

# Optimasi Algoritma *XGBoost Classifier* Menggunakan *Hyperparameter Gridsearch* dan *Random Search* Pada Klasifikasi Penyakit Diabetes

Ginanjar Abdurrahman\*, Hardian Oktavianto \*\*, Mukti Sintawati \*\*\*

\* \*\* Universitas Muhammadiyah Jember , \*\*\* Universitas Ahmad Dahlan

\*abdurrahmanganjar@unmuhjember.ac.id, \*\*hardian@unmuhjember.ac.id , \*\*\*mukti.sintawati@pgsd.uad.ac.id

---

## ABSTRACT

Classification using *XGBoost* in this study was applied to diabetes data originating from the UCI Machine Learning website. The initial step in this research is to deal with missing values. Missing value is found in several features. These missing values need to be handled otherwise the *XGBoost* algorithm will not work. Missing value handling is done by adding a meaningful value as a substitute for the missing value. At the time of modeling, the dataset is divided into training data and test data. The training data used is 80% of the number of patients, while the test data is 20%. In this study, the dataset that had imputed missing values was subjected to three treatments, first without hyperparameters, secondly hyperparameter tuning using gridsearch, and third hyperparameter tuning using random search. In the first treatment, classification using *XGBoost* without hyperparameters obtained a negative log loss value of 25%, which means that the performance accuracy of the algorithm reaches 75%. As for the second treatment and the third treatment, namely by using gridsearch and random search, it produces the same negative log loss value, which is 5%, which means that the performance of the algorithm reaches 95%. Thus, the performance of gridsearch and random search can significantly increase the accuracy value

---

**Keyword:** *XGBoost*, Hyperparameter, Tuning, Gridsearch, Random Search

---

## 1. Pendahuluan

American Diabetes Association (ADA) [1] menyatakan bahwa setiap 21 detik muncul seorang pasien diabetes baru. Menurut [2] diabetes tidak hanya menyebabkan kematian premature di seluruh dunia, tetapi juga menjadi penyebab utama kebutaan, penyakit jantung, dan gagal ginjal. Prevalensi kasus diabetes tipe 2 sebesar 85 – 95% (Bustan [3]. Pada tahun 2019, menurut [2] prevalensi diabetes pada rentang usia 20 – 79 tahun secara global mencapai 8,3%. Adapun sepuluh negara dengan jumlah penderita Diabetes tertinggi pada tahun 2019 menurut [2] adalah sebagai berikut: Cina dengan jumlah penderita sebanyak 116,4 juta, India sebanyak 77 juta, Amerika serikat sebanyak 31 juta, Pakistan sebanyak 19,4 juta, Brasil sebanyak 16,8 juta, Meksiko sebanyak 12,8 juta, Indonesia sebanyak 10,7 juta, Jerman sebanyak 9,5 juta, Mesir sebanyak 8,9 juta, dan Bangladesh sebanyak 8,4 juta.

Berdasarkan Risetdas tahun 2018 dalam [2] menyatakan bahwa hampir seluruh provinsi di Indonesia menunjukkan peningkatan prevalensi pada tahun 2013 – 2018, kecuali provinsi Nusa Tenggara Timur. Empat provinsi dengan prevalensi tertinggi pada tahun 2013 dan 2018, yaitu Daerah Istimewa Yogyakarta (DIY), DKI Jakarta, Sulawesi Utara, dan Kalimantan Timur. Beberapa provinsi menunjukkan peningkatan prevalensi tertinggi hingga 0,9%, yakni Riau, DKI Jakarta, Banten, Gorontalo, dan Papua Barat. Data yang lain menunjukkan deskripsi diabetes menurut provinsi pada tahun 2018 bahwa provinsi Nusa Tenggara Timur memiliki prevalensi terendah, yaitu 0,9% diikuti oleh Maluku dan Papua sebesar 1,1%. Selengkapnya, prevalensi diabetes berdasarkan diagnosis dokter (%) menurut Riskedas tahun 2018 [2] disajikan dalam Gambar 3.

*Machine learning* merupakan teknik yang mengadaptasi cara mesin (komputer) dalam “belajar” dari data (*learn from data*)[4]. Python merupakan Bahasa pemrograman yang banyak digunakan oleh data *analyst*, data *scientist*, dan juga data *engineer* dalam implementasi *machine learning* [5]. Klasifikasi merupakan salah satu algoritma dalam machine learning, khususnya algoritma supervised learning untuk menemukan pola atau fungsi untuk mendeskripsikan dan memisahkan suatu kelas data dengan kelas data yang lainnya. Biasanya algoritma klasifikasi digunakan untuk prediksi data yang belum memiliki kelas data tertentu [6]. Konsep *boosting* berasal dari pertanyaan apakah “*weak learner*” bisa dijadikan “*better learner*” dengan melakukan modifikasi. Algoritma *XGBoost* (*Extreme Gradient Boosting*) merupakan salah satu algoritma machine learning berdasarkan pohon keputusan sebagai classifier dengan *Gradient Boosting* sebagai intinya. Perbedaan antara *Gradient Boosting* dengan *XGBoost*, tidak seperti *Gradient Boosting*, proses penambahan “*weak*

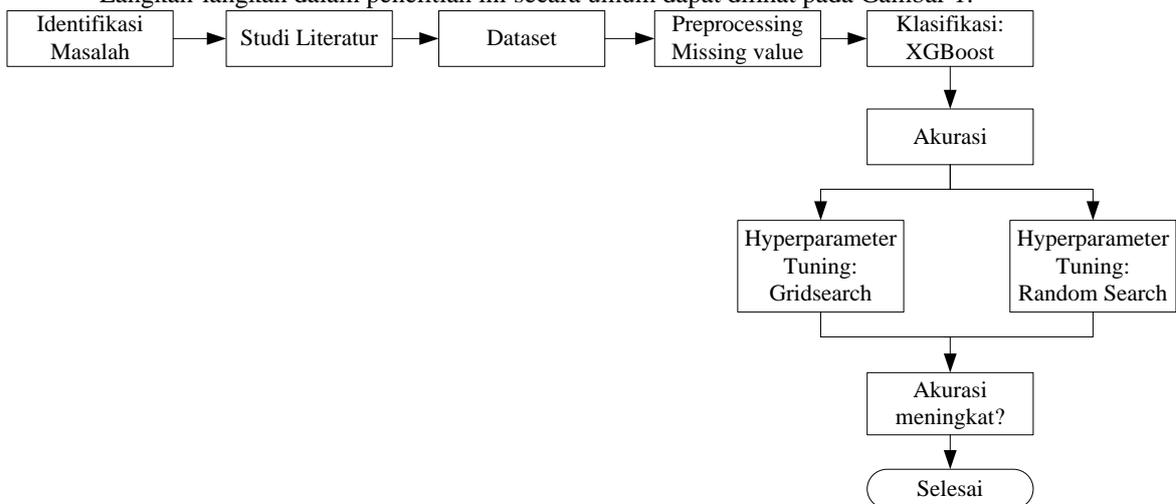
*learner*” pada *XGBoost* tidak terjadi secara berurutan, tetapi secara *multi-threaded*. Dalam hal ini, pemanfaatan inti CPU digunakan secara tepat (efisien), sehingga kecepatan dan kinerja algoritma akan lebih baik [7].

Pada umumnya, algoritma *hyperparameter* mengasumsikan observasi eksak atau mengabaikan ketidakpastian. Kebanyakan algoritma *hyperparameter* akan melakukan pengecekan terhadap pengaturan yang berbeda kemudian menyarankan hasil observasi terbaik. Namun demikian, jika observasi sangat berbeda antara *random seeds* konfigurasi terbaik yang diamati mungkin bukan konfigurasi yang menunjukkan performa terbaik jika dirata-rata diantara banyak *random seeds* [8]. Menurut Bergastra & Bangio dalam [8] Algoritma *random search* merupakan algoritma *hyperparameter* yang sederhana dan bebas-model. Menurut Li & Talwalker dalam [8] *Random search* itu sederhana tetapi mempunyai dasar yang kuat jika dibandingkan dengan algoritma yang lebih kompleks. Dalam penerapannya, ketika ada ketidakpastian, konfigurasi *hyperparameter* yang disarankan tidak berdampak. Namun demikian, pemilihan konfigurasi final yang disarankan mungkin terpengaruh. Dengan adanya ketidakpastian, konfigurasi terbaik yang diamati mungkin bukan konfigurasi terbaik rata-rata. Sedangkan algoritma *gridsearch* [9] akan melakukan pencarian nilai optimal pada semua titik yang ada dalam masalah regresi. Seluruh nilai alternatif pasangan dari *hyperparameter* akan dilakukan pengujian, dan nantinya pasangan yang memiliki nilai MSE minimum merupakan *hyperparameter* terbaiknya.

Pada penelitian ini akan dilakukan klasifikasi penyakit DM menggunakan algoritma *XGBoost Classifier* untuk menentukan apakah seseorang diklasifikasikan sebagai penderita diabetes atau bukan. *State of the art* dalam penelitian ini adalah *extreme gradient boosting (XGBoost)* sebagai model *ensemble*. *XGBoost* merupakan *novelty* dalam algoritma *gradient boosting* dan beberapa kali memenangkan kontes *Kaggle Machine Learning*. Hasil klasifikasi menggunakan *XGBoost*, selanjutnya dilakukan *hyperparameter* menggunakan *Gridsearch* dan *Randomsearch* secara terpisah dengan asumsi, *hyperparameter* dapat meningkatkan akurasi algoritma *XGBoost*. Dataset yang digunakan adalah dataset diabetes yang diambil dari repositori *UCI Machine Learning*. Dataset ini terdiri dari 8 variabel input, 1 variabel target, serta 768 record.

## 2. Metode Penelitian

Langkah-langkah dalam penelitian ini secara umum dapat dilihat pada Gambar 1.



Gambar 1. Langkah-langkah penelitian

### 2.1 Identifikasi masalah

Prevalensi penyakit diabetes melitus yang tinggi, khususnya di Indonesia menjadi perhatian khusus dalam dunia kesehatan. Penanganan yang kurang dalam menentukan penderita atau bukan penderita diabetes memungkinkan *data mining* khususnya untuk membantu dokter untuk mengklasifikasikan penyakit ini.

### 2.2 Studi Literatur

Penelitian yang telah dilakukan [9] yang berjudul *Hyperparameter Tuning Pada XGBoost Untuk Prediksi Keberlangsungan Hidup Pasien Gagal Jantung*, dilakukan untuk menentukan akurasi presisi serta performa *XGBoost* sebagai model prediksi. Penelitian ini menggunakan dataset *heart failure clinical records* yang merupakan *imbalanced dataset*. *Imbalanced dataset* ditangani dengan pendekatan level algoritma. Hal ini dilakukan agar model prediksi tidak cenderung menghasilkan prediksi kelas mayoritas. Hasil penelitian menunjukkan *XGBoost* dengan *tree parzen estimator hyperparameter tuning* lebih baik dari model lainnya

dengan nilai *AUC* 0,94. Nilai *AUC* untuk *XGBoost* dengan *random search* adalah 0,933, untuk *XGBoost* menggunakan *gridsearch* menghasilkan *AUC* sebesar 0,922, kemudian untuk *XGBoost* tanpa *hyperparameter* menghasilkan *AUC* sebesar 0,904.

Penelitian terkait *XGBoost* juga dilakukan oleh [10] dengan judul *Penerapan Metode Extreme Gradient Boosting (XGBoost) pada Klasifikasi Nasabah Kartu Kredit*. Penelitian ini melakukan klasifikasi nasabah kartu kredit yang macet dengan algoritma *XGBoost* dengan melibatkan *hyperparameter tuning grid search cross validation* yang selanjutnya dilakukan validasi menggunakan *10-Fold Cross Validation*. *Hyperparameter* yang dilakukan modifikasi diantaranya *n\_estimator*, *max\_depth*, *subsample*, *gamma*, *colsample\_bylevel*, *min\_child\_weight* serta *learning\_rate*. Performa model mengalami peningkatan setelah penggunaan *hyperparameter tuning*, dengan diperoleh nilai akurasi sebesar 80,039%, presisi sebesar 81,338%, serta *recall* sebesar 96,854%.

Penelitian yang dilakukan oleh [11] yang berjudul “*A XGBoost Risk Model, Via Feature Selection and Bayesian Hyper-Parameter Optimization*”. Dalam penelitian ini, pemodelan resiko ditransformasikan menjadi klasifikasi biner, yakni diberikan pinjaman jika peminjam beresiko rendah, dan tidak diberikan pinjaman jika mempunyai resiko tinggi. Penelitian ini bertujuan untuk membandingkan kinerja dua metode *hyperparameter*, yakni *Bayesian Tree-structured Parzen Estimator (TPE)* dengan *Random Search (RS)* dalam optimasi *XGBoost*. Performa *XGBoost* dengan *hyperparameter TPE (XGB\_TPE)* dan *RS (XGB\_RS)* dievaluasi setelah masing-masing diimplementasikan *feature selection*. Dalam penelitian ini, digunakan lima metode *Feature Selection* yakni *gini index*, bobot menggunakan *chi-square*, *hierarchical variable clustering*, bobot dengan korelasi, dan bobot dengan *information gain ratio*. Hasil dari *feature selection* pada *XGB\_TPE* menunjukkan bahwa metode *Chi-square* memperoleh nilai akurasi, *AUC*, *Recall*, dan *F1-score* tertinggi, sedangkan *Gini* menunjukkan hasil terburuk dengan nilai terendah pada keempat matriks performa tersebut. Performa *XGB\_RS* setelah menerapkan kelima *feature selection* tersebut menunjukkan hasil yang sangat mirip dengan *XGB\_TPE*.

Penelitian dengan judul “*Hyperparameter Optimization in XGBoost for Insurance Claim Prediction*” [12] bertujuan untuk memprediksi klaim asuransi menggunakan *XGBoost* dengan *hyperparameter gridsearch dan Bayesian search*. Penelitian ini menggunakan dua dataset, yakni dataset *Porto Seguro* dan *Allstate*, Pada dataset *Allstate* dilakukan preprocessing berupa konversi data kategorikal menjadi data numerik sehingga machine learning dapat bekerja dengan baik, sedangkan pada dataset *Porto Seguro* dilakukan preprocessing berupa imputasi dengan *imputer* modus. Pada penelitian ini, kinerja algoritma *XGBoost* pada masing-masing dataset dievaluasi menggunakan *mean absolute error (MAE)* dengan lima kali perulangan yang nantinya diperoleh rata-rata dari lima perulangan tersebut. Nilai rata-rata *MAE* pada dataset *Allstate* dari *XGBoost* menggunakan optimasi *gridsearch* diperoleh 1151.3756 sedangkan rata-rata *MAE* untuk *XGBoost* menggunakan optimasi *Bayesian search* diperoleh 1153.4370. Sedangkan nilai rata-rata akurasi dari dataset *Porto Seguro* untuk *XGBoost* dengan optimasi *gridsearch* sebesar 0,9999 sedangkan *XGBoost* menggunakan optimasi *Bayesian* diperoleh nilai akurasi sebesar 0,9938.

Penelitian berjudul *XGBoost dengan Random Search Hyper-parameter Tuning untuk Klasifikasi Situs Phishing* [13]. Penelitian ini bertujuan untuk mendeteksi antara website asli dengan website palsu (*phishing*) menggunakan algoritma *XGBoost* dengan *hyperparameter tuning random search*. Dataset yang digunakan berasal dari *UCI Machine Learning Repository* dengan 11.055 instance yang terdiri dari 30 fitur kategorikal dan dua kelas keputusan, yakni 6.157 kelas bukan *phishing*, dan 4.898 kelas *phishing*. Pembagian dataset dilakukan dengan 80% untuk training set, dan 20% untuk tes set. Hasil penelitian menunjukkan bahwa klasifikasi situs *phishing* dengan *XGBoost* tanpa *hyperparameter* memperoleh akurasi 95.34%, *recall* 97.78%, dan presisi sebesar 95.34%. Sedangkan klasifikasi situs *phishing* dengan *XGBoost* menggunakan *hyperparameter* memperoleh nilai akurasi 97.69%, *recall* 96.33%, dan presisi 98.44%.

Penelitian selanjutnya berjudul “*Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme-Gradient Boosting to Predict Chronic Kidney Failure*” [14]. Penelitian ini memanfaatkan algoritma *XGBoost* untuk prediksi Gagal Ginjal Kronis. Dalam dataset yang digunakan, terdapat beberapa fitur yang mengandung *missing value*, sehingga perlu dilakukan preprocessing data terlebih dahulu. Data juga tidak seimbang, sehingga dilakukan oversampling dengan *SMOTE* dan *random oversampling* untuk menyeimbangkan datanya. Data juga perlu dinormalisasi menggunakan dua metode, yakni normalisasi *min-max* dan *z-score*. Pembagian train-test dataset dilakukan dengan 70% data training dan 30% data testing. Hasil prediksi dengan *XGBoost* menggunakan *gridsearch* menghasilkan nilai akurasi 99.28% dan

$f$ -measure 0.9942 sedangkan hasil prediksi dengan *XGBoost* menggunakan *random search* menghasilkan nilai akurasi 99.33% dan  $f$ -measure 1.0.

Penelitian selanjutnya berjudul “Implementasi *XGBoost* Pada Keseimbangan *Liver Patient Dataset* dengan *SMOTE* dan *Hyperparameter Tuning Bayesian Search*” [15]. Penelitian ini bertujuan untuk klasifikasi seorang pasien merupakan penderita liver atau bukan. Dalam dataset yang digunakan terdapat *imbalanced data* antara kelas penderita liver dengan yang bukan. Untuk mengatasi *imbalanced data* digunakan metode *SMOTE*. Disamping itu, hyperparameter tuning Bayesian search juga digunakan untuk meningkatkan performa model. Model *XGBoost* menghasilkan nilai AUC 0.618, untuk model *XGBoost* dengan Bayesian search menghasilkan AUC 0,658, untuk model *XGBoost* dengan *SMOTE* menghasilkan AUC 0.716, sedangkan model *XGBoost* dengan *Smote* dan Bayesian Search menghasilkan AUC 0,767.

### 2.3 Dataset

Dataset yang digunakan adalah dataset Pima Indian Diabetes yang diperoleh dari UCI Machine Learning. Dataset ini berasal dari National Institute of Diabetes and Kidney Diseases. Semua pasien adalah Wanita dengan usia minimal 21 tahun dan merupakan keturunan India Pima. Dataset yang digunakan terdiri dari 8 variabel input, yakni: Kehamilan, Glukosa, Tekanan Darah, Ketebalan kulit, Insulin, BMI, Fungsi silsilah diabetes, dan umur. Selain itu, juga ada 1 variabel target, yang terdiri dari dua kelas keputusan: Penderita (1), Bukan Penderita (0). Dataset ini terdiri dari 768 record data. Pada dataset ini juga terdapat *missing value*, ditandai dengan nilai nol pada variabel input, yang nantinya akan ditangani dengan Teknik imputing nilai mean. Adapun potongan dataset ditampilkan pada Gambar 2.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
..	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

[768 rows x 9 columns]

Gambar 2. Dataset Penelitian

### 2.4 *XGBoost Classifier*

Algoritma *XGBoost Classifier* pada penelitian ini digunakan pada klasifikasi yang nantinya diimplementasikan dalam Bahasa pemrograman Python 3. Python 3 merupakan Bahasa pemrograman yang sudah umum digunakan dalam data mining dan machine learning. Pada python 3 sudah banyak library-library untuk klasifikasi, clustering, association rule bahkan untuk preprocessing dan seleksi fitur.

### 2.5 Akurasi

Kinerja algoritma klasifikasi pada penelitian ini nantinya akan diukur menggunakan nilai akurasi. Pada python 3 sudah tersedia library untuk mengukur akurasi sebagai ukuran kinerja algoritma klasifikasi. Setelah nilai akurasi diperoleh dari hasil algoritma *XGBoost Classifier*, nantinya dilakukan optimasi menggunakan hyperparameter tuning. *Hyperparameter* tuning yang digunakan adalah *Gridsearch* dan *Random search*, yang nantinya akan diperoleh nilai akurasi *XGBoost* dari setiap *hyperparameter* yang dilakukan.

## 3. Hasil dan Analisis

Klasifikasi menggunakan *XGBoost* pada penelitian ini diterapkan pada dataset diabetes yang berasal dari website *UCI Machine Learning*. Langkah awal pada penelitian ini adalah sebagai berikut:

### 3.1 Penanganan *missing value*

*Missing value* terdapat pada beberapa fitur, yang bisa diidentifikasi menggunakan fungsi *isna()* dan fungsi *isna.sum()*. Fungsi *isna()* digunakan untuk mengidentifikasi letak *missing value*, sedangkan fungsi *isna.sum()* digunakan untuk melihat banyaknya *missing value* dari setiap variable. Adapun tampilan dari fungsi *isna()* dan fungsi *isna.sum()* dapat dilihat pada Gambar 3 dan Gambar 4 berikut.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	False	False	False	False	True	False	False	False
1	False	False	False	False	True	False	False	False
2	False	False	False	True	True	False	False	False
3	False	False	False	False	False	False	False	False
4	True	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
763	False	False	False	False	False	False	False	False
764	False	False	False	False	True	False	False	False
765	False	False	False	False	False	False	False	False
766	False	False	False	True	True	False	False	False
767	False	False	False	False	True	False	False	False

Gambar 3. Tampilan dari pemanggilan fungsi *isna()*

```

Pregnancies      111
Glucose           5
BloodPressure    35
SkinThickness    227
Insulin          374
BMI              11
DiabetesPedigreeFunction  0
Age              0
dtype: int64

```

Gambar 4. Tampilan dari pemanggilan fungsi *isna.sum()*

Dari Gambar 3 dan Gambar 4 dapat diinterpretasikan sebagai berikut: dari Gambar 3 nilai *False* berarti bukan *missing value*, nilai *True* berarti merupakan *missing value*. Adapun interpretasi dari Gambar 4 adalah banyaknya *missing value* dari setiap fitur dalam penelitian ini. Nilai *missing value* ini perlu ditangani, jika tidak algoritma *XGBoost* tidak akan bekerja. Penanganan *missing value* yang dilakukan adalah dengan *imputing* nilai *mean* sebagai pengganti *missing value*.

### 3.2 Klasifikasi menggunakan *XGBoost* tanpa *hyperparameter*

Pada saat pemodelan, dataset dibagi menjadi data latih dan data uji. Data latih yang digunakan adalah 80% dari jumlah pasien, sedangkan data uji sebanyak 20%. Pada penelitian ini, dataset yang telah dilakukan *imputing missing value*. Pada klasifikasi menggunakan *XGBoost* tanpa *hyperparameter* diperoleh nilai *negative log loss* sebesar 25% yang berarti performansi akurasi algoritma mencapai 75%.

### 3.3 Klasifikasi menggunakan *XGBoost* dengan *hyperparameter*

Dengan menggunakan dataset yang sama dengan yang digunakan pada klasifikasi tanpa *hyperparameter*, dilakukan *hyperparameter tuning* menggunakan *gridsearch*, dan *hyperparameter tuning* menggunakan *random search*. Hasil klasifikasi dengan menggunakan *gridsearch* dan *random search* menghasilkan nilai *negative log loss* yang sama, yakni 5%, yang artinya performansi algoritma mencapai 95%. Dengan demikian, performansi *gridsearch* dan *random search* dapat meningkatkan nilai akurasi secara signifikan.

## 4. Kesimpulan

Berdasarkan hasil penelitian, kesimpulan yang dapat diperoleh adalah hasil klasifikasi menggunakan *XGBoost* menggunakan dataset diabetes tanpa menggunakan *hyperparameter tuning* diperoleh nilai akurasi model yang cukup baik yakni sebesar 75%. Sedangkan dengan menggunakan *hyperparameter tuning*, baik *gridsearch* maupun *random search* menghasilkan nilai akurasi model sama, yakni sebesar 95%. Namun demikian, waktu eksekusi program untuk *gridsearch* relatif lebih lama, hal ini dikarenakan *gridsearch* mencari

seluruh kombinasi dan memilih kombinasi terbaik berdasarkan *CV Score* yang paling tinggi, oleh karena itu *gridsearch* sering dinamakan dengan algoritma *exhaustive search*.

### Referensi

- [1] R. Yosmar, D. Almasdy, and F. Rahma, "Jurnal Sains Farmasi Dan Klinis," *Survei risiko penyakit diabetes melitus terhadap Kesehatan. Masy. kota padang*, vol. 5, no. Agustus 2018, pp. 134–141, 2018.
- [2] Kementerian kesehatan republik indonesia, "Tetap Produktif, Cegah Dan Atasi Diabetes Mellitus," *pusat data dan informasi kementerian kesehatan RI*. 2020.
- [3] D. W. Hestiana, "Journal of Health Education," *J. Heal. Educ.*, vol. 25, no. 1, pp. 57–60, 2017, doi: 10.1080/10556699.1994.10603001.
- [4] A. Lukman and Marwana, "Machine Learning Multi Klasifikasi Citra Digital," *Konf. Nas. Ilmu Komput.*, no. December 2014, pp. 1–6, 2014.
- [5] S. S. Purwadhika, "Apa Itu Python dan Fungsinya di Dunia Nyata?," 2019. <https://medium.com/purwadhikacconnect/apa-itu-python-dan-fungsinya-di-dunia-nyata-d5b533117c63>.
- [6] P. Bimo, N. Setio, D. Retno, S. Saputro, and B. Winarno, "Klasifikasi dengan Pohon Keputusan Berbasis Algoritme C4.5," *Prism. Pros. Semin. Nas. Mat.*, vol. 3, pp. 64–71, 2020.
- [7] R. Santhanam, N. Uzir, S. Raman, and S. Banerjee, "Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets," *Int. J. Control Theory Appl.*, vol. 9, no. March, pp. 651–662, 2017.
- [8] L. Hertel, P. Baldi, and D. L. Gillen, "Quantity vs. Quality: On Hyperparameter Optimization for Deep Reinforcement Learning," 2020, [Online]. Available: <http://arxiv.org/abs/2007.14604>.
- [9] M. Rizky Mubarak, Muliadi, and R. Herteno, "Hyper-Parameter Tuning pada XGBoost Untuk Prediksi Keberlangsungan Hidup Pasien Gagal Jantung," *Kumpul. J. Ilmu Komput.*, vol. 9, no. 2, pp. 391–401, 2022.
- [10] S. E. Herni Yulianti, Oni Soesanto, and Yuana Sukmawaty, "Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit," *J. Math. Theory Appl.*, vol. 4, no. 1, pp. 21–26, 2022, doi: 10.31605/jomta.v4i1.1792.
- [11] Y. Wang, "A XGBoost Risk Model Via Feature Selection and Bayesian Hyperparameter optimization," vol. 11, no. 1, pp. 1–17, 2019.
- [12] I. Gede Pajar Bahari and H. Murfi, "Hyperparameter optimization in xg boost for insurance claim prediction," *J. Adv. Res. Dyn. Control Syst.*, vol. 12, no. 4 Special Issue, pp. 1510–1517, 2020, doi: 10.5373/JARDCS/V12SP4/20201630.
- [13] M. R. Afrizal, R. A. Nugroho, D. Kartini, R. Herteno, and K. Selatan, "XGBOOST DENGAN RANDOM SEARCH HYPER- PARAMETER TUNING UNTUK KLASIFIKASI SITUS," vol. 15, pp. 40–47, 1979.
- [14] D. A. Anggoro and S. S. Mukti, "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 6, pp. 198–207, 2021, doi: 10.22266/ijies2021.1231.19.
- [15] R. Ubaidillah, M. Muliadi, D. T. Nugrahadi, M. R. Faisal, and R. Herteno, "Implementasi XGBoost Pada Keseimbangan Liver Patient Dataset dengan SMOTE dan Hyperparameter Tuning Bayesian Search," *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1723, 2022, doi: 10.30865/mib.v6i3.4146.