

# Perbandingan Kinerja Algoritma Optimasi pada Metode Random Forest untuk Deteksi Kegagalan Jantung

Unang Sunarya dan Tita Haryanti  
Fakultas Ilmu Terapan, Universitas Telkom  
Jl. Telekomunikasi No.1, Terusan Buah Batu, Bandung, Jawa Barat 40257  
e-mail: unangsunarya@telkomuniversity.ac.id

**Abstrak**—Jantung merupakan salah satu organ terpenting dalam tubuh manusia. Kegagalan jantung pada pasien dapat mengakibatkan dampak yang vital dan berujung pada kematian. Adapun kegagalan jantung bukan hanya dipengaruhi oleh faktor usia, juga dipengaruhi komorbid dan pola hidup dari pasien. Berbagai upaya medis telah banyak dilakukan untuk mendeteksi kegagalan jantung yang mengharuskan pasien dirawat intensif di rumah sakit yang tentunya membuat pasien merasa kurang nyaman. Maka dari itu, dalam penelitian ini dirancang sebuah aplikasi *Machine Learning* untuk deteksi kegagalan jantung yang dapat mengklasifikasikan kondisi pasien ke arah kematian atau bertahan berdasarkan gejala-gejala yang dimiliki pasien. Adapun algoritma *Machine Learning* yang digunakan adalah *Random Forest* yang dioptimasi dengan tiga buah algoritma optimasi yaitu *Grid Search*, *Random Search* dan *Bayesian Search* sebagai perbandingan. Kinerja ketiga algoritma optimasi kemudian diukur menggunakan akurasi, presisi dan *recall*. Ada 299 sampel pasien yang digunakan dalam penelitian ini. Hasil menunjukkan bahwa *Random Forest* dengan algoritma optimasi *Random Search* mencapai kinerja yang paling unggul dengan akurasi rata-rata sebesar 85,63 %, presisi rata-rata 87,38% dan *recall* 85,63%.

**Kata kunci:** *kegagalan jantung, random forest, grid search, random search, bayesian search*

**Abstract**—The heart is one of the most important organs in the human body. Heart failure in patients can have a strong impact and lead to death. Heart failure is not only influenced by age but also influenced by comorbidities and the patient's lifestyle. Various medical efforts have been made to detect heart failure requiring patients to be treated intensively in the hospital, causing patient discomfort. Therefore, in this study, a Machine Learning application was designed to detect heart failure and classify the patient's condition towards death or survival based on the patient's symptoms. The Machine Learning algorithm, Random Forest, is optimized using three optimization algorithms, namely Grid Search, Random Search, and Bayesian Search, as a comparison. The performance of the three optimization algorithms is then measured using accuracy, precision, and recall. There are 299 patient samples used in this study. The results show that Random Forest with Random Search optimization algorithm achieves the most superior performance with an average accuracy of 85.63%, an average precision of 87.38%, and a recall of 85.63%.

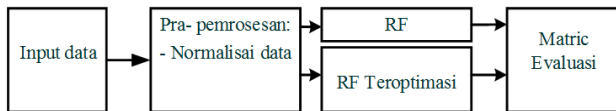
**Keywords:** *heart failure, random forest, grid search, random search, bayesian search*

## I. PENDAHULUAN

Jantung merupakan salah satu organ terpenting dalam tubuh manusia. Kegagalan jantung pada pasien dapat mengakibatkan dampak yang vital dan berujung pada kematian. Studi menunjukkan bahwa penyakit yang berhubungan dengan kegagalan jantung berada pada peringkat pertama yang dapat mengakibatkan kematian [1]. Deteksi dini kegagalan jantung dapat mengurangi risiko fatal pada pasien dengan gagal jantung. Selain itu dapat membantu meringankan petugas medis untuk melakukan perawatan yang lebih baik.

Berbagai penelitian menggunakan *Machine Learning* telah banyak dilakukan untuk memprediksi risiko fatal akibat kegagalan jantung, salah satunya dengan

menghitung peluang kegagalan jantung menggunakan metode pohon keputusan (*Decision Tree*) [2], [3]. Risiko fatal akibat kegagalan jantung juga dapat didiagnostik dari berbagai perangkat implan yang digunakan untuk mendiagnosis berbagai gejala yang berhubungan dengan kegagalan jantung, selanjutnya informasi diagnosa tersebut dikombinasikan dan diidentifikasi menggunakan algoritma *Bayesian Belief Network* (BBN) [4]. Arsitektur *Muti-Task Deep Wide Neural Network* (MT-DWNN) telah digunakan untuk memprediksi pengaruh komplikasi penyakit kegagalan jantung yang dapat membawa pasien ke dalam keadaan kritis [5]. Penelitian tersebut menemukan bahwa memburuknya fungsi ginjal sangat berpengaruh terhadap pasien dengan kegagalan jantung. Analisis kegagalan jantung dapat juga dilakukan dengan



Gambar 1. Diagram blok implementasi deteksi penyakit kegagalan jantung

menghitung perbedaan jarak setiap detak jantung yang berdekatan, *Heart Rate Variability* (HRV), secara terus-menerus dengan menghitung jarak dari titik puncak satu siklus sinyal jantung ke titik puncak dari siklus jantung selanjutnya [6], [7]. Selanjutnya dari HRV dapat ambil fitur yang dapat mewakili kondisi jantung pasien yang sedang didiagnosis. Selain algoritma *Machine Learning*, deteksi kronis kegagalan jantung pun telah dianalisis secara langsung tanpa proses *Feature Engineering* menggunakan metode *End-To-End Deep Learning* [8]. Namun hasil yang didapat dengan menggunakan algoritma ini masih belum maksimal karena ketidakpastian parameter-parameter yang digunakannya, apakah merupakan parameter yang optimal dari algoritma tersebut.

Berbagai upaya telah dilakukan untuk memaksimalkan kinerja algoritma baik pada *Machine Learning* ataupun *Deep Learning* dengan menggunakan algoritma optimasi. Namun kurangnya informasi tentang perbandingan algoritma-algoritma ini yang diuji pada algoritma *Machine Learning* dan dataset yang sama mengakibatkan perbandingan yang tidak ‘*Apple-Tt-Apple*’ sehingga belum dapat ditarik kesimpulan algoritma mana yang lebih optimal untuk digunakan. Oleh karena itu, pada penelitian ini dilakukan pengujian perbandingan algoritma optimasi pada metode *Machine Learning Random Forest* dengan kasus klasifikasi efek kronis kegagalan jantung. Latar belakang dilakukan pengujian perbandingan algoritma pada *Random Forest* (RF) adalah karena mengingat algoritma RF merupakan salah satu algoritma yang handal dalam kasus klasifikasi dan paling banyak digunakan dalam kasus analisis sinyal *physiological* [9], [10]. Sedangkan alasan pemilihan studi kasus dilakukan pada klasifikasi efek kronis kegagalan jantung adalah untuk memberi rekomendasi pemilihan algoritma optimasi yang paling optimal untuk penelitian dalam bidang biomedis yang memerlukan keakuratan analisis dalam mengkaji suatu permasalahan dalam bidang biomedis. Adapun kebaharuan pada penelitian ini adalah menawarkan solusi pemilihan algoritma optimasi yang paling optimal untuk menganalisis permasalahan dalam kasus analisis sinyal *physiological* yang sangat bermanfaat dalam dunia biomedis. Sehingga pada akhirnya dapat menjadi rujukan untuk penelitian-penelitian selanjutnya dalam mengoptimalkan hasil analisis.

## II. METODE

### A. Dataset

Pada penelitian ini studi kasus untuk evaluasi kinerja algoritma menggunakan dataset ‘*Heart Failure Clinical*

*Records Dataset*’ yang merupakan dataset publik dan dapat diunduh secara daring pada [11]. Dataset ini terdiri dari 12 kolom fitur yaitu: *age, anaemia, creatinine phosphokinase, diabetes, ejection fraction, high blood pressure, platelets, serum creatinine, serum sodium, sex, smoking* dan *time*. Berdasarkan fitur tersebut, dataset ini mengklasifikasikan kejadian kegagalan jantung ke dalam dua kelas apakah pasien meninggal selama tindak lanjut atau tidak.

### B. Diagram Blok Implementasi

Gambar 1 menunjukkan keseluruhan proses implementasi kalsifikasi kegagalan jantung. Proses implementasi dimulai dari mengakuisisi fitur data yang diikuti dengan pemrosesan awal data untuk meningkatkan kualitas dan mengurangi redundansi fitur data sebelum dilakukan pengklasifikasian, dalam hal ini digunakan metode normalisasi data. Setelah fitur data dinormalisasi, selanjutnya data tersebut diklasifikasikan dengan metode RF. Selain menggunakan RF, fitur data juga diklasifikasi menggunakan RF yang dioptimasi dengan metode optimasi *Grid Search* (RF-GS), *Random Search* (RF-RS) dan *Bayesian Search* (RF-BS). Pada tahap akhir masing-masing algoritma dievaluasi dengan menggunakan metrik evaluasi untuk menilai algoritma mana yang lebih unggul satu sama lainnya.

### C. Random Forest

*Random Forest* (RF) merupakan salah satu algoritma *Machine Learning* yang bekerja dengan mengombinasikan sejumlah algoritma *Decision Tree* dalam pengambilan keputusannya [12]. RF merupakan jenis *Machine Learning* ansambel yang dapat digunakan untuk masalah klasifikasi maupun regresi [10]. Selain kuat terhadap *overfitting*, RF juga memiliki kinerja yang lebih baik jika dibandingkan dengan beberapa *Machine Learning* lain seperti *Support Vector Machine* (SVM) dan *Discriminant Analysis* [9].

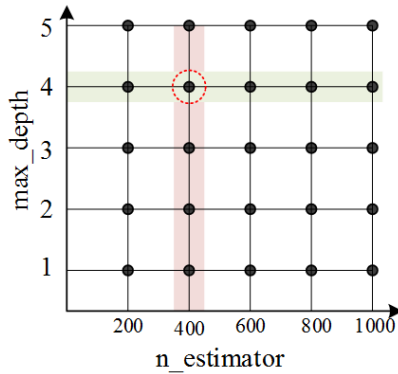
### D. Algoritma Optimasi

#### 1. Grid Search

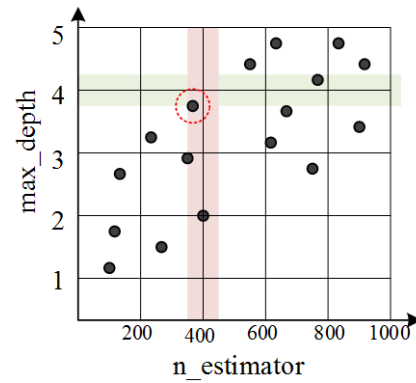
*Grid Search* (GS) adalah metode yang paling mudah untuk diimplementasikan, tetapi kekurangannya adalah tidak efisien ketika jumlah parameternya banyak [13].

Gambar 2 mengilustrasikan cara kerja algoritma GS dengan ciri khas setiap parameter diletakan pada *grid* di antara dua parameter yang berbeda. Nilai optimal parameter terletak di antara persilangan dua parameter tersebut [14]. Sumbu *x* dan *y* dalam hal ini masing-masing menyatakan parameter RF *n\_estimator* dan *max\_depth*.

GS bekerja dengan mencoba semua kombinasi dari *hyperparameter* yang sebelumnya didefinisikan dalam *sample space* sebagai himpunan parameter algoritma GS, kemudian dari *sample space* ini diambil *hyperparameter* yang terbaik berdasarkan hasil pemrosesan data [15]. Algoritma 1 menunjukkan *pseudocode* dari algoritma optimasi GS.



Gambar 2. Mekanisme kerja Grid Search



Gambar 3. Mekanisme kerja Random Search

Algoritma 1: Algoritma GS pada <i>Random Forest</i>	
1:	Inisialisasi model <i>Random Forest</i>
2:	Latih model awal <i>Random Forest</i> dengan data (fitur kegagalan jantung dan label)
3:	Definisikan <i>hyperparameter</i> dalam <i>sample space</i> : <i>'n_estimators'</i> : [100, 250, 500, 750, 1000], <i>'criterion'</i> : ['gini', 'entropy'], <i>'max_features'</i> : [None, 'auto', 'sqrt', 'log2'], <i>'max_depth'</i> : [1, 3, 5]
4:	Model GS ← buat model (model <i>Random Forest</i> , <i>sample space</i> )
5:	Latih model GS dengan data (fitur kegagalan jantung dan label)
6:	Prediksi ← model GS (data uji)
7:	Hitung matriks evaluasi: Akurasi ← model GS (label uji, prediksi) Presisi ← model GS (label uji, prediksi) Recall ← model GS (label uji, prediksi)
8:	Hasil akhir: akurasi, presisi, <i>recall</i>

2. Bayesian Search

*Bayesian Search* (BS) merupakan salah satu metode optimasi parameter yang sangat efisien dalam hal jumlah fungsi evaluasi [16]-[17]. Adapun efisiensi dari BS ini adalah kemampuannya untuk menggabungkan pengetahuan sebelumnya (*prior*) yang digunakan untuk memprediksi hasil berikutnya. Secara umum BS dapat dilihat pada Persamaan (1) sebagai berikut:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{1}$$

dimana  $P(A)$  merupakan *prior probability* dan  $P(B)$  masing-masing menunjukkan probabilitas dari suatu nilai variabel A dan B. Sedangkan  $P(A|B)$  merupakan probabilitas bersyarat, *posterior*, dari variabel A ketika diberikan nilai B, begitu juga sebaliknya  $P(B|A)$  merupakan probabilitas bersyarat *likelihood* dari variabel B jika diketahui A. Adapun Persamaan (1) dapat disederhanakan dengan membuang faktor normalisasi  $P(B)$ , menghasilkan formula seperti terlihat pada Persamaan (2) berikut:

$$P(A|B) = P(B|A)P(A). \tag{2}$$

3. *Random Search*

*Random Search* (RS) merupakan salah satu metode

optimasi *hyperparameter* yang banyak digunakan. Tidak seperti GS, RS bekerja dengan cara mencoba setiap kombinasi nilai parameter secara acak [14]. Pada GS setiap nilai parameter yang telah ditetapkan dicoba satu persatu selama proses optimasi berlangsung, sedangkan pada RS dicari nilai yang optimal sepanjang nilai parameter yang telah ditentukan dengan jelas antara batas bawah dan batas atasnya [18].

Seiring dengan perkembangannya, RS sendiri telah banyak mengalami perubahan yang semula hanya dapat mencari permasalahan dalam *local minimum* kini telah dapat mencari permasalahan dalam *global minimum* melalui modifikasi panjang variabel pencarian dalam algoritmanya [19].

Gambar 3 menunjukkan mekanisme kerja RS, di mana nilai parameter RF yaitu  $n\_estimator$  dan  $max\_depth$  telah didefinisikan sebelumnya kemudian secara acak diambil sebagai parameter sampai didapatkan nilai parameter optimal yang merupakan bagian nilai koordinat yang paling mendekati persilangan antara nilai parameter  $n\_estimator$  dan  $max\_depth$ . Adapun mekanisme kerja RS dapat dilihat pada Algoritma 2 di mana  $n\_estimator$  didefinisikan sebagai sebuah nilai *integer* dengan batas bawah 100 dan batas atas 1000 begitu juga dengan  $max\_depth$  yang merupakan bilangan *integer* antara 1 sampai dengan 5.

Algoritma 2: Algoritma RS pada <i>Random Forest</i>	
1:	Inisialisasi model <i>Random Forest</i>
2:	Latih model awal <i>Random Forest</i> dengan data (fitur kegagalan jantung dan label)
3:	Definisikan <i>hyperparameter</i> dalam <i>sample space</i> : <i>'n_estimators'</i> : int [100 -1000], <i>'criterion'</i> : ['gini', 'entropy'], <i>'max_features'</i> : [None, 'auto', 'sqrt', 'log2'], <i>'max_depth'</i> : int[1 s.d.5]
4:	Model GS ← buat model (model <i>Random Forest</i> , <i>sample space</i> )
5:	Latih model GS dengan data (fitur kegagalan jantung dan label)
6:	Prediksi ← model GS (data uji)
7:	Hitung matriks evaluasi: Akurasi ← model GS (label uji, prediksi) Presisi ← model GS (label uji, prediksi) Recall ← model GS (label uji, prediksi)
8:	Hasil akhir: akurasi, presisi, <i>recall</i>

### E. Metode Evaluasi

Adapun kinerja masing-masing algoritma optimasi pada RF dalam penelitian ini dievaluasi dengan menggunakan nilai akurasi, presisi, *recall*, F1 score dan dianalisis menggunakan *Confussion Matrix*, *Area Under Curve* (AUC) serta waktu komputasi. Akurasi merupakan salah satu metrik evaluasi yang banyak digunakan untuk mengukur kinerja suatu algoritma. Metrik ini dirumuskan seperti pada Persamaan (3).

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FN + FP}, \quad (3)$$

dimana TP adalah *True Positive*, TN: *True Negative*, FN: *False Negative* dan FP adalah *False Positive*.

Selain akurasi, pada penelitian ini juga menggunakan metrik presisi dan *recall* seperti terlihat pada persamaan (4) dan (5).

$$\text{Presisi} = \frac{TP}{TP + FP}, \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (5)$$

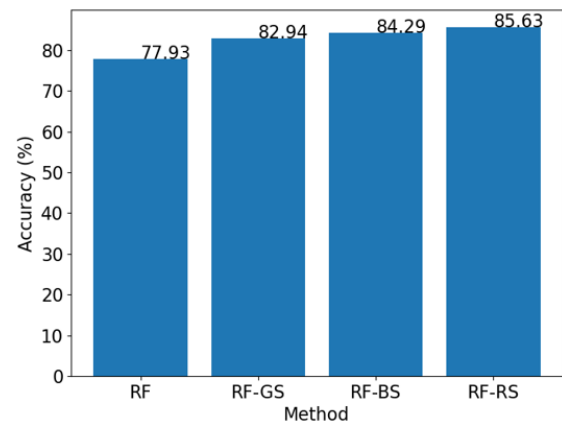
Untuk memverifikasi keandalan kinerja algoritma pada penelitian ini juga ditambahkan analisis F1 score pada setiap algoritma, pengecekan *Confussion Matrix* dan terakhir mengevaluasi setiap algoritma menggunakan pengukuran AUC.

## III. HASIL DAN PEMBAHASAN

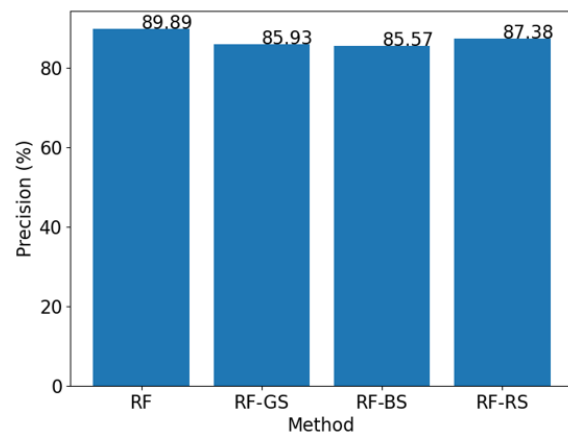
Pada bagian ini, kinerja masing-masing algoritma dibahas dan dianalisis berdasarkan metrik evaluasi untuk melihat algoritma mana yang memiliki kinerja lebih baik secara keseluruhan. Setiap kinerja algoritma dianalisis mulai dari metoda RF tanpa optimasi sampai pada algoritma RF dengan algoritma optimasi GS, BS dan RS.

### A. Kinerja Algoritma Optimasi

Dalam penelitian ini masing-masing kinerja algoritma dievaluasi menggunakan metrik akurasi, presisi dan *recall*. Gambar 4 menunjukkan perbandingan kinerja akurasi dari RF (77,93%) dan RF yang dioptimasi menggunakan algoritma optimasi. Hasil menunjukkan bahwa algoritma RF yang dioptimasi dengan algoritma optimasi RS menunjukkan hasil yang lebih baik (85,63%) jika dibandingkan dengan dua algoritma optimasi lainnya, yaitu GS (82,94 %) dan BS (84,29%). Berdasarkan metrik ini dapat dianalisis bahwa algoritma RF dengan algoritma optimasi RS dapat memberikan kinerja akurasi yang lebih baik untuk mendeteksi penyakit kegagalan jantung berdasarkan fitur-fitur data yang mempengaruhi kegagalan jantung yang diberikan. Ada perbedaan jelas bisa dilihat dari perbandingan akurasi rata-rata dari setiap algoritma yang digunakan dan sangat jelas dapat dianalisis



Gambar 4. Perbandingan kinerja akurasi setiap algoritma

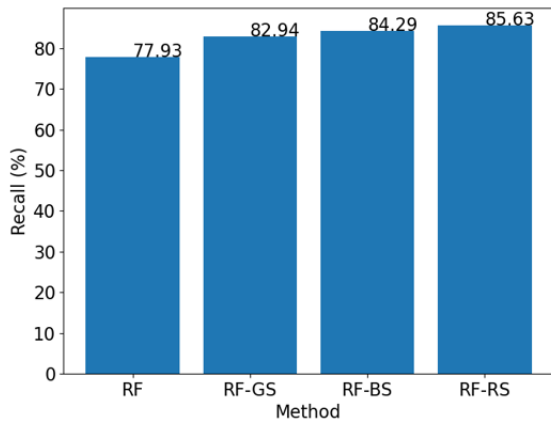


Gambar 5. Perbandingan kinerja presisi setiap algoritma

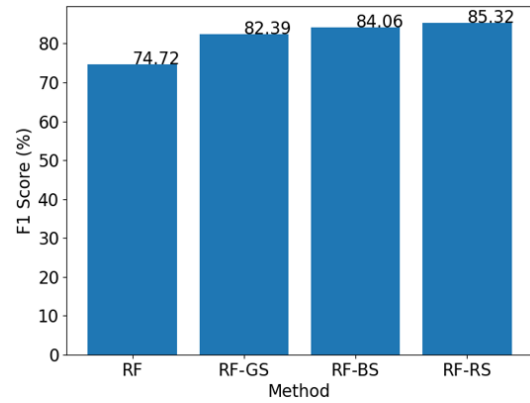
bahwa algoritma optimasi memberikan pengaruh kenaikan akurasi yang signifikan pada algoritma RF. Pada akhirnya setelah dioptimasi mencapai puncak akurasi rata-rata pada RF yang dioptimasi dengan RS.

Gambar 5 menunjukkan kinerja presisi rata-rata dari masing-masing algoritma yang digunakan untuk mengklasifikasi kegagalan jantung. Dalam hal kinerja presisi, algoritma RF tanpa optimasi mencapai nilai yang lebih tinggi (89,89 %) dibanding nilai kinerja pada algoritma dengan optimasi. Sementara nilai kinerja presisi lainnya secara berurutan dicapai oleh RF-RS (87,38%), RF-GS (85,93%) dan RF-BS (85,57 %). Meskipun algoritma RF tanpa optimasi menunjukkan nilai yang lebih tinggi dari algoritma yang lainnya hal ini dapat dianalisis karena perhitungan kinerja presisi cenderung fokus pada kebenaran yang dihitung pada satu kelas saja dan mengabaikan kebenaran pada kelas lainnya. Sehingga ketika perhitungan kinerja presisi dilakukan pada dataset yang jumlah kelasnya tidak seimbang '*imbalance dataset*' dapat mengakibatkan nilai kinerja presisinya lebih tinggi dibandingkan dengan algoritma lainnya, sekalipun nilai kinerja akurasinya rendah. Hal ini dapat dilihat lebih jauh pada Gambar 5.

Gambar 6 menunjukkan kinerja masing-masing algoritma dalam hal perhitungan kinerja *recall*. Dalam hal hasil kinerja *recall*, algoritma RF-RS menunjukkan dominasi kinerja dibandingkan dengan algoritma RF tanpa



Gambar 6. Perbandingan kinerja recall setiap algoritma



Gambar 7. Perbandingan kinerja F1 score setiap algoritma

optimasi maupun algoritma RF dengan optimasi lainnya. Kinerja *recall* pada RF-RS mencapai nilai puncak rata-rata pada 85,63 %. Ada selisih sebesar 1,34 % terhadap algoritma RF-BS pada posisi kedua. Tingginya nilai kinerja *recall* pada algoritma RF-RS dapat dianalisis pada tingginya nilai prediksi yang benar pada semua kelas yang menunjukkan bahwa algoritma RF-RS merupakan algoritma yang andal untuk perhitungan deteksi kegagalan jantung.

B. F1-Score

Untuk memverifikasi keandalan kinerja algoritma pada bagian III.A selanjutnya digunakan perhitungan *F1 score* sebagai *harmonic mean* dari nilai kinerja presisi dan *recall* [20]. Adapun untuk menghitung nilai *F1 score* dapat dilihat pada Persamaan (6),

$$F1\ score = \frac{2 \times Presisi \times Recall}{Presisi + Recall} \tag{6}$$

Gambar 7 menunjukkan hasil perhitungan nilai rata-rata dari *F1 score*. Di mana nilai rata-rata tertinggi *F1 score* diperoleh dengan menggunakan algoritma RF-RS sebesar 85,32 %. Hal ini sangat jelas menunjukkan bahwa algoritma RF-RS merupakan algoritma terbaik dalam kasus deteksi kegagalan jantung pada penelitian ini. Sementara algoritma RF-BS menempati urutan kedua dengan selisih 1,26 % dari RF-BS sebesar 84,06 % dan urutan ketiga RF-GS sebesar 82,39%. Sedangkan algoritma RF tanpa optimasi menunjukkan nilai rata-rata *F1 score* terendah dibandingkan dengan semua algoritma yang menggunakan skema optimasi sebesar 74,72%. Hal ini dapat dianalisis bahwa optimasi yang dilakukan terhadap RF telah berhasil memberikan kontribusi yang signifikan terhadap kinerja algoritma.

C. Confussion Matrix

Gambar 8 menunjukkan *Confussion Matrix* dari algoritma yang digunakan untuk klasifikasi kegagalan jantung. Dari setiap gambar tersebut dapat dianalisis hasil prediksi benar dari setiap kelas. Label ‘0’ mengindikasikan kelas di mana pasien tidak meninggal selama masa tindak

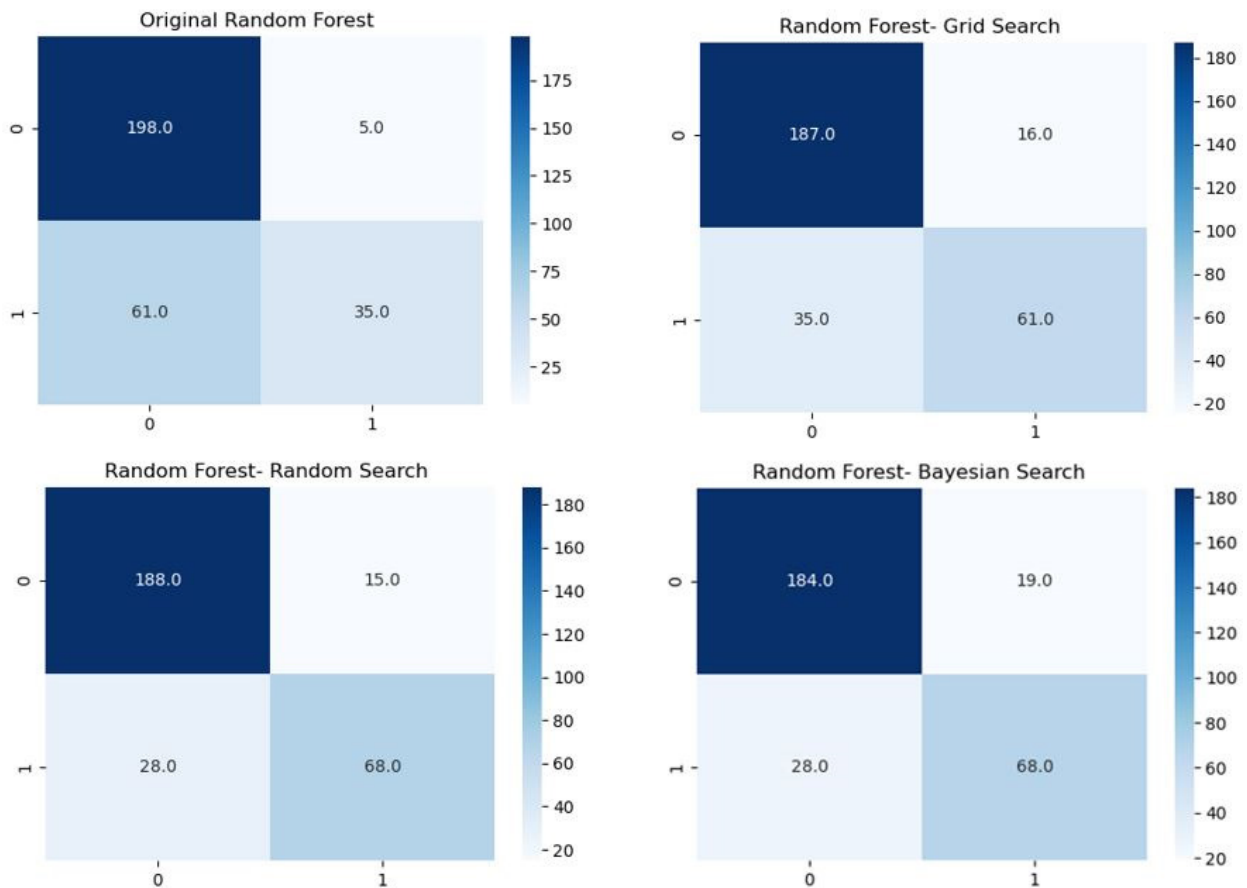
lanjut sedangkan label ‘1’ untuk pasien yang meninggal.

Pada algoritma RF tanpa optimasi sekalipun menunjukkan prediksi 198 (97,5%) pada kelas 0 namun persentase kesalahan pada kelas 1 menunjukkan angka yang lebih besar juga dibandingkan pada algoritma yang lain yaitu sebesar 61 (63,5%). Sehingga secara keseluruhan menunjukkan kinerja algoritma yang lebih rendah dibandingkan algoritma lainnya. Pada algoritma RF-GS mencapai angka prediksi benar pada 187 (92,1%) untuk kelas 0. Angka ini relatif lebih kecil jika dibandingkan dengan persentase pada algoritma RF tanpa optimasi. Namun, algoritma RF-GS memiliki kesalahan prediksi pada kelas 1 jauh lebih rendah jika dibandingkan dengan algoritma RF tanpa optimasi sebesar 35 (36,5%). Sedangkan pada algoritma RF-BS memiliki persentase prediksi benar relatif lebih kecil dari algoritma RF dan RF-BS namun memiliki kesalahan prediksi yang lebih kecil pada kelas 1 yaitu sebesar 28 (29.1 %). Sedangkan pada algoritma RF-RS mampu memprediksi dengan benar kelas 0 lebih baik dibanding dengan algoritma RF-BS dan RF-GS yaitu sebesar 188 (92,6%). Sekalipun nilai prediksi benar ini tidak setinggi nilai prediksi benar pada algoritma RF tanpa optimasi tapi algoritma RF-RS memiliki nilai kesalahan prediksi yang jauh lebih kecil jika dibandingkan dengan tingkat kesalahan pada algoritma RF tanpa optimasi yaitu sebesar 28 (29,1%).

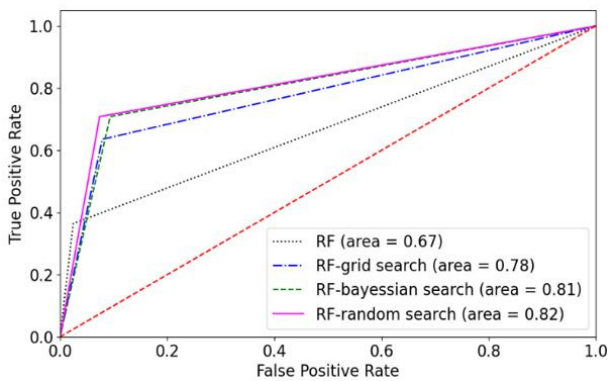
Nilai ini berbeda hampir setengah kali nilai kesalahan pada algoritma RF tanpa optimasi. Sehingga secara keseluruhan algoritma RF-RS menunjukkan kinerja yang lebih baik jika dibandingkan dengan algoritma RF tanpa optimasi maupun algoritma RF-GS dan RF-BS.

D. Area Under Curve

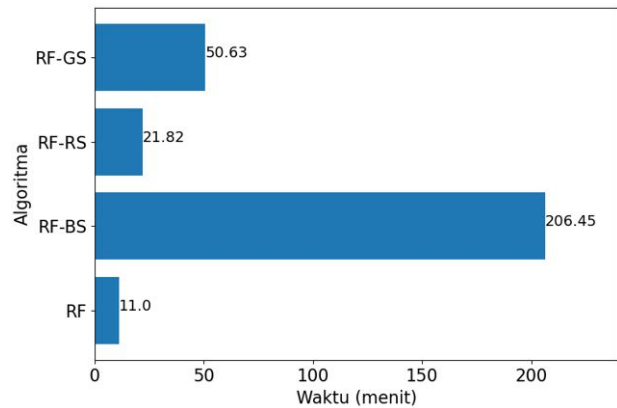
Gambar 9 menunjukkan perbandingan luas kurva, AUC, untuk setiap algoritma yang digunakan. Algoritma RF dengan optimasi RS menunjukkan hasil yang lebih besar (0.82) dibandingkan dengan algoritma RF dengan GS (0.78) dan BS (0.81). Sementara itu, algoritma RF tanpa optimasi menunjukkan nilai yang paling kecil yang berarti memiliki kinerja yang paling rendah dibandingkan dengan algoritma RF lainnya yang dioptimasi dengan algoritma optimasi sebesar 0.67.



Gambar 8. Perbandingan *confusion matrix* setiap algoritma



Gambar 9. Area *under curve* setiap algoritma



Gambar 10. Waktu komputasi algoritma yang diperlukan

Gambar 10 menunjukkan waktu yang diperlukan oleh setiap algoritma untuk memproses semua data sampai mengeluarkan hasil kinerjanya. Sangat terlihat dengan jelas bahwa algoritma RF-BS memerlukan waktu komputasi yang paling lama dibandingkan dengan algoritma lainnya sebesar 206,45 menit untuk menyelesaikan komputasinya.

Di sisi lain algoritma RF tanpa komputasi menunjukkan waktu komputasi yang sangat rendah (11 menit) dibandingkan algoritma lainnya. Hal ini dikarenakan pada algoritma RF tanpa optimasi dalam satu proses perhitungan hanya bergantung pada nilai *hyperparameter*

awal yang sudah ditentukan lebih awal tanpa adanya pilihan lain untuk setiap nilai parameternya. RF-RS membutuhkan waktu 21,82 menit untuk menyelesaikan perhitungan algoritmanya dan menempatkan pada posisi kedua waktu tercepat setelah algoritma RF tanpa optimasi. Sedangkan algoritma RF-GS berada pada posisi ketiga dalam perhitungan waktu komputasi sebesar 50,63 menit.

Dari sini dengan melihat *trade-off* antara kinerja algoritma dengan waktu komputasinya masih dapat dikatakan algoritma RF-RS masih lebih baik dibandingkan dengan algoritma lainnya.

#### IV. KESIMPULAN

Pada penelitian ini dilakukan deteksi kegagalan jantung dengan menggunakan algoritma RF tanpa optimasi dan RF yang dioptimasi dengan menggunakan GS, RS dan BS. Dari hasil implementasi ditunjukkan bahwa kinerja algoritma RF yang dioptimasi menggunakan RS menunjukkan kinerja yang terbaik dibandingkan dengan algoritma RF tanpa optimasi dan RF yang dioptimasi menggunakan GS dan BS yaitu dengan nilai kinerja akurasi rata-rata 85,63 %. Meskipun membutuhkan waktu komputasi yang lebih besar dari algoritma RF tanpa optimasi namun memiliki luas area di bawah kurva yang dominan dibandingkan algoritma lainnya yaitu sebesar 0,82. Dengan demikian dapat disimpulkan bahwa algoritma RF dengan optimasi RS mengungguli algoritma lainnya dalam implementasi deteksi kegagalan jantung ini.

#### REFERENSI

- [1] O. Indicators, *Health at a Glance 2017*. Paris: OECD Indicators OECD Publishing, 2017.
- [2] D. Chicco and G. Jurman, "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone," *BMC Med. Inform. Decis. Mak.*, vol. 20, no. 1, pp. 16, 2020.
- [3] A. J. Aljaaf, D. Al-Jumeily, A. J. Hussain, T. Dawson, P. Fergus, and M. Al-Jumaily, "Predicting the likelihood of heart failure with a multi level risk assessment using decision tree," in 2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE), 2015, pp. 101–106.
- [4] S. Sarkar and J. Koehler, "A Dynamic risk score to identify increased risk for heart failure decompensation," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 147–150, 2013.
- [5] B. Wang *et al.*, "A Multi-task neural network architecture for renal dysfunction prediction in heart failure patients with electronic health records," *IEEE Access*, vol. 7, pp. 178392–178400, 2019.
- [6] H. Wendt, P. Abry, K. Kiyono, J. Hayano, E. Watanabe, and Y. Yamamoto, "Wavelet p-leader non gaussian multiscale expansions for heart rate variability analysis in congestive heart failure patients," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 1, pp. 80–88, 2019.
- [7] P. Melillo, N. De Luca, M. Bracale, and L. Pecchia, "Classification tree for risk assessment in patients suffering from congestive heart failure via long-term heart rate variability," *IEEE J. Biomed. Heal. Informatics*, vol. 17, no. 3, pp. 727–733, 2013.
- [8] M. Gjoreski, A. Gradisek, B. Budna, M. Gams, and G. Poglajen, "Machine learning and end-to-end deep learning for the detection of chronic heart failure from heart sounds," *IEEE Access*, vol. 8, pp. 20313–20324, 2020.
- [9] A. Liaw and M. Wiener, "Classification and regression by random forest," *R news*, Vol.2, no.3, pp.18-22, 2002.
- [10] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *J. Comput. Graph. Stat.*, vol. 15, no. 1, pp. 118–138, 2006.
- [11] UC Irvine Machine Learning Repository. <https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records>.
- [12] C. Brüser, J. Diesel, M. D. H. Zink, S. Winter, P. Schauerte, and S. Leonhardt, "Automatic detection of atrial fibrillation in cardiac vibration signals," *IEEE J. Biomed. Heal. Informatics*, vol. 17, no. 1, pp. 162–171, 2013.
- [13] J. M. Dufour and J. Neves, "Conceptual econometrics using R" *Handbook of Statistics*, 41st ed, North Holland: Elsevier, 2019.
- [14] M. Swamynathan, *Mastering Machine Learning with Python in Six Steps*, 2nd ed. Bangalore, India: Apress Media, 2019.
- [15] L. Vega Ugi, F. Yosef Suratman, and U. Sunarya, "Electrocardiogram feature selection and performance improvement of sleep stages classification using grid search," *Bull. Electr. Eng. Informatics*, vol. 11, no. 4, pp. 2033–2043, 2022.
- [16] H. Bashashati, R. K. Ward, and A. Bashashati, "Bayesian optimization of BCI parameters," in 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2016, pp. 1–5.
- [17] M. Jiang and Y. Chen, "Research on bayesian optimization algorithm selection strategy," in The 2010 IEEE International Conference on Information and Automation, 2010, pp. 2424–2427.
- [18] A. Javeed, S. Zhou, L. Yongjian, I. Qasim, A. Noor, and R. Nour, "An intelligent learning system based on random search algorithm and optimized random forest model for improved heart disease detection," *IEEE Access*, vol. 7, pp. 180235–180243, 2019.
- [19] K. Hirasawa, K. Togo, J. Murata, M. Ohbayashi, N. Shao, and J. Hu, "A new random search method for neural networks learning-random search with variable search length (RasVal)," in Proceeding 1998 IEEE International Joint Conference on Neural Networks, 1998, pp. 1602–1607.
- [20] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, "Automatic sleep stage scoring with single-channel eeg using convolutional neural networks," *ArXiv*, vol. 1610.01683, pp. 1-12, 2016.