BAB II

LANDASAN TEORI

2.1 Stroke

Stroke adalah serangan otak yang timbul secara mendadak dimana terjadi gangguan fungsi otak sebagian atau menyeluruh sebagai akibat dari gangguan aliran darah oleh karena sumbatan atau pecahnya pembuluh darah tertentu diotak. Stroke merupakan gangguan fungsi saraf pusat yang berkembang sangat cepat baik menit maupun jam (Yayasan Stroke Indonesia, 2013). Stroke disebabkan oleh keadaan *ischemic* atau proses *hemorrhagic* yang seringkali diawali oleh adanya lesi atau perlukaan pada pembuluh darah arteri. Dari seluruh kejadian stroke, duapertiganya adalah *ischemic* dan sepertiganya adalah *hemorrhagic*. Disebut *stroke ischemic* karena adanya sumbatan pembuluh darah oleh *thromboembolic* yang mengakibatkan daerah di bawah sumbatan tersebut mengalami *ischemic*. Hal ini sangat berbeda dengan *stroke hemorrhagic* yang terjadi akibat adanya *mycroaneurisme* yang pecah (Guytom, Arthur C; John E Hall, 2007).

WHO menyebutkan stroke adalah terjadinya gangguan fungsional otak fokal maupun global secara mendadak dan akut yang berlangsung lebih dari 24 jam akibat gangguan aliran darah otak. Menurut Neil F. Gordon (2002) stroke adalah gangguan potensial yang fatal pada suplai darah bagian otak. Tidak ada satupun bagian tubuh manusia yang dapat bertahan bila terdapat gangguan suplai darah dalam waktu relatif lama sebab darah sangat dibutuhkan dalam kehidupan terutama oksigen pengangkut bahan makanan yang dibutuhkan pada otak dan otak adalah pusat *control system* tubuh termasuk perintah dari semua gerakan fisik. Dengan kata lain stroke merupakan manifestasi keadaan pembuluh darah *cerebral* yang tidak sehat sehingga bisa disebut juga *"cerebral arterial disease"* atau *"cerebrovascular disease".* Cedera dapat disebabkan oleh sumbatan bekuan darah, penyempitan pembuluh darah, sumbatan dan

penyempitan atau pecahnya pembuluh darah, semua ini menyebabkan kurangnya pasokan darah yang memadai (Irfan, 2010).

Berdasarkan data WHO (*World Healt Organisation*) tahun 2010 setiap tahunnya terdapat 15 juta orang diseluruh dunia menderita stroke. Diantaranya ditemukan jumlah kematian sebanyak 5 juta orang dan 5 juta orang lainnya mengalami kecacatan permanen. Penyakit stroke menduduki urutan ketiga sebagai penyebab utama kematian setelah penyakit jantung koroner dan kanker di negara-negara berkembang. Negara berkembang juga menyumbang 85,5 % dari total kematian akibat stroke di seluruh dunia. Stroke merupakan penyakit terbanyak ketiga setelah penyakit jantung dan kanker, serta merupakan penyakit penyebab kecacatan tertinggi di dunia. Menurut *American Heart Association (AHA),* angka kematian penderita stroke di Amerika setiap tahunnya adalah 50 – 100 dari 100.000 orang penderita (A, Basjiruddin ; darwin Amir (ed.). 2008)

Di negara-negara ASEAN penyakit stroke juga merupakan masalah kesehatan utama yang menyebabkan kematian. Dari data South East Asian Medical Information Centre (SEAMIC) diketahui bahwa angka kematian stroke terbesar terjadi di Indonesia yang kemudian diikuti secara berurutan oleh Filipina, Singapura, Brunei, Malaysia, dan Thailand. Dari seluruh penderita stroke di Indonesia, *stroke ischemic* merupakan jenis yang paling banyak diderita yaitu sebesar 52,9%, diikuti secara berurutan oleh perdarahan *intraserebral, emboli* dan perdarahan *subaraknoid* dengan angka kejadian masing-masingnya sebesar 38,5%, 7,2%, dan 1,4% (A, Basjiruddin ; darwin Amir (ed.). 2008)

2.2 Faktor Risiko Penyakit Stroke

Faktor risiko stroke adalah kondisi atau penyakit atau kelainan yang terdapat pada seseorang yang memiliki potensi untuk memudahkan orang tersebut mengalami serangan stroke pada suatu saat (Yastroki, 2011). Terdapat dua macam faktor risiko penyakit stroke, yaitu faktor risiko yang dapat diubah atau dikendalikan dan faktor risiko yang tidak dapat diubah atau dikendalikan.

Terdapat banyak sekali macam-macam faktor resiko stroke yang dikemukakan oleh beberapa ahli diantaranya adalah sebagai berikut: Faktor risiko yang tidak dapat diubah atau dikendalikan (Valensia, S.A. 2015) meliputi:

1. Usia
2. Jenis Kelamin
3. Ras

Sedangkan faktor risiko yang dapat diubah atau dikendalikan (Valensia, S.A. 2015) meliputi:

1. Tekanan Darah : Yang dimaksud dengan Tekanan Darah adalah jumlah tenaga darah yang ditekan terhadap dinding Arteri (pembuluh nadi) saat Jantung memompakan darah ke seluruh tubuh manusia. Tekanan darah merupakan salah satu pengukuran yang penting dalam menjaga kesehatan tubuh, karena Tekanan darah yang tinggi atau Hipertensi dalam jangka panjang akan menyebabkan perenggangan dinding arteri dan mengakibatkan pecahnya pembuluh darah. Pecahnya pembuluh darah inilah yang menyebabkan terjadinya stroke.
2. Kadar Gula Darah : Gula darah merupakan istilah yang mengacu pada kadar atau banyaknya kandungan gula di dalam sirkulasi darah di dalam tubuh.
3. Kadar Kolesterol Total : Kolesterol total merupakan kadar keseluruhan kolesterol yang beredar dalam tubuh manusia.
4. *Low Density Lipoprotein* (LDL) : Kolestrol LDL adalah lemak yang jahat karena bisa menimbun pada dinding dalam dari pembuluh darah, terutama pembuluh darah kecil yang menyuplai makanan ke jantung dan otak. Timbunan lemak itu semakin lama semakin tebal dan keras, yang dinamakan arteriosklerosis, dan akhirnya menumbat aliran darah.
5. Asam Urat : Zat hasil metabolisme purin dalam tubuh. Zat asam urat ini biasanya akan dikuluarkan oleh ginjal melalui *urine* dalam kondisi normal.
6. *Blood Urea Nitrogen* (BUN) : didefinisikan sebagai jumlah *nitrogen urea* yang hadir dalam darah.

7. Kreatinin (*creatinine*) : adalah *asam amino* yang diproduksi oleh hati, *pankreas* dan ginjal.

2.3 Klasifikasi Stroke dan Etologi

Terdapat dua macam bentuk stroke yaitu stroke iskemik dan stroke hemoragik. Stroke iskemik merupakan 80% dari penyebab stroke, disebabkan oleh gangguan pasokan oksigen dan nutrisi ke sel-sel otak akibat bentukan trombus atau emboli. Keadaan ini dapat diperparah oleh terjadinya penurunan perfusi sistemik yang mengaliri otak. Sedangkan *stroke hemoragik intraserebral* dan *subarakhnoid* disebabkan oleh pecahnya pembuluh darah kranial (Smith et al., 2005). Stroke secara luas diklasifikasikan menjadi stroke iskemik dan hemoragik. Stroke iskemik merupakan 80% kasus stroke dan dibagi menjadi aterotrombosis arteri, emboli otak, stroke lakunar, dan hipoperfusi sistemik. Perdarahan otak merupakan 20% sisa penyebab stroke dan dibagi menjadi perdarahan *intraserebral*, perdarahan *subarakhnoid*, dan *hematoma subdural/ekstradural* (Goldszmidt et al., 2003).

a. Stroke Hemoragik

Stroke perdarahan atau stroke hemoragik adalah perdarahan yang tidak terkontrol di otak. Perdarahan tersebut dapat mengenai dan membunuh sel otak, sekitar 20% stroke adalah stroke hemoragik (Gofir,2009). Jenis perdarahan (stroke hemoragik), disebabkan pecahnya pembuluh darah otak, baik intrakranial maupun subarakhnoid. Pada perdarahan intrakranial, pecahnya pembuluh darah otak dapat karena berry aneurysm akibat hipertensi tak terkontrol yang mengubah morfologi arteriol otak atau pecahnya pembuluh darah otak karena kelainan kongenital pada pembuluh darah otak tersebut. Perdarahan subarakhnoid disebabkan pecahnya aneurysma congenital pembuluh arteri otak di ruang subarakhnoidal (Misbach dkk., 2007).

b. Stroke Iskemik

Stroke iskemik mempunyai berbagai etiologi, tetapi pada prinsipnya disebabkan oleh aterotrombosis atau emboli, yang masing-masing akan

mengganggu atau memutuskan aliran darah otak atau *cerebral blood flow (CBF).* Nilai normal CBF adalah 50–60 ml/100 mg/menit. Iskemik terjadi jika CBF < 30 ml/100mg/menit. Jika CBF turun sampai < 10 ml/mg/menit akan terjadi kegagalan homeostasis, yang akan menyebabkan influks kalsium secara cepat, aktivitas protease, yakni suatu cascade atau proses berantai eksitotoksik dan pada akhirnya kematian neuron. Reperfusi yang terjadi kemudian dapat menyebabkan pelepasan radikal bebas yang akan menambah kematian sel. Reperfusi juga menyebabkan transformasi perdarahan dari jaringan infark yang mati. Jika gangguan CBF masih antara 15–30 ml/100mg/menit, keadaan iskemik dapat dipulihkan jika terapi dilakukan sejak awal (Wibowo dkk., 2001).Stroke iskemik akut adalah gejala klinis defisit serebri fokal dengan onset yang cepat dan berlangsung lebih dari 24 jam dan cenderung menyebabkan kematian. Oklusi pembuluh darah disebabkan oleh proses trombosis atau emboli yang menyebabkan iskemia fokal atau global. Oklusi ini mencetuskan serangkaian kaskade iskemik yang menyebabkan kematian sel neuron atau infark serebri (Adam et al., 2001; Becker et al.,2006). Aliran darah ke otak akan menurun sampai mencapai titik tertentu yang seiring dengan gejala kelainan fungsional, biokimia dan struktural dapat menyebabkan kematian sel neuron yang irreversible(WHO, 1989; Adam et al.,2003; Bandera et al.,2006).

c. Stroke Ringan *(Transient Ischaemic Attack)*

Stroke ringan sendiri dalam bahasa medis dinamakan serangan iskemik transien atau *transient ischaemic attack* (TIA). Penyebab penyakit ini sama dengan stroke, yaitu terhalangnya aliran darah ke otak. Sehingga wajar jika gejala yang muncul mirip dan organ tubuh yang terlibat pun sama. Gejala TIA atau stroke ringan sendiri biasanya mudah dilihat melalui wajah, lengan, dan kemampuan bicara. Stroke ringan dapat menyebabkan kelemahan otot wajah sehingga wajah turun ke salah satu sisi, tidak bisa senyum, mata, atau mulut turun ke bawah. Penderita stroke ringan kemungkinan takkan mampu mengangkat kedua lengan. Hal ini terjadi karena lengan mereka lemas atau mati rasa pada salah satu sisi. Pada

orang yang mengalami stroke ringan, kemampuan bicara mereka juga bisa terganggu. Mereka akan cadel, tidak beraturan, atau bahkan tidak mampu bicara sama sekali. Jika salah satu gejala yang telah disebutkan, maka segeralah menghubungi dokter.

Perbedaan mendasar antara stroke ringan dengan stroke adalah ukuran atau tingkat keparahan sumbatan yang menghalangi aliran darah ke otak. Pada stroke ringan, sumbatan tergolong kecil sehingga bisa diperbaiki kembali oleh sistem pertahanan dalam tubuh. Akibatnya, gejala stroke ringan tidak bersifat permanen dan segera hilang. Jika sumbatan tersebut lebih besar atau parah, maka bisa terjadi stroke sepenuhnya.

2.4 Pengendalian Stroke

Untuk menurunkan angka kesakitan, kecacatan dan kematian diperlukan pengendalian stroke. Kegiatan pengendalian stroke meliputi (Kemenkes, 2013):

1. Pelayanan pra stroke

   Pelayanan pra stroke adalah kegiatan deteksi dini, penemuan dan monitoring faktor risiko stroke pada individu sehat dan berisiko di masyarakat. Pelayanan pra stroke dilakukan di:

   a. Puskesmas
   b. Klinik kesehatan
   c. Posbindu PTM

2. Pelayanan serangan stroke

   Pelayanan serangan stroke dilakukan di:

   a. Rumah sakit dipusatkan pada unit stroke atau pojok stroke
   b. Rumah sakit khusus

3. Pelayanan paska stroke

   Pelayanan paska stroke dilakukan di:

   a. Rumah sakit
   b. Puskesmas
   c. Posbindu PTM

2.5 *Data Mining*

Secara sederhana, *data mining* merupakan ekstraksi informasi yang tersirat dalam sekumpulan data. *Data mining* merupakan sebuah proses untuk menggali kumpulan data dan menemukan informasi di dalamnya. (Turban, E., dkk. 2005). *Data mining* merupakan proses pengekstrakan informasi dari jumlah kumpulan data yang besar dengan menggunakan algoritma dan tehnik gambar dari statistik, mesin pembelajaran dan sistem manajemen *database*. Penggalian data ini dilakukan pada sekumpulan data yang besar untuk menemukan pola atau hubungan yang ada dalam kumpulan data tersebut (Kusrini dan E.T. Lutfi. 2009). Hasil penemuan yang diperoleh setelah proses penggalian data ini, kemudian dapat digunakan untuk analisis yang lebih lanjut.

Data mining yang disebut juga dengan *Knowledge-Discovery in Database* (KDD) adalah sebuah proses secara otomatis atas pencarian data di dalam sebuah memori yang amat besar dari data untuk mengetahui pola dengan menggunakan alat seperti klasifikasi, hubungan (*association*) atau pengelompokan (*clustering*). Proses KDD ini terdiri dari langkah-langkah sebagai berikut (Han, J. dan M. Kamber. 2006):

1. *Data Cleaning*, proses menghapus data yang tidak konsisten dan kotor.
2. *Data Integration*, penggabungan beberapa sumber data.
3. *Data Selection*, pengambilan data yang akan dipakai dari sumber data.
4. *Data Transformation*, proses dimana data ditransformasikan menjadi bentuk yang sesuai untuk diproses dalam data mining.
5. *Data Mining*, suatu proses yang penting dengan melibatkan metode untuk menghasilkan suatu pola data.
6. *Pattern Evaluation*, proses untuk menguji kebenaran dari pola data yang mewakili *knowledge* yang ada didalam data itu sendiri.
7. *Knowledge Presentation*, proses visualisasi dan teknik menyajikan *knowledge* digunakan untuk menampilkan *knowledge* hasil *mining* kepada *user*.

2.6 Metode *Data Mining*

Pada umumnya metode *data mining* dapat dikelompokkan kedalam dua kategori yaitu *deskriptif* dan *prediktif*. Metode *deskriptif* bertujuan untuk mencari pola yang dapat dimengeti oleh manusia yang menjelaskan karakteristik dari data. Metode *prediktif* menggunakan ciri-ciri tertentu dari data untuk melakukan prediksi. Metode-metode yang ada dalam *data mining* adalah sebagai berikut:

1. *Classification*

Klasifikasi (*Classification*) merupakan proses untuk menemukan sekumpulan model yang menjelaskan dan membedakan kelas-kelas data, sehingga model tersebut dapat digunakan untuk memprediksi nilai suatu kelas yang belum diketahui pada sebuah objek. Untuk mendapatkan model, kita harus melakukan analisis terhadap data latih (*training set*). Sedangkan data uji (*test set*) digunakan untuk mengetahui tingkat akurasi dari model yang telah dihasilkan. Dalam proses klasifikasi pohon keputusantradisional, fitur (atribut) dari tupel adalah kategorikal atau numerikal. Biasanya definisi ketepan nilai *(point value)* sudah didefinisikan di awal. Pada banyak aplikasi nyata, terkadang muncul suatu nilai yang tidak pasti. (Tsang, Smith., 2009). Klasifikasi dapat digunakan untuk memprediksi nama atau nilai kelas dari suatu objek data. Metode inilah yang digunakan dalam tugas akhir ini.

2. *Clustering*

Pengelompokkan (*Clustering*) merupakan proses untuk melakukan segmentasi. Digunakan untuk melakukan pengelompokkan secara alami terhadap atribut suatu set data. Termasuk kedalam *unsupervised task*. Contoh *clustering* seperti mengelompokkan dokumen berdasarkan topiknya.

3. *Association*

Tujuan dari metode ini yaitu untuk menghasilkan sejumlah rule yang menjelaskan sejumlah data yang terhubung kuat satu dengan yang lainyya. Sebagai contoh *association analysis* dapat digunakan untuk menentukan produk yang datang dibeli secara bersamaan oleh banyak pelanggan atau bisa juga disebut dengan *market basket analysis*.

4. *Regression*

Regression mirip dengan klasifikasi. Perbedaan utamanya adalah terletak pada atribut yang diprediksi berupa nilai yang kontinyu.

5. *Forecasting*

Prediksi (*Forecasting*) berfungsi untuk melakukan prediksi kejadian yang akan datang berdasarkan data sejarah yang ada.

6. *Sequence Analysis*

Tujuan dari metode ini adalah untuk mengenali pola dari data diskrit. Sebagai contoh adalah menemukan kelompok *gen* dengan tingkat ekspresi yang mirip.

7. *Deviation Analysis*

Tujuan dari metode ini adalah untuk menemukan penyebab perbedaan antara data yang satu dengan data yang lain dan biasa disebut sebagai *oulier detection*. Sebagai contoh adalah apakah sudah terjadi penipuan terhadap pengguna kredit dengan melihat catatan transaksi yang tersimpan dalam basis data perusahaan kartu kredit. (Santosa, Budi. 2007).

2.7 Klasifikasi

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu (1) pembangunan model sebagai prototipe untuk disimpan sebagai memori dan (2) penggunaan model tersebut untuk melakukan pengenalan/ klasifikasi /prediksi pada suatu objek data lain agar diketahui dikelas mana objek data tersebut dalam model yang sudah disimpannya. Model dalam klasifikasi mempunyai arti yang sama denga kotak hitam, dimana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut dan memberikan jawaban sebagai keluaran dari hasil pemikiannya. (Prasetyo, E. 2012).

Tahapan dari klasifikasi dalam data mining terdiri dari(Han, J. dan M. Kamber. 2006) :

1. Pembangunan Model

   Pada tahapan ini dibuat sebuah model untuk menyelesaikan masalahklasifikasi class atau attribut dalam data. Tahap ini merupakan fase pelatihan, dimana data latih dianalisis menggunakan algoritma klasifikasi, sehingga model pembelajaran direpresentasikan dalam bentuk aturan klasifikasi.

2. Penerapan Model

   Pada tahapan ini model yang sudah dibangun sebelumnya digunakan untuk menentukan atribut/kelas dari sebuah data baru yang atribut/kelasnya belum diketahui sebelumnya. Tahap ini digunakan untuk memperkirakan keakuratan aturan klasifikasi terhadap data uji. Jika model dapat diterima, maka aturan dapat diterapkan terhadap klasifikasi data baru.

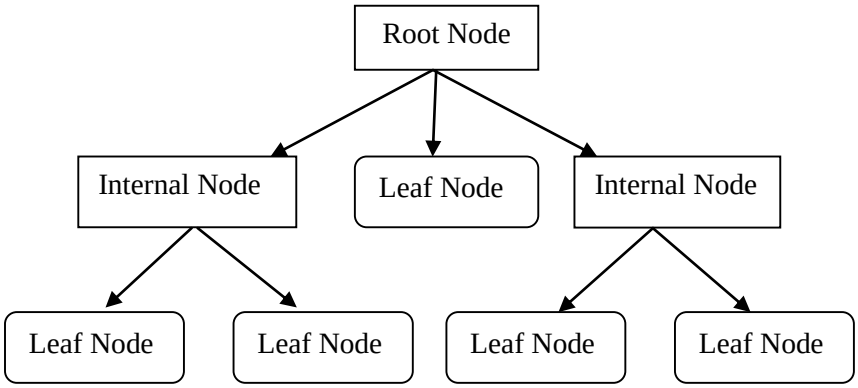
2.8 *Decision Tree*

Decision tree adalah *flow-chart* seperti *struktur tree*, dimana tiap *internal node* menunjukkan sebuah test pada sebuah atribut, tiap cabang menunjukkan hasil dari test, dan *leaf node* menunjukkan *class-class* atau *class distribution*.

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*.

Contoh dari model pohon keputusan yaitu seperti pada **Gambar 2.1** berikut:



Gambar 2.1 Model *Decision Tree*

2.9 *Decision Tree C4.5*

Algoritma C4.5 diperkenalkan oleh Quinlan (1996) sebagai versi perbaikan dari ID3. Dalam ID3, induksi decision tree hanya bisa dilakukan pada fitur bertipe kategorikal (nominal atau ordinal), sedangan tipe numerik (interval atau rasio) tidak dapat digunakan (Eko Prasetyo, 2014).

Yang menjadi hal penting dalam induksi decision tree adalah bagaimana menyatakan syarat pengujian pada node. Ada 3 kelompok penting dalam syarat pengujian node :

1. Fitur biner

   Adalah Fitur yang hanya mempunyai dua nilai berbeda. Syarat pengujian ketika fitur ini menjadi node (akar maupun interval) hanya punya dua pilihan cabang.

2. Fitur kategorikal

   Untuk fitur yang nilainya bertipe kategorikal (nominal atau ordinal) bisa mempunyai beberapa nilai berbeda. Secara umum ada 2 pemecahan yaitu pemecahan biner (*binary splitting*) dan (*multi splitting*).

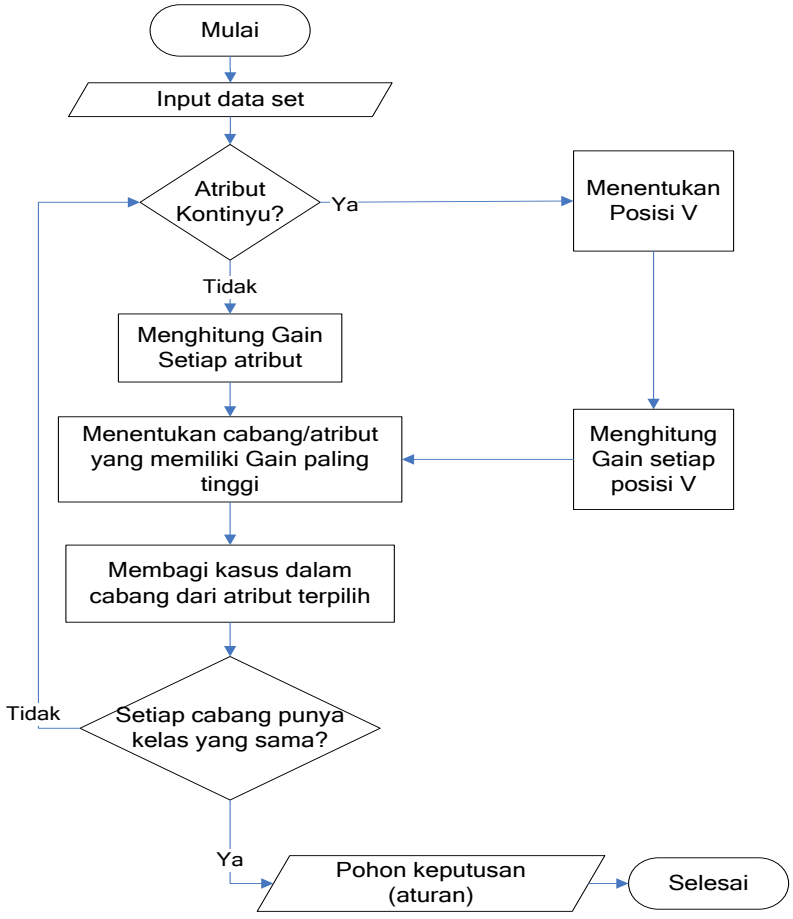
3. Fitur numerik

   Untuk fitur bertipe numerik, Syarat pengujian dalam node (akar maupun internal) dinyatakan dengan pengujian perbandingan ($A \leq V$) atau ($A > V$) dengan hasil biner.

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

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

Berikut ini akan dijelaskan secara lebih detail algoritma C4.5 menggunakan *flowcart* yang disajikan pada gambar 2.2.



Gambar 2.2 *Flowchart* Algoritma *Decision Tree C4.5*

Untuk memilih atribut sebagai simpul akar (*root node*) atau simpul dalam (*internal node*), didasarkan pada nilai *information gain* tertinggi dari atribut-atribut yang ada. Sebelum perhitungan *information gain*, akan dilakukan perhitungan *entropy*. *Entropy* merupakan distribusi probabilitas dalam teori informasi dan diadopsi kedalam algoritma *C*4.5 untuk mengukur tingkat

homogenitas distribusi kelas dari sebuah himpunan data (*data set*). Semakin tinggi tingkat *entropy* dari sebuah data maka semakin homogen distribusi kelas pada data tersebut. Perhitungan *information gain* menggunakan rumus 2.1, sedangkan *entropy* menggunakan rumus 2.2.

$$Gain(S, A) = Entropy\ (S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy\ (S_i) \quad \text{(2.1)}$$

dimana,

S : Himpunan kasus
A : Atribut
n : Jumlah partisi atribut A
|Si| : Jumlah kasus pada partisi ke i
|S| : Jumlah kasus dalam S

$$Entropy(S) = -\sum_{i=1}^{n} p_i * log_2 p_i \quad \text{(2.2)}$$

dimana,

S : Himpunan kasus
A : Fitur
n : Jumlah partisi S
p_i : Proporsi dari S_i terhadap S

Selain *Information Gain* kriteria yang lain untuk memilih atribut sebagai pemecah adalah *Rasio Gain*. Perhitungan *rasio gain* menggunakan rumus 2.3, sedangkan *split information* menggunakan rumus 2.4.

$$GainRasio\ (S, A) = \frac{Gain\ (S,A)}{SplitInformation(S,A)} \quad \text{(2.3)}$$

$$SplitInformation(S, A) = -\sum_{i=1}^{c} \frac{S_i}{S} log_2 \frac{S_i}{S} \quad \text{(2.4)}$$

dimana S_1 sampai S_c adalah c subset yang dihasilkan dari pemecahan S dengan menggunakan atribut A yang mempunyai sebanyak c nilai.

Untuk mengukur nilai akurasi yang didapat dari hasil pengujian, menggunakan rumus 2.5. Sedangkan untuk mengukur tingkat kesalahannya menggunakan rumus 2.6.

$$Akurasi = \frac{Jumlah\ data\ yang\ diprediksi\ secara\ benar}{Jumlah\ prediksi\ yang\ dilakukan} \quad \text{(2.5)}$$

$$Laju\ error = \frac{Jumlah\ data\ yang\ diprediksi\ secara\ salah}{Jumlah\ prediksi\ yang\ dilakukan} \quad \text{(2.6)}$$

Sensitivitas akan mengukur proporsi positif asli yang dikenali (diprediksi) secara benar sebagai positif asli. Rumus perhitungannya menggunakan rumus 2.7. Sedangkan spesifisitas akan mengukur proporsi negatif asli yang dikenali (diprediksi) secara benar sebagai negatif asli. Rumus perhitungannya menggunakan rumus 2.8.

$$Sensitivitas = \frac{TP}{TP + FN} \quad \text{(2.7)}$$

Keterangan :

TP : Kelas acc yang diprediksi secara benar sebagai Kelas acc

FN : Kelas acc yang diprediksi secara salah sebagai Kelas tolak

$$Spesifisitas = \frac{TN}{FP + TN} \quad \text{(2.8)}$$

Keterangan :

TN : Kelas tolak yang diprediksi secara benar sebagai Kelas tolak

FP : Kelas tolak yang diprediksi secara salah sebagai Kelas acc

2.10 Contoh Perhitungan

Berikut ini akan dijelaskan ilustrasi dari alur proses perhitungan algoritma *Decision Tree C4.5*. Data set yang digunakan pada contoh ini adalah data untuk melakukan prediksi "apakah harus bermain *baseball* ?" dengan menjawab Ya atau Tidak. Atribut yang digunakan ada 4 yaitu Cuaca, Suhu, Kelembaban, dan Angin. Dimana atribut Suhu dan Kelembaban bertipe numerik sedangkan Cuaca dan Angin bertipe kategorikal. Sedangkan kolom bermain adalah kelas tujuannya atau label kelasnya.

Tabel 2.1 Contoh data set

Cuaca	Suhu	Kelembaban	Angin	Bermain
Cerah	85	85	Biasa	Tidak
Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Hujan	70	96	Biasa	Ya
Hujan	68	80	Biasa	Ya
Hujan	65	70	Kencang	Tidak
Mendung	64	65	Kencang	Ya
Cerah	72	95	Biasa	Tidak

Tabel 2.1 Contoh Data Set (lanjutan)

Cuaca	Suhu	Kelembaban	Angin	Bermain
Cerah	69	70	Biasa	Ya
Hujan	75	80	Biasa	Ya
Cerah	75	70	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya
Hujan	71	80	Kencang	Tidak

Proses pertama adalah menghitung *entropy* untuk *node* akar (semua data) terhadap komposisi kelas.

Berikut adalah perhitungan *entropy* untuk semua data:

$$Entropy(S) = -\frac{9}{14} * log_2\left(\frac{9}{14}\right) - \frac{5}{14} * log_2\left(\frac{5}{14}\right)$$

$$= 0.9403$$

Selanjutnya, untuk fitur yang bertipe *numeric*, harus ditentukan posisi *v* yang terbaik untuk pemecahan. Dalam contoh ini, digunakan pemecahan biner. Hasil uji coba pada fitur Suhu dengan menghitung nilai Gainnya disajikan pada **Tabel 2.2**. Nilai Gain tertinggi didapatkan pada posisi *v* = 70. Maka untuk fitur Suhu dilakukan diskretisasi pada *v* = 70 ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Tabel 2.2 Posisi *v* Untuk Pemecahan Fitur Suhu Di *Node* Akar

Suhu	**70**		**75**		**80**	
	<=	>	<=	>	<=	>
Ya	4	5	7	2	7	2
Tidak	1	4	3	2	4	1
Gain	**0.0453**		**0.0251**		**0.0005**	

Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada **Tabel 2.3**. Nilai *Gain* tertinggi didapatkan pada posisi *v* = 80. Maka untuk fitur Kelembaban dilakukan diskretisasi pada *v* = 80 ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Tabel 2.3 Posisi *v* untuk pemecahan fitur Kelembaban di *node* akar

Kelembaban	**70**		**75**		**80**		**85**	
	<=	>	<=	>	<=	>	<=	>
Ya	2	7	3	6	7	2	7	2
Tidak	1	4	1	4	2	3	3	2
Gain	**0.0005**		**0.0150**		**0.1022**		**0.0251**	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung gain untuk setiap fitur. Hasilnya disajikan pada **Tabel 2.4.**

Tabel 2.4 Hasil Perhitungan *Entropy* dan *Gain* untuk *Node* Akar

Node			**Jumlah**	**Ya**	**Tidak**	**Entropy**	**Gain**
1	Total		14	9	5	0.9403	
	Cuaca						0.2467
		Cerah	5	2	3	0.9710	
		Mendung	4	4	0	0	
		Hujan	5	3	2	0.9710	
	Suhu						0.0453
		<=70	5	4	1	0.7219	
		>70	9	5	4	0.9911	
	Kelembaban						0.1022
		<=80	9	7	2	0.7642	
		>80	5	3	2	0.9710	
	Angin						0.0481
		Pelan	8	6	2	0.8113	
		Kencang	6	3	3	0.8113	

Hasil yang didapat di **Tabel 2.4** menunjukkan bahwa *Gain* tertinggi ada di fitur Cuaca, maka Cuaca dijadikan sebagai *node* akar. Selanjutnya, dihitung

posisi *split* untuk fitur Cuaca dengan menghitung *Rasio Gain*, selengkapnya disajikan pada tabel 2.5.Hasil peritungan *rasio gain* posisi *split* untuk *opsi* satu sebagai berikut:

$$SplitInfo(Semua, cuaca) = \left(-\frac{5}{14} * log_2\left(\frac{5}{14}\right)\right) + \left(-\frac{4}{14} * log_2\left(\frac{4}{14}\right)\right) + \left(-\frac{5}{14} * log_2\left(\frac{5}{14}\right)\right) = 1.5774$$

$$RasioGain(Semua, cuaca) = \frac{0.2467}{1.5774} = 0.16$$

Dengan cara yang sama, akan didapatkan nilai *rasio gain* untuk *opsi* yang lain.Hasil di **Tabel 2.5** menunjukkan bahwa *rasio gain* tertinggi ada di *opsi* 4 yaitu *split* {cerah, hujan} dengan {mendung}. Itu artinya, cabang untuk akar ada 2, yaitu: {cerah, hujan} dan {mendung}, seperti ditunjukkan pada **Gambar 2.3.**

Tabel 2.5 Perhitungan *Rasio Gain* untuk Fitur Cuaca

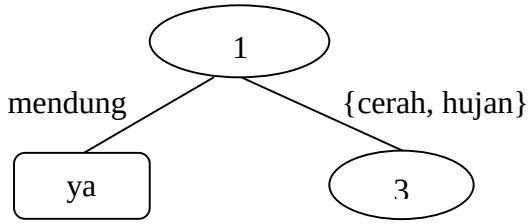
Node			**Jumlah**	**Entropy**	**Gain**	**Rasio Gain**
1	Total		14		0.2467	
Opsi 1	Cuaca	Cerah Mendung Hujan	5 4 5	1.5774		0.16
Opsi 2	Cuaca	Cerah Mendung dan Hujan	5 9	0.9403		0.26
Opsi 3	Cuaca	Cerah, Mendung Hujan	9 5	0.9403		0.26
Opsi 4	Cuaca	Cerah, Hujan Mendung	10 4	0.8631		0.29

Hasil pemisahan data menurut *node* akar disajikan pada **Tabel 2.6.**

Tabel 2.6 Pemisahan Data Menurut *Node* Akar

Cuaca	**Suhu**	**Kelembaban**	**Angin**	**Bermain**
Cerah	85	85	Biasa	Tidak
Cerah	80	90	Kencang	Tidak
Hujan	70	96	Biasa	Ya
Hujan	68	80	Biasa	Ya
Hujan	65	70	Kencang	Tidak
Cerah	72	95	Biasa	Tidak
Cerah	69	70	Biasa	Ya
Hujan	75	80	Biasa	Ya
Cerah	75	70	Kencang	Ya
Hujan	71	80	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Mendung	64	65	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya

Untuk *node* 2, nilai *Entropy* yang didapat adalah 0 (karena semua baris memiliki kelas yang sama) maka dipastikan bahwa node 2 menjadi daun, seperti ditunjukkan pada **Gambar 2.3.**



Gambar 2.3 Hasil Pembentukan Cabang Di Akar Untuk Kasus "Apakah Harus Bermain *Baseball* ?"

Selanjutnya, di *node* 3, harus dihitung dulu *entropy* untuk sisa data terhadap komposisi kelas yang tidak masuk dalam *node* 2. Untuk fitur yang bertipe numerik, harus ditentukan lagi posisi *v* yang terbaik untuk pemecahan. Dalam contoh ini, digunakan pemecahan biner. Hasil uji coba pada fitur Suhu dengan menghitung nilai *Gain* yang disajikan pada **Tabel 2.7.** Nilai *Gain* tertinggi didapatkan pada posisi $v = 75$. Maka untuk fitur Suhu dilakukan diskretisasi pada $v = 75$ ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Tabel 2.7 Posisi *v* untuk Pemecahan Fitur Suhu di *Node* 3

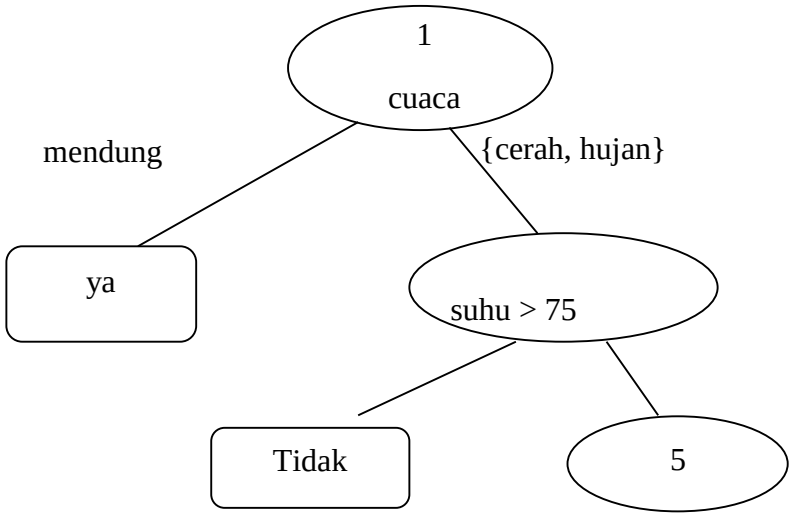
Suhu	**70**		**75**		**80**	
	<=	>	<=	>	<=	>
Ya	3	2	5	0	5	0
Tidak	1	4	3	2	4	1
Gain	**0.1245**		**0.2365**		**0.1080**	

Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada **Tabel 2.8**. Nilai *Gain* tertinggi didapatkan pada posisi *v* = 80. Maka untuk fitur Kelembaban dilakukan diskretisasi pada *v* = 80 ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Node			**Jumlah**	**Ya**	**Tidak**	**Entropy**	**Gain**
3	Total		10	5	5	1.0000	
	Cuaca						0.0290
		Cerah	5	2	3	0.9710	
		Hujan	5	3	2	0.9710	
	Suhu						0.2365
		<=75	8	5	3	0.9544	
		>75	2	0	2	0	
	Kelembaban						0.1245
		<=80	6	4	2	0.9183	
		>80	4	1	3	0.8113	
	Angin						0.1245
		Pelan	6	4	2	0.9183	
		Kencang	4	1	3	0.8113	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada **Tabel 2.9.** Hasil pada tabel tersebut menunjukkan bahwa *gain* tertinggi ada di fitur Suhu,

berarti fitur Suhu dijadikan syarat kondisi di *node* 3, seperti ditunjukan pada **Gambar 2.4.** Pemisahan datanya ditunjukkan pada **Tabel 2.10.**



Gambar 2.4 Hasil Pembentukan Cabang di *Node* 3 untuk Kasus "Apakah Harus Bermain *Baseball*

Tabel 2.10 Pemisahan Data Menurut *Node* 3

Cuaca	**Suhu**	**Kelembaban**	**Angin**	**Bermain**
Hujan	70	96	Pelan	Ya
Hujan	68	80	Pelan	Ya
Hujan	65	70	Kencang	Tidak
Cerah	72	95	Pelan	Tidak
Cerah	69	70	Pelan	Ya
Hujan	75	80	Pelan	Ya
Cerah	75	70	Kencang	Ya
Hujan	71	80	Kencang	Tidak
Cerah	85	85	Pelan	Tidak
Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Mendung	64	65	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya

Node 4, untuk cabang suhu > 75 dimana label kelas bernilai tidak, dipastikan mempunyai entropy 0, maka *node* 4 (yang dituju) dijadikan daun. Seperti ditunjukkan pada **Gambar 2.4.**

Selanjutnya pada *node* 5, untuk fitur numerik kembali dilakukan perhitungan posisi *v* yang terbaik untuk pemecahan. Dalam contoh ini, digunakan pemecahan biner. Hasil uji coba pada fitur Suhu dengan menghitung nilai *Gain* yang disajikan pada **Tabel 2.11**. Nilai *Gain* tertinggi didapatkan pada posisi *v* = 70. Maka untuk fitur Suhu dilakukan diskretisasi pada *v* = 70 ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Tabel 2.11 Posisi *v* untuk Pemecahan Fitur Suhu di *Node* 5

Suhu	**70**		**75**	
	<=	>	<=	>
Ya	3	2	5	0
Tidak	1	2	3	0
Gain	**0.0488**		**0**	

Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada **Tabel 2.12**. Nilai *Gain* tertinggi didapatkan pada posisi *v* = 80. Maka untuk fitur Kelembaban dilakukan diskretisasi pada *v* = 80 ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Tabel 2.12 Posisi *v* untuk Pemecahan Fitur Kelembaban di *Node* 5

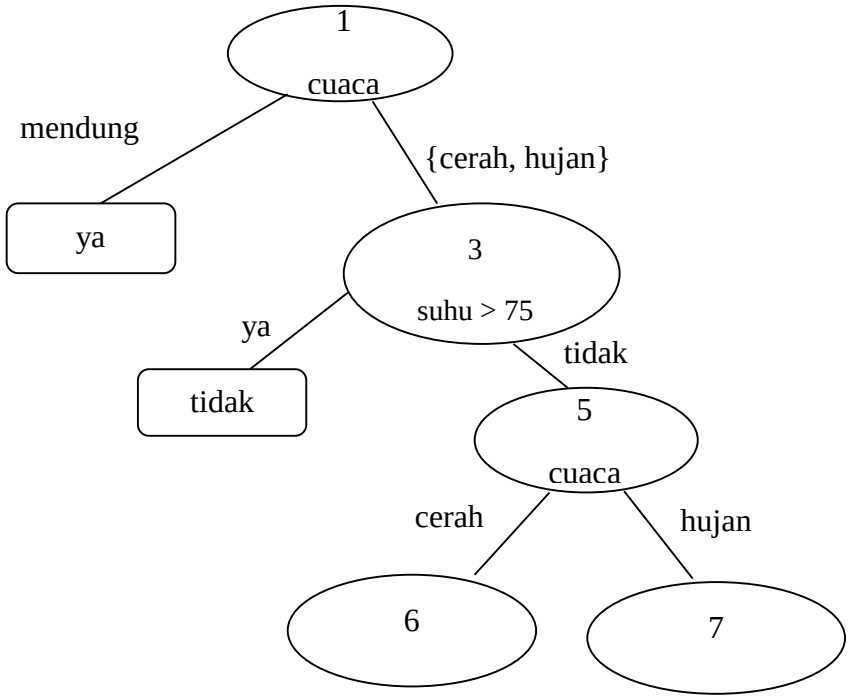
Kelembaban	**70**		**75**		**80**	
	<=	>	<=	>	<=	>
Ya	2	3	2	3	4	1
Tidak	1	2	1	2	2	1
Gain	**0.0032**		**0.0032**		**0.0157**	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada **Tabel 2.13.**

Tabel 2.13 Hasil perhitungan *entropy* dan *gain* untuk *node* 5

Node			**Jumlah**	**Ya**	**Tidak**	**Entropy**	**Gain**
5	Total		8	5	3	0.9544	
	Cuaca						0.2013
		Cerah	3	2	3	0.3900	
		Hujan	5	3	2	0.9710	

Node			Jumlah	Ya	Tidak	Entropy	Gain
	Suhu						0.0488
		<=70	4	3	1	0.8113	
		>70	4	2	2	1.0000	
	Kelembaban						0.0157
		<=80	6	4	2	0.9183	
		>80	2	1	1	1.0000	
	Angin						0.1589
		Pelan	5	4	1	0.7219	
		Kencang	3	1	2	0.9183	



Gambar 2.5 Hasil Pembentukan Cabang di *Node* 5 untuk Kasus Apakah Harus Bermain *Baseball*

Tabel 2.14 Pemisahan data menurut *node* 5

Cuaca	Suhu	Kelembaban	Angin	Bermain
Cerah	72	95	Pelan	Tidak
Cerah	69	70	Pelan	Ya
Cerah	75	70	Kencang	Ya
Hujan	70	96	Pelan	Ya
Hujan	68	80	Pelan	Ya
Hujan	65	70	Kencang	Tidak
Hujan	75	80	Pelan	Ya

Cuaca	Suhu	Kelembaban	Angin	Bermain
Hujan	71	80	Kencang	Tidak
Cerah	85	85	Pelan	Tidak
Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Mendung	64	65	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya

Pada perhitungan berikutnya, fitur Cuaca tidak digunakan lagi karena kedua nilai berbeda yang tersisa sudah digunakan untuk syarat pengujian di *node* 5. Selanjutnya pada *node* 6, untuk fitur numerik kembali dilakukan perhitungan posisi *v* yang terbaik untuk pemecahan. Dalam contoh ini, digunakan pemecahan biner. Hasil uji coba pada fitur Suhu dengan menghitung nilai *Gain* yang disajikan pada **Tabel 2.15.** Nilai *Gain* tertinggi didapatkan pada posisi *v* = 70. Maka untuk fitur Suhu dilakukan diskretisasi pada *v* = 70 ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Tabel 2.15 Posisi *v* untuk Pemecahan Fitur Suhu di *Node* 6

Suhu	**70**		**75**	
	<=	>	<=	>
Ya	1	1	2	0
Tidak	0	1	1	0
Gain	**0.2516**		0	

Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada **Tabel 2.16.** Nilai *Gain* didapatkan pada posisi *v* = 70. Maka untuk fitur Kelembaban dilakukan diskretisasi pada *v* = 70 ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Tabel 2.16 Posisi *v* untuk pemecahan fitur Kelembaban di *node* 6

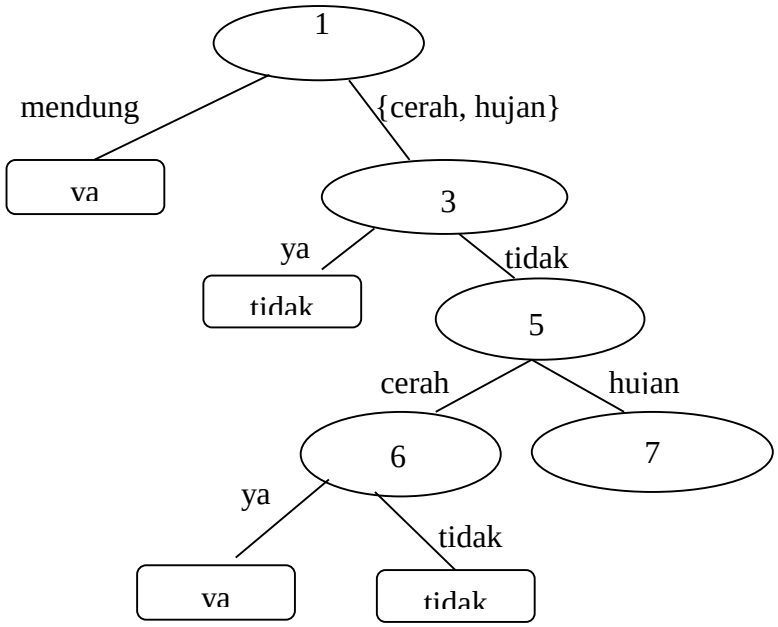
Kelembaban	**70**	
	<=	>
Ya	2	0
Tidak	0	1
Gain	**0.9183**	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada **Tabel 2.17.**

Tabel 2.17 Hasil Perhitungan *Entropy* dan *Gain* untuk *Node* 6

Node			**Jumlah**	**Ya**	**Tidak**	**Entropy**	**Gain**
6	Total		3	2	1	0.9183	
	Suhu						0.2516
		<=70	1	1	0	0	
		>70	2	1	1	1.0000	
	Kelembaban						0.9183
		<=70	2	2	0	0	
		>70	1	0	1	0	
	Angin						0.2516
		Pelan	2	1	1	1.0000	
		Kencang	1	1	0	0	

Hasil yang ditunjukkan pada **Tabel 2.17** menunjukkan bahwa *gain* tertinggi ada di fitur Kelembaban, berarti fitur Kelembaban dijadikan syarat kondisi di *node* 6, seperti ditunjukan pada **Gambar 2.6.** Pemisahan datanya ditunjukkan pada **Tabel 2.18**. Jika diamati **Tabel 2.18,** untuk *node* 8 dan 9 (cabang dari *node* 6) dipastikan menjadi daun karena nilai *entropy* 0, dimana masing-masing cabang jatuh pada label kelas yang sama. Proses berikutnya dilanjutkan untuk *node* 7. *Gain* tertinggi didapatkan hanya pada posisi *v* = 70.



Gambar 2.6 Hasil Pembentukan Cabang di *node* 6 untuk Kasus "Apakah Harus Bermain *Baseball*

Tabel 2.18 Pemisahan Data Menurut *Node* 6

Cuaca	**Suhu**	**Kelembaban**	**Angin**	**Bermain**
Hujan	70	96	Pelan	Ya
Hujan	68	80	Pelan	Ya
Hujan	65	70	Kencang	Tidak
Hujan	75	80	Pelan	Ya
Hujan	71	80	Kencang	Tidak
Cerah	69	70	Pelan	Ya
Cerah	75	70	Kencang	Ya
Cerah	72	95	Pelan	Tidak
Cerah	85	85	Pelan	Tidak
Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Mendung	64	65	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya

Tabel 2.19 Posisi *v* untuk pemecahan fitur Suhu di *node* 7

Suhu	70	
	<=	>
Ya	2	1
Tidak	0	1
Gain	**0.0200**	

Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada **Tabel 2.20.** Nilai *Gain* didapatkan hanya pada posisi $v = 80$. Maka untuk fitur Kelembaban dilakukan diskretisasi pada $v = 80$ ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Tabel 2.20 Posisi *v* untuk Pemecahan Fitur Kelembaban di *Node* 7

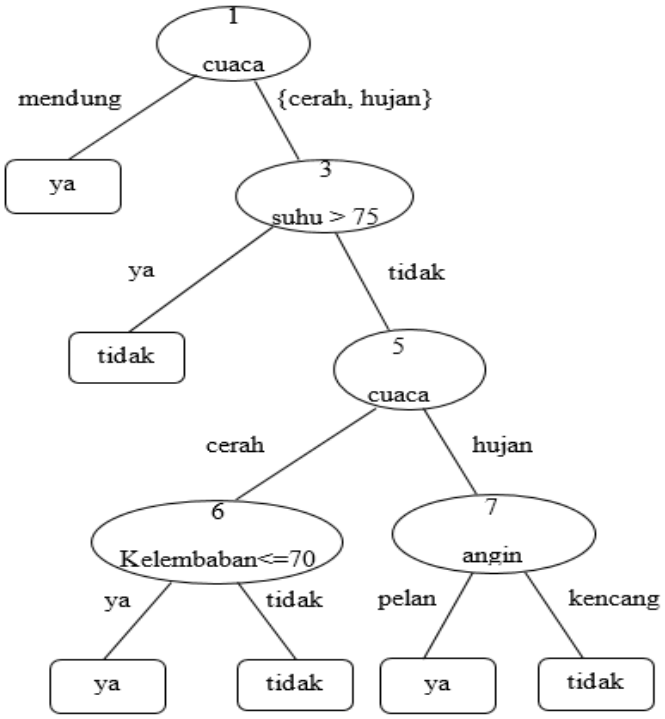
Kelembaban	**70**	
	<=	>
Ya	2	1
Tidak	2	0
Gain	**0.1710**	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada **Tabel 2.21.**

Tabel 2.21 Hasil Perhitungan *Entropy* dan *Gain* untuk *Node* 7

Node			Jumlah	Ya	Tidak	Entropy	Gain
7	Total		5	3	2	0.9710	
	Suhu						0.0200
		<=70	3	2	1	0.9183	
		>70	2	1	1	1.0000	
	Kelembaban						0.1710
		<=80	4	2	2	1.0000	
		>80	1	1	0	0	
	Angin						0.9710
		Pelan	3	3		0	
		Kencang	2	0	2	0	

Hasil yang ditunjukkan pada **tabel 2.21** menunjukkan bahwa *gain* tertinggi ada di fitur Angin, berarti fitur Angin dijadikan syarat kondisi di *node* 7, seperti ditunjukan pada **Gambar 2.8.** Pemisahan datanya ditunjukkan pada **Tabel 2.22.**



Gambar 2.7 Hasil Pembentukan Cabang di *Node* 7 untuk Kasus "Apakah Harus Bermain *Baseball*

Tabel 2.22 Pemisahan Data Menurut *Node* 7

Cuaca	**Suhu**	**Kelembaban**	**Angin**	**Bermain**
Hujan	70	96	Pelan	Ya
Hujan	68	80	Pelan	Ya
Hujan	75	80	Pelan	Ya
Hujan	65	70	Kencang	Tidak
Hujan	71	80	Kencang	Tidak
Cerah	69	70	Pelan	Ya
Cerah	75	70	Kencang	Ya
Cerah	72	95	Pelan	Tidak
Cerah	85	85	Pelan	Tidak
Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Mendung	64	65	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya

Jika diamati **Tabel 2.21,** untuk *node* 10 dan 11 (cabang dari *node* 7) dipastikan menjadi daun karena nilai *entropy* 0, dimana masing-masing cabang jatuh pada label kelas yang sama.

Karena tidak ada lagi node yang harus diproses, maka induksi *decision tree* dinyatakan selesai. Hasil akhir *decision tree* seperti disajikan pada gambar 2.8.

Bentuk aturan *IF THEN* untuk *decision tree* sebagai berikut:

IF cuaca= mendung *THEN playball* = ya

IF cuaca= {cerah, hujan} *AND* suhu > 75 *THEN playball* = tidak

IF cuaca= cerah *AND* suhu <=75 *AND* kelembaban<=70 *THEN playball* = ya

IF cuaca=cerah *AND* suhu<=75 *AND* kelembaban >70 *THEN playball*= tidak

IF cuaca= hujan *AND* suhu <=75 *AND* angin = pelan *THEN playball* = ya

IF cuaca= hujan *AND* suhu<=75 *AND* angin= kencang *THEN playball* = tidak

2.11 Penelitian Sebelumnya

Penelitian sebelumnya yang menggunakan metode *decision tree C4.5* adalah penelitian yang berjudul "Penerapan Algoritma *Decision Tree C4.5* Untuk Diagnosa Penyakit Stroke Dengan Klasifikasi Data Mining Pada Rumah Sakit Santa Maria Pemalang". Penelitian ini disusun oleh Sigit Abdullah yang

berasal dari Program Studi Teknik Informatika Universitas Dian Nuswantoro Semarang. Dalam penelitian ini atribut yang digunakan adalah Jenis Kelamin, Umur, Hipertensi, serta Diabetes dengan variabel target klasifikasi keputusan berupa Stroke atau Non Stroke. Data yang digunakan dalam penelitian ini yaitu sebanyak 156 data pasien yang diperoleh dari rumah sakit, data tersebut dibagi menjadi dua bagian yaitu data training sebanyak 130 data dan sisanya pada data testing yang berjumlah 26 data. Dari metode klasifikasi data mining dengan algoritma C4.5 diperoleh akurasi sebesar 82,31% untuk data training sedangkan akurasi pada data testing sebesar 76,92%. Perhitungan tingkat akurasi keduanya menggunakan confusion matrix.

Liliana Swastina melakukan penelitian dengan judul *"Penerapan Algoritma C4.5 Untuk Penentuan Jurusan Mahasiswa"*. Setelah dilakukan *preprocessing*, data dihitung dengan metode *decision tree C4.5*. Adapun data yang diambil dalam penelitian ini adalah data mahasiswa baru STMIK Indonesia Banjarmasin tahun 2008 s.d 2009. Atribut yang digunakan adalah atribut nama, jenis kelamin, umur, asal sekolah, jurusan asal sekolah, nilai UAN, IPK semester 1, IPK semester 2. Sebanyak 90 % data akan digunakan untuk membangun struktur pohon keputusan melalui metode C4.5. Sedangkan 10 % lainnya digunakan sebagai data uji. Uji pertama melalui data sample yaitu data angkatan 2008, field data NRP, nama, tempat lahir dan tanggal lahir dihilangkan untuk mendapatkan akurasi yang lebih tinggi. Selain itu, Untuk membentuk pohon keputusan maka atribut IPK Semester 1 dan IPK Semester 2 perlu di klasifikasi menjadi: IPK>= 3,00 tergolong kelas A, IPK>= 2,75 tergolong kelas B, dan IPK< 2,75 tergolong kelas C. Kesimpulan dari penelitian tersebut, yakni: dari hasil uji, algoritma *decision tree C4.5* memprediksi lebih akurat dari pada algoritma *naive bayes* dalam penentuan kesesuaian jurusan dan rekomendasi jurusan mahasiswa dan algoritma *Decision Tree C4.5* akurat diterapkan untuk penentuan kesesuaian jurusan mahasiswa dengan tingkat keakuratan 93,31 % dan akurasi rekomendasi jurusan sebesar 82,64%.

Penelitian selanjutanya dilakukan oleh Lailatul Qomariyah, Mahasiswa Teknik Informatika Universitas Muhammadiyah Gresik yang dilakukan tahun 2016. Penelitian ini berjudul "Klasifikasi Calon Pendonor Darah Dengan Metode Decision Tree C4.5 Di Kabupatem Gresik (Studi Kasus: PMI Kabupaten Gresik). Penelitian ini bertujuan untuk menentukan apakah calon pendonor darah dapat melakukan donor darah atau tidak. Atribut yang digunakan dalam penelitian ini yaitu usia, kadar hemoglobin, berat badan dan tekanan darah. Data yang digunakan sebanyak 60 data yang diperoleh dari PMI Kabupaten Gresik. Untuk melakukan pengujian data dilakukan penghitungan nilai akurasi, laju error, sensitivitas dan spesifitas, pengujian dilakukan sebanyak 2 kali dalam 3 variasi data. Variasi data pertama yaitu 30 data latih dan 30 data uji, variasi kedua yaitu 36 data latih dan 24 data uji, dan variasi ketiga sebayak 48 data latih dan 12 data uji. Dari ketiga variasi percobaan tersebut diperoleh rata-rata nilai akurasi sebesar 86,80%, rata-rata laju error sebesar 13,19%, rata-rata sensitivitas sebesar 91,39%, dan rata-rata spesifitas sebesar 82,22%. Dari ketiga variasi peecobaan tersebut diketahui nilai akurasi tertinggi terjadi pada percobaan dengan 48 data latih dan 12 data uji dengan nilai akurasi mencapai 91,67%.

Sedangkan penelitian lain yang digunakan sebagai rujukan dalam tugas akhir ini adalah penelitian yang dibuat oleh Sugarwanto Atmaja, Mahasiswa Teknik Informatika Universitas Muhammadiyah Gresik yang dilakukan pada tahun 2016. Penelitian yang dibuat berjudul "Sistem Pendukung Keputusan Untuk Deteksi Dini Resiko Penyakit Stroke Menggunakan Learning Vector Quantization". Adapun data yang dipakai dalam penelitian ini berasal dari data rekam medis pasien dengan diagnosa dokter umum Puskesmas Glagah Lamongan tahun 2014-2015 sebanyak 128 data pasien dengan diagnosa resiko stroke rendah, sedang dan tinggi. Atribut yang digunakan dalam penelitian ini yaitu berupa tekanan darah, kadar gula darah, kolesterol total, Low Density Lipoprotein (LDL), usia, jenis kelamin, asam urat, Blood Urea Nitrogen (BUN), dan kreatinin. Dari 128 data yang ada kemudian dibagi kedalam data latih dan data uji. Data latih diambil kurang lebih 70% dari jumlah data setiap

kelas dan data uji kurang lebih 30% dari jumlah data setiap kelas. Pengujian dalam penelitian ini dilakukan sebanyak 4 kali dengan laju pembelajaran yang berbeda-beda. Pada percobaan 1 sampai percobaan 3 mempunyai tingkat akurasi total yang sama yaitu sebesar 82% untuk pengujian dengan data uji. Sedangkan pada percobaan 4 nilai tingkay akurasi total hanya 79%. Nilai parameter yang digunakan pada algoritma LVQ ini meliputi laju pembelajaran 0.04 atau 0.05, fungsi pembelajaran 0.6, jumlah otetasi 7, minimal alfa 0.001 dan bobot yang digunakan 0.5. Dari nilai nilai parameter tersebut didapatkan akurasi total sebesar 82%. Namun dalam penelitian ini, untuk hasil prediksi sistem kategori status sedang masih terdistorsi dengan status rendah dan lebih banyak diprediksi dalam kategori status tinggi.