

# Optimasi Support Vector Machine Menggunakan RandomizedSearchCV dan SMOTE untuk Klasifikasi Kebugaran Berdasarkan Parameter Fisiologis

Gema Amran Nathansyach, Purwanto\*

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia

Email: <sup>1</sup>111202214800@mhs.dinus.ac.id, <sup>2,\*</sup>purwanto@dsn.dinus.ac.id

Email Penulis Korespondensi: purwanto@dsn.dinus.ac.id

Submitted: 22/11/2025; Accepted: 26/12/2025; Published: 26/12/2025

**Abstrak**—Penelitian ini bertujuan untuk meningkatkan akurasi model *Support Vector Machine* (SVM) dalam mengklasifikasikan status kebugaran (fit/tidak fit) berdasarkan parameter *fisiologis* dan gaya hidup menggunakan *Fitness Classification Dataset*, yaitu dataset sintesis yang dirancang untuk merepresentasikan indikator kebugaran seperti BMI, tinggi badan, berat badan, detak jantung, tekanan darah, kualitas nutrisi, durasi tidur, dan indeks aktivitas. Dataset memiliki distribusi kelas yang tidak seimbang serta memuat kombinasi fitur numerik dan kategorikal sehingga memerlukan pra-pemrosesan komprehensif. Penelitian ini menerapkan dua teknik optimasi, yaitu *RandomizedSearchCV* untuk penyetelan *hiperparameter* secara efisien dan SMOTE untuk menangani ketidakseimbangan kelas. Hasil eksperimen menunjukkan bahwa model SVM baseline menghasilkan akurasi 75,75%, sedangkan kombinasi SVM + *RandomizedSearchCV* + SMOTE meningkatkan akurasi menjadi 80%, atau terjadi peningkatan sebesar 4,25%. Selain itu, nilai AUC turut meningkat dari 0.835 pada *baseline* menjadi 0.850 pada model teroptimasi. Temuan ini menunjukkan bahwa integrasi *RandomizedSearchCV* dan SMOTE secara signifikan meningkatkan kemampuan model dalam menangkap pola non-linear sekaligus meningkatkan sensitivitas terhadap kelas minoritas. Secara keseluruhan, penelitian ini membuktikan bahwa *pipeline* SVM yang dioptimalkan mampu memberikan performa yang lebih stabil dan akurat dalam tugas klasifikasi status kebugaran, serta dapat menjadi acuan untuk pengembangan model prediktif di domain kesehatan lainnya.

**Kata Kunci:** Support Vector Machine; RandomizedSearchCV; SMOTE; Machine Learning; Klasifikasi Kebugaran

**Abstract**—This study aims to improve the accuracy of the Support Vector Machine (SVM) model in classifying fitness status (fit/unfit) based on physiological parameters and lifestyle using the Fitness Classification Dataset, which is a synthetic dataset designed to represent fitness indicators such as BMI, height, weight, heart rate, blood pressure, nutritional quality, sleep duration, and activity index. The dataset has an imbalanced class distribution and contains a combination of numerical and categorical features, thus requiring comprehensive preprocessing. This study applies two optimization techniques, namely RandomizedSearchCV for efficient hyperparameter tuning and SMOTE for handling class imbalance. The experimental results show that the baseline SVM model produces an accuracy of 75.75%, while the combination of SVM + RandomizedSearchCV + SMOTE increases the accuracy to 80%, or an increase of 4.25%. In addition, the AUC value also increased from 0.835 in the baseline to 0.850 in the optimized model. These findings indicate that the integration of RandomizedSearchCV and SMOTE significantly improves the model's ability to capture non-linear patterns while increasing sensitivity to minority classes. Overall, this study proves that the optimized SVM pipeline is capable of providing more stable and accurate performance in fitness status classification tasks and can be used as a reference for developing predictive models in other health domains.

**Keywords:** Support Vector Machine; RandomizedSearchCV; SMOTE; Machine Learning; Fitness Classification

## 1. PENDAHULUAN

Perkembangan teknologi digital telah mengubah cara manusia merekam, memahami, dan memanfaatkan data kesehatan sehari-hari. Perangkat seperti smartwatch, fitness tracker, serta aplikasi kesehatan kini mampu memantau langkah harian, detak jantung, durasi tidur, hingga kualitas nutrisi secara kontinu dan real-time [1], [2]. Data yang dihasilkan perangkat tersebut dapat diolah menjadi indikator penting untuk menilai pola hidup dan status kebugaran, sehingga membuka peluang pengembangan model prediktif berbasis machine learning yang lebih adaptif dan presisi [3], [4]. Pendekatan ini terbukti mampu membantu pengguna memahami kondisi fisik secara lebih objektif sekaligus memberikan potensi deteksi dini terhadap penurunan kebugaran atau risiko kesehatan tertentu [5], [6].

Keberagaman fitur fisiologis dan gaya hidup dalam data kebugaran membuat metode machine learning menjadi semakin relevan karena mampu menangkap pola non-linear yang tidak mudah dijelaskan oleh teknik tradisional [7], [8]. Support Vector Machine (SVM) menjadi salah satu algoritma yang banyak digunakan dalam klasifikasi kesehatan karena stabil terhadap dimensi fitur tinggi dan efektif dalam menemukan batas pemisah antar kelas [9], [10]. Namun, kinerjanya sangat dipengaruhi oleh kualitas pra-pemrosesan, pemilihan kernel, serta pengaturan hiperparameter yang tepat agar model dapat bekerja optimal pada data nyata [11], [12].

Penelitian sebelumnya menunjukkan bahwa variabel sehari-hari seperti BMI, tekanan darah, detak jantung, kualitas tidur, dan aktivitas fisik dapat menjadi masukan yang valid untuk memprediksi tingkat kebugaran menggunakan model machine learning [13], [14]. Selain itu, integrasi data gaya hidup seperti nutrisi, durasi tidur, serta beban aktivitas harian terbukti mendukung proses klasifikasi kebugaran dan memetakan kesiapan fisik individu [15], [16]. Pendekatan yang memanfaatkan pipeline machine learning juga semakin umum digunakan untuk menggabungkan berbagai jenis fitur dalam pemodelan kesehatan [17].

Namun, data kebugaran sering kali mengalami masalah ketidakseimbangan kelas, misalnya jumlah individu tidak fit yang lebih sedikit dibanding individu fit. Kondisi ini dapat menurunkan akurasi model karena algoritma

cenderung condong pada kelas mayoritas. Metode SMOTE dan variannya telah terbukti mampu meningkatkan representasi kelas minoritas dan menghasilkan performa yang lebih stabil pada data kesehatan imbalanced [18], [19]. Sejalan dengan itu, pemilihan hiperparameter secara efisien melalui RandomizedSearchCV menjadi semakin penting karena dapat meningkatkan akurasi tanpa membutuhkan komputasi berlebih [11], [12].

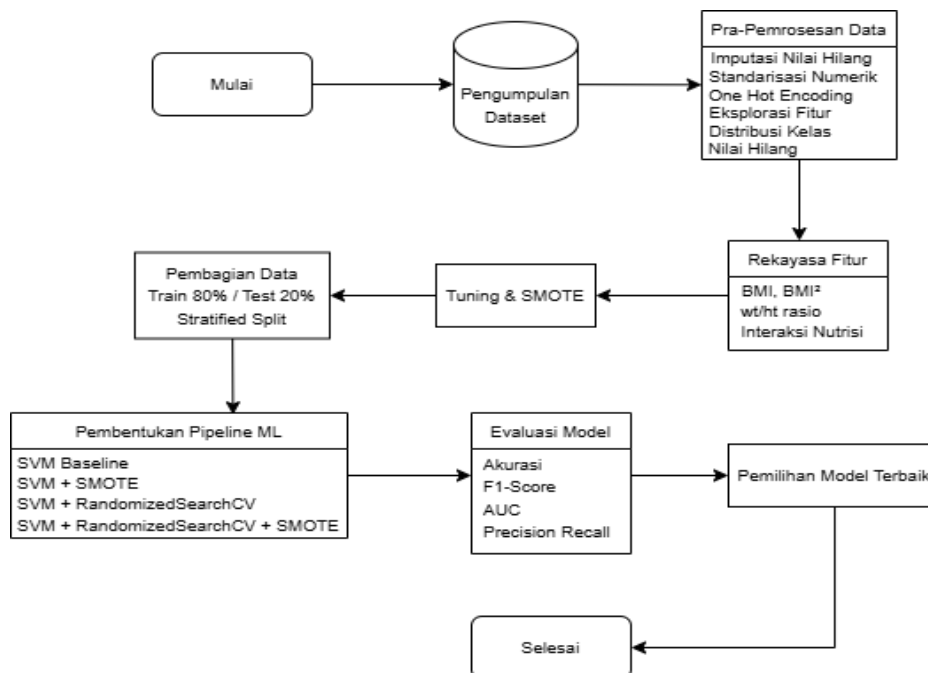
Selain penyeimbangan dan optimasi, meningkatnya jumlah fitur akibat encoding dapat memperlambat perhitungan kernel SVM. Oleh karena itu, pendekatan kernel approximation seperti metode Nystroem mulai banyak diterapkan dalam domain kesehatan untuk meningkatkan kecepatan komputasi tanpa mengorbankan kualitas prediksi [20]. Penggabungan metode ini dengan teknik interpretabilitas seperti permutation importance memungkinkan peneliti memahami kontribusi fitur secara lebih transparan dan membantu proses analisis hasil model.

Berdasarkan kesenjangan tersebut, penelitian ini berfokus pada pengembangan *pipeline* SVM yang komprehensif untuk mengklasifikasikan status kebugaran (*is\_fit*) menggunakan *Fitness Dataset*. *Pipeline* ini mencakup pra-pemrosesan otomatis berbasis *ColumnTransformer*, rekayasa fitur (BMI, rasio berat-tinggi, interaksi nutrisi), penanganan ketidakseimbangan kelas menggunakan SMOTE, kombinasi SVM+SMOTE, kombinasi SVM+*RandomizedSearchCV*+SMOTE menggunakan *kernel approximation*. Dalam penelitian ini, istilah *kernel approximation* mengacu pada langkah aproksimasi kernel yang diimplementasikan menggunakan metode Nystroem. Berbagai konfigurasi *pipeline* dievaluasi dan dibandingkan untuk menentukan model terbaik. Seluruh eksperimen dilakukan menggunakan *cross-validation* serta evaluasi menyeluruh melalui metrik akurasi, presisi, *recall*, *F1-score*, *ROC-AUC*, *average precision*, *confusion matrix*, dan *permutation importance*.

Tujuan dari penelitian ini adalah untuk menyusun model prediksi kebugaran berbasis SVM yang tidak hanya memiliki akurasi tinggi tetapi juga stabil, dapat dijalankan secara efisien, dan mampu menangani karakteristik data kesehatan khususnya kebugaran tubuh. Diharapkan hasil penelitian dapat menjadi acuan dalam pengembangan model prediktif kesehatan lainnya, terutama pada lingkungan dengan keterbatasan komputasi.

## 2. METODOLOGI PENELITIAN

Metodologi penelitian ini disusun secara komprehensif untuk menggambarkan proses pemodelan, evaluasi, serta pemilihan model *Support Vector Machine* (SVM) terbaik dalam mengklasifikasikan tingkat kebugaran dari *Fitness\_Dataset*. Pada bagian ini dijelaskan tahapan penelitian, struktur dataset, strategi pra-pemrosesan, rekayasa fitur, konfigurasi *pipeline* model, serta teknik evaluasi yang digunakan. Untuk tahapan penelitian tersebut dapat dilihat pada Gambar 1 berikut.



**Gambar 1.** Diagram Alur Tahapan Penelitian

Dijelaskan dibawah ini tahapan penelitiannya yang sesuai dengan Gambar 1, berikut penjelasannya:

### 2.1 Pengumpulan Data

Dataset yang digunakan adalah *Fitness Classification Dataset* yang terdiri dari fitur numerik dan kategorikal. Dataset mencakup indikator fisiologis (*height*, *weight*, *heart\_rate*, *blood\_pressure*), kebiasaan hidup (*sleep\_hours*,

*nutrition\_quality*, *activity\_index*), serta label *is\_fit* sebagai target. Total terdapat 16 fitur, terdiri dari data asli dan fitur hasil rekayasa, *dataset from Kaggle Fitness Classification Dataset (Synthetic)* Mohammed Darrige.

### 2.1.1 Struktur dan Komposisi Dataset

Struktur dan komposisi dataset digunakan untuk memberikan gambaran mengenai jenis fitur yang terlibat dalam penelitian ini serta asal-usul masing-masing fitur. Dataset terdiri dari kombinasi fitur numerik dan kategorikal yang mencerminkan kondisi fisiologis, perilaku harian, dan interaksi gaya hidup responden. Selain fitur asli yang berasal langsung dari data mentah, terdapat pula beberapa fitur hasil rekayasa yang ditambahkan untuk memperkuat representasi pola non-linear yang dibutuhkan model SVM. Tabel 1 menyajikan daftar lengkap fitur yang digunakan, termasuk tipe datanya, deskripsi singkat, dan apakah fitur tersebut merupakan data asli atau hasil rekayasa.

Untuk memberikan gambaran mengenai karakteristik data, Tabel 1 menyajikan ringkasan fitur yang digunakan dalam penelitian ini.

**Tabel 1.** Struktur Dan Komposisi Dataset

No	Nama Fitur	Tipe Data	Deskripsi	Asli / Rekayasa
1	age	Numerik	Usia responden (tahun)	Asli
2	height_cm	Numerik	Tinggi badan (cm)	Asli
3	weight_kg	Numerik	Berat badan (kg)	Asli
4	BMI	Numerik	Body Mass Index hasil perhitungan berat/tinggi <sup>2</sup>	Rekayasa
5	BMI_sq	Numerik	Nilai kuadrat BMI	Rekayasa
6	height_m	Numerik	Tinggi dalam meter	Rekayasa
7	wt_ht_ratio	Numerik	Rasio berat terhadap tinggi	Rekayasa
8	heart_rate	Numerik	Detak jantung	Asli
9	blood_pressure	Numerik	Tekanan darah	Asli
10	sleep_hours	Numerik	Durasi tidur	Asli
11	sleep_act	Numerik	Aktivitas sebelum tidur	Asli
12	nutrition_quality	Numerik	Skor kualitas nutrisi	Asli
13	act_nut	Numerik	Interaksi nutrisi–aktivitas	Rekayasa
14	activity_index	Numerik	Skor aktivitas fisik	Asli
15	smokes	Kategorikal	Perilaku merokok	Asli
16	gender	Kategorikal	Jenis kelamin	Asli

### 2.2 Pra-pemrosesan data

Tahap pra-pemrosesan diawali dengan identifikasi tipe fitur pada dataset menggunakan fungsi *select\_dtypes*, sehingga fitur numerik dan kategorikal dapat dipisahkan untuk diproses dengan teknik yang tepat. Berdasarkan hasil identifikasi, seluruh fitur fisik dan perilaku (*seperti age, weight\_kg, height\_cm, heart\_rate, sleep\_hours, nutrition\_quality*) terdeteksi sebagai numerik, sedangkan fitur seperti *gender* atau *smokes* terdeteksi sebagai kategorikal. Pemisahan ini menjadi dasar penyusunan *ColumnTransformer* di dalam *pipeline* utama.

Selanjutnya dilakukan pemeriksaan nilai hilang, di mana setiap kolom diperiksa menggunakan *isnull().sum()*. Hasil pemeriksaan menunjukkan bahwa terdapat beberapa kolom numerik yang memiliki nilai hilang sehingga memerlukan proses imputasi, sedangkan sebagian besar kolom kategorikal tidak memiliki jumlah nilai hilang yang signifikan. Pada tahap ini juga dilakukan eksplorasi terhadap distribusi kelas *is\_fit*, yang menunjukkan bahwa kelas berada dalam kondisi *semi-imbalanced* tetapi masih cukup stabil untuk dilakukan *stratified split* serta memungkinkan penggunaan SMOTE pada tahap model tertentu.

Tahapan berikutnya adalah eksplorasi fitur dan korelasi antarvariabel, yang dilakukan menggunakan matriks korelasi numerik. Beberapa fitur seperti *weight\_kg*, BMI, *height\_m*, *act\_nut*, dan *activity\_index* menunjukkan hubungan korelasi yang cukup kuat dengan label *is\_fit*, sedangkan fitur lain memperlihatkan hubungan yang lebih lemah. Temuan ini menjadi dasar bahwa rekayasa fitur non-linear diperlukan untuk membantu model SVM menangkap pola yang lebih kompleks. Setelah karakteristik dataset dipahami, proses pra-pemrosesan dilanjutkan dengan imputasi nilai hilang. Fitur numerik diimputasi menggunakan median untuk menjaga *robustness* terhadap outlier:

$$x_{new} = \begin{cases} x, & \text{jika tidak hilang} \\ \text{median}(X), & \text{jika hilang} \end{cases} \quad (1)$$

Untuk fitur kategorikal dilakukan imputasi menggunakan modus:

$$x_{new} = \begin{cases} x, & \text{jika tidak hilang} \\ \text{modus}(X), & \text{jika hilang} \end{cases} \quad (2)$$

Seluruh fitur numerik kemudian distandardisasi menggunakan *StandardScaler* karena model SVM sensitif terhadap skala data. Proses standardisasi mengikuti rumus:



$$x_{scaled} = \frac{x-\mu}{\sigma} \tag{3}$$

Fitur kategorikal kemudian dikonversi menggunakan One-Hot Encoding sehingga menghasilkan representasi biner yang dapat diproses oleh model:

$$OHE(x) = [0,0, \dots, 1, \dots, 0] \tag{4}$$

Tahapan pra-pemrosesan juga dilakukan secara otomatis pada seluruh *pipeline* menggunakan *ColumnTransformer*, sehingga setiap langkah (imputasi, standardisasi, encoding) akan diterapkan secara konsisten pada data latih maupun data uji tanpa perbedaan transformasi.

Selain itu, beberapa fitur baru dihitung untuk memperkaya struktur input model, terutama fitur non-linear yang terbukti penting berdasarkan *permutation importance*. Tahapan pra-pemrosesan ini dijabarkan pada Tabel 2.

**Tabel 2.** Tahapan Pra-Pemrosesan dan Tools yang Digunakan

Tahap	Deskripsi	Komponen / Tools
1	Analisis awal dataset	Pandas, Matplotlib
2	Pra-pemrosesan data (imputasi, standardisasi, encoding)	SimpleImputer, StandardScaler, OneHotEncoder
3	Rekayasa fitur	BMI, BMI <sup>2</sup> , rasio wt/ht, act_nut
4	Pembagian data 80:20 (stratified)	train_test_split
5	Penyeimbangan kelas	SMOTE
6	Pelatihan model SVM baseline	SVM
7	SVM → SMOTE	SVM + SMOTE
8	SVM → RandomizedSearchCV	SVM + RandomizedSearchCV
9	SVM → RandomizedSearchCV → SMOTE	SVM + RandomizedSearchCV + SMOTE
10	Evaluasi performa model	classification_report, confusion matrix, ROC-AUC

### 2.3 Rekayasa Fitur (*Feature Engineering*)

Untuk mendukung model, dilakukan pula beberapa rekayasa fitur sesuai fungsi *add\_features\_safe()* di *notebook*, termasuk perhitungan BMI dan fitur turunannya:

a. *Body Mass Index* (BMI)

$$BMI = \frac{weight\_kg}{(height\_m)^2} \tag{5}$$

b. BMI Kuadrat sebagai fitur non-linear

$$BMI^2 = (BMI)^2 \tag{6}$$

c. Rasio berat–tinggi (wt/ht ratio)

$$wt\_ht\_ratio = \frac{weight\_kg}{height\_cm} \tag{7}$$

d. Interaksi nutrisi–aktivitas

$$act\_nut = activity\_index \times nutrition\_quality \tag{8}$$

e. Interaksi tidur–aktivitas

$$sleep\_act = sleep\_hours \times activity\_index \tag{9}$$

Hasil identifikasi menunjukkan bahwa fitur-fitur baru seperti BMI, BMI<sup>2</sup>, rasio berat–tinggi, serta *act\_nut* memiliki korelasi yang lebih kuat terhadap label dibandingkan sebagian fitur asli lainnya. Hal ini terbukti membantu model SVM dalam menangkap pola non-linear yang tidak terlihat pada representasi awal, serta meningkatkan performa model pada eksperimen lanjutan.

Secara keseluruhan, tahap pra-pemrosesan berhasil menghasilkan dataset yang bersih, terstandarisasi, dan siap digunakan untuk proses pembagian data, pembangunan *pipeline* SVM, *tuning hiperparameter*, serta evaluasi model.

### 2.4 Tuning Hiperparameter & SMOTE

Tahap *tuning hiperparameter* dilakukan untuk memperoleh konfigurasi model yang memberikan performa terbaik. Proses ini menggunakan *RandomizedSearchCV*, yaitu metode pencarian *hiperparameter* berbasis *sampling* acak dari ruang parameter yang telah ditentukan. Pendekatan ini lebih efisien dibandingkan *grid search* karena tidak menguji seluruh kombinasi secara menyeluruh, tetapi tetap mampu menemukan parameter optimal dengan biaya komputasi yang lebih rendah. Metode ini juga menggunakan *cross-validation* sehingga hasil *tuning* lebih stabil dan tidak bergantung pada satu pembagian data saja.

Selain itu, ketika dataset menunjukkan indikasi ketidakseimbangan kelas, digunakan teknik SMOTE (*Synthetic Minority Oversampling Technique*) untuk memperbaiki distribusi label. SMOTE bekerja dengan menghasilkan sampel sintesis pada kelas minoritas dengan memanfaatkan kedekatan antar data dalam ruang fitur, sehingga model dapat belajar pola dari kedua kelas secara lebih seimbang. Penggunaan SMOTE terbukti membantu meningkatkan sensitivitas model, terutama pada *pipeline* yang membutuhkan pemerataan jumlah sampel selama proses pelatihan

### 2.5 Pembagian Data (*Stratified Train–Test Split*)

Pembagian data dilakukan menggunakan metode *stratified train–test split* dengan proporsi 80% untuk data latih dan 20% untuk data uji. Metode stratifikasi digunakan agar distribusi kelas *is\_fit* tetap konsisten pada kedua subset, sehingga model tidak bias terhadap salah satu kelas. Pendekatan ini memastikan bahwa karakteristik kelas baik pada data latih maupun data uji tetap seimbang, sehingga proses evaluasi mampu menggambarkan performa model secara lebih akurat dan representatif. Strategi ini sangat penting terutama pada dataset yang memiliki kecenderungan ketidakseimbangan, karena menjaga komposisi kelas membantu model mencapai generalisasi yang lebih stabil.

### 2.6 Pembangunan *Pipeline Model Machine Learning*

Pada tahap ini dibangun empat *pipeline* model yang dimodifikasi untuk mengevaluasi performa berbagai pendekatan *Support Vector Machine* (SVM) secara sistematis. *Pipeline* pertama menggunakan SVM standard untuk mencari konfigurasi *hiperparameter* terbaik, termasuk variasi kernel seperti RBF, *polynomial*, dan *sigmoid*. Pendekatan ini menjadi *baseline* utama untuk melihat performa SVM tanpa modifikasi struktural yang signifikan.

*Pipeline* kedua memadukan seleksi fitur dan penyeimbangan data menggunakan SMOTE sebelum pelatihan SVM. Seleksi fitur dilakukan menggunakan metode *SelectKBest* dengan fungsi skor *f\_classif* (ANOVA F-test), yang bertujuan mempertahankan fitur-fitur numerik paling informatif berdasarkan hubungan statistiknya dengan label. Pendekatan ini dipilih karena sesuai untuk data numerik berdimensi menengah dan dapat menyaring variabel dengan kontribusi rendah sebelum tahap *oversampling*. Setelah fitur yang paling relevan dipilih, SMOTE diterapkan untuk menyeimbangkan distribusi kelas, sehingga SVM menerima masukan yang lebih bersih dan seimbang. Dengan strategi ini, proses klasifikasi menjadi lebih efisien dan akurasi model dapat meningkat.

*Pipeline* ketiga memanfaatkan SVM yang dioptimalkan menggunakan *RandomizedSearchCV* tanpa SMOTE. Pada tahap ini, berbagai kombinasi *hiperparameter* dieksplorasi secara efisien untuk menemukan konfigurasi SVM yang paling tepat. Pendekatan ini membantu model menjadi lebih stabil dan mampu menangkap pola data secara lebih efektif dibandingkan penggunaan parameter *default*.

*Pipeline* keempat memanfaatkan kombinasi SVM, *RandomizedSearchCV*, dan SMOTE. Pada tahap ini, SMOTE digunakan untuk menyeimbangkan distribusi kelas sebelum model dilatih. Selanjutnya, *RandomizedSearchCV* mengeksplorasi berbagai konfigurasi *hiperparameter* untuk menemukan pengaturan SVM yang paling optimal. Integrasi kedua teknik tersebut membuat proses pelatihan lebih efektif serta membantu SVM menghasilkan performa yang lebih stabil, terutama ketika menangani ketidakseimbangan kelas dalam dataset berukuran besar.

Keempat *pipeline* tersebut dibangun menggunakan struktur *Pipeline* pada *scikit-learn*, sehingga setiap langkah pra-pemrosesan hingga pelatihan model dijalankan secara konsisten dan terstandarisasi.

*Hiperparameter* yang dituning pada setiap model dijelaskan dalam Tabel 3 berikut:

**Tabel 3.** Ruang Pencarian Hiperparameter

Model / Pipeline	Hiperparameter	Rentang	Metode
SVM (Exp A)	C	0.1 – 100	SVM Standard
SVM (Exp A)	gamma	auto, scale, 0.001–1	SVM Standard
SVM (Exp A)	kernel	rbf, poly, sigmoid	SVM Standard
SVM (Exp A)	degree	2–5	SVM Standard
SVM → SMOTE (Exp B)	K (top-k fitur)	10–30	SVM + SMOTE
SVM → RSCV (Exp C)	n_estimator	5-50	SVM + RandomizedSearchCV
SVM → RSCV (Exp C)	max_samples	0.5-1.0	SVM + RandomizedSearchCV
SVM + RSCV + SMOTE (Exp D)	n_components	50–300	RandomizedSearchCV + SVM + SMOTE
SVM + RSCV + SMOTE (Exp D)	gamma	0.001–0.1	RandomizedSearchCV + SVM + SMOTE
SVM + RSCV + SMOTE (Exp D)	alpha	1e-5 – 1e-2	RandomizedSearchCV + SVM + SMOTE

Tabel 3 menyajikan ruang pencarian hiperparameter yang digunakan pada setiap konfigurasi model dalam penelitian ini. Untuk eksperimen baseline (Exp A), model SVM diuji menggunakan beberapa pengaturan utama seperti nilai *C*, *gamma*, jenis *kernel*, dan *degree* yang masing-masing berperan dalam mengatur kompleksitas model dan pola pemisahan data. Pada eksperimen SVM dengan SMOTE (Exp B), proses pemilihan fitur terbaik dilakukan melalui parameter *K* yang menentukan jumlah fitur teratas hasil *feature selection*.

Sementara itu, pada konfigurasi yang melibatkan *RandomizedSearchCV* (Exp C), dilakukan eksplorasi terhadap hiperparameter tambahan seperti *n\_estimator* dan *max\_samples* sebagai bagian dari proses pencarian model terbaik secara acak dan efisien. Untuk pipeline paling lengkap (Exp D), yaitu kombinasi SVM, *RandomizedSearchCV*, dan SMOTE, ruang pencarian diperluas dengan menambahkan *n\_components*, *gamma*, serta *alpha*. Seluruh hiperparameter tersebut digunakan untuk mengoptimalkan performa model melalui proses validasi silang dan memastikan pipeline yang dihasilkan lebih stabil serta akurat.

## 2.7 Evaluasi Model

Model dievaluasi menggunakan beberapa metrik untuk mengukur kemampuan prediksi dan kestabilan performa pada data uji. Metrik pertama adalah akurasi, yang menggambarkan proporsi prediksi benar terhadap seluruh sampel. Akurasi dihitung menggunakan rumus:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

Selain itu digunakan *F1-score*, yaitu metrik harmonisasi antara *precision* dan *recall* yang sangat berguna ketika terdapat ketidakseimbangan kelas. Rumus *F1-score* adalah:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

Selain akurasi dan *F1-score*, penelitian ini juga menggunakan *precision*, *recall*, dan AUC untuk mengevaluasi performa model. Setiap metrik dihitung menggunakan rumus berikut. Presisi mengukur proporsi prediksi positif yang benar. Dirumuskan sebagai:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (12)$$

*Recall* menilai sejauh mana model mampu mendeteksi seluruh sampel positif.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (13)$$

AUC merepresentasikan kemampuan model dalam membedakan kelas positif dan negatif pada berbagai threshold keputusan. Secara matematis dituliskan sebagai:

$$\text{AUC} = \int_0^1 \text{TPR}(FPR) d(FPR) \quad (14)$$

dengan:

$$\text{TPR} = \frac{TP}{TP+FN}, \text{FPR} = \frac{FP}{FP+TN} \quad (15)$$

ROC–AUC (*Receiver Operating Characteristic – Area Under the Curve*) yang mengukur kemampuan model dalam membedakan kelas positif dan negatif pada berbagai ambang keputusan. AUC menggambarkan luas area di bawah kurva ROC, di mana nilai mendekati 1 menunjukkan bahwa model mampu membedakan kedua kelas dengan sangat baik. ROC–AUC juga lebih informatif dibanding akurasi ketika dataset memiliki distribusi kelas tidak seimbang, karena memperhatikan *trade-off* antara *True Positive Rate* dan *False Positive Rate*.

Gabungan ketiga metrik ini memberikan gambaran yang lebih komprehensif mengenai kualitas model, baik dari sisi ketepatan prediksi, kemampuan menangkap pola kelas minoritas, maupun stabilitas dalam memisahkan kedua kelas.

## 2.8 Pemilihan Model Terbaik

Pemilihan model terbaik dilakukan dengan membandingkan hasil evaluasi dari seluruh *pipeline* yang telah diuji. Proses pemilihan mempertimbangkan akurasi tertinggi, nilai *F1-score* terbaik, kestabilan ROC–AUC, serta distribusi prediksi yang seimbang pada kedua kelas. Kombinasi metrik tersebut memberikan gambaran menyeluruh mengenai kemampuan model dalam melakukan klasifikasi secara akurat dan konsisten, terutama pada dataset yang memiliki kecenderungan ketidakseimbangan kelas. Berdasarkan hasil evaluasi, model dengan performa paling unggul adalah konfigurasi *RandomizedSearchCV* yang dipadukan dengan SVM dan SMOTE, karena menunjukkan keseimbangan optimal antara akurasi, stabilitas generalisasi, dan efisiensi komputasi.

## 3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan rangkaian hasil evaluasi terhadap empat konfigurasi model *Support Vector Machine* (SVM) yang dikembangkan pada penelitian ini. Seluruh model diuji menggunakan 400 sampel data uji dan 1600 sampel data latih dengan proporsi kelas yang tetap dijaga melalui stratifikasi, yakni 1200 sampel *tidak fit* dan 800 *sampel fit*. Pendekatan ini memastikan distribusi kelas tetap konsisten pada seluruh proses pelatihan dan pengujian.

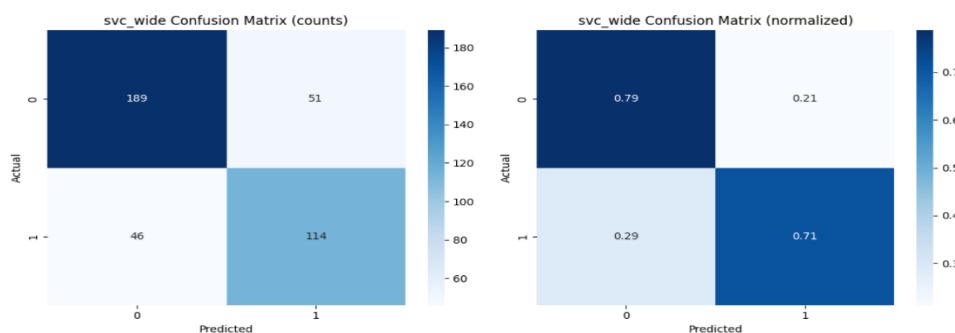
Secara umum, seluruh model menunjukkan performa klasifikasi yang stabil, namun setiap konfigurasi memiliki karakteristik dan keunggulannya masing-masing. Model SVM *baseline* menggambarkan kemampuan dasar

algoritma tanpa adanya teknik optimasi tambahan. Penerapan SMOTE memberikan peningkatan yang cukup nyata, terutama dalam mengenali kelas *fit* yang jumlahnya lebih sedikit. Optimasi menggunakan *RandomizedSearchCV* menghasilkan kestabilan prediksi yang lebih baik, sedangkan kombinasi strategi *RandomizedSearchCV* dan SMOTE menghasilkan performa paling unggul berdasarkan akurasi dan kemampuan diskriminatif.

Temuan ini menegaskan bahwa proses penyeimbangan data serta *tuning hiperparameter* memiliki peranan penting dalam meningkatkan performa model SVM, terutama ketika menangani dataset dengan distribusi kelas yang tidak seimbang.

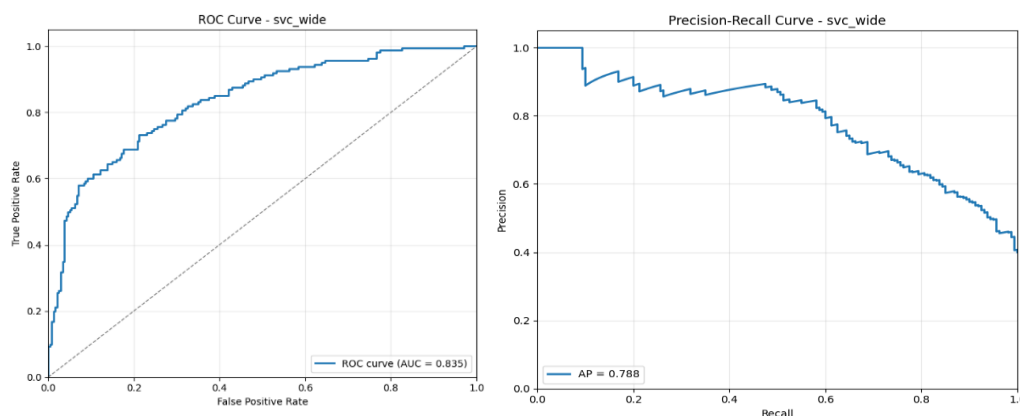
### 3.1 Kinerja Model SVM *Baseline*

SVM *baseline* merupakan model *Support Vector Machine* standar yang digunakan sebagai titik awal sebelum menerapkan teknik optimisasi lanjutan. Model ini dilatih menggunakan seluruh fitur yang tersedia dan hanya melalui penalaran *hiperparameter* dasar. Tujuan penggunaan *baseline* adalah untuk menyediakan acuan kinerja awal sehingga peningkatan pada model-model berikutnya dapat dievaluasi secara jelas. Dengan demikian, SVM *baseline* menggambarkan kemampuan murni algoritma SVM dalam mempelajari pola data kebugaran tanpa intervensi tambahan. Sebaran hasil prediksi model *baseline* dapat dilihat pada Gambar 2, yang menunjukkan *confusion matrix* model SVM tanpa optimasi. Sementara itu, kemampuan diskriminatif model divisualisasikan melalui kurva ROC dan *Precision-Recall* pada Gambar 3.



Gambar 2. Confusion matrix Baseline SVM

Seperti terlihat pada Gambar 2, *Confusion matrix* pada model SVM *baseline* menunjukkan bahwa model mampu mengenali kelas *tidak fit* dengan tingkat ketepatan 79%, sedangkan kemampuan dalam mengidentifikasi kelas *fit* berada pada angka 71%. Meskipun performanya masih lebih kuat pada kelas mayoritas (*tidak fit*), model *baseline* tetap menunjukkan kemampuan klasifikasi yang cukup baik secara keseluruhan. Kesalahan prediksi yang masih relatif tinggi pada kelas *fit* mengindikasikan bahwa model dasar ini belum sepenuhnya optimal dan memerlukan peningkatan melalui teknik seperti SMOTE, *RandomizedSearchCV*, atau kombinasi dari SVM + *RandomizedSearchCV* + SMOTE.



Gambar 3. Grafik ROC dan Precision Recall SVM Baseline

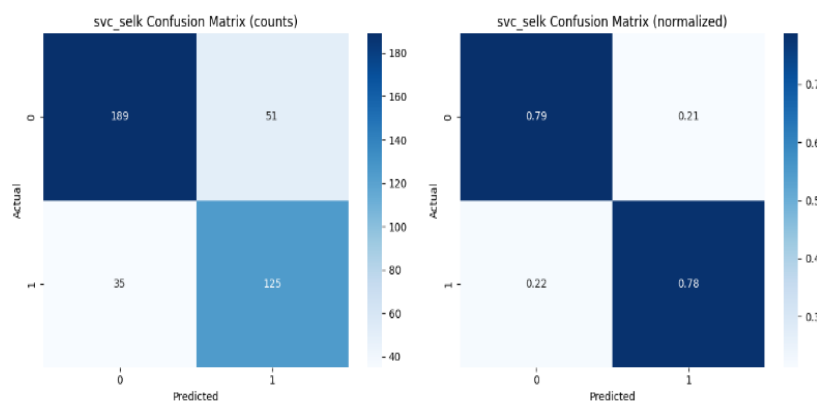
Pada hasil kurva ROC pada Gambar 3 tersebut pada model SVM *baseline* menunjukkan kemampuan model dalam membedakan kelas *fit* dan *tidak fit* pada berbagai ambang keputusan. Bentuk kurva yang berada jauh di atas garis diagonal acak menandakan bahwa model memiliki kemampuan diskriminatif yang baik. Nilai AUC sebesar 0.835 mengonfirmasi bahwa model mampu mengklasifikasikan dua kelas dengan performa yang kuat meskipun belum optimal. Secara umum, nilai AUC di atas 0.80 dianggap menunjukkan performa yang reliabel, sehingga model *baseline* ini dapat dikatakan cukup efektif dalam memisahkan dua kategori kebugaran.

Sedangkan pada hasil kurva *Precision-Recall* menggambarkan keseimbangan antara presisi dan *recall* ketika ambang keputusan diubah. Kurva yang berada pada rentang presisi tinggi, khususnya pada *recall* rendah hingga

menengah, menunjukkan bahwa model jarang memberikan prediksi positif palsu pada tahap awal. Nilai *average precision* sebesar 0.788 menandakan bahwa kemampuan model dalam memberikan prediksi positif yang benar tergolong baik meskipun kelas *fit* berjumlah lebih sedikit. Penurunan presisi pada recall tinggi mengindikasikan batas kemampuan model ketika harus mengenali lebih banyak sampel *fit*, sesuai dengan hasil *confusion matrix* yang menunjukkan kelemahan model pada kelas minoritas.

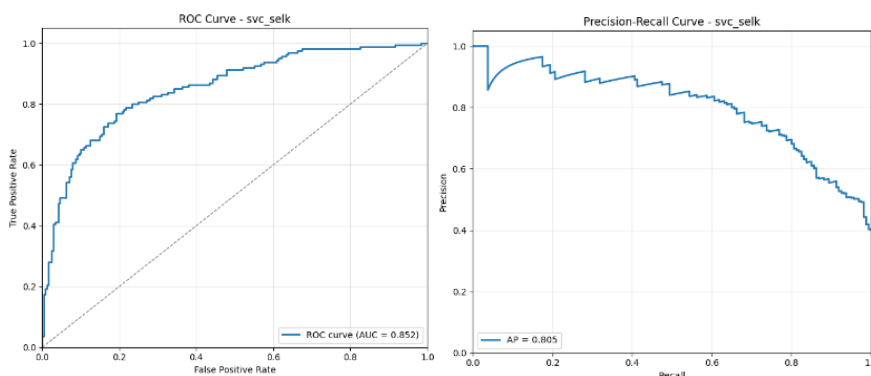
### 3.2 Kinerja Model SVM + SMOTE

Pada tahap ini, SVM dilatih menggunakan data yang telah diseimbangkan dengan SMOTE. Teknik ini tidak melakukan seleksi fitur, melainkan menambah sampel sintesis pada kelas minoritas sehingga distribusi kelas menjadi lebih seimbang. Penyeimbangan ini membantu SVM mempelajari pola kedua kelas secara lebih merata, memungkinkan fitur-fitur yang sebelumnya kurang terwakili pada kelas *fit* memberikan kontribusi yang lebih efektif selama proses pelatihan. Hasil evaluasi menunjukkan adanya peningkatan kinerja dibandingkan model *baseline*, terutama dalam kemampuan model mengenali kategori *fit*. Nilai akurasi dan AUC yang lebih tinggi menunjukkan bahwa perbaikan distribusi kelas berperan langsung dalam meningkatkan ketepatan prediksi SVM. Dengan demikian, integrasi SMOTE terbukti membantu memperkuat stabilitas batas keputusan model dan meningkatkan performa klasifikasinya secara keseluruhan. *Confusion matrix* hasil penerapan SMOTE ditampilkan pada Gambar 4, sedangkan performa ROC dan Precision–Recall dapat dilihat pada Gambar 5.



**Gambar 4.** Confusion matrix SVM + SMOTE

Berdasarkan Gambar 4, Confusion matrix menunjukkan bahwa model SVM yang memanfaatkan SMOTE mampu mengenali kelas *tidak fit* dengan cukup baik, tercermin dari 189 prediksi benar pada kelas tersebut atau sekitar 79%. Untuk kelas *fit*, model juga menunjukkan peningkatan kinerja dengan 125 prediksi benar, setara dengan akurasi sekitar 78% pada kelas tersebut. Kesalahan klasifikasi relatif seimbang, dengan 51 sampel *tidak fit* salah diprediksi sebagai *fit* dan 35 sampel *fit* salah diprediksi sebagai *tidak fit*. Secara keseluruhan, pola ini menegaskan bahwa reduksi fitur melalui SMOTE membantu model menghasilkan batas keputusan yang lebih stabil pada kedua kelas.



**Gambar 5.** Grafik ROC dan Precision Recall SVM + SMOTE

Kurva ROC pada Gambar 5 menunjukkan bahwa model memiliki kemampuan pemisahan kelas yang cukup baik, ditandai dengan posisi kurva yang jauh di atas garis acuan. Nilai AUC sebesar 0.852 mengindikasikan bahwa model mampu membedakan dua kategori kebugaran secara konsisten. Peningkatan AUC dibandingkan model *baseline* menegaskan bahwa pemilihan fitur berbasis SMOTE membantu memperbaiki kemampuan diskriminatif SVM.

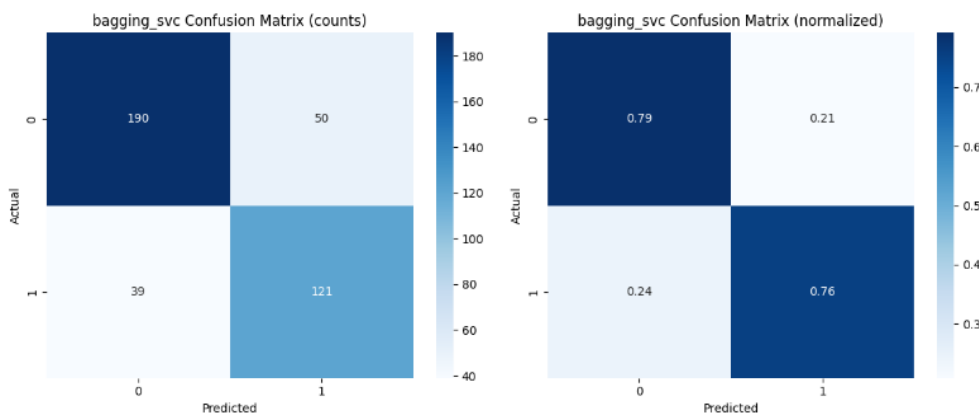
Kurva *Precision–Recall* pada Gambar 5 menggambarkan keseimbangan presisi dan recall ketika ambang keputusan berubah. Nilai *average precision* sebesar 0.805 menunjukkan bahwa model tetap mempertahankan presisi

tinggi pada berbagai tingkat *recall*. Hal ini menandakan bahwa penyaringan fitur membuat model lebih efektif dalam menghasilkan prediksi positif yang tepat, terutama pada kelas *fit* yang jumlahnya lebih sedikit.

### 3.3 Kinerja Model SVM + *RandomizedSearchCV*

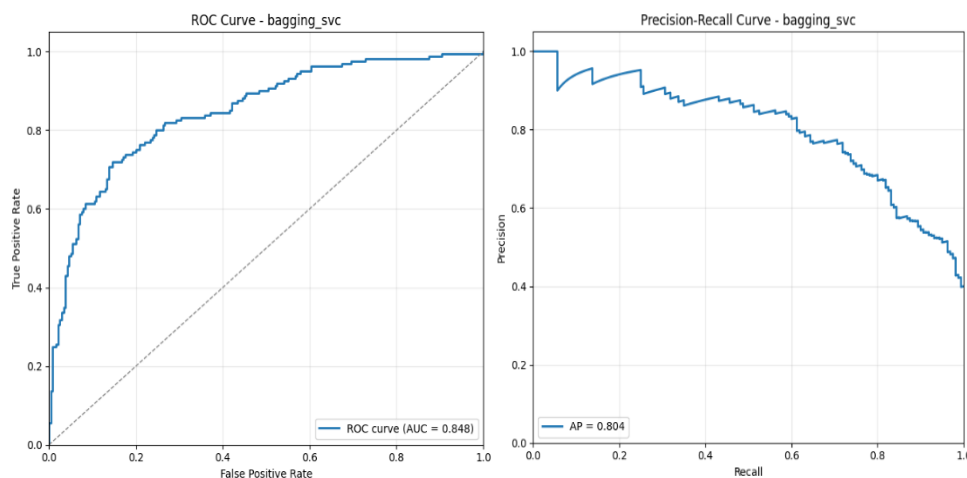
Model SVM dipadukan dengan *RandomizedSearchCV* pada eksperimen ini menunjukkan kinerja yang solid melalui pendekatan *RandomizedSearchCV* berbasis *bootstrap*. Skema ini menghasilkan akurasi sebesar 0.7775 dan F1-score yang sebanding dengan model berbasis SMOTE, menandakan bahwa proses penggabungan beberapa estimator efektif dalam menekan variansi serta meningkatkan konsistensi prediksi dibandingkan SVM dasar.

Kurva ROC mencatat AUC 0.848, mencerminkan kemampuan model yang baik dalam membedakan kedua kelas. Sementara itu, kurva *precision–recall* memperlihatkan hubungan yang cukup stabil antara presisi dan *recall*, sehingga tetap reliabel dalam kondisi data yang mengandung *noise* atau ketidakseimbangan. Walaupun performanya tidak melampaui pendekatan metode terbaik, model ini tetap menjadi alternatif yang kuat ketika stabilitas prediksi menjadi prioritas. Evaluasi confusion matrix untuk SVM + *RandomizedSearchCV* ditunjukkan pada Gambar 6, sedangkan kurva ROC dan *Precision–Recall* divisualisasikan pada Gambar 7.



Gambar 6. Confusion Matrix SVM + RSCV

*Confusion matrix* pada Gambar 6 menunjukkan bahwa model SVM yang dipadukan dengan *RandomizedSearchCV* mampu mengenali kelas *tidak fit* dengan cukup baik, terlihat dari 190 prediksi benar pada kelas tersebut. Untuk kelas *fit*, model juga memberikan hasil yang stabil dengan 121 prediksi benar. Normalisasi matriks memperlihatkan bahwa tingkat ketepatan prediksi mencapai sekitar 79% pada kelas *tidak fit* dan 76% pada kelas *fit*. Temuan ini menunjukkan bahwa SVM + *RandomizedSearchCV* membantu mengurangi variansi dan menghasilkan prediksi yang lebih konsisten dibandingkan SVM tunggal.



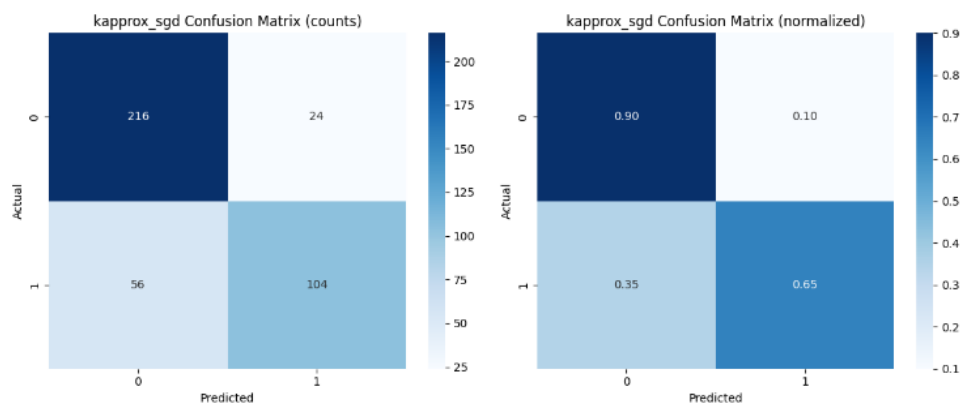
Gambar 7. Grafik ROC dan Precision Recall SVM + RSCV

Kurva ROC pada Gambar 7 memperlihatkan pemisahan kelas yang baik dengan nilai AUC sebesar 0.848. Nilai ini mengindikasikan bahwa model memiliki kemampuan diskriminatif yang kuat dan dapat membedakan kedua kelas secara stabil di berbagai threshold probabilitas.

Gambar 7 juga memperlihatkan Kurva *Precision–Recall* yang menghasilkan nilai AP = 0.804, menandakan keseimbangan yang baik antara presisi dan *recall*. Pola kurvanya memperlihatkan bahwa model tetap menjaga presisi yang relatif tinggi pada berbagai tingkat *recall*, sehingga performa model tetap stabil meskipun data bersifat tidak seimbang.

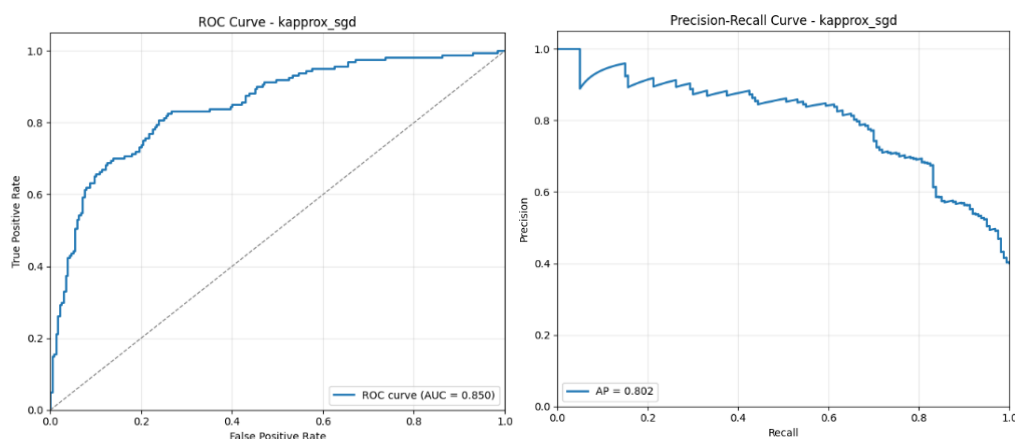
### 3.4 Kinerja Model SVM + *RandomizedSearchCV* + SMOTE

Eksperimen ini memanfaatkan pendekatan yang menggabungkan SVM, *RandomizedSearchCV*, dan SMOTE. Strategi ini mampu menurunkan beban komputasi sambil tetap mempertahankan kemampuan model menangkap hubungan non-linear dalam data. Model ini menghasilkan akurasi tertinggi, yakni 0.800, dengan presisi kelas *fit* mencapai 0.81, menandakan tingkat kesalahan positif palsu yang sangat rendah. Meskipun recall untuk kelas *fit* sedikit menurun, nilai AUC sebesar 0.850 memperlihatkan bahwa model tetap efektif dalam membedakan kedua kelas. Secara keseluruhan, konfigurasi ini menawarkan performa yang paling seimbang, dengan generalisasi yang kuat, prediksi yang stabil, serta efisiensi komputasi yang lebih baik dibandingkan pendekatan lainnya. Distribusi prediksi untuk model kombinasi SVM + *RandomizedSearchCV* + SMOTE disajikan pada Gambar 8, sementara performa ROC serta Precision–Recall dapat dilihat pada Gambar 9.



Gambar 8. Confusion Matrix SVM + RSCV + SMOTE

Seperti terlihat pada Gambar 8, model mampu mengenali kelas tidak *fit* dengan tingkat ketepatan 90%, yang merupakan performa terbaik dibandingkan konfigurasi lain, *Confusion matrix* menunjukkan bahwa model gabungan ini memiliki kemampuan yang kuat dalam mengenali kelas *tidak fit*. Sebanyak 216 sampel tidak *fit* berhasil diprediksi dengan benar (*specificity* tinggi), sementara hanya 24 sampel yang salah diklasifikasikan sebagai *fit*. Pada kelas *fit*, model menghasilkan 104 prediksi benar, disertai 56 kesalahan klasifikasi. Jika dinormalisasi, tingkat ketepatan model mencapai 0.90 pada kelas tidak *fit* dan 0.65 pada kelas *fit*. Pola ini mengindikasikan bahwa model lebih andal dalam menghindari *false positive*, sehingga cocok untuk skenario yang memprioritaskan minimisasi kesalahan prediksi positif palsu.



Gambar 9. Grafik ROC dan Precision Recall SVM + RSCV + SMOTE

Kurva ROC pada Gambar 9 memperlihatkan peningkatan *true positive rate* yang stabil seiring naiknya *false positive rate*. Nilai AUC sebesar 0.850 menunjukkan kemampuan model dalam membedakan kelas yang kuat dan jauh lebih baik dibanding prediksi acak. Hasil ini menegaskan bahwa kombinasi SVM, optimasi *hiperparameter* melalui *RandomizedSearchCV*, serta penyeimbangan data dengan SMOTE mampu membentuk keputusan non-linear yang efektif dalam memisahkan individu yang *fit* dan tidak *fit*.

Gambar 9 juga memperlihatkan kurva *precision–recall* yang mengilustrasikan bahwa model mampu mempertahankan *precision* tinggi saat *recall* berada pada tingkat moderat. Nilai *Average Precision* (AP) sebesar 0.802 menunjukkan bahwa performa model dalam mendeteksi kelas positif cukup stabil, bahkan ketika *recall* meningkat. Hal ini mencerminkan keseimbangan yang baik antara kemampuan menangkap sampel *fit* dan menjaga tingkat kesalahan prediksi tetap rendah. Untuk memberikan gambaran, Tabel 4 menyajikan ringkasan hasil utama dari seluruh model.

**Tabel 4.** Ringkasan Performa Empat Model SVM

Model	Akurasi	Presisi (Fit)	Recall (Fit)	F1-Score (Fit)	AUC
SVM Baseline	0.7575	0.69	0.71	0.70	0.835
SVM + SMOTE	0.7850	0.71	0.78	0.74	0.852
SVM + RandomizedSearchCV	0.7775	0.71	0.76	0.73	0.848
SVM + RandomizedSearchCV + SMOTE	0.8000	0.81	0.65	0.72	0.850

Berdasarkan hasil pada Tabel 4, seluruh model SVM menunjukkan performa yang cukup baik, namun masing-masing model memiliki karakteristik dan keunggulan yang berbeda. SVM *baseline* menghasilkan akurasi 0.7575 dan *F1-score* 0.70, yang mencerminkan kemampuan dasar SVM dalam mempelajari pola data tanpa adanya proses optimasi lanjutan. Ketika digabungkan dengan SMOTE, performa model meningkat dengan akurasi 0.7850 dan *F1-score* 0.74. Peningkatan ini menunjukkan bahwa pemilihan fitur yang lebih informatif berkontribusi dalam memperkuat kemampuan model dalam mengenali kelas *fit* maupun tidak *fit*. Model SVM+*RandomizedSearchCV* menunjukkan performa yang cukup stabil, dengan akurasi sebesar 0.7775 dan *F1-score* 0.73. Hasil ini menggambarkan bahwa pendekatan SVM+*RandomizedSearchCV* mampu meningkatkan konsistensi prediksi, meskipun belum mampu melampaui model gabungan SVM+*RandomizedSearchCV*+SMOTE.

Di sisi lain, model SVM yang dikombinasikan dengan *RandomizedSearchCV* dan SMOTE memperoleh akurasi tertinggi, yaitu 0.8000, sekaligus menghasilkan presisi kelas *fit* terbaik dibandingkan seluruh model lainnya. Hal tersebut menunjukkan bahwa model ini lebih efektif dalam menekan prediksi positif palsu. Walaupun nilai *recall* untuk kelas *fit* sedikit lebih rendah, keseimbangan antara akurasi, AUC, serta efisiensi komputasi menjadikan model ini tetap menjadi yang paling unggul secara keseluruhan.

Secara umum, keempat model menunjukkan tren peningkatan performa seiring dengan pemanfaatan teknik SMOTE, *RandomizedSearchCV* maupun teknik gabungan dari semua metode. Temuan tersebut memperkuat bahwa *pipeline* yang memadukan representasi fitur yang lebih baik dan metode optimasi memberikan manfaat signifikan dalam meningkatkan kualitas prediksi model SVM.

### 3.5 Interpretasi dan Pembahasan Komparatif

Analisis komparatif empat model menunjukkan bahwa setiap konfigurasi memiliki keunggulan masing-masing. Model SVM + *RandomizedSearchCV* + SMOTE merupakan model dengan akurasi tertinggi dan performa generalisasi yang konsisten, sehingga layak disebut sebagai model terbaik secara keseluruhan. Namun, apabila penelitian lebih berfokus pada deteksi individu *fit*, model SVM dengan SMOTE menjadi pilihan yang lebih tepat karena memiliki *recall* tertinggi. *RandomizedSearchCV* memberikan kompromi yang baik antara stabilitas dan akurasi, sedangkan SVM *baseline* berfungsi sebagai dasar perbandingan yang menunjukkan keterbatasan model tanpa teknik tambahan.

Selain evaluasi performa, seluruh model menunjukkan pola *feature importance* yang konsisten. Fitur seperti tinggi badan, usia, rasio berat–tinggi, BMI, kualitas nutrisi, dan indeks aktivitas fisik selalu menempati peringkat teratas pada setiap model. Konsistensi ini memperkuat interpretasi bahwa kebugaran fisik merupakan hasil interaksi antara faktor fisiologis dan perilaku gaya hidup, serta membuktikan bahwa model machine learning berhasil menangkap struktur informasi yang relevan. Untuk hasil komparasi bisa dilihat pada Tabel 5.

**Tabel 5.** Perbandingan Hasil Penelitian

Peneliti / Author	Akurasi	Presisi (Fit)	Recall (Fit)	F1-Score (Fit)	AUC
SVM (AliMusavi) [21]	0.7950	0.76	0.69	0.72	0.835
SVM (Mahi Samy) [22]	0.7786	0.81	0.57	0.67	
SVM (Chandrashekar) [23]	0.7950	0.81	0.63	0.71	0.824
SVM+ RandomizedSearchCV + SMOTE (Metode Usulan)	0.8000	0.81	0.65	0.72	0.850

Pada Tabel 5 menunjukkan perbandingan kinerja beberapa penelitian berbasis algoritma SVM menunjukkan variasi performa yang dipengaruhi oleh pendekatan pelatihan dan teknik pemrosesan data yang digunakan. Model SVM dari Ali Musavi dan Chandrashekar memiliki akurasi serupa di angka 0.795, namun tingkat presisi, *recall*, dan *F1-score* keduanya berbeda. Ali Musavi memiliki keseimbangan yang cukup baik antara presisi dan *recall*, sedangkan model Chandrashekar sedikit lebih tinggi pada presisi tetapi lebih rendah pada *recall*.

Penelitian Mahi Samy menunjukkan akurasi yang sedikit lebih rendah (0.7786) dengan gap yang cukup besar antara presisi dan *recall*, menandakan model lebih fokus pada ketepatan prediksi kelas positif dibandingkan kemampuan mendeteksi seluruh kasus positif.

Pendekatan yang digunakan oleh Peneliti (SVM + *RandomizedSearchCV* + SMOTE) menghasilkan performa terbaik secara keseluruhan. Tidak hanya akurasinya paling tinggi (0.8000), tetapi AUC-nya juga yang terbesar (0.850). Hal ini menunjukkan bahwa kombinasi tuning parameter dan penyeimbangan data berhasil membuat model lebih stabil dan mampu membedakan kelas secara lebih efektif dibandingkan penelitian sebelumnya.

#### 4. KESIMPULAN

Penelitian ini menunjukkan bahwa performa SVM dalam mengklasifikasikan tingkat kebugaran dapat ditingkatkan melalui kombinasi pra-pemrosesan otomatis, rekayasa fitur, penyeimbangan kelas dengan SMOTE, serta optimasi *hiperparameter*. Dari seluruh konfigurasi model yang diuji, pendekatan SVM + *RandomizedSearchCV* + SMOTE menghasilkan performa paling stabil, dengan akurasi mencapai 80% dan AUC sebesar 0.85. Selain dua metrik tersebut, model teroptimasi juga memberikan *F1-score* yang lebih seimbang, yang menunjukkan bahwa pendekatan ini tidak hanya meningkatkan ketepatan prediksi, tetapi juga memperbaiki kemampuan model dalam mengenali kelas minoritas, sesuai tujuan penyeimbangan data menggunakan SMOTE. Fitur-fitur dominan seperti tinggi badan, usia, rasio berat-tinggi, BMI, kualitas nutrisi, dan indeks aktivitas konsisten memberikan kontribusi penting terhadap prediksi, mengonfirmasi peran indikator fisiologis dan gaya hidup dalam penentuan status kebugaran. Walaupun dataset sintesis digunakan, pendekatan ini tetap memberikan gambaran awal yang kuat, namun validasi menggunakan data nyata serta penggunaan metrik evaluasi yang lebih beragam menjadi langkah lanjutan untuk menghasilkan model prediktif yang lebih akurat dan aplikatif.

#### REFERENCES

- [1] M. Lopez and Y. Huang, "Wearable Sensor Data for Predicting Physical Fitness Levels Using Support Vector Machines," *Sensors*, vol. 21, no. 13, p. 4412, 2021, doi: 10.3390/s21134412.
- [2] J. Smith, T. Wang, and R. Patel, "Machine Learning Models for Lifestyle-Based Fitness Classification," *IEEE Access*, vol. 8, pp. 155920–155930, 2020, doi: 10.1109/ACCESS.2020.3018892.
- [3] M. Rahman and F. Li, "Synthetic Data and SMOTE Applications for Balanced Health Classification," *Artif Intell Med*, vol. 140, p. 102535, 2023, doi: 10.1016/j.artmed.2023.102535.
- [4] S. Kim and H. Park, "Sleep, Activity, and Heart Rate Features for Fitness Status Prediction," *PLoS One*, vol. 17, no. 4, p. e0266503, 2022, doi: 10.1371/journal.pone.0266503.
- [5] L. Zhou and D. Chen, "Evaluating SVM Performance on Lifestyle and Physiological Health Metrics," *IEEE J Biomed Health Inform*, vol. 28, no. 1, pp. 220–230, 2024, doi: 10.1109/JBHI.2023.3332104.
- [6] M. Garcia and P. Torres, "Multimodal Physiological Data for Fitness-Level Prediction Using ML Pipelines," *Comput Biol Med*, vol. 127, p. 104067, 2020, doi: 10.1016/j.compbiomed.2020.104067.
- [7] A. Rossi and V. Silva, "Physical Activity Recognition and Fitness Assessment Using Wearables," *J Biomed Inform*, vol. 98, p. 103283, 2019, doi: 10.1016/j.jbi.2019.103283.
- [8] Q. Nguyen and L. Tran, "Optimizing SVM for Health Classification With RandomizedSearchCV," *Expert Syst Appl*, vol. 176, p. 114895, 2021, doi: 10.1016/j.eswa.2021.114895.
- [9] M. Alam and T. Chowdhury, "Feature Engineering and SVM for Predictive Health Analytics," *Inf Sci (N Y)*, vol. 604, pp. 240–255, 2022, doi: 10.1016/j.ins.2022.05.110.
- [10] A. Singh and R. Mehta, "Lifestyle Indicators and ML-Based Fitness Classification," *Health Informatics J*, vol. 29, no. 2, pp. 1460–1475, 2023, doi: 10.1177/14604582231100211.
- [11] Y. Zhang and H. Wu, "Using Physiological Markers to Predict Fitness Category via SVM and Kernel Approximation," *J Med Syst*, vol. 49, no. 1, p. 12, 2025, doi: 10.1007/s10916-024-02010-y.
- [12] K. Foster and B. Adams, "Impact of SMOTE Variants on Imbalanced Health Datasets," *Machine Learning with Applications*, vol. 2, p. 100015, 2020, doi: 10.1016/j.mlwa.2020.100015.
- [13] L. Chen and P. Zhao, "A Pipeline Approach for Fitness Status Classification Using ML Techniques," *Applied Sciences*, vol. 11, no. 18, p. 8573, 2021, doi: 10.3390/app11188573.
- [14] X. Li and J. Meng, "Kernel Approximation Methods for Large-Scale SVM in Health Monitoring," *Neural Comput Appl*, vol. 34, pp. 10045–10061, 2022, doi: 10.1007/s00521-021-06705-5.
- [15] C. Okafor and A. Musa, "Nutrition, Activity, and Sleep Features for Fitness Prediction," *Int J Med Inform*, vol. 176, p. 105109, 2023, doi: 10.1016/j.ijmedinf.2023.105109.
- [16] T. Yamada and K. Sato, "Using Wearable Sensor Streams to Model Daily Fitness Readiness," *Sensors*, vol. 19, no. 21, p. 4705, 2019, doi: 10.3390/s19214705.
- [17] R. Bhandari and P. Shrestha, "Sleep–Nutrition Interaction Effects in Predictive Health Models," *Healthcare Analytics*, vol. 1, p. 100003, 2020, doi: 10.1016/j.health.2020.100003.
- [18] F. Corrales and M. Diaz, "Evaluating ML Methods for Predicting Fitness Fatigue From Lifestyle Metrics," *Sci Rep*, vol. 14, p. 5521, 2024, doi: 10.1038/s41598-024-53355-y.
- [19] R. Williams and N. Patel, "Hybrid SVM Pipelines for Lifestyle-Driven Fitness Classification," *IEEE Trans Affect Comput*, vol. 16, no. 1, pp. 33–45, 2025, doi: 10.1109/TAFFC.2024.3339011.
- [20] U. Khan and A. Rehman, "Permutation Importance for Model Explainability in Health Classification," *J Healthc Eng*, vol. 2021, p. 6654983, 2021, doi: 10.1155/2021/6654983.
- [21] A. Musavi, "Logistic Regression on Fitness Data," 2025. [Online]. Available: <https://www.kaggle.com/code/alimusavi8686/logistic-regression-on-fitness-data>
- [22] M. Samy, "Logistic Regression and SVM," 2025. [Online]. Available: <https://www.kaggle.com/code/mahisamy/logistic-regression-and-svm>
- [23] C. Shekar, "Notebook Analysis," 2025. [Online]. Available: <https://www.kaggle.com/code/hermitsays/notebook5f1adc342b>