Indonesian Journal of Computer Science
ISSN 2302-4364 (*print*) dan 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam, No. 1, Padang, Indonesia, Telp. (0751) 7056199, 7058325
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Hyperparameter Tuning Algoritma Supervised Learning untuk Klasifikasi Keluarga Penerima Bantuan Pangan Beras

Joshua Agung Nurcahyo[1], Theopilus Bayu Sasongko[2]
joshuagung@students.amikom.ac.id, theopilus.27@amikom.ac.id
Universitas Amikom Yogyakarta

Informasi Artikel

Diterima : 26 Jun 2023
Direview : 29 Jun 2023
Disetujui : 30 Jun 2023

Kata Kunci

Klasifikasi, *Supervised learning*, *Hyperparameter tuning*, Bantuan pangan, IFLS

Abstrak

Indonesia memiliki berbagai macam program untuk menekan kemiskinan, salah satunya adalah program bantuan pangan beras. Namun, berdasarkan temuan di lapangan, program bantuan ini tidak tepat sasaran. Melalui klasifikasi *supervised learning* dengan *hyperparameter tuning*, penelitian ini bertujuan untuk mengetahui algoritma klasifikasi umum yang paling optimal dan akurat dalam menentukan keluarga penerima bantuan pangan beras. Algoritma *Support Vector Machine* (SVM), *Decision Tree*, *Naïve Bayes*, dan *K-Nearest Neighbor* (KNN) serta metode *hyperparameter tuning grid search*, *random search*, dan *Bayesian optimization* digunakan dalam penelitian. Data pada penelitian ini bersumber dari IFLS. Berdasarkan hasil analisis, penerapan *hyperparameter tuning* memiliki manfaat dalam meningkatkan kinerja algoritma KNN, *Decision Tree*, dan SVM. Algoritma KNN dengan *random search* serta *Bayesian optimization* dan SVM dengan *Bayesian optimization* memberikan nilai *accuracy* yang sama, yakni sebesar 74%. Oleh karena itu, model tersebut memiliki kinerja yang setara dan sama baiknya dalam mengklasifikasikan keluarga penerima bantuan pangan beras.

Keywords

Classification, Supervised learning, Hyperparameter tuning, Food aid, IFLS

Abstrak

Indonesia has various programs to reduce poverty, including the rice food assistance program. However, based on findings in the field, this assistance program was not on target. Through supervised learning classification with hyperparameter tuning, this study aims to determine the most optimal and accurate general classification algorithm in determining rice food assistance beneficiary families. Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and K-Nearest Neighbor (KNN) algorithms, as well as the hyperparameter tuning grid search, random search, and Bayesian optimization methods, were used in the research. The data in this study were sourced from IFLS. Based on the analysis results, the application of hyperparameter tuning has benefits in improving the performance of the KNN, Decision Tree, and SVM algorithms. The KNN algorithm with random search and Bayesian optimization and SVM with Bayesian optimization gives the same accuracy value, which is 74%. Therefore, the model performs equally well in classifying families receiving rice food assistance.

A. Pendahuluan

Kemiskinan merupakan permasalahan sosial fundamental yang terjadi di Indonesia. Berdasarkan data Badan Pusat Statistik (BPS), penduduk miskin pada tahun 2022 mencapai 26,36 juta individu atau setara dengan 9,57% dari total penduduk indonesia [1]. Angka kemiskinan tersebut menunjukkan penurunan dibandingkan dengan sepuluh tahun sebelumnya. Pada tahun 2012, penduduk miskin mencapai 28,59 juta orang atau setara dengan 11,66% dari total penduduk [2]. Penurunan kemiskinan di Indonesia dapat terjadi karena pemerintah mengeluarkan berbagai macam kebijakan dan program untuk mengurangi angka kemiskinan. Salah satu program pemerintah yang ditetapkan untuk menekan angka kemiskinan adalah program bantuan pangan beras [3].

Program bantuan pangan beras terus mengalami perkembangan dari tahun ke tahun. Pada tahun 1998, diperkenalkan program operasi pasar khusus beras. Pada tahun 2002, program ini mengalami perubahan nama menjadi Raskin (Beras untuk Keluarga Miskin) guna meningkatkan akurasi sasaran bantuan. Perubahan nama tersebut memiliki maksud agar keluarga yang tidak tergolong miskin merasa enggan untuk mengambil bagian dalam mendapatkan Raskin [4]. Pada tahun 2015, Raskin kemudian berganti nama menjadi Rastra (Beras Sejahtera) [5]. Namun, pada tahun 2017, program Rastra berakhir dan digantikan oleh program Bansos Rastra (Bantuan Sosial Beras Sejahtera) [6]. Secara keseluruhan, dapat disimpulkan bahwa program bantuan pangan beras mengalami perubahan dalam terminologi dan penyempurnaan program dari waktu ke waktu. Meskipun demikian, tujuan utama program tersebut tetap sama, yakni menyediakan bantuan bulanan kepada keluarga miskin demi ketahanan pangan dan mengurangi beban pengeluaran rumah tangga melalui penyediaan sebagian kebutuhan pangan pokok dalam bentuk beras [7].

Penerima Bansos Rastra adalah keluarga dengan kondisi sosial ekonomi 25% terendah di daerah pelaksanaan program. Penerima ini disebut sebagai Keluarga Penerima Manfaat (KPM) Bansos Rastra dengan syarat namanya termasuk dalam daftar KPM dan ditetapkan oleh Menteri Sosial. Sumber data yang digunakan untuk menentukan KPM Bansos Rastra berasal dari Data Terpadu Program Penanganan Fakir Miskin (DT-PPFM) yang merupakan hasil dari Pemutakhiran Basis Data Terpadu (PBDT) [8]. Data terpadu ini langsung dikelola oleh Kementerian Sosial sebagai acuan dalam penanggulangan kemiskinan. Namun, berdasarkan temuan di lapangan terdapat ketidakakuratan sehingga penargetan kelompok penerima manfaat tidak tepat sasaran [9]. Kondisi tersebut menunjukkan perlunya teknis yang lebih baik dalam menentukan keluarga penerima bantuan pangan beras.

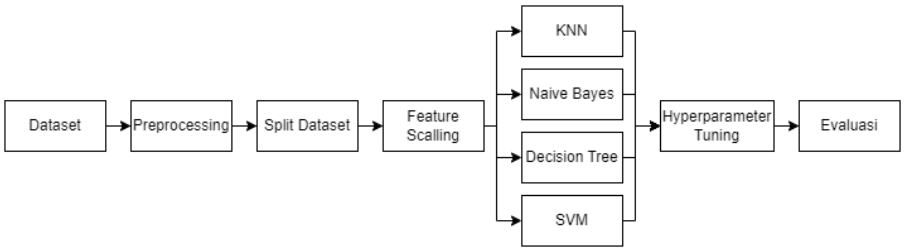
Klasifikasi dengan pendekatan *supervised learning* dapat diterapkan untuk menentukan kelayakan penerima bantuan pangan beras. Klasifikasi dengan pendekatan ini merupakan salah satu teknik dalam *machine learning* yang dapat memprediksi nilai dari sekelompok atribut dalam menggambarkan dan membedakan kelas data yang bertujuan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui [10]. Dengan kata lain, klasifikasi adalah teknik yang dapat menentukan label kelas dari *dataset* sehingga memungkinkan pengelompokan objek ke dalam kategori tertentu [11]. Klasifikasi dengan pendekatan *supervised learning* secara signifikan unggul dibandingkan dengan metode pengelompokan objek sejenis, yaitu *clustering* pada *unsupervised learning* [12]. Beberapa algoritma umum yang digunakan dalam klasifikasi *supervised*

learning meliputi *K-Nearest Neighbor* (KNN), *Naïve Bayes*, *Decision Tree*, dan *Support Vector Machine* (SVM) [13]. Dalam rangka membangun model klasifikasi yang memiliki kinerja maksimal, perlu dilakukan penyelidikan terhadap berbagai kemungkinan *hyperparameter* yang ada. Proses menemukan konfigurasi *hyperparameter* yang optimal dikenal sebagai *hyperparameter tuning* [14].

Berdasarkan permasalahan yang telah diuraikan, penelitian ini memiliki fokus untuk membandingkan berbagai algoritma yang umum digunakan dalam klasifikasi *supervised learning*, yaitu KNN, *Naïve Bayes, Decision Tree*, dan SVM disertai dengan *hyperparameter tuning* dengan metode *grid search*, *random search*, dan *Bayesian optimization*. Melalui penerapan berbagai algoritma dan *hyperparameter tuning*, penelitian ini bertujuan untuk mengetahui model yang paling optimal dan akurat dalam menentukan keluarga penerima bantuan pangan beras. Hasil penelitian ini diharapkan dapat memberikan wawasan dalam pengambilan keputusan program bantuan pangan beras.

B. Metode Penelitian

Tahapan dalam metode penelitian terdiri dari beberapa langkah, mulai dari pengumpulan data, *preprocessing*, *split dataset*, *feature scaling*, penerapan algoritma, *hyperparameter tuning* hingga evaluasi. *Preprocessing* merupakan teknik dalam data mining yang bertujuan mengubah data mentah menjadi format yang terstruktur dan dapat dipahami [15]. Data mentah seringkali tidak lengkap atau mengandung kesalahan sehingga dalam proses *preprocessing* data yang tidak lengkap dihilangkan dari *dataset*. Setelah *preprocessing*, tahap selanjutnya adalah membagi *dataset* menjadi dua bagian, yaitu *data train* dan *data test*. Pembagian dilakukan dengan proporsi 70% untuk *data train* dan 30% untuk *data test*. Setelah itu, dilakukan *feature scaling* menggunakan metode *Standard scaler*. *Standard scaler* adalah teknik standarisasi fitur yang menghilangkan rata-rata dan menormalkan unit varians yang bertujuan untuk mencegah adanya data yang memiliki nilai terlalu besar dibanding dengan nilai yang lain [16]. Setelah tahapan *feature scaling*, langkah berikutnya adalah menerapkan algoritma, *hyperparameter tuning*, dan evaluasi model. Gambar 1 dapat menunjukkan tahapan penelitian secara visual.



Gambar 1. Tahapan Metode Penelitian

1. Data

IFLS (*Indonesia Family Life Survey*) adalah survei individu, rumah tangga, dan komunitas yang dilakukan oleh RAND Corporation. Survei ini dilakukan di 13 provinsi di Indonesia, antara lain Provinsi Kalimantan Selatan, Sulawesi Selatan, Sumatera Utara, Sumatera Barat, Lampung, Sumatera Selatan, Nusa Tenggara Barat, DKI Jakarta, D.I. Yogyakarta, Jawa Tengah, Jawa Barat, Jawa Timur, dan Bali. Sampel

yang terdapat dalam survei IFLS 5 dapat mewakili sekitar 83% dari populasi Indonesia [17].

2. K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* (KNN) adalah metode klasifikasi yang digunakan untuk mengklasifikasikan data berdasarkan pembelajaran dari data yang sudah terklasifikasikan sebelumnya. Metode ini termasuk dalam *supervised learning*, di mana hasil dari *instance query* yang baru akan diklasifikasikan berdasarkan mayoritas kedekatan jaraknya dengan kategori yang ada dalam K-NN [18]. Kedekatan ini diukur menggunakan jarak metrik, seperti jarak Euclidean. Jarak Euclidean dapat dihitung menggunakan persamaan berikut ini [19].

$$jarak(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2} \quad (1)$$

Keterangan:
n = Dimensi data
X_1 = *Data train*
X_2 = *Data test*

3. Naïve Bayes

Naïve Bayes adalah salah satu metode yang dapat digunakan dalam klasifikasi data. *Naïve Bayes* merupakan metode klasifikasi yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas [20]. Metode Bayes adalah pendekatan statistik untuk melakukan inferensi induktif dalam masalah klasifikasi [21]. Dalam *Naïve Bayes*, diasumsikan bahwa keberadaan suatu fitur dalam suatu kelas tidak terkait dengan keberadaan fitur lain dalam kelas yang sama. Pada saat melakukan klasifikasi, pendekatan Bayes akan menghasilkan label kategori dengan probabilitas tertinggi [22]. Persamaan teorema Bayes adalah sebagai berikut.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2)$$

Keterangan:
A = Hipotesis data B merupakan suatu kelas spesifik
B = Data dengan kelas yang belum diketahui
P(A) = Probabilitas dari A
P(B) = Probabilitas dari B
P(A|B) = Probabilitas hipotesis A berdasarkan kondisi B
P(B|A) = Probabilitas hipotesis B berdasarkan kondisi A

4. Decision Tree

Algoritma *Decision Tree* adalah salah satu algoritma dalam *supervised machine learning* yang umumnya digunakan untuk menyelesaikan masalah klasifikasi [23]. *Decision Tree* mencari solusi untuk masalah dengan menggunakan kriteria sebagai *node* yang saling berhubungan yang membentuk struktur seperti pohon. Setiap pohon memiliki cabang dan setiap cabang mewakili fitur yang harus dipenuhi untuk menuju cabang selanjutnya hingga akhirnya mencapai daun sebagai hasil akhir [24]. Salah satu penerapan algoritma *Decision Tree* adalah dengan menggunakan metode C4.5. Langkah-langkah dalam algoritma C4.5 meliputi perhitungan nilai *entropy*,

nilai *information gain*, nilai *split info*, dan nilai *gain ratio* [25]. *Entropy* dihitung untuk setiap fitur menggunakan rumus sebagai berikut:

$$Entropy(S) = -P_{(+)} \log_2 P_{(+)} - P_{(-)} \log_2 P_{(-)} \quad (3)$$

Menghitung nilai *information gain* setiap fitur dengan menggunakan rumus:

$$Information\ Gain(S, A) = Entropi(S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} Entropi(Si) \quad (4)$$

Menghitung nilai *split info* setiap fitur dengan menggunakan rumus:

$$Split\ Info(S, A) = \sum_{i=1}^{n} \frac{|Si|}{|S|} \log_2 \frac{|Si|}{|S|} \quad (5)$$

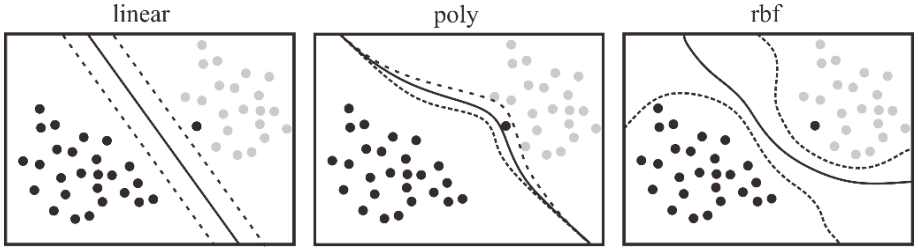
Menghitung *gain ratio* tiap fitur menggunakan rumus:

$$Gain\ ratio(S, A) = \frac{Information\ Gain(S, A)}{Split\ Info(S, A)} \quad (6)$$

Keterangan:
S = Himpunan kasus
A = Atribut
n = Jumlah atribut A
|Si| = Jumlah kasus pada partisi ke-i
|S| = Jumlah kasus dalam S

5. Support Vector Machine

Support Vector Machine (SVM) merupakan algoritma pembelajaran mesin yang didasarkan pada konsep statistik [26]. Dalam SVM, digunakan *hyperplane* sebagai pemisah antara kelompok kelas. *Hyperplane* ini terbentuk melalui dimensi vektor dengan ukuran n. Tujuan SVM adalah mencari *hyperplane* yang optimal dengan memaksimalkan jarak antara pemisah kelas yang disebut sebagai *margin*. Ilustrasi klasifikasi dengan SVM menggunakan berbagai jenis *hyperplane* dapat ditemukan pada Gambar 2. Gambar tersebut memberikan gambaran visual tentang bagaimana SVM mencari *hyperplane* yang optimal untuk memisahkan data dengan kelas-kelas yang berbeda [27].



Gambar 2. Jenis Hyperplane SVM

6. Grid Search

Grid search merupakan salah satu metode yang paling umum digunakan dalam eksplorasi *hyperparameter tuning*. Metode ini bekerja dengan melakukan pencarian secara menyeluruh terhadap semua kombinasi *hyperparameter* yang telah

ditentukan pada grid konfigurasi. Kelebihan *grid search* adalah kemampuannya untuk dijalankan secara paralel, di mana setiap percobaan berjalan secara mandiri tanpa dipengaruhi oleh urutan waktu. Oleh sebab itu, hasil dari satu percobaan tidak bergantung pada hasil percobaan lainnya. Selain itu, *grid search* juga memberikan fleksibilitas tinggi dalam alokasi sumber daya komputasi [28].

7. Random Search

Random search merupakan peningkatan dasar dari *grid search*. Metode ini melibatkan pencarian secara acak pada *hyperparameter* dengan menggunakan distribusi tertentu pada kemungkinan nilai parameter. Keunggulan *random search* adalah kemampuannya untuk dijalankan secara paralel pada arsitektur komputer yang memungkinkan pemrosesan yang efisien dan cepat. Selain itu, *random search* juga memberikan kemudahan dalam mengatasi kegagalan komputer saat menjalankan percobaan. Dengan berbagai fleksibilitas dan keunggulan yang dimiliki, *random search* menjadi metode yang efektif dan dapat diandalkan dalam *hyperparameter tuning* [29].

8. Bayesian Optimization

Bayesian optimization adalah metode populer dalam *hyperparameter tuning*. Metode ini memanfaatkan informasi hasil evaluasi sebelumnya untuk menentukan titik evaluasi selanjutnya. Dalam *Bayesian optimization*, terdapat dua komponen utama, yaitu model pengganti dan fungsi akuisisi. Model pengganti digunakan untuk memodelkan hubungan antara *hyperparameter* dan fungsi tujuan berdasarkan data evaluasi sebelumnya. Dari model pengganti tersebut, distribusi prediksi dapat diperoleh. Fungsi akuisisi digunakan untuk memilih titik evaluasi selanjutnya dengan mempertimbangkan *trade-off* antara eksplorasi dan eksploitasi. Eksplorasi dilakukan untuk mengeksplorasi area yang belum diuji, sedangkan eksploitasi dilakukan untuk memperoleh sampel di daerah yang menjanjikan berdasarkan distribusi posterior. Metode *Bayesian optimization* mencapai keseimbangan antara eksplorasi dan eksploitasi guna mendeteksi daerah yang paling mungkin optimal saat ini dan menghindari kehilangan konfigurasi yang lebih baik di daerah yang belum dijelajahi [30].

9. Evaluasi

Pada perbandingan kinerja beberapa algoritma, diperlukan suatu standar yang sama agar dapat menentukan algoritma terbaik dari perbandingan tersebut [31]. Metode evaluasi klasifikasi yang dapat diterapkan adalah dengan *confusion matrix*. *Confusion matrix* adalah tabel yang menyajikan hasil klasifikasi antara jumlah data uji yang benar dan jumlah data uji yang salah dari setiap kelas atau kategori [32]. Tabel *confusion matrix* ditunjukkan pada tabel [33] :

Tabel 1. *Confusion Matrix*

		Kelas Aktual	
		Positif	Negatif
Kelas Prediksi	Positif	TP	FP
	Negatif	FN	TN

Berdasarkan tabel *confusion matrix*, dapat diketahui nilai *accuracy*, *precision*, *recall*, dan nilai F-1 *score*. *Accuracy* merupakan ukuran yang menggambarkan sejauh mana sistem klasifikasi dapat mengklasifikasikan data dengan benar. Nilai *precision* memberikan gambaran tentang seberapa baik sistem dapat menghindari kesalahan

dalam mengklasifikasikan data negatif sebagai positif. *Recall* memberikan gambaran tentang seberapa baik sistem dalam menemukan semua data positif yang ada. F-1 *Score* memberikan ukuran yang seimbang antara *precision* dan *recall*. Nilai *accuracy*, *precision*, *recall*, dan nilai F-1 *score* dapat diperoleh dengan persamaan berikut [34].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$F1\ Score = \frac{2\ x\ Precision\ x\ Recall}{Precision\ +\ Recall} \quad (10)$$

Keterangan :
TP (*True Positive*) = jumlah data dalam kategori positif dan diklasifikasikan dengan benar sebagai positif.
TN (*True Negative*) = jumlah data dalam kategori negatif dan diklasifikasikan dengan benar sebagai negatif.
FN (*False Negative*) = jumlah data dalam kategori positif, tetapi diklasifikasikan sebagai kategori negatif.
FP (*False Positive*) = jumlah data dalam kategori negatif, tetapi diklasifikasikan sebagai kategori positif.

C. Hasil dan Pembahasan

1. Dataset

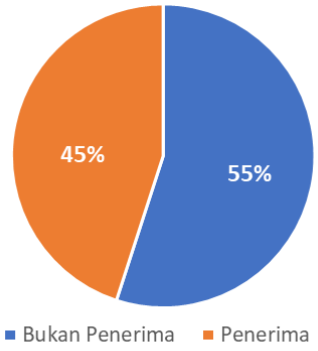
Kelas yang digunakan dalam penelitian ini adalah status keluarga penerima bantuan pangan beras. Kelas ini digunakan sebagai label untuk membagi data menjadi dua set label, yaitu 0 dan 1. Nilai 0 menunjukkan bahwa keluarga tersebut tidak menerima bantuan pangan beras, sedangkan nilai 1 menunjukkan bahwa keluarga tersebut menerima bantuan pangan beras. Variabel yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 2. Variabel Penelitian

No	Kode IFLS	Variabel	Keterangan
1	KSR24a	Status penerima bantuan pangan beras (Y)	0 = Tidak menerima 1 = Menerima
2	AR16	Pendidikan tertinggi kepala keluarga (X)	0 = Tidak sekolah 1 = Setara SD 2 = Setara SMP 3 = Setara SMA 4 = Diploma 5 = S-1 6 = S-2 7 = S-3

3	KR03	Kepemilikan rumah (X)	0 = Milik sendiri 1 = Menempati 2 = Kontrak 3 = Lainnya
4	KR11	Penggunaan listrik (X)	0 = Tidak menggunakan listrik 1 = Menggunakan listrik
5	KR13	Sumber air minum utama (X)	0 = Air kemasan 1 = Ledeng atau pipa 2 = Sumur 3 = Mata air 4 = Air hujan 5 = Sungai 6 = Kolam 7 = Bak penampungan 8 = Lainnya
6	KR20	Pembuangan air besar (X)	0 = Jamban sendiri 1 = Jamban bersama atau umum 2 = Sungai atau selokan 3 = Kebun atau sawah 4 = Kolam atau empang 5 = Kandang ternak 6 = Laut atau danau 7 = Lainnya
7	KR24	Bahan bakar untuk memasak (X)	0 = Listrik 1 = Gas elpiji 2 = Minyak tanah 3 = Kayu atau arang 4 = Lainnya
8	KR26	Kepemilikan asuransi (X)	0 = Tidak memiliki asuransi 1 = Memiliki asuransi
9	KRK06	Jumlah ruangan rumah (X)	*Continuous*
10	KRK08	Jenis lantai rumah (X)	0 = Keramik, marmer, ubin, granit, tegel, atau teraso 1 = Semen atau bata merah 2 = Kayu atau bambu 3 = Tanah 4 = Lainnya
11	KRK09	Jenis dinding luar rumah (X)	0 = Tembok 1 = Kayu, papan, atau triplek 2 = Bambu 3 = Lainnya
12	KRK10	Jenis atap rumah (X)	0 = Genteng 1 = Kayu 2 = Seng 3 = Asbes 4 = Daun 5 = Lainnya

2. Preprocessing



Gambar 3. Proporsi Keluarga Penerima Bantuan Pangan Beras

Dataset awal dalam penelitian ini sebesar 15.133 data dengan 12 variabel. Setelah dilakukan *preprocessing*, terdapat sebesar 11.175 data dengan 12 variabel yang siap diproses untuk membangun model penentuan keluarga penerima bantuan pangan beras. Berdasarkan gambar 3, terdapat 55% atau 6186 keluarga yang tidak terdaftar sebagai KPM sehingga termasuk ke dalam kelompok yang tidak menerima bantuan pangan beras. Sementara itu, terdapat 45% atau 4989 keluarga yang terdaftar sebagai KPM sehingga termasuk ke dalam kelompok penerima bantuan pangan beras.

3. Pemodelan

Berikut ini adalah tabel yang menunjukkan hasil dari *hyperparameter tuning* yang digunakan dalam algoritma KNN, *Naïve Bayes, Decision Tree*, dan SVM:

Tabel 3. *Hyperparameter Tuning* KNN

No	Metode	*leaf_size*	*n_neighbors*	*p*
1	*Grid search*	7	23	1
2	*Random search*	8	25	1
3	*Bayesian optimization*	1	29	1

Tabel 4. *Hyperparameter Tuning Naïve Bayes*

No	Metode	*ar_smoothing*
1	*Grid search*	0.12328467394420659
2	*Random search*	0.6990754548165822
3	*Bayesian optimization*	0.1873817422860384

Tabel 5. *Hyperparameter Tuning Decision Tree*

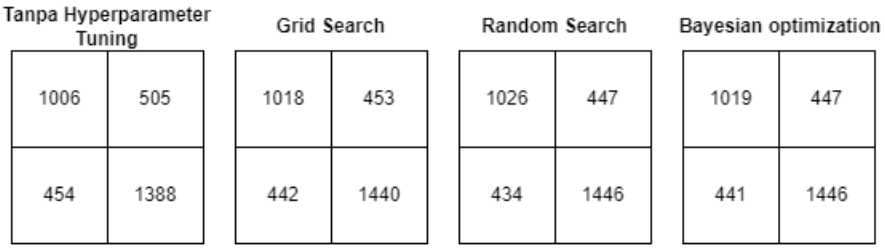
No	Metode	*Ccp Alpha*	*Criterion*	*Max Depth*	*Max Features*	*Min Samples Split*
1	*Grid search*	0.001	*entropy*	7	log2	-
2	*Random search*	-	*entropy*	17	log2	2
3	*Bayesian optimization*	-	*gini*	7	sqrt	18

Tabel 6. *Hyperparameter Tuning* SVM

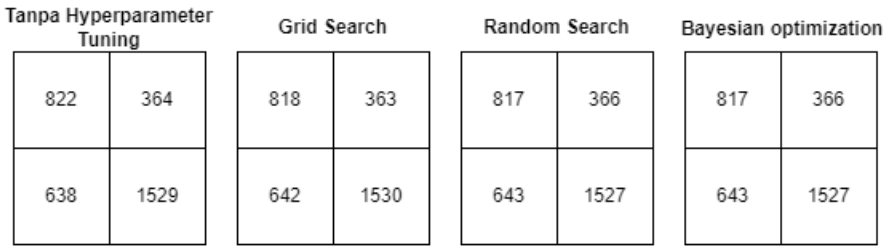
No	Metode	*gamma*	*kernel*	*C*
1	*Grid search*	0.2	rbf	1
2	*Random search*	0.2	rbf	10
3	*Bayesian optimization*	0.2	rbf	1

4. Evaluasi

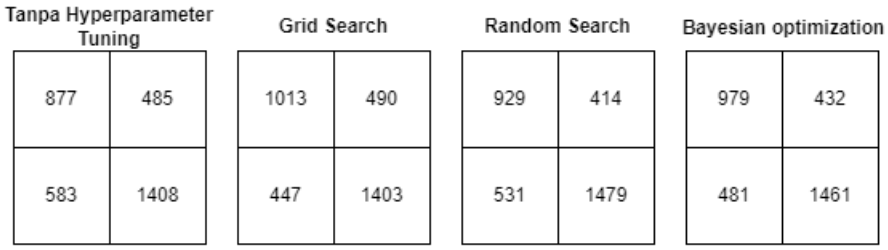
Setelah mengembangkan model klasifikasi menggunakan empat algoritma yang berbeda, langkah selanjutnya adalah melakukan evaluasi antara keempat model tersebut. Gambar 4. merupakan hasil dari evaluasi dengan *confusion matrix*.



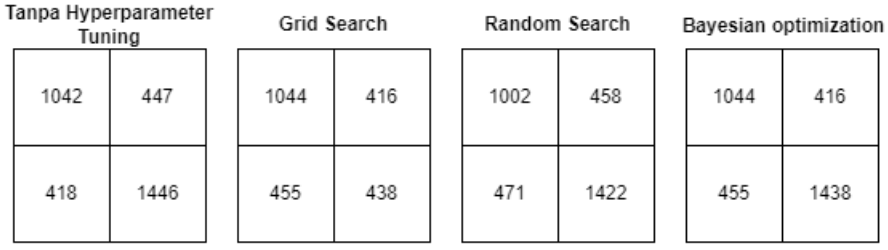
Gambar 4. Hasil *Confusion Matrix* KNN



Gambar 5. Hasil *Confusion Matrix Naïve Bayes*



Gambar 6. Hasil *Confusion Matrix Decision Tree*



Gambar 7. Hasil *Confusion Matrix* SVM

Dalam tabel yang diberikan, terdapat evaluasi performa algoritma SVM, KNN, *Naïve Bayes, dan Decision Tree* dengan atau tanpa penerapan *hyperparameter tuning*. Evaluasi performa ini diukur dengan menggunakan beberapa metrik, yaitu

accuracy, *precision*, *recall*, dan nilai F-1 *score*. Selain itu, juga diberikan nilai *train set score dan test set score*.

Tabel 7. Perbandingan Hasil Algoritma

No	Algoritma	*Hyperparameter Tuning*	*Accuracy*	*Recall*	*Precision*	*F1Score*	*Train Set Score*	*Test Set Score*
1	KNN	Tanpa *tuning*	0,71	0,68	0,66	0,67	0,78	0,71
		Grid search	0,73	0,70	0,69	0,69	0,75	0,73
		Random search	0,74	0,70	0,70	0,70	0,74	0,74
		Bayesian optimization	0,74	0,70	0,70	0,70	0,74	0,74
2	*Naïve Bayes*	Tanpa *tuning*	0,70	0,56	0,69	0,62	0,68	0,70
		Grid search	0,70	0,56	0,69	0,62	0,68	0,70
		Random search	0,70	0,56	0,69	0,62	0,68	0,70
		Bayesian optimization	0,70	0,56	0,69	0,62	0,68	0,70
3	*Decision Tree*	Tanpa *tuning*	0,68	0,68	0,64	0,62	0,89	0,68
		Grid search	0,71	0,71	0,68	0,66	0,72	0,71
		Random search	0,72	0,72	0,69	0,66	0,74	0,72
		Bayesian optimization	0,73	0,73	0,69	0,68	0,73	0,73
4	SVM	Tanpa *tuning*	0,74	0,71	0,70	0,71	0,75	0,74
		Grid search	0,74	0,72	0,70	0,71	0,77	0,74
		Random search	0,72	0,69	0,68	0,68	0,81	0,72
		Bayesian optimization	0,74	0,72	0,70	0,71	0,77	0,74

Berdasarkan hasil tabel, dapat dilihat perbandingan performa antara keempat algoritma yang digunakan dalam penelitian ini serta pengaruh *hyperparameter tuning* terhadap performa klasifikasi. Dari data tersebut, dapat diketahui bahwa algoritma KNN memiliki model yang paling baik dengan metode *random search* dan *Bayesian optimization*. *Accuracy* tertinggi sebesar 0,74 atau 74% yang berarti bahwa model KNN mampu mengklasifikasikan data dengan tingkat kebenaran sebesar 74%. Selanjutnya, nilai *recall* sebesar 0,7 menunjukkan bahwa model KNN berhasil mengidentifikasi sekitar 70% dari keseluruhan data positif yang tersedia. Lebih lanjut, nilai *precision* sebesar 0,7 menunjukkan bahwa sekitar 70% dari prediksi positif yang dibuat oleh model KNN benar-benar merupakan data positif. Nilai *train set score* dan nilai *test set score* yang sama, yakni sebesar 0,74 yang menandakan bahwa KNN memiliki kemampuan yang cukup stabil dalam generalisasi.

Algoritma *Naïve Bayes* memiliki a*ccuracy* yang sama baik dengan *hyperparameter tuning* maupun tanpa *hyperparameter tuning*, yaitu sebesar 70%. *Accuracy* ini menunjukkan bahwa model *Naïve Bayes* mampu melakukan klasifikasi dengan tingkat kebenaran sebesar 70%. Selanjutnya, nilai *recall* sebesar 0,56 menunjukkan bahwa model *Naïve Bayes* berhasil mengidentifikasi sekitar 56% dari total data positif yang tersedia. Kemudian, nilai *precision* sebesar 0,69 menunjukkan bahwa sekitar 69% dari prediksi positif yang dibuat oleh model *Naïve Bayes* benar-benar merupakan data positif. Nilai *train set score* sebesar 0,68 dan nilai *test set score* sebesar 0,70 yang menandakan bahwa *Naïve Bayes* memiliki kemampuan yang cukup stabil dalam generalisasi.

Algoritma *Decision Tree* menunjukkan kinerja model yang lebih baik dengan penerapan *hyperparameter tuning*. *Accuracy* tertinggi *Decision Tree* diperoleh dengan menggunakan metode *Bayesian optimization*, yaitu sebesar 0,73 atau 73%. Nilai ini mengindikasikan bahwa model *Decision Tree* mampu melakukan klasifikasi dengan tingkat kebenaran sebesar 73%. Selanjutnya, nilai *recall* sebesar 0,73

menunjukkan bahwa model *Decision Tree* berhasil mengidentifikasi sekitar 73% dari total data positif yang tersedia. Lebih lanjut, nilai *precision* sebesar 0,69 menunjukkan bahwa sekitar 69% dari prediksi positif yang dibuat oleh model *Decision Tree* benar-benar merupakan data positif. Nilai *train set score* dan nilai *test set score* yang sama, yakni sebesar 0,73 yang menandakan bahwa *Decision Tree* memiliki kemampuan yang cukup stabil dalam generalisasi.

Algoritma SVM memiliki hasil yang hampir mirip baik dengan *hyperparameter tuning* maupun tanpa *hyperparameter tuning*. Namun, model SVM yang paling optimal berdasarkan evaluasi tersebut adalah dengan metode *grid search* dan *Bayesian optimization*. Dengan metode tersebut, algoritma SVM memiliki *accuracy* tertinggi sebesar 0,74 atau 74%. Hasil ini menggambarkan kemampuan SVM dalam mengklasifikasikan data dengan tingkat kebenaran 74%. Nilai *accuracy* yang dicapai dalam penelitian ini lebih tinggi dibandingkan dengan penelitian serupa yang dilakukan oleh Iman dkk (2021) mengenai klasifikasi rumah tangga penerima Raskin/Rastra di Provinsi Jawa Barat Tahun 2017 dengan Metode *Random Forest* dan *Support Vector Machine* [27]. Selanjutnya, nilai *recall* sebesar 0,72 mengindikasikan bahwa model SVM mampu mengidentifikasi sekitar 72% dari keseluruhan data positif yang ada. Kemudian, nilai *precision* sebesar 0,7 menunjukkan bahwa sekitar 70% dari prediksi positif yang dibuat oleh model SVM benar-benar merupakan data positif. Selain itu, kedua nilai *train set score* dan *test set score* berada pada rentang 0,77 – 0,74 menunjukkan konsistensi model SVM dalam menghasilkan performa yang baik saat dilatih dan diuji pada data yang belum pernah dilihat sebelumnya. Keberadaan nilai yang seimbang ini menunjukkan bahwa model SVM memiliki kemampuan yang stabil dalam generalisasi.

D. Simpulan

Melalui implementasi algoritma klasifikasi menggunakan pendekatan *supervised learning* disertai dengan *hyperparameter tuning* pada keluarga penerima bantuan pangan beras, dapat disimpulkan bahwa penerapan *hyperparameter tuning* memiliki manfaat dalam meningkatkan kinerja algoritma KNN, *Decision Tree*, dan SVM. Algoritma tersebut menjadi lebih stabil dalam generalisasi serta dapat mengenali pola umum yang ada dalam data latihan dan menerapkannya pada data uji yang berbeda. Berdasarkan hasil evaluasi, dapat disimpulkan bahwa algoritma KNN dengan *random search* serta *Bayesian optimization* dan SVM dengan *Bayesian optimization* memberikan nilai *accuracy* yang sama, yakni sebesar 74%. Oleh karena itu, model tersebut memiliki kinerja yang setara dan sama baiknya dalam mengklasifikasikan keluarga penerima bantuan pangan beras.

E. Referensi

[1] Badan Pusat Statistik, "Persentase Penduduk Miskin (P0) Menurut Daerah 2021-2022," 2023, Available : https://www.bps.go.id/indicator/23/184/1/persentase-penduduk-miskin-p0-menurut-daerah.html. [Accessed 10 Juni 2023]

[2] Badan Pusat Statistik, "Jumlah Penduduk Miskin (Ribu Jiwa) Menurut Provinsi dan Daerah 2011-2012," *2013*, Available : https://www.bps.go.id/indicator/23/185/6/jumlah-penduduk-miskin-ribu-jiwa-menurut-provinsi-dan-daerah.html. [Accessed 10 Juni 2023]

[3] M. Hanafi & S. Hartini, "Metode AHP pada Seleksi Penerima Raskin Desa Sukakerta, Sukawangi Bekasi," *Junal Teknika*, vol. 15, No.01, pp. 63-69. 2021.
[4] Badan Urusan Logistik (Bulog), "Sekilas RASTRA / RASKIN," *2023*, Available : https://www.bulog.co.id/beraspangan/rastra/sekilas-raskin-beras-untuk-rakyat-miskin. [Accessed 10 Juni 2023]
[5] Dewan Perwakilan Rakyat Republik Indonesia (DPR RI), "Raskin Berubah Menjadi Rastra," *2015,* Available : https://www.dpr.go.id/berita/detail/id/11143. [Accessed 10 Juni 2023]
[6] JawaPos.com, "Ini Kata Darmin yang Membedakan BPNT dan Bansos Rastra," *2018*, Available : https://www.jawapos.com/ekonomi/0110130/ini-kata-darmin-yang-membedakan-bpnt-dan-bansos-rastra [Accessed 10 Juni 2023]
[7] M. Parida and M. Merina, "SISTEM PENDUKUNG PENGAMBILAN KEPUTUSAN SELEKSI PENERIMAAN BERAS (RASKIN) MENGGUNAKAN METODE AHP Studi Kasus: Kelurahan Tanjung Harapan Kotabumi Lampung Utara," *Jurnal Informasi Dan Komputer*, Oct. 2019, doi: 10.35959/jik.v7i2.134.
[8] Kementerian Koordinator Bidang Pembangunan Manusia dan Kebudayaan Republik Indonesia, *Pedoman Umum Bantuan Sosial Beras Sejahtera*. 2018.
[9] E. Wijoyono, "The utilization of village-information system for integrated social welfare data management: actor-network theory approach in Gunungkidul regency," *Jurnal Teknosains: Jurnal Ilmiah Sains Dan Teknologi*, vol. 11, no. 1, p. 13, Dec. 2021, doi: 10.22146/teknosains.60798.
[10] C. Vercellis, *Business Intelligence: Data Mining and Optimization for Decision Making*. Wiley, 2009.
[11] N. Frastian, S. A. Hendrian, & V. H. Valentino, "Komparasi Algoritma Klasifikasi Menentukan Kelulusan Mata Kuliah Pada Universitas," *Factor Exacta: Membangun Kreativitas Insan Ilmiah*, Mar. 2018, doi: 10.30998/faktorexacta.v11i1.1826.
[12] P. Laskov, P. Düssel, C. Schäfer, & K. Rieck, "Learning Intrusion Detection: Supervised or Unsupervised?," in *Lecture Notes in Computer Science*, Springer Science+Business Media, 2005, pp. 50–57. doi: 10.1007/11553595_6.
[13] P. C. Sen, M. Hajra, & M. Ghosh, "Supervised Classification Algorithms in Machine Learning: A Survey and Review," in *Advances in intelligent systems and computing*, Springer Nature, 2020, pp. 99–111. doi: 10.1007/978-981-13-7403-6_11.
[14] Springer, *Automated Machine Learning*. Springer Cham, 2019. doi: 10.1007/978-3-030-05318-5.
[15] L. A. Andika, P. N. Azizah, & R. Respatiwulan, "Analisis Sentimen Masyarakat terhadap Hasil Quick Count Pemilihan Presiden Indonesia 2019 pada Media Sosial Twitter Menggunakan Metode Naive Bayes Classifier," *Indonesian Journal of Applied Statistics*, Jul. 2019, doi: 10.13057/ijas.v2i1.29998.
[16] A. Ambarwari, Q. J. Adrian, & Y. Herdiyeni, "Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman," *Jurnal Rekayasa Sistem dan Teknol ogi Informasi*, vol. 1, no. 3, pp. 117–122, 2019.
[17] J. Strauss, F. Witoelar, & B. Sikoki, *The Fifth Wave of the Indonesia Family Life Survey: Overview and Field Report: Volume 1*. 2016. doi: 10.7249/wr1143.1.

[18]D. Cahyanti, A. N. Rahmayani, & S. A. Husniar, "Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara," *Indonesian Journal of Data and Science*, vol. 1, no. 2, pp. 39–43, Jul. 2020, doi: 10.33096/ijodas.v1i2.13.

[19] Z. Arifin, W. J. Shudiq, & S. Maghfiroh, "PENERAPAN METODE KNN (K-NEAREST NEIGHBOR) DALAM SISTEM PENDUKUNG KEPUTUSAN PENERIMAAN KIP (KARTU INDONESIA PINTAR) DI DESA PANDEAN BERBASIS WEB DAN MYSQL," *NJCA (Nusantara Journal of Computers and Its Applications)*, vol. 4, no. 1, Jul. 2019, doi: 10.36564/njca.v4i1.101.

[20] R. Indra Borman & M. Wati, "Penerapan Data Maining Dalam Klasifikasi Data Anggota Kopdit Sejahtera Bandarlampung Dengan Algoritma Naïve Bayes," *Jurnal Ilmiah Fakultas Ilmu Komputer*, vol. 09, no. 01, 2020.

[21] H. Annur, "Klasifikasi Masyarakat Miskin Menggunakan metode naive Bayes," ILKOM Jurnal Ilmiah, vol. 10, no. 2, pp. 160–165, 2018. doi:10.33096/ilkom.v10i2.303.160-165

[22]M. A. Handoko & N. Neneng, "SISTEM PAKAR DIAGNOSA PENYAKIT SELAMA KEHAMILAN MENGGUNAKAN METODE NAIVE BAYES BERBASIS WEB," *Jurnal Teknologi Dan Sistem Informasi*, vol. 2, no. 1, pp. 50–58, Mar. 2021, doi: 10.33365/jtsi.v2i1.739.

[23]M. Bansal, A. Goyal, & A. Choudhary, "A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning," *Decision Analytics Journal*, vol. 3, p. 100071, Jun. 2022, doi: 10.1016/j.dajour.2022.100071.

[24]I. Sutoyo, "IMPLEMENTASI ALGORITMA DECISION TREE UNTUK KLASIFIKASI DATA PESERTA DIDIK," *Jurnal Pilar Nusa Mandiri*, vol. 14, no. 2, p. 217, Sep. 2018, doi: 10.33480/pilar.v14i2.926.

[25]D. A. Larasati & T. Sutrisno, "Tourism site recommendation in Jakarta using decision tree method based on web review," SSRN Electronic Journal, 2018. doi:10.2139/ssrn.3268964

[26] M. Muhathir, M. H. Santoso, & D. A. Larasati, "Wayang Image Classification Using SVM Method and GLCM Feature Extraction," *JITE (Journal of Informatics and Telecommunication Engineering)*, vol. 4, no. 2, pp. 373–382, Jan. 2021, doi: 10.31289/jite.v4i2.4524.

[27] Q. Iman & A. W. Wijayanto, "Klasifikasi Rumah Tangga Penerima Beras Miskin (Raskin)/Beras Sejahtera (Rastra) di Provinsi Jawa Barat Tahun 2017 dengan Metode Random Forest dan Support Vector Machine," *Jurnal Sistem Dan Teknologi Informasi (Justin)*, vol. 9, no. 2, p. 178, Apr. 2021, doi: 10.26418/justin.v9i2.44137.

[28] T. Yu and H. Zhu, "Hyper-Parameter Optimization: A Review of Algorithms and Applications," *Arxiv*, pp. 1–54, Mar. 2020, doi: 10.48550/arXiv.2003.05689.

[29] R. Andonie, "Hyperparameter optimization in learning systems," *Journal of Membrane Computing*, vol. 1, no. 4, pp. 279–291, Oct. 2019, doi: 10.1007/s41965-019-00023-0.

[30] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.

[31] Y. I. Kurniawan, "Perbandingan Algoritma Naive Bayes dan C.45 dalam Klasifikasi Data Mining," *Jurnal Teknologi Informasi Dan Ilmu Komputer*, Oct. 2018, doi: 10.25126/jtiik.201854803.

[32] D. Normawati & S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, vol. 5, no. 2, pp. 697–711, Sep. 2021, doi: 10.30645/j-sakti.v5i2.369.

[33] B. P. Pratiwi, A. S. Handayani, & S. Sarjana, "PENGUKURAN KINERJA SISTEM KUALITAS UDARA DENGAN TEKNOLOGI WSN MENGGUNAKAN CONFUSION MATRIX," *Jurnal Informatika UPGRIS*, vol. 6, no. 2, Jan. 2021, doi: 10.26877/jiu.v6i2.6552.

[34] T. N. Wijaya, R. Indriati, & M. N. Muzaki, "ANALISIS SENTIMEN OPINI PUBLIK TENTANG UNDANG-UNDANG CIPTA KERJA PADA TWITTER," *Jambura Journal of Electrical and Electronics Engineering*, vol. 3, no. 2, pp. 78–83, Jul. 2021, doi: 10.37905/jjeee.v3i2.10885.