



Prediksi pembatalan pemesanan hotel menggunakan optimalisasi hiperparameter pada algoritme Random Forest

Prediction of hotel bookings cancellation using hyperparameter optimization on Random Forest algorithm

Yufis Azhar, Galang Aji Mahesa, Moch. Chamdani Mustaqim *)

Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Malang
Jl. Raya Tlogomas No.246, Malang, Jawa Timur, Indonesia 65144

Cara sitasi: Y. Azhar, G. A. Mahesa, and M. C. Mustaqim, "Prediksi pembatalan pemesanan hotel menggunakan optimalisasi hiperparameter pada algoritme Random Forest," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 1, pp. 15-21, 2021. doi: [10.14710/jtsiskom.2020.13790](https://doi.org/10.14710/jtsiskom.2020.13790), [Online].

Abstract – Cancellation of hotel bookings by customers greatly influences hotel managerial decision making. To minimize losses by this problem, the hotel management made a fairly rigid policy that could damage the reputation and business performance. Therefore, this study focuses on solving these problems using machine learning algorithms. To get the best model performance, hyperparameter optimization is applied to the random forest algorithm. It aims to obtain the best combination of model parameters in predicting hotel booking cancellations. The proposed model is proven to have the best performance with the highest accuracy results of 87 %. This study's results can be used as a model component in hotel managerial decision-making systems related to future bookings' cancellation.

Keywords – classification; hyperparameter optimization; random forest

Abstrak – Pembatalan pemesanan hotel oleh pelanggan sangat mempengaruhi pengambilan keputusan manajerial hotel. Untuk meminimalkan kerugian akibat masalah ini, pihak pengelola hotel membuat kebijakan yang cukup kaku yang dapat merusak reputasi dan kinerja bisnis. Oleh karena itu, penelitian ini berfokus pada penyelesaian masalah tersebut menggunakan algoritme pembelajaran mesin. Untuk mendapatkan performa model terbaik, optimalisasi hiperparameter diterapkan pada algoritme random forest. Hal tersebut bertujuan untuk mendapatkan kombinasi parameter model yang terbaik dalam memprediksi pembatalan pemesanan hotel. Model yang diusulkan terbukti memiliki kinerja terbaik dengan hasil akurasi tertinggi 87 %. Hasil dari penelitian ini dapat digunakan sebagai komponen model dalam sistem pengambilan keputusan manajerial hotel terkait pembatalan pemesanan di masa mendatang.

Kata kunci – klasifikasi; optimalisasi hiperparameter; random forest

I. PENDAHULUAN

Dalam industri perhotelan, travel dan pariwisata, pembatalan pesanan sangat mempengaruhi manajemen industri dalam mengambil keputusan manajerialnya. Untuk mengurangi kerugian akibat masalah tersebut, pihak pengelola sering kali menerapkan kebijakan pembatalan yang sangat kaku. Belum lagi ditambah dengan ketentuan biaya pembatalan pesanan yang harus dibayar oleh pelanggan. Hal tersebut dapat merusak reputasi dan kinerja bisnis hotel di masa mendatang.

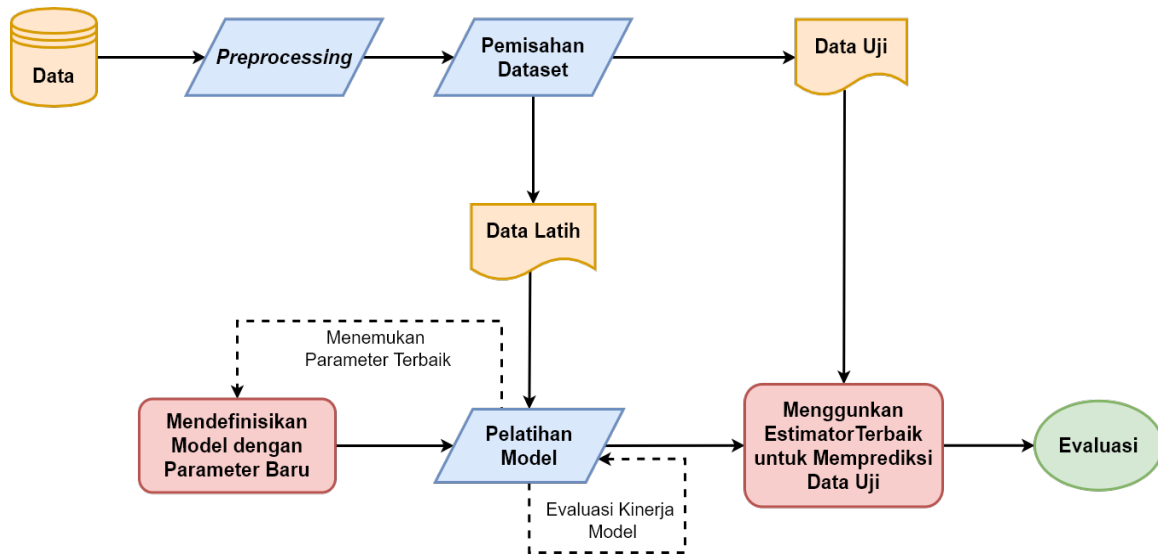
Kitamori dkk. [1] mengatakan bahwa para pemilik industri rata-rata ingin mengetahui keuntungan bisnisnya berdasarkan peramalan dari data yang telah mereka miliki. Namun, saat mencoba menerapkan metode tersebut seringkali ditemukan masalah pada variabel data yang rata-rata terlalu umum dan tidak spesifik.

Berbagai studi telah dilakukan untuk memecahkan permasalahan tersebut menggunakan berbagai macam pendekatan pembelajaran mesin. Antonio dkk. [2] telah mengembangkan sebuah model *xgboost* (XGB) untuk memprediksi pembatalan pesanan hotel dengan akurasi sebesar 84 %. Antonio dkk. [3] membandingkan beberapa algoritme pembelajaran mesin untuk studi kasus serupa dengan dataset yang berbeda dan menyatakan bahwa algoritme *Decision Forest* (DF) memiliki nilai akurasi terbaik.

Kajian yang telah ada menunjukkan model yang cukup baik untuk digunakan dalam sistem manajemen pendapatan hotel. Meskipun begitu, kedua algoritme tersebut masih memiliki kelemahan, seperti XGB yang memiliki kerentanan terhadap *overfit* apabila jumlah kombinasi *tree* yang digunakan tidak tepat [4]. Begitu pula dengan DF, yang perlu mendefinisikan jumlah *tree* yang tepat untuk dapat mengurangi efek ketidakstabilan dari algoritme dasarnya, yaitu *Decision Tree* (DT). Oleh sebab itu, untuk menangani permasalahan tersebut diperlukan suatu metode yang memiliki performa lebih stabil dan tahan terhadap kemungkinan *overfit*.

Kelemahan dari metode tersebut dapat ditangani oleh metode *Random Forest* (RF). Hal ini dikarenakan metode RF merupakan metode yang cocok untuk menangani kasus klasifikasi [5]. Selain itu, Rokach dkk.

*) Penulis korespondensi (Moch. Chamdani Mustaqim)
Email: chamdani@webmail.um.ac.id



Gambar 1. Implementasi metode OHRF

[4] menyatakan bahwa model RF merupakan sebuah model *ensemble* yang istimewa sebab memiliki *bagging* di dalamnya. Metode *bagging* ini membuat model RF menjadi lebih tahan terhadap kemungkinan *overfit* dan juga lebih stabil daripada algoritme DT.

Meskipun dalam praktiknya, untuk mendefinisikan model RF yang optimal merupakan sebuah tantangan yang cukup sulit untuk dilakukan. Hingga saat ini algoritme RF tidak memiliki komposisi parameter model yang pasti. Oo dkk. [6] mengatakan, bahwa masalah yang sering muncul ketika membangun model RF adalah penentuan parameter terbaik untuk memperoleh model yang lebih akurat. Oleh karena itu, perlu adanya metode yang tepat untuk mendapatkan komposisi parameter model yang optimal.

Sebuah studi menemukan bahwa teknik optimalisasi hiperparameter pada model pembelajaran mesin dapat digunakan sebagai alat untuk mendapatkan komposisi parameter model yang lebih optimal [7]–[9]. Menurut Bergstra dan Bengio [10], metode optimalisasi hiperparameter berbasis *random search algorithm* (RSA) adalah teknik yang efisien untuk meningkatkan performa model pembelajaran mesin.

Kajian [2], [3] menggunakan teknik prapemrosesan yang berbeda. Kedua teknik tersebut telah menghasilkan komposisi variabel data yang berbeda antara satu sama lain. Dalam kajian [3], kombinasi variabel data latihan masukan yang lebih spesifik diperoleh berkat metode seleksi fitur yang digunakan. Pada kajian [2], transformasi data dipilih dan dapat menghasilkan komposisi variabel yang lebih banyak dari [3].

Meskipun demikian, kedua teknik prapemrosesan yang digunakan masih menyisakan data derau yang dapat berpengaruh besar terhadap hasil klasifikasi model. Data derau yang dimaksud adalah daftar pengunjung hotel yang tidak memiliki jumlah pengunjung dewasa, anak-anak, dan bayi. Hal ini dapat menyebabkan *overfit* pada model pembelajaran mesin yang didefinisikan.

Penelitian ini menggunakan model pendekatan RF untuk menangani efek ketidakstabilan dari algoritme DT dan kemungkinan *overfit* algoritme XGB. Selain itu, teknik transformasi data dan penanganan data derau dipilih untuk menghasilkan data latihan masukan yang lebih kaya dan bersih. Metode optimalisasi hiperparameter berbasis RSA diaplikasikan pada algoritme RF untuk dapat mengoptimalkan kinerja model tradisional. Model yang diusulkan selanjutnya disebut sebagai *Optimized Hyperparameter on Random Forest algorithm* (OHRF).

II. METODE PENELITIAN

Gambar 1 menunjukkan implementasi metode OHRF yang digunakan secara keseluruhan. Pembahasan meliputi dataset, prapemrosesan, pemisahan dataset, pelatihan model, optimalisasi model, dan evaluasi.

A. Dataset

Penelitian ini menggunakan dua dataset pemesanan hotel yang berasal dari *City Hotel* (H1) dan *Resort Hotel* (H2) [11]. Dataset ini merupakan data pengamatan nyata dari dua hotel dengan klasifikasi bintang 4. Kedua dataset ini terbagi dalam struktur yang sama, dengan 31 variabel yang menggambarkan 40.060 pengamatan untuk H1 dan 79.330 pengamatan untuk H2. Masing-masing dataset mewakili pemesanan hotel yang tiba antara 1 Juli 2015 hingga 31 Agustus 2017 termasuk data pemesanan berhasil dan dibatalkan. Semua variabel data yang terkait dengan identitas pelanggan dan juga hotel telah dihapus untuk menjaga privasi data. Semua variabel dataset dapat dilihat pada Tabel 1.

Sebelum melakukan proses prapemrosesan data, dataset hotel yang digunakan perlu dipastikan bahwa tidak ada data *corrupt* atau hilang sebagian. Hal ini telah dipastikan oleh sistem manajemen properti hotel bahwa tidak ada data yang hilang dalam tabel basis datanya [11].

Tabel 1. Deskripsi variabel

Jenis Data	Variabel
Numerik	ADR
Bilangan bulat	Adults, ArrivalDateDayOfMonth, ArrivalDateWeekNumber, ArrivalDateYear, Babies, BookingChanges, Children, DaysInWaitingList, LeadTime, PreviousBookingsNotCanceled, PreviousCancellations, RequiredCardParkingSpaces, StaysInWeekendNights, StaysInWeekNights, TotalOfSpecialRequests
Kategoris	Agent, ArrivalDateMonth, AssignedRoomType, Company, Country, CustomerType, DepositType, DistributionChannel, IsCanceled, IsRepeatedGuest, MarketSegment, Meal, ReservationStatus, ReservedRoomType
Tanggal	ReservationStatusDate

B. Prapemrosesan

Dalam penelitian ini, prapemrosesan data diadopsi dari penelitian [2], [3] dengan menambahkan proses baru untuk menghasilkan data latih masukan yang lebih baik lagi. Langkah demi langkah prapemrosesan data dalam penelitian ini dijabarkan pada beberapa paragraf berikut.

Pertama, nilai *NULL* dalam beberapa variabel kategori seperti *Agent* dan *Company*, disajikan sebagai salah satu data berjenis kategori. Data ini seharusnya tidak dianggap sebagai nilai yang hilang, tetapi lebih dianggap sebagai data yang "tidak diketahui". Misalnya, apabila "*Agent*" didefinisikan dengan *NULL*, maka hal ini berarti proses pemesanan hotel tidak berasal dari agen perjalanan. Namun, sedikit berbeda dengan variabel numerik seperti *Children* yang memiliki nilai *NULL*. Nilai ini dianggap sebagai nilai yang hilang. Oleh sebab itu, diberikan nilai pengganti berupa angka 0.

Kedua, proses penggantian nilai *NULL* menjadi hal yang sangat penting untuk dilakukan sebagai inisiasi awal dalam proses prapemrosesan. Data yang telah diisi nilai pengganti selanjutnya dilakukan pembersihan data. Hal ini bertujuan untuk menghilangkan data irasional pada jumlah pengunjung di setiap pengamatan. Data irasional yang dimaksud adalah data pengamatan yang memiliki jumlah pengunjung dewasa, anak-anak, dan bayi sebanyak 0 pengunjung.

Ketiga, *Country* dihapus dari dataset pemodelan. Selanjutnya, semua variabel data yang berisi format waktu dihapus dari data pengamatan. Hal ini disebabkan karena model dalam penelitian ini ditujukan untuk memprediksi pembatalan pesanan hotel di masa mendatang. Selain itu, atribut yang tidak memberi manfaat seperti *AssignedRoomType*, *RequiredCarParkingSpaces*, dan *ReservedRoomType*, juga dihilangkan dari data pengamatan. Variabel data tersebut hanya dapat menambah derau ke dalam elaborasi model yang didefinisikan. Fitur *ReservationStatus* juga dihapus dari dataset pengamatan karena memiliki nilai yang mirip dengan label pada dataset pengamatan sehingga sangat berpotensi menimbulkan *overfit* pada model.

Keempat, variabel data kategoris seperti *Company*, *DepositType*, *CustomerType*, *DistributionChannel*, *Meal*, dan *Agent* dikodekan ulang menggunakan metode *One-*

Tabel 2. Data latih masukan hasil prapemrosesan

Jenis data	Variabel
Label	IsCanceled
Fitur	LeadTime, MarketSegment, ADR, Adults, Children, Babies, Meal, CustomerType, TotalOfSpecialRequests, PreviousCancellations, PreviousBookingsNotCanceled, BookingChanges, DaysInWaitingList, IsRepeatedGuest, DistributionChannel, StaysInWeekendNights, StaysInWeekNights
Fitur kategoris	DistributionChannel, Agent, MarketSegment, DepositType, CustomerType, Company, Meal

Hot-Encoding. Hal ini disebabkan karena teknik pengkodean ulang dapat mengurangi efek kardinalitas yang tinggi pada variabel kategoris seperti *Agency* dan *Company*. Selain itu, teknik ini juga dapat menggeneralisir efek variabel yang tidak umum seperti *MarketSegment* dan *Meal*. Seger dkk. [12] menyatakan bahwa prediktor yang dikodekan ulang dapat memiliki dampak yang signifikan pada kinerja model.

Terakhir, seluruh rangkaian prapemrosesan data ini menghasilkan data pemodelan dengan fitur yang lebih bersih dan lebih banyak daripada [2], [3]. Data latih masukan yang terbentuk berisi label, fitur, dan fitur kategoris seperti dinyatakan dalam Tabel 2.

Dataset hasil prapemrosesan yang diterapkan menyisakan 40.047 data pengamatan untuk H1 dan 79.163 data pengamatan untuk H2. Selanjutnya, dataset H1 dan H2 tersebut dipisahkan menjadi data latih dan data uji dengan rasio 0,90 : 0,10. Data latih digunakan untuk melatih masing-masing model yang dibandingkan, sedangkan data uji digunakan untuk memverifikasi hasil evaluasi model.

C. Pendefinisian model

DT merupakan model *tree* yang paling sederhana, namun sangat penting dalam pembelajaran mesin [8]. Semua metode berbasis *tree* berasal dari algoritme ini,

termasuk algoritme RF. Algoritme ini membangun DT-nya menggunakan strategi *divide-and-conquer* yang mencari partisi terbaik dari data yang ada untuk dijadikan sebagai cabang simpul baru [13], [14]. Metode XGB merupakan algoritme *gradient lifting* yang dapat mengungguli beberapa model *ensemble* lain dengan cara menjumlahkan beberapa pengklasifikasi lemah melalui iterasinya [15]–[17]. Dalam setiap iterasinya fungsi *loss* dan regularisasi ditambahkan pada fungsi objektif yang dapat dioptimalkan kembali sebagai sebuah langkah perbaikan [18], [19].

Pendekatan RF tradisional dalam penelitian ini digunakan sebagai model verifikator untuk memvalidasi peningkatan kinerja model yang diusulkan. RF sendiri merupakan metode yang diusulkan oleh Breiman [20] berdasarkan algoritme DT yang sangat cocok untuk kasus klasifikasi dan regresi. Algoritme ini dapat mencapai akurasi yang lebih baik dari DT karena menerapkan metode *bagging* pada skema pemilihan keputusannya. Skema *bagging* tersebut memberikan unsur generalisasi yang baik dengan adanya penurunan varian serta peningkatan generalisasi kesalahan secara menyeluruh. Hal inilah yang dapat meningkatkan akurasi metode DT dengan ringkasan yang berasal dari sebagian besar model [21].

Metode pendekatan RF juga memiliki kelebihan lain berupa skema *bootstrap* dan *ensemble*. Skema ini membuat RF menjadi lebih tangguh terhadap masalah *overfit* sehingga tidak perlu melakukan pemangkasan pada *tree* yang telah didefinisikan. Dalam melakukan prediksinya RF mengkombinasikan hasil dari setiap DT dengan cara *majority vote* untuk kasus klasifikasi [22]. Selanjutnya, sebuah prediksi yang akurat dan generalisasi yang lebih baik dapat dicapai dengan pemanfaatan strategi *ensemble* dan pengambilan sampel prediktor acak.

D. Optimalisasi model

Prapemrosesan dalam penelitian ini telah menghasilkan data latih masukan dengan dimensi yang cukup besar (466 variabel untuk H1 dan 456 variabel untuk H2). Meskipun pendekatan RF memiliki kemampuan dalam menangani data berdimensi besar, bukan berarti untuk mendefinisikan model yang optimal dapat dilakukan dengan begitu saja. Hingga saat ini tidak ada formulasi pasti untuk mendefinisikan model RF yang optimal menangani pemodelan data berskala dimensi yang besar. Sementara itu, terdapat beberapa parameter model yang perlu didefinisikan untuk memperoleh kinerja model terbaik.

Metode optimalisasi hiperparameter pada algoritme RF menjadi pilihan yang tepat untuk mendapatkan kombinasi parameter model terbaik [7], [10]. Kombinasi parameter yang dihasilkan pada proses optimalisasi dapat mewakili nilai-nilai yang menentukan proses pelatihan model RF. Jumlah hiperparameter pada setiap metode dapat berbeda antara satu sama lainnya, semakin banyak jumlah parameter maka waktu pelatihan model pun juga semakin meningkat [23]. Optimalisasi hiperparameter adalah proses pencarian kemungkinan

kombinasi parameter model terbaik untuk digunakan oleh suatu algoritme pembelajaran mesin tertentu. Proses pencarian hiperparameter dinyatakan dengan Persamaan 1.

$$\lambda^* = \arg \arg f(\lambda), \lambda \in \Lambda \quad (1)$$

Notasi Λ sebagai ruang kombinasi hiperparameter dan $f(\lambda)$, $\lambda \in \Lambda$, untuk mengukur kinerja pembelajaran mesin dengan kombinasi hiperparameter λ . Algoritme optimalisasi mencari kombinasi hiperparameter $\lambda = \lambda^*$ yang memaksimalkan $f(\lambda)$. Untuk mencari solusi λ^* yang optimal merupakan hal yang tidak mudah sehingga permasalahan tersebut diganti dengan mencari kombinasi hiperparameter yang baik, misalnya $\Lambda' \subset \Lambda$ menjadi bagian yang cukup besar untuk kombinasi hiperparameter. Λ' ditentukan secara statis atau dapat menggunakan algoritma otomatis, seperti *Grid Search Algorithm*(GSA) dan RSA [23].

Kajian ini memilah RSA untuk optimalisasi model karena memiliki kemampuan yang cenderung lebih baik daripada GSA dalam menentukan kombinasi parameter optimal dari sekian banyak hiperparameter yang didefinisikan. RSA mendefinisikan Λ' dengan cara mengambil sampel acak pada rentang nilai tertentu hingga pencarian habis. Algoritme ini memiliki *baseline* yang berguna sebab tidak membuat asumsi pada algoritme pembelajaran mesin yang dioptimalkan. Dengan sumber daya yang cukup, RSA bisa memberikan kinerja mendekati optimal meski membutuhkan waktu yang lama untuk menemukan hasilnya [10], [24], [25].

Sebelum mendefinisikan model OHRF menggunakan teknik RSA, kumpulan parameter model ditentukan dan disimpan dalam sebuah *grid dictionary*. Parameter model yang ada pada *grid dictionary* digunakan, yaitu *n_estimators*, *max_features*, *max_depth*, *min_samples_split*, *min_samples_leaf*, dan parameter *bootstrap*. Parameter *n_estimators* berisi daftar angka bilangan bulat kelipatan 200 yang dimulai dari angka 100 untuk menentukan jumlah *tree* dalam RF. Parameter *max_features* berisi dua jenis parameter model untuk menentukan jumlah variabel penting yang perlu dipertimbangkan di setiap pemisahan data. Parameter *max_depth* didefinisikan sebagai daftar *linspace* dengan 11 *generate number* dari rentang nilai 10 sampai 110 untuk menentukan jumlah maksimum level dalam *tree*. Parameter *min_samples_split* berisi nilai bilangan bulat 2, 5 dan 10 yang berguna sebagai nilai minimum sampel untuk memisahkan simpul dalam *tree*. Parameter *min_samples_leaf* berisi daftar bilangan bulat 1, 2 dan 4 yang berfungsi sebagai jumlah sampel minimum yang dibutuhkan di setiap simpul *leaf*. Daftar parameter *bootstrap* yang berisi nilai *boolean True* dan *False* sebagai metode pemilihan sampel untuk melatih *tree*.

Setelah menentukan parameter model dalam *grid*, selanjutnya *estimator model* RF didefinisikan untuk digunakan oleh fungsi RSA. *Estimator model* dan

kumpulan parameter yang tersimpan dalam *grid* digunakan oleh fungsi RSA sebagai nilai parameter fungsinya. Parameter fungsi lain seperti 100 buah *n_iter* dan *4-folds cross-validation* digunakan untuk menentukan jumlah *fitting* model yang dilakukan oleh fungsi RSA. Dalam hal ini, terdapat 400 kombinasi *fitting* model dengan susunan hiperparameter yang berbeda-beda. Untuk menyelesaikan seluruh proses pelatihan model tersebut diperlukan waktu rata-rata 100 menit untuk dataset H1 dan 175 menit untuk dataset H2.

Seluruh model yang telah dilatih kemudian dievaluasi menggunakan beberapa parameter matriks umum yang telah ditentukan sebelumnya. Proses evaluasi bertujuan untuk memverifikasi seberapa baik kinerja model yang diusulkan dalam memprediksi pembatalan pemesanan hotel oleh pelanggan di masa depan.

E. Evaluasi model

Dalam studi ini, matriks evaluasi umum digunakan untuk mengukur performa masing-masing model yang dibandingkan. Matriks evaluasi yang digunakan, yaitu akurasi, presisi, sensitivitas (*recall*), spesifisitas, dan nilai *f1-score*. Matriks evaluasi tersebut dapat dihitung menggunakan Persamaan 2 sampai Persamaan 6 berdasarkan nilai yang telah diekstrak dari matriks konfusi. Akurasi menyatakan rasio total data yang diklasifikasikan benar (dibatalkan dan tidak dibatalkan) dalam sebuah set data uji. Presisi menyatakan rasio prediksi benar (dibatalkan) dibandingkan dengan keseluruhan hasil yang diprediksi dibatalkan. Persentase data yang diklasifikasikan dengan benar ditunjukkan oleh sensitivitas (*recall*). Spesifisitas menunjukkan klasifikasi akurat dari data yang tidak dibatalkan. *F1-score* menunjukkan perbandingan rata-rata antara presisi dan *recall*.

$$Akurasi = \frac{TP + TN}{P + N} \quad (2)$$

$$Presisi = \frac{TP}{TP + FP} \quad (3)$$

$$Sensitivitas (recall) = \frac{TP}{TP + FN} \quad (4)$$

$$Spesifisitas = \frac{TN}{TN + FN} \quad (5)$$

$$F1-Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (6)$$

True Positive (TP) adalah jumlah kasus yang diprediksi positif dibatalkan dan sebenarnya dibatalkan. *True Negative* (TN) merupakan jumlah kasus prediksi tidak dibatalkan dan sebenarnya tidak dibatalkan. *False Negative* (FN) adalah variabel total kasus yang diprediksi tidak dibatalkan namun sebenarnya dibatalkan (disebut juga kesalahan tipe dua). *False Positive* (FP) merupakan variabel jumlah kasus yang diprediksi dibatalkan, sedangkan sebenarnya tidak

Tabel 3. Nilai hasil ekstraksi matriks konfusi

Model	Data	TP	TN	FP	FN
DT	H1	781	2512	381	331
	H2	2598	3984	671	664
XGB [2]	H1	521	2792	101	591
	H2	2216	4350	305	1046
RF	H1	718	2705	188	394
	H2	2520	4254	401	742
OHRF	H1	745	2728	165	367
	H2	2514	4316	339	748

Tabel 4. Hasil pengukuran nilai akurasi, presisi, *recall*, *f-score* data H1 dan H2 pada skema kedua.

Model	Data	Akurasi	Presisi	Recall	F1
DT	H1	0.8222	0.6721	0.7023	0.6869
	H2	0.8314	0.7947	0.7964	0.7956
XGB [2]	H1	0.8272	0.8376	0.4685	0.6009
	H2	0.8294	0.8790	0.6793	0.7664
RF	H1	0.8547	0.7925	0.6457	0.7116
	H2	0.8556	0.8627	0.7725	0.8151
OHRF	H1	0.8672	0.8187	0.6700	0.7369
	H2	0.8627	0.8812	0.7707	0.8222

dibatalkan atau kesalahan tipe satu. Nilai hasil ekstraksi matriks konfusi dinyatakan dalam [Tabel 3](#).

III. HASIL DAN PEMBAHASAN

Hasil pengukuran nilai akurasi, presisi, *recall*, dan *f-score* atas data H1 dan H2 dari masing-masing metode pembelajaran mesin yang dibandingkan dapat dilihat pada [Tabel 4](#). Berdasarkan hasil tersebut diketahui bahwa nilai *recall* tertinggi dimiliki oleh algoritme DT. Namun, secara keseluruhan metode OHRF yang diusulkan memiliki kinerja yang lebih baik daripada pendekatan pembelajaran mesin lainnya (nilai evaluasi tertinggi ditandai dengan **bold**). Hal ini disebabkan oleh jumlah varian yang tinggi serta dan yang rendah akibat dari penggunaan *bagging* pada RF [4], [5]. Meskipun begitu, hal inilah yang menjadi karakteristik penting dari sub-model RF saat menggabungkan prediksinya. Selain itu, metode optimalisasi hiperparameter menggunakan pendekatan RSA [7], [10] terbukti dapat meningkatkan kinerja RF tradisional sebesar 0,71% sampai 2,62%.

Jumlah data yang tidak seimbang untuk setiap kelasnya membuat perbedaan nilai evaluasi model yang bisa saja menipu. Oleh karena itu, dalam penelitian ini menerapkan uji evaluasi model menggunakan *weighted average matrix* dan *balanced accuracy score* (BAS). *Weighted average matrix* merupakan fungsi yang menghitung nilai evaluasi dari nilai rata-rata terbobot untuk setiap label dengan mempertimbangkan proporsi ketidakseimbangan data. Evaluasi menggunakan teknik *weighted average matrix* dan BAS untuk dataset H1 dan H2 dapat dilihat pada [Tabel 5](#) dan [Tabel 6](#).

Berdasarkan hasil pengujian nilai BAS dan *weighted average matrix* (presisi, *recall*, *F1-score*) dapat diketahui bahwa OHRF memiliki performa yang jauh lebih baik daripada metode lainnya. Di sisi lain, metode XGB yang diusung dalam [2] diketahui memiliki performa yang paling rendah. Optimalisasi hiperparameter [6]–[10] pada RF tradisional terbukti bisa meningkatkan nilai BAS hingga 1.62% untuk dataset H1 dan 0.57% untuk dataset H2.

Untuk memverifikasi kinerja model yang diusulkan, kajian ini melakukan pengukuran matrik AUC, sensitivitas, dan spesifisitas seperti yang ditunjukkan pada Tabel 7. Meskipun metode OHRF memiliki nilai sensitivitas dan spesifisitas yang tidak lebih baik dari DT dan XGB, metode yang diusulkan telah berhasil meraih kinerja yang paling baik untuk dataset H2. Selain itu, teknik RSA yang digunakan juga telah terbukti dapat meningkatkan AUC dari RF tradisional sebesar 2,18 % untuk dataset H1 dan 0,87 % untuk H2. Dataset yang berasal dari dua catatan hotel yang berbeda dari [11] ditambah dengan banyaknya variasi data akibat proses *feature encoding* [12] yang digunakan dapat memicu perbedaan nilai evaluasi model meskipun tidak secara signifikan. Begitu pula dengan data yang tidak seimbang [7] antara masing-masing kelasnya, juga dapat mempengaruhi hasil evaluasi dari setiap model yang ada.

Seluruh rangkaian pengujian yang dilakukan dapat memperkuat kesimpulan bahwa pendekatan algoritme RF yang dioptimalisasi menggunakan metode RSA dapat melahirkan model OHRF yang jauh lebih tangguh dan stabil daripada model pendekatan pembelajaran mesin yang digunakan dalam [2] untuk prediksi pembatalan pesanan hotel.

IV. KESIMPULAN

Optimalisasi hiperparameter dapat meningkatkan performa model RF tradisional yang digunakan sehingga meraih nilai akurasi tertinggi sebesar 0,87. Pengukuran performa model lain menggunakan matriks AUC dan sensitivitas juga telah menunjukkan nilai evaluasi tertinggi sebesar 0,87 dan 0,88. Selain itu, prapemrosesan yang digunakan dapat memberikan data latih masukan yang jauh lebih baik dan lebih spesifik lagi untuk kasus prediksi pembatalan pemesanan hotel oleh pelanggan. Model OHRF yang diusulkan juga telah terbukti memiliki kinerja yang lebih baik daripada model lainnya. Hal tersebut ditunjukkan oleh kecenderungan evaluasi model OHRF yang lebih tinggi daripada model lain yang didefinisikan.

Hasil penelitian ini dapat dikembangkan lebih lanjut pada penelitian-penelitian lainnya, salah satunya adalah dalam membangun sistem manajemen pendapatan hotel berbasis metode pendekatan OHRF. Metode ini juga dapat diterapkan berupa aplikasi berbasis *mobile* maupun *website* yang terbatas untuk kalangan staf manajerial hotel yang bersangkutan.

Tabel 5. Hasil evaluasi *weighted average matrix* dan BAS untuk data H1

Model	Presisi	BAS	Recall	F1-Score
DT	0.8249	0.7853	0.8222	0.8234
XGB [2]	0.8287	0.7168	0.8272	0.8095
RF	0.8505	0.7903	0.8547	0.8498
OHRF	0.8640	0.8065	0.8672	0.8628

Tabel 6. Hasil evaluasi *weighted average matrix* dan BAS untuk data H2

Model	Presisi	BAS	Recall	F1-Score
DT	0.8314	0.8261	0.8314	0.8314
XGB [2]	0.8362	0.8069	0.8294	0.8247
RF	0.8561	0.8432	0.8556	0.8542
OHRF	0.8642	0.8489	0.8627	0.8610

Tabel 7. Nilai pengukuran AUC, sensitivitas dan spesifisitas data H1 dan H2 pada skema kedua

Model	Data	AUC	Sensitivitas	Spesifisitas
DT	H1	0.7286	0.8836	0.6721
	H2	0.8375	0.8571	0.7947
XGB [2]	H1	0.7269	0.8253	0.8376
	H2	0.8452	0.8062	0.8790
RF	H1	0.7683	0.8729	0.7925
	H2	0.8645	0.8515	0.8627
OHRF	H1	0.7901	0.8814	0.8187
	H2	0.8732	0.8523	0.8812

DAFTAR PUSTAKA

- [1] S. Kitamori, H. Sakai, and H. Sakaji, "Extraction of sentences concerning business performance forecast and economic forecast from summaries of financial statements by deep learning," in *IEEE Symposium Series on Computational Intelligence*, Honolulu, USA, Dec. 2017, pp. 1-7. doi: [10.1109/SSCI.2017.8285335](https://doi.org/10.1109/SSCI.2017.8285335)
- [2] N. Antonio, A. de Almeida, and L. Nunes, "Predicting hotel bookings cancellation with a machine learning classification model," in *IEEE International Conference on Machine Learning and Applications*, Cancun, Mexico, Dec. 2017, pp. 1049–1054. doi: [10.1109/ICMLA.2017.00-11](https://doi.org/10.1109/ICMLA.2017.00-11)
- [3] N. Antonio, A. de Almeida, and L. Nunes, "Predicting hotel booking cancellations to decrease uncertainty and increase revenue," *Tourism & Management Studies*, vol. 13, no. 2, pp. 25–39, 2017. doi: [10.18089/tms.2017.13203](https://doi.org/10.18089/tms.2017.13203)
- [4] L. Rokach, "Decision forest: Twenty years of research," *Information Fusion*, vol. 27, pp. 111–125, 2016. doi: [10.1016/j.inffus.2015.06.005](https://doi.org/10.1016/j.inffus.2015.06.005)
- [5] P. Fernandez-Gonzalez, C. Bielza, and P. Larranaga, "Random forests for regression as a weighted sum of k-potential nearest neighbors,"

- IEEE Access*, vol. 7, pp. 25660–25672, 2019. doi: [10.1109/ACCESS.2019.2900755](https://doi.org/10.1109/ACCESS.2019.2900755)
- [6] M. C. M. Oo and T. Thein, “Hyperparameters optimization in scalable random forest for big data analytics,” in *4th International Conference on Computer and Communication Systems*, Singapore, Singapore, Feb. 2019, pp. 125-129. doi: [10.1109/CCOMS.2019.8821752](https://doi.org/10.1109/CCOMS.2019.8821752)
- [7] B. H. Shekar and G. Dagnev, “Grid search-based hyperparameter tuning and classification of microarray cancer data,” in *International Conference on Advanced Computational and Communication Paradigms*, Gantok, India, Feb. 2019, pp. 1-8. doi: [10.1109/ICACCP.2019.8882943](https://doi.org/10.1109/ICACCP.2019.8882943)
- [8] T. Wang et al., “Random forest-bayesian optimization for product quality prediction with large-scale dimensions in process industrial cyber-physical systems,” *IEEE Internet Things Journal*, vol. 7, no. 9, pp. 8641-8653, 2020. doi: [10.1109/JIOT.2020.2992811](https://doi.org/10.1109/JIOT.2020.2992811)
- [9] E. Hazan, A. Klivans, and Y. Yuan, “Hyperparameter optimization: A spectral approach,” 2017, [arXiv:1706.00764](https://arxiv.org/abs/1706.00764)
- [10] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, pp. 281-305, 2012.
- [11] N. Antonio, A. de Almeida, and L. Nunes, “Hotel booking demand datasets,” *Data in Brief*, vol. 22, pp. 41–49, 2019. doi: [10.1016/j.dib.2018.11.126](https://doi.org/10.1016/j.dib.2018.11.126)
- [12] C. Seger, “An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing,” *thesis*, KTH Royal Institute of Technology, Stockholm, Sweden, 2018.
- [13] J. S. Lee, “AUC4.5: AUC-based C4.5 decision tree algorithm for imbalanced data classification,” *IEEE Access*, vol. 7, pp. 106034 – 106042, 2019, doi: [10.1109/ACCESS.2019.2931865](https://doi.org/10.1109/ACCESS.2019.2931865)
- [14] J. Li, S. Ma, T. Le, L. Liu, and J. Liu, “Causal decision trees,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 2, pp. 257–271, 2017. doi: [10.1109/TKDE.2016.2619350](https://doi.org/10.1109/TKDE.2016.2619350)
- [15] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *International Conference on Knowledge Discovery and Data Mining*, New York, USA, Aug. 2016, pp. 785-794. doi: [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785)
- [16] M. Chen, Q. Liu, S. Chen, Y. Liu, C. H. Zhang, and R. Liu, “XGBoost-Based algorithm interpretation and application on post-fault transient stability status prediction of power system,” *IEEE Access*, vol. 7, pp. 13149-13158, 2019. doi: [10.1109/ACCESS.2019.2893448](https://doi.org/10.1109/ACCESS.2019.2893448)
- [17] N. Li, B. Li, and L. Gao, “Transient stability assessment of power system based on XGBoost and factorization machine,” *IEEE Access*, vol. 8, pp. 28403-28414, 2020. doi: [10.1109/ACCESS.2020.2969446](https://doi.org/10.1109/ACCESS.2020.2969446)
- [18] S. Georganos, T. Grippa, S. Vanhuyse, M. Lennert, M. Shimoni, and E. Wolff, “Very high resolution object-based land use-land cover urban classification using extreme gradient boosting,” *IEEE Geoscience and Remote Sensing Letter*, vol. 15, no. 4, pp. 607-611, 2018. doi: [10.1109/LGRS.2018.2803259](https://doi.org/10.1109/LGRS.2018.2803259)
- [19] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, “A data-driven design for fault detection of wind turbines using Random Forests and XGboost,” *IEEE Access*, vol. 6, pp. 21020–21031, 2018. doi: [10.1109/ACCESS.2018.2818678](https://doi.org/10.1109/ACCESS.2018.2818678)
- [20] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, 2001. doi: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324)
- [21] S. Liu, H. Li, Y. Zhang, B. Zou, and J. Zhao, “Random forest-based track initiation method,” *Journal of Engineering*, vol. 2019, no. 19, pp. 6175-6179, 2019. doi: [10.1049/joe.2019.0180](https://doi.org/10.1049/joe.2019.0180)
- [22] A. Primajaya and B. N. Sari, “Random Forest algorithm for prediction of precipitation,” *Indonesian Journal of Artificial Intelligence and Data Mining*, vol. 1, no. 1, pp. 27, 2018. doi: [10.24014/ijaidm.v1i1.4903](https://doi.org/10.24014/ijaidm.v1i1.4903)
- [23] D. Marinov and D. Karapetyan, “Hyperparameter optimisation with early termination of poor performers,” in *Computer Science and Electronic Engineering*, Colchester, UK, Sept. 2019, pp. 160–163. doi: [10.1109/CEEC47804.2019.8974317](https://doi.org/10.1109/CEEC47804.2019.8974317)
- [24] B. Nakisa, M. N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, “Long short term memory hyperparameter optimization for a neural network based emotion recognition framework,” *IEEE Access*, vol. 6, pp. 49325–49338, 2018. doi: [10.1109/ACCESS.2018.2868361](https://doi.org/10.1109/ACCESS.2018.2868361)
- [25] M. Feurer and F. Hutter, *Hyperparameter Optimization*. Springer, 2019. doi: [10.1007/978-3-030-05318-5_1](https://doi.org/10.1007/978-3-030-05318-5_1)