BAB II
TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 *Data Mining*

Data mining adalah sebuah ilmu komputer dalam proses pencarian dan analisis dari kumpulan data yang besar untuk menentukan pola tertentu yang memiliki makna serta aturan (Barhate dkk., 2018). *Data mining* memiliki tujuan untuk merancang pola tertentu dari sejumlah kumpulan data yang besar (*database*) dengan secara efisien untuk menghasilkan suatu informasi atau pengetahuan sehingga dikenal dengan istilah *knowledge discovery*, sedangkan untuk pengenalan pola yang tepat pada *data mining* yang digunakan untuk menemukan pola tersembunyi dalam kumpulan data dikenal dengan istilah *pattern recognition* (Nabila dkk, .2021).

Proses *data mining* yaitu dengan mengekstraksi pengetahuan dari sejumlah data yang besar, *data mining* sebagai proses mengeksplorasi dan menganalisis kumpulan data untuk menemukan pola dan aturan untuk memecahkan suatu masalah, serta dalam penggunaan *data mining* didorong dari kebutuhan akan teknik mengambil keputusan dengan cara menganalisis, memahami, dan melihat data yang dikumpulkan dari *data warehouse* (PASCU, 2018). Proses pengerjaan pada *data mining* memiliki metodologi yang dapat digunakan, metodologi dapat dengan mudah digunakan oleh seorang pemula dalam bidang *data mining* karena setiap fase pengerjaan terstruktur, terdokumentasi dengan baik serta terdefinisi dengan jelas. Metodologi penelitian ini terdiri dari beberapa fase utama, yaitu fase

pemahaman bisnis, fase pemahaman data, fase pengolahan data, fase pemodelan, fase evaluasi, serta fase penyebaran hasil. (Fadillah, 2015).

Data mining memiliki beberapa tahapan dalam pengerjaannya yaitu:

1. Pembersihan Data (*Data Cleaning)*

   Pembersihan data merupakan proses menghilangkan noise dan data yang tidak relevan. Pada umumnya data yang didapatkan memiliki isi yang tidak sempurna, seperti data yang hilang, data yang tidak *valid*, atau data yang salah dalam pengetikan atau *human error,* sehingga baik untuk dihilangkan karena akan mempengaruhi hasil ataupun proses dari teknik *data mining*.

2. Integrasi Data (*Data Integration*)

   Proses integrasi data merupakan penggabungan data dari berbagai *database* kedalam satu *database* yang baru. Proses integrasi harus dikerjakan dengan teliti karena jika terjadi kesalahan saat proses integrasi data dapat menghasilkan hasil yang salah dan akan mempengaruhi saat pengambilan keputusan.

3. Seleksi Data (*Data Selection*)

   Data yang terdapat pada *database* untuk dilakukan analisis tidak selalu semuanya dipakai, akan tetapi hanya data yang sesuai yang digunakan oleh karena itu dilakukan proses seleksi data.

4. Transformasi Data (*Data Transformation*)

   Transformasi data merupakan perubahan data baik itu data yang digabungkan maupun data yang diubah sesuai dengan format untuk diproses.

5. Proses *Mining*

   Proses *mining* merupakan proses utama saat metode diterapkan untuk menemukan pengetahuan baru yang tersembunyi dari suatu data.

6. Evaluasi Pola (*Pattern evaluation*)

   Proses evaluasi pola digunakan untuk mengidentifikasi pola-pola menarik yang ditemukan dalam *knowledge based*, sehingga tahap ini akan menghasilkan pola-pola yang baru dan unik maupun model prediksi dievaluasi untuk menilai tercapainya hipotesa yang ada.

2.1.2 Klasifikasi

Klasifikasi merupakan salah satu teknik pada *data mining* yang berfungsi untuk mengelompokan data ke dalam kelas tertentu dari sejumlah kelas yang ada berdasarkan atribut-atribut tertentu (Leomongga Oktaria Sihombing, Hannie, 2021). Dasar yang diturunkan dari model klasifikasi yaitu pada analisis dari *training data*. Menurut (Ulfatul *et al.*, 2022) proses klasifikasi terbagi menjadi dua fase yatu *fase learning* dan *testing. Fase training* merupakan fase untuk membangun sebuah model dari data yang akan digunakan, sedangkan *fase testing* merupakan pengujian dari model yang telah dibuat dengan data lainnya untuk mengetahui akurasi dari model tersebut.

Klasifikasi memiliki 3 tahapan (Nasution, Khotimah and Chamidah, 2019).

1. Pembangunan Model

   Pembangunan model merupakan data latih yang telah memiliki atribut dan kelas digunakan untuk membangun sebuah model.

2. Penerapan Model

   Penerapan model merupakan penerapan model yang telah dibangun untuk menentukan kelas dari data atau objek baru.

3. Evaluasi

   Evaluasi merupakan proses yang dilakukan untuk melihat akurasi terhadap data baru dari pembangunan dan penerapan model yang telah dilakukan pada tahap sebelumnya.

2.1.3 *Adaptive boosting (Adaboost)*

Algoritma *boosting* adalah konsep dari *machine learning* yang dapat digunakan untuk mengkombinasikan algoritma klasifikasi lain untuk meningkatkan performa klasifikasi. *Adaptive boosting* adalah salah satu algoritma *boosting* yang mampu menyelesaikan nilai *error* secara adaptif yang dihasilkan dari klasifikasi lemah untuk dijadikan acuan proses pelatihan klasifikasi berikutnya (Latief, Subekti and Gata, 2021). Algoritma *adaboos*t termasuk algoritma yang populer dan cukup banyak digunakan, karena *adaboost* ini mudah digunakan dan diimplementasikan juga fleksibel sehingga mudah dikombinasikan dengan algoritma lain. *Adaboost* banyak berhasil digunakan pada beberapa bidang karena memiliki teori yang kuat, prediksi yang besar, juga sederhana (Prasetio and Susanti, 2019). *Weak learner* yang digunakan pada algoritma ini yaitu algoritma *Decision Tree*.

Algoritma Decision Tree merupakan suatu model berbasis diagram alur yang memiliki struktur menyerupai pohon, setiap *internal node* merepresentasikan proses pengujian terhadap suatu atribut, setiap cabang (*branch*) menunjukkan hasil

dari pengujian tersebut, dan *leaf node* merepresentasikan kategori kelas atau distribusi kelas yang dihasilkan. Node yang terletak di bagian paling atas disebut sebagai *root node* atau simpul akar, yang memiliki sejumlah *edge* keluar tanpa *edge* masuk. Sementara itu, *internal node* memiliki satu *edge* masuk dan beberapa *edge* keluar, sedangkan *leaf node* hanya memiliki satu *edge* masuk tanpa *edge* keluar. Decision Tree umumnya digunakan dalam proses klasifikasi untuk menentukan kelas dari suatu sampel data yang belum diketahui kategorinya, berdasarkan kelas-kelas yang telah ditetapkan sebelumnya (Septhya *et al.*, 2023).

2.1.4 *K-Nearest Neighbor (KNN)*

Algoritma *K-Nearest Neighbor (KNN)* merupakan algoritma tertua dan paling sederhana dalam penerapannya dan efektif, (Gamadarenda and Waspada, 2020), serta akurat untuk melakukan klasifikasi pola dan regresi. Algoritma ini sederhana namun memiliki kinerja yang dapat bersaing dengan klasifikasi yang kompleks (Suwirmayanti, 2017), algoritma ini memiliki kemampuan dalam mendeteksi dan menganalisa permasalahan yang cukup kompleks dan *non-linier*, serta perhitungan dilakukan secara paralel sehingga waktu komputasi akan lebih cepat (Abu Alfeilat *et al.*, 2019).

Menurut (Arifin, 2019) Algoritma *K-Nearest Neighbor (KNN))* merupakan pengklasifikasian objek berdasarkan data pembelajaran terdekat berdasarkan dengan data sebelumnya yang dimiliki sebagai sampel untuk menemukan suatu hasil akhir. Algoritma *K-NN* bersifat nonparametrik yang berarti tidak memiliki jumlah parameter, namun parameter akan ditentukan oleh ukuran kumpulan data

pelatihan, meskipun tidak ada asumsi yang perlu dibuat untuk distribusi data yang mendasarinya (Setiyorini and Asmono, 2019).

Terdapat beberapa langkah-langkah dalam melakukan perhitungan *KNN (K-Nearest Neighbor)* (Arifin, 2019), berikut adalah langkah-langkah tersebut :

a. Menentukan parameter k atau jumlah tetangga paling dekat.
b. Menghitung jarak *euclidean* atau *query instance* pada masing-masing objek terhadap data sampel yang diberikan.
c. Mengurutkan objek-objek ke dalam kelompok
d. yang mempunyai jarak *euclidean* terkecil.
e. Mengumpulkan kategori Y atau klasifikasi *KNN (K-Nearest Neighbor)*.
f. Dengan menggunakan kategori *nearest neighbor* yang paling banyak maka dapat dilakukan prediksi nilai *query instance* yang telah dihitung.

Klasifikasi algoritma *K-NN* dilakukan berdasarkan *data training* dilihat dari jarak paling dekat dengan objek berdasarkan nilai k (Setianto, Kusrini and Henderi, 2019). Jarak terdekat dapat dilakukan dengan membagi data menjadi *data training* dan *data testing*, jika *data training* dan *data testing* telah dibuat maka bisa menggunakan metode perhitungan jarak untuk menghitung jarak masing-masing data *testing* terhadap data *training*. Terdapat beberapa metode perhitungan jarak pada algoritma *KNN (K-Nearest Neighbor)* yaitu :

1. *Euclidean Distance*

   Metode *euclidean* merupakan salah satu metode perhitungan jarak yang digunakan untuk mengitung jarak antara dua titik dalam ruang *euclidean*. Pada metode ini, semakin kecil jarak maka akan semakin kecil

jarak antar kedua titik. Metode ini merupakan metode yang paling sederhana dan umum digunakan (Yudhana, Sunardi and Hartanta, 2020), dan untuk mengkur tingkat kemiripan data dengan *euclidean* mengunakan rumus sebagai berikut:

$$d(x,y) = \sqrt{\sum_i^n = 1 \left(x^i_{training} - y^i_{testing}\right)^2} \text{……..….. (2.1)}$$

Keterangan :

d(x,y) = jarak

i = variabel data

n = dimensi data

$x^i_{training}$ = data *training*

$y^i_{testing}$ = data *testing*

2. *Chebyshev Distance*

Chebyshev disebut sebagai *maximum distance* karena menerapkan pencarian nilai maksimum dari selisih jarak titk data x dan y di demensi ruang d (Hidayati *et al.*, 2021). *Chebyshev* merupakan metode pengukuran jarak yang dihitung dari nilai *absolute* atau nilai mutlak dari selisih sepasang koordinat (Rani, Aziz and Sulistyono, 2022). Jarak yang diukur dengan *chebyshev* berdasarkan nilai mutlak dari perbedaan anta elemen-elemen pada vektor dan jumlah data yang secara otomatis harus sama. Rumus yang dapat digunakan untuk menghitung jarak dua buah objek berdasarkan *chebyshev* sebagai berikut:

$$d(x,y) = max_{i=1}^{n}|x_i - y_i| \text{ …….(2.2)}$$

3. *Manhattan Distance*

Manhattan (*city distance*) diartikan sebagai jumlah jarak dari semua atribut. *Manhattan* merupakan jarak geometri dari dua buah objek, digunakan untuk mengambil kasus yang cocok dari basis kasus dengan menghitung jumlah bobot *absolute* dari perbedaan antara kasus sekarang dan kasus lain dalam basis kasus. Untuk melakukan perhitungan jarak dapat menggunakan rumus berikut:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \ldots\ldots\ldots (2.3)$$

2.1.5 *Synthetic Minority Oversampling Technique* (*SMOTE*)

Metode *sytnthetic minority oversampling technique* atau sering dikenal dengan *SMOTE* merupakan model pengembangan dari metode *oversampling*, yang diterapkan dalam rangka menangani ketidakseimbangan kelas yang ada pada *dataset*. Masalah ketidakseimbangan data banyak sekali muncul pada proses klasifikasi, ketidakseimbangan dapat terjadi apabila jumlah objek suatu kelas data lebih banyak (*mayor*) dari kelas lainnya (*minor*). Perbedaan jumlah data yang besar akan berdampak pada model klasifikasi, sehingga model klasifikasi tidak dapat memprediksikan kelas minor dengan tepat sehingga banyak data tes yang seharusnya ada pada kelas minor diprediksikan salah oleh model klasifikasi (Prianggi, Nilogiri and Umilasari, 2021).

Cara kerja pada metode *SMOTE* ini yaitu dengan membuat data sintesis pada kelas data minor sehingga data menjadi seimbang (Arifiyanti and Wahyuni, 2020). Beberapa algoritma yang dikombinasikan dengan metode *SMOTE* terbukti

mengatasi ketidakseimbangan data dan mendapatkan hasil klasifikasi yang lebih baik (Sutoyo and Fadlurrahman, 2020).

2.1.6 *Confusion Matrix*

Confusion matrix merupakan proses untuk melakukan perhitungan akurasi pada konsep *data mining* (Ernawati and Wati, 2018). Terdapat 4 kombinasi yang berbeda pada *confusion matrix* yaitu terdapat nilai prediksi dan aktual. Adapun tabel evaluasi model *confusion matrix* (Said *et al.*, 2022), sebagai berikut :

Tabel 2. 1 *Confusion Matrix*

Confusion Matrix		Nilai Aktual	
		Positif	**Negatif**
Nilai Prediksi	**Positif**	*True Positive*	*False Negative*
	Negatif	*False Positive*	*True Negative*

Keterangan :

1. *True Positive* (TP) : jumlah data bernilai positif yang diklasifikasikan sebagai positif.
2. *False Positive* (FP) : jumlah data bernilai negatif yang diklasifikasikan positif.
3. *False Negative* (FN) : jumlah data bernilai positif yang diklasifikasikan negatif,
4. *True Negative* (TN) : jumlah data bernilai negatif yang diklasifikasikan negatif.

2.1.7 *Recall*

Recall merupakan rasio antara jumlah dengan prediksi benar pada kelas positif atau *true positif* (TP) terhadap total data yang sebenarnya masuk dalam kelas positif, dengan menjumlahkan antara *true positif* dan *false negatif* (TP + FN). Nilai ini merepresentasikan sejauh mana model klasifikasi mampu mengidentifikasi dan menemukan informasi yang benar-benar relevan (Clara *et al.*, 2021)

2.2 Penelitian Terkait dan Kebaruan Penelitian

2.2.1 *State Of The Art* (SOTA)

Tabel 2. Membahas mengenai penelitian sebelumnya yang telah dilakukan sebagai perbandingan dengan fokus pada penelitian yang akan dilakukan yaitu pada peningkatan akurasi algoritma klasifikasi. Terdapat beberapa kesamaan serta perbedaan dari masing-masing penelitian yang dapat dilihat dari penggunaan metode dan algoritmanya.

Tabel 2. 2 *State Of The Art*

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
1.	Tessy Badriyah, Nur Sakinah, Iwan Syarif, Daisy Rahmania Syarif	2020	*Machine Learning Algorithm for Stroke Disease Classification*	Stroke adalah penyebab utama kematian dan obesitas nomor satu di banyak negara. Penelitian ini melakukan *preprocessing* untuk meningkatkan kualitas citra dan mengurangi *noise,* serta menerapkan algoritma *machine learning* untuk mengklasifikasikan pasien menjadi dua sub tipe penyakit stroke, yaitu stroke iskemik dan stroke pendarahan. Delapan algoritma digunakan dalam penelitian ini salah satunya yaitu *KNN (K-Nearest Neighbor).*	Sebelum dilakukan klasifikasi, dilakukan pengolahan citra dan ekstraksi ciri pada data citra. Dan setelah itu digunakan perbandingan 8 (delapan) algoritma untuk melakukan klasifikasi yaitu : *KNN (K-Nearest Neighbor)s, naive bayes, logistic regression, decision tree, random forest, multi-layer perceptron (mlp-nn), deep learning and support vector machine.* Dari hasil perbandingan ini algoritma *K-NN* memiliki nilai akurasi 96%.

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
2.	Iswanto, Tulus, Poltak Sihombing	2021	*Comparison of Distance Models on KNN (K-Nearest Neighbor) Algorithm in Stroke Disease Detection*	Stroke adalah penyakit kardiovaskular (CVD) yang disebabkan oleh kegagalan sel-sel otak untuk mendapatkan suplai oksigen sehingga menimbulkan risiko kerusakan iskemik dan mengakibatkan kematian. Mendeteksi stroke membutuhkan metode pembelajaran mesin. Dalam penelitian ini, menggunakan salah satu dari metode klasifikasi pembelajaran terbimbing yaitu *KNN (K-Nearest Neighbor) (K NN)*. Penelitian ini membandingkan model jarak *euclidean, minkowski, manhattan , chebyshev* untuk mendapatkan hasil yang optimal.	Berdasarkan hasil pengujian, model *chebyshev* memiliki tingkat akurasi tertinggi dibandingkan ketiga model jarak lainnya dengan nilai akurasi rata-rata 95,49%, akurasi tertinggi 96,03%, pada K = 10. Model jarak *euclidean* dan *minkowski* memiliki tingkat akurasi yang sama pada setiap nilai K dengan nilai akurasi rata-rata 95,45%, akurasi tertinggi 95,93% pada K = 10. Sedangkan *Manhattan* memiliki rata-rata terendah dibandingkan model jarak lainnya, yaitu 95,42% tetapi memiliki akurasi tertinggi 96,03% pada nilai K = 6. Dengan demikian, model jarak yang paling optimal yang digunakan untuk

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
					dataset penyakit stroke adalah model jarak *chebyshev* dengan nilai K optimal 10. Selanjutnya model perhitungan jarak *manhattan* dengan nilai K optimal th adalah 6
3.	Kartika Resiandi, Adiwijaya, Dody Qori Utama	2018	*Detection of Atrial Fibrillation Disease Based on Electrocardiogram Signal Classification Using RR Interval and KNN (K-Nearest Neighbor)*	Seseorang yang tidak pernah memiliki riwayat penyakit jantung bahkan berpeluang menderita AF. Risiko yang ditimbulkan oleh AF, yaitu kemungkinan stroke, gagal jantung, dan kematian. Bagi seseorang yang sudah memiliki gejala AF sebaiknya segera memeriksakan diri salah satunya dengan menggunakan alat elektrokardiogram (EKG). Karena dengan adanya deteksi dini dapat menurunkan jumlah persentase	Struktur sistem diperoleh dengan menggunakan metode klasifikasi *K-NN*. Penelitian ini menggunakan parameter K=1,3,5,7,9, dan 11 dengan train/test split untuk mendapatkan akurasi tertinggi. Kinerja algoritma *K-NN* diukur dengan menggunakan confusion matrix. Berdasarkan akurasi skema keseluruhan, hasil terbaik adalah k = 1 dengan akurasi rata-rata sebesar 91,75% dan tingkat akurasi,

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
				populasi AF, dan prognosis penyakit AF juga lebih baik. Ada tiga tahapan dalam penelitian ini; yaitu pra-pemrosesan sebagai proses penyeragaman dimensi data, ekstraksi ciri, dan klasifikasi *K-NN*.	sensitivitas, spesifisitas tertinggi sebesar 95,45%, 91,67%, dan 100% dengan data train/test split sebesar 60 :40 persen. Disimpulkan bahwa diagnosis AF jantung dengan metode *K-NN* sudah memadai untuk pemeriksaan medis.
4.	Agus Byna, Muhammad Basit	2020	Penerapan Metode *Adaboost* Untuk Mengoptimasi Prediksi Penyakit Stroke Dengan Algoritma *Naïve Bayes*	Stroke adalah penyakit paling mematikan nomor dua di dunia menurut WHO. Saat ini perkembangan Era Revolusi Industri 4.0 yang berkolaborasi di bidang teknologi dan ilmu kesehatan sehingga menjadi sesuatu yang bermanfaat dengan menggunakan *Machine Learning*. Banyak sekali manfaat yang digunakan dalam memprediksi beberapa penyakit	Hasil penelitian untuk nilai akurasi algoritma *Naïve Bayes* memiliki nilai 0.976 dengan *Split data* 80/20, sedangkan untuk nilai akurasi optimasi *adaboost* dengan *naïve bayes* senilai 0.981 *split data* 70/30 kedua model tersebut memiliki diagnosa *Excellent Classification*, dalam pengujian prediksi penyakit stroke dengan dengan 11 variabel

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
				yang dapat diantisipasi. Khususnya penyakit stroke dengan menggunakan algoritma *adaboost* yang mempunyai kelebihan bisa digabung dengan algoritma *naïve Bayes.* Penerapan metode ini menggunakan *split data* yaitu data *training* dan data *testing* dibuat porsi dalam melakukan pengujian	dengan jumlah data 28,500 untuk data *training* dan 572 untuk data *testing* (hasil dari kuisioner dan aplikasi di rumah sakit). Algoritma *adaboost* dapat meningkatkan dan mengoptimasi yang dapat digabung dengan *naïve bayes* sebagai algoritma estimator sehingga menghasilkan akurasi yang dapat meningkatkan hasil yang terbaik dengan memiliki margin *error* yang kecil.
5.	Nia Novianti, Muhammad Zarlis, Poltak Sihombing	2022	Penerapan Algoritma *Adaboost* Untuk Peningkatan Kinerja Klasifikasi *Data Mining* Pada	Data – data penyakit penderita diabetes terdahulu sudah tersimpan dan tersusun data sebuah gudang penyimpanan data atau yang biasa disebut dengan *dataset*. Maka dari itu perlu dilakukan proses untuk	Berdasarkan hasil penelitian dan pengujian yang telah dilakukan maka dapat ditarik kesimpulan bahwasannya algoritma *adaboost* dapat dipergunakan dengan baik untuk membantu kinerja dari pada

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
			Imbalance Dataset Diabetes	melakukan pengolahan data yang terdapat pada *dataset*. Tetapi penggunaan dari pada teknik *data mining* sendiri harus dibantu dengan menggunakan teknik yang terdapat pada *data mining* tersebut yaitu teknik klasifikasi. Pada penelitian ini, menggunakan algoritma *KNN (K-Nearest Neighbor)* dengan menerapkan algoritma *adaboost* untuk meningkatkan akurasi metode klasifikasi.	algoritma *KNN (K-Nearest Neighbor)* untuk proses klasifikasi pada *dataset* penyakit diabetes. Hal ini dapat dilihat pada hasil pengujian dengan nilai K = 7, 13, 19, 25 dan 31 terdapat peningkatan hasil akurasi yang didapatkan setelah menggunakan algoritma *adaboost.* Akurasi tertinggi terdapat pada nilai K=7, sebelum menerapkan algoritma adaboost memiliki akurasi 92,70%, sedangkan setelah menerapkan algoritma *adaboost* meningkat menjadi 95,40%.
6.	Aah Sumiah, Nita Mirantika	2020	Perbandingan Metode *KNN (K-Nearest Neighbor)* dan *Naive Bayes*	Membandingkan algoritma *KNN (K-Nearest Neighbor)* dan algoritma *naïve bayes* untuk rekomendasi penerimaan beasiswa pada Universitas Kuningan.	Hasil dari implementasi ini algoritma *KNN (K-Nearest Neighbor)* diperoleh akurasi 100%, sedangkan algoritma *naïve bayes* akurasinya 99,89%. Hal

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
			untuk Rekomendasi Penentuan Mahasiswa Penerima Beasiswa pada Universitas Kuningan	Algoritma ini digunakan karena kedua algoritma tersebut biasa digunakan dalam klasifikasi data. Dari penelitian ini mengimplementasikan menjadi sistem informasi yang menggunakan *visual basic.net* dan *sql server*.	ini menunjukan bahwa pada data penerimaan beasiswa, algoritma *KNN (K-Nearest Neighbor)* lebih baik karena memiliki nilai akurasi lebih tinggi dibandingkan dengan *naïve bayes*. Banyaknya data latih akan mempengaruhi keakuratan data.
7.	Dita Noviana, Yuliana Susanti, Irwan Susanto	2019	Analisis Rekomendasi Penerima Beasiswa Menggunakan Algoritma *KNN (K-Nearest Neighbor)* (*K-Nn*) Dan Algoritma C4.5	Beasiswa merupakan bantuan biaya pendidikan yang sangat membantu prestasi mahasiswa. Seiring dengan meningkatnya jumlah mahasiswa yang mengajukan beasiswa, maka dibutuhkan suatu metode klasifikasi yang dapat membantu menentukan siapa yang layak mendapatkan beasiswa PPA dari Universitas Sebelas Maret. Metode yang digunakan dalam	Dari penelitian ini, dalam merekomendasikan penerima beasiswa PPA dapat dilakukan menggunakan algoritma *K-NN* dan C.45 dengan akurasi masing-masing sebesar 90.7% dan 88.3%. Hasil penelitian menunjukkan bahwa variabel memiliki pengaruh adalah IPK, prestasi, penghasilan, jumlah orang tua bekerja, dan jumlah

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
				analisis ini adalah *KNN (K-Nearest Neighbor)* dan C4.5. Hasil klasifikasi digunakan sebagai keputusan dalam rekomendasi penerima beasiswa.	tanggungan. Variabel yang paling berpengaruh adalah variabel IPK. Dalam mengukur kedua fungsi algoritma tersebut menggunakan *confusion matrix* dengan hasil bahwa algoritma *K-NN* memiliki tingkat akurasi yang lebih tinggi dibandingkan algoritma C4.5.
8.	Riski Tri Prasetio, dan Sari Susanti	2019	Prediksi Harapan Hidup Pasien Kanker Paru Pasca Operasi Bedah Toraks Menggunakan *Boosted KNN (K-Nearest Neighbor)*	Operasi bedah toraks menjadi salah satu solusi utama untuk kanker paru-paru. Akan tetapi,terdapat banyak resiko dan komplikasi pasca operasi bedah toraks hingga berujung pada kematian. Maka prediksi ini dilakukan dengan menganalisis kondisi pasien sebelum dan sesudah operasi. *Adaptive boost* digunakan sebagai optimasi	Untuk melakukan prediksi ini yaitu dengan mengkombinasikan teknik *boosting Adaboost* sebagai optimasi level algoritma *KNN (K-Nearest Neighbor)* dengan didapatkan akurasi 85.11% dengan menggunakan validasi *10 fold cross validation* dengan parameter nilai k pada algoritma *K-NN* bernilai 5 untuk

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
				level algoritma pada algoritma *KNN (K-Nearest Neighbor)*.	prediksi harapan hidup pasien pasca operasi bedah toraks.
9.	Ikhsan Wisnuadji Gamadarenda, dan Indra Waspada	2020	Implementasi *Data Mining* Untuk Deteksi Penyakit Ginjal Kronis (Pgk) Menggunakan *KNN (K-Nearest Neighbor)* Dengan *Backward Elimination*	Penyakit ginjal kronis (PGK) merupakan masalah kesehatan yang dihadapi oleh dunia dengan angka kejadian yang terus meningkat. Deteksi PGK membutuhkan banyak atribut, sehingga membutuhkan biaya yang cukup mahal. Penelitian ini menggunakan metode *data mining* dengan mengimplementasikan algoritma *KNN (K-Nearest Neighbor)* dengan menambahkan algoritma *backward elimination* pada tahap *preprocessing* sehingga lebih optimal dan dapat menekan biaya pemeriksaan laboratorium.	Hasil pemodelan *data mining* terbaik dari sistem dibuat menggunakan *Backward Elimination* ($\alpha = 0{,}05$) dan kNN ($k = 3$) dengan menghitung kenaikan biaya inspeksi dan sensitivitas tertinggi. Rekomendasi sistem menghasilkan 10 atribut terpilih dari 24 atribut awal yang digunakan yaitu : berat jenis (sg), albumin (al), ureum darah (bu), kreatinin serum (sc), natrium (tanah), hemoglobin (hemo), sel darah merah (rbc), hipertensi (htn), diabetes mellitus (dm), dan nafsu makan (nafsu makan). Penggunaan atribut

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
					terpilih berhasil mencapai biaya pemeriksaan hingga 73,36%. Selanjutnya pendeteksian penyakit menggunakan algoritma *KNN (K-Nearest Neighbor)* menghasilkan nilai akurasi sebesar 99,25%, sensitivitas 99,5%, dan spesifisitas 98,745%.
10.	Irene Lishania, Rito Geojantoro, dan Yuki Novia Nasution	2019	Perbandingan Klasifikasi Metode *Naïve Bayes* dan Metode *Decision Tree* Algoritma *(J48)* pada Pasien Penderita Penyakit Stroke di RSUD Abdul Wahab	Penelitian ini akan membandingkan hasil akurasi klasifikasi yaitu *naïve bayes* dan *decision tree (J48*) pada pasien stroke. Artinya, seseorang yang mengalami stroke akan diklasifikasikan dengan menggunakan data pasien di RSUD Abdul Wahab Sjahranie Samarinda dengan 7 faktor yaitu usia, jenis kelamin, tekanan	Hasil ketepatan klasifikasi penyakit stroke pada data pasien di RSUD Abdul Wahab Sjahranie bulan November dan Desember 2017 dengan metode *naive bayes* adalah 81,25% dan metode *decision tree algoritma (J48)* diperoleh tingkat akurasi sebesar 87,5%. Hal ini menunjukkan bahwa pada penelitian

No.	Nama Pengarang	Tahun	Judul	Isi Ringkasan	Hasil
			Sjahranie Samarinda	darah, diabetes melitus, dislipidemia, kadar asam urat dan penyakit jantung.	ini, metode *decision tree algoritma (J48)* memberikan ketepatan prediksi klasifikasi yang lebih baik.

Pada *state of the art* tersebut belum ada penelitian yang membandingkan dua algoritma klasifikasi *k-nearst neighbor* dan *adaptive boosting* pada kasus prediksi penyakit stroke. Algoritma *k-nearst neighbor* dan *adaptive boosting* memiliki kekurangan dan kelebihan masing-masing, oleh karena itu penelitian ini akan membandingkan kedua algoritma tersebut untuk meperoleh hasil algoritma yang terbaik dan dengan menerapkan teknik *SMOTE* untuk mengatasi *imbalance data* pada data prediksi penyakit stroke ini.

2.2.2 Matriks Penelitian

Tabel 2. 3 Matriks Penelitian

No.	Penulis/ Tahun	Judul	Ruang Lingkup						
			Algoritma/Metode				Tujuan		
			Naïve Bayes	KNN	C.45	Ada Boost	Klasifikasi	Mengatasi *Imbalance* Data	Prediksi
1.	(Badriyah,dkk, 2020)	*Machine Learning Algorithm for Stroke Disease Classification*	√	√	√	-	-	√	-
2.	(Tulus dan Sihombing, 2021)	*Comparison of Distance Models on KNN (K-Nearest Neighbor) Algorithm in Stroke Disease Detection.*	-	√	-	-	-	√	-

No.	Penulis/ Tahun	Judul	Ruang Lingkup						
			Algoritma/Metode				Tujuan		
			Naïve Bayes	KNN	C.45	Ada Boost	Klasifikasi	Mengatasi *Imbalance* Data	Prediksi
3.	(Novianti,dkk, 2022)	Penerapan Algoritma *Adaboost* Untuk Peningkatan Kinerja Klasifikasi *Data Mining* Pada *Imbalance Dataset* Diabetes	-	√	-	√	-	√	-
4.	(Sumiah dan Mirantika, 2020)	Perbandingan Metode *K-Nearest Neighbor* dan *Naive Bayes* untuk Rekomendasi Penentuan Mahasiswa Penerima Beasiswa pada Universitas Kuningan	√	√	-	-	-	√	-
5.	(Prasetio dan Susanti, 2019)	Prediksi Harapan Hidup Pasien Kanker Paru Pasca Operasi Bedah	-	√	-	√	-	√	-

No.	Penulis/ Tahun	Judul	Ruang Lingkup						
			Algoritma/Metode				Tujuan		
			Naïve Bayes	KNN	C.45	Ada Boost	Klasifikasi	Mengatasi *Imbalance* Data	Prediksi
		Toraks Menggunakan *Boosted KNN (K-Nearest Neighbor)*							
6.	(Gamadarendra dan Waspada, 2020)	Implementasi *Data Mining* Untuk Deteksi Penyakit Ginjal Kronis (Pgk) Menggunakan *K-Nearest Neighbor* (*K-NN*) Dengan *Backward Elimination.*	-	√	-	-	√	√	-
7.	(Nopi, 2022)	Penerapan Metode *Adaboost* Untuk Optimalisasi Kerja Algoritma *K-Nearest Neighbor* Untuk Prediksi Penyakit Stroke	-	√	-	√	-	√	√