

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Kelangsungan hidup dan pertumbuhan dari suatu perusahaan bukan hanya ditentukan dari keberhasilan dalam mengelola keuangan yang berdasarkan pada kekuatan modal atau uang semata, tetapi juga ditentukan dari keberhasilannya mengelola sumber daya manusia. Pengelolaan sumber daya manusia yang dimaksudkan adalah bahwa perusahaan harus mampu untuk menyatukan persepsi atau cara pandang karyawan dan pimpinan perusahaan dalam rangka mencapai tujuan perusahaan antara lain melalui pembentukan mental bekerja yang baik dengan dedikasi dan loyalitas yang tinggi terhadap pekerjaannya, memberikan motivasi kerja, bimbingan, pengarahan dan koordinasi yang baik dalam bekerja oleh seorang pemimpin kepada bawahannya.

Menciptakan kepuasan kerja karyawan adalah tidak mudah karena kepuasan kerja dapat tercipta jika variabel-variabel yang mempengaruhinya antara lain motivasi kerja, kepemimpinan dan budaya organisasi/perusahaan dapat diakomodasikan dengan baik dan diterima oleh semua karyawan di dalam suatu organisasi/perusahaan. Brahmasari dan Suprayetno (2008) mengemukakan bahwa kinerja organisasi tergantung dari kinerja individu atau dengan kata lain kinerja individu akan memberikan kontribusi pada kinerja organisasi, artinya bahwa perilaku anggota organisasi baik secara individu maupun kelompok memberikan kekuatan atas kinerja organisasi sebab motivasinya akan mempengaruhi pada kinerja organisasi.

Salah satu cara dalam melakukan evaluasi kinerja dalam dunia pendidikan adalah melalui pendekatan *data mining* dengan model pohon keputusan dan algoritma C4.5 adalah salah satu metode untuk membangun pohon keputusan. Kamagi (2014) dalam penelitiannya mengimplementasikan algoritma C4.5 untuk memprediksi tingkat kelulusan mahasiswa dan menghasilkan prediksi dengan tingkat keakuratan yang tinggi, yaitu 87,5%.

Sesuai dengan Perak UNDIP Nomor 11 tahun 2017 tentang pedoman perhitungan insentif kinerja wajib, insentif kelebihan kinerja pegawai tetap Universitas Diponegoro, dan beban kerja dosen, maka dosen diwajibkan untuk menyusun Sasaran Kerja Pegawai (SKP) sebagai dasar penilaian prestasi kerja mereka.

Sasaran Kinerja Pegawai (SKP) adalah rencana dan target kinerja yang harus dicapai oleh pegawai dalam kurun waktu penilaian yang bersifat nyata dan dapat diukur serta disepakati pegawai dan atasannya. Menurut peraturan rektor tersebut, setiap dosen wajib menyusun SKP (Sasaran Kerja Pegawai), yang dalam pelaksanaannya harus sesuai dengan tanggung jawab dan kewajiban yang diatur oleh pejabat terkait. SKP merupakan pengganti DP3 (Daftar Penilaian Prestasi Pegawai) yang mulai awal 2014 ini ditiadakan. Penilaiannya diharapkan lebih komprehensif dibandingkan dengan DP3.

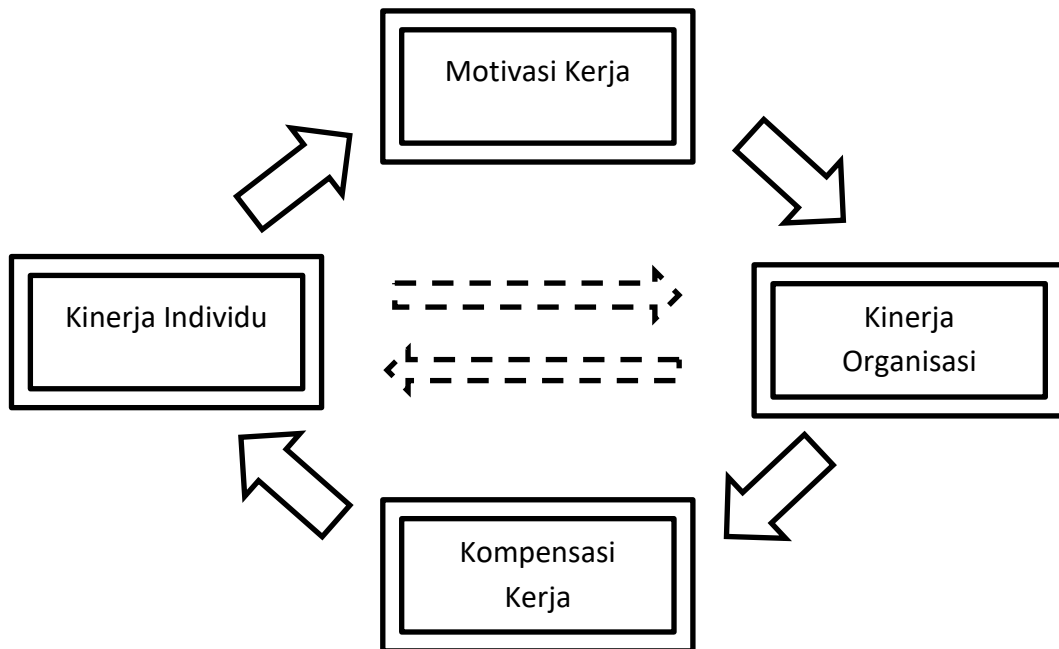
2.2 Dasar Teori

2.2.1 Pengaruh Kinerja Individu Terhadap Kinerja Organisasi

Menurut Brahmasari dan Suprayetno (2008) istilah kinerja merupakan suatu hasil kerja yang dicapai seorang dalam melaksanakan tugas-tugas yang diwajibkan kepadanya yang didasarkan atas kecakapan, pengalaman dan kesungguhan serta waktu. Terdapat tiga faktor utama yang berpengaruh pada kinerja pada seorang individu yaitu kemampuan bekerja, usaha kerja (keinginan untuk bekerja), dan dukungan organisasional (kompensasi kerja).

Ailabouni dkk (2009) dalam jurnalnya menunjukkan bahwa pemberian insentif secara praktis dapat meningkatkan kinerja bagi tenaga kerja, dengan pemberian insentif mereka dapat meningkatkan produktivitas, menurunkan biaya produksi, memperpendek waktu dalam membangun proyek dan tentunya dapat meningkatkan penghasilan para pekerja. Dalam jurnal tersebut juga mendaftar faktor yang memotivasi dalam pekerjaan seperti penghasilan lembur, tunjangan kesehatan, penyediaan transportasi, promosi, kenaikan gaji, pengakuan, kebijakan perusahaan, kondisi kerja, hubungan dengan rekan kerja. Dalam jurnalnya juga menunjukkan bahwa ada hubungan yang jelas antara remunerasi, motivasi dan

efisiensi pada pekerjaan dan juga menekankan bahwa produktivitas dan semangat tenaga kerja adalah kunci faktor dalam mengukur produktivitas organisasi.



Gambar 2.1 Hubungan Kinerja Individu dengan Kinerja Organisasi

Ailabouni dkk (2009), menggambarkan empat kategori faktor utama yang mempengaruhi produktivitas organisasi berdasarkan *Association of Researchers in Construction Management (ARCOM)* yaitu faktor lingkungan kerja, faktor organisasi, faktor grup dan faktor individu.

2.2.2 Kinerja Perguruan Tinggi

Sebelum mengetahui bagaimana cara untuk menganalisis kinerja perguruan tinggi berikut ini ada beberapa definisi pada kinerja. Menurut Lockett (1992), kinerja didefinisikan sebagai konstruksi multidimensi dan faktor-faktor umum yang sering dikaitkan dengan organisasi kinerja seperti efisiensi, kualitas, responsif, biaya dan efektivitas keseluruhan. Armstrong (1994) juga memperluas definisi untuk kinerja manajemen sebagai sarana untuk mendapatkan lebih baik hasil dari organisasi, tim dan individu dengan memahami dan mengelola kinerja dalam rangka menyepakati rencana tujuan, sasaran, dan standar prestasi dan kompetensi.

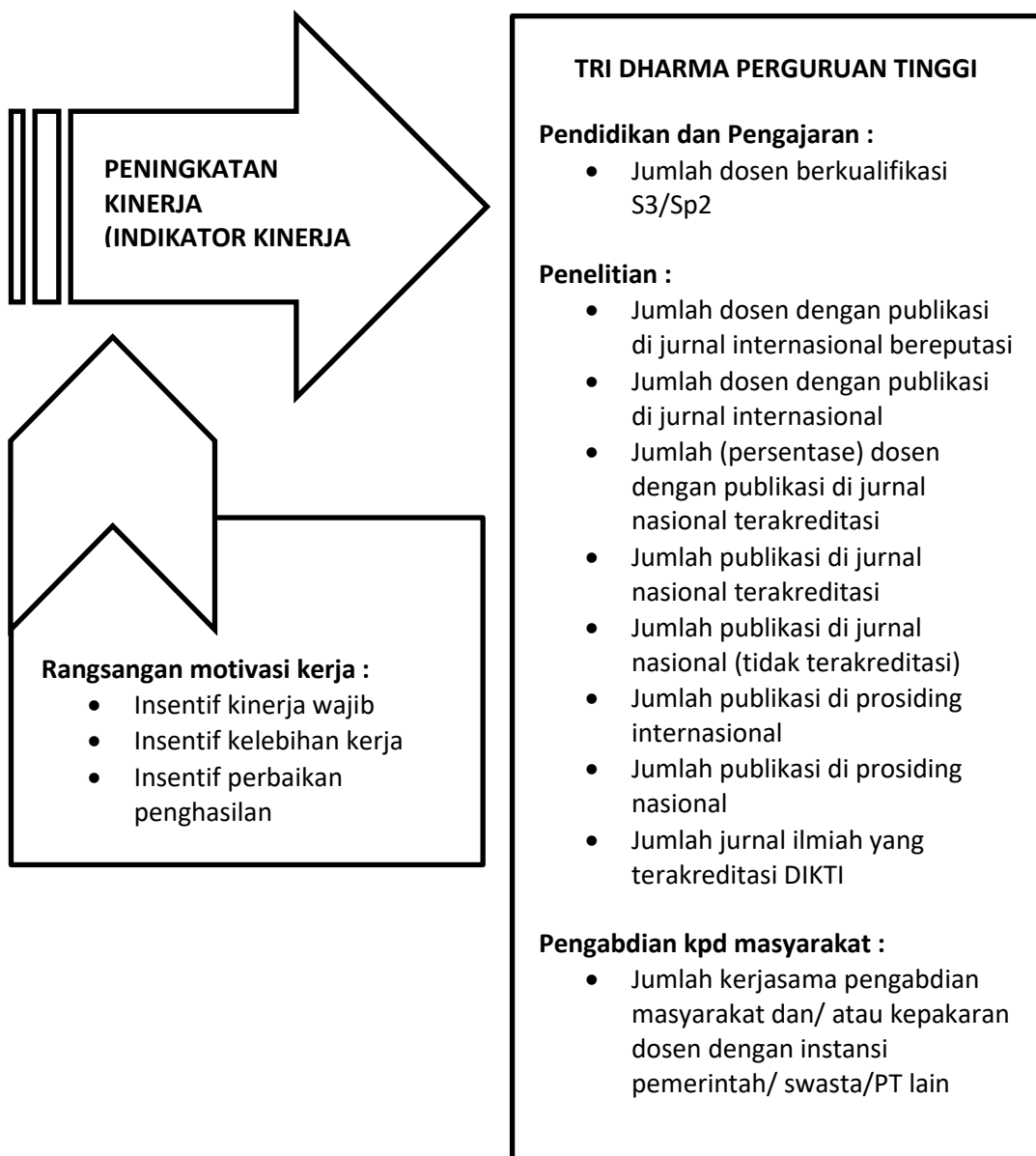
Evaluasi kinerja universitas yang telah banyak dilakukan oleh banyak universitas adalah berdasarkan *key performance indicator* (KPI) mereka seperti pengajaran, penelitian, pengawasan, publikasi dan konsultasi. Meskipun indikator-indikator tersebut adalah elemen penting dalam pengukuran kinerja dari staf akademik dari sebuah perguruan tinggi, terdapat satu lagi aspek kinerja yang diabaikan dan diantisipasi karena memiliki konsekuensi negatif jika otoritas perguruan tinggi tidak berencana untuk benar-benar mengamati masalah ini.

Indikator tersebut adalah berupa penghormatan atas kontribusi kinerja dari staf akademik contohnya seperti tunjangan dan uang kinerja. Populasi dari studi kasus tersebut ini ditentukan oleh *purposive sampling* dan terdiri dari staf, profesor, profesor Madya, dosen senior dari universitas publik di semua kategori di Malaysia. Hasil penelitian menunjukkan bahwa indikator kinerja tersebut memiliki implikasi positif dan signifikan pada staf akademik (Masron dkk, 2012).

Kerangka analisis kinerja yang akan dibangun yang berdasarkan kerangka analisis yang dirancang oleh Masron dkk (2012) dalam menghubungkan antara penghargaan kinerja staf pengajar (dosen) dalam perguruan tinggi dengan indikator kinerja universitas (IKU) sebagai target kinerja organisasi di gambarkan seperti pada Gambar 2.2.

Sehubungan dengan indikator kinerja universitas sebagai akibat dari tekanan akuntabilitas dalam sektor publik, pengukuran kinerja menjadi agenda di lembaga-lembaga pendidikan tinggi. Chen dkk (2006) menyebutkan bahwa menghadapi tantangan persaingan, perlu adanya tindakan yang diambil untuk mereformasi kegiatan-kegiatan yang ada pada lembaga pendidikan tinggi.

Dalam menanggapi kebutuhan ini, pemerintah dan perguruan tinggi telah berusaha untuk membuat kebijakan dan merancang praktek terbaik untuk mengukur, mendorong dan menghargai kinerja staf akademik seperti dosen, dalam kinerja mereka meliputi pendidikan, pengajaran, penelitian dan publikasi. Kebijakan tersebut selain digunakan untuk meningkatkan kinerja para dosen juga dapat sekaligus meningkatkan kinerja universitas karena kinerja dosen merupakan salah satu indikator kinerja universitas.



Gambar 2.2 *Framework KPI Universitas.*

Dari indikator kinerja pada Gambar 2.2 maka dibuatlah target yang digunakan sebagai acuan atau tolok ukur peningkatan kinerja dari perguruan tinggi. Visi Universitas Diponegoro sebagai salah satu universitas riset yang unggul memiliki target dan capaian dari indikator kinerja yang disebut IKU (Indikator Kinerja Universitas) sebagai bagian dari rencana strategis perguruan tinggi.

2.2.3 *Data mining dan Educational Data mining*

Data mining merupakan langkah-langkah analisis dari proses "penemuan pengetahuan dalam basis data". Dengan *data mining* dilakukan proses analisis data dari berbagai perspektif dan merangkumnya menjadi informasi yang bermanfaat. Dalam proses *data mining* diperlukan adanya perangkat lunak sebagai salah satu dari sejumlah alat analitis untuk menganalisis data. Dengan bantuan perangkat lunak ini dimungkinkan pengguna dapat menganalisis data dari berbagai dimensi atau sudut, mengkategorikannya, dan merangkum hubungan yang diidentifikasi. Dalam beberapa tahun terakhir, telah ada peningkatan minat dalam penggunaan *data mining* untuk menyelidiki beberapa masalah-masalah ilmiah dalam dunia pendidikan (Romero dkk, 2010). Dalam penelitian *data mining* tersebut menerapkan beberapa teknik seperti *K- nearest neighbor*, *decision tree*, *Naïve Bayes*, *Neural network*, *Fuzzy* dan lainnya (Carlos dkk, 2013).

Algoritma yang digunakan dalam *data mining* memberikan suatu mekanisme untuk membuat model pemrosesan data. Untuk membuat model dalam algoritma *data mining* tersebut pertama-tama menganalisis sekumpulan data dan mencari pola dan tren tertentu. Hasil analisis algoritma tersebut menentukan parameter model penambangan, parameter ini diterapkan pada seluruh rangkaian data untuk mengekstraksi pola yang layak dan informasi statistik terperinci. Kriteria pada evaluasi dalam *data mining* digunakan untuk memprediksi kelas target dan beberapa set data uji lainnya dapat digunakan untuk mengevaluasi akurasi.

Han dkk (2014) menjelaskan bahwa ada 3 proses dalam *data mining* yang pertama adalah eksplorasi data. Eksplorasi data terdiri dari pembersihan data, transformasi data, normalisasi data, penanganan data yang salah, pengurangan dimensi data, pemilihan subset fitur, dan sebagainya. Tahap kedua adalah pembangunan model dan validasinya, pada tahap ini dilakukan pembangunan model dan melakukan validasi terhadap model tersebut dan memilih model yang memiliki performa prediksi terbaik. Metode-metode yang dapat digunakan untuk membangun dan melakukan validasi model adalah metode klasifikasi, regresi, analisis cluster, deteksi anomali, analisis asosiasi, analisis pola sekuensial, dan sebagainya.

Kemudian yang terakhir adalah penerapan model. Langkah terakhir dalam proses *data mining* ini menerapkan model pada data baru untuk menghasilkan peramalan atau estimasi yang cocok terhadap masalah-masalah yang diinvestigasi. Pada proses tahap ini diharapkan dapat mengidentifikasi pola yang muncul secara berulang pada suatu data dan mengubah pola tersebut menjadi aturan dan kriteria yang dapat mudah dimengerti oleh *user*.

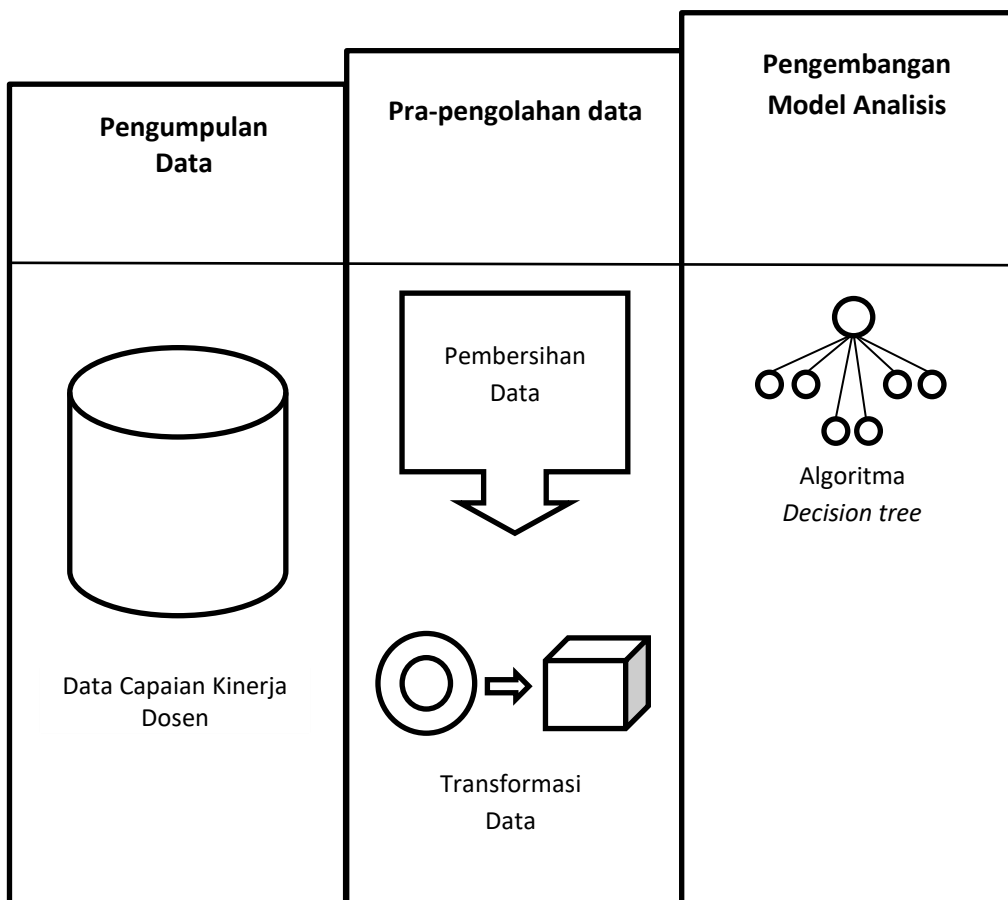
Konsep klasifikasi menurut Han (2014) adalah klasifikasi dalam *data mining* adalah turunan dari suatu fungsi atau model yang menggambarkan kelas dari suatu objek berdasarkan atributnya. Atribut adalah *field* data yang menggambarkan karakteristik atau fitur dalam suatu objek data. Fungsi atau model klasifikasi dibangun dengan menganalisis hubungan antara atribut dan kelas objek dalam *dataset* pelatihan (*training set*). Han juga menyatakan bahwa komponen dasar dari proses klasifikasi terdiri dari empat komponen, yaitu kelas (*class*), prediktor (*predictors*), set data pelatihan (*training dataset*), dan set data pengujian (*testing dataset*)

Salah satu penerapan *data mining* adalah pada sektor pendidikan atau yang disebut *educational data mining* (EDM), EDM dapat didefinisikan sebagai aplikasi teknik pertambangan data tradisional untuk analisis data pendidikan yang bertujuan untuk memecahkan masalah dalam konteks pendidikan (Baker dan Yacef, 2009). Beberapa aplikasi EDM terdiri dari pengembangan sistem *e-learning* (Lara dkk, 2014), dukungan pedagogik (Hung dan Crooks, 2009), clustering data pendidikan (Chakraborty dkk, 2016), dan untuk memprediksi prestasi pada siswa (Kabra dan Bichkar, 2011).

EDM menggunakan database dalam sistem informasi dunia pendidikan untuk memahami karakter siswa dan gaya pembelajaran yang lebih komprehensif dalam upaya untuk merancang kebijakan pendidikan yang akan meningkatkan kinerja akademis mereka, dan mengurangi tingkat kegagalan pada akhir masing-masing tahun sekolah. EDM memfasilitasi, misalnya, penemuan pola-pola baru tentang proses pembelajaran siswa. Menggunakan model ini, seseorang dapat memvalidasi dan mengevaluasi beberapa aspek dari sistem pendidikan dengan tujuan meningkatkan kualitas pendidikan (Romero dkk, 2010).

2.2.4 Metode Analisis *Data mining*

Metode *data mining* yang digunakan dalam penelitian ini terbagi ke dalam 3 tahapan yakni pengumpulan data (*data collection*), pra-proses data (*data pre-processing*), dan pengembangan model analisis (*analysis model development*) seperti yang terlihat pada Gambar 2.3.



Gambar 2.3 Metode analisis *data mining*

1. Pengumpulan Data

Sumber data yang digunakan dalam penelitian ini yakni data dosen yang didapat dari sistem informasi kepegawaian universitas. Data yang digunakan dalam penelitian ini adalah data jumlah pendidikan, jumlah penelitian, dan jumlah pengabdian kepada masyarakat yang merupakan kewajiban dosen dalam Tridharma perguruan tinggi.

2. **Pra Pengolahan Data**

Setelah data dikumpulkan, langkah selanjutnya adalah melakukan pra-proses data (*data preprocessing*). Teknik praproses data yang digunakan dalam penelitian ini dibagi ke dalam empat bagian yakni pembersihan data (*data cleaning*). Pada tahap ini data yang tidak lengkap, data yang tidak konsisten maupun data yang tidak memiliki nilai (*missing values*) akan dibersihkan.

3. **Pengembangan Model Analisis**

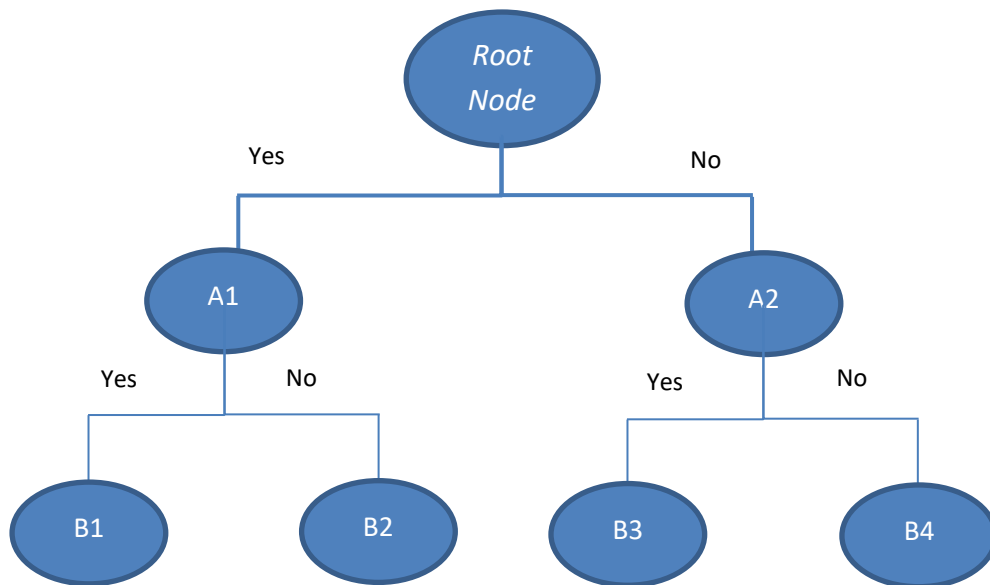
Tahapan selanjutnya dari penelitian ini yakni tahap pengembangan model analisis. Pada tahapan ini teknik *data mining* dan *machine learning* diaplikasikan. Algoritma yang digunakan pada penelitian ini adalah algoritma *C4.5 Decision tree*.

2.2.5 **Pohon Keputusan (*Decision tree*)**

Decision tree atau Pohon Keputusan adalah sebuah algoritma dengan pendekatan *divide-and-conquer* untuk klasifikasi suatu masalah. Pendekatan algoritma ini menggunakan struktur sebuah pohon, dimana setiap *node* dari pohon merepresentasikan atribut yang telah diuji, setiap cabang merupakan suatu pembagian hasil uji, dan *node* daun atau *leaf* merepresentasikan kelompok kelas tertentu. Level *node* teratas dari sebuah *Decision tree* adalah *node* akar (*root*) yang biasanya berupa atribut yang paling memiliki pengaruh terbesar pada suatu kelas tertentu.

Secara teknis algoritma *decision tree* melakukan strategi pencarian masalah secara *top-down*. Pada proses mengklasifikasi data yang tidak diketahui, nilai atribut akan diuji dengan cara melacak jalur dari *root node* sampai *leaf node* dan kemudian melakukan prediksi kelas yang dijadikan target pada data *testing*. Secara detail algoritma *decision tree* mencari nilai kemungkinan terbaik pada setiap atribut kemudian membaginya ke atribut di bawahnya, dan memproses secara rekursif pada submasalah yang dihasilkan dari pembagian tersebut. Strategi ini menghasilkan sebuah *decision tree* yang dapat diubah menjadi satu set *classification rules* (Witten dkk, 2011).

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*. Seperti ditunjukkan dalam Gambar 2.4, *decision tree* tergantung pada aturan *if-then*, tetapi tidak membutuhkan parameter dan metrik. Struktur sederhana dan dapat ditafsirkan memungkinkan *decision tree* untuk memecahkan masalah atribut *multi-type*. *Decision tree* juga dapat melakukan pemrosesan data dengan nilai-nilai yang hilang atau data noise (Dua dan Xian, 2011).



Gambar 2.4 Struktur *decision tree* (Dua dan Xian, 2011)

Dasar dari algoritma pohon keputusan adalah algoritma *greedy* yang membangun sebuah pohon keputusan dari atas kebawah secara rekursif secara *divide and conquer* (membagi dan memecah) (Han dkk, 2014). Berikut ini adalah pseudocode dari algoritma pohon keputusan (*decision tree*) :

Algoritma: Generate_decision_tree

Input : data training samples; list of attributes; attribute_selection_method.

Output : Pohon keputusan

Metode :

- (1) Buat simpul N
- (2) **IF** sampel memiliki kelas yang sama, C, **THEN**
- (3) **RETURN** N sebagai simpul daun yang berlabel kelas C;
- (4) **IF** daftar atribut kosong **THEN**
- (5) **RETURN** N sebagai simpul daun yang dilabeli dengan kelas mayoritas dalam sampel.
- (6) Memilih *attribute-test* (atribut yang diuji) yang memiliki nilai Gain Ratio terbesar.
- (7) Menjadikan *node* N sebagai label *test-attribute*.
- (8) **FOR EACH** a_i pada *test-attribute*;
- (9) Menambahkan cabang pada *node* N untuk dijadikan atribut uji dengan nilai *test-attribute* = a_i
- (10) Membuat partisi untuk sampel S_i dari sampel dimana *test-attribute* = a_i
- (11) **IF** $S_i = \text{empty}$ **THEN**
- (12) Menjadikan *node* leaf dengan kelas yang paling banyak pada himpunan sampel;
- (13) **ELSE** menjadikan *node* yang dihasilkan oleh Generate_decision_tree (S_i , attribute-list, test-attribute);
- (14) **ENDFOR**
- (15) **RETURN** N;

2.2.6 Algoritma C4.5

Selama akhir 1970-an dan awal 1980-an, J. Ross Quinlan, seorang peneliti dalam pembelajaran mesin, mengembangkan algoritma pohon keputusan yang dikenal sebagai ID3 (*Iterative Dichotomiser*). Quinlan kemudian mengembangkan algoritma ID3 tersebut menjadi algoritma C4.5, algoritma inilah yang menjadi tolak

ukur yang sering dibandingkan dengan algoritma pembelajaran yang lebih baru (Han dkk, 2014).

Data mining dengan algoritma C4.5 diperkenalkan oleh Quinlan mengacu pada model klasifikasi juga disebut sebagai pohon keputusan, dari data yang diamati. Dalam kumpulan data yang diamati, setiap catatan berisi struktur data yang sama. Data dapat memiliki sejumlah atribut atau pasangan nilai. Salah satu atribut ini mewakili kategori catatan. Masalahnya adalah untuk membangun pohon keputusan berdasarkan pengamatan tentang atribut non-kategori memprediksi dengan benar nilai atribut kategori. Atribut kategori dapat mengambil nilai seperti {benar, salah}, atau {diprediksi, tidak diprediksi}, atau {sukses, gagal}, atau sesuatu yang setara.

Banyak algoritma yang dapat dipakai dalam pembentukan pohon keputusan, antara lain ID3, CART, dan C.45. Sedangkan algoritma C.45 sendiri merupakan pengembangan dari algoritma ID3. Secara umum algoritma C4.5 digunakan untuk membangun pohon keputusan adalah sebagai berikut (Quinlan, 1993) :

1. Menghitung jumlah kasus total, jumlah kasus masing masing kelas, dan Entropi dari semua kasus dan kasus yang dibagi berdasarkan nilai atribut. Entropi digunakan untuk menentukan seberapa informatif sebuah *input* atribut untuk menghasilkan *output* atribut. Rumus dasar dari Entropi tersebut adalah sebagai berikut :

$$Entropi (S) = \sum_{i=1}^k - p_i \log_2 p_i \quad (2.1)$$

dimana :

S adalah populasi pada kasus yang digunakan

k adalah banyaknya kelas atau partisi pada S

p_i adalah probabilitas dari atribut i terhadap S

Entropi bisa dikatakan sebagai kebutuhan bit untuk menyatakan suatu kelas. Semakin kecil nilai Entropi maka semakin baik digunakan dalam mengekstraksi suatu kelas. Apabila nilai Entropi = 0 maka otomatis pohon keputusan akan berhenti pada *node* tersebut dan atribut tersebut menjadi *leaf node*.

- Menghitung *Information Gain* dari setiap variabel dengan menggunakan rumus seperti tertera dalam persamaan :

$$Gain(S, A) = Entropi(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \times Entropy(S_i) \quad (2.2)$$

dimana :

S adalah populasi pada kasus yang digunakan

A adalah atribut yang dipilih

n adalah banyaknya kelas atau partisi pada A .

S_i adalah jumlah populasi pada atribut i .

- Menghitung *Split Info* untuk setiap atribut.

$$Split\ Info(S, A) = - \sum_{i=1}^n \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \quad (2.3)$$

- Menghitung *Gain Ratio* untuk setiap atribut.

$$Gain\ Ratio(A) = \frac{Gain(S,A)}{SplitInfo(S,A)} \quad (2.4)$$

- Memilih atribut dengan nilai *Gain Ratio* terbesar sebagai *node*.
- Membagi data berdasarkan nilai atribut dari atribut terpilih. Kemudian menggunakannya untuk melakukan langkah selanjutnya
- Ulangi langkah 1 sampai 6 hingga seluruh atribut digunakan atau memenuhi suatu kondisi berhenti yaitu apabila seluruh kelas tidak ada yang tersisa untuk dipilih, kondisi ini terjadi bila seluruh percabangan telah dibentuk dan tidak dimungkinkan lagi untuk dibentuk percabangan baru.

2.2.7 *Forward Chaining*

Forward Chaining atau *forward reasoning* adalah salah satu dari dua metode utama penalaran saat menggunakan mesin inferensi dan dapat digambarkan secara logis sebagai aplikasi modus ponens yang diulang. *Forward Chaining* merupakan salah satu strategi implementasi populer untuk sistem pakar, sistem aturan dalam bisnis dan produksi. Kebalikan dari *Forward Chaining* adalah *backward chaining*.

Dasar dari metode *Forward Chaining* merupakan pelacakan ke depan yang dimulai dari sekumpulan fakta-fakta dengan mencari kaidah yang cocok dengan dugaan/hipotesa yang ada menuju kesimpulan. *Forward Chaining* dimulai dengan

data yang tersedia dan menggunakan aturan inferensi untuk mengekstraksi lebih banyak data (dari pengguna akhir, misalnya) hingga tujuan tercapai. Mesin inferensi menggunakan *Forward Chaining* mencari aturan inferensi sampai menemukan satu di mana anteseden (klausa IF) diketahui benar. Ketika aturan tersebut ditemukan, mesin dapat menyimpulkan, atau menyimpulkan, konsekuensinya (klausa THEN), menghasilkan penambahan informasi baru ke datanya. (Edward, 1988).

Menurut Giarratano dan Riley (2005), *Forward Chaining* adalah salah satu metode dari sistem pakar yang mencari atau menelusuri solusi melalui masalah. Dengan kata lain metode ini melakukan pertimbangan dari fakta-fakta yang kemudian berujung pada sebuah kesimpulan yang berdasarkan pada fakta-fakta. Metode ini merupakan kebalikan dari metode *backward chaining* yang melakukan pencarian yang berawal dari hipotesis menuju ke fakta-fakta untuk mendukung hipotesis tersebut. Fakta yang ada disebut sebagai bukti atau kesimpulan dalam metode *backward chaining* yang mengindikasikan jika fakta tersebut digunakan untuk mendukung hipotesa, dengan cara yang sama bukti atau kesimpulan yang dikehendaki digunakan untuk membuktikan sebuah *rule*. Secara khusus, penjelasan menjadi sederhana dalam *backward chaining* karena sistem dapat dengan mudah menjelaskan dengan tepat tujuan yang ingin dicapai.

Sedangkan menurut Lucas dan Van der Gaag (1991), *Forward Chaining* adalah model komputasi dari bawah ke atas. Model penalaran yang dimulai dengan serangkaian fakta-fakta yang diketahui dan diterapkan pada aturan untuk menghasilkan fakta-fakta baru yang sesuai dengan fakta-fakta dikenal, dan melanjutkan proses penalaran ini sampai mencapai tujuan yang telah ditetapkan, atau sampai ada fakta-fakta yang lebih lanjut dapat diperoleh yang sesuai dengan fakta-fakta yang diketahui. Metode penalaran ini memeriksa fakta-fakta terhadap permintaan atau tujuan yang telah ditentukan, dan menunjukkan bahwa kesimpulan bergerak maju terhadap fakta-fakta yang dituju. *Forward Chaining* juga digunakan untuk meningkatkan dan mengembangkan permodelan *Expert Systems* (ES) dan pemodelan otak manusia di ranah *Artificial Intelligence* (AI).

2.2.8 Pengukuran Kinerja dan Validasi Sistem

1. *Confusion Matrix*

Pengukuran kinerja dari suatu algoritma klasifikasi sangat penting dilakukan, hal ini akan menggambarkan seberapa baik sistem dalam menjalankan klasifikasi data. Dalam penelitian ini untuk mengukur kinerja *Decision tree* dengan algoritma C4.5 menggunakan *Confusion Matrix*. *Confusion Matrix* bekerja dengan membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil yang seharusnya. Dalam *Confusion Matrix* terdapat empat istilah sebagai representasi hasil dari proses klasifikasi, keempat istilah itu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)* dan *False Negative (FN)*. Nilai *True Negative (TN)* merupakan jumlah data negatif yang terdeteksi dengan benar oleh sistem, sedangkan *False Positive (FP)* merupakan data positif namun terdeteksi sebagai data negatif/terklasifikasi salah oleh sistem. Sementara itu, *True Positive (TP)* merupakan jumlah data positif yang terklasifikasi dengan benar oleh sistem, *False Negative (FN)* merupakan jumlah data negatif namun terklasifikasi sebagai data positif/terdeteksi salah oleh sistem. Berdasarkan nilai *True Negative (TN)*, *False Positive (FP)*, *False Negative (FN)* dan *True Positive* dapat diperoleh nilai akurasi, presisi dan recall. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasi data dengan benar, nilai presisi menggambarkan jumlah data kategori positif yang diklasifikasi dengan benar dibagi dengan total data yang diklasifikasi positif. Sementara recall menunjukkan berapa persen data kategori positif yang terklasifikasi dengan benar oleh sistem. *Confusion Matrix* ditunjukkan dengan Gambar 2.5.

		<i>Predicted</i>	
		<i>Positive (1)</i>	<i>Negative (0)</i>
<i>Actual</i>	<i>Positive (1)</i>	<i>TP</i>	<i>FP</i>
	<i>Negative (0)</i>	<i>FN</i>	<i>TN</i>

Gambar 2.5 *Confusion Matrix*

Sedangkan persamaan untuk mencari nilai akurasi, presisi, dan *recall* sebagai berikut:

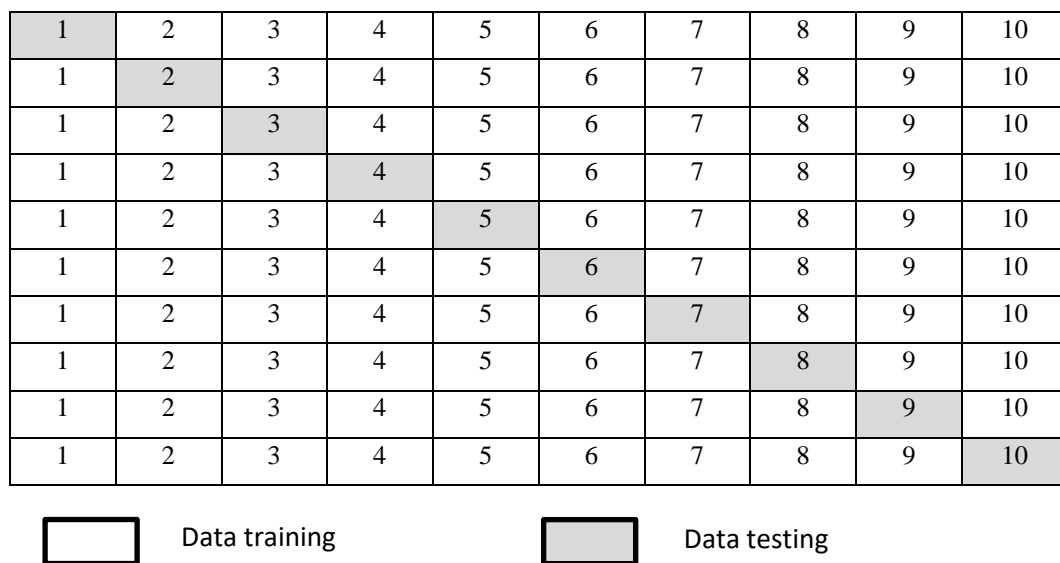
$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (2.5)$$

$$Presisi = \frac{TP}{TP+FP} * 100\% \quad (2.6)$$

$$Recall = \frac{TP}{TP+FN} * 100\% \quad (2.7)$$

2. *K-Fold Cross Validation*

Dalam proses kerja algoritma klasifikasi perlu dilakukan evaluasi model dan algoritma yang digunakan. Pada penelitian ini digunakan metode *K-Fold Cross Validation* untuk mengevaluasi suatu model algoritma C4.5. Dalam mengevaluasi kinerja model, *K-Fold Cross Validation* dipisah menjadi dua *dataset* yaitu satu bagian *dataset* uji/evaluasi dan satu bagian lagi menjadi *dataset* latihan. *K-Fold Cross Validation* merupakan salah satu pemilihan model terbaik karena cenderung memberikan estimasi akurasi lebih baik dan terukur. Karena pada penelitian ini menggunakan 10 *Fold Cross Validation* maka *dataset* yang telah dibagi menjadi dua bagian tadi kemudian dilakukan pelatihan sebanyak 10 kali dan hasil dari dari pelatihan tersebut dihitung nilai rata-rata akurasi dari masing-masing iterasi untuk mendapatkan nilai akurasi. Gambar 2.6 merupakan ilustrasi dari 10 *Fold Cross Validation*.



Gambar 2.6 Skema *K-Fold Cross Validation*

Cara kerja *K-Fold Cross Validation* adalah sebagai berikut:

1. Total *instance* dibagi menjadi 10 bagian.
2. Fold ke-1 adalah ketika bagian ke-1 menjadi data uji (*data testing*) dan sisanya menjadi data latih (*data training*). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut. Perhitungan akurasi tersebut dengan menggunakan persamaan akurasi *Confusion Matrix*.
3. Fold ke-2 adalah ketika bagian ke-2 menjadi data uji (*data testing*) dan sisanya menjadi data latih (*data testing*). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut.
4. Demikian seterusnya hingga mencapai fold ke-K. Hitung rata-rata akurasi dari K buah akurasi di atas. Rata-rata akurasi ini menjadi akurasi final.