

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Dalam penelitian ini, pengumpulan data ditempuh dengan menyebar kuesioner berbasis Google Form kepada para responden yang merupakan Generasi Z (individu lahir pada pertengahan 1997 hingga 2012). Kuesioner ini dirancang untuk menilai dua aspek penting, yaitu intensitas gaya hidup digital responden dan tingkat kecemasan sosial yang mereka alami.

Alat pengukur untuk kecemasan sosial yang digunakan mengacu pada skala *Liebowitz Social Anxiety Scale (LSAS)*, yang telah dimodifikasi agar sesuai dengan kebutuhan penelitian ini. Kuesioner terdiri dari 20 pertanyaan, di mana setiap pertanyaan dijawab melalui skala ordinal dengan empat pilihan jawaban, yaitu: “Tidak Pernah”, “Jarang”, “Sering”, dan “Sangat Sering”.

Proses pengumpulan data melalui kuesioner online ini berlangsung selama periode yang telah ditentukan hingga jumlah responden memenuhi target penelitian. Berdasarkan hasil penyebaran kuesioner, jumlah data yang berhasil dikumpulkan adalah sebanyak 302 responden. Data mentah tersebut selanjutnya digunakan sebagai dasar pada tahap pemrosesan data (preprocessing) untuk memastikan kualitas data sebelum diolah lebih lanjut.

Instrumen lengkap kuesioner penelitian, yang disusun berdasarkan hasil studi literature, dan kuesioner dapat dilihat pada Lampiran 6. Contoh data mentah hasil kuesioner untuk beberapa responden pertama dan terakhir sebelum melalui tahap preprocessing disajikan pada tabel berikut:

Tabel 4. 1 Data Hasil Kuesioner

NO.	Timesta mp	Nama	Jenis Kela min	Umu r	Pertan yaan	Pertan yaan	Pertan yaan	Pert anya an
1.	14/04/20 26 8:41:54	Dewi Intan	P	19-23	Sering	Sering	Jarang	...
2.	14/04/20 26 8:44:02	Indri An	P	19-23	Sering	Jarang	Sering	...
3.	14/04/20 26 8:55:48	Ulfa Lailatul Sofia	P	19-23	Jarang	Sangat Sering	Sangat Sering	...
4.	14/04/20 26 8:56:04	Salsadel a Mayza Sabira	P	19-23	Sangat Sering	Sering	Jarang	...
...	...	...	...	...	...	...	...	...
299	07/06/20 26 17:25:45	Grace Marchilv ia	P	24-28	Sering	Sering	Sangat Sering	...
300	07/06/20 26 18:42:14	Gadis Melsiana	P	19-23	Jarang	Sering	Sangat Sering	...
301	07/06/20 26 21:44:03	Lala	P	19-23	Sering	Sangat Sering	Sangat Sering	...
302	07/06/20 26 22:11:21	Neya	P	19-23	Jarang	Sering	Sangat Sering	...

4.2 Hasil Pemrosesan Data (Preprocessing)

Data mentah dari kuesioner umumnya masih mengandung ketidakkonsistenan seperti spasi berlebihan, data duplikat dan data kosong (missing values), sehingga perlu dilakukan tahapan preprocessing agar data siap diolah. Proses ini meliputi data cleaning, transformasi, pelabelan skor, dan normalisasi data. Berikut adalah hasil pemrosesan data setiap tahapannya.

4.2.1 Data Cleaning

Tahap ini bertujuan untuk membersihkan dataset dari spasi yang berlebihan pada teks, data duplikat, dan data kosong (missing values). Proses ini dilakukan dengan menggunakan bahasa Python dan memanfaatkan pustaka Pandas.

Hasil proses cleaning pada tahap ini ditampilkan pada tabel dibawah ini.

Tabel 4. 2 Hasil *Data Cleaning*

NO.	Keterangan	Jumlah Data
1.	Total Data Awal	302 baris
2.	Data Duplikat/Kosong	7 baris
3.	Data Bersih	295 baris

Setelah menjalankan proses cleaning, data mentah telah dibersihkan dari spasi yang berlebihan, entri duplikat, serta nilai yang kosong. Data berkurang dari 302 menjadi 295 baris, dengan 7 baris dihilangkan karena dianggap tidak valid atau duplikat.

4.2.2 Data Transformation

Sesudah data bersih diperoleh, selanjutnya ke proses tranformasi data. Bagian ini ditujukan untuk mengonversi data mentah yang masih berupa teks jawaban kuesioner bertentuk numerik. Transformasi dilakukan melalui dua langkah. Pertama, nama asli responden disamarkan dan diganti dengan kode anonim (R1, R2, R3, dan seterusnya) guna menjaga privasi data. Kedua, jawaban berbentuk teks pada setiap butir pertanyaan dikonversi ke dalam nilai numerik

berdasarkan peta ordinal yang ditetapkan, yaitu: “Tidak Pernah” = 0, “Jarang” = 1, Sering = 2, dan “Sangat Sering” = 3. Nama kolom pertanyaan juga disederhanakan menjadi P1 hingga P20 untuk kemudahan pengolahan data.

Contoh hasil transformasi data pada lima responden pertama ditampilkan pada tabel berikut:

Tabel 4. 3 Hasil *Data Transformation* (Lima Responden Pertama)

No	Nama	P1	P2	P3	P4	P5	P6	P7	P8	...
1.	R1	2	2	1	3	2	1	1	0	...
2.	R2	2	1	2	1	2	1	0	1	...
3.	R3	1	3	3	3	3	2	2	1	...
4.	R4	3	2	2	3	3	1	1	0	...
5.	R5	2	2	2	2	1	0	1	0	...

Berdasarkan hasil transformasi data pada tabel diatas, seluruh jawaban berskala ordinal untuk 20 butir pertanyaan (P1 sampai P20) telah berhasil dikonversi ke dalam nilai numerik (0–3). Proses ini memastikan bahwa tidak terdapat nilai yang gagal dikonversi, sehingga data telah siap untuk dianalisis lebih lanjut.

4.2.3 Data Labeling

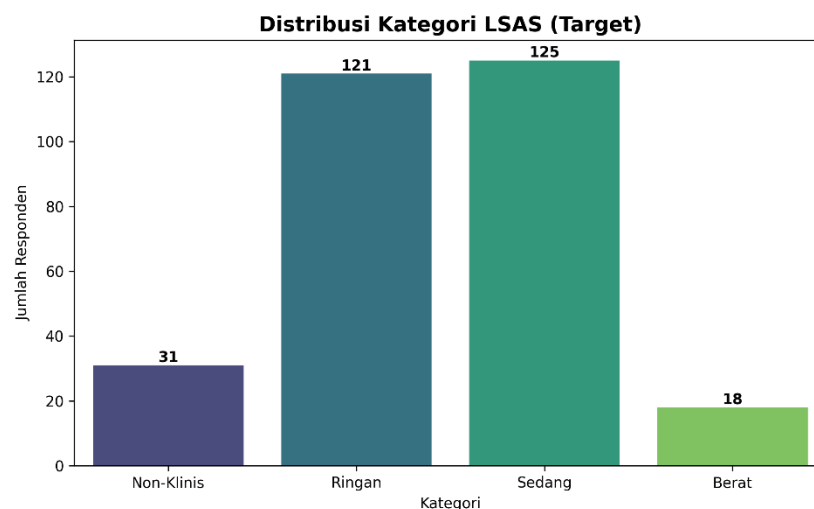
Setelah data bertransformasi menjadi nilai numerik, langkah selanjutnya adalah pemberian label kelas (*data labeling*). Total skor dari 20 item pertanyaan dijumlahkan untuk setiap responden, kemudian dikategorikan ke dalam empat kelas tingkat kecemasan sosia berdasarkan ambang batas skor yang mengacu pada adaptasi skala LSAS.

Tabel 4. 4 Ambang Batas Kategori Skor LSAS

Rentang Skor Total	Kategori Kecemasan Sosial
0 - 20	Non-Klinis
21 - 38	Ringan
39 - 50	Sedang
51 - 60	Berat

Berdasarkan tabel diatas, fungsi kategorisasi membagi total skor LSAS (rentang 0-60) ke dalam empat kategori tingkat kecemasan sosial. Fungsi ini diterapkan pada seluruh baris data, sehingga responden memperoleh label kategori kecemasan sesuai dengan total skor yang diperolehnya. Hasil pelabelan kemudian disimpan sebagai kolom baru bernama target yang akan menjadi variabel keluaran (output) pada proses klasifikasi.

Distribusi jumlah responden untuk masing-masing kategori target dapat diamati pada gambar berikut:



Gambar 4. 1 Distribusi Kategori LSAS (Target)

Gambar di atas menunjukkan bahwa dari total 295 data yang telah melalui tahap cleaning, distribusi kategori kecemasan sosial tidak merata antar kelas. Kategori "Sedang" memiliki jumlah responden terbanyak yaitu 125 orang, diikuti kategori "Ringan" sebanyak 121 orang, kategori "Non-Klinis" sebanyak 31 orang, dan kategori "Berat" merupakan kelas dengan jumlah responden paling sedikit yaitu

18 orang. Kondisi ini menunjukkan adanya ketidakseimbangan kelas (class imbalance), khususnya pada kategori "Berat" dan "Non-Klinis" yang jumlah sampelnya jauh lebih sedikit dibandingkan kategori "Ringan" dan "Sedang". Oleh karena itu, pada tahap pembagian data (data splitting), teknik stratified sampling diterapkan agar proporsi masing-masing kelas tetap terjaga baik pada data latih maupun data uji.

4.2.4 Data Normalization

Sebelum data dimasukkan ke dalam model *machine learning*, data terlebih dahulu melalui proses normalisasi dengan menerapkan metode *Min-Max Scaling*. Normalisasi bertujuan untuk menyeragamkan skala seluruh fitur karena bergantung pada perhitungan jarak (*distance-based*). Apabila normalisasi tidak dilakukan, fitur dengan rentang nilai yang lebih besar berpotensi mendominasi proses perhitungan jarak sehingga dapat menurunkan tingkat akurasi model.

Formula Min-Max Scaling yang digunakan adalah sebagai berikut:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (7)$$

Contoh hasil normalisasi untuk lima responden pertama, dibandingkan dengan nilai sebelum dinormalisasi, disajikan pada tabel dibawah ini:

Tabel 4. 5 Hasil *Data Normalization*

Nama	P1 (Sebelum)	P1 (Sesudah)	...	Total Skor (Sebelum)	Total Skor (Sesudah)
R1	2	0,667	...	22	0,315
R2	2	0,667	...	22	0,315
R3	1	0,333	...	37	0,593
R4	3	1,000	...	23	0,333
R5	2	0,667	...	17	0,222

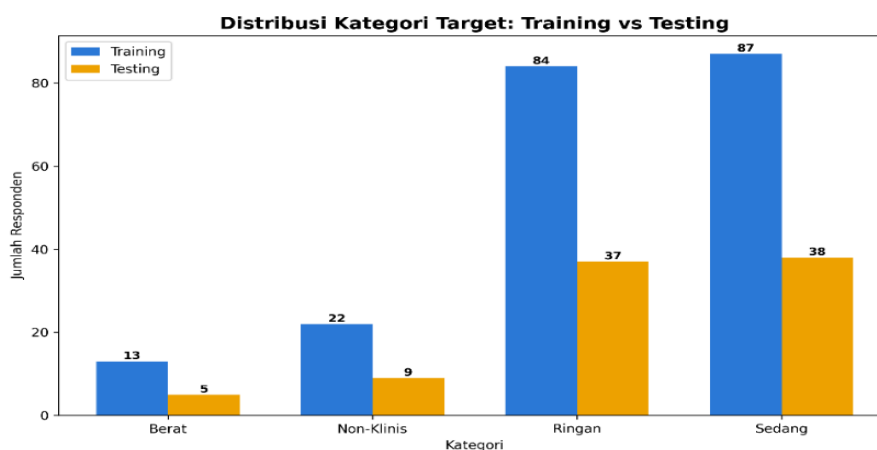
Berdasarkan tabel di atas, proses normalisasi menggunakan Min-MaxScaler dari pustaka Scikit-learn berhasil mengubah seluruh nilai fitur P1 sampai P20 dan Total_Score ke dalam rentang skala 0 sampai 1, sementara kolom

Nama dan target tetap dipertahankan dalam bentuk aslinya karena bukan merupakan fitur numerik yang dihitung jaraknya. Statistik deskriptif pada kolom Total_Score setelah normalisasi menunjukkan nilai minimum 0 dan maksimum 1, yang mengonfirmasi bahwa proses penyeragaman skala telah berhasil dilakukan sebelum data digunakan pada tahap pelatihan model, khususnya untuk algoritma yang sensitif terhadap perbedaan skala fitur seperti KNN.

4.3 Data Splitting

Sebelum pelatihan model dimulai, dataset yang telah melalui seluruh tahap pra-pemrosesan dibagi kedalam dua subset: data latih (training data) dan data uji (testing data) dengan proporsi 70:30 menggunakan teknik stratified sampling, sehingga distribusi proporsi kelas pada masing-masing subset tetap merepresentasikan proporsi yang terdapat pada dataset asli. Hasil pembagian data ditunjukkan pada Gambar 4.2 dan tabel 4.

Gambar 4. 2 Distribusi Kategori Target: Training vs Testing



Tabel 4. 6 Hasil *Data Splitting*

Kategori	Data Latih (70%)	Data Uji (30%)	Total
Berat	13	5	18
Non-Klinis	22	9	31
Ringan	84	37	121
Sedang	87	38	125
Total	206	89	295

Dari total 295 data, sebanyak 206 data (70%) dialokasikan sebagai data latih dan 89 data (30%) sebagai data uji. Proporsi tiap kategori pada kedua subset relative konsisten dengan proporsi pada dataset asli, sehingga model tidak bias terhadap kelas mayoritas saat pelatihan maupun pengujian.

4.4 Implementasi Model Klasifikasi

Pada tahap ini, tiga algoritma klasifikasi machine learning diterapkan secara terpisah menggunakan data training yang telah disiapkan sebelumnya. Ketiga algoritma yang digunakan meliputi Gaussian Naïve Bayes, K-Nearest Neighbor (KNN), dan Random Forest. Performa masing-masing model diukur melalui empat metrik evaluasi, yaitu akurasi, presisi, recall, dan F1-score dengan pembobotan (weighted), mengingat adanya ketidakseimbangan kelas pada dataset.

4.4.1 Implementasi Algoritma *Naïve Bayes*

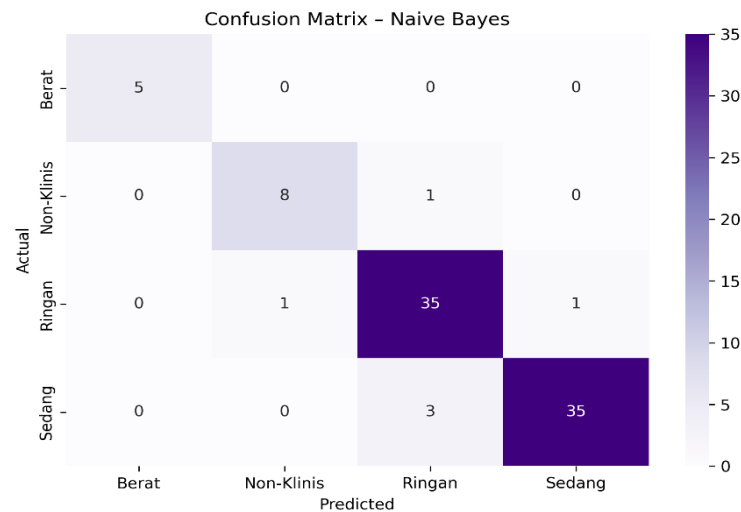
Algoritma pertama yang dibangun pada penelitian ini menggunakan algoritma Gaussian Naïve Bayes (GNB). Varian Gaussian dari Naïve Bayes dipilih karena data fitur dalam penelitian ini bersifat kontinu (nilai numerik hasil normalisasi), berbeda dengan Multinomial Naïve Bayes yang lazim digunakan pada data frekuensi teks. Gaussian Naïve Bayes berasumsi bahwa nilai setiap fitur pada suatu kelas mengikuti distribusi Gaussian (distribusi normal).

Proses pelatihan model dilakukan menggunakan fungsi `fit()` dengan memasukkan data latih (`X_train` dan `y_train`). Setelah proses pelatihan selesai, model kemudian digunakan untuk melakukan prediksi pada data uji (`X_test`), yang hasilnya selanjutnya dibandingkan dengan label sesungguhnya (`y_test`) guna mengukur tingkat performanya. Hasil evaluasi model untuk setiap kategori dapat dilihat pada tabel berikut.

Tabel 4. 7 Hasil Evaluasi Model Naïve Bayes

Kategori	Precision	Recall	F1-Score	Support
Berat	1.00	1.00	1.00	5
Non-Klinis	0.89	0.89	0.89	9
Ringan	0.90	0.95	0.92	37
Sedang	0.97	0.92	0.95	38
Accuracy			0.93	89
Macro avg	0.94	0.94	0.94	89
Weighted avg	0.93	0.93	0.93	89

Berdasarkan Tabel 4.7, model Naive Bayes memperoleh akurasi keseluruhan sebesar 93,2%, dengan presisi 93,4%, recall 93,2%, dan F1-score 93,2%. Nilai presisi dan recall tertinggi dicapai pada kategori "Berat" yang mencapai 100%, sedangkan performa sedikit lebih rendah terjadi pada kategori "Non-Klinis" dan "Ringan". Rincian distribusi kesalahan klasifikasi antar kategori dapat dilihat pada confusion matrix berikut.



Gambar 4. 3 Confusion Matrix Model Naïve Bayes

Gambar confusion matrix di atas menunjukkan bahwa model Naïve Bayes berhasil mengklasifikasikan seluruh 5 data kategori “Berat” dengan benar, serta 8 dari 9 data kategori “Non-Klinis”. Pada kategori “Ringan”, 35 dari 37 data diklasifikasikan dengan tepat, sedangkan 2 data lainnya keliru diklasifikasikan ke kategori lain, masing-masing 1 data sebagai “Non-Klinis” dan 1 data sebagai “Sedang”. Pada kategori “Sedang”, 35 dari 38 data diklasifikasikan dengan benar, sementara 3 data lainnya keliru diklasifikasikan sebagai “Ringan.” Secara keseluruhan, model Naïve Bayes memperoleh nilai akurasi sebesar 93,2%, dengan nilai presisi 93,4%, recall 93,2%, dan F1-score 93,2%, yang menunjukkan performa yang cukup baik meskipun terdapat sedikit kesalahan klasifikasi pada kelas yang berdekatan skornya (Ringan dan Sedang).

4.4.2 Implementasi Algoritma *K-Nearest Neighbor*

Model kedua yang diimplementasikan adalah algoritma K-Nearest Neighbor (KNN). KNN merupakan algoritma berbasis instance (lazy learner) yang melakukan klasifikasi berdasarkan kemiripan dengan data latih terdekat. Ketika data uji baru dimasukkan, algoritma menghitung jarak antara data uji tersebut

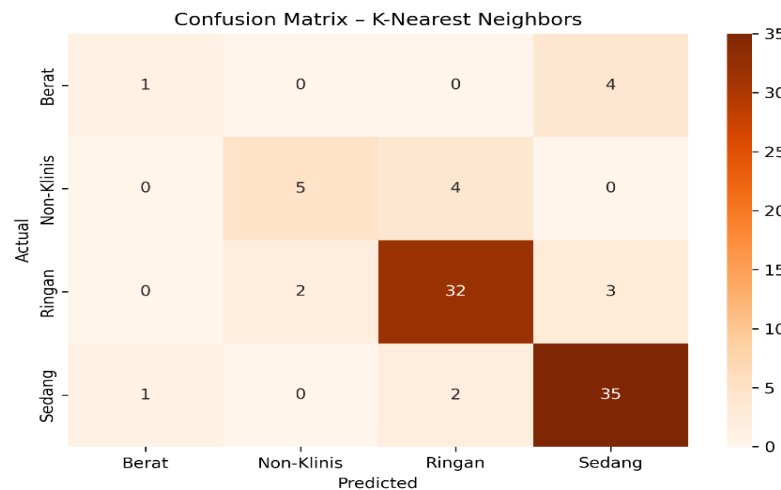
dengan seluruh titik data latih, kemudian menentukan kelas berdasarkan suara terbanyak (majority voting) dari k tetangga terdekat.

Dalam penelitian ini, digunakan nilai $k = 5$, yang merupakan nilai default umum (default) dan sering digunakan sebagai titik awal eksperimen. Penerapan normalisasi data pada tahap sebelumnya menjadi sangat krusial untuk algoritma ini, mengingat KNN sangat dipengaruhi oleh perbedaan skala antar fitur. Hasil evaluasi model perkategori ditunjukkan pada tabel dibawah ini.

Tabel 4. 8 Hasil Evaluasi Model KNN

Kategori	Precision (%)	Recall (%)	F1-Score (%)	Junlah Data Uji
Berat	0.50	0.20	0.29	5
Non-Klinis	0.71	0.56	0.62	9
Ringan	0.84	0.86	0.85	37
Sedang	0.83	0.92	0.88	38
Accuracy			0.82	89
Macro avg	0.72	0.64	0.66	89
Weighted avg	0.81	0.82	0.81	89

Berdasarkan Tabel 4.8, model KNN memperoleh akurasi keseluruhan sebesar 82,0%, dengan presisi 80,6%, recall 82,0%, dan F1-score 80,7%, yang merupakan nilai terendah di antara ketiga algoritma yang diuji. Performa paling lemah terlihat pada kategori "Berat" dengan recall hanya 20,0%, sedangkan kategori "Ringan" dan "Sedang" menunjukkan hasil yang relatif lebih baik. Rincian distribusi kesalahan klasifikasi antar kategori dapat dilihat pada confusion matrix berikut.



Gambar 4. 4 Confusion Matrix Model KNN

Gambar confusion matrix di atas menunjukkan bahwa model KNN hanya mampu mengklasifikasikan 1 dari 5 data kategori "Berat" dengan benar, serta 5 dari 9 data kategori "Non-Klinis". Pada kategori "Ringan", 32 dari 37 data diklasifikasikan dengan tepat, sementara pada kategori "Sedang", 35 dari 38 data diklasifikasikan dengan benar. Secara keseluruhan, model KNN memperoleh nilai akurasi sebesar 82,0%, dengan nilai presisi 80,6%, recall 82,0%, dan F1-score 80,7%, yang menunjukkan performa terendah di antara ketiga algoritma yang diuji, dengan kesalahan klasifikasi yang cukup banyak terjadi pada kategori "Berat" dan "Non-Klinis".

4.4.3 Implementasi Algoritma *Random Forest*

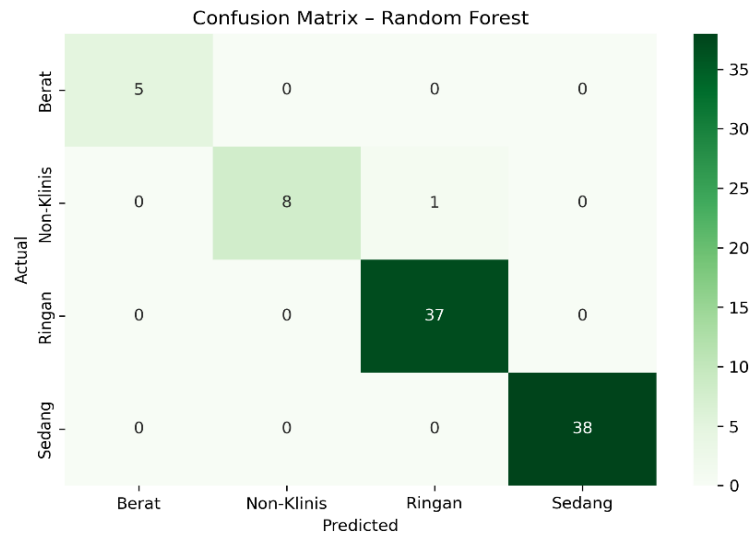
Algoritma ketiga yang diterapkan dalam penelitian ini adalah Random Forest (RF), yakni sebuah metode ensemble yang memanfaatkan kumpulan pohon keputusan (*decision trees*) yang masing-masing dilatih secara independen. Penentuan hasil akhir prediksi dilakukan melalui mekanisme voting mayoritas dari keseluruhan pohon yang telah terbentuk. Dengan pendekatan semacam ini, risiko overfitting yang kerap muncul pada decision tree tunggal dapat ditekan, sehingga model yang dihasilkan menjadi lebih stabil dan tahan terhadap variasi data.

Dalam penelitian ini, model Random Forest diinisialisasi dengan parameter `random_state=42` `n_estimators=100` dan `random_state=42`. Hasil evaluasi model per kategori ditunjukkan pada Tabel 4.9.

Tabel 4. 9 Hasil Evaluasi Model Random Forest

Kategori	Precision (%)	Recall (%)	F1-Score (%)	Junlah Data Uji
Berat	1.00	1.00	1.00	5
Non-Klinis	1.00	0.89	0.94	9
Ringan	0.97	1.00	0.99	37
Sedang	1.00	1.00	1.00	38
Accuracy			0.99	89
Macro avg	0.99	0.97	0.98	89
Weighted avg	0.99	0.99	0.99	89

Berdasarkan Tabel 4.12, model Random Forest mencatatkan akurasi keseluruhan sebesar 98,8%, dengan presisi 98,9%, recall 98,8%, dan F1-score 98,8%, menjadikan Random Forest sebagai model dengan performa terbaik di antara ketiga algoritma yang diuji. Seluruh kategori mencapai nilai precision dan recall di atas 97%, dengan kategori "Berat" dan "Sedang" mencapai nilai sempurna 100%. Rincian distribusi kesalahan klasifikasi antar kategori dapat dilihat pada confusion matrix berikut.



Gambar 4. 5 Confusion Matrix Model Random Forest

Gambar confusion matrix di atas menunjukkan performa klasifikasi yang sangat baik, dimana Random Forest berhasil memprediksi dengan benar seluruh data pada kategori “Berat” (5 dari 5), “Ringan” (37 dari 37), dan “Sedang” (38 dari 38). Hanya terdapat 1 kesalahan klasifikasi pada kategori “Non-klinis”, yaitu 1 dari 9 data yang keliru diprediksi sebagai “Ringan”. Hasil ini menghasilkan nilai akurasi sebesar 98,8%, dengan presisi 98,9%, recall 98,8%, dan F1-score 98,8%, menjadikan Random Forest sebagai model dengan performa terbaik di antara ketiga algoritma yang diuji dalam penelitian ini.

4.5 Evaluasi dan Perbandingan Model

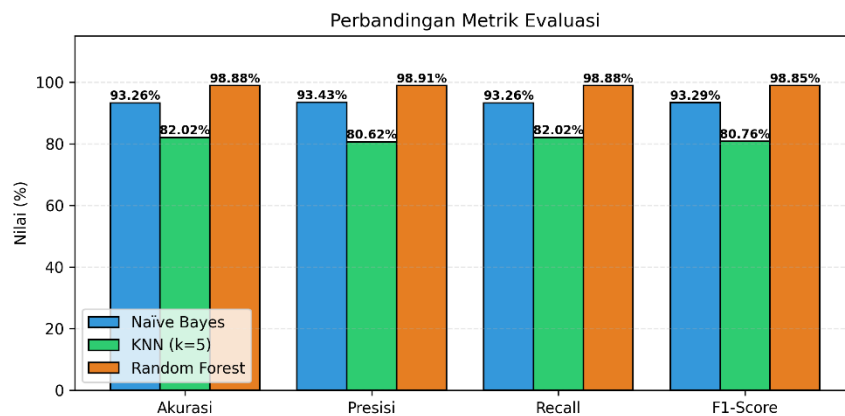
Setelah ketiga model klasifikasi selesai dilatih dan dievaluasi secara terpisah, tahap selanjutnya adalah membandingkan algoritma performa ketiga model tersebut secara menyeluruh guna mengidentifikasi model dengan kemampuan prediksi terbaik dalam memprediksi tingkat kecemasan sosial pada Generasi Z. Perbandingan performa dari ketiga model tersebut disajikan pada tabel berikut:

Tabel 4. 10 Perbandingan Performa Ketiga Model Klasifikasi

Model	Akurasi	Presisi	Recall	F1-Score
Naïve Bayes	93,2%	93,4%	93,2%	93,2%
KNN (k=5)	82,0%	80,6%	82,0%	80,7%
Random Forest	98,8%	98,9%	98,8%	98,8%

Tabel diatas menyusun seluruh nilai metric evaluasi (akurasi, presisi, recall, dan F1-score) dari ketiga model ke dalam satu tabel perbandingan (df_performance). Selanjutnya, dilakukan pencarian otomatis terhadap model dengan nilai akurasi tertinggi (model terbaik) dan terendah (model terlemah) menggunakan fungsi `idxmax()` dan `idxmin()`, serta perhitungan selisih akurasi antara kedua model tersebut.

Perbandingan visual keempat evaluasi dari ketiga algoritma dapat dilihat pada gambar berikut:

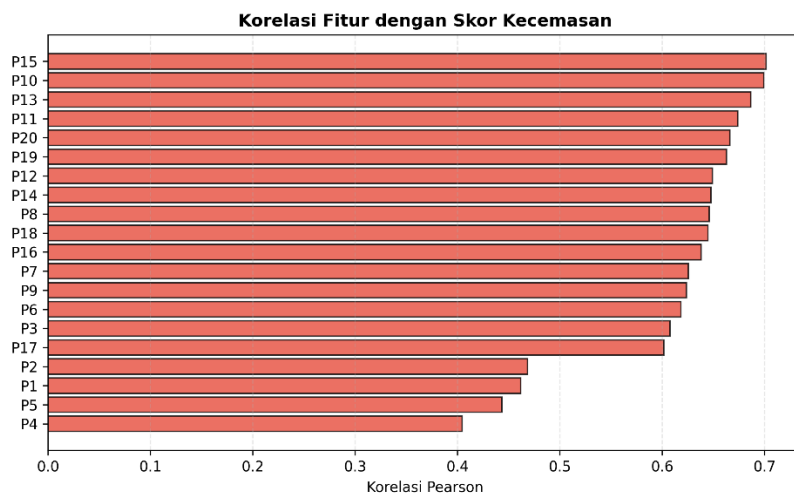


Gambar 4. 6 Grafik Perbandingan Akurasi, Presisi, Recall, dan F1-Score

Mengacu gambar di atas, Random Forest secara konsisten meraih nilai paling tinggi pada seluruh indikator evaluasi, yaitu akurasi 98,8%, presisi 98,9%, recall 98,8%, dan F1-score 98,8%. Model Naïve Bayes berada di posisi kedua dengan nilai akurasi 93,2%, presisi 93,4%, recall 93,2% dan F1-score 93,2%. Sementara itu, model KNN (k=5) menunjukkan performa terendah di antara ketiga algoritma, dengan nilai akurasi 82,0%, presisi 80,6%, recall 82,0% dan F1-score 80,7%. Dengan demikian dapat disimpulkan bahwa Random Forest menjadi model

terbaik berdasarkan nilai akurasi, sedangkan KNN merupakan model dengan performa terlemah, dengan selisih akurasi antara kedua model tersebut sebesar 16,8%. Keunggulan Random Forest ini sejalan dengan karakteristiknya sebagai metode ensemble yang menggabungkan banyak pohon keputusan sehingga lebih tahan terhadap ketidakseimbangan kelas dan variasi data dibandingkan model tunggal seperti Naïve Bayes maupun pendekatan berbasis jarak seperti KNN.

Selain evaluasi performa model klasifikasi, dilakukan pula analisis korelasi Pearson antara setiap fitur gaya hidup digital (P1 hingga P20) dengan skor kecemasan (Total_Score) untuk mengidentifikasi fitur yang paling berpengaruh terhadap tingkat kecemasan sosial. Hasil analisis korelasi ditampilkan pada gambar berikut:



Gambar 4. 7 Hasil Analisis Korelasi Pearson

Berdasarkan gambar di atas, fitur P15 dan P10 tercatat sebagai dua fitur dengan korelasi tertinggi terhadap skor kecemasan, masing-masing dengan nilai korelasi Pearson sekitar 0,70. Fitur-fitur lain seperti P13, P11, P20, P19, P12, P14, P8, P18, P16, P7, P9, P6, P3, dan P17 juga menunjukkan korelasi yang cukup kuat, berada pada rentang 0,60 hingga 0,69. Sementara itu, fitur P2, P1, P5, dan P4 memiliki korelasi yang relatif lebih rendah, yaitu pada rentang 0,40 hingga 0,46. Hasil ini mengindikasikan bahwa sebagian besar fitur gaya hidup digital yang diukur memiliki hubungan linear yang cukup kuat dengan tingkat kecemasan sosial, dengan P15 dan P10 menjadi indikator yang paling representatif dalam memprediksi skor kecemasan pada Gen Z.