BAB 2 LANDASAN TEORI

Bab ini berisi penjelasan mengenai sejumlah teori yang digunakan penulis dalam penelitian ini. Adapun teori yang dijelaskan meliputi sistem penunjang keputusan, metode prakiraan cuaca jangka pendek, serta mengenai metode pohon keputusan yang akan digunakan sebagai model prakiraan dalam penelitian ini.

2.1 Landasan Teori Prakiraan Cuaca Jangka Pendek

Proses prakiraan cuaca yang berlangsung di Indonesia dan dilakukan oleh Badan Meteorologi Klimatologi dan Geofisika didasarkan pada pengamatan 178 jejaring stasiun BMKG di seluruh Indonesia. Data meteorologi ini digunakan sebagai dasar pembuatan prakiraan cuaca harian, mingguan, dan bahkan bulanan untuk kebutuhan berbagai macam penggunanya seperti maskapai penerbangan maupun industri pelayaran.

Badan Meteorologi Klimatologi dan Geofisika sendiri menerbitkan dua jenis prakiraan dalam setahun yaitu Prakiraan Musim Hujan (awal bulan September) dan Prakiraan Musim Kemarau (awal bulan Maret) [9].

Proses pembuatan prakiraan cuaca harian yang dilakukan oleh BMKG telah mengikuti standar Internasional *World Meteorological Organization* (WMO) sebagai berikut:

- Memperhatikan unsur cuaca 24 jam yang lalu, dan unsur cuaca yang sedang terjadi (peta sipnotik). Tujuan dari kegiatan ini adalah untuk mengetahui apakah ada unsur cuaca yang cukup ekstrem
- Membuat kontur tekanan udara. Tujuan dari kegiatan ini adalah untuk mengetahui sumber massa udara yang mendukung pertumbuhan awan
- Membuat gambar angin (*streamline)* pada lapisan permukaan hingga pada lapisan 20,000 kaki bahkan lebih. Tujuan dari kegiatan ini adalah untuk memantau pergerakan massa udara apakah ikut berinteraksi dengan massa udara pada daerah yang dilalui.
- Membuat kontur kelembaban dan suhu udara, untuk memantau tingkat kebasahan atmosfer

- Membuat prakiraan model tekanan, angin, kelembaban, suhu udara, dan curah hujan
- Memperhatikan ada atau tidaknya badai tropis yang tumbuh di dekat perairan Indonesia
- Memantau satelit awan dan radar awan atau hujan untuk memantau distribusi awan dan hujan
- Memprakirakan cuaca 1 hingga 3 hari dan 1 minggu ke depan.

Dalam prakiraan cuaca jangka pendek, kondisi cuaca yang terjadi pada hari ini tidak dapat digunakan untuk memprakirakan cuaca pada 1 atau 2 bulan yang akan datang tapi cenderung untuk memprakirakan cuaca 1 hingga 2 hari ke depan saja [17]. Proses prakiraan cuaca jangka pendek dalam penelitian ini adalah proses prakiraan cuaca untuk 1 hingga 3 hari ke depan dengan memperhatikan 7 buah faktor atau unsur cuaca yang diamati setiap hari oleh Stasiun Meteorologi 745 Kemayoran Jakarta. Proses pengamatan yang dilakukan oleh prakirawan di stasiun cuaca tersebut berlangsung setiap jamnya dengan mengukur sejumlah unsur cuaca yakni:

- Temperatur udara
- Kelembaban udara
- Tekanan udara
- Arah angin
- Kecepatan angin
- Curah hujan
- Lama penyinaran matahari

2.1.1 Proses Pengukuran Unsur Cuaca

Pengukuran untuk masing-masing unsur cuaca tersebut dilakukan sebagai berikut:

- Temperatur udara

  Dengan menggunakan *Psychrometer Standard* yang terdiri dari 4 buah *termometer* yaitu *termometer* maksimum, *termometer* minimum, *termometer* bola kering (BK), serta *termometer* bola basah (BB). Temperatur udara

maksimum dapat dilihat pada *termometer* maksimum sedangkan temperatur udara minimum dapat dilihat pada *termometer* minimum yang ada. Sedangkan untuk melihat temperatur udara pada saat pengamatan dilakukan dapat dilihat pada *termometer* bola kering (BK) [7].

- Kelembaban udara

Masih dengan menggunakan *Psychrometer Standard*, pengukuran kelembaban udara dilakukan dengan melihat termometer bola basah dan bola kering. Selisih dari angka yang ditunjukkan pada bola kering dan bola basah kemudian dibandingkan dengan angka persentase yang terdapat dalam tabel kelembaban relatif [7].

- Tekanan udara

Pengukuran tekanan udara dilakukan dengan menggunakan *barometer* dalam satuan *milibar* (mb). *Barometer* yang digunakan dilengkapi dengan *termometer* untuk mengetahui suhu yang ada pada ruangan pengamatan. *Barometer* ini tidak boleh terkena sinar matahari dan angin secara langsung serta dipasang tegak lurus dengan ketinggian bejana 1 meter dari lantai [7]. Pada dasarnya tekanan atmosfer berbeda antara satu tempat dengan tempat lainnya dan dari waktu ke waktu. Pada ketinggian permukaan laut, rentang nilai tekanan udara ini berkisar antara 970 hingga 1040 mb. Karena tekanan menurun seiring dengan kenaikan ketinggian dari permukaan laut (berbanding terbalik dengan ketinggian dari permukaan laut) maka tekanan udara hasil observasi pada stasiun yang berbeda-beda harus disesuaikan dengan ketinggiannya dari permukaan laut [11].

- Arah angin

Arah angin dapat dilihat dari *anemometer* dengan ketinggian 10 meter. Arah angin yang dimaksud adalah arah darimana angin berhembus. Adapun arah angin yang dijadikan tolak ukur ada 8 penjuru yaitu utara, selatan, barat, timur, barat daya, barat laut, tenggara, dan timur laut. [7]

- Kecepatan angin

Kecepatan angin diukur dengan menggunakan *cup counter anemometer* dengan prinsip kerja seperti *speedometer* yang ada pada kendaraan bermotor.

Satuan yang digunakan pada alat ini adalah km per jam. Sebagai konvensi, 1 knot kecepatan angin sama dengan 1.8 km per jam. [7]

- Curah hujan

Untuk mengukur curah hujan digunakan sejumlah alat yang memiliki fungsi yang sama namun cara kerja berbeda. Alat tersebut antara lain penakar hujan otomatis (Hellman) dimana dengan alat ini dapat diketahui waktu terjadi dan berakhirnya hujan dan keluaran yang dihasilkan adalah berupa grafik. Grafik terjal menunjukkan hujan dengan intensitas lebat sedangkan grafik landai menunjukkan hujan dengan intensitas ringan. Alat yang ke-2 adalah *ombrometer* dimana curah hujan diukur dengan gelas penakar dan pengamatan dilakukan setiap 3 jam sekali. Satuan untuk alat ini adalah *millimeter* (mm) dimana 1 mm sama dengan 10 cc [7].

- Lama penyinaran matahari

Stasiun pengamatan cuaca BMKG mencatat jumlah atau persentase lama penyinaran matahari setiap harinya mulai dari jam 08.00 hingga 16.00. Instrumen yang digunakan dalam pengamatan lama penyinaran matahari ini disebut *Campbell-Stokes*. Alat ini terdiri dari sebuah bola kaca berisi air yang memfokuskan cahaya matahari sehingga membakar kartu indeks (pias) dan meninggalkan lubang pembakaran pada kartu tersebut. Seiring dengan pergerakan matahari, lubang hasil pembakaran tersebut juga ikut bergerak dan menunjukkan berapa lama waktu penyinaran yang terjadi pada hari itu [11]. Jenis pias yang digunakan pun terdiri dari 3 macam yaitu lengkung panjang (digunakan pada 11 Oktober hingga 28 Februari), lurus (digunakan pada 11 September hingga 10 Oktober dan 1 Maret hingga 10 April), dan lengkung pendek (digunakan pada 11 April hingga 10 Agustus). Selain *Campbell-Stokes*, dapat pula digunakan *actinograph bimetal* yang mengukur intensitas penyinaran matahari secara otomatis dengan satuan pengukuran *K Cal/m*^{2} (Langley).

2.2 Landasan Teori Sistem Penunjang Keputusan

2.2.1 Definisi Sistem Penunjang Keputusan

Sistem penunjang keputusan didefinisikan sebagai suatu sistem yang ditujukan untuk mendukung manajer suatu organisasi dalam mengambil keputusan dalam situasi keputusan yang kurang terstruktur. Berikut ini adalah sejumlah definisi dari sistem penunjang keputusan [14]:

- *Little*

  "Himpunan prosedur berbasiskan model pemrosesan data untuk membantu manajer dalam mengambil keputusan"

- *Moore and Chang*

  "Sistem yang dapat diperluas dan mampu menganalisis data ad hoc serta memodelkan keputusan, dengan orientasi kepada perencanaan di masa yang akan datang"

- *Bonczek*

  "Sistem berbasiskan komputer yang terdiri dari tiga komponen dasar yaitu: sistem bahasa (mekanisme komunikasi antara pengguna dan komponen lain dari SPK), sistem pengetahuan (tempat penyimpanan masalah sesuai dengan domain pengetahuan dalam SPK dalam bentuk data atau prosedur), serta sistem pemrosesan masalah (penghubung antara dua komponen lainnya dengan kemampuan untuk memanipulasi masalah yang dibutuhkan untuk proses pengambilan keputusan)

- *Keen*

  "Produk proses pengembangan dimana pengguna, pengembang serta SPK itu sendiri saling mempengaruhi satu sama lain yang pada akhirnya akan menghasilkan evolusi sistem serta pola penggunaan tertentu"

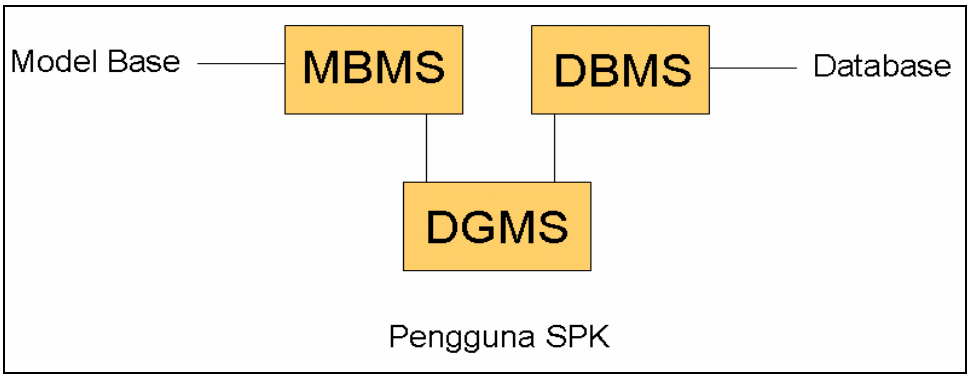
Berdasarkan sejumlah definisi tersebut, dapat ditarik kesimpulan bahwasanya SPK adalah sebuah sistem berbasiskan komputer yang dapat membantu pengambilan keputusan dengan mengolah serta menganalisis data sesuai orientasi pada kondisi yang akan terjadi di masa yang akan datang dalam berbagai situasi yang mungkin terjadi.

2.2.2 Komponen Sistem Penunjang Keputusan

Aplikasi SPK dapat terdiri dari sejumlah komponen sebagai berikut [1]:

- *Data-management system*. Komponen ini terdiri dari basis data yang menyimpan data yang relevan serta diatur oleh sebuah DBMS (*Database Management System*). DBMS berfungsi sebagai sebuah bank data untuk SPK. DBMS ini menyimpan data dalam jumlah besar yang dianggap relevan dengan domain pengetahuan SPK. DBMS memisahkan pengguna dari aspek fisik struktur basis data dan pemrosesannya.
- *Model base management system*. Peranan dari MBMS adalah untuk mentransformasikan data dari DBMS ke dalam bentuk informasi yang bermanfaat dalam pengambilan keputusan. Dengan banyaknya masalah yang tidak terstruktur, MBMS harus mampu menyediakan suatu bentuk pemodelan masalah kepada pengguna agar dapat direpresentasikan dalam bentuk yang lebih terstruktur
- *Dialog generation and management system*. SPK harus dilengkapi dengan komponen antarmuka (*interface*) yang intuitif dan mudah untuk digunakan. Antarmuka ini tidak hanya akan mempermudah pengguna dalam membangun model penunjang keputusan melainkan juga untuk memperoleh rekomendasi dari model tersebut terhadap masalah yang dihadapi pengguna.

Interaksi antara 3 komponen SPK ini dapat direpresentasikan dalam gambar berikut:



Gambar 2. 1 Arsitektur SPK

Dari sejumlah komponen utama yang membangun SPK, pemodelan atau *modelling* merupakan komponen yang sulit untuk ditentukan. Sejumlah model yang cukup sering diimplementasikan dalam SPK antara lain *decision analysis, optimization, search methods, heuristic programming*, serta *simulation*.

Dalam situasi keputusan yang melibatkan jumlah alternatif yang terbatas dan tidak terlalu besar biasanya digunakan pendekatan dengan model *decision analysis* [14]. Situasi seperti ini dapat dimodelkan dengan *decision tables* atau *decision trees*. *Decision tables* merupakan sebuah cara untuk mengorganisir informasi secara sistematis. Dengan menggunakan sebuah tabel yang berisi *decision variables (*disebut juga *alternatives)* dan *uncontrollable variables*. Contoh dari *decision tables* adalah sebagai berikut:

Tabel 2. 1 Contoh *Decision Table*

	State of Nature (Uncontrollable Variables)		
Alternative	*Solid Growth (%)*	*Stagnation (%)*	*Inflation (%)*
Bonds	12	6	3
Stocks	15	3	-2
CDs	6.5	6.5	6.5

Tabel 2.1 menggambarkan estimasi investasi dalam berbagai kondisi ekonomi yang terjadi. Jika masalah pengambilan keputusan berlangsung dalam kondisi yang penuh dengan kepastian maka investasi mana yang terbaik dapat dengan

mudah diketahui, namun lebih sering yang dihadapi adalah ketidakpastian (*uncertainty*) dan resiko (*risk*). Perbedaan antara 2 kondisi ini adalah pada *uncertainty*, probabilitas dari tiap *State of Nature* yang ada tidak diketahui, sedangkan pada *risk* asumsi probabilitas untuk setiap *State of Nature* yang mungkin muncul pada saat itu turut diperhitungkan.

Representasi alternatif dari *decision table* adalah *decision tree* atau pohon keputusan. Pohon keputusan mampu menunjukkan hubungan masalah secara grafis dan dalam situasi yang kompleks. Dalam *decision tree*, masalah pengambilan keputusan diasumsikan berlangsung dalam kondisi yang penuh kepastian. Maksudnya adalah sebuah rangkaian kondisi diproyeksikan hanya ke dalam satu alternatif (rangkaian kondisi dari *root* atau akar pohon hingga *leaf* atau simpul daun hanya memiliki satu alternatif atau solusi). Dalam penelitian ini, pohon keputusan digunakan mengingat proses prakiraan cuaca jangka pendek melibatkan sejumlah variabel dengan keluaran atau alternatif berjumlah terbatas. Dengan pohon keputusan, dapat terlihat variabel mana yang paling berpengaruh dalam menentukan kondisi hujan di hari mendatang berdasarkan perhitungan matematis yang diterapkan dalam algoritma klasifikasi *data mining C*4.5.

2.3 Metode Pohon Keputusan

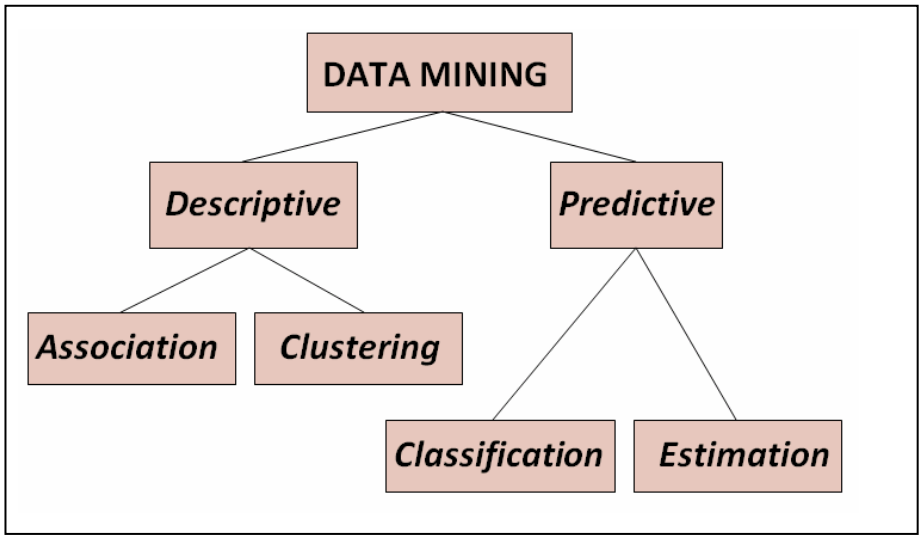
Pohon keputusan dapat dibangun dengan menggunakan teknik *data mining*. Teknik *data mining* ini berfungsi untuk mencari pola aturan dalam kumpulan data yang berjumlah besar. Dengan mengetahui aturan-aturan ini, sebuah kasus atau masalah baru dapat diklasifikasikan ke dalam suatu alternatif berdasarkan nilai dari variabelnya.

2.3.1 *Data Mining*

Pemanfaatan basis data memungkinkan keberadaan “tambang emas” pengetahuan. Dalam basis data banyak terkubur pengetahuan-pengetahuan yang tidak diketahui oleh manusia. Padahal dengan mengetahui, memahami, dan menggunakan pengetahuan ini dapat memberikan keuntungan yang cukup signifikan bagi suatu organisasi. Teknik yang dapat digunakan untuk mengeksplorasi ke dalam basis

data dan untuk mencari pola yang ada di dalamnya disebut dengan *knowledge discovery in databases* (KDD) atau yang lebih umum dikenal dengan nama *data mining* (DM). *Data mining* mampu menganalisis kumpulan data yang begitu besar menjadi informasi yang dapat digunakan untuk menunjang pengambilan suatu keputusan.

Tujuan dari *data mining* secara garis besar adalah untuk mendeskripsikan apa yang telah terjadi (*descriptive data mining*) , dan untuk memprediksikan apa yang akan terjadi (*predictive data mining*). *Descriptive data mining* mencari pola pada kejadian yang telah lampau yang mempengaruhi kejadian yang terjadi pada masa sekarang. Teknik *data mining* yang termasuk dalam kategori ini adalah *association* dan *clustering*. Sedangkan, *predictive data mining* mengacu pada kejadian yang telah lampau untuk memprediksikan apa yang terjadi pada masa yang akan datang. Yang termasuk ke dalam kategori *predictive data mining* ini adalah *classification* dan *estimation*. Teknik yang penulis gunakan dalam penelitian ini adalah *classification*. Teknik *data mining* ini membangun model klasifikasi berdasarkan *training data* yang digunakan. Model ini dapat digunakan untuk memprediksikan kelas atau kategori dari suatu data yang baru.



Gambar 2. 2 Pengkategorian *Data Mining*

Teknik-teknik *data mining* dapat diterapkan dengan sejumlah pendekatan seperti *symbolic* dan *inductive, connectionist*, dan *statistical*. Pendekatan yang digunakan dalam penelitian ini adalah *inductive.* Berdasarkan kamu *Webster*, induksi adalah:

"reasoning from particular facts or individual cases to general conclusion"
[penalaran berdasarkan fakta atau kasus-kasus untuk mencapai kesimpulan yang umum]

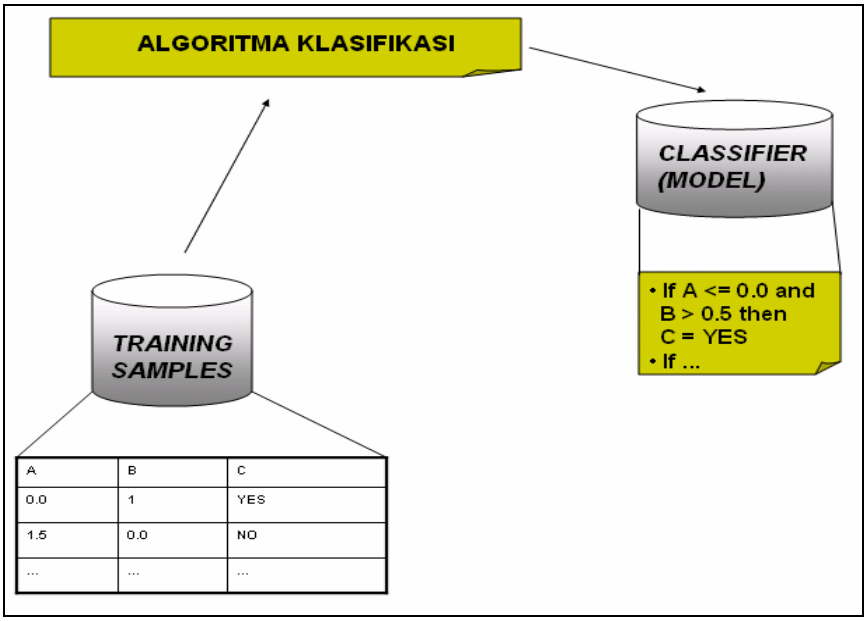
Induksi dianggap sebagai elemen dasar dalam penelitian ilmiah dan kesimpulan akhir yang diperoleh berupa hubungan antar atribut yang menyusun tiap kasus yang menjadi obyek penelitian tersebut [3]. Dalam merepresentasikan suatu hubungan antar atribut yang menyusun suatu kasus, dibutuhkan suatu representasi grafis yang membuat hubungan tersebut terlihat jelas dan mudah untuk dipahami. Representasi grafis ini dapat berupa sebuah pohon keputusan.

2.3.2 Definisi Pohon Keputusan

Pohon keputusan (*decision tree)* dan aturan keputusan (*decision rule)* merupakan metodologi *data mining* yang banyak diterapkan sebagai solusi untuk mengklasifikasikan masalah. Klasifikasi sendiri merupakan proses pembelajaran yang memetakan komponen data ke dalam sejumlah kelas yang telah didefinisikan sebelumnya (*predefined class)*. Proses klasifikasi yang menggunakan pendekatan induksi menggunakan sejumlah data sampel yang terdiri dari sejumlah vektor atribut beserta nilainya (disebut *feature vectors)* dan sebuah atribut kelas. Tujuan dari proses pembelajaran ini adalah untuk memperoleh model klasifikasi yang dikenal dengan sebutan *classifier* yang akan memprediksikan kelas untuk sebuah sampel berdasarkan nilai dari atribut-atributnya [6]. Proses klasifikasi terdiri dari 2 tahap [4]:

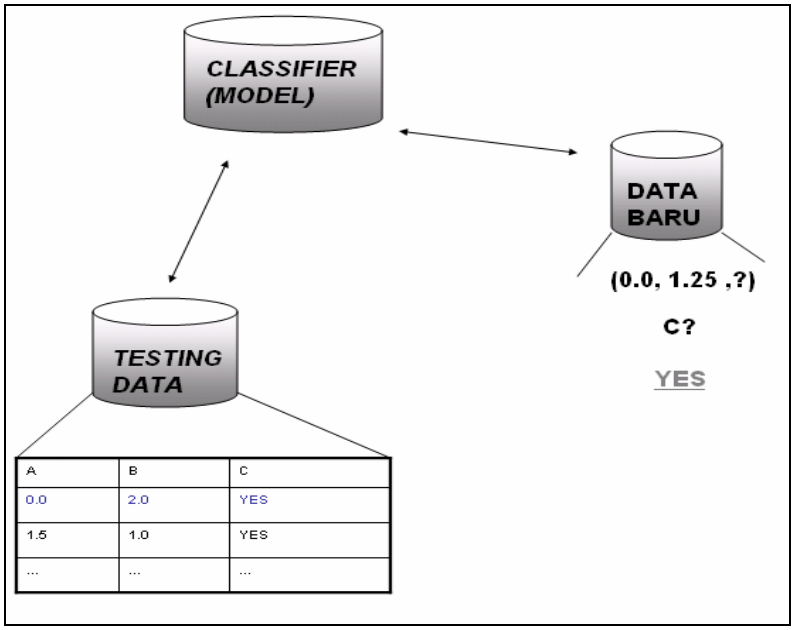
- Pembuatan model, pada tahap ini setiap data diasumsikan telah digolongkan ke dalam sejumlah kelas (*predefined class)*. Himpunan data yang akan menyusun model ini disebut sebagai *training data.* Model yang dihasilkan direpresentasikan dalam bentuk aturan klasifikasi, pohon keputusan, atau formula matematika

.



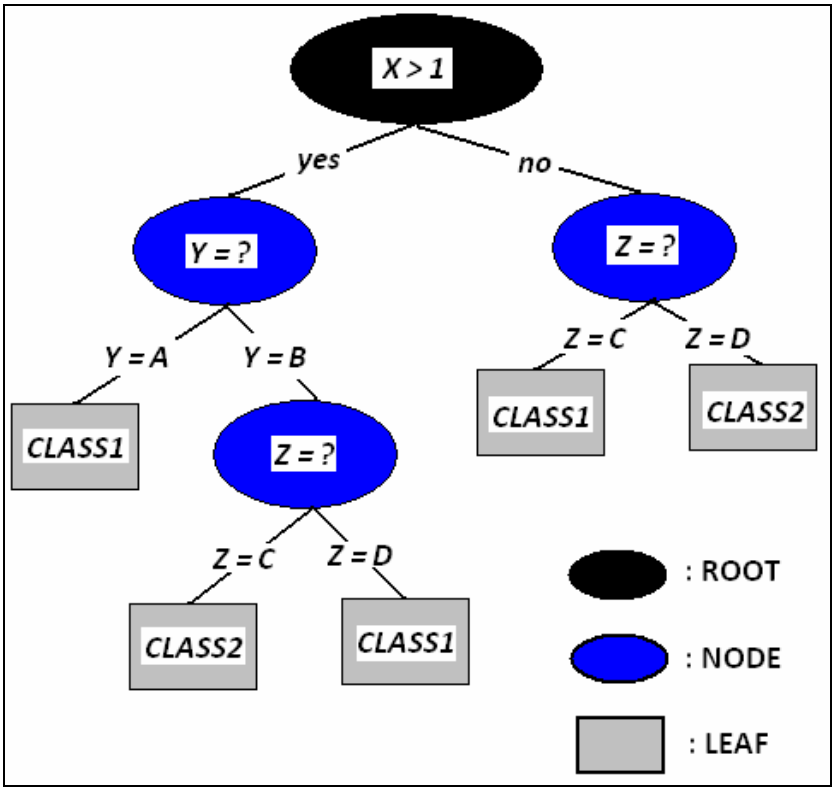
Gambar 2. 3 Ilustrasi Pembuatan Pohon Keputusan

- Pemanfaatan model, tahap ini digunakan untuk mengklasifikasikan obyek yang belum diketahui kelasnya. Estimasi akurasi dilakukan dengan membandingkan kelas dari *testing data* dengan kelas hasil klasifikasi model. Tingkat akurasi adalah ratio jumlah *testing data* yang diklasifikasikan secara benar berdasarkan model klasifikasi dengan seluruh jumlah *testing data*. Jika tingkat akurasi ini diterima maka model klasifikasi kemudian dapat digunakan untuk mengklasifikasikan data yang belum diketahui kelasnya.



Gambar 2. 4 Ilustrasi Pemanfaatan Pohon Keputusan

Representasi pohon keputusan ini dianggap sebagai metode logis yang sering digunakan pada bahasan mengenai statistik terapan dan pembelajaran mesin (*machine learning)*. Pembuatan pohon keputusan sendiri menggunakan metode *supervised learning* yaitu proses pembelajaran dimana data baru diklasifikasikan berdasarkan *training samples* yang ada [4]. Pohon keputusan ini terdiri dari *nodes* atau simpul yang merupakan atribut dari data sampel. Cabang (*branches*) yang keluar dari *node* tersebut merupakan nilai atau *outcome* yang dimiliki oleh atribut (*nodes)* bersangkutan. Sedangkan daun yang ada pada pohon keputusan tersebut menunjukkan kelas dari data sampel yang diuji. Sebagai ilustrasi dapat dilihat pada contoh gambar berikut ini:



Gambar 2. 5 Model Pohon Keputusan

Pada gambar 2.5 terlihat ada 3 atribut berbeda yaitu X, Y, dan Z yang terletak pada simpul (*node*) berbentuk oval. Atribut X terletak pada simpul akar (*root node)* sedangkan Y dan Z terdapat di dalam *internal node* atau simpul dalam. Tiap cabang yang keluar dari simpul tersebut menunjukkan nilai masing-masing atribut yang dimiliki oleh data pengujian. Pada simpul daun (*leaf node)* terdapat kelas yang menjadi keluaran akhir dari *classifier*. Untuk mengetahui kelas dari suatu data pengujian maka jalur yang ada dari akar hingga daun dapat ditelusuri [6].

2.3.3 Algoritma *C*4.5

Algoritma *C*4.5 merupakan generasi baru dari algoritma ID3 yang dikembangkan oleh J. Ross Quinlan pada tahun 1983. Untuk membuat sebuah pohon keputusan, algoritma ini dimulai dengan memasukkan *training samples* ke dalam simpul akar pada pohon keputusan. *Training samples* adalah sampel yang digunakan untuk membangun model *classifier* dalam hal ini pohon keputusan. Kemudian sebuah atribut dipilih untuk mempartisi sampel ini. Untuk tiap nilai yang dimiliki atribut

ini, sebuah cabang dibentuk. Setelah cabang terbentuk maka subset dari himpunan data yang atributnya memiliki nilai yang bersesuaian dengan cabang tersebut dimasukkan ke dalam simpul yang baru. Algoritma ID3 ini pada dasarnya hanya mengulang langkah pemartisian ini hingga pada akhirnya diperoleh keadaan dimana semua sampel pada sebuah simpul tergolong ke dalam kelas yang sama. Setiap jalur dari akar menuju daun pada pohon keputusan ini merepresentasikan aturan keputusan (*decision rule*) yang pada akhirnya nanti dapat digunakan sebagai prediktor kelas data berikutnya.

Pada algoritma ini, pemilihan atribut mana yang akan menempati suatu simpul dilakukan dengan melakukan perhitungan entropi informasi (*information entropy)* dan mencari nilai yang paling minimum. Pemilihan atribut pada algoritma ini berdasarkan pada asumsi bahwa kompleksitas yang dimiliki oleh pohon keputusan sangat berkaitan erat dengan jumlah informasi yang diberikan oleh nilai-nilai atributnya. Dengan kata lain, teknik heuristik berbasiskan informasi ini memilih atribut yang memberikan perolehan informasi terbesar (*highest information gain)* dalam menghasilkan subpohon (*subtree*) untuk mengklasifikasikan sampel [6].

Algoritma dengan pendekatan induktif seperti *C*4.5 memiliki sejumlah kriteria yaitu [6]:

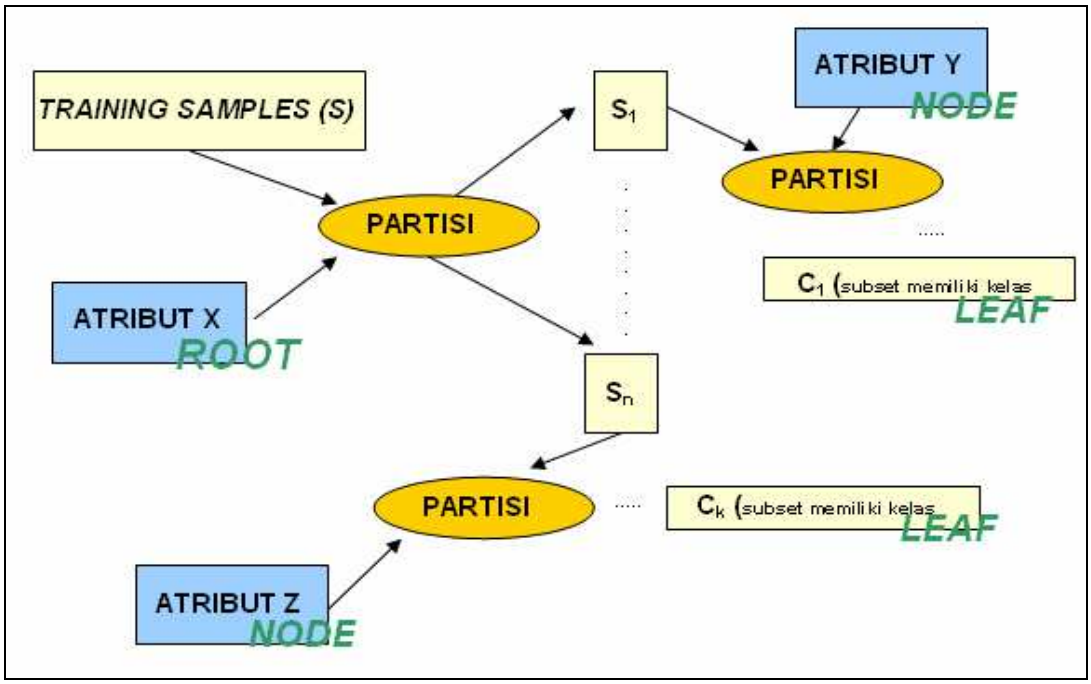
- Pasangan atribut-nilai (*attribute-value description).* Himpunan data yang digunakan untuk menganalisis harus dapat direpresentasikan dalam bentuk himpunan atribut. Tiap atribut ini dapat memiliki nilai diskret atau kontinu
- Kelas yang telah didefinisikan (*predefined-classes).* Kategori yang akan diberikan kepada tiap sampel harus ditentukan terlebih dahulu. Hal inilah yang menyebabkan pendekatan induktif ini disebut dengan *supervised-learning*.
- Kelas diskret. Sebuah kasus atau sampel harus tergolong atau tidak tergolong ke dalam sebuah kelas tertentu dan jumlah sampel harus jauh lebih besar daripada jumlah kelas yang ada.

- Jumlah data yang mencukupi. Jumlah data yang dibutuhkan dipengaruhi oleh jumlah atribut dan kelas serta kompleksitas dari model klasifikasi yang digunakan.
- Model klasifikasi logis. Pendekatan induktif digunakan untuk membangun *classifier* yang dapat diekspresikan sebagai pohon keputusan atau aturan keputusan.

Dengan asumsi *training samples T,* atribut (A_1, A_2, A_3, $A_{4,...}$) dan kelas terdiri dari (K_1, K_2, K_3, $K_{4,..}$) Kerangka utama dari algoritma *C*4.5 dapat dijelaskan sebagai berikut:

- Jika *T* tidak kosong dan semua sampel yang ada didalamnya memiliki kelas K_i yang sama maka pohon keputusan untuk *T* adalah sebuah simpul daun (*leaf node*) dengan label K_i
- Jika atribut kosong maka pohon keputusan berisi sebuah simpul daun dengan label K_j dimana K_j adalah kelas yang paling dominan pada *training samples T*
- Jika *T* terdiri dari sampel yang memiliki kelas yang berbeda-beda maka partisi *T* ke dalam T_1, T_2, T_3, ...T_n. *Training samples T* dipartisi berdasarkan *distinct value* dari atribut A_k yang pada saat itu menjadi *node parent* (simpul orang tua). Misalkan A_k terdiri dari 3 jenis nilai yaitu: n_1, n_2, n_3 maka *T* akan dipartisi ke dalam 3 subset yaitu yang nilai $A_k = n_1$, $A_k = n_2$, dan $A_k = n_3$.

Proses ini terus dilakukan secara rekursif dengan *base case* langkah 1 dan langkah 2. Cara untuk mencari atribut yang akan menjadi *node parent* pada suatu iterasi dilakukan dengan menghitung sebuah kriteria yang disebut *gain*. *Gain* berfungsi untuk memilih atribut yang akan diuji berdasarkan konsep teori informasi *entropy*. Berikut ini merupakan ilustrasi gambar dari proses jalannya algoritma *C*4.5:



Gambar 2. 6 Proses Algoritma *C*4.5

Berikut ini akan dijelaskan komponen-komponen yang menyusun algoritma *C*4.5 dalam membentuk pohon keputusan:

2.3.3.1 *Entropy*

Entropi merupakan distribusi probabilitas dalam teori informasi dan diadopsi ke dalam algoritma *C*4.5 untuk mengukur tingkat homogenitas distribusi kelas dari sebuah himpunan data (*data set*). Sebagai ilustrasi, semakin tinggi tingkat entropi dari sebuah *data set* maka semakin homogen distribusi kelas pada *data set* tersebut.

Jika distribusi probabilitas dari kelas didefinisikan dengan $P = (p_1, p_2, p_3, ..., p_k)$ maka entropi dapat dituliskan sebagai persamaan dari [12]:

$$E(p_1, p_2, ..., p_k) = -\sum_{i=1}^{k} (p_i \bullet \log_2(p_i)) \qquad (2.1)$$

Persamaan 2.1 sama dengan persamaan *Info(T)* sebagai berikut:

$$Info(T) = -\sum_{i=1}^{k}\left(\left(\frac{frequency(Ci,T)}{|T|}\right)\bullet \log_2\left(\frac{frequency(Ci,T)}{|T|}\right)\right) \qquad (2.2)$$

Dimana $frequency(Ci,T)$ adalah jumlah sampel di himpunan T yang memiliki kelas $C_1, C_2, C_3, ..., C_k$.

Sebagai contoh, distribusi kelas (0.5, 0.5) lebih homogen bila dibandingkan dengan distribusi (0.67, 0.33) sehingga distribusi (0.5, 0.5) memiliki entropi yang lebih tinggi dari distribusi (0.67, 0.33). Hal ini apat dibuktikan sebagai berikut:

$$E(0.5,0.5) = -0.5\times\log_2(0.5) - 0.5\times\log_2(0.5) = 1$$

$$E(0.67,0.33) = -0.67\times\log_2(0.67) - 0.33\times\log_2(0.33) = 0.91$$

Setelah T dipartisi ke dalam sejumlah subset $T_1, T_2, T_3, ...T_n$ berdasarkan atribut X maka perhitungan *Info* dilakukan dengan menggunakan himpunan *training data* yang merupakan hasil partisi sebagai berikut:

$$Info_x(T) = \sum_{i=1}^{n}\left(\left(\frac{|T_i|}{|T|}\right)\bullet Info(T_i)\right) \qquad (2.3)$$

2.3.3.2 *Information Gain*

Setelah membagi *data set* berdasarkan sebuah atribut kedalam subset yang lebih kecil, entropi dari data tersebut akan berubah. Perubahan entropi ini dapat digunakan untuk menentukan bagus tidaknya pembagian data yang telah dilakukan. Perubahan entropi ini disebut dengan *information gain* dalam algoritma *C*4.5. *Information gain* ini diukur dengan menghitung selisih antara entropi *data set* sebelum dan sesudah pembagian (*splitting*) dilakukan. Pembagian yang terbaik akan menghasilkan entropi subset yang paling kecil, dengan demikian berdampak pada *information gain* yang terbesar [18].

Jika sebuah *data set D* dipartisi berdasarkan nilai dari sebuah atribut *X* sehingga menghasilkan subset *(T_1, T_2, ..., T_n)* maka *information gain* dapat dihitung dengan persamaan:

$$Gain(x) = Info(T) - Info_x(T) \tag{2.4}$$

Dalam persamaan 2.4, $Info(T)$ adalah entropi dari *data set* sebelum dipartisi berdasarkan atribut *X*, dan $Info_x(T)$ adalah *Info* dari *subset* setelah dilakukan pemartisian berdasarkan atribut *X*.

2.3.3.3 *Gain Ratio*

Perhitungan *information gain* masih memiliki sejumlah kekurangan. Salah satu kekurangan yang mungkin terjadi adalah pemilihan atribut yang tidak relevan sebagai pemartisi yang terbaik pada suatu simpul. *Gain ratio* merupakan normalisasi dari *information gain* yang memperhitungkan entropi dari distribusi probabilitas subset setelah dilakukan proses partisi. Secara matematis, *gain ratio* dihitung sebagai berikut [12]:

$$GainRatio(X) = \frac{Gain(X)}{SplitInfo(X)} \tag{2.5}$$

Dimana $SplitInfo(X)$ merupakan entropi dari seluruh distribusi probabilitas *subset* setelah dilakukan pemartisian (*splitting*)

$$SplitInfo(x) = -\sum_{i=1}^{n}\left(\left(\frac{|T_i|}{|T|}\right) \bullet \log_2\left(\frac{|T_i|}{|T|}\right)\right) \tag{2.6}$$

Dari persamaan 2.6, $|T_i|$ adalah kardinalitas dari subset T_i yang berada dalam *training data T*.

2.3.3.4 Penanganan *Continuous Attribute*

Penanganan atribut dengan nilai kontinu merupakan salah satu kelebihan yang dimiliki algoritma *C*4.5 bila dibandingkan dengan pendahulunya, ID3. *Distinct value* dari *training data T* harus diurutkan terlebih dahulu sehingga diperoleh *value* dengan urutan $\{v_1, v_2, ..., v_m\}$. Cari setiap *threshold* yang berada di antara v_i dan v_{i+1} dengan menggunakan persamaan $\frac{v_i + v_{i+1}}{2}$. Dengan begitu hanya akan ada *m-1* kemungkinan *threshold*. Untuk masing-masing *threshold* ini dilakukan perhitungan *gain ratio*. *Threshold* yang terpilih adalah *threshold* dengan *gain ratio* terbesar. Langkah ini dilakukan untuk setiap atribut dengan tipe data numerik atau memiliki nilai yang kontinu [12].

2.3.3.5 *Penanganan Missing Value*

Salah satu fitur atau kelebihan lain yang dimiliki oleh *C*4.5 dibandingkan dengan pendahulunya (ID3) selain kemampuannnya dalam menangani atribut dengan nilai kontinu adalah kemampuannya dalam menangani atribut dengan nilai *null* atau *missing value*. Perhitungan $Gain(X)$ dan $SplitInfo(X)$ dilakukan seperti biasa kecuali *training data* yang digunakan adalah *training data* tanpa *missing value*. Dengan kata lain perhitungan $Gain(X)$ dan $SplitInfo(X)$ tidak memperhitungkan data dengan *missing value*. Ketika menghitung $GainRatio(X)$, probabilitas nilai yang diketahui (dinotasikan dengan *F*) harus diperhitungkan. Sehingga rumus $GainRatio(X)$ dapat ditulis sebagai berikut [12]:

$$GainRatio(X) = F \bullet \frac{Gain(X)}{SplitInfo(X)} \tag{2.7}$$

Dengan *F* adalah:

$$\frac{|NilaiXYangDiketahui|}{|NilaiX|} \tag{2.8}$$

2.3.3.6 *Error Based Pruning*

Pohon keputusan biasanya disederhanakan dengan menghapuskan satu atau beberapa subpohon kemudian menggantikannya dengan simpul daun (*leaf node*). Algoritma *C*4.5 menerapkan proses penggantian sebuah subpohon dengan salah satu cabang yang dimiliki oleh subpohon tersebut. Secara garis besar, proses pemangkasan subpohon yang dilakukan dalam algoritma ini dimulai dari bagian bawah pohon keputusan. Jika sebuah subpohon digantikan oleh sebuah simpul daun atau oleh salah satu cabang yang keluar dari subpohon tersebut akan menghasilkan tingkat prediksi *error* (*predicted error rate*) yang lebih rendah maka proses pemangkasan (*pruning*) dapat dilakukan. Karena tingkat *error* dari keseluruhan pohon keputusan akan menurun seiring menurunnya tingkat *error* dari subpohon yang menyusunnya maka proses pemangkasan ini akan menghasilkan sebuah pohon keputusan yang memiliki *predicted error rate* paling kecil.

Bagaimana tingkat *error* dari sebuah pohon keputusan dapat diprediksikan? Quinlan dalam bukunya menyatakan pada dasarnya ada 2 jenis teknik dalam memperoleh *predicted error rate*. Teknik yang pertama dilakukan dengan menggunakan himpunan data baru yang berbeda dari *training data*. Adapun yang tergolong ke dalam jenis ini adalah:

- *Cost-complexity pruning*, yang memodelkan *predicted error rate* sebagai total kompleksitas dari pohon keputusan yang berbobot (himpunan data baru digunakan untuk menentukan porsi dari bobot ini) serta jumlah *error* yang dimilikinya pada *training data*.
- *Reduced error pruning*, yang menilai tingkat *error* dari sebuh pohon keputusan dan komponen-komponennya langsung berdasarkan himpunan kasus yang baru.

Kelemahan yang dimiliki oleh jenis *pruning* ini adalah sejumlah data tambahan harus disiapkan untuk menjadi himpunan data baru, hal ini menyebabkan jumlah *training data* menjadi lebih sedikit. Teknik yang *C*4.5 gunakan adalah teknik

pruning yang hanya menggunakan *training data* yang sama ketika membangun pohon keputusan.

Ketika *N* buah *training data* tergolong ke dalam sebuah simpul daun, dan ada *E* data dari *training data* tersebut yang tergolong ke dalam kelas yang salah maka tingkat resubstitusi *error* yang dimiliki oleh simpul daun ini adalah *E/N*. Secara naif hal ini dapat dilihat sebagai jumlah *E* kejadian (*events*) dalam *N* kali percobaan (*trials*). Jika *N training data* ini dianggap sebagai sampel percobaan maka tingkat resubstitusi *error E/N* dapat dianggap sebagai probabilitas terjadinya *error* (*E*) pada populasi kasus (*N*) yang tergolong ke dalam simpul daun bersangkutan. Tingkat *error* tidak dapat ditentukan secara mutlak namun dapat diperoleh dari distribusi probabilitas yang berupa batas kepercayaan (*confidence limits*). Untuk suatu tingkat kepercayaan (*confidence level*) *CF*, batas atas probabilitas ini dapat ditentukan dari *confidence limits* distribusi Binomial (*Binomial distribution*) yaitu $U_{CF}(E,N)$. Algoritma *C*4.5 mencari *predicted error rate* untuk sebuah simpul daun dengan menggunakan batas atas ini pada pohon keputusan yang telah dibuat untuk meminimalisir tingkat *error*.

Penjelasan mengenai *predicted error rate* yang digunakan dalam algoritma *C*4.5 memang tidak sesuai dengan pemahaman akan batas kepercayaan (*confidence limits*) dan teknik *sampling* yang ada pada ilmu statistika, namun sebagaimana pendekatan heuristik lainnya, estimasi dengan pendekatan *confidence level* ini memberikan hasil yang dapat diterima.

Estimasi *error* dari simpul daun maupun subpohon dihitung berdasarkan asumsi bahwa mereka digunakan untuk mengklasifikasikan sejumlah kasus-kasus baru (*unseen cases*) yang memiliki ukuran yang sama dengan *training data*. Sehingga sebuah simpul daun yang menggolongkan *N* buah kasus dengan *predicted error rate* $U_{CF}(E,N)$ diprediksi akan memiliki $N \times U_{CF}(E,N)$ buah *error*. Begitu juga halnya dengan subpohon, jumlah *predicted error* yang dimiliki sebuah subpohon merupakan total *predicted error* dari cabang-cabangnya [12]. Berikut ini merupakan rumus perhitungan *upper confidence limit* yang digunakan [16]:

$$e = \frac{\left(f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right)}{\left(1 + \frac{z^2}{N} \right)} \quad (2.9)$$

Default confidence level yang digunakan dalam *C*4.5 adalah 25%, sehingga nilai *z* untuk persamaan 2.9 adalah 0.69 dengan melihat pada tabel distribusi normal untuk peluang 25%. Nilai *f* menunjukkan probabilitas *error* pada *training data* dan *N* menunjukkan jumlah kasus yang tergolong ke dalam simpul daun yang sedang dihitung *predicted error*-nya. Sebagai contoh misalkan terdapat 20 buah data dari himpunan *training data* yang diklasifikasikan ke dalam sebuah simpul daun, dari 20 data ini terdapat 8 buah data yang tergolong ke dalam kelas yang salah pada simpul daun tersebut. Dari keterangan ini dapat dihitung nilai *upper confidence limit* dari tingkat *error* yang mungkin terjadi pada simpul daun tersebut sebagai berikut:

$$N = 20$$
$$z = 0.69$$
$$f = \frac{8}{20} = 0.4$$
$$e = \frac{\left(0.4 + \frac{0.69^2}{40} + 0.69\sqrt{\frac{0.4}{20} - \frac{0.4^2}{20} + \frac{0.69^2}{1600}} \right)}{\left(1 + \frac{0.69^2}{20} \right)} = \frac{0.4 + 0.012 + 0.0759}{1.023} = 0.4769$$

Dengan tingkat kepercayaan 75% dapat disimpulkan bahwa tingkat *error* maksimum yang mungkin terjadi pada simpul ini adalah sebesar 47.69%.

2.3.3.7 Contoh Proses Algoritma C4.5

Bagian ini akan menjelaskan ilustrasi dari alur proses yang ada pada algoritma *C*4.5.

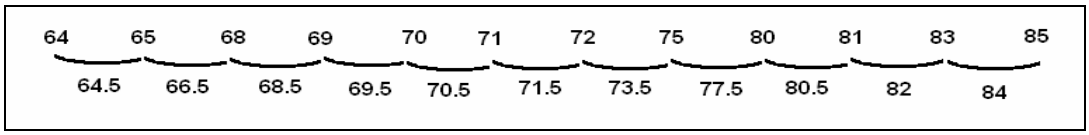
Contoh *training set* [12]:

Tabel 2. 2 Contoh *Training Data*

Outlook	Temperature	Humidity	Windy	Class
sunny	75	70	TRUE	Play
sunny	80	90	TRUE	Don't Play
sunny	85	85	FALSE	Don't Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
overcast	72	90	TRUE	Play
overcast	83	78	FALSE	Play
overcast	64	65	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play
rain	65	70	TRUE	Don't Play
rain	75	80	FALSE	Play
rain	68	80	FALSE	Play
rain	70	96	FALSE	Play

Berdasarkan tabel 2.2, *training data* disusun oleh sejumlah atribut yaitu *Outlook, Temperature, Humidity,* dan *Windy* serta memiliki sebuah *predefined class* yaitu *Class.* Untuk atribut yang memiliki tipe diskret seperti *Outlook*, nilai *GainRatio* harus dihitung untuk seluruh nilai yang dimiliki oleh atribut ini (*sunny, overcast,* dan *rain*). Sedangkan untuk atribut yang memiliki tipe kontinu seperti *Temperature*, nilai *GainRatio* harus dihitung untuk seluruh *threshold* yang merupakan *mean* atau rata-rata dari 2 nilai berbeda (*distinct value*) yang tersusun secara urut dari yang terkecil hingga terbesar:

Nilai yang berbeda untuk atribut *Temperature*:



Gambar 2. 7 Pembagian *Threshold* Atribut Kontinu

Untuk mengetahui *GainRatio* setiap atribut, nilai *entropy* awal sebelum himpunan *training data* pada tabel 2.2 dipartisi perlu dihitung sebagai berikut:

Jumlah kelas *Play*: 9
Jumlah kelas *Don't play*: 5

$$Info(T) = -\frac{9}{14} \bullet \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \bullet \log_2\left(\frac{5}{14}\right) = 0.940$$

Setelah menghitung nilai *entropy training data* sebelum dipartisi dengan atribut apapun, berikutnya nilai *entropy* untuk *training data* setelah dipartisi oleh masing-masing atribut harus dihitung. Berikut ini akan dijelaskan proses penghitungan *entropy* himpunan setelah dipartisi oleh atribut yang bertipe diskret (*Outlook*) dan kontinu (*Temperature*) hingga diperoleh *GainRatio* untuk masing-masing atribut tersebut.

- **Contoh penghitungan *GainRatio* himpunan data yang dipartisi dengan atribut bertipe diskret**

  Berdasarkan tabel 2.2, jumlah distribusi kelas *training data* berdasarkan atribut *Outlook* adalah sebagai berikut:

Tabel 2. 3 Contoh Distribusi Kelas Berdasarkan Atribut Diskret

Nilai *Outlook*	∑ *Play*	∑ *Don't Play*	Total
Sunny	2	3	5
Overcast	4	0	4

Tabel 2. 3 Contoh Distribusi Kelas Berdasarkan Atribut Diskret (Lanjutan)

Rain	3	2	5

Berdasarkan tabel 2.3, maka nilai *entropy* untuk atribut *Outlook* adalah:

$$Info_x(T) = \frac{5}{14} \bullet \left(-\frac{2}{5} \bullet \log_2\left(\frac{2}{5}\right) - \frac{3}{5} \bullet \log_2\left(\frac{3}{5}\right) \right)$$
$$+ \frac{4}{14} \bullet \left(-\frac{4}{4} \bullet \log_2\left(\frac{4}{4}\right) - \frac{0}{4} \bullet \log_2\left(\frac{0}{4}\right) \right)$$
$$+ \frac{5}{14} \bullet \left(-\frac{3}{5} \bullet \log_2\left(\frac{3}{5}\right) - \frac{2}{5} \bullet \log_2\left(\frac{2}{5}\right) \right)$$
$$= 0.694$$

Setelah menghitung nilai *entropy* sebelum dan sesudah proses pemartisian, maka nilai *Gain*, *SplitInfo*, dan *GainRatio* dapat dihitung sebagai berikut:

$$Gain(X) = 0.940 - 0.694 = 0.246$$

$$SplitInfo(X) = -\frac{5}{14} \bullet \log_2\left(\frac{5}{14}\right) - \frac{4}{14} \bullet \log_2\left(\frac{4}{14}\right) - \frac{5}{14} \bullet \log_2\left(\frac{5}{14}\right) = 1.577$$

$$GainRatio(X) = \frac{0.246}{1.577} = 0.156$$

- **Contoh penghitungan *GainRatio* himpunan data yang dipartisi dengan atribut bertipe kontinu**

Untuk atribut yang memiliki tipe kontinu, *GainRatio* dihitung untuk semua *threshold* yang dimiliki atribut tersebut. Dengan melihat contoh pada atribut *Temperature*, *GainRatio* untuk atribut ini adalah nilai terbesar dari *GainRatio* yang dihasilkan pada nilai 64.5, 66.5, 68.5, 69.5, 70.5, 71.5, 73.5, 77.5, 80.5, 82, dan 84. Bagian berikut ini akan menjelaskan proses penghitungan *GainRatio* untuk salah satu *threshold* dari atribut *Temperature*:

Tabel 2. 4 Contoh Distribusi Kelas Berdasarkan Atribut Kontinu

Threshold = 69.5	$\sum$ *Play*	$\sum$ *Don't Play*	Total
<= *threshold*	3	1	4
> *threshold*	6	4	10

Berdasarkan tabel 2.4, maka nilai *entropy* untuk atribut *Temperature* berdasarkan *threshold* 69.5 adalah:

$$Info_x(T) = \frac{4}{14} \bullet \left(-\frac{3}{4} \bullet \log_2\left(\frac{3}{4}\right) - \frac{1}{4} \bullet \log_2\left(\frac{1}{4}\right)\right)$$
$$+ \frac{10}{14} \bullet \left(-\frac{6}{10} \bullet \log_2\left(\frac{6}{10}\right) - \frac{4}{10} \bullet \log_2\left(\frac{4}{10}\right)\right)$$
$$= 0.925$$

Setelah menghitung nilai *entropy* sebelum dan sesudah proses pemartisian, maka nilai *Gain*, *SplitInfo*, dan *GainRatio* dapat dihitung sebagai berikut:

$$Gain(X) = 0.940 - 0.925 = 0.015$$

$$SplitInfo(X) = -\frac{4}{14} \bullet \log_2\left(\frac{4}{14}\right) - \frac{10}{14} \bullet \log_2\left(\frac{10}{14}\right) = 0.863$$

$$GainRatio(X) = \frac{0.015}{0.863} = 0.017$$

Penghitungan di atas hanya menghitung *GainRatio* untuk satu *threshold* atribut *Temperature* saja dan untuk mengetahui *GainRatio* untuk atribut ini perlu dilakukan penghitungan untuk seluruh *threshold* yang dimilikinya kemudian mencari nilai terbesar yang ada.

Atribut dengan *GainRatio* terbesar akan menempati simpul akar (*root node*) jika penghitungan terjadi pada iterasi pertama dan akan menempati simpul dalam

(*internal node*) dari pohon keputusan jika penghitungan terjadi pada iterasi berikutnya.

2.3.4 Aplikasi Berbasis Pohon Keputusan

Bagian ini akan menjelaskan sejumlah penerapan algoritma *C*4.5 dan metode pohon keputusan dalam sejumlah penelitian maupun aplikasi yang pernah dikembangkan sebelumnya.

2.3.4.1 Sistem Prakiraan Hujan Jangka Pendek di Timur Laut Thailand

Bagian timur laut Thailand terdiri dari daerah yang gersang dan curah hujan yang bervariasi. Untuk meningkatkan daya penyerapan tanah di area ini, operasi penanaman awan dilakukan dalam *Royal Rain Making Project*. Karena tidak ada jaminan keberhasilan dalam menjalankan operasi ini, sangatlah penting untuk menentukan atau memprediksikan tingkat kesuksesan sebelum operasi ini dijalankan. Sejumlah faktor iklim, catatan penyerapan, serta hasil prediksi dari model awan seperti *Great Plains Cumulus Model* (GPCM) biasanya digunakan dalam menentukan keputusan apakah operasi penanaman awan ini akan dijalankan atau tidak. Hal paling prinsip yang harus diperhatikan dalam menentukan tingkat efektifitas dari program penanaman awan ini adalah curah hujan.

Secara tradisional, estimasi curah hujan dapat diperoleh atau diprediksi dari pemodelan angka dengan menggunakan radar dan pengamatan permukaan. Sebagai alternatif, digunakan metodologi pembelajaran mesin (*machine learning*) untuk memprediksi tingkat curah hujan dalam jangka pendek. Sumber data yang digunakan dalam penelitian ini berasal dari *Bureau of the Royal Rain Making and Agricultural Aviation and Department of Meteorology*, Thailand berupa data GPCM dan RADAR. Tiap data GPCM terdiri dari 52 variabel atau fitur seperti temperatur, kelembaban udara, tekanan udara, angin, stabilitas atmosfer, potensi penanaman awan, serta curah hujan. Himpunan data GPCM dan RADAR terdiri dari 179 rekord dengan cara menghubungkan data GPCM dengan hasil

pengamatan RADAR dari *Chalermprakiat Royal Rainmaking Research center* di *Pimai,* Provinsi *Nakhon Ratchasima* selama Maret 2004 hingga September 2006. Fitur tambahan yang dihasilkan dari penggabungan himpunan data ini adalah jumlah awan, ketinggian awan, intensitas awan dan jangkauan hujan sehingga total fitur atau variabel yang digunakan berjumlah 57.

Algoritma *C*4.5 digunakan sebagai salah satu metode selain *Artificial Neural Network* (ANN) dan *Support Vector Machine* (SVM). Percobaan pertama dengan algoritma *C*4.5 terdiri dari 2 kategori prediksi yaitu Hujan dan Tidak Hujan dan memberikan tingkat akurasi sebesar 94.41% dengan menggunakan 5-*fold cross validation*. Sedangkan percobaan ke-2 terdiri dari 3 kategori prediksi yaitu Tidak Hujan (0 – 0.1 *mm*), Hujan Sedikit (0.1 – 10 *mm*), dan Hujan Sedang (> 10 *mm*). Hasil dari percobaan ke-2 memberikan tingkat akurasi 62.57%. Percobaan ini dilakukan dengan menggunakan WEKA version 3.5.4 [5].

2.3.4.2 Pengujian Sistem Penunjang Keputusan Berbasis Petunjuk Klinis

ASTI merupakan sebuah sistem penunjang keputusan berbasis petunjuk klinis yang dikembangkan di Perancis. SPK ini bertujuan untuk memperbaiki perawatan terapis bagi pasien dengan penyakit kronis dengan cara membantu dokter dalam mempertimbangkan berbagai rekomendasi yang tercantum dalam petunjuk klinis. ASTI memiliki modul kritisisasi yang merupakan sistem berbasis aturan (*rule based system*) dan terdiri dari *knowledge base* dan *inference engine*. Modul ini dapat memberikan peringatan ketika resep yang ditulis oleh dokter berbeda dengan resep yang direkomendasikan oleh petunjuk klinis.

Metode yang digunakan dalam pengembangan sistem ini terdiri dari 3 tahap, yaitu [8]:

- *Generating input vectors and outputs*

  Data yang digunakan sebagai masukan dalam sistem ini berasal dari kondisi klinis pasien, sejarah perawatan pasien, perawatan baru yang dianjurkan dokter, serta efisiensi dan toleransi dari perawatan yang sedang dilakukan.

- *Building the decision tree*

Data yang telah diproses dari tahap sebelumnya digunakan sebagai variabel klasifikasi. Pohon keputusan dibangun dengan menggunakan algoritma *C*4.5. Adapun sejumlah variabel yang digunakan antara lain penemuan sejarah penyakit Diabetes, indeks massa tubuh (*body mass index*), tipe perawatan yang sedang dijalankan, tingkat Hemoglobin, dan lain sebagainya.

- *Comparing the Decision Tree with Clinical Guidelines*

Proses perbandingan dilakukan dengan meminta 2 orang dokter untuk membandingkan pohon keputusan yang telah dibangun dengan petunjuk klinis yang ada. Para ahli menemukan bahwa pohon keputusan lebih mudah dibaca dan seluruh rekomendasi terapi yang ada pada petunjuk klinis terdapat dalam pohon keputusan tersebut. Mereka juga menyatakan bahwa berbagai cabang yang ada dalam pohon keputusan tersebut mengandung resep yang sama sebagaimana dianjurkan dalam petunjuk klinis.

2.3.4.3 Sistem Prediksi Diagnosis Demam *Dengue*

Demam *dengue* sangat sering muncul di daerah tropis sehingga sering menjadi wabah atau epidemik di daerah tersebut. Pemeriksaan awal pada gejala demam ini sering menghasilkan diagnosis yang rancu dengan kemunculan gejala penyakit lain. Strategi pemeriksaan berdasarkan sejumlah petunjuk atau gejala yang muncul pada diri pasien pada awal kemunculan demam dapat menggolongkan mereka ke dalam sejumlah kategori sehingga proses pemeriksaan dan penanganan selanjutnya dapat dilakukan dengan lebih mudah.

Penelitian yang dibiayai oleh *Biomedical Research Council* (BMRC) dari *Agency for Science*, Singapura ini menggunakan sampel data pasien sebanyak 1200 orang yang mengalami kemunculan demam dalam 72 jam pertama. Dari seluruh sampel tersebut, 364 diantaranya mengalami positif RT-PCR, 173 hanya demam *dengue*, 171 demam berdarah, dan 20 orang mengalami *shock syndrome*. Penelitian ini menggunakan pohon keputusan *C*4.5 dengan menganalisis data klinis, hematologi, dan virologi. Algoritma ini berhasil mengklasifikasikan penyakit ke dalam golongan *dengue* dan *non-dengue* dengan tingkat akurasi 84.7%.

Adapun kesimpulan yang dapat ditarik dari penelitian ini adalah algoritma keputusan (*decision algorithms*) dapat digunakan dengan menggunakan parameter klinis dan hematologi yang sederhana untuk memprediksikan diagnosis dan prognosis dari penyakit *dengue* dan menjadi sebuah penemuan yang bermanfaat dalam manajemen dan pengawasan penyakit [13].