

BAB 2

TINJAUAN PUSTAKA

Bab ini berisi penjelasan mengenai kajian pustaka dari beberapa referensi seperti buku, jurnal, dan paper yang diperlukan sebagai penunjang pada penelitian Tugas Akhir.

2.1 UKT

Uang Kuliah Tunggal atau biasa disingkat UKT merupakan besaran biaya kuliah secara tunggal dimana biaya ini ditanggung oleh setiap individu mahasiswa berdasarkan pada kemampuan ekonomi dari masing-masing mahasiswa (Permenristekdikti No. 39, 2016). Sistem ini mulai diterapkan pada mahasiswa baru program S1 di seluruh PTN di Indonesia pada tahun akademik 2013/2014. Sistem UKT meniadakan uang pangkal yang harus dibayarkan oleh setiap mahasiswa baru menjadi sebuah biaya tunggal. Tujuan dari penerapan sistem ini adalah untuk meringankan beban biaya pendidikan mahasiswa khususnya yang berasal dari ekonomi menengah ke bawah. Besaran UKT mahasiswa ditetapkan dengan sistem subsidi silang yaitu didasarkan pada kondisi ekonomi dan sosial dari orang tua/wali mahasiswa, sehingga besaran UKT setiap mahasiswa tidaklah sama. Perbedaan ini dipengaruhi oleh nilai-nilai dari variabel penentu besaran UKT seperti penghasilan orang tua/wali, rekening air, rekening listrik, PBB, PKB motor, PKB mobil, hingga jumlah tanggungan anak (Larytasari dan Susanti, 2019).

2.2 *Artificial Intelligence*

Menurut H. A. Simon (1987), Kecerdasan buatan (*Artificial Intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu yang dalam pandangan manusia adalah cerdas. Menurut Rich dan Knight (1991), Kecerdasan buatan (AI) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang saat ini hanya dapat dilakukan oleh manusia. Kecerdasan buatan merupakan bagian dari ilmu komputasi yang mempelajari bagaimana membuat mesin

(komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Sehingga dapat ditarik kesimpulan bahwa *Artificial Intelligence* atau kecerdasan buatan merupakan studi yang mempelajari bagaimana membuat mesin komputer melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan oleh manusia (Marcania, 2019).

2.3 *Machine Learning*

Secara definisi, *machine learning* atau pembelajaran mesin merupakan ilmu atau studi yang menerapkan algoritma pada sistem komputer untuk dapat menyelesaikan *task* tertentu tanpa instruksi secara eksplisit. *Machine learning* merupakan metode analisis yang dapat menangani data dalam jumlah besar dengan cara mengembangkan algoritma komputer. Algoritma *machine learning* memiliki kemampuan otomatis dalam melakukan perhitungan yang kompleks pada data yang besar. Dengan menggunakan data, mesin pembelajaran memungkinkan komputer untuk menemukan pola tersembunyi tanpa dilakukan program secara eksplisit saat mencarinya. Untuk mendapatkan pola dan kesimpulan dari data yang besar, algoritma *machine learning* menghasilkan model matematika yang didasari dari data sampel atau biasa disebut dengan *data training* (Marcania, 2019).

2.4 *Data Mining*

Data mining merupakan suatu disiplin ilmu yang bertujuan untuk menggali, menambang atau menemukan informasi yang selama ini tidak diketahui secara langsung dari suatu data yang besar. *Data mining* merupakan gabungan dari teknik statistik, matematika, kecerdasan buatan dan *machine learning* dalam mengekstraksi informasi yang tersembunyi dari *database* yang besar. *Data mining* juga dapat diartikan sebagai proses eksplorasi pola atau karakteristik dari suatu data yang tidak diperkirakan sebelumnya untuk membantu dalam pengambilan keputusan di waktu mendatang. Pola ini akan dikenali dan dipelajari dengan teliti oleh perangkat yang diharapkan nantinya dapat memberikan analisa yang berguna. *Data mining* merupakan proses pembelajaran komputer (*machine learning*) dengan menggunakan satu atau lebih teknik pembelajaran (algoritma), sehingga diharapkan

komputer mampu menganalisa pengetahuan secara otomatis. Berdasarkan tugas yang dapat dilakukan, *data mining* dibagi menjadi beberapa kelompok, yaitu: deskripsi, estimasi, prediksi, klasifikasi, pengklusteran dan asosiasi (Hutabarat, 2018; Kastawan, Wiharta dan Sudarma, 2018; Manik, Pristiwanto dan Tampubolon, 2018; Marcania, 2019; Pardede, Buulolo dan Ndruru, 2019).

Data mining menjadi salah satu tahapan dari proses *knowledge discovery in database* (KDD). KDD merupakan keseluruhan proses dalam mengidentifikasi pola (*pattern*) dari suatu data, yang mana pola tersebut bersifat baru dan dapat bermanfaat. Proses KDD secara garis besar dapat dijelaskan sebagai berikut (Hutabarat, 2018; Wijaya, Hasibuan dan Ramadhani, 2018; Marcania, 2019).

1. *Data Selection*

Langkah pertama dari rangkaian proses KDD yaitu melakukan *data selection* atau seleksi data. Data yang telah diperoleh akan dilakukan pemilihan atribut untuk digunakan pada proses selanjutnya. Seleksi data perlu dilakukan sebelum memasuki tahap penggalian informasi. Data yang telah diseleksi baru akan digunakan pada proses *data mining*.

2. *Data Cleaning*

Setelah dilakukan seleksi data, langkah selanjutnya adalah proses pembersihan (*cleansing*) data, yang tahapannya mencakup membuang duplikasi data, memperbaiki kesalahan (*error*) pada data seperti kesalahan penulisan, sampai mengecek data yang hilang (*missing*).

3. *Data Transformation*

Merupakan proses untuk merubah bentuk data menjadi bentuk yang sesuai untuk di-*mining*. Beberapa teknik *data mining* membutuhkan format data yang khusus sebelum diproses dalam *data mining*. Tahap ini *data training* akan diubah formatnya menjadi format yang dapat dipakai dalam *data mining* (Nofriansyah, 2014).

4. *Data Mining*

Data mining merupakan proses mencari suatu informasi atau pola tersembunyi dari suatu data dengan menggunakan teknik atau metode tertentu.

Pemilihan metode atau algoritma sangatlah penting karena sangat berpengaruh pada tujuan yang ingin dicapai.

5. *Interpretation (Evaluation)*

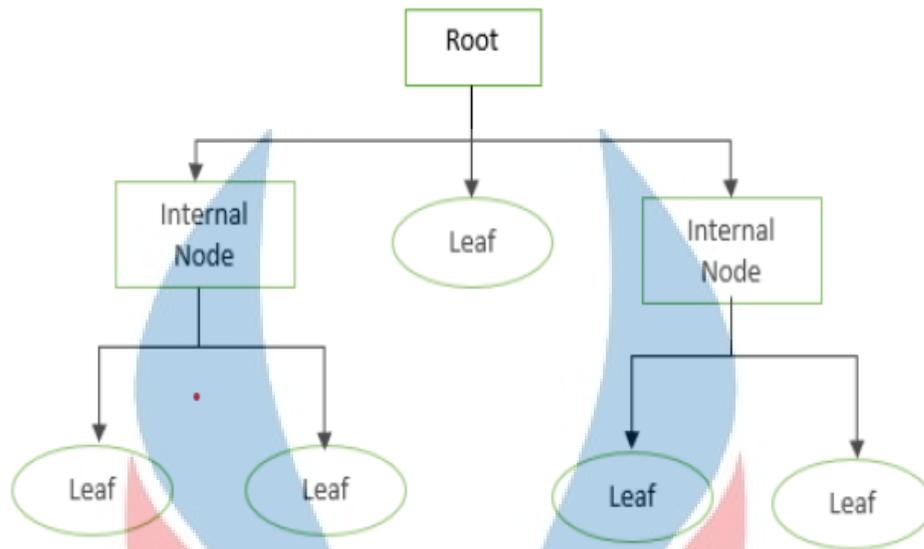
Pada tahap ini, pola yang dihasilkan perlu ditampilkan dalam bentuk yang mudah dimengerti. Selain itu, tahapan ini juga mencakup pemeriksaan apakah pola atau informasi yang dihasilkan bertentangan dengan fakta yang ada sebelumnya atau tidak.

Terdapat beberapa teknik dalam *data mining*, salah satunya adalah teknik klasifikasi. Dalam klasifikasi, variabel targetnya berbentuk kategori. Adapun salah satu teknik klasifikasi yang cukup umum yaitu pohon keputusan (*decision tree*) (Kastawan, Wiharta dan Sudarma, 2018).

2.5 *Decision Tree*

Menurut Kusriani dan Luthfi (2017), *Decision tree* atau pohon keputusan merupakan metode klasifikasi yang sangat kuat dan populer. *Decision tree* dapat mengubah data yang sangat besar menjadi sebuah pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami, karenanya *decision tree* merupakan teknik klasifikasi yang mudah untuk dipelajari dan sangat populer digunakan. *Decision tree* merupakan teknik yang digunakan untuk melakukan klasifikasi terhadap sekumpulan objek atau data dengan representasi struktur pohon. Pohon keputusan umumnya digunakan untuk eksplorasi pola pada sekumpulan data serta menemukan hubungan antara sejumlah variabel *input* dengan variabel *output*. Adapun bagan *decision tree* terdiri dari tiga bagian yaitu: (Hadi, 2017; Itiqomah, Susanti dan Zukhronah, 2019; Marcania, 2019; Manurung, 2020).

1. *Root Node* : Merupakan *node* yang terletak paling atas dari pohon keputusan.
2. *Internal Node* : Merupakan *node* percabangan, dimana hanya terdapat satu *input* dan menghasilkan minimal dua *output*.
3. *Leaf Node* : Merupakan *node* akhir, hanya mempunyai satu *input* dan tidak memiliki *output*.



Gambar 2. 1 Struktur Pohon Keputusan
*(Riadi, Azhar dan Wicaksono, 2020)

Layaknya pohon, *decision tree* dihubungkan oleh cabang, bergerak dari atas ke bawah, dimulai dari *root node* hingga berakhir di *leaf node*. *Root node* ditempatkan di bagian atas *decision tree*, kemudian seluruh atribut dievaluasi di bagian *node*, dengan kemungkinan menghasilkan cabang. Tiap cabang dapat masuk ke *node* lainnya ataupun menuju *leaf node*. Setiap *node* merepresentasikan atribut, cabangnya merepresentasikan nilai dari atribut, dan *leaf node* merepresentasikan kelas. Pada kasus ini, *leaf node* terbentuk berdasarkan mayoritas kelas dari masing-masing kategori atribut. Adapun syarat yang harus dipenuhi dalam penerapan algoritma *decision tree* adalah sebagai berikut (Hadi, 2017; Hutabarat, 2018).

1. Algoritma *decision tree* merepresentasikan *supervised learning* yang artinya diperlukan *training data set* yang menyediakan variabel target *preclassified*.
2. *Training dataset* harus kaya dan bervariasi.
3. Variabel target harus diskrit.

Pohon keputusan dapat dengan mudah melihat hubungan antar faktor-faktor yang mempengaruhi suatu masalah. Algoritma ini dapat melakukan prediksi hanya dengan menggunakan struktur pohon, mengubah data menjadi pohon keputusan serta menjadi beberapa aturan (*rules*) keputusan (Kastawan, Wiharta dan Sudarma, 2018).

2.6 Algoritma C4.5

Algoritma C4.5 dipopulerkan oleh Quinlan pada tahun 1996 sebagai penyempurnaan dari algoritma ID3. Algoritma ID3 hanya bisa dilakukan pada fitur bertipe kategorik dengan skala (nominal atau ordinal), sedangkan tipe numerik dengan skala (interval atau rasio) tidak dapat digunakan. Penyempurnaan ini terdapat pada kemampuan algoritma C4.5 dalam menangani fitur bertipe numerik, melakukan pemangkasan pohon (*pruning tree*) dan penurunan (*deriving*) rule set. Algoritma ini menggunakan kriteria *Gain* dalam menentukan fitur yang menjadi pemecah node pada *decision tree* (Prasetyo, 2013).

Hal penting dalam pembentukan *decision tree* adalah menyatakan syarat pengujian pada node. Ada tiga kelompok penting dalam syarat pengujian node (Prasetyo, 2013) :

1. Fitur biner

Fitur yang hanya memiliki dua nilai berbeda disebut fitur biner. Syarat pengujian ketika fitur ini menjadi node (akar maupun internal) hanya memiliki dua pilihan cabang.

2. Fitur bertipe kategorikal

Untuk fitur yang bertipe kategorik (nominal atau ordinal) memungkinkan memiliki beberapa nilai yang berbeda dan ini bisa memiliki banyak kombinasi dalam syarat pengujian pemecahan. Secara umum terdapat dua jenis pemecahan, yaitu pemecahan biner (*biner splitting*) dan (*multi splitting*). Seperti pada algoritma CART, maka akan memberikan kemungkinan jumlah kombinasi pemecahan sebanyak 2^{k-1} , di mana k adalah jumlah nilai berbeda dalam fitur.

3. Fitur bertipe numerik

Pada fitur bertipe numerik, syarat pengujian dalam node (akar ataupun internal) dinyatakan dengan pengujian perbandingan ($A > v$) atau ($A \leq v$) dengan hasil biner. Dalam kasus pemecahan biner, algoritma akan memeriksa semua kemungkinan posisi pemecahan v dan memilih posisi v terbaik yaitu yang memiliki kriteria *gain* paling besar. Pada kasus pemecahan multi, dengan hasil berupa nilai interval dalam bentuk $v_i \leq A < v_{i+1}$ untuk $i = 1, 2, \dots, k$, maka algoritma akan memeriksa seluruh kemungkinan nilai interval dan memilih nilai interval terbaik.

2.7 Algoritma C5.0

Algoritma C5.0 merupakan salah satu metode *data mining* dengan algoritma klasifikasi yang berbasis pada teknik *decision tree*. Algoritma ini merupakan hasil penyempurnaan dari algoritma ID3 dan C4.5 yang dibentuk oleh Ross Quinlan pada tahun 1987. Algoritma C5.0 lebih baik dibanding algoritma C4.5 dalam hal akurasi dan memori. Kecepatan dalam membuat model dinilai sangat cepat dibanding algoritma lainnya. Algoritma ini menerapkan *rule based model* atau model berbasis aturan sehingga memudahkan dalam memahami aturan dari pohon keputusan. Kelebihan inilah yang membuat algoritma C5.0 dinilai unggul dibanding algoritma lainnya. Algoritma ini dapat menangani atribut yang bernilai diskrit maupun kontinu. Algoritma ini memproses atribut menggunakan *Information Gain* dan *Entropy*. Atribut dengan *Gain* tertinggi akan dipilih sebagai *root node* dan *parent node* selanjutnya. Algoritma ini bermula dari semua atribut yang dijadikan akar dari pohon keputusan. Kemudian dipilih atribut yang memiliki nilai *Gain* tertinggi untuk dijadikan *root node*. Selanjutnya atribut lainnya akan dievaluasi dengan cara yang serupa untuk mendapatkan akar node selanjutnya. *Information Gain* merupakan nilai informasi untuk mengukur tingkat keberagaman (heterogenitas) dari sekumpulan data. Adapun persamaan untuk menghitung *Information Gain* adalah sebagai berikut (Hutabarat, 2018; Manik, Pristiwanto dan Tampubolon, 2018; Itiqomah, Susanti dan Zukhronah, 2019; Pardede, Buulolo dan Ndruru, 2019).

$$I(S_1, S_2, \dots, S_m) = - \sum_{i=1}^m P_i \log_2(P_i) \quad (2.1)$$

Dengan :

$I(S_1, S_2, \dots, S_m)$ = Informasi dari keseluruhan kasus pada kelas i

S = Jumlah kasus

S_i = Jumlah kasus pada kelas i

m = Banyaknya kelas

P_i = Proporsi kelas i , $P_i = S_i/S$

Langkah selanjutnya yaitu menghitung *Information Gain* dari keseluruhan kasus pada kelas i dan kategori j sebagai berikut.

$$I(S_{1j}, S_{2j}, \dots, S_{mj}) = - \sum_{i,j=1}^{my} P_{ij} \log_2(P_{ij}) \quad (2.2)$$

$I(S_{1j}, S_{2j}, \dots, S_{mj})$ merupakan informasi dari keseluruhan kasus pada kelas i dan kategori j . P_{ij} merupakan proporsi kelas i pada kategori j dari S_{ij} terhadap S , $P_{ij} = S_{ij}/S$. S_{ij} menunjukkan jumlah kasus pada kelas i dan kategori j . serta y menunjukkan banyaknya kategori. $\log_2(x)$ umum digunakan dalam konteks teknik komputer, kebanyakan bahasa komputer mengandung logaritma natural berbasis dua. Sedangkan untuk menghitung nilai *Entropy* ditunjukkan pada persamaan berikut.

$$E(A) = \sum_{i,j=1}^{my} \frac{S_{1j} + \dots + S_{mj}}{S} \times I(S_{1j}, S_{2j}, \dots, S_{mj}) \quad (2.3)$$

Entropy digunakan untuk mengukur kecenderungan suatu kelas dari sekumpulan data. $E(A)$ merupakan *Entropy* atribut A pada kategori j , S_{ij} merupakan jumlah kasus kelas i dan kategori j dari atribut A , dan $\frac{S_{1j} + \dots + S_{mj}}{S}$ merupakan informasi dari jumlah kasus kelas i dan kategori j terhadap S . Untuk mendapatkan nilai *Gain* selanjutnya dihitung dengan formula sebagai berikut.

$$Gain(A) = I(S_1, S_2, \dots, S_m) - E(A) \quad (2.4)$$

Gain dapat diartikan sebagai ukuran efektifitas atau seberapa informatif suatu atribut dalam melakukan klasifikasi data. $E(A)$ merupakan total *entropy* dari keseluruhan kategori pada atribut A . Proses dilakukan hingga kategori sampel tidak dapat dilakukan *split* atau pemisahan. Dalam klasifikasi kelas biner, kriteria *Information Gain*, *Entropy* dan *Gain* memiliki *range* nilai dari nol sampai satu. Pada kasus dengan kelas lebih dari dua, *range* nilai dapat lebih besar dari satu, yang artinya sama bahwa tingkat keberagaman sangat tinggi.

2.8 Prediksi

Prediksi merupakan proses memperkirakan sesuatu yang paling mungkin terjadi di masa mendatang berdasarkan informasi masa lalu, agar kesalahannya dapat diperkecil. Prediksi tidak harus memberikan jawaban secara tepat, melainkan

berusaha untuk mencari jawaban sedekat mungkin dengan apa yang akan terjadi. Prediksi dilakukan berdasarkan hasil observasi atau data yang telah dikumpulkan (Hutabarat, 2018).

2.9 Confusion Matrix

Confusion matrix merupakan metode evaluasi yang menggunakan tabel matriks. Matriks ini terdiri dari beberapa sel, setiap sel berisi angka yang menyatakan jumlah data uji yang diklasifikasi dengan benar dan jumlah data uji yang salah diklasifikasikan. Pada klasifikasi biner, matriks ini berdimensi 2x2. Adapun tabel *confusion matrix* sebagai berikut (Kastawan, Wiharta dan Sudarma, 2018; Itiqomah, Susanti dan Zukhronah, 2019).

Tabel 2.1 Confusion Matrix

Actual	Prediction	
	Class +	Class -
Class +	TP	FN
Class -	FP	TN

*(Itiqomah, Susanti dan Zukhronah, 2019)

Dimana:

TP : *True Positive* yaitu jumlah data aktual berlabel positif yang terklasifikasi dengan benar oleh sistem prediksi.

TN : *True Negative* yaitu jumlah data aktual berlabel negatif yang terklasifikasi dengan benar oleh sistem prediksi.

FN : *False Negative* yaitu jumlah data aktual berlabel positif tapi diklasifikasi model sebagai negatif.

FP : *False Positive* yaitu jumlah data aktual berlabel negatif tapi diklasifikasi model sebagai positif.

Confusion matrix umum digunakan untuk mengukur akurasi pada konsep *data mining*. Akurasi menyatakan jumlah data yang diprediksi dengan benar. Metrik ini digunakan untuk mengetahui seberapa baik model dalam melakukan prediksi. Adapun rumus yang digunakan untuk menghitung akurasi adalah sebagai berikut (Kastawan, Wiharta dan Sudarma, 2018).

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

Sedangkan *error* menyatakan jumlah data yang salah diprediksi, atau dengan kata lain untuk mengetahui seberapa buruk model dalam melakukan prediksi. Adapun rumus yang digunakan untuk menghitung *error* adalah sebagai berikut (Kastawan, Wiharta dan Sudarma, 2018).

$$Error (\%) = \frac{FN + FP}{TP + TN + FP + FN} \times 100\% \quad (2.6)$$

Akurasi dan *error* tidak cukup dijadikan penentu seberapa baik model melakukan prediksi, banyak hal yang perlu diperhatikan. Selain akurasi, metrik pengukur lainnya yaitu *precision*, sensitivitas (*recall*), dan *specifity*.

Presisi menyatakan tingkat ketepatan antara data sebenarnya dengan jawaban yang diberikan oleh sistem. Metrik ini digunakan untuk melihat seberapa handal model memberikan prediksi positif sesuai dengan label aktual. Presisi adalah metrik yang tepat untuk meminimalkan kejadian FP (model gagal mendeteksi data yang sebenarnya negatif). Adapun rumus yang digunakan untuk menghitung presisi adalah sebagai berikut.

$$Precision (\%) = \frac{TP}{TP + FP} \times 100\% \quad (2.7)$$

Sedangkan sensitivitas atau *recall* menyatakan tingkat keberhasilan model dalam mendeteksi data berlabel positif dengan benar. Metrik ini digunakan untuk melihat seberapa sensitif model dalam mendeteksi data berlabel positif. Sensitivitas adalah metrik yang tepat untuk meminimalkan kejadian FN (model gagal mendeteksi data yang sebenarnya positif). Adapun rumus yang digunakan untuk menghitung sensitivitas adalah sebagai berikut.

$$Recall (\%) = \frac{TP}{TP + FN} \times 100\% \quad (2.8)$$

Specifity menyatakan derajat kehandalan model dalam mendeteksi data berlabel negatif dengan benar. *Specifity* merupakan sensitivitas dalam konteks

negatif. Adapun rumus yang digunakan untuk menghitung *specifity* adalah sebagai berikut.

$$Specifity (\%) = \frac{TN}{TN + FP} \times 100\% \quad (2.9)$$

2.10 Penelitian Terdahulu

Berikut adalah rangkuman hasil penelitian terdahulu yang memiliki keterkaitan dengan penelitian ini.

Tabel 2. 2 Rangkuman Penelitian Terdahulu

No	Nama dan Tahun Publikasi	Hasil
1	Manik dkk, 2018	Judul : Prediksi Kolektibilitas Kredit Anggota Dengan Algoritma C5.0 (Studi Kasus : CU Sejahtera Medan). Metode : Algoritma C5.0 berbasis pohon keputusan. Hasil : Kolektibilitas kredit menggunakan 6 atribut pendukung seperti : besar pinjaman, lama pinjaman, simpanan, agunan, pendapatan keluarga, dan pengeluaran keluarga. Berdasarkan perhitungan <i>gain</i> , besar pinjaman dijadikan <i>root node</i> . Pohon keputusan membantu membuat keputusan dan melihat hubungan antar atribut penentu. Algoritma ini dapat dijadikan acuan dalam pemecahan suatu masalah.
2	Kastawan dkk, 2018	Judul : Implementasi Algoritma C5.0 Pada Penilaian Kinerja Pegawai Negeri Sipil. Metode : Algoitma C5.0 berbasis <i>Decision Tree, Pruning, Confusion Matrix</i> . Hasil : Berdasarkan hasil evaluasi pada data <i>training</i> sebanyak 184 data, didapatkan akurasi sebesar 96,08%. Algoritma C5.0 mampu memproses data kinerja pegawai menjadi pohon keputusan dan aturan-aturan. Hasil dapat dikembangkan menjadi sistem penentu keputusan, sehingga dapat membantu menentukan keputusan kinerja pegawai.
3	Marcania, M. 2019	Judul : Prediksi Pengangkatan Karyawan Dengan Metode Klasifikasi Algoritma C5.0 (Studi Kasus : PT. Kiyokuni Indonesia Factory-2). Metode : Algoritma C5.0

		<p>Hasil : Berdasarkan hasil pengujian dalam memprediksi pengangkatan karyawan pada PT. Kiyokuni Indonesia Factory-2, diperoleh akurasi sebesar 96.75%. Maka disimpulkan bahwa teknik algoritma C5.0 dinilai berhasil dalam memprediksi pengangkatan karyawan PT. Kiyokuni Indonesia Factory-2.</p>
4	Nastiti dkk, 2019	<p>Judul : Penerapan Algoritma C5.0 Pada Analisis Faktor-Faktor Pengaruh Kelulusan Tepat Waktu Mahasiswa Teknik Informatika Universitas Muhammadiyah Malang. Metode : Algoritma C5.0</p> <p>Hasil : Penelitian ini menggunakan algoritma C5.0 untuk melakukan seleksi fitur penting. Hasil implementasi algoritma C5.0 pada penelitian ini mampu melakukan seleksi fitur dengan menghasilkan 8 dari total keseluruhan 15 fitur dengan nilai akurasi yang lebih baik dibandingkan nilai akurasi yang menggunakan keseluruhan fitur.</p>
5	Wisri dkk, 2019	<p>Judul : Analisa Kesejahteraan Rumah Tangga Menggunakan Metode C5.0. Metode : Algoritma C5.0</p> <p>Hasil : Penelitian ini dilakukan untuk menentukan pola status rumah tangga dengan mengklasifikasi berdasarkan status kesejahteraan, jenis kelamin, pendidikan terakhir, lapangan usaha dan jumlah anggota keluarga.</p>

