

BAB II

LANDASAN TEORI

2.1. Kajian Empiris

Menurut Kamus Besar Bahasa Indonesia (KBBI) empiris adalah berdasarkan pengalaman (terutama yang diperoleh dari penemuan, percobaan, pengamatan yang telah dilakukan). Kajian empiris adalah kajian yang diperoleh dari observasi atau percobaan. Berikut ini kajian empiris tentang penggunaan Algoritma C4.5 dari beberapa penelitian diantaranya :

Dalam penelitian yang berjudul Mining Educational Data to Analyze Students Performance (Baradwaj & Pal, 2012) dengan menggunakan data dari 50 siswa yang diperoleh dari Universitas VBS Purvanchal, Jaunpur (Uttar Pradesh) jurusan Application Computer MCA (Master of Computer Applications) dari sesi 2007 sampai 2010. dengan variable PSM (Previous Semester Marks), CTG (Class Test Grade), SEM (Seminar Performance), ASS (Assignment), GP (General Proficiency), ATT (Attendance), LW (Lab Work), ESM (End Semester Marks). Pemilihan split criteria menggunakan gain ratio, dan tujuan akhirnya untuk mengetahui kinerja pada akhir semester dengan pembagian first, second, third, dan fail. Penelitian ini juga berguna untuk mengidentifikasi siswa yang memerlukan perhatian khusus untuk mengurangi ransum atau rasio yang gagal dan mengambil tindakan yang tepat untuk semester berikutnya.

Dalam penelitian yang berjudul implementasi *data mining* dengan algoritma C4.5 untuk memprediksi tingkat kelulusan mahasiswa (Kamagi &

Hansun, 2015) menggunakan 100 data alumni angkatan 2007-2009 pada Universitas Multimedia Nusantara, berdasarkan hasil implementasi dan uji coba menggunakan algoritma C4.5 berhasil memprediksi kelulusan mahasiswa dengan persentase 87,5% dari 60 data training dan 40 data testing.

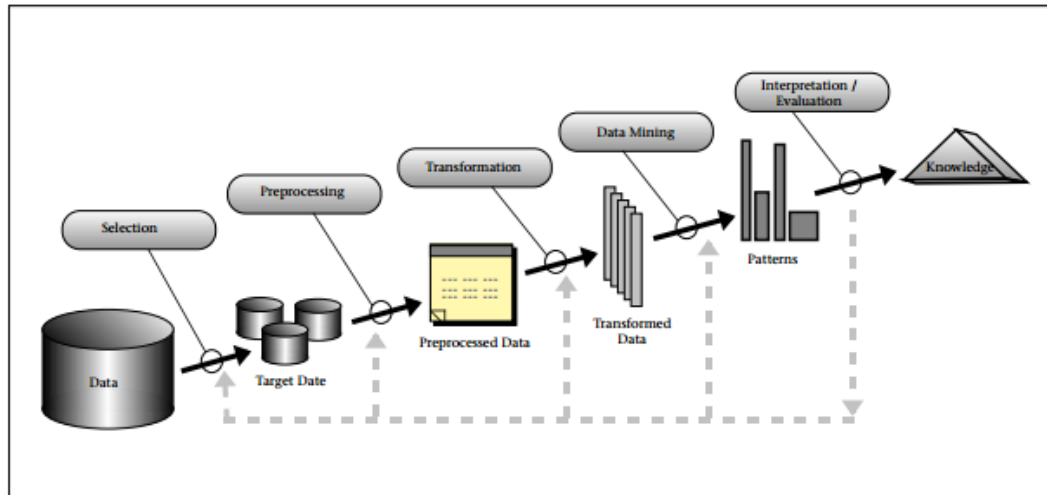
2.2. Data Mining

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam database. *Data mining* adalah proses yang menggunakan teknik statistic, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar (Turban.et al, 2005).

Menurut Gartner Group *data mining* adalah suatu proses menemukan hubungan yang berarti, pola, dan kecenderungan dengan memeriksa dalam sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik statistik dan matematika (Larose, 2005). *Data mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual (Pramudiono, 2003).

Data mining merupakan langkah penting atau bagian dari *knowledge discovery in databases* (KDD) yang menghasilkan pola atau model yang berguna dari data. *Data mining* dan *knowledge discovery in databases* (KDD) memiliki konsep yang berbeda namun saling berkaitan. *Knowledge discovery in databases* (KDD) mengacu pada proses keseluruhan menemukan pengetahuan yang bermanfaat dari data sedangkan *data mining* merupakan bagian dari KDD

menemukan pola baru dari kekayaan data dalam database dengan fokus pada algoritma. (Fayyad, Piatetsky-Shapiro, & Smyth, 1996)



Gambar 2. 1 Data mining dan Knowledge discovery in databases (KDD)

Proses *Knowledge discovery in databases* (KDD) pada gambar 2.1 secara garis besar dapat dijelaskan sebagai berikut :

1. *Selection* adalah proses penyeleksian data yang relevan untuk dianalisis atau digunakan dari kumpulan database.
2. *Preprocessing* adalah pembersihan data sebelum diproses pencarian informasi baru yang mencakup antara lain memeriksa dan menghapus data yang tidak konsisten, dan membuang duplikasi data.
3. *Transformation* adalah proses transformasi data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*.
4. *Data mining* adalah proses penggalian informasi atau pola dalam kumpulan data dengan teknik, metode atau algoritma dalam *data mining*.

5. *Interpretation/Evaluation* adalah proses menafsirkan pola atau informasi yang dihasilkan oleh *data mining* tersebut sehingga mudah dimengerti oleh manusia. Pada tahap ini juga mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

2.2.1. Fungsi Data Mining

Menurut (Chen, et al., 2015), *data mining* memiliki beberapa fungsi berdasarkan tugas yang dapat dilakukan, yaitu :

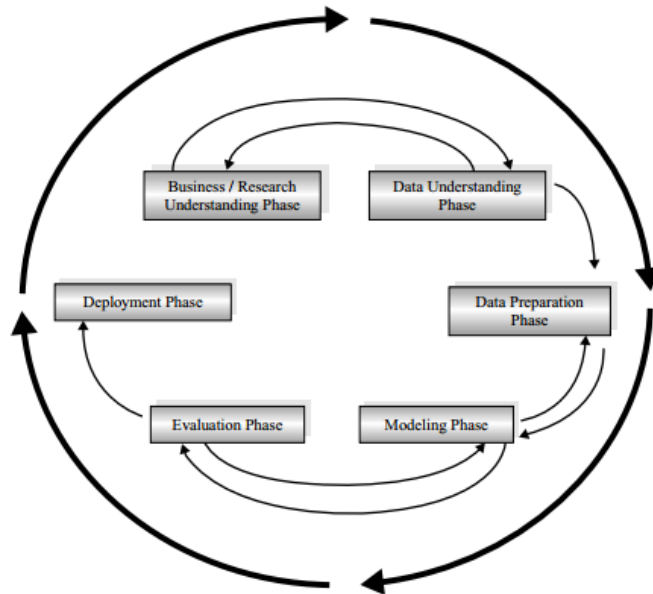
1. *Classification* adalah proses pencarian suatu set model atau fungsi yang menjelaskan dan membedakan kelas data atau konsep, untuk tujuan memprediksi kelas dari objek yang label kelasnya belum diketahui dalam data.
2. *Clustering* adalah pengelompokan data, hasil observasi dan kasus ke dalam class yang mirip. Suatu klaster (*cluster*) adalah koleksi data yang mirip antara satu dengan yang lain, dan memiliki perbedaan bila dibandingkan dengan data dari klaster lain.
3. *Association analysis* adalah penemuan aturan asosiasi menampilkan kondisi atribut nilai yang sering terjadi bersama-sama dalam himpunan data.
4. *Time series analysis* adalah metode dan teknik untuk menganalisis perubahan data secara berjangka untuk mengambil informasi dan karakteristik lain dari data.

5. *Other analysis*, pencarian model atau pola keteraturan atau tren untuk objek yang perilakunya berubah dari waktu ke waktu.

2.3. Cross Industry Standard Process (CRISP-DM)

Cross-Industry Standard Process for Data Mining (CRISP-DM) yang dikembangkan tahun 1996 oleh analis dari beberapa industri seperti DaimlerChrysler, SPSS dan NCR. CRISP-DM menyediakan standar proses *data mining* sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian (Larose, 2005).

Dalam CRISP-DM, sebuah proyek *data mining* memiliki siklus hidup yang terbagi dalam enam fase seperti pada Gambar 2.2. Keseluruhan fase yang berurutan tersebut bersifat adaptif atau bias berubah menyesuaikan permasalahan. Fase berikutnya dalam urutan bergantung kepada keluaran dari fase sebelumnya. Hubungan penting antar fase digambarkan dengan panah. Sebagai contoh, jika fase berada pada fase *modeling*, berdasarkan pada perilaku dan karakteristik model, proses mungkin harus kembali kepada fase *data preparation* untuk perbaikan lebih lanjut terhadap data atau berpindah maju kepada fase *evaluation*.



Gambar 2. 2 Proses *data mining* menurut CRISP-DM (Larose, 2005)

Enam fase CRISP-DM (Larose, 2005) :

1. Fase Pemahaman Bisnis (*Business Understanding Phase*)
 - a. Penentuan tujuan proyek dan kebutuhan secara detail dalam lingkup bisnis atau unit penelitian secara keseluruhan.
 - b. Menerjemahkan tujuan dan Batasan menjadi formula dari permasalahan *data mining*.
 - c. Menyiapkan strategi awal untuk mencapai tujuan.
2. Fase Pemahaman Data (*Data Understanding Phase*)
 - a. Mengumpulkan data.
 - b. Menggunakan analisis penyelidikan data untuk mengenali lebih lanjut data dan pencarian pengetahuan awal.
 - c. Mengevaluasi kualitas data.
 - d. Jika diinginkan, pilih sebagian kecil group data yang mungkin mengandung pola dari permasalahan.

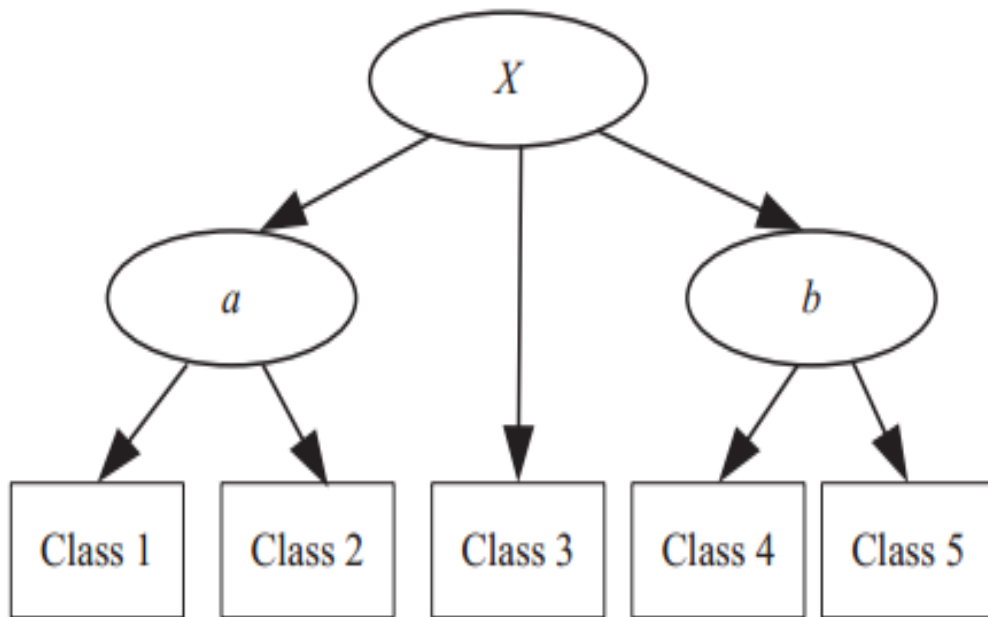
3. Fase Pengolahan Data (*Data Preparation Phase*)
 - a. Siapkan dari data awal, kumpulkan data yang akan digunakan.
 - b. Pilih kasus dan variable yang ingin dianalisis dan yang sesuai analisis yang akan dilakukan.
 - c. Lakukan perubahan pada beberapa *variable* jika dibutuhkan.
 - d. Siapkan data awal sehingga siap untuk perangkat pemodelan.
4. Fase pemodelan (*Modeling Phase*)
 - a. Pilih dan aplikasi teknik pemodelan yang sesuai.
 - b. Kalibrasi aturan model untuk mengoptimalkan hasil.
 - c. Perlu diperhatikan bahwa beberapa teknik mungkin untuk digunakan pada permasalahan *data mining* yang sama.
 - d. Jika diperlukan, proses dapat kembali ke fase pengolahan data untuk menjadikan data ke dalam bentuk yang sesuai dengan spesifikasi kebutuhan teknik *data mining* tertentu.
5. Fase Evaluasi (*Evaluation Phase*)
 - a. Mengevaluasi satu atau lebih model yang digunakan dalam fase pemodelan untuk mendapatkan kualitas dan efektivitas sebelum disebarkan untuk digunakan.
 - b. Menetapkan apakah terdapat model yang memenuhi tujuan pada fase awal.
 - c. Menentukan apakah terdapat permasalahan penting dari bisnis atau penelitian yang tidak tertangani dengan baik.
 - d. Mengambil keputusan berkaitan dengan penggunaan hasil dari *data mining*.

6. Fase Penyebaran (*Deployment Phase*)
 - a. Menggunakan model yang dihasilkan. Terbentuknya model tidak menandakan telah terselesaikannya proyek.
 - b. Contoh sederhana penyebaran : Pembuatan laporan.
 - c. Contoh kompleks penyebaran : Penerapan proses *data mining* secara parallel pada departemen lain.

2.4. Decision tree

Menurut (Kusrini & Luthfi, 2009), *Decision tree* atau 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.

Decision tree adalah pohon yang ada dalam analisis pemecahan masalah, pemetaan mengenai alternatif-alternatif pemecahan masalah yang dapat diambil dari masalah. *Decision tree* adalah salah satu metode klasifikasi yang banyak digunakan karena mudah untuk dibaca dan diinterpretasi oleh manusia (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). *Decision tree* adalah model prediksi menggunakan struktur pohon atau struktur berhirarki. Konsep dari *decision tree* adalah mengubah data menjadi pohon keputusan dan aturan-aturan keputusan. *Decisio tree* ini cocok untuk memprediksi hasil kategoris dan kurang sesuai untuk aplikasi dengan data time series



Gambar 2. 3 Model *Decision Tree* (Dongming, et al., 2016)

2.5. Algoritma C4.5

Algoritma C4.5 adalah algoritma yang digunakan untuk membentuk pohon keputusan (Dhika & Destiwati, 2015). Algoritma C4.5 merupakan bagian dari kelompok algoritma *decision trees*. Algoritma C4.5 diperkenalkan oleh J. Ross Quinlan diakhir tahun 1970 hingga awal tahun 1980-an. J. Ross Quinlan seorang peneliti dibidang mesin pembelajaran yang merupakan pengembangan dari algoritma ID3 (*Iterative Dichotomiser*), algoritma tersebut digunakan untuk membentuk pohon keputusan (Jailani, Defit, & Nurcahyo, 2015).

Algoritma C4.5 membaca seluruh sampel data training dari storage dan memuatnya ke memori. Hal inilah yang menjadi salah satu kelemahan algoritma C4.5 dalam kategori “skalabilitas” adalah algoritma ini hanya dapat digunakan jika data training dapat disimpan secara keseluruhan dan pada waktu yang bersamaan di memori (Kamagi & Hansun, 2015). Didalam pohon keputusan node

pusat merupakan attribute dari data yang diuji (tuple), cabang merupakan hasil dari pengujian atribut, dan daun merupakan kelas yang terbentuk (Han, Kamber, & Pei, 2012)

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 (Julianto, Yunitarini, & Sophan, 2014).

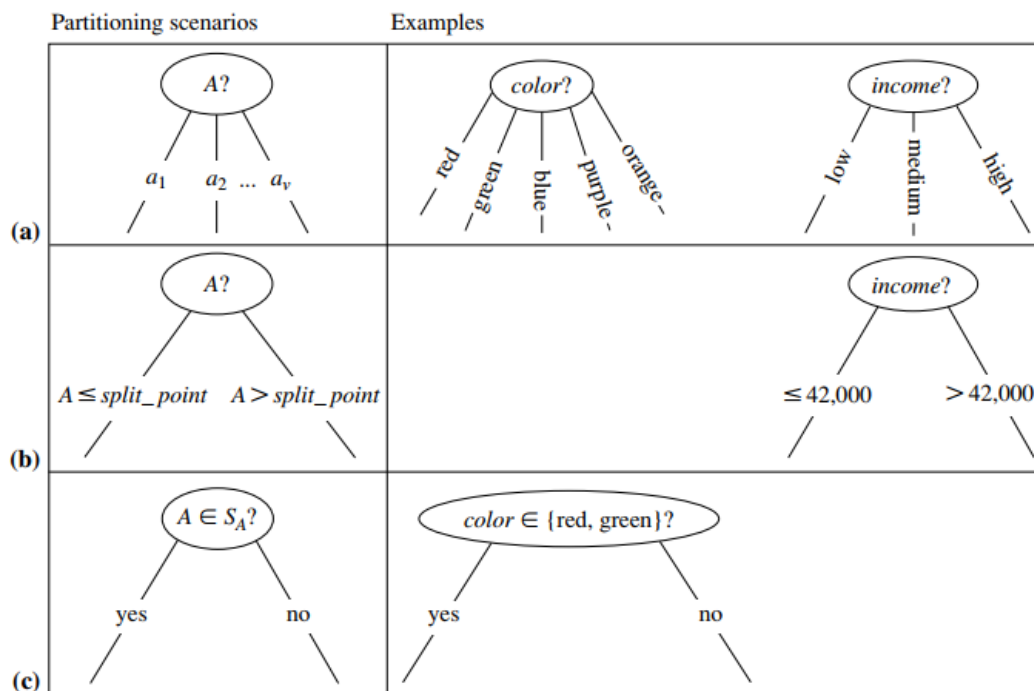
Menurut (Kusrini & Luthfi, 2009), secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut :

1. Pilih attribute 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.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. *Entropy* digunakan untuk menentukan seberapa informatif sebuah masukan atribut untuk menghasilkan sebuah atribut (Kamagi & Hansun, 2015). Dalam pembuatan cabang, ada tiga kemungkinan yang akan terjadi sesuai dengan jenis data. Jika A merupakan salah satu atribut dari data

yang diuji, maka kemungkinan yang akan terjadi akan bergantung pada nilai A (Han, Kamber, & Pei, 2012) :

- 1) Jika A bernilai diskrit, maka cabang akan terbentuk untuk setiap nilai A. Atribut A akan dikeluarkan dari daftar atribut yang perlu diperiksa karena setelah cabang A terbentuk nilai atribut A pada setiap cabang akan selalu sama.
- 2) Jika A bernilai kontinyu, maka akan terbentuk 2 buah cabang, dimana $A \leq$ dari nilai perpecahan, dan $A >$ nilai perpecahan. Nilai perpecahan ditentukan oleh metode pemilihan atribut saat membuat daftar attribute.
- 3) Jika A bernilai Diskrit dan biner, maka akan terbentuk 2 cabang yaitu cabang untuk nilai benar, dan cabang untuk nilai salah.



Gambar 2. 4 Pembuatan cabang pohon keputusan (Han, Kamber, & Pei, 2012).

2.5.1. Attribute Selection Measures

Attribute Selection Measures adalah teknik pencarian heuristik untuk memilih kriteria pemisah (*split criteria*) yang terbaik memisahkan partisi data tertentu, *Data partition* (D) , dari label *class training* ke kelas individu (Han, Kamber, & Pei, 2012). Jika kita membagi *data partition* (D) menjadi partisi yang lebih kecil sesuai dengan hasil kriteria pemisahan, idealnya setiap partisi akan murni (yaitu, semua tupel yang masuk dalam partisi tertentu akan termasuk dalam kelas yang sama). Proses *Attribute Selection Measures*, yaitu :

1) *Information Gain*

Information Gain ini dirintis oleh Claude Shannon pada teori informasi, yang mempelajari nilai atau "isi informasi" dari pesan (Han, Kamber, & Pei, 2012). *Information Gain* menggunakan nilai tersebut sebagai acuan dalam menentukan atribut yang akan digunakan dalam menyusun pohon keputusan. *Info* (D) juga biasa disebut dengan *Entropy*, perhitungan nilai *Entropy* dapat dilihat pada persamaan 1 berikut :

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

Keterangan :

S : himpunan kasus

A : fitur

n : jumlah partisi S

p_i : proporsi dari S_i terhadap S

Untuk menghitung nilai *gain* digunakan rumus seperti tertera dalam persamaan 2 :

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (2)$$

Keterangan :

S : himpunan kasus

A : atribut

n : jumlah partisi atribut A

|S_i|: jumlah kasus pada partisi ke -i

|S| : jumlah kasus dalam S

2) *Gain Ratio*

Gain Ratio merupakan modifikasi dari *information gain* untuk mengurangi bias atribut yang memiliki banyak cabang (Han, Kamber, & Pei, 2012).

Untuk menghitung nilai *gain ratio* dengan persamaan 3. *Log2*

$$Gain Ratio(S, A) = \frac{Gain(S, A)}{Split Information(S, A)} \quad (3)$$

Sementara itu, perhitungan nilai *split information* dapat dilihat pada persamaan 4 berikut :

$$Split Information(S, A) = - \sum_{i=1}^n \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \quad (4)$$

Keterangan :

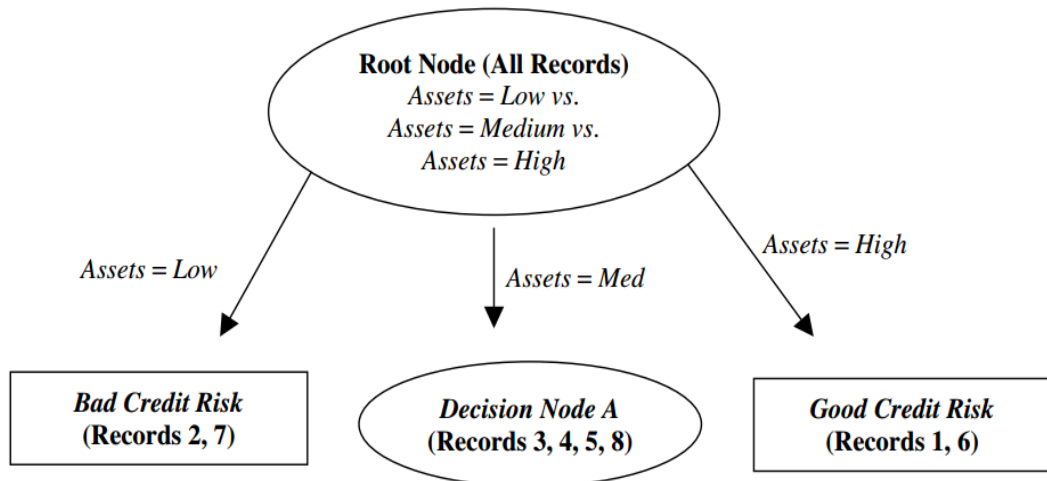
S : himpunan kasus

A : atribut

n : jumlah partisi atribut A

|S_i|: jumlah kasus pada partisi ke -i

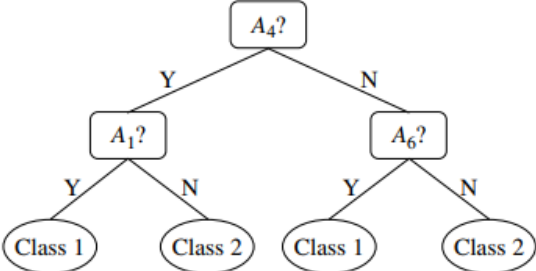
|S| : jumlah kasus dalam S



Gambar 2. 5 *Decision tree* Algoritma C4.5 (Larose, 2005)

2.6. Forward Selection

Pemilihan subset attribute berfungsi untuk mengurangi ukuran kumpulan data dengan menghapus atribut atau dimensi yang tidak relevan atau berlebihan (Han, Kamber, & Pei, 2012). Tujuan pemilihan subset atribut adalah untuk menemukan sekumpulan atribut minimum sehingga distribusi probabilitas yang dihasilkan dari kelas data sedekat mungkin dengan distribusi asli yang diperoleh dengan menggunakan semua atribut. Prosedur dimulai dengan himpunan kosong dari atribut sebagai set yang dikurangi, atribut yang terbaik dari atribut asli ditentukan dan ditambahkan pada set yang kurang. Pada setiap iterasi berikutnya yang terbaik dari atribut asli yang tersisa ditambahkan ke set.

Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ \Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ \Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  <pre> graph TD A4["A4?"] -- Y --> A1["A1?"] A4 -- N --> A6["A6?"] A1 -- Y --> C1_1((Class 1)) A1 -- N --> C2_1((Class 2)) A6 -- Y --> C1_2((Class 1)) A6 -- N --> C2_2((Class 2)) </pre> \Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$

Gambar 2. 6 Seleksi atribut metode greedy

2.7. Confusion Matrix

Confusion matrix adalah sebuah metode yang digunakan untuk mengevaluasi kinerja klasifikasi pohon keputusan (Tayefi, et al., 2017). *Confusion matrix* merupakan sebuah table yang terdiri dari banyaknya baris data uji yang diprediksi benar dan tidak benar oleh model klasifikasi. Tabel ini diperlukan untuk mengukur kinerja suatu model klasifikasi (Swastina L. , 2013).

Confusion matrix adalah alat yang berguna untuk menganalisis seberapa baik *classifier* Anda dapat mengenali tupel dari kelas yang berbeda (Han, Kamber, & Pei, 2012). Dalam pembuatan table *confusion matrix* ada empat hal yang harus diketahui, yaitu :

1. **True Positives (TP)** : jumlah tupel positif yang bernilai benar dari hasil klasifikasi pohon keputusan.
2. **True Negatives (TN)** : jumlah tupel negatif yang bernilai salah dari hasil klasifikasi pohon keputusan.

3. **False Positives (FP)** : jumlah tupel positif yang bernilai salah.
4. **False Positives (FP)** : jumlah tupel negatif yang bernilai salah.

Tabel 2. 1 Confusion Matrix

Classification as	Correct Classification		Total
	-	+	
-	True negative	False negative	N'
+	False positive	True positive	P'
Total	N	P	P+N

Confusion matrix, dapat menghitung hitung nilai *sensitivity (recall positive)*, *Specifity(recall negative)*, *precision*, dan *accuracy*. *Sensitivity* digunakan untuk membandingkan jumlah *True Positive (TP)* terhadap jumlah record yang positif sedangkan *Specifity*, *precision* dalah perbandingan jumlah *True Negative (TN)* terhadap jumlah record yang negatif. Untuk menghitung digunakan persamaan dibawah ini (Han, Kamber, & Pei, 2012) :

$$\text{Recall, True Positive Rate (Sensitivity)} = \frac{TP}{P} = \frac{TP}{TP+FN}$$

$$\text{Recall, True Negative Rate (Specifity)} = \frac{TN}{N} = \frac{TN}{TN+FP}$$

$$\text{Precision, Positive Predictive Value} = \frac{TP}{TP+FP}$$

$$\text{Precision, Negative Predictive Value} = \frac{TN}{TN+FN}$$

$$\text{Accuracy} = \frac{P}{(P+N)} + \frac{N}{(P+N)} = \frac{TP+TN}{P+N} = \frac{TP+TN}{\text{Jumlah Populasi (Classification as)}}$$

Ukuran tingkat kesalahan klasifikasi juga dapat dihitung dengan mencari

Error Rate:

$$\text{Error Rate} = \frac{FP+FN}{P+N} = \frac{FP+FN}{\text{Jumlah Populasi (Classification as)}}$$

2.8. ROC Curves

Kurva ROC (*Receiver Operating Characteristic*) merupakan alat visualisasi dari akurasi model dan perbandingan perbedaan antar model klasifikasi (Han, Kamber, & Pei, 2012). ROC adalah grafik dua dimensi dengan false positives sebagai garis horizontal dan true positive untuk mengukur perbedaan performansi metode yang digunakan (Gorunescu, 2011). Kurva ROC adalah teknik untuk memvisualisasi dan menguji kinerja pengklasifikasian berdasarkan performanya. Model klasifikasi yang lebih baik adalah yang mempunyai kurva ROC yang lebih besar. Performa keakurasian AUC dapat diklasifikasikan menjadi lima kelompok yaitu (Gorunescu, 2011):

- ✓ 0.90 - 1.00 = *Excellent classification* (sangat bagus)
- ✓ 0.80 - 0.90 = *Good classification* (bagus)
- ✓ 0.70 - 0.80 = *Fair classification* (cukup bagus)
- ✓ 0.60 - 0.70 = *Poor classification* (kurang bagus)
- ✓ 0.50 - 0.60 = *Failure* (gagal)

2.9. Uji Kompetensi Keahlian (UKK)

Peserta didik wajib menyelesaikan dan lulus Ujian Nasional (UN) sebagai salah satu syarat kelulusan. Menurut (Peraturan Menteri Pendidikan dan Kebudayaan RI Nomor 3, 2017), Ujian Nasional (UN) adalah kegiatan pengukuran capaian kompetensi lulusan pada mata pelajaran tertentu secara nasional dengan mengacu pada Standar Kompetensi Lulusan. Untuk jenjang Sekolah Menengah Kejuruan (SMK) peserta didik wajib menyelesaikan dan lulus

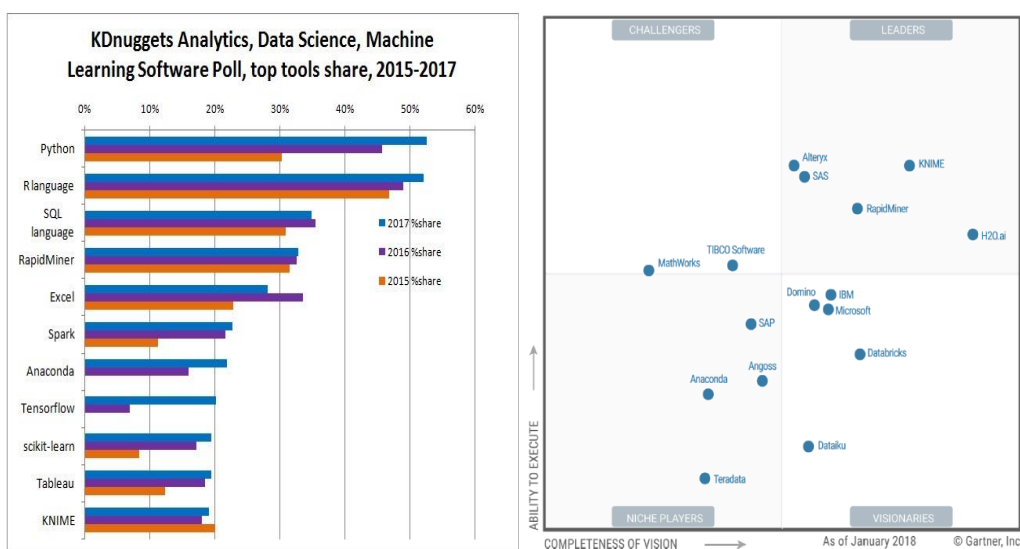
Uji Kompetensi Keahlian (UKK) sebagai syarat kelulusan. Menurut (Direktur Pembinaan SMK, 2017), Uji Kompetensi Keahlian (UKK) adalah ujian nasional yang terdiri atas ujian teori kejuruan dan ujian praktik kejuruan dimana proses penilaian baik teknis maupun non teknis melalui pengumpulan bukti yang relevan untuk menentukan apakah seseorang kompeten atau belum kompeten pada suatu unit kompetensi atau kualifikasi tertentu. Uji Kompetensi Keahlian (UKK) bertujuan untuk mengukur pencapaian kompetensi siswa pada level tertentu sesuai kompetensi keahlian yang ditempuh di Sekolah Menengah Kejuruan (SMK).

2.10. RapidMiner

RapidMiner merupakan perangkat lunak yang bersifat terbuka (*open source*). RapidMiner adalah sebuah solusi untuk melakukan analisis terhadap *data mining*, *text mining* dan analisis prediksi. RapidMiner menggunakan berbagai teknik deskriptif dan prediksi dalam memberikan wawasan kepada pengguna sehingga dapat membuat keputusan yang paling baik. RapidMiner memiliki kurang lebih 500 operator *data mining*, termasuk operator untuk *input*, *output*, *data preprocessing* dan visualisasi. RapidMiner merupakan *software* yang berdiri sendiri untuk analisis data dan sebagai mesin *data mining* yang dapat diintegrasikan pada produknya sendiri. RapidMiner ditulis dengan menggunakan bahasa java sehingga dapat bekerja di semua sistem operasi. Polling *Software data mining* yang dilakukan oleh KDnuggets pada tahun 2014 yang berjudul “*What Analytics, Data Mining, Data Science software/tools you used in the past 12 months for a real project Poll*” (KDnuggets, 2018), RapidMiner menempati urutan pertama dan dalam hasil *polling* yang dilakukan 3 tahun terakhir sejak

2015 – 2017 oleh KDnuggets berjudul “*New Leader, Trends, and Surprises in Analytics, Data Science, Machine Learning Software Poll*”, RapidMiner tetap menjadi platform umum yang paling populer untuk data mining, dengan sekitar 33% penggunaan.

Dalam laporan gartner yang berjudul “*Magic Quadrant for Data Science and Machine-Learning Platforms*” menempatkan posisi RapidMiner pada quadrant Leaders. Gartner berpendapat bahwa RapidMiner masih menjadi *Leader* sebagai *platform* yang lengkap dan mudah digunakan untuk *data science*.



Gambar 2. 7 KDnuggets Analytics / Data Science 2017 Software Poll

