matematika. Algoritma *C4.5* merupakan salah satu metode *data mining* untuk memprediksi kemampuan siswa dalam menyelesaikan bidang studi dilihat dari faktor eksternal siswa. Algoritma *C4.5* digunakan untuk mengetahui tingkat akurasi prediksi kemampuan siswa sekolah menengah. Parameter pemilihan fitur adalah faktor-faktor yang mempengaruhi kemampuan siswa sekolah menengah dalam bidang studi matematika. Hasil pengujian dan analisis menunjukkan bahwa Algoritma *Decision Tree C4.5* akurat diterapkan untuk prediksi nilai akhir siswa sekolah menengah dengan tingkat akurasi 60%.

Penelitian yang dilakukan oleh Saifur Rohman Cholil dkk pada tahun 2021, penelitian tentang Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa. Penelitian bertujuan mendapatkan

calon penerima beasiswa yang tepat sasaran yaitu untuk mengklasifikasikan calon penerimaan beasiswa berdasarkan data yang diambil dari data siswa penerima beasiswa sebelumnya sebagai *data training* dengan data yang diambil dari calon penerima beasiswa sebagai *data testing*. Penelitian ini membantu proses seleksi beasiswa di SMA menggunakan algoritma K-Nearest Neighbor (KNN) supaya penerima beasiswa tepat sasaran. Hasil dari penelitian ini adalah terseleksinya 30 orang dari 89 data setelah diklasifikasi, pengujian menggunakan pengujian metode *confusion matrix*, menghasilkan akurasi sebesar 90%.

Menurut Aulia Dina dkk, pada tahun 2023, penelitian tentang Perbandingan Algoritma NBC, KNN dan C45 untuk Klasifikasi Penerima Bantuan Program Keluarga Harapan (PKH) bertujuan untuk mengatasi permasalahan yang terjadi pada PKH yaitu penyaluran bantuan yang masih belum tepat sasaran. Maka dari itu penelitian ini bertujuan untuk membuat model klasifikasi penerima bantuan PKH untuk mengatasi permasalahan tersebut. Algoritma yang digunakan untuk membuat model klasifikasi adalah Naïve Bayes Classifier (NBC), K-Nearest Neighbor (K-NN), dan C4.5. Metode validasi yang digunakan adalah K-Fold Cross Validation (K = 10). Jumlah atribut yang digunakan adalah 33 atribut. Data yang digunakan untuk pembuat model klasifikasi (data setelah praproses) adalah sebanyak 378 data calon penerima PKH. Berdasarkan hasil percobaan algoritma NBC menghasilkan nilai akurasi sebesar 77,51%, algoritma K-NN (K = 3) menghasilkan nilai akurasi sebesar 76,72%, algoritma C4.5 menghasilkan nilai akurasi sebesar 80,16%. Selain itu, algoritma C4.5 berhasil mereduksi jumlah atribut, dari 33 atribut menjadi 8 atribut saja, yaitu: jumlah art, fasbab, rumah lain, ada emas, ada lemari es, jumlah kamar, dinding, dan pembuangan tinja. Hal ini mengurangi kompleksitas dari model klasifikasi yang dihasilkan oleh algoritma C4.5.

Menurut Yholanda Maldini dkk, pada tahun 2021, melakukan penelitian tentang perbandingan algoritma C-45 dan KNN untuk menentukan pemberian kredit bagi nasabah koperasi. Pada proses simpan pinjam sering terjadi kredit macet, yang disebabkan nasabah koperasi memiliki masalah ekonomi yang buruk sehingga pembayaran kredit sedikit terhambat sehingga dilakukan penelitian untuk

menentukan pemberian kredit bagi nasabah dengan cara memperhatikan data yang dimasukkan nasabah untuk proses peminjaman. Penelitian ini membandingkan dua algoritma klasifikasi yaitu algoritma C 4.5 dan algoritma K-Nearest Neighbor (KNN). Hasil akurasi yang diperoleh dengan algoritma KNN sebesar 62%, sedangkan hasil akurasi algoritma C4.5 sebesar 57.5% sehingga hasil akurasi terbaik dari algoritma KNN bisa dijadikan acuan untuk menentukan pemberian kredit kepada nasabah.

2.2. Dasar Teori

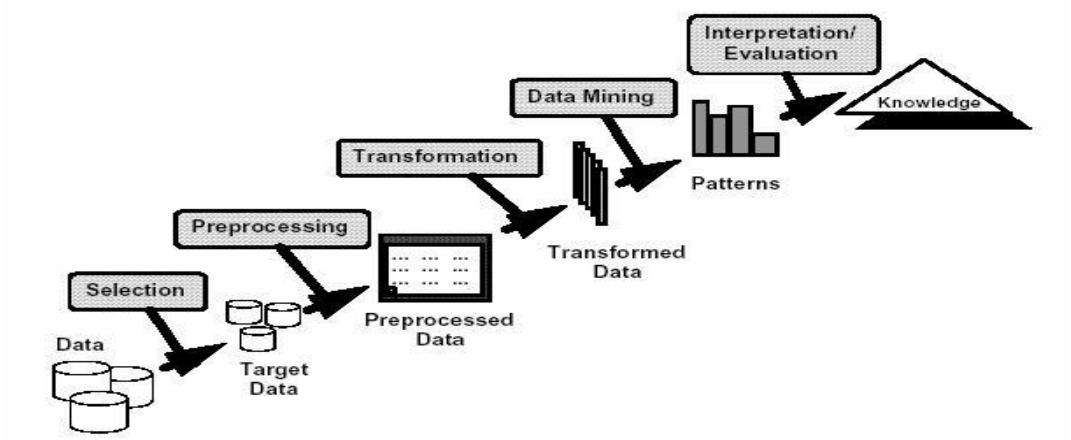
2.2.1.Definisi Data Mining

Data mining didefinisikan sebagai satu set teknik yang digunakan secara otomatis untuk mengeksplorasi secara menyeluruh dan membawa ke permukaan relasi-relasi yang kompleks pada set data yang sangat besar. Set data tersebut adalah set data yang berbentuk tabulasi, sebagaimana yang banyak diimplementasikan dalam teknologi manajemen basis data relasional. Akan tetapi, teknik-teknik *data mining* dapat juga diaplikasikan pada representasi data yang lain, seperti *domain* data *spatial*, berbasis *text*, dan multimedia (citra).

Data mining dapat juga didefinisikan sebagai "pemodelan dan penemuan pola-pola yang tersembunyi dengan memanfaatkan data dalam volume yang besar". *Data mining* merupakan sebuah proses untuk menemukan hubungan, pola, *trend* baru, yang bermakna dengan menyaring data yang sangat besar, yang tersimpan dalam penyimpanan, menggunakan teknik pengenalan pola seperti teknik statistik dan matematika. Selain pendapat diatas, ada yang berpendapat bahwa *Data mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan, dimana sebelumnya data tersebut tidak diketahui secara manual. Dari beberapa pendapat diatas dapat diketahui bahwa *Data mining* merupakan proses untuk mengetahui informasi baru dari data yang besar, dimana informasi tersebut tidak diketahui sebelumnya.

2.2.2.Proses Data Mining

Istilah *data mining* dan *knowledge discovery in databases* (KDD) sering digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Kedua istilah tersebut memiliki konsep yang berbeda, tetapi saling berkaitan. Salah satu tahapan dalam keseluruhan proses KDD adalah *data mining*.



Gambar 2.1 Tahapan Proses KDD

Berikut adalah penjelasan proses KDD secara garis besar:

1. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing/ Cleaning*

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi). Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

3. *Transformation*

Coding adalah adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

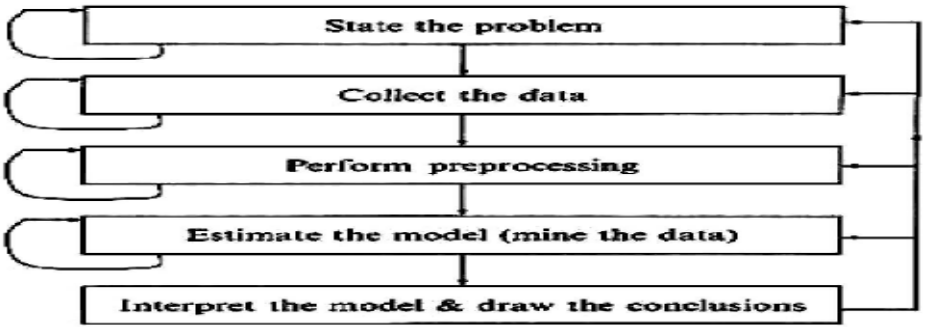
4. *Data mining*

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation / Evaluation*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut dengan *interpretation.* Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

Proses KDD secara garis besar terdiri dari 5 tahap seperti yang telah dijelaskan diatas. Tetapi, dalam proses KDD yang sebenarnya dapat terjadi iterasi atau pengulangan pada tahap-tahap tertentu. Sebagai contoh, pada saat *coding* atau *data mining*, analis menyadari proses *cleaning* belum dilakukan dengan sempurna, atau belum menemukan data atau informasi baru untuk memperkaya data yang sudah ada. Menurut *Kantardzic (2003)*, ada beberapa prosedur umum untuk menyelesaikan permasalahan dalam *data mining*, yaitu:



Gambar 2.2 Prosedur Umum Data Mining

1. Merumuskan permasalahan

   Pada tahap ini ditetapkan sebuah rumusan masalah serta variable-variabel yang terlibat.
2. Mengumpulkan data

   Pada prosedur ini, konsentrasi ditujukan pada proses pembuatan atau pengumpulan data.
3. Preprocessing data

   Untuk menyeleksi data yang akan digunakan dalam proses.
4. Estimasi model

   Dapat disebut sebagai proses utama pada prosedur ini, sebab implementasi dari teknik *data mining* dilakukan pada prosedur ini.
5. Menafsirkan informasi yang dihasilkan dari proses sebelumnya

2.2.3.Teknik Data Mining

Ada beberapa teknik yang dimiliki *data mining* berdasarkan tugas yang bisa dilakukan, yaitu :

1. Deskripsi

   Para peneliti / analis biasanya mencoba menemukan cara untuk mendeskripsikan pola dan *trend* yang tersembunyi dalam data.
2. Estimasi

   Estimasi mirip dengan klasifikasi, kecuali variabel tujuan yang lebih ke arah numerik daripada kategori. Misalnya, akan dilakukan estimasi tekanan *systolic* dari pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah.
3. Prediksi

   Prediksi memiliki kemiripan dengan estimasi dan klasifikasi. Hanya saja, prediksi hasilnya menunjukkan sesuatu yang belum terjadi (mungkin terjadi dimasa depan). Misalnya, ingin diketahui prediksi harga beras tiga bulan yang akan datang.

4. Klasifikasi

   Dalam klasifikasi variabel, tujuan bersifat kategorik. Misalnya, kita akan mengklasifikasikan pendapatan dalam tiga kelas, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

5. Clustering

   Clustering lebih ke arah pengelompokan *record*, pengamatan, atau kasus dalam kelas yang memiliki kemiripan. Sebuah *cluster* adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lain dan memiliki ketidak miripan dengan *record-record* dalam *cluster* yang lain, misalnya untuk tujuan *audit* akuntasi akan dilakukan segmentasi perilaku *financial* dalam kategori dan mencurigakan.

6. Asosiasi

   Mengidentifikasi hubungan antara berbagai peristiwa yang terjadi pada satu waktu. Pendekatan asosiasi tersebut menekankan sebuah kelas masalah yang dicirikan dengan analisis keranjang pasar.

Faktor-faktor Faktor – faktor yang mempengaruhi tingkat akurasi algoritma data mining adalah sebagai berikut :

1. *Tree Pruning*

   Tree Prunning dilakukan untuk menyederhanakan tree sehingga akurasi dapat bertambah.

2. Pembersihan data atau *pre-processing*

   Data-data yang tidak relevan juga dibuang karena apabila terdapat data yang tidak konsisten, data hilang, noise, data tidak valid, ataupun sekedar salah ketik maka dapat mengurangi tingkat akurasi dalam *data mining*. Penentuan *data training* sangat menentukan tingkat *akurasi tree* yang dibuat.

3. Jumlah *data training*

   Dalam beberapa penelitian telah dibuktikan bahwa jumlah *data training* dapat mempengaruhi tingkat akurasi. Semakin banyak *data training* maka semakin tinggi pula tingkat akurasi yang didapat.

4. Jumlah atribut dan *class*

Salah satu faktor yang mempengaruhi tingkat akurasi adalah banyaknya atribut dan *class*, semakin banyak atribut dan *class* yang digunakan semakin banyak juga variasi rule yang didapat untuk mencari akurasi.

2.2.4.Cross Industry Standard Process for Data Mining (CRISP-DM)

Untuk melaksanakan *project-project* dalam *Data mining* (DM) secara sistematis, suatu proses umum biasanya dilakukan. Berdasarkan '*best practice*', para praktisi dan peneliti DM mengusulkan beberapa proses (*workflow* atau pendekatan *step-by-step* yang sederhana) untuk memperbesar peluang keberhasilan dalam melaksanakan *project-project* DM. Usaha-usaha itu akhirnya menghasilkan beberapa proses yang dijadikan sebagai standar, beberapa diantaranya (yang paling popular) dibahas dalam bagian ini.

Salah satu proses yang sudah dijadikan standar tersebut, boleh dibilang sebagai yang paling populer, yaitu '*Cross-Industry Standard Process for Data mining*' – CRISP-DM – diusulkan pada pertengahan 1990an oleh konsorsium perusahaan-perusahaan Eropa untuk dijadikan *methodology standard nonproprietary* bagi DM (CRISP-DM, 2009).

Proses *data mining* berdasarkan CRISP-DM terdiri dari 6 fase, yaitu:

1. *Business Understanding* / *Research Understanding Phase*

   Pemahaman domain (penelitian). Pada fase ini dibutuhkan pemahaman tentang substansi dari kegiatan *data mining* yang akan dilakukan, kebutuhan dari perspektif bisnis.

2. *Data Understanding*

   Pemahaman data adalah fase mengumpulkan data awal, mempelajari data untuk bisa mengenal data yang akan dipakai.

3. *Data Preparation*

   Persiapan data. Fase ini sering disebut fase padat karya.

4. *Modelling*

   Fase menentukan teknik *data mining* yang digunakan, menentukan *tools data mining,* teknik *data mining,* algoritma *data mining,* menentukan parameter dengan nilai yang optimal.

5. *Evaluation*

   Fase interpretasi terhadap hasil *data mining* yang ditunjukkan dalam proses pemodelan pada fase sebelumnya.

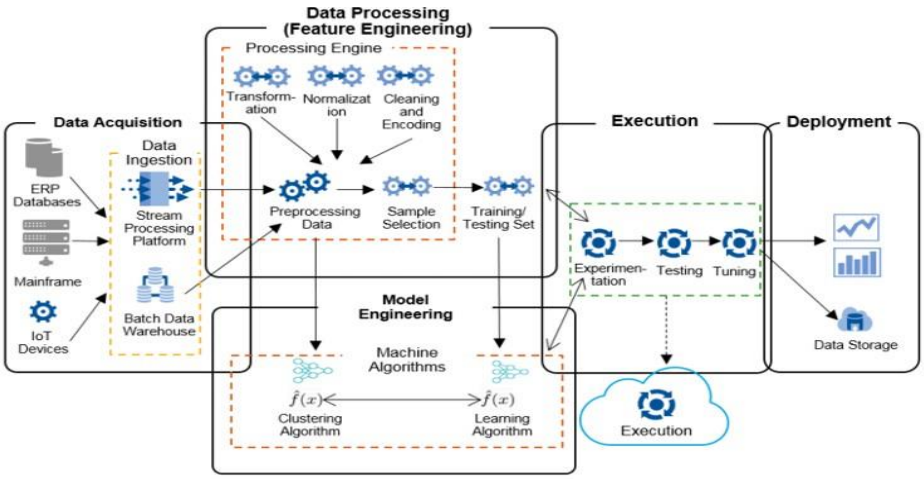
6. *Deployment*

   Fase penyusunan laporan atau presentasi dari pengetahuan yang didapat dari evaluasi pada proses *data mining.*

2.2.5.*Machine Learning*

Machine learning adalah salah satu bidang di mana kecerdasan buatan yang mengembangkan algoritma yang dapat mempelajari pola dan aturan keputusan daridata (Seffens dkk., 2015). *Machine learning* berfokus pada pengembangan program komputer yang dapat mengakses data dan menggunakannya untuk belajar sendiri.

Infrastruktur aplikasi *machine learning* sangat fleksibel disesuaikan dengan kebutuhan skala proses dan volume data yang proses. Arsitektur *machine learning* diilustrasikan pada Gambar 2.1



Gambar 2.3 Arsitektur Machine Learning

Gambar menunjukkan ilustrasi arsitektur dari *machine learning* yang terdiri dari*Data acquisition, Data processing, Data modeling or model engineering, Execution, Deployment.* Penjelasan dari masing-masing bagian adalah sebagai berikut :

a. Pengambilan data (*Data acquisition*).

   Data dikumpulkan, disiapkan dan kemudian diteruskan untuk diproses

b. Pemrosesan data (*Data processing*).

   Langkah-langkah seperti preprocessing, pemilihan sampel dan pelatihan dataset berlangsung, dalam persiapan untuk pelaksanaan *machine learning*. Analisis fitur atau rekayasa fitur (*Feature analysis or feature engineering*) (bagian dari komponen pemrosesan data), fitur yang menggambarkan struktur yang melekatdalam data dipilih dan dianalisis.

c. Pemodelan data atau rekayasa model (*Data modeling or model engineering*). Desain model data dan algoritma yang digunakan dalam pemrosesan data ML(termasuk algoritma pengelompokan dan pelatihan):

   1. *Model fitting* (di mana satu set data pelatihan ditugaskan ke model untukmembuat prediksi yang dapat diandalkan pada data baru atau tidak terlatih
   2. Evaluasi model (*Model evaluation*). Model dievaluasi berdasarkan kinerjadan kemanjuran.

d. Eksekusi (*Execution*).

   Lingkungan di mana data yang diproses, dan dilatih diteruskan untuk digunakan dalam pelaksanaan rutinitas ML (seperti eksperimen, pengujian, dan penyetelan)

e. Penempatan (*Deployment*).

   Pemanfaatan dari ML - seperti model atau wawasan - digunakan untuk aplikasi perusahaan, sistem atau penyimpanan data (misalnya, untuk pelaporan)

Proses pembelajaran dimulai dengan pengamatan atau data, seperti contoh,pengalaman langsung, atau instruksi, untuk mencari pola dalam data dan membuatkeputusan yang lebih baik dimasa depan berdasarkan contoh yang diberikan. Tujuan utamanya adalah untuk memungkinkan

komputer belajar secaraotomatis tanpa campur tangan atau bantuan manusia dan menyesuaikan tindakan yang sesuai

Machine learning memungkinkan analisis data dalam jumlah besar. *Machine learning* umumnya memberikan hasil yang lebih cepat, lebih akurat untuk mengidentifikasi peluang yang menguntungkan atau risiko berbahaya, mungkin juga memerlukan waktu dan sumber daya tambahan untuk melatih *machine learning* dengan benar. M*achine learning* dengan digabungkan dengan teknologi kognitif dapat membuat pekerjaan lebih efektif dalam memproses informasi dalam volume besar.

2.2.6.*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*.

2.2.7.*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 simpul 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 atribut A
      3) Jika sampel Svi kosong
         i. Dibawah 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*, 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* yang jarang dipotong akan menjadi *leaf* (*node* akhir) dengan kelas yang paling sering muncul.

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 C4.5 berikut ini disertakan contoh kasus yang dituangkan dalam Tabel 2.1

Tabel 2 .1 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 2.1 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)*.

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

Keterangan:

S = Himpunan Kasus

A = Atribut

n = Jumlah Partisi Atribut A

|Si| = Jumlah Kasus pada partisi ke-i

|S| = Jumlah Kasus dalam S

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

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

Keterangan :

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 2.1

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 2.2

Tabel 2 .2 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 2.2 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 :

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

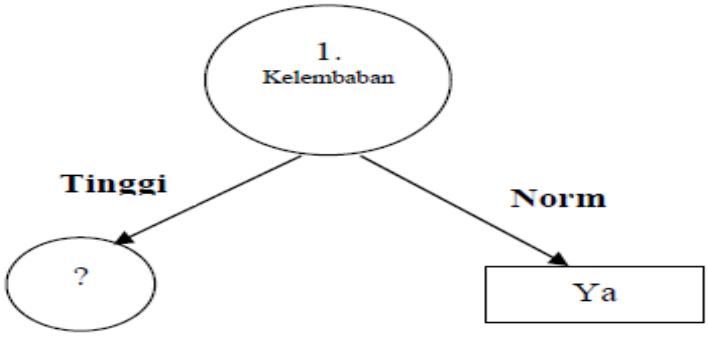
$$Gain(Total, Cuaca) = 0.863120569 - (\frac{4}{-14 * 0)} + \frac{5}{(14 * 0.723)} + \frac{45}{(14 * 0.97))}$$

$$Gain(Total, Cuaca) = 0.23$$

Dari hasil pada Tabel 2.2 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 2.4



Gambar 2.4 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 2.3

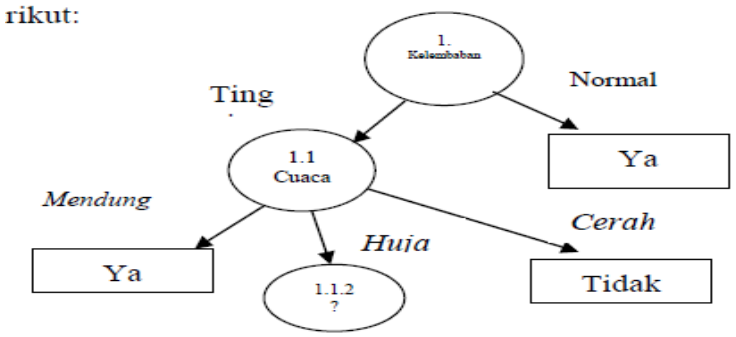
Tabel 2 .3 Perhitungan Node 1.1

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 2.3 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 2.4 berikut:



Gambar 2.5 Pohon Keputusan Hasil Perhitungan Node 1.1

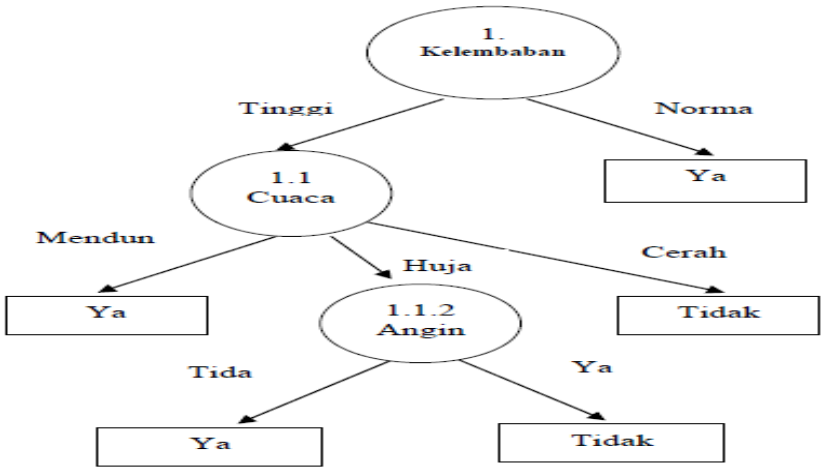
3. Menghitung jumlah kasus, jumlah kasus untuk keputusan Ya, jumlah kasus untuk keputusan Tidak, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut temperatur dan angin yang dapat menjadi *node* cabang dari nilai atribut hujan. Setelah itu lakukan penghitungan *Gain* untuk masingmasing atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.4

Tabel 2. 4 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 2.4 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 2.5



Gambar 2.6 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.

2.2.7.1. *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:

$$\boldsymbol{Entropy(S) = -p + \log_{p+} -p - \log_2 p -}$$

Keterangan :

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.

2.2.7.2.*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. Secara matematis, *infomation gain* dari suatu atribut A, dituliskan sebagai berikut:

$$\boldsymbol{Gain(S,A) = Entropy(S) - \sum \frac{|Si|}{|S|} * Entropy(Sv)}$$

Keterangan :

A	= atribut
v	= menyatakan suatu nilai yang mungkin untuk atribut A
Values(A)	= himpunan yang mungkin untuk atribut A
|S|	= jumlah seluruh sempel data
Entropy(Sv)	= entropy untuk *sample-sample* yang memiliki nilai v

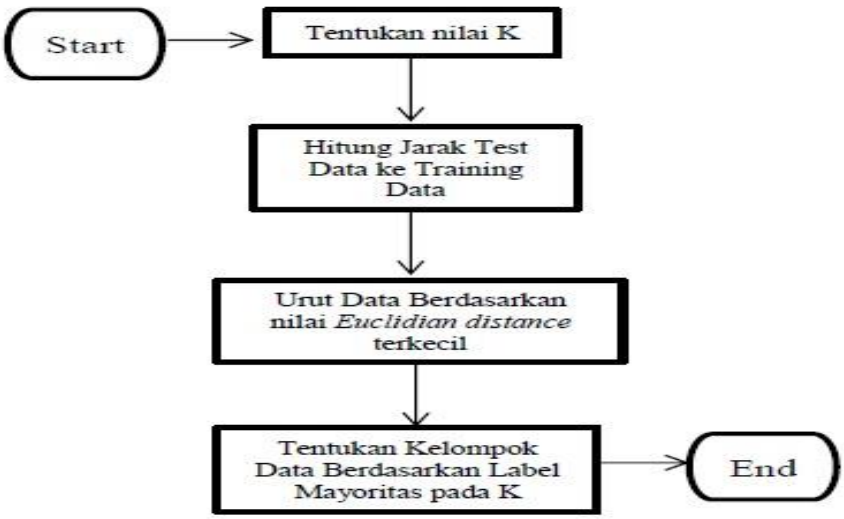
2.2.8. *Algoritma K-Nearest Neighbor (K-NN)*

Algoritma Nearest Neighbor adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada. (Kusrini dan Luthfi, 2009) Algoritma K-NN adalah suatu metode yang menggunakan algoritma supervised. Perbedaan antara supervised learning dengan unsupervised learning adalah pada supervised learning bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Tujuan dari

algoritma K-NN adalah untuk mengklasifikasi objek baru berdasarkan atribut atau training samples.

K-Nearest Neighbor sering digunakan dalam klasifikasi dengan tujuan dari algoritma ini adalah untuk mengklasifikasi objek baru berdasarkan atribut dan *training samples*. *Algoritma K-Nearest Neighbor* (K-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Teknik ini sangat sederhana dan mudah diimplementasikan. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruang ini ditandai kelas *c* jika kelas *c* merupakan klasifikasi yang paling banyak ditemui pada *k* buah tetangga terdekat titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak *Euclidean*. Untuk mendefinisikan jarak antara dua titik yaitu titik pada *data testing* (x) dan titik pada *data training* (y) maka digunakan rumus *Euclidean distance* (Santoso, 2007).

Dalam pengenalan pola, algoritma KNN merupakan metode non parametrik yang digunakan untuk klasifikasi dan regresi. Algoritma KNN memiliki kelebihan yaitu dapat menghasilkan data yang kuat atau jelas dan efektif jika digunakan pada data besar. Algoritma KNN dapat ditunjukkan pada *flowchart* sebagai berikut :



Gambar 2.7 Algoritma K-Nearest Neighbor

Penentuan nilai k yang baik bergantung pada data yang digunakan. Umumnya, nilai k yang lebih besar mengurangi efek *noise* pada klasifikasi, namun menyebabkan batasan antar kelas sedikit berbeda. Jarak antar data yang digunakan dalam algoritma KNN dapat dihitung menggunakan rumus jarak *Euclidean Distance* dan *Manahattan Distance.* Rumus jarak yang biasa digunakan untuk KNN adalah *Euclidean Distance,* yang mempresentasikan cara berpikir manusia tentang jarak pada kehidupan nyata. Berikut adalah rumus *Euclidean Distance :*

$$\mathbf{D}_{\boldsymbol{Euclidean}}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{\boldsymbol{k}}^{\boldsymbol{n}} = \mathbf{1} (\boldsymbol{X_k} - \boldsymbol{Y_k})^2}$$

Keterangan : Xk dan Yk merupakan atribut ke-k dari X dan Y berturut-turut.

Berikut ini merupakan contoh perhitungan menggunakan algoritma KNN dengan mengambil data training sebanyak 20 data dari alumni:

Tabel 2 .5 Data Training

NO	NIM	IPS 1	IPS 2	IPS 3	IPS 4	TOTAL SKS	STATUS
1	135610075	3,19	3,52	3,50	3,13	91	LC
2	135610082	3,19	3,52	3,33	3,22	91	LC
3	135610123	3,86	3,65	3,36	3,68	88	LC
4	135610140	3,38	3,38	3,29	3,33	93	LC
5	135610152	2,62	2,86	3,60	3,42	86	LC
6	135610157	3,62	3,38	3,26	3,67	89	LC
7	085610034	1,00	1,38	0,00	1,13	70	LL
8	095610111	2,14	0,79	1,62	0,13	68	LL
9	095610116	1,10	1,55	0,00	2,20	63	LL
10	105610080	1,48	2,62	2,29	1,54	70	LL
11	105610087	2,24	0,00	1,92	2,57	74	LL
12	115610022	1,76	1,93	2,43	1,90	70	LL
13	115610062	2,05	2,19	1,59	1,15	82	LL
14	115610118	2,14	2,50	2,80	2,33	82	LL
15	125610042	2,10	2,24	2,50	2,52	81	LT
16	125610050	2,14	2,56	2,90	2,67	83	LT
17	125610055	1,56	2,19	2,19	2,00	71	LT
18	125610056	1,86	1,94	2,39	1,72	73	LT
19	125610069	2,33	2,67	2,35	1,26	81	LT
20	125610083	2,71	2,05	2,61	2,50	80	LT

Keterangan Status :

Lulus Cepat (LC) : 7 semester

Lulus Tepat (LT) : 8 sampai dengan 10 semester

Lulus Lambat (LL) : lebih dari 10 semester

Terdapat data baru mahasiswa sebagai data uji kemudian dilakukan perhitungan berdasarkan metode K-NN dengan *euclidean distance* sebagai berikut

Nim : 165610122 IPS 2 : 3,73 IPS 4 : 3,10

IPS 1 : 3,83 IPS 3 : 3,50 Total SKS Semester 4 : 77

Dari data baru untuk mahasiswa dengan nim 165610122 tersebut dihitung jarak dengan menggunakan rumus (2.2) nilai K adalah 5 sebagai berikut :

1. Nilai d(x21, y1) =

$$\sqrt{(3,83-3,19)^2+(3,73-3,52)^2+(3,50-3,50)^2+(3,10-3,13)^2+(77-91)^2}$$

$$= \sqrt{0,4096+0,0441+0+0,0009+196} = \sqrt{196,4546}$$

$$= 14,0162$$

2. Nilai d(x21, y2)

$$= \sqrt{(3,83-3,19)^2+(3,73-3,52)^2+(3,50-3,33)^2+(3,10-3,22)^2+(77-91)^2}$$

$$= \sqrt{0,4096+0,0441+0,0289+0,0144+196} = \sqrt{196,4970}$$

$$= 14,0177$$

3. Nilai d(x21, y3)

$$= \sqrt{(3,83-3,86)^2+(3,73-3,65)^2+(3,50-3,36)^2+(3,10-3,38)^2+(77-88)^2}$$

$$= \sqrt{0,0009+0,0064+0,0196+0,3364+121} = \sqrt{121,3633}$$

$$= 11,0165$$

4. Nilai d(x21,y4)

$$= \sqrt{(3,83-3,38)^2+(3,73-3,38)^2+(3,50-3,29)^2+(3,10-3,33)^2+(77-93)^2}$$

$$= \sqrt{0,2025+0,12225+0,0441+0,0529+256} = \sqrt{256,4220}$$

$$= 16,0132$$

Berikut ini adalah tabel hasil perhitungan jarak menggunakan *Euclidean Distance* :

Tabel 2 .6 Perhitungan Jarak Menggunakan Euclidean Distance

NO	NIM	JARAK EUCLIDEAN	RANKING
1	135610075	14.0162	17
2	135610082	14.0177	18
3	135610023	11.0165	15
4	135610140	16.0132	20
5	135610152	9.1287	13
6	135610157	12.0229	16
7	135610034	8.8692	12
8	135610111	10.2399	14
9	135610116	14.875	19
10	135610080	7.7235	11
11	135610087	5.3121	5
12	135610022	7.6883	10
13	135610062	6.1637	7
14	135610118	5.5183	6
15	135610042	4.7486	2
16	135610050	6.3851	8
17	135610055	6.8155	9
18	135610056	5.1207	4
19	135610069	4.9073	3
20	135610083	3.7721	1

Nearest Neighbor ditentukan pada awal adalah K = 5, yaitu 5 jarak yang paling kecil.

Tabel 2 .7 Jarak Terdekat Sebanyak K=5

No	NIM	Jarak	Rangking	Status
1	125610083	3,7721	1	LT
2	125610042	4,7486	2	LT
3	125610069	4,9073	3	LT
4	125610056	5,1207	4	LT
5	105610087	5,3121	5	LL

a. Menghitung jumlah status yang lebih banyak muncul. Pada bagian ini, kemunculan status terbanyak adalah LT sebanyak 4 kali, sedangkan kemunculan status LL sebanyak satu kali.

b. Kesimpulan dari hasil perkiraan masa studi mahasiswa dengan 165610122 adalah Lulus Tepat Waktu (LT).

2.2.9. Rapid Miner

2.2.9.1 Sejarah Rapid Miner

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 RapidMiner 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 *data mining*. 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.

2.2.9.2 Pengertian *Rapid Miner*

Rapid Miner adalah aplikasi *data mining open-source* yang terkemuka dan ternama di dunia. Dirancang sebagai aplikasi yang berdiri sendiri untuk analisis data dan sebagai mesin pengolah *data mining* 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.

2.2.9.3 Terminologi Dasar Rapid Miner

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.

2.2.9.4 Desain Proses Analisa Dalam Rapid Miner

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.

2.2.9.5.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.

2.2.9.6.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*).

2.2.9.7.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.

2.2.9.8.Operator dan Proses

Proses *data mining* 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*.

2.2.10.Status Gizi

2.2.10.1 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)*.

2.2.10.2 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)*.

2.2.10.3 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 :

Tabel 2 .8 Indeks Massa Tubuh

Indeks	**Kategori Status Gizi**	**Ambang Batas (Z-Score)**
Berat Badan menurut Umur **(BB/U) anak usia 0 - 60 bulan**	Berat badan sangatkurang (*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**	Gizi buruk (*severely wasted*)	<-3 SD
	Gizi kurang (*wasted*)	- 3 SD sd <- 2 SD

atauBB/TB) anak usia 0 - 60 bulan	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 menurutUmur **(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

Sumber : Permenkes No. 2 Tahun 2020