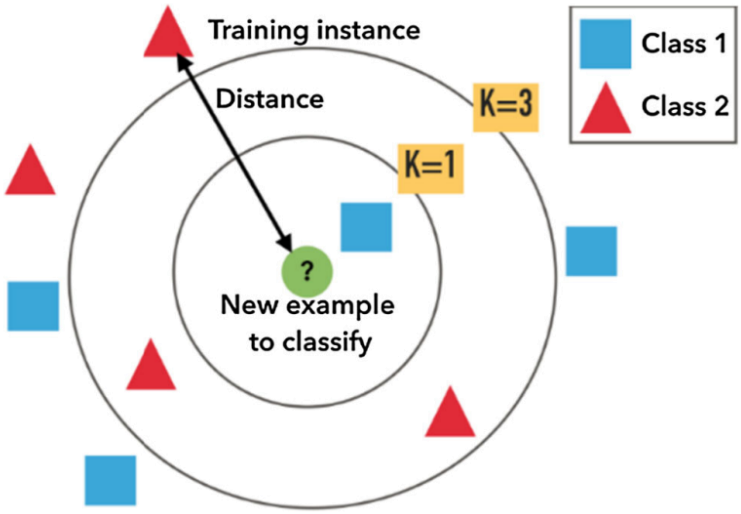
BAB II

LANDASAN TEORI

2.1 K-Nearest Neighbors (KNN)

K-Nearest Neighbors merupakan salahsatu algoritma *nonparametic* dalam permasalahan klasifikasi dan regresi (Cover & Hart, 1967). KNN mengasumsikan bahwa data yang serupa cenderung berada dalam kelompok yang sama atau memiliki nilai yang serupa. Saat diberikan data baru, KNN mencari tetangga terdekat dalam dataset pelatihan menggunakan metrik jarak seperti *Euclidean distance* atau *Manhattan distance* (Du & Li, 2019). Selanjutnya, KNN menentukan kelas mayoritas atau nilai rata-rata dari tetangga-tetangga tersebut sebagai prediksi untuk data baru tersebut.



Gambar 2. 1 Ilustrasi *K-Nearest Neighbors* (KNN)

Konsep KNN dapat diilustrasikan dalam Gambar 2.1. Sebagai contoh, sampel lingkaran uji hijau dapat dikelompokkan ke dalam kategori pertama persegi biru atau kategori kedua segitiga merah. Jika nilai K yang berarti jumlah tetangga adalah satu maka sampel tersebut akan dikategorikan sebagai kelas persegi karena jarak

tetangga yang terdekat dari sampel adalah persegi. Sedangkan jika jumlah tetangga adalah tiga (lingkaran eksternal), sampel tersebut akan diklasifikasikan ke dalam kelas segitiga karena terdapat dua segitiga dan hanya satu persegi di dalam lingkaran.

2.2 Distance Function

Jarak antara titik A dan B dalam ruang fitur telah diukur dalam penelitian sebelumnya menggunakan berbagai fungsi jarak yang berbeda. Terdapat beberapa fungsi jarak yang tersedia untuk klasifikasi KNN, seperti *Euclidean*, *Minkowski*, *correlation* dan *chi-square* (Thaseen et al., 2019). Fungsi tersebut diuraikan dalam persamaan (1), (2), (3) dan (4).

$$dist\ (A,B) = \sqrt{\sum_{i=1}^{m}(xi - yi)^2} \tag{1}$$

$$\text{dist\_Minkowsky}(A,\ B) = \left(\sum_{i=1}^{m} |x_i - y_i|^r\right)^{1/r} \tag{2}$$

$$\text{dist\_correlation}(A,\ B) = \frac{\sum_{i=1}^{m}(x_i - \mu_i)\,(y_i - \mu_i)}{\sqrt{\sum_{i=1}^{m}(x_i - \mu_i)^2 \sum_{i=1}^{m}(y_i - \mu_i)^2}} \tag{3}$$

$$\text{dist\_Chi} - \text{square}(A,\ B) = \sum_{i=1}^{m} \frac{1}{\text{sum}_i}\left(\frac{x_i}{\text{size}_Q} - \frac{y_i}{\text{size}_I}\right)^2 \tag{4}$$

Dari semua metode yang ada, fungsi yang akan digunakan dalam penelitian ini adalah Euclidean karena merupakan fungsi jarak yang paling paling umum digunakan dan memiliki tingkat identifikasi kemiripan lebih baik dibanding fungsi yang lain (Nishom, 2019; Pribadi et al., 2022) . Dalam fungsi euclidean, titik A dan B direpresentasikan oleh vektor fitur A = (x1, x2,..., xm) dan B = (y1, y2,..., ym),

di mana m merujuk pada dimensi dari ruang fitur. Untuk mengukur kesamaan antara dokumen-dokumen dalam pencarian uji, biasanya digunakan pengukuran kesamaan kosinus. Persamaan untuk menghitung kesamaan kosinus ini diberikan pada persamaan (5).

$$\text{sim}(A, B) = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} \tag{5}$$

Dalam persamaan ini, pembilang menggambarkan hasil perkalian titik (*dot product*) dari panjang A dan B, sedangkan hasil perkalian panjang Euclidean keduanya disajikan dalam penyebut.

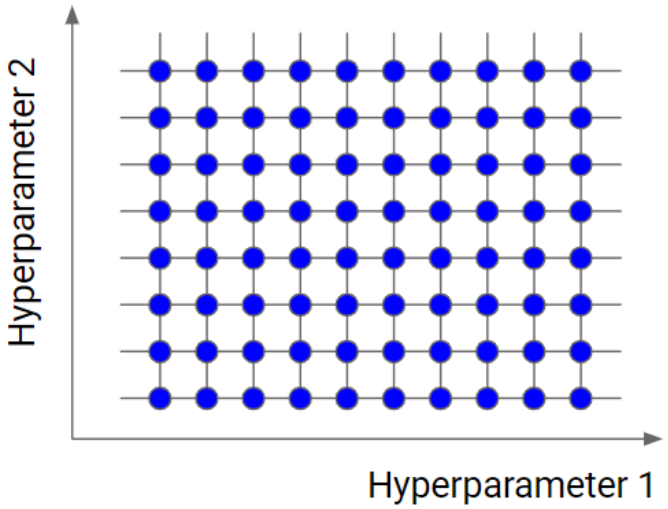
2.3 Hyperparameter Tuning

Hyperparameter Tuning adalah proses penyetelan parameter yang tidak dipelajari secara otomatis dalam algoritma *machine learning* untuk menemukan nilai yang paling oprtimal pada suatu model (Yang & Shami, 2020). Dalam konteks ini, *hyperparameter* merujuk pada parameter yang digunakan untuk mengontrol proses pembelajaran algoritma, seperti *learning rate* dalam algoritma *Neural Networks,* nilai K dalam algoritma *K-Nearest Neighbors* (KNN), dan *kernel size* dalam algoritma *Support Vector Machine* (SVM) (Yu & Zhu, 2020).

Proses *Hyperparameter Tuning* dilakukan dengan menguji berbagai nilai *hyperparameter* untuk mencari kombinasi yang paling optimal untuk meminimalkan kesalahan model atau meningkatkan kinerja model pada dataset tertentu. Ada beberapa metode yang umum digunakan dalam *Hyperparameter Tuning*, seperti *Grid Search, Random Search, Bayesian optimization*, dan *algoritma genetika.*

2.4 Grid Search

Grid Search merupakan salah satu algoritma yang sering digunakan dalam mencari parameter optimal bagi suatu model dalam *Machine Learning*. Pendekatan ini bekerja dengan cara mencoba semua kombinasi yang mungkin berdasarkan pada nilai parameter yang telah ditentukan sebelumnya oleh pengguna. Proses ini mirip dengan pembentukan sebuah *"grid"* di mana setiap sel dalam *grid* mewakili satu kombinasi parameter yang diuji. Selama proses pencarian, parameter-parameter yang telah dikombinasikan dievaluasi menggunakan suatu metrik evaluasi, seperti galat (*error*) atau akurasi model (Saputra et al., 2019).

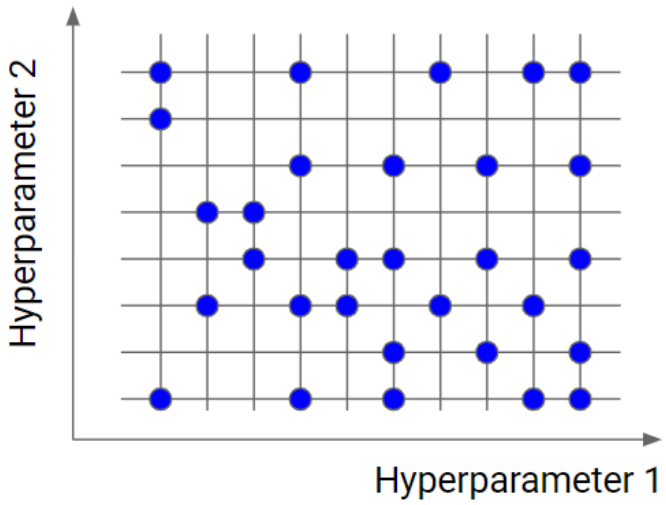


Gambar 2. 2 Visual of *Grid Search*

Hasil dari setiap kombinasi parameter disimpan dalam grid, dan parameter terbaik dipilih berdasarkan kriteria tertentu, seperti memiliki nilai galat terkecil. Dengan demikian, *Grid Search* memberikan cara sistematis untuk mengeksplorasi ruang parameter dan membantu pengguna dalam menemukan konfigurasi parameter yang optimal untuk meningkatkan kinerja model.

2.5 Random Search

Random Search merupakan metode alternatif dalam penyetelan parameter model yang berbeda dengan pendekatan *Grid Search*. Jika *Grid Search* mencoba semua kombinasi parameter secara sistematis, *Random Search* mengambil sampel secara acak dari ruang parameter yang telah ditentukan, kemudian menggabungkannya menjadi kombinasi. Pendekatan ini memungkinkan eksplorasi yang lebih efisien dalam ruang parameter, tanpa perlu menguji setiap kemungkinan kombinasi (Valarmathi & Sheela, 2021).

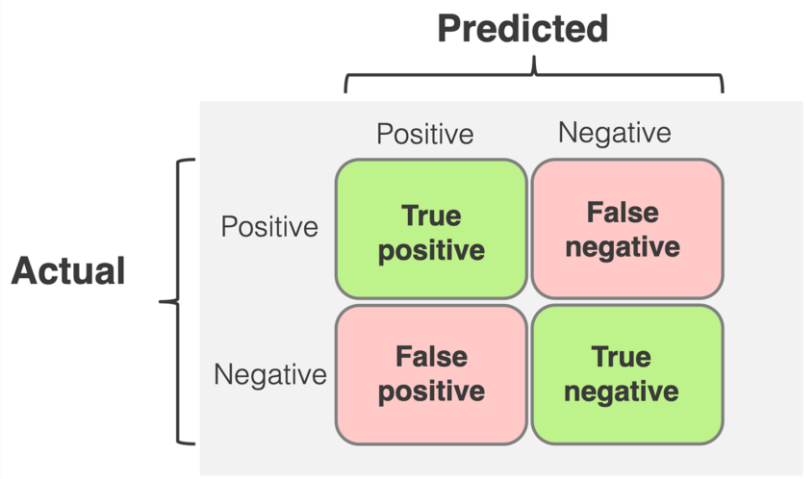


Gambar 2. 3 Visual of *Random Search*

Random Search lebih fokus pada eksplorasi nilai parameter yang berpotensi memiliki dampak signifikan pada performa model, dengan memprioritaskan kombinasi yang lebih mungkin memberikan hasil terbaik. Dengan kata lain, *Random Search* tidak terikat pada struktur grid tetap seperti *Grid Search*, sehingga memungkinkan untuk menemukan solusi optimal dengan cara yang lebih fleksibel dan efisien.

2.6 Confusion Matrix

Confusion matrix merupakan alat evaluasi yang digunakan dalam *Machine Learning* untuk mengukur kinerja dari sebuah model klasifikasi. *Confusion Matrix* menyajikan ringkasan hasil prediksi dengan menyediakan informasi perbandingan antara hasil klasifikasi yang diprediksi oleh model (*Predict value*) dengan hasil yang seharusnya (*Actual value*) (Satria et al., 2020).



Gambar 2. 4 Tabel Confusion Matrix

Dalam *confusion matrix*, terdapat empat istilah yang digunakan sebagai representasi hasil dari proses klasifikasi. yaitu, *True Positive* (TP) adalah ketika model dengan benar mengidentifikasi sampel positif, *True Negative* (TN) adalah ketika model dengan benar mengidentifikasi sampel negatif. Sementara itu, *False Positive* (FP) terjadi ketika model salah mengidentifikasi sampel negatif sebagai positif, dan *False Negative* (FN) terjadi ketika model salah mengidentifikasi sampel positif sebagai negatif (Suryadewiansyah & Tju, 2022).

Nilai dari Confusion Matrix tersebut nantinya akan digunakan untuk menghitung *accuracy, precision, recall, specificity* dan *f-measure* sebagai hasil nilai

evaluasi dari suatu model dengan menggunakan persamaan (6), (7), (8), (9) dan (10) (Afrillia et al., 2022).

$$Accurracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

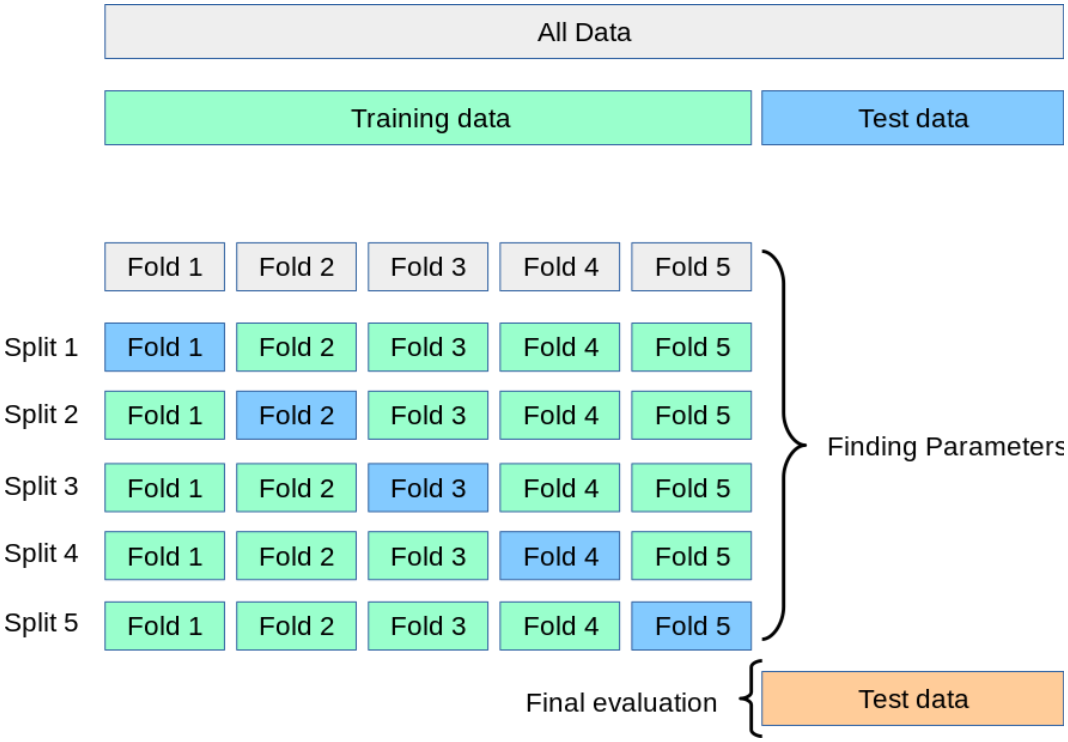
$$Specificity = \frac{TN}{TN+FP} \quad (9)$$

$$F1\ Score = 2 \times \frac{Recall \times Precission}{Recall + Precission} \quad (10)$$

2.7 K-Fold Cross Validation

K-fold cross-validation merupakan metode statistik yang sering digunakan untuk mengevaluasi performa model *machine learning*. Metode ini membagi dataset menjadi beberapa subset atau *"fold"* yang sama besar (FUADAH et al., 2022). Model dilatih menggunakan semua fold kecuali satu, yang digunakan sebagai data validasi. Proses ini diulangi k kali, sehingga setiap *fold* digunakan sekali sebagai data validasi dan k-1 kali sebagai data pelatihan. Penelitian ini akan menggunakan 5-fold untuk mengevaluasi kinerja model. Penting untuk memperhatikan nilai k pada k-fold cross-validation karena k tersebut dapat mempengaruhi hasil akurasi model. K yang lebih kecil dapat menyebabkan estimasi akurasi yang lebih bervariasi dan kurang stabil karena lebih sedikit data untuk pelatihan. Sebaliknya, k yang lebih besar memberikan estimasi akurasi yang lebih stabil dan andal, namun meningkatkan waktu komputasi (Nti et al., 2021). Metode

K-Fold Cross direkomendasikan karena memberikan estimasi yang baik tentang kemampuan model untuk menggeneralisasi data yang tidak terlihat (Peryanto et al., 2020a).



Gambar 2. 5 Ilustrasi Proses 5-fold Cross Validation

Gambar 2.5 menunjukan dataset dibagi menjadi lima bagian yang sama besar. Pada setiap iterasi, empat bagian digunakan sebagai data pelatihan dan satu bagian sebagai data validasi. Misalnya, dari total 2000 data Diabetes, 1600 data digunakan untuk pelatihan dan 400 data untuk pengujian pada setiap iterasi. Proses ini diulangi sampai setiap fold dari 5-fold telah digunakan sebagai set pengujian. Metode ini mengatasi masalah variabilitas kinerja model yang mungkin terjadi jika hanya satu set data uji digunakan. Dengan menggunakan k-fold cross-validation, setiap bagian

dari dataset berkontribusi sebagai data pelatihan dan data validasi, menghasilkan evaluasi model yang lebih stabil dan akurat.

2.8 Penelitian Terkait

Penelitian tentang *tuning hyperparameter* telah menjadi fokus perhatian beberapa peneliti sebelumnya yang bertujuan untuk mengidentifikasi opsi terbaik dalam menentukan parameter yang tepat untuk suatu algoritma tertentu. Penelitian tersebut dilakukan dalam berbagai konteks aplikasi di dunia nyata. Tabel 2.1 akan menjadi dasar landasan dari penelitian ini yang merangkum *state of the art* dari penelitian terkait yang relevan.

Tabel 2. 1 *State of the Art*

Penulis	**Judul**	**Metode Klasifikasi**	**Metode Tuning**	**Hasil Penelitian**
(Ambesange et al., 2020a)	*Liver Diseases Prediction using KNN with Hyper Parameter Tuning Techniques*	*K-Nearest Neighbors (KNN)*	*Grid Search*	Penelitian ini menggunakan K-Nearest Neighbors dengan penyetelan hiperparameter *Grid Search* untuk memprediksi penyakit liver. Hasilnya menunjukkan peningkatan performa dan akurasi dari 80% menjadi 91% setelah dilakukan tuning pada hyperparameter.
(Khatib Sulaiman et al., 2023)	*Hyperparameter Tuning Algoritma Supervised Learning untuk Klasifikasi Keluarga Penerima*	Support Vector Machine (SVM), Decision Tree, Naïve Bayes, K-	*Grid Search, Random Search, Bayesian Optimization*	Penelitian ini bertujuan mengklasifikasikan penerima bantuan pangan beras menggunakan berbagai metode klasifikasi dengan pendekatan hyperparameter yang berbeda. Hasil pengujian menunjukkan bahwa secara keseluruhan, metode *Hyperparameter*

Penulis	Judul	Metode Klasifikasi	Metode Tuning	Hasil Penelitian
	Bantuan Pangan Beras	Nearest Neighbors (KNN)		*Tuning* memiliki performa yang setara dengan akurasi mencapai 74%.
(Wazirali, 2020)	*An Improved Intrusion Detection System Based on KNN Hyperparameter Tuning and Cross-Validation*	K-Nearest Neighbors (KNN)	*Grid Search*	Penelitian ini mengembangkan metode baru untuk deteksi intrusi menggunakan pembelajaran mesin. Pendekatan yang diusulkan mengoptimalkan kinerja dengan memanfaatkan k-nearest neighbors, penyetelan hyperparameter, dan cross validation. Eksperimen yang dilakukan pada dataset NSL-KDD menunjukkan peningkatan akurasi yang signifikan.
(Fajri & Primajaya, 2023b)	*Komparasi Teknik Hyperparameter Optimization pada SVM untuk Permasalahan Klasifikasi dengan Menggunakan Grid Search dan Random Search*	Support Vector Machine (SVM)	*Grid Search*, *Random Search*	Dari hasil percobaan penelitian ini, disimpulkan bahwa dari segi nilai akurasi, baik *Grid Search* maupun *Random Search* tidak menunjukkan perbedaan yang signifikan, dengan nilai rata-rata dari kedua metode sebesar 84%. Meskipun demikian, penggunaan memori untuk *Random Search* cenderung lebih tinggi sekitar 3,4 Mebibytes dibandingkan dengan *Grid Search*, meskipun secara umum perbedaannya tidak signifikan.
(Pramudhyta &	*Perbandingan Optimasi Metode*	XGBoost	*Grid Search*,	Hasil penelitian menunjukkan bahwa optimasi parameter melalui *Grid*

Penulis	Judul	Metode Klasifikasi	Metode Tuning	Hasil Penelitian
Rohman, 2024b)	*Grid Search dan Random Search dalam Algoritma XGBoost untuk Klasifikasi Stunting*		*Random Search*	*Search* dan *Random Search* meningkatkan akurasi, presisi, dan recall dibandingkan dengan model XGBoost awal. Model XGBoost awal memiliki akurasi 83.28%, presisi 83.92%, recall 82.66%, dan F1-Score 83,28%. Model *Grid Search* mencapai akurasi tertinggi 89.09%, presisi tertinggi 94.55%, recall 85.12%, dan F1-Score tertinggi 89.58%. Sedangkan Model *Random Search* juga memberikan hasil yang baik dengan akurasi 88.71%, presisi 93.10%, recall tertinggi 85.47%, dan F1-Score 89.12%.
(Abdulraheem et al., 2023)	*Hyper-parameter tuning for support vector machine using an improved cat swarm optimization algorithm*	Support Vector Machine (SVM)	Cat Swarm Optimization (CSO)	Hasil eksperimen menunjukkan bahwa ICSO-SVM dan CSO-SVM mencapai akurasi rata-rata 91.2% dan 85.71% untuk lebih dari 15 dataset. ICSO-SVM mengungguli CSO-SVM dengan standar deviasi yang lebih rendah pada beberapa dataset.
(Villalobos-Arias et al., 2020)	*Evaluating hyper-parameter tuning using Random Search in support vector machines*	Support Vector Machine (SVM)	*Random Search*	Hasil penelitian ini menunjukan bahwa penyetelan hiperparameter dan pemrosesan data krusial dalam membangun model SVR yang prediktif, stabilitas dan akurasi model dapat ditingkatkan melalui penyetelan

Penulis	Judul	Metode Klasifikasi	Metode Tuning	Hasil Penelitian
	for software effort estimation			hiperparameter yang tepat, seperti *Grid Search* atau *Random Search*, dan pencarian acak yang kurang intensif memiliki kinerja yang sebanding dengan pencarian grid, sehingga menjadi alternatif baik untuk penyetelan hiperparameter.
(Agrawal & Chakraborty, 2021)	*On the use of acquisition function-based Bayesian optimization method to efficiently tune SVM hyperparameters for structural damage detection*	Support Vector Machine (SVM)	Bayesian Optimization	Penelitian ini mencari metode efisien untuk melatih klasifikasi SVM secara optimal dalam aplikasi deteksi kerusakan. Validasi dilakukan dengan mencari solusi lengkap untuk masalah deteksi kerusakan pada sistem eksperimental. Dibandingkan dengan metode PSO yang membutuhkan kelompok besar anggota, Bayesian menunjukkan jumlah evaluasi fungsi yang lebih baik. Hal ini ditunjukan dengan nilai tinggi dari metode optimisasi Bayesian dalam membandingkan berbagai metode statistic dalam konteks pengoptimalan.
(Kalita et al., 2021)	*A dynamic framework for tuning SVM hyper parameters based on Moth-Flame*	Support Vector Machine (SVM)	Moth-Flame Optimization, knowledge-	Penelitian ini menekankan penyetelan hiperparameter SVM dalam lingkungan yang dinamis, yang umum terjadi pada model klasifikasi karena dinamikanya data dunia nyata.

Penulis	Judul	Metode Klasifikasi	Metode Tuning	Hasil Penelitian
	Optimization and knowledge-based-search		based-search	Penelitian diperluas untuk masalah klasifikasi dunia nyata yang dinamis terutama di bidang kedokteran. Kerangka kerja yang ada pada penelitian ini lebih unggul daripada berbagai kerangka kerja lain yang menggunakan optimisasi modern seperti PSO, MVO, GWO, CS, WOA, GA, FFA, dan SSA.
(Muhajir et al., 2021)	*Improving classification algorithm on education dataset using Hyperparameter Tuning*	K-Nearest Neighbors (KNN)	*Random Search*	Dalam penelitian ini, teknik-teknik supervised *Machine Learning* diterapkan pada dataset Rekrutmen Kampus dan faktor-faktor keterampilan penempatan. Penyetelan hiperparameter dilakukan untuk meningkatkan hasil, dengan evaluasi menunjukkan peningkatan rata-rata 2% pada setiap algoritma.
(Assegie et al., 2023)	*Early Prediction of Gestational Diabetes with Parameter-Tuned K-Nearest Neighbor Classifier*	K-Nearest Neighbors (KNN)	*Grid Search*	Dalam penelitian ini, tuning parameter dan standarisasi fitur input digunakan untuk meningkatkan kinerja model KNN dalam prediksi diabetes. Hasil eksperimen menunjukkan model KNN yang dikembangkan mengungguli model KNN yang ada, dengan penskalaan fitur dianggap berperan penting dalam peningkatan kinerja.

Penulis	Judul	Metode Klasifikasi	Metode Tuning	Hasil Penelitian
(Assegie, 2021)	*An optimized K-Nearest neighbor based breast cancer detection*	K-Nearest Neighbors (KNN)	*Grid Search*	Penelitian ini mengusulkan model KNN yang dioptimalkan untuk prediksi kanker payudara dengan pendekatan *Grid Search*. Eksperimen menunjukkan peningkatan signifikan dalam kinerja dengan akurasi meningkat dari 90,10% menjadi 94,35% menggunakan hiperparameter terbaik.
(Aghaabbasi et al., 2023)	*On Hyperparameter Optimization of Machine Learning Methods Using a Bayesian Optimization Algorithm to Predict Work Travel Mode Choice*	K-Nearest Neighbors (KNN)	Bayesian Optimization	Penelitian ini menggunakan metode RF untuk memilih input yang relevan dari dataset Iowa dan Ohio. Hasilnya menunjukkan model SVM memiliki kesalahan klasifikasi minimum, sementara model KNNBO mencapai akurasi tertinggi. Metode BO efektif meningkatkan akurasi prediksi kelas minoritas pada model EDT dan SVM, serta meningkatkan klasifikasi dan AUC pada model KNN standar. Temuan ini menegaskan bahwa pendekatan BO dapat meningkatkan kemampuan model ML dasar dalam memprediksi pilihan mode perjalanan kerja, terutama pada model KNN.

Penulis	Judul	Metode Klasifikasi	Metode Tuning	Hasil Penelitian
(Andonie & Florea, 2020)	*Weighted Random Search for CNN hyperparameter optimization*	Convolutional Neural Network (CNN)	*Random Search*	Berdasarkan hasil eksperimen, algoritma optimisasi WRS memiliki kinerja yang lebih baik daripada beberapa metode optimisasi hiperparameter terkini. Dengan kode yang di buat tersedia secara publik, peneliti berharap bahwa eksperimen lebih lanjut pada arsitektur lain akan mengonfirmasi hasil ini.
(Firdaus et al., 2021)	*Deep Neural Network with Hyperparameter Tuning for Detection of Heart Disease*	Deep Neural Network (DNN)	*Grid Search*, *Random Search*, Bayesian optimization	Penelitian ini mengusulkan metode deteksi penyakit jantung menggunakan jaringan saraf tiruan dalam deep learning dengan penyetelan *hiperparameter* menggunakan *Grid Search*, *Random Search*, dan optimisasi *Bayesian*. DNN dengan optimisasi Bayesian menunjukkan kinerja terbaik dengan akurasi pengujian 91,67%, sensitivitas 95,83%, spesifisitas 88,89%, presisi 85,19%, F1-score 90,20%, dan nilai AUC 0,9514, menunjukkan keunggulan dalam mendeteksi penyakit jantung.

Mengacu pada temuan-temuan yang telah ada pada penelitian sebelumnya, penelitian ini diharapkan mampu memberikan wawasan yang lebih mendalam mengenai metode yang paling efektif dalam menentukan hyperparameter yang

optimal pada dataset yang beragam terutama pada konteks membandingkan metode *Grid Search* dan *Random Search* sebagai teknik *Hyperparameter Tuning* pada Algoritma K-Nearest Neighbors (KNN).

Tabel 2.2 menunjukkan perbandingan capaian yang akan didapatkan pada penelitian ini dengan capaian yang telah didapatkan oleh penelitian sebelumnya dalam cakupan pengembangan dan implementasi *Hyperparameter Tuning*.

Tabel 2. 2 Matriks penelitian

No	Penelitian	Ruang Lingkup						
		Classification		Tuning method		Parameter uji		
		KNN	*Other method*	*Grid Search*	*Random Search*	*accuracy*	*Memory Usage*	*Running Time*
1	(Ambesange et al., 2020b)	✓	○	✓	○	✓	○	○
2	(Khatib Sulaiman et al., 2023)	✓	✓	○	✓	✓	○	○
3	(Wazirali, 2020)	✓	○	✓	○	✓	○	○
4	(Fajri & Primajaya, 2023b)	○	✓	✓	✓	✓	✓	✓
5	(Pramudhyta & Rohman, 2024b)	○	✓	✓	✓	✓	✓	✓
6	(Abdulraheem et al., 2023)	○	✓	○	○	✓	○	○
7	(Villalobos-Arias et al., 2020)	○	✓	○	✓	✓	○	○

No	Penelitian	Ruang Lingkup						
		Classification		Tuning method		Parameter uji		
		KNN	*Other method*	*Grid Search*	*Random Search*	*accuracy*	*Memory Usage*	*Running Time*
8	(Agrawal & Chakraborty, 2021)	○	✓	○	○	✓	○	○
9	(Kalita et al., 2021)	○	✓	○	○	✓	○	○
10	(Muhajir et al., 2021)	✓	○	○	✓	✓	○	○
11	(Assegie et al., 2023)	✓	○	✓	○	✓	○	○
12	(Assegie, 2021)	✓	○	✓	○	✓	○	○
13	(Aghaabbasi et al., 2023)	✓	○	○	○	✓	○	○
14	(Andonie & Florea, 2020)	○	✓	○	✓	✓	○	○
15	(Firdaus et al., 2021)	○	✓	✓	✓	✓	○	○
16	Our research	✓	○	✓	✓	✓	✓	✓

2.9 Kebaruan Penelitian

Penelitian ini dilatarbelakangi oleh belum adanya kajian komprehensif yang membandingkan metode *Grid Search* dan *Random Search* dalam konteks pengoptimalan *hyperparameter* pada algoritma *K-Nearest Neighbors* (KNN). Hal ini mengingat setiap algoritma *machine learning* memiliki jenis dan karakteristik *hyperparameter* yang berbeda, seperti *kernel* pada *Support Vector Machine* (SVM) dan *learning rate* pada *XGBoost*, yang memerlukan pendekatan pengoptimalan

spesifik untuk mencapai performa terbaik. Hingga saat ini, penelitian mengenai pengoptimalan *hyperparameter* pada KNN cenderung hanya fokus pada implementasi praktis tanpa analisis perbandingan yang mendalam terhadap metode optimasi yang digunakan. Penelitian ini bertujuan untuk mengisi kekosongan tersebut dengan melakukan perbandingan sistematis antara *Grid Search* dan *Random Search* dalam mengoptimalkan *hyperparameter* pada algoritma KNN.