

Klasifikasi Halaman SEO berbasis *Machine Learning* melalui *Mutual Information* dan *Random Forest Feature Importance*

SITI NURADILLA, KUSMAN SADIK, CICI SUHAENI, AGUS M SOLEH

Program Studi Statistika dan Sains Data, IPB University
Email: sitinuradilla@apps.ipb.ac.id

Received 5 Mei 2025 | Revised 20 Mei 2025 | Accepted 5 Juni 2025

ABSTRAK

Proses optimasi SEO melibatkan banyak faktor yang saling terkait, sehingga sulit bagi tim SEO dalam menentukan halaman mana yang memerlukan perbaikan lebih lanjut. Penelitian ini bertujuan untuk mengembangkan model berbasis machine learning yang tidak hanya akurat dalam mengklasifikasikan halaman, tetapi juga efisien dalam memilih fitur yang paling informatif. Metode yang digunakan dalam penelitian ini melibatkan seleksi fitur menggunakan Mutual Information (MI) dan Random Forest Feature Importance (RFFI) untuk mengidentifikasi faktor-faktor yang paling penting untuk optimasi SEO, yang dimodelkan menggunakan Random Forest dan Weighted Voting Ensemble (WVE). Model dievaluasi berdasarkan Accuracy, Precision, Recall, dan ROC AUC. Hasil penelitian menunjukkan bahwa model Random Forest dengan 20 fitur berdasarkan RFFI, memberikan performa terbaik dengan ROC AUC sebesar 75.87%, Accuracy 77.74%, Precision 60.51%, dan Recall 71.29%. Model mampu membedakan secara efektif halaman yang membutuhkan optimasi SEO atau tidak.

Kata kunci: Feature Importance, Random Forest, SEO, Seleksi Variabel, WVE

ABSTRACT

The SEO optimization process involves many interrelated factors, making it challenging to identify which pages need further improvement. This study proposes a machine learning-based model that is accurate in classifying web pages and efficient in selecting the most relevant features. Feature selection is performed using Mutual Information (MI) and Random Forest Feature Importance (RFFI) to identify key factors for SEO optimization, followed by modeling with Random Forest and Weighted Voting Ensemble (WVE). The model is evaluated using Accuracy, Precision, Recall, and ROC AUC. Results indicate that the Random Forest model with 20 features selected via RFFI delivers the best performance, achieving a ROC AUC of 75.87%, an Accuracy of 77.74%, a Precision of 60.51%, and a Recall of 71.29%. The model effectively distinguishes between pages that require SEO optimization and those that do not.

Keywords: Feature Importance, Random Forest, SEO, Variable Selection, WVE

1. PENDAHULUAN

Era revolusi industri 4.0 dan *society* 5.0 membawa transformasi besar dalam berbagai sektor salah satunya dunia bisnis dan pemasaran digital. Pemangku bisnis berbondong-bondong menggunakan teknik yang efektif dan efisien untuk menawarkan produknya. Salah satu teknik pemasaran digital yang unggul yaitu *Search Engine Optimization* (SEO). SEO menjadi aspek penting dalam meningkatkan visibilitas dan performa suatu *website* di mesin pencari seperti Google (**Alfiana, dkk., 2023**). Pada konteks ini, halaman *website* yang dioptimasi dengan baik dapat meningkatkan *traffic* pengunjung dan konversi yang menguntungkan bagi pemilik *website*. Pemahaman yang baik tentang optimasi SEO sangat penting, khususnya pada SEO *On-page* yang dapat memengaruhi peringkat web.

Banyak faktor yang mempengaruhi peringkat SEO, seperti konten, struktur halaman, kecepatan *loading*, serta pengalaman pengguna mengakibatkan tim SEO sering kali mendapatkan tantangan dalam menentukan faktor mana yang diprioritaskan untuk optimasi SEO (**Ziakis, dkk., 2019**). Proses ini menjadi semakin kompleks karena berbagai elemen tersebut saling terkait dan dapat mempengaruhi peringkat halaman *website* secara berbeda, tergantung pada algoritma mesin pencari yang terus berkembang (**Aryani, dkk., 2023**). Selain itu, banyak pemilik *website* atau tim SEO yang masih mengandalkan pendekatan manual untuk menentukan halaman mana yang memerlukan perbaikan lebih lanjut (**Rachita & Pandey, 2024**). Hal ini rentan terhadap kesalahan dan keterbatasan waktu.

Pada praktiknya, proses optimasi SEO yang efektif memerlukan identifikasi elemen-elemen yang paling berpengaruh terhadap kinerja halaman di mesin pencari. Di statistika, konsep ini disebut seleksi fitur yang mampu meningkatkan generalisasi dan pemahaman yang lebih sederhana terhadap data. Proses ini dapat menggunakan pendekatan *Mutual Information* (MI) dan *Random Forest Feature Importance* (RFFI) (**Alduailij, dkk., 2022**). Pendekatan ini dapat menganalisis dan menentukan secara objektif fitur-fitur mana yang berkontribusi signifikan terhadap kinerja SEO. MI mengukur ketergantungan informasi antara fitur dan target, yang sangat berguna ketika ada hubungan *non-linear* antara keduanya (**Beraha, dkk., 2019**). Sementara itu, RFFI sebagai metode berbasis pohon keputusan, mengidentifikasi fitur-fitur yang paling berpengaruh dengan cara mengukur pengurangan impuritas yang terjadi ketika fitur tersebut digunakan dalam *splitting* data (**Probst, dkk., 2019**).

Keunggulan utama dari penggunaan MI dan RFFI dalam seleksi fitur adalah kemampuannya untuk menangkap relasi yang kompleks dan *non-linear* yang dapat terlewat oleh metode konvensional lainnya (**Hoilijoki, dkk., 2022**). Berdasarkan penelitian (**Alduailij, dkk., 2022**), MI dan RFFI terbukti efektif dalam seleksi fitur pada sistem deteksi serangan, termasuk dalam *cloud computing* dan *network intrusion detection systems* (IDS). Penggunaan kedua teknik ini memudahkan identifikasi faktor-faktor SEO yang paling berpengaruh dalam mengklasifikasikan halaman yang perlu dioptimasi atau tidak. Hal ini akan memudahkan tim SEO untuk mengklasifikasikan halaman secara terarah dan objektif.

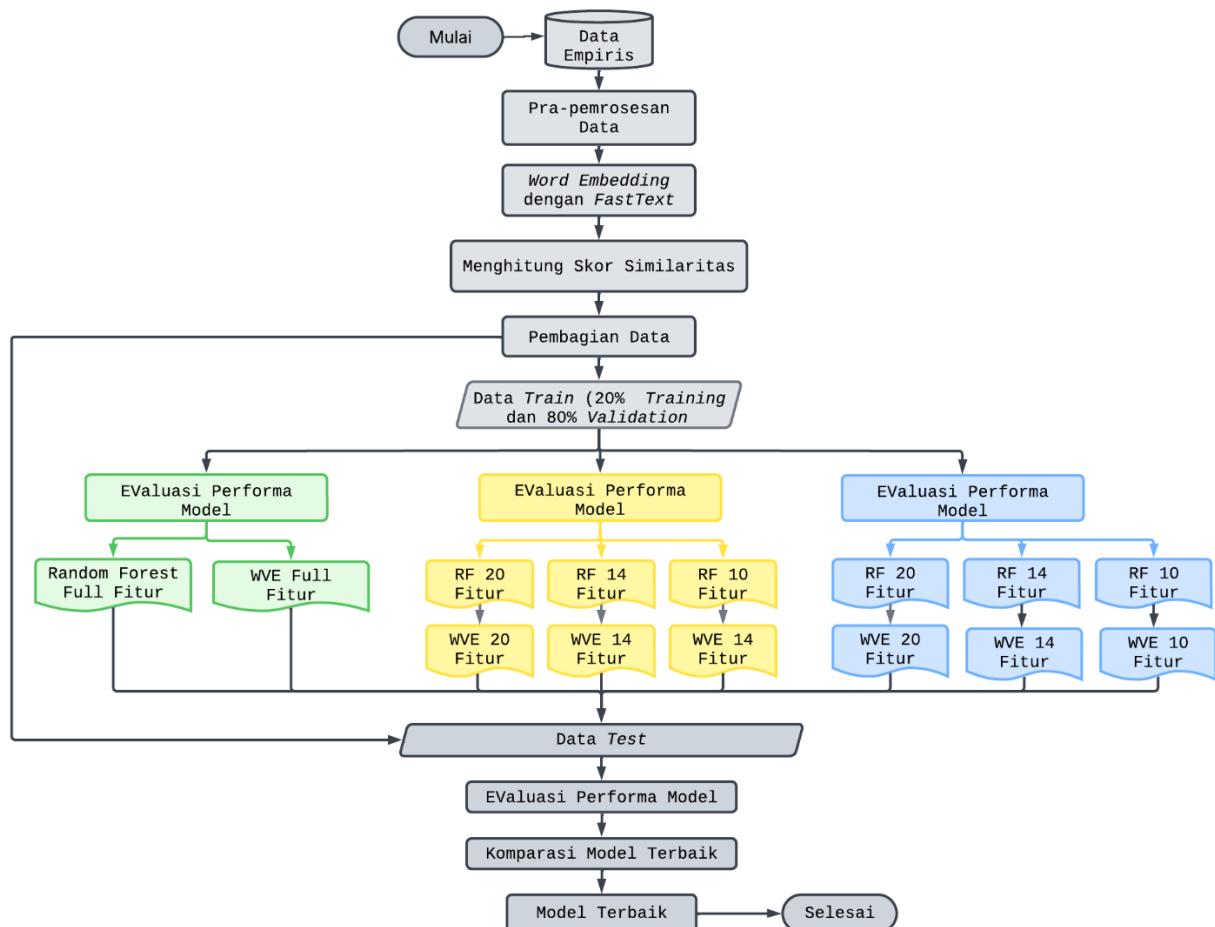
Penelitian mengenai klasifikasi performa SEO pada halaman *website* yang sudah dilakukan yakni penerapan *machine learning* berdasarkan konten dan *backlink* menggunakan pohon keputusan, *Naive Bayes*, Regresi Logistik, SVM dan KNN (**Matošević, dkk., 2021**). Namun, penelitian tersebut tidak mempertimbangkan seleksi fitur yang mampu meningkatkan efisiensi model. Selain itu, penelitian juga tidak melihat fitur semantik, sehingga belum diketahui sejauh mana representasi teks berperan dalam model tersebut. Penelitian lainnya mengklasifikasikan peringkat SEO ke dalam tiga kategori, menggunakan 36 faktor yang dianggap berpengaruh. Hasilnya menunjukkan bahwa model *Random Forest* memberikan kinerja terbaik dengan akurasi rata-rata 72% yang dapat membantu mengidentifikasi halaman web yang

memerlukan perbaikan SEO (**Pandey & Pandeya, 2023**). Penelitian tersebut tidak menggunakan seleksi fitur sehingga kurang efisien dalam memprioritaskan elemen yang perlu diperhatikan dalam penulisan artikel.

Menyikapi kekurangan yang ada pada penelitian sebelumnya, penelitian ini akan menggunakan *Random Forest* (RF) dan *Weighted Voting Ensemble* (WVE) dengan teknik seleksi fitur berbasis *Mutual Information* (MI) dan *Random Forest Feature Importance* (RFFI). *Random Forest* terbukti efektif dalam mengidentifikasi fitur-fitur penting dalam dataset yang besar dan kompleks, serta menghasilkan model yang lebih akurat (**Capitaine, dkk., 2021**). Sementara WVE dengan pendekatan *soft voting*, menggabungkan kekuatan *Random Forest* dan *Extreme Gradient Boosting* untuk menghasilkan prediksi yang lebih *robust* dan lebih baik dalam menangani ketidakseimbangan data (**Chakravarthy & Rajaguru, 2022**). Di samping itu, peneliti juga memanfaatkan informasi semantik antar komponen SEO *On-Page* menggunakan teknik *Word Embedding* yaitu *FastText* yang efisien dalam memetakan kata-kata dalam artikel menjadi representasi numerik (**Badri, dkk., 2022**). Hasil representasi tersebut digunakan untuk menghitung koherensi semantik antar fitur SEO. Penelitian ini bertujuan untuk menginisiasi model yang tidak hanya akurat dalam mengklasifikasikan halaman yang perlu dioptimasi, tetapi juga efisien dalam memilih fitur yang berpengaruh.

2. METODOLOGI PENELITIAN

Metodologi yang digunakan disusun secara sistematis meliputi pengumpulan data, pra-pemrosesan, perhitungan similaritas, hingga pelatihan dan evaluasi model. Gambar 1 berikut ini menyajikan diagram alir analisis data yang digunakan dalam penelitian.



Gambar 1. Diagram Alir Analisis Data

2.1 Pengumpulan Data

Penelitian ini memanfaatkan data yang diekstrak dari *Screaming Frog* dan *Semrush*. Data yang dikumpulkan meliputi komponen *SEO On-Page* yang tercantum pada Tabel 1. Data tersebut berasal dari *platform* yang telah melalui proses *web crawling*. Fitur X22 sampai X26 diperoleh dengan perhitungan similaritas yang akan dijelaskan selanjutnya.

Tabel 1. Fitur-fitur pada Data SEO

| Fitur | Keterangan Fitur | Fitur | Keterangan Fitur |
|-------|-------------------------------------|-------|------------------------|
| X1 | Panjang Judul | X15 | <i>Inlinks</i> |
| X2 | Lebar Pixel Judul | X16 | <i>Outlinks</i> |
| X3 | Panjang <i>Meta Description</i> | X17 | Waktu Respon |
| X4 | Lebar Pixel <i>Meta Description</i> | X18 | Kesulitan Kata Kunci |
| X5 | Panjang H1 | X19 | Volume Pencarian |
| X6 | Ukuran (<i>kilobytes</i>) | X20 | <i>Traffic</i> |
| X7 | Total Transfer (<i>kilobytes</i>) | X21 | Perubahan Peringkat |
| X8 | Jumlah Kata dalam Halaman | X22 | Similaritas URL_Title |
| X9 | Jumlah Kalimat dalam Halaman | X23 | Similaritas URL_Meta |
| X10 | Rata-rata Kata per Kalimat | X24 | Similaritas URL_H1 |
| X11 | Skor <i>Flesch Reading Ease</i> | X25 | Similaritas Title_Meta |
| X12 | <i>Readability</i> | X26 | Similaritas Title_H1 |
| X13 | Rasio Teks | X27 | Similaritas Meta_H1 |
| X14 | Kedalaman <i>Crawling</i> | Y | Status SEO |

Selain menampilkan fitur pada data, Tabel 2 dan Tabel 3 berikut menyajikan sampel baris data sebanyak 4 observasi yang digunakan dalam penelitian ini.

Tabel 2. Sampel Data SEO pada Fitur X1 – X14

| X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | X11 | X12 | X13 | X14 |
|-----|------|-----|-----|----|-----|------|------|-----|------|-------|--------|------|-----|
| 51 | 491 | 143 | 942 | 51 | 227 | 37,9 | 1976 | 317 | 6,23 | 14,18 | Hard | 7,59 | 2 |
| 109 | 1055 | 132 | 863 | 46 | 232 | 39,2 | 2537 | 401 | 6,33 | 10,21 | Hard | 9,00 | 2 |
| 60 | 565 | 127 | 844 | 60 | 224 | 38,8 | 2188 | 357 | 6,13 | 13,71 | Hard | 8,14 | 2 |
| 60 | 547 | 142 | 885 | 60 | 212 | 37,8 | 1876 | 294 | 6,38 | 13,53 | Normal | 7,60 | 2 |

Tabel 3. Sampel Data SEO pada Fitur X15 – X27 dan Fitur Respon

| X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X25 | X26 | X27 | Y |
|-----|-----|------|-------|-------|------|-------|------|------|------|------|-----|------|---|
| 38 | 126 | 0,20 | 25,06 | 361,1 | 248 | 47,22 | 0,85 | 0,75 | 0,85 | 0,83 | 1 | 0,83 | 0 |
| 35 | 131 | 0,19 | 20,68 | 370,5 | 31 | 48,77 | 0,61 | 0,73 | 0,82 | 0,54 | 0,8 | 0,72 | 0 |
| 48 | 135 | 0,13 | 17,35 | 530,2 | 696 | 58,38 | 0,90 | 0,87 | 0,90 | 0,82 | 1 | 0,82 | 0 |
| 35 | 126 | 0,14 | 23,02 | 669,2 | 4486 | 37,16 | 0,90 | 0,84 | 0,90 | 0,86 | 1 | 0,86 | 1 |

2.2 Pra-Pemprosesan Data

Data yang dikumpulkan terdiri dari dua jenis yaitu data teks dan numerik, yang perlu diproses agar dapat digunakan secara efektif pada model. Data teks umumnya mengandung informasi yang lebih kompleks dan membutuhkan pra-pemrosesan yang lebih intensif, sementara data numerik dapat langsung digunakan dalam model. Langkah-langkah pra-pemprosesan data teks maupun numerik mencakup (**Naseem, dkk., 2021**):

- Pembersihan data: Menghapus data duplikat dan menangani data yang hilang, serta menyaring karakter tidak relevan yang dapat mengganggu analisis.
- Penghapusan tanda baca: Menghilangkan tanda baca seperti koma, titik, tanda seru, dan tanda tanya untuk mereduksi noise dalam teks.

- Penghapusan *Stopwords*: Mengeleminasi kata-kata umum (*stopwords*) yang tidak memberikan kontribusi makna yang relevan dalam konteks *SEO*.
- Penghapusan nilai numerik: Menghapus angka yang tidak relevan dalam pemrosesan teks, khususnya jika angka tersebut tidak memberikan kontribusi arti spesifik.
- Tokenisasi: Memecah teks menjadi unit-unit kata untuk mempermudah pemrosesan lebih lanjut dan *Word Embedding* akan lebih efektif.
- *Lemmatization*: Mengubah kata ke bentuk dasarnya (*lemmatization*) agar lebih konsisten.
- Pengkodean data kategorikal: Mengubah fitur kategori menjadi format numerik, untuk memudahkan pemrosesan oleh model *machine learning*. Hal ini termasuk pengkodean fitur seperti "Status SEO".

2.3 Representasi Teks dalam Format Numerik

Fitur dari SEO terdiri dari data teks dan numerik, sehingga diperlukan representasi teks ke numerik agar dapat diproses oleh algoritma *machine learning*. *FastText* dikembangkan oleh tim riset AI Facebook sebagai penyempurnaan dari *Word2Vec* (**Khomsah, dkk., 2022**). Model ini memiliki arsitektur yang mirip dengan *Continuous Bag of Words* (CBOW), namun dengan representasi kata dalam bentuk vektor yang padat. *FastText* efisien dalam pelatihan, yang memproses hingga satu miliar kata hanya dalam sepuluh menit dengan hasil yang dapat bersaing dengan model lainnya (**Bojanowski, dkk., 2017**). *FastText* menggunakan informasi *subword* (*n-gram* karakter), sehingga menghasilkan representasi kata yang lebih akurat dan tangguh terhadap kata yang belum pernah dilihat (**Hastuti, dkk., 2023**). Melalui kemampuan ini, *FastText* unggul dibandingkan *Word Embedding* lain dalam menangani kata baru.

2.4 Perhitungan Similaritas

Setelah representasi teks berhasil dilakukan, vektor hasil representasi kemudian digunakan dalam menghitung similaritas antar fitur SEO. Similaritas ini sangat berpengaruh dalam proses perankingan website di mesin pencari (**Confetto and Covucci, 2021**). Similaritas diperoleh dengan menghitung *Cosine Similarity* yang terdapat pada Persamaan (1).

$$\text{Similarity Score} = \cos \theta = \frac{A \cdot B}{|A||B|} \quad (1)$$

dengan *A* dan *B* merupakan representasi vektor dari dua dokumen yang dibandingkan.

2.5 Proses Seleksi Fitur

Agar website tetap memiliki visibilitas tinggi, tim SEO perlu melakukan optimasi secara efektif dalam waktu terbatas. Dengan banyaknya aspek SEO yang perlu diperhatikan (± 27 fitur), penelitian ini menawarkan solusi seleksi fitur prioritas menggunakan *Mutual Information* dan *Random Forest Feature Importance* untuk menentukan halaman yang perlu dioptimasi.

2.5.1 Mutual Information (MI)

Mutual Information (MI) mengukur seberapa banyak informasi yang dibagikan antara dua fitur acak atau menggambarkan tingkat ketergantungan antara dua fitur (**Laarne, dkk., 2021**). Secara sederhana, MI dapat dipahami sebagai ukuran yang menunjukkan seberapa besar informasi mengenai suatu fitur dapat mengurangi ketidakpastian pada fitur lainnya. Oleh karena itu, MI sangat berguna dalam proses seleksi fitur, di mana fitur yang lebih relevan diprioritaskan, model tetap dapat mempertahankan informasi yang penting meskipun dengan jumlah fitur yang lebih sedikit. Salah satu keunggulan MI adalah kemampuannya untuk menangkap hubungan *non-linier* antara fitur dan fitur target, yang sering kali menjadi batasan pada metode pemilihan fitur lainnya (**Zaidan, dkk., 2019**). Pada pembelajaran mesin, MI digunakan untuk memilih fitur yang memiliki ketergantungan tinggi dengan fitur target, sehingga meningkatkan akurasi model.

MI dihitung dengan mengukur Entropi dari suatu fitur dan mengamati perubahan Entropi tersebut ketika fitur lain diketahui. MI antara dua fitur X dan Y yaitu selisih antara Entropi Y dan Entropi kondisional $Y|X$, yang menggambarkan pengurangan ketidakpastian Y setelah mengamati X (**Beraha, dkk., 2019**), yang terdapat pada Persamaan (2).

$$I(X; Y) = H(Y) - H(Y|X) \quad (2)$$

Dimana:

- $H(Y)$: Entropi dari fitur Y yang mengukur ketidakpastian informasi
- $H(Y|X)$: Entropi kondisional Y setelah diberikan X, untuk mengukur ketidakpastian Y setelah diketahui X.

2.5.2 Random Forest Feature Importance (RFFI)

Random Forest merupakan salah satu algoritma *Ensemble Learning* yang mengandalkan kombinasi teknik *Bagging (Bootstrap Aggregating)* dan pemilihan fitur secara acak dalam membangun pohon keputusan (**Sohil, dkk., 2022**). Pada konteks seleksi fitur, *Random Forest* memiliki mekanisme internal yang memungkinkan penilaian kontribusi masing-masing fitur terhadap performa model, yang dikenal sebagai *Random Forest Feature Importance* (RFFI). Proses ini bekerja dengan mengevaluasi seberapa besar suatu fitur menyebabkan penurunan *Gini Impurity* pada setiap percabangan (*node*) pohon (**Disha & Waheed, 2022**). Semakin besar total penurunan *Impurity* yang disebabkan oleh sebuah fitur di seluruh pohon dalam hutan, maka semakin tinggi skor kepentingannya.

Pada saat proses pembentukan pohon, fitur yang dipilih untuk memecah simpul ditentukan berdasarkan penurunan *Impurity* terbesar. Penurunan dihitung dari selisih *Impurity* sebelum dan sesudah *split* (**Xie, dkk., 2023**). Skor kepentingan fitur dinyatakan pada Persamaan (3).

$$Vim_j = \sum_{t \in T_j}^K p(t) \cdot \Delta i(t) \quad (3)$$

di mana:

- Vim_j : skor kepentingan fitur ke- j
- T_j : himpunan semua simpul di seluruh pohon yang menggunakan fitur j
- $p(t)$: proporsi sampel yang mencapai simpul t
- $\Delta i(t)$: penurunan *Impurity* akibat pemisahan data oleh fitur j di simpul t

Skor ini digunakan untuk mengurutkan fitur berdasarkan pengaruhnya terhadap akurasi model. Fitur dengan skor tertinggi dianggap paling penting dalam membentuk pohon.

2.6 Pemodelan *Machine learning*

Setelah melalui prapemprosesan data hingga seleksi fitur, selanjutnya dilakukan pemodelan menggunakan *machine learning* untuk mengklasifikasikan halaman web berdasarkan performa *SEO*. Untuk setiap skenario (tanpa seleksi, MI, dan RFFI), *pipeline* dibangun ulang secara independen, termasuk *scaling*, SMOTE, dan *tuning*. Model yang digunakan yaitu:

2.6.1 Random Forest (RF)

Random Forest menjadi salah satu algoritma yang bersifat *ensemble* dengan teknik *Bagging (Bootstrap Aggregating)* dalam membentuk beberapa pohon menjadi hutan (*Forest*). *Random Forest* melakukan perbaikan pada *Bagged Trees* dengan cara mengurangi korelasi antar pohon (**Sohil, dkk., 2022**). Artinya, pohon-pohon diatur sedemikian rupa agar tidak bergantung satu dengan yang lainnya. *Random Forest* mengkombinasikan hasil prediksi dari seluruh pohon untuk memberikan keputusan akhir menggunakan *Majority Vote* (**Mienye & Sun, 2022**).

Algoritma *Random Forest* dimulai dengan membentuk beberapa *bootstrap sample* dari data pelatihan berukuran n . Setiap *bootstrap sample* digunakan untuk membangun satu pohon keputusan tanpa proses *pruning*. Pada setiap simpul dalam pohon, algoritma akan memilih subset fitur secara acak sebanyak m dari total p fitur yang tersedia, dengan ketentuan $m < p$ (**Sohil, dkk., 2022**). Pemilihan fitur terbaik dari subset ini dilakukan berdasarkan kriteria *Gini Impurity* atau *Information Gain*. Proses pembentukan pohon dilanjutkan hingga tercapai kedalaman maksimum atau batasan parameter tertentu. Strategi pemilihan subset acak ini efektif dalam mengurangi korelasi antar pohon, sehingga menghasilkan variasi struktur pohon yang lebih tinggi dibandingkan metode *Bagged Trees* (**Capitaine, dkk., 2021**). Dengan pendekatan ini, *Random Forest* mampu menghasilkan prediksi yang lebih stabil dan akurat.

Pada *Random Forest* terdapat beberapa *hyperparameter* yang secara langsung berpengaruh terhadap performa model (**Probst, dkk., 2019**). *Hyperparameter* tersebut meliputi *max depth*, yang mengatur kedalaman maksimum pohon dan *max features* yang menentukan jumlah fitur pada setiap simpul. Selain itu, *min samples split* mengatur jumlah minimum sampel di simpul, dan *min samples leaf* menetapkan jumlah minimum sampel. Terakhir, *n_estimators* menunjukkan jumlah pohon dalam hutan yang dibangun.

2.6.2 *Weighted Voting Ensemble (WVE) Classifier*

Weighted Voting Ensemble (WVE) merupakan teknik *ensemble learning* yang menggabungkan prediksi dari beberapa *base classifiers* dengan memberikan bobot berbeda pada masing-masing model berdasarkan performanya (**Dogan & Birant, 2019**). WVE bekerja dengan memberikan bobot besar pada model yang lebih andal. Pada WVE, penggabungan algoritma seperti *Random Forest* (RF) dan *Extreme Gradient Boosting* (XGB) sering digunakan. RF bekerja dengan prinsip *Bagging*, menghasilkan banyak pohon keputusan dari sampel acak yang independen dan mampu mengurangi *overfitting* (**Sohil, dkk., 2022**). Sebaliknya, XGB menerapkan *Boosting*, membangun pohon secara berurutan untuk memperbaiki kesalahan pohon sebelumnya, sehingga menghasilkan model yang lebih presisi (**Fatima, dkk., 2023**).

Menggabungkan RF dan XGB dalam WVE memungkinkan pengambilan keputusan yang lebih adaptif dan akurat. Model dengan performa terbaik akan lebih berpengaruh terhadap hasil prediksi akhir. Kombinasi ini memanfaatkan kekuatan RF dalam generalisasi dan keunggulan presisi XGB, menghasilkan model yang lebih stabil dan akurat. Berdasarkan (**Osamor & Okezie, 2021**), penerapan WVE mampu meningkatkan performa klasifikasi secara signifikan pada studi terkait diagnosis prediktif *tuberculosis*. Pada kajian tersebut, WVE menunjukkan akurasi yang lebih tinggi dibandingkan *classifier* tunggal, yakni mencapai 95% akurasi, dibandingkan 92% dan 87% yang diperoleh oleh SVM dan Naïve Bayes secara individual. Proses pelatihan model dilakukan dengan tahapan berikut:

1. Seleksi model: Pilih beberapa *base classifiers* yang berbeda (RF dan XGB).
2. Evaluasi performa: Uji masing-masing model pada data validasi untuk memperoleh nilai akurasi atau metrik lain.
3. Pemberian bobot: Setiap model diberikan bobot yang proporsional terhadap performanya. Formula bobot WVE terdapat pada Persamaan (4).

$$w_i = \frac{Accuracy_i}{\sum_{j=1}^k Accuracy_j} \quad (4)$$

4. *Voting* tertimbang: Untuk setiap prediksi, kelas akhir ditentukan berdasarkan jumlah suara dari model yang dikalikan dengan bobotnya.

2.7 Evaluasi Model

Kinerja model diukur menggunakan beberapa metrik evaluasi, salah satunya *ROC AUC*. *ROC AUC* (*Receiver Operating Characteristic - Area Under the Curve*) mengukur kemampuan model

dalam membedakan antara kelas positif dan negatif. Metrik ini cocok untuk data tidak seimbang karena menggabungkan sensitivitas dan spesifikasi model (**Carrington, dkk., 2023**). Skor ROC AUC dijabarkan pada Persamaan (5).

$$Skor ROC AUC = \int_0^1 Sensitivity(t)d(1 - Specificity(t)) \quad (5)$$

Dimana:

- *Sensitivity (True Positive Rate, TPR):* $Sensitivity = \frac{TP}{TP+FN}$
- *Specificity (True Negative Rate, TNR):* $Specificity = \frac{TN}{TN+FP}$
- *FPR (False Positive Rate):* $FPR = \frac{FP}{FP+TN}$

Selain *ROC AUC*, metrik *Accuracy*, *Precision*, dan *Recall* digunakan untuk mengevaluasi keseimbangan deteksi kelas (**Naidu, dkk., 2023**). *Accuracy* mengukur proporsi prediksi yang benar secara keseluruhan, *Precision* menilai ketepatan prediksi positif, dan *Recall* menunjukkan kemampuan model menangkap seluruh kelas positif. Model dengan nilai *ROC AUC* terbaik dipilih sebagai model akhir untuk mengklasifikasikan halaman.

3. HASIL DAN PEMBAHASAN

Penelitian ini menguji efektivitas metode seleksi dan model *machine learning* yang tepat dalam mengklasifikasikan halaman yang perlu dioptimasi secara SEO. Metode seleksi yang digunakan yaitu *Mutual Information* (MI) dan *Random Forest Feature Importance* (RFFI), dengan tiga skenario (20, 14, dan 10 fitur terpenting) yang dimodelkan menggunakan *Random Forest* dan *Weighted Voting Ensemble* (WVE).

3.1 Mutual Information (MI)

Teknik seleksi fitur *Mutual Information* (MI) digunakan untuk mengevaluasi kontribusi masing-masing fitur terhadap fitur target. Proses seleksi fitur dimulai dari pemisahan fitur penjelas dan fitur target, dilanjutkan dengan pembagian data latih dan data uji, penanganan *missing value* menggunakan rata-rata, dan normalisasi. Skor MI dihitung menggunakan *SelectKBest*, menghasilkan skor relevansi fitur yang ditampilkan pada Tabel 4. Hasil seleksi dapat mengidentifikasi fitur yang paling berkontribusi untuk menentukan status SEO pada suatu halaman web dan menjadi dasar dalam pemodelan lebih lanjut. Performa model diuji menggunakan seluruh fitur serta skenario 20, 14, dan 10 fitur terpenting menggunakan *Random Forest* (RF) dan WVE. 20 fitur yang terpenting dari hasil MI sebagai berikut.

Tabel 4. 20 Fitur Terpenting dari Mutual Information

| No. | Fitur | Skor MI | No. | Fitur | Skor MI |
|-----|---------------------------------|----------|-----|--------------------------------|----------|
| 1 | <i>Traffic</i> | 0,069838 | 11 | Similaritas Meta_H1 | 0,008906 |
| 2 | Perubahan Peringkat | 0,046130 | 12 | <i>Readability_Fairly Hard</i> | 0,007044 |
| 3 | Ukuran (<i>bytes</i>) | 0,023198 | 13 | Similaritas Title_Meta | 0,005464 |
| 4 | Total Transfer (<i>bytes</i>) | 0,020599 | 14 | <i>Outlinks</i> | 0,005241 |
| 5 | Similaritas URL_Title | 0,018583 | 15 | Panjang H1 | 0,004381 |
| 6 | Kesulitan Kata Kunci | 0,016109 | 16 | Rata-rata Kata per Kalimat | 0,004358 |
| 7 | Jumlah Kata pada Halaman | 0,011535 | 17 | Skor Flesch Reading Ease | 0,002656 |
| 8 | <i>Inlinks</i> | 0,010354 | 18 | Panjang Meta Description | 0,001915 |
| 9 | Volume Pencarian | 0,009841 | 19 | <i>Readability_Very Hard</i> | 0,000815 |
| 10 | <i>Readability_Fairly Easy</i> | 0,009294 | 20 | Panjang Judul | 0,000447 |

3.1.1 Random Forest dengan Fitur Hasil Mutual Information (MI)

Pada penelitian ini, model *Random Forest* dioptimalkan dengan parameter terbaik melalui *GridSearchCV*, yang menunjukkan kombinasi parameter optimal. Beberapa parameter yang diatur mencakup kedalaman pohon (*max_depth*) untuk membatasi kompleksitas model. Selain itu, penurunan jumlah sampel minimum untuk membelah pohon (*min_samples_split* dan *min_samples_leaf*) memberikan fleksibilitas lebih dalam menangani data dengan variasi yang lebih besar. Jumlah pohon yang digunakan (*n_estimators*) juga diatur untuk mengurangi variansi dan meningkatkan performa model. Setelah dilakukan pelatihan, diperoleh model dengan performa sebagai berikut.

Tabel 5. Hasil Evaluasi Model *Random Forest* dengan *Mutual Information*

| Model | Accuracy | Precision | Recall | ROC AUC |
|------------------------------------|------------------|------------------|------------------|------------------|
| <i>Random Forest</i> – Semua Fitur | 0,7650685 | 0,5917526 | 0,6643519 | 0,7358724 |
| <i>Random Forest</i> – 20 Fitur | 0,7630137 | 0,5860000 | 0,6782407 | 0,7384394 |
| <i>Random Forest</i> – 14 Fitur | 0,7630137 | 0,5830116 | 0,6990741 | 0,7444787 |
| <i>Random Forest</i> – 10 Fitur | 0,7575342 | 0,5744275 | 0,6967593 | 0,7399166 |

Berdasarkan Tabel 5, *Random Forest* menggunakan semua fitur, menghasilkan *Accuracy* 76,51%, *Precision* 59,18%, *Recall* 66,44%, dan *ROC AUC* 73,59%. Skor ini menunjukkan bahwa model dengan semua fitur memberikan performa yang cukup baik secara keseluruhan. Namun, nilai *Precision* yang relatif rendah mengindikasikan bahwa model cenderung menghasilkan beberapa kesalahan klasifikasi positif (menyatakan artikel dengan "Optimasi" sebagai "Tidak Optimasi"). *Di samping itu*, *Random Forest* dengan 20 Fitur menunjukkan sedikit penurunan dalam *Accuracy* (76,30%), tetapi terjadi peningkatan pada *Recall* menjadi 67,82%. Hal ini menunjukkan bahwa model lebih baik dalam menangkap artikel dengan kelas "Tidak Optimasi" (kelas positif) meskipun sedikit mengorbankan *Precision*. Nilai *ROC AUC* juga meningkat menjadi 73,84%, menunjukkan kemampuan model untuk membedakan kelas positif dan negatif lebih baik.

Random Forest dengan 14 fitur menunjukkan peningkatan yang signifikan pada *Recall* (69,91%), artinya model ini lebih sensitif dalam mengidentifikasi artikel dengan kelas "Tidak Optimasi". Meskipun ada penurunan kecil dalam *Accuracy* (76,30%), dan *Precision* (58,30%), *ROC AUC* meningkat menjadi 74,45%, yang mengindikasikan peningkatan kemampuan model untuk membedakan antara kedua kelas. Disisi lain, *Random Forest* dengan 10 Fitur menunjukkan sedikit penurunan dalam *Accuracy* (75,75%) dan *Precision* (57,44%). Meskipun *Recall* tetap tinggi, *ROC AUC* juga mengalami penurunan menjadi 73,99%, yang menunjukkan bahwa pengurangan fitur yang banyak tidak cocok dalam klasifikasi SEO.

Jika dilihat dari tren metriks evaluasi, *Recall* meningkat secara konsisten seiring dengan berkurangnya jumlah fitur, yang menunjukkan bahwa model menjadi lebih sensitif terhadap identifikasi artikel dengan "Tidak Optimasi". *Precision* mengalami penurunan saat fitur yang digunakan semakin sedikit, yang mengindikasikan bahwa model cenderung mengklasifikasikan lebih banyak artikel "Optimasi" sebagai "Tidak Optimasi" saat menggunakan subset fitur yang lebih kecil. *ROC AUC* meningkat pada model dengan 14 fitur dan menurun pada model dengan 10 fitur. Ini menunjukkan bahwa dengan menggunakan fitur yang lebih relevan, model memiliki kemampuan yang lebih baik.

Secara umum, *Random Forest* dengan 14 fitur menunjukkan hasil yang sangat baik dalam *Recall* dan *ROC AUC*, meskipun ada sedikit penurunan dalam *Precision*. Melalui kinerja *ROC AUC* yang mencapai 74,45%, model ini juga menunjukkan kemampuan yang baik dalam membedakan antara artikel yang perlu "Optimasi" atau "Tidak Optimasi". Hal ini menegaskan

bahwa model ini mampu memisahkan kedua kelas secara lebih jelas dibandingkan dengan model yang menggunakan lebih banyak fitur. Oleh sebab itu, *Random Forest* dengan 14 fitur melalui pendekatan MI memberikan keseimbangan yang baik antara *Recall* dan *ROC AUC*, menjadikannya pilihan yang sangat efektif untuk tugas klasifikasi.

3.1.2 **Weighted Voting Ensemble** dengan Fitur Hasil *Mutual Information*

WVE memanfaatkan *voting* berbobot, di mana prediksi yang dihasilkan oleh model dengan kinerja lebih baik diberi bobot lebih besar dalam keputusan akhir. Pada kasus ini, WVE menggabungkan hasil dari *Random Forest* dan *XGBoost* yang masing-masing telah melalui *hyperparameter tuning* untuk memperoleh hasil terbaik. Model *Random Forest* yang digunakan pada WVE merupakan model *Random Forest* yang telah dilatih sebelumnya. Berikut ini hasil evaluasi model WVE dengan seleksi fitur menggunakan *Mutual Information*.

Tabel 6. Hasil Evaluasi Model WVE dengan Metode *Mutual Information* (MI)

| Model | Accuracy | Precision | Recall | ROC AUC |
|-------------------|------------------|------------------|------------------|------------------|
| WVE – Semua Fitur | 0,7643836 | 0,5890688 | 0,6736111 | 0,7380701 |
| WVE – 20 Fitur | 0,7657534 | 0,5900000 | 0,6828704 | 0,7417270 |
| WVE – 14 Fitur | 0,7616438 | 0,5807692 | 0,6990741 | 0,7435059 |
| WVE – 10 Fitur | 0,7513699 | 0,5662188 | 0,6828704 | 0,7315130 |

Berdasarkan Tabel 6, WVE dengan semua fitur menghasilkan *Accuracy* sebesar 76,44%, *Precision* 58,91%, *Recall* 67,36%, dan *ROC AUC* 73,81%. Jika dibandingkan dengan model dengan jumlah fitur yang lebih sedikit, model WVE dengan semua fitur masih belum cukup unggul. WVE dengan 20 fitur lebih baik dalam hal *Accuracy* (76,58%) dan *Precision* (59%) dibandingkan model dengan skenario lainnya yang menunjukkan ketepatannya dalam memprediksi kelas positif (artikel yang "Tidak Optimasi"). *ROC AUC* juga meningkat menjadi 74,17% dibandingkan WVE semua fitur, artinya model lebih baik dalam membedakan kelas positif dan negatif.

WVE dengan 14 Fitur memiliki *Accuracy* sedikit lebih rendah (76,16%) dibandingkan dengan model yang menggunakan 20 fitur, namun *Recall* meningkat menjadi 69,91%, yang menunjukkan bahwa model lebih sensitif dalam menangkap artikel dengan kelas "Tidak Optimasi". Meskipun *Precision* sedikit menurun (58,08%), *ROC AUC* meningkat menjadi 74,35%, menunjukkan bahwa model ini lebih baik dalam membedakan antara kelas positif dan negatif dibandingkan dengan model lainnya. Terakhir, WVE dengan 10 Fitur menunjukkan penurunan kinerja yang signifikan dengan *Accuracy* (75,13%) dan *Precision* (56,62%) yang lebih rendah. Meskipun *Recall* tetap relatif stabil (68,29%), *ROC AUC* mengalami penurunan menjadi 73,15%, menunjukkan bahwa penggunaan fitur yang terlalu sedikit mengurangi kemampuan model dalam membedakan kelas positif dan negatif.

Secara keseluruhan, berdasarkan penekanan pada metrik *ROC AUC*, WVE dengan 14 fitur adalah pilihan terbaik. Meskipun model ini menunjukkan *Accuracy* yang sedikit lebih rendah dibandingkan dengan model 20 fitur, *ROC AUC* yang lebih tinggi (74,35%) menunjukkan bahwa model ini lebih efektif dalam membedakan antara kelas Tidak Optimasi dan Optimasi SEO. Peningkatan *ROC AUC* ini menunjukkan bahwa model dengan 14 fitur mampu memisahkan kedua kelas dengan baik.

3.2 **Random Forest Feature Importance (RFFI)**

Seleksi fitur menggunakan *Random Forest Feature Importance* (RFFI) memanfaatkan algoritma *Random Forest* yang menggunakan ukuran *Gini Impurity*, yaitu metrik yang mengukur seberapa murni pembagian data pada setiap simpul dalam pohon keputusan.

Semakin rendah nilai *Gini Impurity*, semakin baik kemampuan fitur dalam membedakan kelas target, yaitu “Optimasi” atau “Tidak Optimasi”.

Sama halnya dengan proses MI, RFFI dimulai dengan memisahkan fitur penjelas dan target, lalu membagi data menjadi data latih dan data uji, serta normalisasi dengan *StandardScaler*. Selanjutnya, setiap fitur dianalisis untuk menentukan kontribusinya terhadap prediksi “Status SEO” dengan menghasilkan skor yang menggambarkan relevansi masing-masing fitur. Model *Random Forest* dan WVE dievaluasi berdasarkan seluruh fitur, 20 fitur, 14 fitur, dan 10 fitur. Pada Tabel 7, dijabarkan 20 fitur terpenting dari hasil seleksi.

Tabel 7. 20 Fitur Terpenting dari Hasil Seleksi Menggunakan RFFI

| No. | Fitur | Skor MI | No. | Fitur | Skor MI |
|-----|--------------------------|----------|-----|-----------------------------|----------|
| 1 | <i>Traffic</i> | 0,182795 | 11 | Similaritas URL_Meta | 0,030790 |
| 2 | Perubahan Peringkat | 0,096632 | 12 | Rata-rata Kata per Kalimat | 0,030407 |
| 3 | Kesulitan Kata Kunci | 0,060983 | 13 | Jumlah Kalimat pada Halaman | 0,030371 |
| 4 | Volume Pencarian | 0,045993 | 14 | Total Transfer (bytes) | 0,030348 |
| 5 | Ukuran (bytes) | 0,041667 | 15 | Similaritas Meta_H1 | 0,030303 |
| 6 | Jumlah Kata pada Halaman | 0,035206 | 16 | Similaritas URL_H1 | 0,030250 |
| 7 | Rasio Teks | 0,034093 | 17 | Similaritas Title_Meta | 0,028215 |
| 8 | Similaritas URL_Title | 0,032916 | 18 | <i>Outlinks</i> | 0,027041 |
| 9 | Waktu Respon | 0,032259 | 19 | Similaritas Title_H1 | 0,026934 |
| 10 | Skor Flesch Reading Ease | 0,031027 | 20 | Lebar Pixel Judul | 0,025811 |

3.2.1 *Random Forest* dengan Fitur Hasil RFFI

Berikut ini merupakan hasil evaluasi kinerja model *Random Forest* menggunakan seleksi fitur dengan *Random Forest Feature Importance* (RFFI).

Tabel 8. Hasil Evaluasi Kinerja *Random Forest* dengan RFFI

| Model | Accuracy | Precision | Recall | ROC AUC |
|------------------------------------|------------------|------------------|------------------|------------------|
| <i>Random Forest</i> – Semua Fitur | 0,7650685 | 0,5917526 | 0,6643519 | 0,7358724 |
| <i>Random Forest</i> – 20 Fitur | 0,7773973 | 0,6051081 | 0,7129630 | 0,7587188 |
| <i>Random Forest</i> – 14 Fitur | 0,7664384 | 0,5897436 | 0,6921296 | 0,7448975 |
| <i>Random Forest</i> – 10 Fitur | 0,7650685 | 0,5844402 | 0,7129630 | 0,7499640 |

Berdasarkan Tabel 8, *Random Forest* dengan semua fitur menghasilkan *Accuracy* 76,51%, *Precision* 59,18%, *Recall* 66,46%, dan *ROCAUC* 73,59%. Hal ini memberikan hasil yang cukup baik karena model mampu mendeteksi sebagian besar artikel “Tidak Optimasi”, namun terdapat beberapa kesalahan dalam mengklasifikasikan artikel yang memerlukan “Optimasi”. *Di samping itu*, performa *Random Forest* dengan 20 Fitur lebih baik dibandingkan dengan model lainnya dalam semua aspek, dengan *Accuracy* 77,74%, *Recall* meningkat menjadi 71,29%, dan *Precision* lebih tinggi (60,51%). Model dengan 20 fitur juga memiliki *ROC AUC* yang paling tinggi yaitu 75,87%, hal ini menunjukkan kemampuannya yang unggul dalam membedakan kelas positif dan negatif.

Random Forest dengan 14 Fitur menunjukkan *Accuracy* yang sedikit menurun (76,64%), dengan *Recall* dan *Precision* yang masih stabil. *ROC AUC* menurun menjadi 74,49%, menunjukkan bahwa model ini tidak sebagus model dengan 20 fitur dalam memisahkan kelas positif dan negatif. Di sisi lain, *Random Forest* dengan 10 fitur menunjukkan penurunan kinerja pada *Accuracy* (76,51%) dan *Precision* (58,44%). Namun, *Recall* tetap tinggi (71,29%) dan

ROC AUC juga tetap baik (74,99%) dibandingkan dengan model dengan 14 fitur maupun seluruh fitur. Secara keseluruhan, *Random Forest* dengan 20 fitur dari metode RFFI menjadi model yang terbaik dengan keseimbangan prediksi mencapai 75,87%. Penelitian pada beberapa skenario menunjukkan bahwa penggunaan fitur yang sangat banyak maupun sangat sedikit, memiliki performa yang kurang baik.

3.2.2 **Weighted Voting Ensemble (WVE) dengan Fitur Hasil RFFI**

Berikut ini merupakan hasil evaluasi model *Weighted Voting Ensemble* (WVE) menggunakan metode seleksi fitur dengan RFFI pada masing-masing skenario.

Tabel 9. Hasil Kinerja Model WVE dengan Seleksi Fitur Menggunakan RFFI

| Model | Accuracy | Precision | Recall | ROC AUC |
|-------------------|------------------|-----------|------------------|------------------|
| WVE – Semua Fitur | 0,7643836 | 0,5890688 | 0,6736111 | 0,7380701 |
| WVE – 20 Fitur | 0,7678082 | 0,5899420 | 0,7060185 | 0,7498964 |
| WVE – 14 Fitur | 0,7630137 | 0,5814394 | 0,7106481 | 0,7478338 |
| WVE – 10 Fitur | 0,7520548 | 0,5660377 | 0,6944444 | 0,7353545 |

Berdasarkan Tabel 9, model WVE dengan semua fitur menghasilkan *Accuracy* 76,44%, *Precision* 58,91%, *Recall* 67,36%, dan *ROC AUC* 73,81%. Model ini memberikan *Accuracy* yang baik, *Recall* dan *ROC AUC* sedikit lebih rendah dibandingkan dengan model-model yang menggunakan subset fitur yang lebih kecil yaitu 20 fitur dan 14 fitur. Hal ini menunjukkan bahwa meskipun menggunakan semua fitur, model ini tidak efektif dalam membedakan kelas “Tidak Optimasi” dan “Optimasi”. Ketika hanya menggunakan 20 fitur, model WVE meningkat dengan *Accuracy* 76,78%, *Recall* 70,60%, *Precision* tetap stabil di 58,99%, dan terjadi peningkatan pada *ROC AUC* menjadi 74,99%, menunjukkan model lebih seimbang dalam memprediksi kelas positif dan negatif.

Penggunaan 14 fitur pada model menunjukkan sedikit penurunan dalam *Accuracy* (76,30%) dan *Precision* (58,14%), namun *Recall* meningkat menjadi 71,06%, yang menandakan model lebih sensitif dalam mendeteksi artikel dengan “Tidak Optimasi”. *ROC AUC* juga mencapai 74,78%, menandakan model ini efektif dalam membedakan antara kedua kelas dibandingkan model dengan semua fitur dan 10 fitur. Sebaliknya, model dengan 10 fitur mengalami penurunan signifikan dalam kinerja, dengan *Accuracy* lebih rendah (75,21%) dan *Precision* yang menurun (56,66%). Meskipun *Recall* cukup baik (69,44%) dan *ROC AUC* hanya mencapai 73,54%, yang menunjukkan bahwa model ini kehilangan kemampuan dalam membedakan kedua kelas dengan optimal ketika hanya menggunakan fitur terbatas. Hasil ini lebih bagus dibandingkan dengan model semua fitur, yang memiliki *Recall* dan *ROC AUC* terendah.

Tabel 10. Hasil Perbandingan Kinerja Setiap Model dan Pendekatan Seleksi Fitur

| Model | Accuracy | Precision | Recall | ROC AUC |
|---|-----------|-----------|-----------|-----------|
| Tanpa Seleksi Fitur (Base Model) | | | | |
| Random Forest | 0,7650685 | 0,5917526 | 0,6643519 | 0,7358724 |
| WVE | 0,7643836 | 0,5890688 | 0,6736111 | 0,7380701 |
| Seleksi Fitur Menggunakan Mutual Information | | | | |
| Random Forest – 20 Fitur | 0,7630137 | 0,5860000 | 0,6782407 | 0,7384394 |
| Random Forest – 14 Fitur | 0,7630137 | 0,5830116 | 0,6990741 | 0,7444787 |
| Random Forest – 10 Fitur | 0,7575342 | 0,5744275 | 0,6967593 | 0,7399166 |
| WVE – 20 Fitur | 0,7657534 | 0,5900000 | 0,6828704 | 0,7417270 |
| WVE – 14 Fitur | 0,7616438 | 0,5807692 | 0,6990741 | 0,7435059 |
| WVE – 10 Fitur | 0,7513699 | 0,5662188 | 0,6828704 | 0,7315130 |
| Seleksi Fitur Menggunakan Random Forest Feature Importance | | | | |

| Model | Accuracy | Precision | Recall | ROC AUC |
|--------------------------|------------------|------------------|------------------|------------------|
| Random Forest – 20 Fitur | 0,7773973 | 0,6051081 | 0,7129630 | 0,7587188 |
| Random Forest – 14 Fitur | 0,7664384 | 0,5897436 | 0,6921296 | 0,7448975 |
| Random Forest – 10 Fitur | 0,7650685 | 0,5844402 | 0,7129630 | 0,7499640 |
| WVE – 20 Fitur | 0,7678082 | 0,5899420 | 0,7060185 | 0,7498964 |
| WVE – 14 Fitur | 0,7630137 | 0,5814394 | 0,7106481 | 0,7478338 |
| WVE – 10 Fitur | 0,7520548 | 0,5660377 | 0,6944444 | 0,7353545 |

Berdasarkan hasil evaluasi model pada Tabel 10, model *Random Forest* dengan 20 Fitur dari *Random Forest Feature Importance* (RFFI) menunjukkan kinerja yang lebih baik dibandingkan dengan model yang menggunakan *Mutual Information* (MI) atau seluruh fitur. Keunggulan ini berasal dari kemampuan *Random Forest* untuk secara efektif menangani interaksi non-linear antar fitur, serta kemampuannya dalam memilih fitur yang memiliki kontribusi terbesar terhadap penurunan ketidakmurnian (*Impurity*) pada pohon keputusan. RFFI memberikan informasi yang lebih jelas tentang fitur yang benar-benar penting untuk klasifikasi, sementara MI hanya mengukur ketergantungan statistik antara fitur dan target tanpa mempertimbangkan interaksi kompleks antar fitur. Dengan menggunakan hanya 20 fitur terpenting, model ini mampu mencapai *Accuracy* (77,74%) dan *Precision* (60,51%) yang paling tinggi dan optimal.

Penerapan 20 fitur terbaik yang dipilih melalui RFFI juga mengurangi kompleksitas model, menghindari *overfitting*, dan meningkatkan efisiensi dalam pengambilan keputusan. Jumlah fitur yang lebih sedikit menjadikan model lebih fokus pada fitur yang paling relevan, yang pada akhirnya meningkatkan *Recall* dan *ROC AUC* (75,87%), menunjukkan bahwa model ini lebih baik dalam membedakan antara kelas "Tidak Optimasi" dan "Optimasi". Pengurangan kompleksitas ini juga mempercepat waktu pelatihan dan membuat model lebih mudah diinterpretasikan, tanpa mengorbankan akurasi prediksi.

4. KESIMPULAN

Hasil seleksi fitur menggunakan *Random Forest Feature Importance* (RFFI) berhasil mengidentifikasi 20 fitur utama yang relevan dalam konteks optimasi SEO. Fitur-fitur tersebut yaitu *Traffic*, perubahan peringkat, kesulitan kata kunci, volume pencarian, dan fitur terpilih lainnya memiliki pengaruh besar terhadap kemampuan halaman untuk menentukan apakah perlu dioptimasi atau tidak. Model *Random Forest* dengan 20 fitur yang terpilih melalui RFFI menunjukkan *Accuracy* mencapai 77,74%, *Precision* sebesar 60,51%, *Recall* sebesar 71,29%, dan *ROC AUC* mencapai 75,87%. Model ini efektif dalam membedakan antara halaman yang memerlukan optimasi SEO dan yang tidak. Melalui pemilihan fitur-fitur yang paling relevan, model ini tidak hanya meningkatkan akurasi prediksi tetapi juga mengurangi kompleksitas model, menghindari *overfitting*, dan meningkatkan efisiensi dalam proses seleksi dan pemodelan. Oleh karena itu, *Random Forest* dengan 20 fitur memberikan solusi yang optimal dan efisien untuk memprediksi halaman-halaman yang perlu dioptimasi SEO, sesuai dengan tujuan penelitian untuk menciptakan model yang akurat dan efisien.

Meskipun penelitian ini menunjukkan hasil yang baik dalam mengintegrasikan kajian *Natural Language Processing* (NLP) dalam klasifikasi halaman web, pendekatan ini masih dapat ditingkatkan melalui beberapa strategi tambahan. Strategi yang dapat dilakukan meliputi penggunaan teknik representasi fitur yang lebih canggih berbasis transformer dan menerapkan teknik seleksi fitur lainnya seperti *Recursive Feature Elimination* (RFE) atau *Feature Importance* dari model *Gradient Boosted Tree*.

DAFTAR RUJUKAN

- Alduailij, M., Khan, Q. W., Tahir, M., Sardaraz, M., Alduailij, M., & Malik, F. (2022). Machine-Learning-Based DDoS Attack Detection Using *Mutual Information* and *Random Forest Feature Importance* Method. *Symmetry*, 14(6), 1095. <https://doi.org/10.3390/sym14061095>
- Alfiana, F., Khofifah, N., Ramadhan, T., Septiani, N., Wahyuningsih, W., Azizah, N. N., & Ramadhona, N. (2023). Apply the Search Engine Optimization (SEO) Method to determine Website Ranking on Search Engines. *International Journal of Cyber and IT Service Management*, 3(1), 65–73. <https://doi.org/10.34306/ijcitsm.v3i1.126>
- Aryani, D., Shine Pintor Siolemba Patiro, Setiawan, A., & Tjahjono, B. (2023). Comparative Analysis Of On-Page And Off-Page White Hat Search Engine Optimization (SEO) Techniques On Website Popularity. *International Journal of Science, Technology & Management*, 4(3), 527–533. <https://doi.org/10.46729/ijstm.v4i3.815>
- Badri, N., Kboubi, F., & Chaibi, A. H. (2022). Combining *FastText* and *Glove* Word Embedding for Offensive and Hate speech Text Detection. *Procedia Computer Science*, 207, 769–778. <https://doi.org/10.1016/j.procs.2022.09.132>
- Beraha, M., Metelli, A. M., Papini, M., Tirinzoni, A., & Restelli, M. (2019). *Feature Selection via Mutual Information: New Theoretical Insights* (No. arXiv:1907.07384). arXiv. <https://doi.org/10.48550/arXiv.1907.07384>
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, 135–146. https://doi.org/10.1162/tacl_a_00051
- Capitaine, L., Genuer, R., & Thiébaut, R. (2021). *Random Forests* for high-dimensional longitudinal data. *Statistical Methods in Medical Research*, 30(1), 166–184. <https://doi.org/10.1177/0962280220946080>
- Carrington, A. M., Manuel, D. G., Fieguth, P. W., Ramsay, T., Osmani, V., Wernly, B., Bennett, C., Hawken, S., Magwood, O., Sheikh, Y., McInnes, M., & Holzinger, A. (2023). Deep ROC Analysis and AUC as Balanced Average *Accuracy*, for Improved Classifier Selection, Audit and Explanation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2022.3145392>
- Chakravarthy, S. R., & Rajaguru, H. (2022). Ensemble-Based Weighted *Voting* Approach for the Early Diagnosis of Diabetes Mellitus. Dalam P. Karrupusamy, V. E. Balas, & Y. Shi (Ed.), *Sustainable Communication Networks and Application* (hlm. 451–460). Springer Nature. https://doi.org/10.1007/978-981-16-6605-6_33

- Disha, R. A., & Waheed, S. (2022). *Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique.*
- Dogan, A., & Birant, D. (2019). A Weighted Majority Voting Ensemble Approach for Classification. *IEEE Access*.
- Fatima, S., Hussain, A., Amir, S. B., Ahmed, S. H., & Aslam, S. M. H. (2023). XGBoost and Random Forest Algorithms: An in Depth Analysis. *Pakistan Journal of Scientific Research*, 3(1), 26–31. <https://doi.org/10.57041/pjusr.v3i1.946>
- Hastuti, R. P., Riona, V., & Hardiyanti, M. (2023). Content Retrieval dengan FastText Word Embedding pada Learning Management System Olimpiade. *Journal of Internet and Software Engineering*, 4(1), 18–22. <https://doi.org/10.22146/jise.v4i1.6766>
- Hoilijoki, S., Kilpua, E. K. J., Osmane, A., Kalliokoski, M. M. H., George, H., Savola, M., & Asikainen, T. (2022). Using Mutual Information to investigate non-linear correlation between AE index, ULF Pc5 wave activity and electron precipitation. *Frontiers in Astronomy and Space Sciences*, 9, 987913. <https://doi.org/10.3389/fspas.2022.987913>
- Khomsah, S., Ramadhani, R. D., & Wijaya, S. (2022). The Accuracy Comparison Between Word2Vec and FastText On Sentiment Analysis of Hotel Reviews. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 6(3), 352–358. <https://doi.org/10.29207/resti.v6i3.3711>
- Laarne, P., Zaidan, M. A., & Nieminen, T. (2021). *ennemi: Non-linear correlation detection with Mutual Information*.
- Matošević, G., Dobša, J., & Mladenović, D. (2021). Using Machine Learning for Web Page Classification in Search Engine Optimization. *Future Internet*, 13(1), 9. <https://doi.org/10.3390/fi13010009>
- Mienye, I. D., & Sun, Y. (2022). A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects. *IEEE Access*, 10, 99129–99149. <https://doi.org/10.1109/ACCESS.2022.3207287>
- Naidu, G., Zuva, T., & Sibanda, E. M. (2023). A Review of Evaluation Metrics in Machine Learning Algorithms. Dalam R. Silhavy & P. Silhavy (Ed.), *Artificial Intelligence Application in Networks and Systems* (hlm. 15–25). Springer International Publishing. https://doi.org/10.1007/978-3-031-35314-7_2
- Naseem, U., Razzak, I., & Eklund, P. W. (2021). A survey of pre-processing techniques to improve short-text quality: A case study on hate speech detection on twitter. *Multimedia*

Tools and Applications, 80(28–29), 35239–35266. <https://doi.org/10.1007/s11042-020-10082-6>

Osamor, V. C., & Okezie, A. F. (2021). Enhancing the weighted *voting* ensemble algorithm for tuberculosis predictive diagnosis. *Scientific Reports*, 11(1), 14806. <https://doi.org/10.1038/s41598-021-94347-6>

Pandey, P., & Pandeya, Y. R. (2023). *Machine Learning Techniques for Web Page Classification with Search Engine Optimization*. 8(2).

Probst, P., Wright, M., & Boulesteix, A.-L. (2019). *Hyperparameters and Tuning Strategies for Random Forest*. *WIREs Data Mining and Knowledge Discovery*, 9(3), e1301. <https://doi.org/10.1002/widm.1301>

Rachita, R., & Pandey, S. (2024). A Comprehensive Examination of Search Engine Optimization (SEO) Audit and Optimization Strategies. *International Scientific Journal of Engineering and Management*, 03(05), 1–9. <https://doi.org/10.55041/ISJEM01723>

Sohil, F., Sohali, M. U., & Shabbir, J. (2022). An introduction to statistical learning with applications in R: By Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, New York, Springer Science and Business Media, 2013, \$41.98, eISBN: 978-1-4614-7137-7. *Statistical Theory and Related Fields*, 6(1), 87–87. <https://doi.org/10.1080/24754269.2021.1980261>

Xie, X.-R., Yuan, M.-J., Bai, X.-T., Gao, W., & Zhou, Z.-H. (2023). On the Gini-*Impurity* Preservation For Privacy *Random Forests*. *Advances in Neural Information Processing Systems*.

Zaidan, M. A., Dada, L., Alghamdi, M. A., Al-Jeelani, H., Lihavainen, H., Hyvärinen, A., & Hussein, T. (2019). *Mutual Information Input Selector and Probabilistic Machine Learning Utilisation for Air Pollution Proxies*.

Ziakis, C., Vlachopoulou, M., Kyrkoudis, T., & Karagkiozidou, M. (2019). Important Factors for Improving Google Search Rank. *Future Internet*, 11(2), 32. <https://doi.org/10.3390/fi11020032>