



# **Analisis Perbandingan Algoritma Random Forest dan K-Nearest Neighbors pada Klasifikasi Tingkat Stres Pekerja**

**Syalom Kristian Manurung\*, Irfan Pratama**

Fakultas Teknologi Informasi, Sistem Informasi, Universitas Mercu Buana Yogyakarta, Yogyakarta, Indonesia

Email: <sup>1,\*</sup>syalomkristian09@gmail.com, <sup>2</sup>irfanp@mercubuana-yogya.ac.id

Email Penulis Korespondensi: syalomkristian09@gmail.com

Submitted: 15/06/2025; Accepted: 01/09/2025; Published: 02/09/2025

**Abstrak**—Stres kerja merupakan isu signifikan dalam dunia profesional modern karena dapat berdampak pada penurunan produktivitas, kualitas kerja, dan kesejahteraan mental karyawan. Penelitian ini bertujuan untuk mengevaluasi serta membandingkan performa dua algoritma *machine learning*, yaitu *Random Forest* dan *K-Nearest Neighbors (KNN)*, dalam mengklasifikasikan tingkat stres kerja. Data yang digunakan diperoleh melalui kuesioner daring dari 212 responden yang bekerja di berbagai sektor di Indonesia. Data tersebut kemudian dikonversi dari skala *Likert* ke bentuk numerik, dikelompokkan menggunakan metode *K-Means*, dan dikategorikan ke dalam lima tingkat stres, mulai dari tidak stres hingga stres berat. Untuk mengatasi ketidakseimbangan distribusi data, digunakan metode *Synthetic Minority Over-sampling Technique (SMOTE)*. Pemodelan dilakukan dalam tiga skenario pembagian data, yaitu 90:10, 80:20, dan 70:30, serta dievaluasi menggunakan metrik akurasi, presisi, *recall*, *f1-score*, dan *cross-validation*. Hasil penelitian menunjukkan bahwa algoritma *Random Forest* secara konsisten memberikan kinerja yang lebih unggul dibandingkan algoritma KNN pada semua skenario pembagian data. Setelah penerapan SMOTE, kedua algoritma mengalami peningkatan performa, dengan *Random Forest-Balanced* mencatat akurasi dan *f1-score* tertinggi sebesar 92% pada skenario 70:30. Temuan ini menunjukkan bahwa kombinasi antara *Random Forest* dan SMOTE merupakan pendekatan yang efektif dalam klasifikasi tingkat stres kerja serta memiliki potensi untuk dikembangkan sebagai sistem deteksi dini stres kerja yang objektif dan efisien.

**Kata Kunci:** Stres Kerja; Random Forest; K-Nearest Neighbors; SMOTE; Klasifikasi

**Abstract**—Work stress has become a prominent concern in the modern professional landscape, as it can lead to reduced productivity, diminished work quality, and decreased mental well-being among employees. This study aims to evaluate and compare the performance of two machine learning algorithms, namely Random Forest and K-Nearest Neighbors (KNN), in classifying levels of work stress. The data were obtained through an online questionnaire completed by 212 respondents from various employment sectors in Indonesia. The responses were converted from Likert scale to numerical values, grouped using the K-Means clustering method, and categorized into five levels of stress, ranging from no stress to very high stress. To address data imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied. The modeling process was conducted using three different data split scenarios, namely 90:10, 80:20, and 70:30, and evaluated using metrics such as accuracy, precision, recall, f1-score, and cross-validation. The findings indicate that the Random Forest algorithm consistently outperformed KNN across all scenarios. After applying SMOTE, both algorithms showed improved performance, with the Balanced Random Forest model achieving the highest accuracy and f1-score of 92 percent in the 70:30 scenario. These results suggest that combining Random Forest with SMOTE offers an effective and reliable solution for classifying work stress levels and could be developed as an objective and efficient early detection system.

**Keywords:** Work Stress; Random Forest; K-Nearest Neighbors; SMOTE; Classification

## **1. PENDAHULUAN**

Setiap lingkungan kerja dituntut untuk mengelola sumber daya manusia secara profesional untuk mengoptimalkan kinerja pekerja sebagai aset utama perusahaan. Kinerja dipengaruhi berbagai faktor, termasuk lingkungan kerja, sehingga pengelolaan tenaga kerja perlu diarahkan untuk membangun suasana kerja yang mendukung kesejahteraan pekerja dan mengatasi stres kerja sebagai tantangan utama [1]. Stres kerja merupakan kondisi psikologis yang dialami karyawan akibat adanya tekanan dalam menyelesaikan tuntutan pekerjaan yang melebihi kapasitas atau sumber daya yang dimiliki [2].

Menurut World Health Organization (WHO), stres merupakan suatu keadaan yang muncul sebagai respons terhadap interaksi antara individu dan lingkungan yang dianggap menimbulkan ketidakseimbangan antara tuntutan situasi dengan kemampuan individu untuk mengatasinya. Keadaan ini melibatkan faktor-faktor biologis, psikologis, dan sosial. Secara umum, stres dipahami sebagai tekanan atau gangguan yang bersifat tidak menyenangkan dan berasal dari luar diri seseorang [3]. Tingkat stres kerja yang tinggi memberikan dampak negatif terhadap produktivitas karyawan serta berpengaruh signifikan terhadap kinerja. Semakin rendah tingkat stres yang dirasakan, maka kinerja cenderung meningkat, sebaliknya, semakin tinggi tingkat stres, maka kinerja akan mengalami penurunan, yang pada akhirnya dapat menghambat pencapaian tujuan perusahaan [4]. Berdasarkan laporan State of the Global Workplace yang dirilis oleh Gallup, pada tahun 2022 tercatat bahwa 44% pekerja secara global mengalami stres secara rutin. Angka ini sama dengan tahun sebelumnya 2021 dan merupakan tingkat stres tertinggi dalam lebih dari sepuluh tahun terakhir. Di kawasan Asia Tenggara, Filipina mencatat tingkat stres tertinggi dengan 45% responden menyatakan sering mengalami stres, disusul oleh Myanmar, Thailand, dan Kamboja yang masing-masing menunjukkan angka sebesar 39%. Sementara itu, Indonesia tercatat sebagai negara dengan tingkat stres pekerja terendah di kawasan tersebut, dengan hanya 21% pekerja yang melaporkan sering merasa stres pada tahun 2022 [5]. Fakta ini menunjukkan



urgensi pengembangan metode yang efektif untuk memantau dan mengelola stres kerja guna menjaga kualitas hidup dan kinerja karyawan.

Pendekatan dalam identifikasi stres kerja secara konvensional, seperti wawancara masih umum digunakan dalam berbagai penelitian. Penelitian yang dilakukan oleh Nurtika pada tahun 2024 di PT Perkebunan Nusantara I Kebun Baru Kota Langsa menunjukkan bahwa tingkat stres kerja pada karyawan tergolong rendah, dengan skor rata-rata sebesar 1.90. Sementara itu, kinerja karyawan berada dalam kategori sangat tinggi dengan skor 4.29 [4]. Sejalan dengan temuan tersebut, Amri pada tahun 2020 dalam penelitiannya di Toko Jawa Timur Kota Sorong juga menyatakan bahwa tingkat stres kerja karyawan berada pada kategori rendah, ditunjukkan oleh skor rata-rata 2.33 [6]. Namun, kedua penelitian tersebut belum menerapkan metode klasifikasi otomatis berbasis *machine learning*. Di sisi lain, beberapa penelitian mulai menggunakan algoritma *machine learning* untuk mendeteksi stres, meskipun belum secara khusus membahas stres kerja. Astari menggunakan algoritma *Random Forest* untuk mengklasifikasikan stres fisiologis saat tidur dan berhasil mencapai akurasi sebesar 93,65% [7]. Anisa menerapkan algoritma *K-Nearest Neighbors (KNN)* untuk mendeteksi stres mahasiswa berdasarkan indikator psikologis, dengan akurasi mencapai 91,58% [8]. Sementara itu, penelitian yang dilakukan oleh Adi menunjukkan bahwa Random Forest memperoleh nilai evaluasi sebesar 93%, unggul dibandingkan KNN yang hanya mencapai 81% dalam klasifikasi penyakit gagal jantung [9].

Berbagai studi sebelumnya membuktikan keandalan algoritma *Random Forest* dalam tugas klasifikasi di berbagai domain. Mahing menerapkannya pada data teks Twitter untuk klasifikasi tingkat stres dan mencapai akurasi sebesar 80% [10]. Dalam konteks klasifikasi bencana, Rahmasari menunjukkan bahwa *Random Forest* unggul dengan akurasi 95,59% pada prediksi status banjir, mengungguli KNN yang memiliki performa terendah [11]. Sementara itu, Rahman melaporkan bahwa *Random Forest* mencapai akurasi sebesar 100% dalam klasifikasi data Iris, sedangkan KNN hanya mencapai 97% [12]. Selanjutnya, Rofi dan Hasan dengan dua kasus penelitian yang berbeda menunjukkan bahwa *Random Forest* unggul dalam hal stabil dan akurat terkait klasifikasi hasil belajar dan lama studi [13], [14]. Konsistensi ini menunjukkan bahwa *Random Forest* memiliki keunggulan dalam hal akurasi, stabilitas, dan kemampuan generalisasi dibandingkan KNN, menjadikannya kandidat yang kuat untuk digunakan dalam klasifikasi data kompleks seperti tingkat stres kerja.

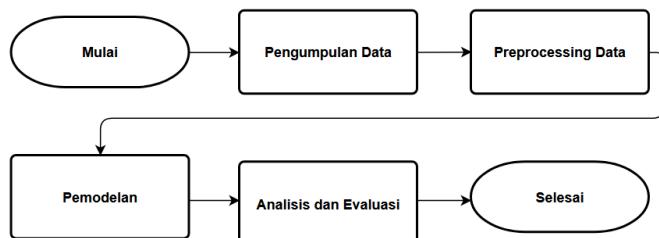
Meskipun kedua algoritma, *Random Forest* dan *K-Nearest Neighbors*, telah terbukti memberikan hasil klasifikasi yang baik dalam berbagai studi, hingga kini belum banyak penelitian yang secara khusus membandingkan kinerjanya dalam konteks identifikasi tingkat stres di lingkungan kerja. Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengisi kekosongan tersebut dengan mengevaluasi dan membandingkan efektivitas kedua algoritma dalam memprediksi tingkat stres kerja. Fokus utama meliputi pengolahan data, pengujian performa melalui metrik *akurasi*, *presisi*, *recall*, dan *f1-score*, serta penyusunan rekomendasi algoritma terbaik untuk diterapkan dalam sistem pemantauan stres kerja berbasis teknologi. Pendekatan ini diharapkan menjadi alternatif yang lebih objektif dan efisien dibandingkan metode konvensional.

Sebagai respons terhadap hal tersebut, penelitian ini dilakukan untuk menyoroti dan membandingkan efektivitas algoritma *Random Forest* serta *K-Nearest Neighbors (KNN)* dalam mengklasifikasikan tingkat stres kerja, menggunakan data kuesioner yang diperoleh dari pekerja di Indonesia. Studi ini juga berupaya mengidentifikasi kelebihan dan kekurangan masing-masing algoritma dalam konteks klasifikasi psikologis yang kompleks. Hal ini akan memungkinkan penentuan algoritma yang lebih efektif untuk mengklasifikasikan tingkat stres kerja di Indonesia, sekaligus menunjukkan kinerja kedua algoritma berdasarkan metrik evaluasi yang relevan.

## 2. METODOLOGI PENELITIAN

### 2.1 Alur Penelitian

Penelitian ini dilaksanakan melalui serangkaian proses sistematis, yang mencakup pengumpulan data, preprocessing, pemodelan, serta analisis dan evaluasi hasil. Alur lengkap proses tersebut disajikan pada Gambar 1.



**Gambar 1.** Alur lengkap penelitian.

Gambar 1 menyajikan alur proses penelitian yang dimulai dari tahapan awal hingga evaluasi model. Proses dimulai dengan pengumpulan data dari sumber yang relevan, kemudian dilanjutkan dengan tahapan preprocessing untuk mempersiapkan data sebelum dilakukan pemodelan. Setelah model dibangun menggunakan algoritma yang sesuai, dilakukan analisis dan evaluasi untuk menilai kinerja model.



## 2.2 Pengumpulan Data

Pengumpulan data merupakan langkah krusial dalam suatu penelitian yang dilakukan untuk memperoleh informasi yang diperlukan sebagai dasar analisis [15]. Pada penelitian ini, jenis data yang digunakan adalah data primer. Pengambilan data primer dilakukan melalui pengisian kuesioner digital yang disebarluaskan secara online menggunakan Google Forms. Responden yang menjadi sasaran adalah individu yang secara aktif bekerja di berbagai sektor di Indonesia.

Instrumen yang digunakan dalam penelitian ini dirancang dengan merujuk pada indikator stres kerja yang diadaptasi dari teori Robbins yang menyebutkan stres kerja terdiri dari tiga aspek utama, yaitu faktor individu, organisasi, dan lingkungan [16]. Kuesioner yang disebarluaskan terdiri dari dua bagian utama, yaitu informasi demografis responden serta 20 butir pernyataan yang berkaitan dengan tingkat stres kerja yang dialami.

Untuk mengukur jawaban responden terhadap setiap pernyataan dalam kuesioner, penelitian ini menggunakan format jawaban berupa skala *Likert*. Skala ini merupakan metode pengukuran yang umum dipakai dalam studi sosial guna mengetahui opini, sikap, atau persepsi seseorang terhadap suatu topik tertentu. Melalui pendekatan ini, responden diminta untuk memilih satu dari lima tingkat persetujuan, yaitu: a) Sangat Setuju, b) Setuju, c) Netral/Ragu-ragu, d) Tidak Setuju, dan e) Sangat Tidak Setuju. Masing-masing pilihan mencerminkan tingkat persetujuan responden dan diberi skor tertentu yang selanjutnya digunakan dalam tahap analisis data [17].

## 2.3 Preprocessing Data

Data *Preprocessing* adalah proses mengonversi data yang awalnya tidak terstruktur menjadi format yang lebih terorganisir agar dapat diolah lebih lanjut sesuai dengan kebutuhan [18]. Proses *preprocessing* pada penelitian ini terdapat empat tahapan utama yang saling berkaitan. Setiap tahapan dilakukan secara sistematis untuk mempersiapkan data agar layak digunakan dalam tahap pemodelan dan evaluasi, sekaligus meminimalisasi bias yang mungkin muncul akibat ketidakseimbangan atau ketidakteraturan data mentah.

### 2.3.1 Transformasi Data Likert

Selaras dengan penjabaran sebelumnya, tahap pertama dalam proses *preprocessing* merupakan tahapan transformasi data mentah ke dalam format numerik, yaitu dengan mengonversi setiap jawaban responden pada skala *Likert* lima poin. Dalam hal ini, respons kualitatif dikodekan sebagai berikut: "Sangat Tidak Setuju" diberikan skor 1, "Tidak Setuju" skor 2, "Netral" skor 3, "Setuju" skor 4, dan "Sangat Setuju" skor 5 [17].

### 2.3.2 Klustering K-Means

Tahap kedua dalam penelitian ini melibatkan proses klaster menggunakan algoritma *K-Means* untuk mengelompokkan data berdasarkan skor total yang mencerminkan tingkat stres kerja. *K-Means* merupakan teknik pengelompokan non-hierarkis yang membagi data ke dalam sejumlah klaster dengan mempertimbangkan kesamaan karakteristik antar data. Tujuan utama dari metode ini adalah untuk meminimalkan variasi data dalam setiap klaster serta memaksimalkan perbedaan antar kluster [19].

Untuk mendukung interpretasi hasil pengelompokan tersebut, dilakukan analisis statistika deskriptif. Analisis deskriptif merupakan salah satu teknik dalam pendekatan kuantitatif yang digunakan untuk memperoleh gambaran secara menyeluruh terhadap data sampel yang telah dikumpulkan dalam penelitian [20]. Analisis ini mencakup pengukuran nilai pusat data seperti *mean*, *median*, *modus*, serta kuartil bawah dan kuartil atas. Penyajian informasi dalam penelitian ini dituangkan dalam bentuk tabel guna menampilkan karakteristik utama dari setiap klaster, sehingga memudahkan pemahaman pola dan perbedaan antar kelompok secara informatif. Setelah klaster terbentuk, dilakukan pengurutan berdasarkan rata-rata skor dari yang terendah hingga tertinggi agar pengelompokan tersusun secara sistematis sesuai tingkat stres responden. Pengurutan ini memudahkan pemetaan setiap klaster ke kategori tingkat stres yang relevan.

### 2.3.3 Pelabelan

Setelah data tingkat stres kerja berhasil dikelompokkan menggunakan algoritma K-Means menjadi lima klaster. Masing-masing klaster kemudian diurutkan berdasarkan rata-rata skor dari yang terendah hingga tertinggi. Tahapan penting selanjutnya adalah proses pelabelan, yang bertujuan untuk memberikan makna interpretatif terhadap setiap klaster hasil pengelompokan. Pelabelan ini merujuk pada kategori tingkat stres yang lazim digunakan dalam skala DASS-21, meskipun perlu dicatat bahwa penelitian ini tidak menggunakan kuesioner DASS-21 secara langsung. Skala DASS-21 (Depression Anxiety Stress Scales) merupakan alat ukur psikologis yang terdiri atas tiga subskala, yaitu depresi, kecemasan, dan stres. Dalam konteks penelitian ini, hanya subskala stres yang dijadikan acuan klasifikasi. Subskala stres pada DASS-21 membagi tingkat stres ke dalam lima kategori, yaitu: Normal, Ringan, Sedang, Parah, dan Sangat Parah. [21]. Dengan adanya lima kategori stres yang sudah terbentuk dan terurut secara sistematis, setiap klaster akan dipetakan ke dalam lima kategori stres.

### 2.3.4 Teknik Penyeimbangan SMOTE

Setelah proses pelabelan dilakukan dan setiap klaster dimasukkan ke dalam kategori tingkat stres tertentu, langkah selanjutnya adalah mempersiapkan data untuk kemungkinan adanya keterdistribusian antar kategori. Oleh karena itu,



diterapkan teknik *Synthetic Minority Over-sampling Technique (SMOTE)* sebagai bagian dari proses *preprocessing* data. *SMOTE* adalah metode yang sering digunakan untuk menangani masalah data yang tidak seimbang dengan cara menghasilkan sampel sintetis baru pada kelas minoritas sehingga jumlah data antar kelas menjadi lebih seimbang [22]. Pada tahap ini, dilakukan penyesuaian distribusi data agar setiap kategori tingkat stres memiliki jumlah data yang lebih seimbang. Prosedur ini penting untuk mengurangi bias yang mungkin terjadi akibat ketidakseimbangan jumlah sampel antar kategori, sehingga data yang dihasilkan lebih representatif untuk analisis selanjutnya. Dengan demikian, tahap ini membantu memastikan bahwa model yang dikembangkan dapat memproses data secara optimal tanpa dipengaruhi oleh ketidakseimbangan kelas yang signifikan.

## 2.4 Pemodelan

Pemodelan adalah tahapan penting untuk membangun model klasifikasi atau prediksi berdasarkan data yang telah diproses melalui tahap *preprocessing* [23]. Pada penelitian ini, tahapan pemodelan meliputi pemilihan *parameter* algoritma dan pelatihan model. Dua algoritma utama yang digunakan, yaitu *Random Forest* dan *K-Nearest Neighbors (KNN)*.

### 2.4.1 Random Forest

*Random Forest* adalah algoritma hasil pengembangan dari metode *Decision Tree* yang membangun sejumlah pohon keputusan, di mana tiap pohon dilatih dengan sampel data yang berbeda dan atribut yang dipilih secara acak dari subset tertentu. Algoritma ini memiliki berbagai keunggulan, seperti mampu meningkatkan akurasi meskipun terdapat data yang tidak lengkap, tahan terhadap *outlier*, serta efisien dalam penggunaan memori. Selain itu, *Random Forest* juga dilengkapi dengan fitur seleksi otomatis yang dapat mengidentifikasi atribut paling berpengaruh untuk meningkatkan kinerja model klasifikasi [24].

### 2.4.2 K-Nearest Neighbors

*K-Nearest Neighbors (KNN)* merupakan salah satu metode klasifikasi yang bekerja dengan mengukur kedekatan suatu data terhadap data lainnya berdasarkan nilai atribut yang dimiliki [25]. Salah satu rumus umum yang digunakan untuk menghitung jarak tersebut adalah *Euclidean Distance*. *Euclidean Distance* merupakan metode untuk mengukur jarak kedekatan antara dua variabel, yang digunakan untuk menunjukkan tingkat kesamaan antara dua citra. Semakin kecil nilai jarak yang dihitung, maka tingkat kemiripan antara kedua citra tersebut semakin tinggi [26]. Algoritma KNN dipilih karena proses pelatihannya yang cepat, konsepnya yang mudah dipahami, serta kemampuannya yang efektif saat diterapkan pada *dataset* berukuran besar [25]. Alur algoritma *K-Nearest Neighbor (KNN)* adalah sebagai berikut:

- Menentukan nilai  $k$  sebagai jumlah tetangga terdekat yang akan digunakan dalam klasifikasi.
- Hitung jarak antara data baru dengan seluruh data latih menggunakan rumus Euclidean berikut:

$$d_{(x_i x_j)} = \sqrt{\sum_{r=1}^n (x_i - x_j)^2} \quad (1)$$

Jarak  $d_{(x_i x_j)}$  adalah ukuran kedekatan antara data latih  $x_i$  dan data uji  $x_j$ , di mana  $r$  menunjukkan indeks data dan  $n$  adalah jumlah dimensi atau fitur pada data.

- Susun data latih berdasarkan jarak terkecil ke yang terbesar, kemudian pilih  $k$  data yang paling dekat.
- Identifikasi kelas dari data tetangga yang telah terpilih.
- Prediksi kelas data baru berdasarkan kelas mayoritas dari tetangga terdekat tersebut [27], [28].

## 2.5 Analisis dan Evaluasi

### 2.5.1 Pembagian Data Skenario dan Analisis Performa

Analisis hasil eksperimen ini secara mendalam mengkaji akurasi model klasifikasi dengan meninjau dua kondisi dataset yang berbeda, yaitu data mentah dan data yang telah diseimbangkan (balanced). Evaluasi performa model dilakukan melalui tiga skenario pembagian data yang bervariasi, yaitu 90:10, 80:20, dan 70:30. Prioritas utama analisis ini adalah mengungkap pengaruh setiap proporsi pembagian data terhadap tingkat akurasi model, serta membandingkan konsistensi akurasi antara penggunaan data asli dan data yang telah mengalami proses balancing. Pembagian data 90% untuk pelatihan dan 10% untuk pengujian pada penelitian Anisa, dirancang untuk memaksimalkan data yang digunakan dalam pelatihan, sehingga model dapat lebih efektif mempelajari pola yang ada, meskipun dengan uji data yang lebih sedikit untuk menguji kinerja [8]. Dalam penelitian yang dilakukan oleh Rofi, pembagian data sebesar 80% untuk pelatihan dan 20% untuk pengujian dimaksudkan agar model mendapatkan lebih banyak informasi selama proses belajar, sementara tetap menyediakan jumlah data yang memadai untuk menguji kinerja model secara tepat [13]. Pembagian data dengan proporsi 70% untuk pelatihan dan 30% untuk pengujian dalam studi Astari memberikan keseimbangan yang ideal antara jumlah data yang dimanfaatkan untuk melatih model sekaligus menyediakan porsi data yang memadai guna menilai performa secara komprehensif [7].

### 2.5.2 Confusion Matrix

*Confusion Matrix* adalah representasi dalam bentuk tabel yang berfungsi untuk menilai kinerja suatu klasifikasi model, khususnya dalam menguji kemampuan model dalam membedakan dan memprediksi kelas dari data uji. Matriks ini



digunakan sebagai indikator dalam proses evaluasi algoritma pembelajaran mesin yang digunakan untuk menyelesaikan permasalahan klasifikasi. Di dalamnya terdapat empat elemen utama, yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*, yang masing-masing merepresentasikan hasil prediksi model terhadap kondisi aktual. Struktur dasar matriks konfusi ditunjukkan pada tabel berikut [29].

**Tabel 1.** Hasil perbandingan akurasi tiga skenario

Kelas Aktual	Kelas Prediksi	
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Melalui informasi yang diperoleh dari matriks konfusi, sejumlah evaluasi metrik dapat dihitung guna menilai kinerja klasifikasi model secara menyeluruh. Beberapa metrik penting yang digunakan dalam evaluasi ini antara lain akurasi, presisi, *recall* dan *f1-score*, di mana masing-masing memiliki peran tersendiri dalam mengukur efektivitas model prediksi. Uraian mengenai metrik-metrik tersebut dijelaskan pada bagian berikut [30]:

a. Akurasi (Accuracy)

Akurasi menunjukkan model kemampuan dalam menghasilkan prediksi yang tepat secara keseluruhan terhadap seluruh data yang diuji.

$$\text{Rumus Akurasi} = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (2)$$

b. Recall

*Recall* menilai efektivitas model dalam mendekripsi seluruh sampel positif yang sebenarnya ada dalam data.

$$\text{Rumus Recall} = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

c. Presisi (Precision)

Presisi mengukur presisi model ketika memutuskan suatu sampel sebagai positif, yaitu seberapa banyak prediksi positif yang memang benar.

$$\text{Rumus Presisi} = \frac{TP}{TP+FP} \times 100\% \quad (4)$$

d. F1-Score

*F1-score* adalah rata-rata harmonik dari nilai presisi dan recall, yang digunakan untuk merepresentasikan keseimbangan antara kedua metrik evaluasi tersebut.

$$\text{Rumus F1-score} = \frac{2 \times \text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \times 100\% \quad (5)$$

### 2.5.3 Cross-Validation

*Cross-validation* merupakan salah satu pendekatan evaluasi model yang digunakan untuk menguji keandalan hasil analisis prediktif [31]. Salah satu metode validasi silang yang sering digunakan adalah *k-fold cross-validation*, yang membagi *dataset* ke dalam *k* subset berukuran sama. [32]. Metode ini berfungsi untuk menilai kapasitas model dalam menangani data di luar sampel pelatihan serta memberikan dasar yang valid dalam menentukan kesesuaian model untuk digunakan [33]. Studi ini menggunakan *k-fold cross-validation* dengan nilai *k*=5. Penggunaan *k-fold cross-validation* dengan nilai *k*=5 bertujuan untuk mengurangi bias yang mungkin timbul dalam data [32].

## 3. HASIL DAN PEMBAHASAN

### 3.1 Hasil Pengumpulan Data

Proses pengumpulan data menghasilkan respon dari 212 partisipan yang semuanya saat ini bekerja di Indonesia. Setiap responden memberikan jawaban lengkap untuk seluruh item dalam instrumen, tanpa ada data yang hilang atau tidak terisi dalam keseluruhan *dataset*. Data yang dikumpulkan terdiri dari 22 kolom, yang meliputi informasi identifikasi responden serta serangkaian pernyataan terkait pengukuran tingkat stres kerja. Pada tahapan ini, *dataset* telah tersusun lengkap dan telah siap untuk diproses lebih lanjut. Tabel 2 sampel menampilkan data sebagai representasi dari keseluruhan pengumpulan data.

**Tabel 2.** Data hasil kuesioner.

No	Email	Nama	Q1	Q2	Q3-Q19	Q20
1	Resp1	R1	Sangat Setuju	Setuju	...	Setuju
2	Resp2	R2	Netral	Setuju	...	Netral
...	...	...	...	...	...	...
211	Resp211	R211	Sangat Setuju	Sangat Setuju	...	Netral



No	Email	Nama	Q1	Q2	Q3-Q19	Q20
212	Resp212	R212	Netral	Netral	...	Sangat Tidak Setuju

### 3.2 Preprocessing Data

Pada tahap ini, data yang dikumpulkan dari responden telah melalui proses pengolahan awal dengan mengubah setiap jawaban pada skala *Likert* menjadi bentuk numerik. Selanjutnya, nilai total skor diperoleh dengan menjumlahkan skor dari butir pertanyaan Q1 hingga Q20 pada kuesioner yang digunakan untuk mengukur tingkat stres kerja. Nilai total tersebut menjadi representasi kuantitatif dari tingkat stres masing-masing individu. Rangkuman hasil konversi skor ini disajikan pada Tabel 3.

**Tabel 3.** Hasil Konversi.

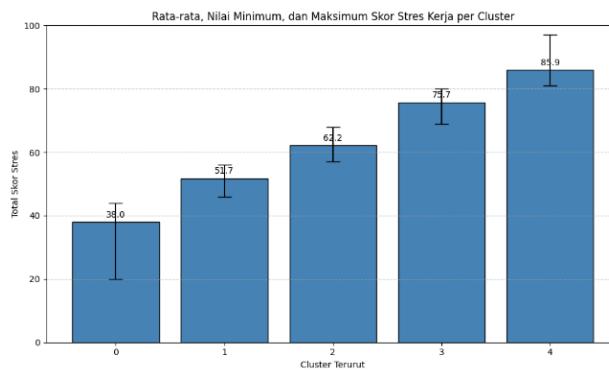
No	Email	Nama	Q1	Q2	Q3-Q19	Q20	Total Skor
1	Resp1	R1	5	4	...	4	79
2	Resp2	R2	3	4	...	3	77
...	...	...	...	...	...	...	...
211	Resp211	R211	4	4	...	4	79
212	Resp212	R212	3	3	...	1	42

Dari Tabel 3 terlihat adanya variasi skor total tingkat stres yang cukup signifikan antar responden. Variasi ini menunjukkan adanya perbedaan tingkat stres kerja dalam populasi yang diteliti. Skor total ini kemudian digunakan sebagai dasar utama untuk mengelompokkan tingkat stres secara lebih sistematis. Proses klasterisasi dengan metode *K-Means* membagi responden ke dalam lima klaster berdasarkan tingkat stres kerja. Hasil klasterisasi ini disajikan pada Tabel 4, yang menunjukkan nilai rata-rata atau *mean*, nilai minimum, dan nilai maksimum skor total pada masing-masing klaster.

**Tabel 4.** Hasil klasterisasi *K-Means*.

Klaster	Jumlah Data	Mean	Min	Max
0	28	37.964286	20.0	44.0
1	21	51.714286	46.0	56.0
2	38	62.157895	57.0	68.0
3	56	75.714286	69.0	80.0
4	69	85.927536	81.0	97.0

Hasil pada Tabel 4 mengungkapkan perbedaan mencolok dalam distribusi skor stres responden, yang berhasil dikelompokkan ke dalam lima klaster utama. Klaster 0, yang terdiri dari 28 responden, memiliki nilai *Mean* 37.96 dengan rentang skor 20.0-44.0, merepresentasikan kelompok tidak stres atau tingkat stres normal. Klaster 1, berisi 21 responden, menunjukkan *Mean* 51.71 dengan rentang 46.0-56.0, mengindikasikan tingkat stres rendah. Klaster 2, dengan 38 responden, memiliki *Mean* 62.16 dan rentang 57.0-68.0, menggambarkan kelompok dengan tingkat stres sedang. Klaster 3, terdiri dari 56 responden, menunjukkan *Mean* 75.71 dengan rentang 69.0-80.0, diasosiasikan dengan tingkat stres berat. Terakhir, Klaster 4, klaster terbesar dengan 69 responden, memiliki *Mean* 85.93 dan rentang 81.0-97.0, merepresentasikan kelompok dengan tingkat stres sangat berat. Hasil klasterisasi ini secara jelas memperlihatkan variasi tingkat stres kerja pada populasi responden, yang dapat dijadikan dasar untuk penanganan dan intervensi yang disesuaikan bagi setiap kelompok. Perbedaan karakteristik pada masing-masing klaster tersebut divisualisasikan pada Gambar 2, yang menampilkan distribusi nilai rata-rata (*mean*) skor total tingkat stres.



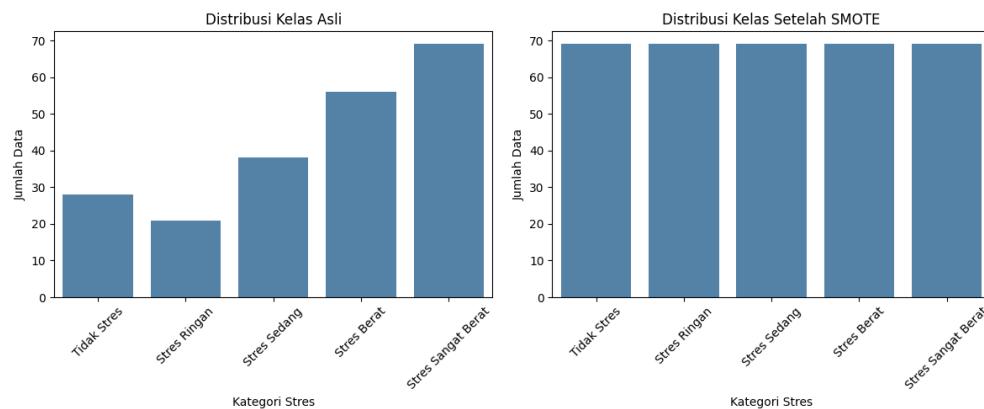
**Gambar 2.** Rata-rata, nilai minimum dan maksimum skor stres kerja per-klaster.

Berdasarkan hasil klasterisasi pada Tabel 3, dibuat tabel baru yang mengelompokkan rentang skor total tingkat stres kerja ke dalam kategori yang lebih mudah dipahami. Tabel ini menyajikan batas minimum dan maksimum skor untuk setiap kategori stres, yaitu tidak stres, stres rendah, stres sedang, dan stres berat, stres sangat berat.

**Tabel 5.** Kategori tingkat stres berdasarkan klaster.

Klaster	Jumlah Data	Rentang Total Skor	Kategori Stres
0	28	20-44	Tidak Stres
1	21	46-56	Stres Rendah
2	38	57-68	Stres Sedang
3	56	69-80	Stres Berat
4	69	81-97	Stres Sangat Berat

Melihat distribusi jumlah responden pada masing-masing kategori tingkat stres pada Tabel 5, ditemukan adanya ketidakseimbangan yang cukup signifikan antar kelompok. Kategori "Tidak Stres" hanya memiliki 28 data, jumlah yang jauh lebih sedikit jika dibandingkan dengan kategori "Stres Sangat Berat" yang mencapai 69 responden. Ketidakseimbangan ini berpotensi menimbulkan bias dalam proses klasifikasi, karena model cenderung mengoptimalkan performa pada kategori dengan jumlah data terbanyak, sehingga mengurangi akurasi prediksi untuk kategori yang lebih sedikit datanya. Untuk mengatasi permasalahan ke distribusi data antar kategori tingkat stres, diterapkan teknik Synthetic Minority Over-sampling Technique (SMOTE). Perbandingan distribusi data sebelum dan sesudah penerapan SMOTE ditampilkan pada Gambar 3.

**Gambar 3.** Perbandingan Distribusi Kelas Asli dan Kelas setelah SMOTE.

Gambar 3 memperlihatkan bahwa penerapan metode SMOTE berhasil menyamakan jumlah data pada setiap kategori tingkat stres menjadi 69 data per kelas. Akibatnya, total data mengalami peningkatan dari semula 212 menjadi 345. Temuan ini mengindikasikan bahwa SMOTE mampu menangani masalah ketidakseimbangan kelas secara efektif, sehingga menghasilkan dataset yang lebih merata dan siap digunakan untuk proses pelatihan model klasifikasi.

### 3.3 Pemodelan

Pada penelitian ini, algoritma *Random Forest* menggunakan konfigurasi dengan jumlah *n\_estimators* sebanyak 200 dan *max\_depth* sebesar 10. Pengaturan tersebut dipilih untuk menjaga keseimbangan antara tingkat akurasi dan kompleksitas model.

Sedangkan untuk algoritma KNN, nilai *k* = 5 dipilih berdasarkan pertimbangan teoritis. Jika nilai *k* yang dipilih terlalu kecil, maka model cenderung terlalu peka terhadap data yang menyimpang (*noise*), sehingga prediksi yang dihasilkan menjadi kurang konsisten. Sebaliknya, apabila nilai *k* terlalu besar, maka terlalu banyak tetangga dari kelas lain dapat mempengaruhi proses klasifikasi, menyebabkan informasi dari data yang paling relevan menjadi kurang berpengaruh dan berpotensi menurunkan tingkat akurasi [14].

### 3.4 Analisis dan Evaluasi

Pada langkah ini, model klasifikasi *Random Forest* dan *K-Nearest Neighbors (KNN)* diuji pada data asli serta data yang telah diseimbangkan dengan metode SMOTE. Data kemudian dipisah menjadi set pelatihan dan pengujian dengan perbandingan 90%:10%, 80%:20%, dan 70%:30%. Selain itu, dilakukan juga *Cross-validation* untuk mengevaluasi seberapa baik model dapat menggeneralisasi data baru. Nilai akurasi dan *f1-score* untuk setiap kondisi tersebut ditampilkan pada Tabel 6.

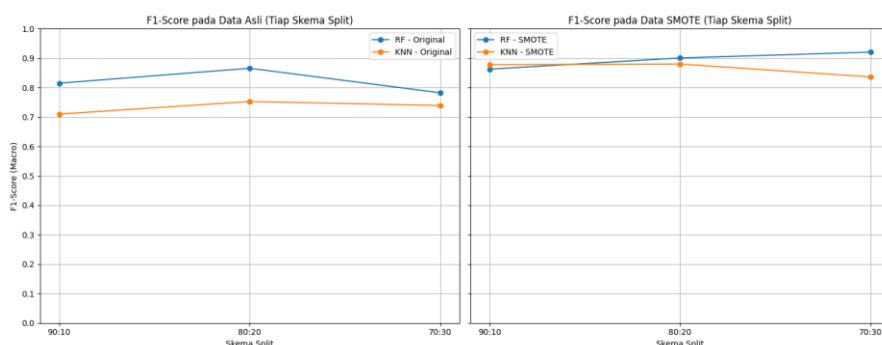
**Tabel 6.** Hasil perbandingan akurasi tiga skenario.

Algoritma	Skenario 1		Skenario 2		Skenario 3	
	90:10		80:20		70:30	
	Akurasi	<i>F1-score</i>	Akurasi	<i>F1-score</i>	Akurasi	<i>F1-score</i>
Random Forest	0.86	0.82	0.88	0.87	0.83	0.78
K-Nearest Neighbors	0.77	0.71	0.77	0.75	0.80	0.74
Random Forest-Balanced	0.89	0.86	0.90	0.90	0.92	0.92

Algoritma	Skenario 1		Skenario 2		Skenario 3	
	90:10		80:20		70:30	
	Akurasi	F1-score	Akurasi	F1-score	Akurasi	F1-score
K-Nearest Neighbors-Balanced	0.89	0.88	0.88	0.88	0.84	0.84

Analisis akurasi menunjukkan bahwa pada data asli tanpa keseimbangan, algoritma Random Forest secara konsisten memiliki akurasi yang lebih tinggi dibandingkan algoritma K-Nearest Neighbors di seluruh skenario pembagian data. Random Forest mencapai nilai tertinggi sebesar 0.88 (88%) pada skenario 80:20, sedangkan K-Nearest Neighbors memperoleh akurasi tertinggi sebesar 0.80 (80%) pada skenario 70:30. Penerapan metode Synthetic Minority Over-sampling Technique (SMOTE) terbukti efektif dalam meningkatkan kinerja kedua algoritma. Pada skenario 90:10, baik Random Forest Balanced maupun K-Nearest Neighbors Balanced menunjukkan peningkatan akurasi menjadi 0.89 (89%). Selanjutnya Random Forest Balanced mencatat peningkatan bertahap dengan akurasi sebesar 0.90 (90%) pada skenario 80:20 dan mencapai performa puncak pada skenario 70:30 dengan akurasi sebesar 0.92 (92%). Hasil ini menunjukkan bahwa metode SMOTE memiliki peran krusial dalam optimalisasi akurasi model, khususnya pada algoritma Random Forest yang secara signifikan mengungguli kinerja model tanpa penyeimbangan data. Selain itu, distribusi data yang tidak seimbang sejak awal dapat menyebabkan paradoks akurasi, yaitu kondisi di mana nilai akurasi tampak tinggi namun tidak mewakili kinerja model secara menyeluruh akibat bias terhadap kelas sebagian besar. Temuan ini selaras dengan penelitian yang dilakukan oleh Utami, yang menekankan pentingnya penyeimbangan kelas sebagai strategi utama dalam meningkatkan validitas dan kejelasan hasil klasifikasi [34]. Oleh karena itu, *f1-score* digunakan sebagai metrik utama karena mampu merepresentasikan keseimbangan antara presisi dan recall secara lebih adil, terutama saat menghadapi distribusi kelas yang tidak seimbang.

Untuk mendukung interpretasi terhadap hasil dari proses klasifikasi yang disajikan pada Tabel 6, visualisasi grafis disediakan guna memperjelas perbandingan *f1-score* antar algoritma pada berbagai skenario pengujian, baik pada *dataset* asli maupun *dataset* yang telah diseimbangkan menggunakan metode SMOTE. Visualisasi tersebut ditampilkan pada Gambar 4.



Gambar 4. Perbandingan Akurasi RF dan KNN pada data asli.

Gambar 4 menyajikan perbandingan nilai *f1-score* (*macro*) antara algoritma *Random Forest* (RF) dan *K-Nearest Neighbors* (KNN) pada dua jenis *dataset*, yaitu data asli (tanpa penyeimbangan) dan data yang telah diproses menggunakan teknik Synthetic Minority Over-sampling Technique (SMOTE). Evaluasi dilakukan berdasarkan tiga skenario pembagian data pelatihan dan pengujian, yaitu 90:10, 80:20, dan 70:30. Pada data asli, RF menunjukkan kinerja terbaik pada skenario 80:20 dengan *f1-score* tertinggi sekitar 0,87, sedangkan performa menurun pada skenario 70:30 menjadi sekitar 0,78. KNN pada data asli menunjukkan performa yang lebih stabil, dengan nilai *f1-score* berada di kisaran 0,71 hingga 0,75. Sementara itu, pada data hasil SMOTE, RF menunjukkan tren peningkatan performa secara konsisten, dari sekitar 0,87 (90:10), menjadi 0,90 (80:20), dan mencapai sekitar 0,92 pada skenario 70:30. Di sisi lain, KNN juga mengalami peningkatan pada skenario 80:20, namun nilainya menurun kembali pada skema 70:30. Secara keseluruhan, RF menunjukkan kinerja yang lebih baik dan stabil dibandingkan KNN, terutama pada data yang telah diseimbangkan menggunakan SMOTE.

Berdasarkan hasil visualisasi dan data pada Tabel 5, berikut ini dijelaskan secara terperinci performa masing-masing algoritma pada tiga skenario pengujian yang telah diterapkan, beserta hasil *confusion matrix* dan *cross-validation* sebagai bagian dari evaluasi model secara menyeluruh.

a. Skenario 90:10 (Proporsi Data Uji Sangat Terbatas)

Pada skenario pembagian data adalah 90% dan pelatihan dan 10% adalah pengujian, algoritma Random Forest menunjukkan nilai akurasi sebesar 0,86 (86%) dan f1-score sebesar 0,82 (82%). Sementara itu, algoritma K-Nearest Neighbors mencatat akurasi sebesar 0,77 (77%) dan f1-score sebesar 0,71 (71%). Setelah dilakukan penyeimbangan data, performa kedua algoritma mengalami peningkatan. Random Forest Balanced mencapai akurasi sebesar 0,89 (89%) dan f1-score sebesar 0,86 (86%), sedangkan K-Nearest Neighbors Balanced mencapai akurasi yang sama, yaitu sebesar 0,89 (89%), dan f1-score sebesar 0,88 (88%). Hasil ini menunjukkan bahwa penyeimbangan data tidak hanya memperkuat kinerja Random Forest, tetapi juga memberikan peningkatan yang signifikan terhadap kinerja K-Nearest Neighbors. Temuan ini sejalan dengan hasil penelitian yang dilakukan oleh



Adi, yang membandingkan kinerja kedua algoritma pada skenario serupa dan melaporkan bahwa f1-score Random Forest mencapai 0.93 (93%), lebih tinggi dibandingkan K-Nearest Neighbors yang hanya memperoleh nilai sebesar 0.81 (81%) [9].

b. Skenario 80:20 (Proporsi Data Uji Umum)

Selanjutnya pada skenario pembagian data sebesar 80% untuk pelatihan dan 20% untuk pengujian, Random Forest menunjukkan akurasi tertinggi pada data asli, yaitu sebesar 0.88 (88%), sementara K-Nearest Neighbors mencatat nilai akurasi yang konsisten sebesar 0.77 (77%). Setelah penerapan penyeimbangan data, kinerja kedua algoritma meningkat. Nilai Random Forest Balanced sebesar 0,90 (90%) dan nilai K-Nearest Neighbors Balanced sebesar 0,88 (88%). Temuan ini mengindikasikan bahwa teknik penyeimbangan data memberikan kontribusi positif terhadap peningkatan kinerja model, khususnya pada proporsi data uji yang lebih besar. Penelitian yang dilakukan oleh Rahman mendukung hasil tersebut, yang mana pada tugas klasifikasi data bunga Iris dengan skenario pembagian data 80:20, Random Forest memperoleh akurasi sebesar 1.00 (100%) dan f1-score sebesar 1.00 (100%), sementara K-Nearest Neighbors mencatat akurasi sebesar 0.97 (97%) dan f1-score sebesar 0.98 (98%) [12]. Meskipun konteks data yang digunakan berbeda, kesamaan dalam kebutuhan akan akurasi dan keseimbangan prediksi menunjukkan bahwa Random Forest secara umum memiliki kinerja yang lebih stabil dan andal dibandingkan K-Nearest Neighbors.

c. Skenario 70:30 (Proporsi Data Uji Lebih Besar)

Pada skenario terakhir, dengan proporsi data pelatihan sebesar 70% dan data pengujian sebesar 30%, kinerja Random Forest pada data asli mengalami penurunan, dengan akurasi sebesar 0.83 (83%). Sebaliknya, K-Nearest Neighbors menunjukkan peningkatan performa dengan akurasi sebesar 0.80 (80%). Setelah dilakukan penyeimbangan data, kedua algoritma menunjukkan performa terbaik. Random Forest Balanced memperoleh akurasi sebesar 0.92 (92%), sedangkan K-Nearest Neighbors Balanced mencatat akurasi sebesar 0.84 (84%). Hasil ini menegaskan bahwa dalam kondisi jumlah data latih yang lebih terbatas, penerapan teknik penyeimbangan data menjadi semakin penting untuk mempertahankan efektivitas model klasifikasi. Temuan ini selaras dengan penelitian yang dilakukan oleh Rahmasari, yang menunjukkan bahwa Random Forest mencatat nilai f1-score tertinggi di antara beberapa algoritma yang diuji, sementara K-Nearest Neighbors berada pada posisi terbawah berdasarkan metrik yang sama [11]. Secara keseluruhan, Random Forest menunjukkan konsistensi performa yang unggul, terutama dalam menjaga keseimbangan antara presisi dan recall pada data yang bersifat kompleks dan tidak seimbang.

d. Evaluasi Performa Model

Untuk memberikan gambaran yang lebih komprehensif mengenai performa algoritma *Random Forest* dan *K-Nearest Neighbors (KNN)*, selain metrik *akurasi*, dilakukan evaluasi menggunakan metrik presisi, *recall*, dan *f1-score*. Tabel 7 menampilkan perbandingan nilai presisi, *recall*, dan *f1-score* dari kedua algoritma baik pada data asli maupun data yang telah diseimbangkan.

**Tabel 7.** Perbandingan presisi, *recall*, dan *f1-Score*

Algoritma	Skenario	Presisi	Recall	F1-Score
Random Forest	90:10	0.89	0.85	0.82
K-Nearest Neighbors	90:10	0.80	0.78	0.71
Random Forest-Balanced	90:10	0.88	0.90	0.86
K-Nearest Neighbors-Balanced	90:10	0.93	0.87	0.88
Random Forest	80:20	0.89	0.88	0.87
K-Nearest Neighbors	80:20	0.81	0.80	0.75
Random Forest-Balanced	80:20	0.90	0.91	0.90
K-Nearest Neighbors-Balanced	80:20	0.90	0.89	0.88
Random Forest	70:30	0.81	0.77	0.78
K-Nearest Neighbors	70:30	0.83	0.78	0.74
Random Forest-Balanced	70:30	0.92	0.93	0.92
K-Nearest Neighbors-Balanced	70:30	0.86	0.84	0.84

Analisis performa menunjukkan bahwa Random Forest (RF) secara konsisten unggul dalam tugas klasifikasi, terutama setelah penerapan teknik SMOTE. Pada data asli, RF mencatat f1-score tertinggi sebesar 0.87 di skenario pembagian data 80:20, meskipun menurun menjadi 0.78 pada 70:30. Sebaliknya, KNN menunjukkan performa yang lebih rendah dan stabil di kisaran 0.71 hingga 0.75. Setelah data diseimbangkan, kedua algoritma mengalami peningkatan signifikan. RF-Balanced menunjukkan lonjakan performa yang stabil di seluruh skenario, dengan f1-score mencapai 0.92 pada 70:30. KNN-Balanced juga membaik, terutama pada 90:10 dan 80:20 dengan f1-score 0.88, meski sedikit menurun pada 70:30. Hasil ini menegaskan bahwa penyeimbangan data tidak hanya meningkatkan akurasi, tetapi juga memperkuat konsistensi model. Secara keseluruhan, RF-Balanced tampil sebagai algoritma yang paling tangguh dan seimbang dalam menangani variasi distribusi data.

e. Evaluasi Confusion Matrix

Sebagai pelengkap evaluasi performa model, disajikan *confusion matrix* pada Tabel 8 yang menampilkan confusion matrix dari algoritma *Random Forest* dengan data yang telah diseimbangkan menggunakan teknik



*SMOTE* pada skenario pembagian data 70:30. Pemilihan skenario ini didasarkan pada hasil analisis performa sebelumnya yang menunjukkan bahwa kombinasi tersebut menghasilkan nilai *f1-score* tertinggi dibandingkan skenario lainnya. Penyajian *confusion matrix* ini bertujuan untuk memberikan gambaran rinci mengenai ketepatan klasifikasi model terhadap setiap kelas serta mengevaluasi keseimbangan antara prediksi yang benar dan kesalahan yang terjadi.

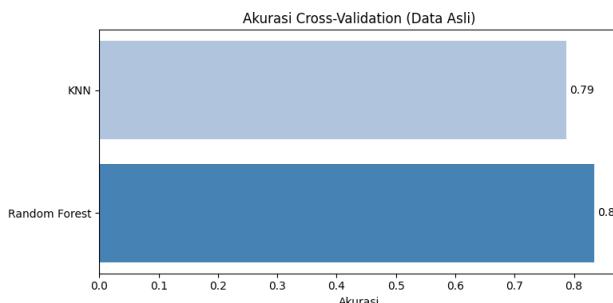
**Tabel 8.** Confusion matrix algoritma Random Forest dengan *SMOTE* pada skema 70:30.

Kelas Aktual	Kelas Prediksi				
	Tidak Stres	Stres Ringan	Stres Sedang	Stres Berat	Stres Sangat Berat
Tidak Stres	21	1	0	0	0
Stres Ringan	1	25	0	0	0
Stres Sedang	0	0	16	0	0
Stres Berat	0	0	3	20	2
Stres Sangat Berat	0	0	0	1	14

Jika merujuk pada Tabel 8, sebagian besar prediksi model berada pada diagonal utama, yang menunjukkan kemampuan model dalam mengklasifikasikan setiap kategori stres secara akurat. Meski demikian, masih ditemukan beberapa kesalahan klasifikasi, terutama pada kelas "Stres Berat" yang sering tertukar dengan "Stres Sedang" dan "Stres Sangat Berat". Kesalahan serupa, meskipun dalam jumlah kecil, juga terjadi antara kelas "Tidak Stres" dan "Stres Ringan", serta antara "Stres Sangat Berat" dan "Stres Berat". Hal ini mengindikasikan adanya kemiripan karakteristik antar kelas yang berdekatan dan menjadi tantangan tersendiri dalam proses klasifikasi.

#### f. Cross-validation

Untuk menilai kemampuan model dalam memberikan hasil yang konsisten pada data baru, pengujian menggunakan metode 5-Fold *Cross-Validation* telah dilakukan. Hasil visualisasi dari proses ini disajikan pada Gambar 5 di bawah ini.



**Gambar 5.** Perbandingan akurasi cross-validation RF dan KNN pada data asli.

Gambar 5 menyajikan visualisasi akurasi algoritma *Random Forest* (RF) dan *K-Nearest Neighbors* (KNN) yang dievaluasi melalui penerapan metode *cross-validation* pada data asli. Observasi menunjukkan bahwa *Random Forest* mencapai akurasi *cross-validation* sebesar 0.83, nilai yang secara komparatif lebih tinggi. Sementara itu, *K-Nearest Neighbors* menunjukkan akurasi *cross-validation* sebesar 0.79. Hasil ini secara konsisten mengindikasikan bahwa *Random Forest* memiliki kapabilitas generalisasi dan performa klasifikasi yang superior pada data asli dibandingkan dengan KNN, sebagaimana divalidasi secara objektif melalui prosedur 5-Fold *Cross-Validation*.

### 3.5 Limitasi Penelitian

Terdapat sejumlah keterbatasan dalam studi ini yang patut dicermati. Pertama, jumlah data yang digunakan terbatas pada 212 responden, sehingga dapat mempengaruhi kemampuan model generalisasi terhadap populasi yang lebih luas. Meskipun teknik SMOTE telah diterapkan untuk menyeimbangkan distribusi data, ukuran sampel yang relatif kecil tetap menjadi tantangan dalam menjamin validitas hasil. Kedua, penelitian ini hanya membandingkan dua algoritma, yaitu *Random Forest* dan *K-Nearest Neighbors*, tanpa melibatkan metode lain seperti *Support Vector Machine* (*SVM*), *Naive Bayes*, dan *Neural Network* yang berpotensi menghasilkan performa klasifikasi yang lebih baik.

## 4. KESIMPULAN

Hasil analisis menunjukkan bahwa algoritma *Random Forest* (RF) unggul dalam kinerja klasifikasi dibandingkan dengan *K-Nearest Neighbors* (KNN) dalam mengidentifikasi tingkat stres kerja karyawan. Pada data asli, RF menunjukkan akurasi dan *f1-score* yang lebih tinggi dibandingkan KNN di semua skenario pembagian data (90:10, 80:20, dan 70:30). Penerapan teknik *Synthetic Minority Over-sampling Technique* (*SMOTE*) terbukti mampu



meningkatkan performa kedua algoritma, terutama *Random Forest* yang mencapai puncak *akurasi* sebesar 0.92 dan *f1-score* 0.92 pada skenario 70:30. Sementara itu, meskipun performa KNN juga mengalami peningkatan setelah penyeimbangan data, algoritma ini menunjukkan kecenderungan penurunan kinerja ketika proporsi data uji semakin besar. Secara umum, penyeimbangan data terbukti menjadi langkah penting dalam membangun model klasifikasi yang akurat dan adil, khususnya saat menghadapi distribusi kelas yang tidak merata. Temuan ini memperkuat bahwa *Random Forest-Balanced* merupakan algoritma yang paling direkomendasikan untuk klasifikasi tingkat stres kerja karena mampu memberikan hasil yang stabil, akurat, dan konsisten di berbagai kondisi data. Penelitian ini sekaligus menjawab rumusan masalah dan tujuan yang telah ditetapkan, yaitu membandingkan efektivitas dua algoritma *machine learning* dalam klasifikasi stres kerja serta menentukan pendekatan yang paling optimal untuk diterapkan pada data psikologis karyawan. Dengan demikian, penggunaan algoritma *Random Forest* yang didukung oleh teknik penyeimbangan SMOTE dapat menjadi solusi berbasis teknologi yang efektif untuk monitoring dan penanganan stres kerja secara lebih objektif, efisien, dan sistematis di lingkungan profesional.

## UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih dan penghargaan setinggi-tingginya kepada Ibu Nia Kusuma Wardhani, M.Psi., Psikolog, atas kesediaannya memberikan masukan dan validasi terhadap instrumen kuesioner yang digunakan dalam penelitian ini. Kontribusi beliau, sebagai pihak yang memiliki keahlian di bidang psikologi, sangat membantu dalam memastikan kesesuaian dan kelayakan instrumen secara teoritis maupun praktis. Masukan yang diberikan telah meningkatkan kualitas serta validitas data penelitian, sehingga mendukung pencapaian hasil yang lebih akurat dan dapat dipertanggungjawabkan secara ilmiah.

## REFERENCES

- [1] J. Heruwanto, R. Wahyuningsih, R. Rasipan, and E. Nurpatria, "Pengaruh Lingkungan Kerja dan Stres Kerja terhadap Kinerja Karyawan pada PT Nusamulti Centralestari Tangerang," *Jurnal Manajemen Kewirausahaan*, vol. 17, no. 1, pp. 69–78, Jun. 2020, doi: 10.33370/jmk.v17i1.391.
- [2] A. A. P. Mangkunegara, *Manajemen Sumber Daya Manusia*. Bandung: PT. Remaja Rosdakarya, 2017.
- [3] N. Azizah, F. P. Idris, and A. Asrina, "Hubungan Umur dengan Stres Kerja pada Pedagang New Makassar Mall Kota Makassar," *Window of Public Health Journal*, vol. 4, no. 2, pp. 297–304, Apr. 2023, doi: 10.33096/woph.v4i4.765.
- [4] A. D. Nurtika, A. Deli, and N. Juliaviani, "Pengaruh Stress Kerja Terhadap Kinerja Karyawan Pada PT Perkebunan Nusantara I Kebun Baru Kota Langsa (The Influence of Work Stress on Employee Performance at PT Perkebunan Nusantara I Kebun Baru Langsa City)," *Jurnal Ilmiah Mahasiswa Pertanian*, vol. 9, no. 3, pp. 165–172, Aug. 2024, doi: 10.17969/jimfp.v9i3.31588.
- [5] H. Akbar *et al.*, "Hubungan Umur, Masa Kerja dan Tuntutan Kerja dengan Stres Kerja Pada Karyawan PDAM di Kabupaten X," *Promotif: Jurnal Kesehatan Masyarakat*, vol. 14, no. 1, pp. 1–6, Jun. 2024, doi: 10.56338/promotif.v14i1.5518.
- [6] I. Amri, Ashar, and I. Lagat, "Analisis Tingkat Stres Kerja Pada Karyawan di Toko Jawa Timur Kota Sorong," *Metode Jurnal Teknik Industri*, vol. 6, no. 2, pp. 46–54, Oct. 2020, doi: 10.33506/mt.v6i2.1642.
- [7] D. F. Astari, Y. H. Chrisnanto, and Melina, "Klasifikasi Tingkat Stres Saat Tidur Menggunakan Algoritma Random Forest," *Jurnal Mahasiswa Teknik Informatika*, vol. 7, no. 5, pp. 3676–3684, Oct. 2023, doi: 10.36040/jati.v7i5.7750.
- [8] S. Anisa, A. Komarudin, and E. Ramadhan, "Sistem Klasifikasi untuk Menentukan Tingkat Stress Mahasiswa secara Umum Menggunakan Metode K-Nearest Neighbors," *Jurnal Informatika Teknologi dan Sains (JINTEKS)*, vol. 6, no. 3, pp. 568–578, Aug. 2024, doi: 10.51401/jinteks.v6i3.4317.
- [9] S. Adi and A. Wintarti, "Komparasi Metode Support Vector Machine (SVM), K-Nearest Neighbors (KNN), dan Random Forest (RF) untuk Prediksi Penyakit Gagal Jantung," *MATHunesa: Jurnal Ilmiah Matematika*, vol. 10, pp. 258–268, Jul. 2022, doi: 10.26740/mathunesa.v10n2.p258-268.
- [10] N. F. Mahing, A. L. Gunawan, A. F. A. Zen, A. B. Fitra, and S. A. Wicaksono, "Klasifikasi Tingkat Stres dari Data Berbentuk Teks dengan Menggunakan Algoritma Support Vector Machine (SVM) dan Random Forest," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 11, no. 5, pp. 1067–1076, Oct. 2024, doi: 10.25126/jtik2024118010.
- [11] F. Rahmasari, M. Rifany, T. Priharyanto, and R. Kurniawan, "Perbandingan Algoritma Machine Learning dalam Klasifikasi Status Banjir di Sumatera Utara," *Seminar Nasional Sains Data*, vol. 4, no. 1, pp. 307–318, Sep. 2024, doi: 10.33005/senada.v4i1.198.
- [12] B. Rahman, F. Fauzi, and S. Amri, "Perbandingan Hasil Klasifikasi Data Iris menggunakan Algoritma K-Nearest Neighbor dan Random Forest," *Journal of Data Insights*, vol. 1, no. 1, pp. 19–26, Jun. 2023, doi: 10.26714/jodi.v1i1.135.
- [13] M. M. Rofiq, F. A. Setiawan, and F. Riana, "Perbandingan Metode K-NN dan Random Forest pada Klasifikasi Mahasiswa Berpotensi Dropout," *INFOTECHjournal*, vol. 10, no. 1, pp. 84–89, Mar. 2024, doi: 10.31949/infotech.v10i1.8856.
- [14] I. K. Hasan, R. Resmawan, and J. Ibrahim, "Perbandingan K-Nearest Neighbor dan Random Forest dengan Seleksi Fitur Information Gain untuk Klasifikasi Lama Studi Mahasiswa," *Indonesian Journal of Applied Statistics*, vol. 5, no. 1, p. 58, May 2022, doi: 10.13057/ijas.v5i1.58056.
- [15] D. M. Musa *et al.*, "Penerapan Data Mining Untuk Klasifikasi Data Penjualan Pakan Ternak Terlaris Dengan Algoritma C4.5," *Jurnal Teknologi Informatika dan Komputer*, vol. 10, no. 1, pp. 172–186, Mar. 2024, doi: 10.37012/jtik.v10i1.1985.
- [16] S. P. Robbins and T. A. Judge, *Organizational Behavior 16ed*. Pearson Education, 2015.
- [17] Marselim and I. Fenriana, "Analisis Pola Pembelian Pelanggan Menggunakan Algoritma Apriori serta Metode Pengujian SUS," *bit-Tech(Binary Digital - Technology)*, vol. 7, no. 3, pp. 705–714, Apr. 2025, doi: 10.32877/bt.v7i3.1769.
- [18] J. Dwi Muthohhar and A. Prihanto, "Analisis Perbandingan Algoritma Klasifikasi untuk Penyakit Jantung," *Journal of Informatics and Computer Science*, vol. 04, no. 3, pp. 298–304, Jan. 2023, doi: 10.26740/jinacs.v4n03.p298-304.



- [19] S. D. K. Wardani, A. S. Ariyanto, M. Umroh, and D. Rolliawati, "Perbandingan Hasil Metode Clustering K-Means, DB Scanner & Hierarchical untuk Analisa Segmentasi Pasar," *JIKO (Jurnal Informatika dan Komputer)*, vol. 7, no. 2, p. 191, Sep. 2023, doi: 10.26798/jiko.v7i2.796.
- [20] L. U. Khasanah, Y. N. Nasution, and F. D. T. Amijaya, "Klasifikasi Penyakit Diabetes Melitus Menggunakan Algoritma Naïve Bayes Classifier," *BASIS: Jurnal Ilmiah Matematika*, vol. 1, no. 1, pp. 41–50, Sep. 2022, doi: 10.30872/basis.v1i1.918.
- [21] A. Ramadhan, A. Z. Firdausi, and U. Karyani, "Stres Pada Mahasiswa Selama Pandemi Covid-19," *Jurnal Psikologi Insight*, vol. 5, no. 2, pp. 130–136, Oct. 2021, doi: 10.17509/insight.v5i2.62774.
- [22] A. F. B. Sajiw, B. Rahmat, and A. Junaidi, "Klasifikasi Indeks Standar Pencemaran Udara (ISPU) Menggunakan Algoritma XGBoost dengan Teknik Imbalanced Data (SMOTE)," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Aug. 2024, doi: 10.23960/jitet.v12i3.4699.
- [23] L. Tanti, B. S. Riza, Y. Y. Thanri, and N. Panjaitan, "Model Klasifikasi Penyakit Kanker Payudara Menggunakan Algoritma Naive Bayes," *Informatic Technique Journal*, vol. 12, no. 2, pp. 1–12, Oct. 2024, doi: 10.22303/it.1.1.2021.01-10.
- [24] R. Chairunisa, Adiwijaya, and W. Astuti, "Perbandingan CART dan Random Forest untuk Deteksi Kanker berbasis Klasifikasi Data Microarray," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 5, pp. 805–812, Oct. 2020, doi: 10.29207/resti.v4i5.2083.
- [25] A. A. D. Halim and S. Anraeni, "Analisis Klasifikasi Dataset Citra Penyakit Pneumonia Menggunakan Metode K-Nearest Neighbor (KNN)," *Indonesian Journal of Data and Science (IJODAS)*, vol. 2, no. 1, pp. 1–12, Mar. 2021, doi: 10.33096/ijodas.v2i1.23.
- [26] I. Riadi, A. Fadlil, and P. Annisa, "Identifikasi Tulisan Tangan Huruf Katakan Jepang Dengan Metode Euclidean," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 4, no. 1, pp. 29–37, Mar. 2020, doi: 10.30645/j-sakti.v4i1.184.
- [27] A. Yudhana, Sunardi, and A. J. S. Hartanta, "Algoritma K-NN dengan Euclidean Distance untuk Prediksi Hasil Penggergajian Kayu Sengon," *TRANSMISI: Jurnal Ilmiah Teknik Elektro*, vol. 22, no. 4, pp. 123–129, Oct. 2020, doi: 10.14710/transmisi.22.4.107-141.
- [28] N. S. Rifda, Y. T. S. Sitorus, A. M. Naomi, S. Najwa, A. P. Anindya, and F. Indrayatna, "Penerapan Algoritma K-Nearest Neighbor pada Klasifikasi Penyakit Jantung," *BIASStatistics: Jurnal Statistika Teori dan Aplikasi*, vol. 2022, no. Special Issue 1, pp. 222–229, Jan. 2023, doi: 10.1234/bias.v2022i1.192.
- [29] Suci Amaliah, M. Nusrang, and A. Aswi, "Penerapan Metode Random Forest Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konijiwa Bantaeng," *VARIANSI: Journal of Statistics and Its application on Teaching and Research*, vol. 4, no. 3, pp. 121–127, Dec. 2022, doi: 10.35580/variansiunnm31.
- [30] E. Tangkelobo, W. Mayaut, H. Listanto, I. Binanto, and N. F. Sianipar, "Perbandingan Algoritma Klasifikasi Random Forest, Gaussian Naive Bayes, dan K-Nearest untuk Data Tidak Seimbang dan Data yang diseimbangkan dengan metode Random Undersampling pada dataset LCMS Tanaman Keladi Tikus," *Seminar Nasional Teknik Elektro, Informatika dan Sistem Informasi (SINTaKS)*, Aug. 2023, doi: 10.35842/sintaks.v2i1.28.
- [31] W. A. Firmansyah, U. Hayati, and Y. A. Wijaya, "Analisa Terjadinya Overfitting dan Underfitting pada Algoritma Naive Bayes dan Decision Tree dengan Teknik Cross Validation," *Jurnal Mahasiswa Teknik Informatika*, vol. 7, no. 1, pp. 262–269, Feb. 2023, doi: 10.36040/jati.v7i1.6329.
- [32] H. Azis, P. Purnawansyah, F. Fattah, and I. P. Putri, "Performa Klasifikasi K-NN dan Cross Validation Pada Data Pasien Pengidap Penyakit Jantung," *ILKOM Jurnal Ilmiah*, vol. 12, no. 2, pp. 81–86, Aug. 2020, doi: 10.33096/ilkom.v12i2.507.81-86.
- [33] W. Wijiyanto, A. I. Pradana, S. Sopangi, and V. Atina, "Teknik K-Fold Cross Validation untuk Mengevaluasi Kinerja Mahasiswa," *Jurnal Algoritma*, vol. 21, no. 1, May 2024, doi: 10.33364/algoritma.v.21-1.1618.
- [34] H. Utami, "Analisis Sentimen dari Aplikasi Shopee Indonesia Menggunakan Metode Recurrent Neural Network," *Indonesian Journal of Applied Statistics*, vol. 5, no. 1, p. 31, May 2022, doi: 10.13057/ijas.v5i1.56825.