

RUN-IO: APLIKASI UNTUK REKOMENDASI LATIHAN PELARI MENGGUNAKAN RANDOM FOREST

1st Josey Alexander Takesan
Teknologi Informasi
Telkom University Surabaya
Surabaya, Indonesia

[joseyalexander@student.telkomuniversi
ty.ac.id](mailto:joseyalexander@student.telkomuniversi
ty.ac.id)

2nd Yohanes Setiawan
Program Studi Teknologi Informasi
Universitas Telkom, Kampus Surabaya
Surabaya, Jawa Timur, Indonesia

[yohanessetiawan@telkomuniversity.ac.
id](mailto:yohanessetiawan@telkomuniversity.ac.
id)

3rd Bernadus Anggo Seno Aji
Program Studi Teknologi Informasi
Universitas Telkom, Kampus Surabaya
Surabaya, Jawa Timur, Indonesia

[bernadusanggosenoaji@telkomuniversi
ty.ac.id](mailto:bernadusanggosenoaji@telkomuniversi
ty.ac.id)

Abstrak — Meningkatnya popularitas olahraga lari diiringi dengan tingginya risiko cedera atau *running related injuries* (RRIs) dimana seringkali kebingungan dalam menentukan program latihan yang aman dan sesuai dengan kemampuan individu. Penelitian ini bertujuan merancang, membangun, dan mengevaluasi RUN-IO, sebuah aplikasi *mobile* yang memberikan rekomendasi latihan lari yang dipersonalisasi untuk meminimalkan risiko tersebut. Menggunakan algoritma *Random Forest Regressor*, aplikasi memproses data pengguna (usia, tinggi, berat, jenis kelamin) untuk memprediksi tiga parameter kunci: Kecepatan, Waktu, dan Jarak Lari. Model dilatih menggunakan dataset "*Running Calorie Burn*" dari *Kaggle* dan dievaluasi kinerjanya dengan metrik *Mean Absolute Error* (MAE) dan *R-Squared* (R^2). Hasil evaluasi menunjukkan bahwa model prediksi memenuhi kriteria kelayakan implementasi. Model *Running Speed* menunjukkan performa paling unggul dengan nilai R^2 mencapai 0.8990 dan MAE 0.56 km/jam. Pengujian fungsional aplikasi dan evaluasi antarmuka oleh pengguna menunjukkan aplikasi dinilai sangat baik. Penelitian ini berhasil mengimplementasikan *model machine learning* yang valid ke dalam aplikasi RUN-IO yang fungsional, menyediakan alat bantu praktis bagi pelari untuk berlatih secara lebih aman dan terukur agar dapat menghindari cedera.

Kata kunci— Aplikasi *mobile*, *Running Related Injuries*, *Running*, *Random Forest Regressor*, Personalisasi latihan

I. PENDAHULUAN

Lari adalah salah satu olahraga fisik yang paling populer di dunia [1]. Penelitian menunjukkan bahwa bahkan lari intensitas sedang selama 10 menit pada level 50% VO_2 peak dapat memberikan manfaat signifikan bagi fungsi kognitif dan suasana hati. Hal ini dibuktikan dengan peningkatan tingkat *arousal* dan *pleasure* yang signifikan, dengan *effect size Cohen's d* sebesar 1.55 untuk *arousal* dan 0.98 untuk *pleasure* (Damrongthai et al., 2021). Semenjak tahun 2020 jumlah orang yang melakukan kegiatan lari semakin meningkat, namun dengan meningkatnya jumlah orang yang melakukan kegiatan lari, risiko cedera oleh karena berlari juga semakin meningkat (García-Arrabé et al., 2024). Risiko cedera meningkat 1,40 kali akibat berlari, ini menunjukkan

bahwa intensitas lari tanpa adaptasi yang tepat dapat meningkatkan risiko cedera [4].

Dalam konteks ini, *smartphone* dapat digunakan untuk memantau dan mengelola intensitas serta jarak lari seperti yang dijelaskan dalam penelitian Straczkiwicz, *smartphone* memiliki berbagai sensor *built-in* seperti *accelerometer* dan GPS yang dapat mengukur aktivitas fisik secara objektif dan berkelanjutan [5]. Kemampuan *smartphone* untuk mengumpulkan data secara *real-time*, dikombinasikan dengan algoritma *Random Forest Regressor*, menghasilkan aplikasi yang dapat memberikan rekomendasi latihan sesuai dengan atribut pengguna seperti umur, tinggi badan, berat badan dan juga *gender*. Salah satu aplikasi serupa yang telah ada di pasaran adalah aplikasi "Strava", yang menawarkan fitur pelacakan aktivitas lari dan analisis performa. Strava memungkinkan pengguna untuk merekam aktivitas lari, membagikan hasil dengan komunitas, dan mendapatkan wawasan tentang kinerja mereka. Namun, Strava tidak secara khusus menargetkan pelari dengan rekomendasi latihan yang dipersonalisasi berdasarkan karakter fisik individu.

Untuk memperdalam pemahaman dan memposisikan kebaruan penelitian ini, dilakukan tinjauan terhadap beberapa studi relevan sebelumnya. Penelitian oleh Martínez-Gramage mengatasi tingginya insiden cedera pada atlet triathlon muda menggunakan metode *Random Forest* untuk mengidentifikasi pola biomekanis penyebab cedera melalui program *gait retraining*. Hasilnya menunjukkan performa model yang baik (AUC-ROC 0.8), namun memiliki keterbatasan pada ukuran sampel yang kecil dan belum adanya integrasi data *smartphone* untuk rekomendasi yang dipersonalisasi [6]. Selanjutnya, tinjauan oleh Amendolara terhadap berbagai metode *machine learning* menunjukkan bahwa *Random Forest* mencapai akurasi prediksi cedera hingga 96%, namun menyoroti tantangan pada kecilnya dataset dan kurangnya personalisasi rekomendasi [7]. Penelitian oleh Straczkiwicz juga mengonfirmasi bahwa sensor *smartphone* mampu mengenali aktivitas dengan akurasi di atas 95%, tetapi menghadapi masalah pada penanganan data yang hilang [5]. Studi lain oleh Van Eetvelde menyimpulkan bahwa metode *tree-based ensemble*

seperti *Random Forest* adalah yang paling menjanjikan untuk prediksi cedera (AUC 0.87), namun menekankan perlunya standar evaluasi yang lebih baik [8]. Terakhir, penelitian oleh Dimmick membuktikan bahwa model *Random Forest* yang dipersonalisasi lebih akurat (hingga 68,9%) dalam mendeteksi kelelahan dibandingkan model berbasis kelompok, meskipun terbatas oleh variabel dari perangkat komersial [9].

Oleh karena itu, penelitian-penelitian tersebut menjelaskan bahwa penggunaan *Random Forest* dan *smartphone* dalam mencegah cedera pada saat olahraga sangat penting dan efektif. Hasil penelitian dari beberapa sumber menunjukkan bahwa model *Random Forest* memiliki akurasi yang tinggi dan transparansi yang baik sehingga dapat menghindari *overfitting*. *Smartphone* juga efisien untuk digunakan karena memiliki sensor *Human Activity Recognition* (HAR) yang baik sehingga dapat mengukur kegiatan sehari-hari. Penggabungan antara model *Random Forest* dan sensor *smartphone* akan menjadi peruntukan yang membedakan dari penelitian sebelumnya.

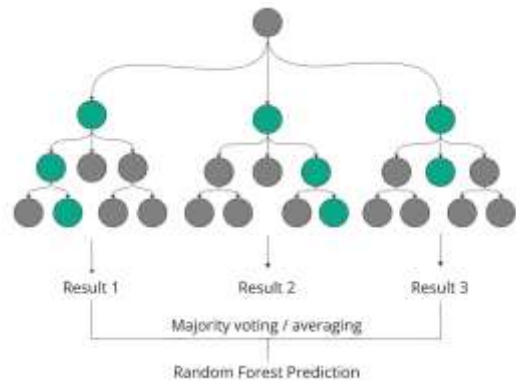
Berdasarkan kesenjangan tersebut, penelitian ini merumuskan dua permasalahan utama yang perlu dipecahkan. Pertama, bagaimana membuat model dan mengevaluasi kinerja model untuk memberikan rekomendasi latihan yang efektif. Kedua, bagaimana cara mengembangkan aplikasi yang memanfaatkan model yang sudah dibuat untuk membantu pelari dalam memantau dan memberikan rekomendasi latihan yang efektif sehingga dapat mengurangi cedera.

Oleh karena itu, penelitian ini bertujuan untuk mengembangkan aplikasi lari yang menggunakan *Random Forest Regressor* untuk analisis pelari seperti kecepatan, jarak tempuh dengan memberikan porsi latihan yang cukup untuk setiap pelari. *Random Forest Regressor* dipilih karena kelebihanannya sebagai metode *ensemble* yang menggabungkan beberapa *Decision Tree*, sehingga meningkatkan akurasi dan mengurangi risiko *overfitting* dibandingkan model tunggal. Model ini juga mampu menangani data berdimensi tinggi dan mengelola variabel yang hilang dengan baik, serta memberikan informasi penting tentang fitur yang berkontribusi terhadap prediksi, memungkinkan pengembangan rekomendasi latihan yang lebih tepat. Aplikasi ini diharapkan dapat membantu pelari terutama bagi orang yang baru mulai melakukan olahraga lari agar dapat melakukan aktivitas lari secara teratur dan dapat mengurangi risiko terjadinya cedera.

II. KAJIAN TEORI

Dasar teori berisi penjelasan konsep penting yang memastikan bahwa suatu penelitian dapat fokus pada topik yang telah ditentukan. Dasar teori digunakan untuk memahami konsep, metode, dan teknologi yang dipakai. Berikut dasar teori yang digunakan dalam penelitian. Running Related Injuries (RRIs) dapat mempengaruhi hingga 85% dari semua pelari. Pelari yang berlari 5 hari per minggunya dapat memiliki prevalensi cedera sebesar 39%, dibandingkan dengan 12% untuk pelari yang berlari 3 hari per minggu. Selain itu, ada kecenderungan bahwa berlari selama 60 menit dalam seminggu dapat melindungi terhadap cedera terkait lari (Fredette et al., 2022).

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan penggunaan metode *machine learning*. Salah satu algoritma yang relevan adalah *Random Forest*, yang termasuk dalam metode *ensemble* berbasis *decision tree*, yaitu model yang bekerja dengan membagi data menjadi subset lebih kecil untuk membuat keputusan [11]. Algoritma ini didasarkan pada kumpulan pohon yang dilatih dan dari data pelatihan tersebut akan rata-rata dari hasil semua pohon dalam model [12].



GAMBAR 1
(Konsep *Random Forest Regressor*)

Berdasarkan Gambar 1, proses kerja *Random Forest Regressor* dimulai dengan membangun beberapa *Decision Tree* dari sampel data yang diambil menggunakan teknik *bootstrap*. Untuk setiap pohon, hanya sebagian fitur yang dipilih secara acak untuk menentukan pembagian (*split*), sebuah pendekatan yang efektif untuk mengurangi *overfitting*. Prediksi akhir untuk regresi diperoleh dengan mengambil nilai rata-rata (*averaging*) dari hasil semua pohon, sehingga menghasilkan prediksi yang lebih stabil dan akurat [13].

Implementasi model ini didukung oleh teknologi yang tepat. Kemampuan *Python* sebagai bahasa pemrograman untuk membangun situs dan aplikasi, sebagaimana dinyatakan oleh Dicoding.com [14], salah satu *framework* yang efisien adalah *Python Flask*. *Python Flask* adalah *framework* mikro web yang ringan dan fleksibel, banyak digunakan untuk membangun aplikasi web dan API dengan cepat menggunakan bahasa *Python* [15].

Setelah model dibangun, kinerjanya perlu dievaluasi dengan menggunakan metrik evaluasi. Metrik evaluasi yang digunakan dalam penelitian ini adalah *Mean Absolute Error* (MAE) dan *R-Squared* (R^2). MAE memberikan informasi tentang kesalahan rata-rata dalam prediksi, sementara R^2 menunjukkan seberapa baik model menjelaskan varians dalam data.

A. *Mean Absolute Error* (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1)$$

B. *R-Squared* (R^2)

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (2)$$

Keterangan:

n : Jumlah total pengamatan
 y_i : Nilai Aktual

$\hat{y}_i$: Nilai yang diprediksi oleh model

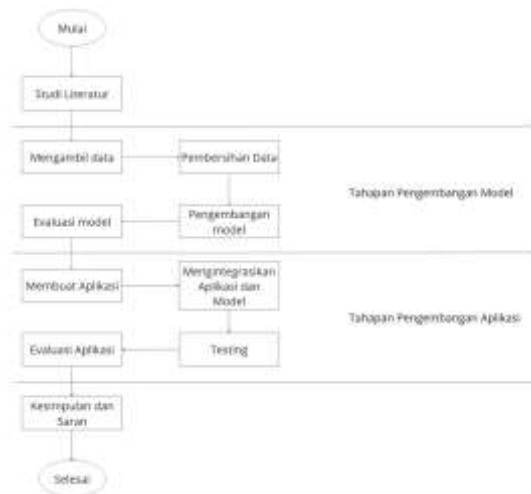
SS_{res} : Jumlah selisih antara nilai aktual dan nilai prediksi

SS_{tot} : Jumlah selisih antara nilai aktual dan rata² nilai aktual

Setelah itu aplikasi akan diuji agar menentukan status kelayakan aplikasi dapat digunakan atau tidak dengan 2 tahap yaitu dengan menggunakan Appium dan Blackbox Testng. Appium adalah alat pengujian otomatisasi yang digunakan untuk menguji aplikasi *mobile*, baik yang berbasis *native*, *hybrid*, maupun *web*. Appium menggunakan *JSON Wire Protocol* untuk komunikasi klien/server, memungkinkan penulisan klien dalam berbagai bahasa yang berbeda [16].

Blackbox Testing adalah metode pengujian perangkat lunak di mana penguji tidak memiliki pengetahuan tentang struktur internal atau kode sumber aplikasi yang diuji. *Black-box testing* sering digunakan untuk menguji aplikasi interaktif, seperti aplikasi *Android*, dan dapat dilakukan dengan alat seperti *Monkey* dan *ARES*, yang menghasilkan *input* acak atau menggunakan strategi eksplorasi yang lebih cerdas untuk meningkatkan efisiensi pengujian [17].

III. METODE



GAMBAR 2
(Alur sistematis penyelesaian)

Seperti yang bisa dilihat pada gambar 2 penelitian ini dimulai dengan studi literatur untuk memahami teori dan konsep mengenai algoritma *Random Forest Regressor* serta kegunaan *smartphone* dalam memonitor dan menganalisis kebiasaan olahraga. Dari pemahaman teori yang sudah didapatkan, selanjutnya adalah mengambil data untuk dikembangkan nantinya menjadi rekomendasi.

A. Pengumpulan Data

Pada penelitian ini, data yang digunakan adalah data sekunder yang diperoleh dari dataset yang tersedia pada situs *Kaggle* yang dapat diakses pada link berikut [Running-Calorie-Burn | Kaggle](#). Metode pengumpulan data yang digunakan dalam penelitian ini adalah metode pengunduhan data dari website *Kaggle*. Berikut adalah langkah-langkah yang dilakukan dalam pengumpulan data yaitu mengakses dataset yang tersedia di platform *Kaggle*, Mengunduh file CSV yang berisi data yang relevan untuk analisis, dan memastikan bahwa data yang diunduh lengkap dan sesuai

dengan kebutuhan penelitian seperti yang bisa dilihat pada tabel 1.

TABEL 1
(Contoh Data)

Gender	Age	Height	Weight	BMI	Running Time	Running Speed	Distance
Male	49	184	74	20.4	78	9.5	3

B. Pengolahan Data

Pada tahap pengolahan data, beberapa proses dilakukan untuk mempersiapkan data yang telah dikumpulkan agar dapat digunakan dalam analisis. Proses ini mencakup pembersihan data, transformasi fitur, dan pelatihan model prediksi.

Proses pengolahan data diawali dengan pembersihan untuk menghilangkan data yang tidak valid atau kosong, memastikan data yang digunakan akurat. Selanjutnya, dilakukan proses *feature engineering* untuk memperkaya dataset. Tahap ini meliputi pembuatan fitur-fitur baru seperti BMI (yang dikategorikan menjadi *Underweight*, *Normal*, *Overweight*, *Obese*), *Pace* (min/km), *Speed Category* (mulai dari *Very Slow* hingga *Very Fast*), dan *Age Group*. Selain itu, dibuat juga fitur interaksi (*Age x Weight*, *Height x Weight*, *Gender x Age*, *BMI x Age*) untuk menangkap hubungan non-linear antar variabel. Terakhir, seluruh data kategorikal seperti *Gender* dan kategori yang baru dibuat ditransformasikan menjadi nilai numerik menggunakan *Label Encoder*. Melalui proses ini, dataset yang semula memiliki 6 fitur berhasil dikembangkan menjadi lebih dari 20 fitur informatif, yang secara signifikan meningkatkan informasi yang dapat digunakan oleh model *machine learning*.

Setelah proses *feature engineering* selesai, data disiapkan untuk tahap pelatihan model. Sebelum model dilatih, seluruh fitur numerik distandarisasi menggunakan *StandardScaler* untuk memastikan semua fitur memiliki skala yang sama, sehingga mencegah fitur dengan nilai besar mendominasi proses pembelajaran. Data kemudian dibagi menjadi data latih (70%) dan data uji (30%) untuk memungkinkan evaluasi performa model yang objektif. Proses pelatihan dilakukan secara terpisah untuk setiap variabel target yaitu *Running Time*, *Running Speed*, dan *Distance* menggunakan subset fitur yang paling relevan.

C. Pengembangan Model

Tahap pelatihan model menggunakan algoritma *Random Forest Regressor*, yang dipilih karena kemampuannya menangani banyak fitur dan mengurangi *overfitting*. Proses pelatihan dilakukan secara terpisah untuk setiap variabel target: *Running Time*, *Running Speed*, dan *Distance*.

Proses optimasi dilakukan melalui *hyperparameter tuning* yang komprehensif, mencakup parameter seperti *n_estimators* (jumlah pohon dalam *forest*), *max_depth* (kedalaman maksimum setiap pohon), *min_samples_split* (jumlah minimum sampel untuk membagi *node*), *min_samples_leaf* (jumlah minimum sampel di setiap daun), *max_features* (proporsi fitur yang dipertimbangkan untuk setiap split), *bootstrap* (penggunaan sampling dengan replacement), dan *max_samples* (proporsi sampel yang digunakan untuk setiap pohon).

Untuk menemukan konfigurasi terbaik, dilakukan eksperimen dengan tiga skenario *hyperparameter* yang

berbeda: pendekatan moderat, kompleksitas tinggi, dan pencegahan *overfitting* yang bisa dilihat pada tabel 2.

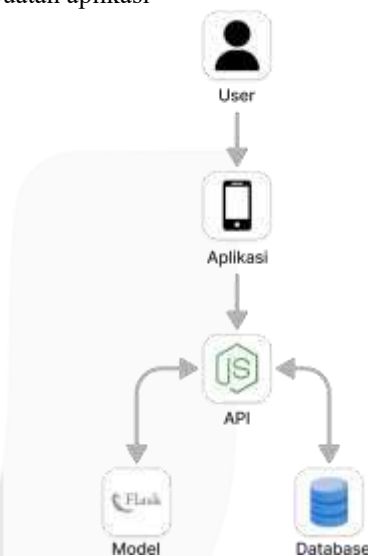
Proses optimasi menggunakan *mean absolute error* dan *R-Squared* untuk memilih model terbaik, karena MAE memberikan interpretasi yang mudah dipahami dalam konteks prediksi nilai kontinu. *RandomizedSearchCV* digunakan untuk mengeksplorasi kombinasi *hyperparameter* secara efisien.

TABEL 2
(Skenario *hyperparameter* yang digunakan)

Skenario	Hyperparameter	Value	Keterangan
Skenario 1	n_estimators	200 – 300	Jumlah pohon yang cukup stabil untuk performa baik tanpa membebani komputasi. Cocok untuk dataset ukuran menengah.
	max_depth	10 – 20	Kedalaman sedang agar model tidak terlalu kompleks, menjaga generalisasi
	min_samples_split	2 – 5	Ukuran minimal pemisahan untuk mencegah overfitting tanpa terlalu membatasi pertumbuhan pohon.
	min_samples_leaf	1 – 2	Nilai rendah agar menangkap detail data, cocok untuk prediksi numerik seperti waktu/tempuh.
	max_features	0.6 – 0.8	Kompromi antara kekayaan fitur vs. variansi model.
	max_samples	0.8 – 0.9	Sampling sebagian data training untuk menjaga keanekaragaman pohon dan mencegah overfitting.
Skenario 2	n_estimators	500 – 1000	Banyak pohon meningkatkan stabilitas dan akurasi, ideal untuk pencapaian performa maksimal.
	max_depth	20 – 30	Kedalaman besar memungkinkan eksplorasi relasi kompleks antar fitur.
	min_samples_split	2 – 3	Relatif kecil agar pohon dapat tumbuh dalam hingga kedalaman maksimum.
	min_samples_leaf	1	Lebih fleksibel untuk memaksimalkan akurasi.
	max_features	0.7 – 1.0	Membolehkan model untuk memanfaatkan lebih banyak fitur—efektif untuk banyak dimensi.
	max_samples	0.9 – 1.0	Menggunakan hampir seluruh data agar performa maksimum bisa tercapai.

Skenario 3	n_estimators	100 – 300	Jumlah pohon dibatasi agar waktu latih cepat dan menghindari overfitting.
	max_depth	5 – 12	Kedalaman pohon kecil untuk membatasi kompleksitas dan menjaga generalisasi. Cocok jika data berisik.
	min_samples_split	5 – 15	Memaksa pohon hanya membelah jika cukup data tersedia, untuk mencegah overfitting.
	min_samples_leaf	2 – 6	Menjaga agar daun tidak terlalu spesifik.
	max_features	0.4 – 0.6	Mengurangi jumlah fitur yang dipertimbangkan, memperkuat regularisasi model.
	max_samples	0.7 – 0.8	Menghindari pelatihan pada keseluruhan data, memperkenalkan variasi dan generalisasi.

D. Pembuatan aplikasi



GAMBAR 3
(Arsitektur Sistem)

Arsitektur sistem ini dirancang dengan tiga komponen utama yang bekerja secara terintegrasi untuk memberikan rekomendasi lari yang dipersonalisasi seperti yang bisa dilihat pada gambar 3.

- **Front-End:** Aplikasi *mobile* dibangun menggunakan *React Native* untuk memastikan kompatibilitas lintas platform. Antarmukanya mencakup fitur-fitur penting seperti autentikasi pengguna, formulir *input* data, halaman utama, pelacakan lari, dan menu "tantangan" untuk menampilkan rekomendasi.
- **Back-End:** Menggunakan *Node.js* untuk membangun API (*Application Programming Interface*) utama yang menghubungkan *front-end* dengan berbagai layanan, termasuk API pihak ketiga untuk cuaca, berita, dan peta. Sementara itu,

model prediksi *Random Forest Regressor* diimplementasikan menggunakan *Python Flask*, yang bertugas menganalisis data pengguna dan menghasilkan rekomendasi latihan.

- **Database:** Seluruh data pengguna, riwayat aktivitas, dan hasil analisis disimpan dalam *Firestore*, yang dipilih karena kemudahan integrasi dan kemampuan sinkronisasi data secara *real-time*.

Alur kerja dimulai saat pengguna memasukkan data di aplikasi. Data ini dikirim melalui API ke *back-end*, di mana model *Random Forest Regressor* akan memprosesnya. Hasil rekomendasi yang telah dipersonalisasi kemudian dikirim kembali ke aplikasi dan ditampilkan di menu "tantangan" untuk diakses oleh pengguna.

TABEL 3
(Spesifikasi perangkat keras)

Kategori	Komponen	Spesifikasi
Pengembangan Model	Platform	Kaggle Notebooks
	GPU	NVIDIA T4 / P100 (disediakan oleh Kaggle)
	RAM	16 GB
Pengembangan Lokal	Processor (CPU)	Intel Core i5 Generasi ke-10 atau setara
	Memory (RAM)	8 GB DDR4
	Penyimpanan	256 GB SSD
Target Pengguna	Perangkat	Android
	Memory (RAM)	Minimal 4 GB
	Penyimpanan	100 MB ruang kosong
	Sensor	GPS, Accelerometer

TABEL 4
(Spesifikasi perangkat lunak)

Kategori	Komponen	Spesifikasi / Versi
Sistem Operasi	Pengembangan	Windows 11 / macOS Monterey
	Target Aplikasi	Android 8.0 (Oreo) atau lebih baru
Pengembangan Model	Bahasa	Python 3.9
	Framework API	Flask 2.0.2
	Library ML	scikit-learn (v1.0.2), pandas (v1.3.4), numpy (v1.20.3)
Pengembangan Frontend	Framework	React-Native
	Backend API	Node JS
Database	Layanan	Firestore
Pengujian	Automation	Appium
	Metode	Blackbox Testing

E. Testing



GAMBAR 4
(Alur testing)

Sesuai dengan gambar 4 tahap *testing* ini akan dilakukan *testing* untuk mencoba semua fitur yang sudah dibuat di dalam aplikasi. Proses *testing* dimulai dengan rencana pengujian dimana tujuan dilakukannya *testing* dan jenis pengujian akan ditentukan. Pengujian ini akan membuat *study case* yang dirancang untuk mencakup semua fitur aplikasi, seperti input data, statistik, *overview*, *tracking running* dan juga memberikan rekomendasi sesuai inputan

data sebelumnya. Pada tahap pengujian ini akan menggunakan Appium untuk membantu dalam pengujian aplikasi. Evaluasi yang menyeluruh ini penting untuk meningkatkan keandalan dan efektivitas aplikasi dalam konteks dunia nyata.

Evaluasi aplikasi bertujuan untuk memastikan fungsionalitas, mengidentifikasi *bug*, dan menilai akurasi prediksi dari model *Random Forest Regressor* yang terintegrasi. Proses ini dilakukan menggunakan alat otomatisasi Appium pada emulator atau perangkat fisik untuk menguji fitur-fitur penting seperti input data, navigasi, dan penyajian rekomendasi.

Kinerja aplikasi dievaluasi berdasarkan tiga kriteria utama: waktu respons aplikasi, akurasi *output* model yang diukur dengan metrik MAE dan R^2 , serta kepuasan pengguna yang dikumpulkan melalui umpan balik. Pengujian fungsional dan kinerja dilakukan secara menyeluruh untuk memastikan setiap fitur berfungsi dengan benar dan aplikasi dapat menangani beban pengguna secara efisien. Evaluasi komprehensif ini sangat penting untuk memperbaiki aplikasi dan memastikan pengalaman pengguna yang memuaskan sebelum siap digunakan.

IV. HASIL DAN PEMBAHASAN

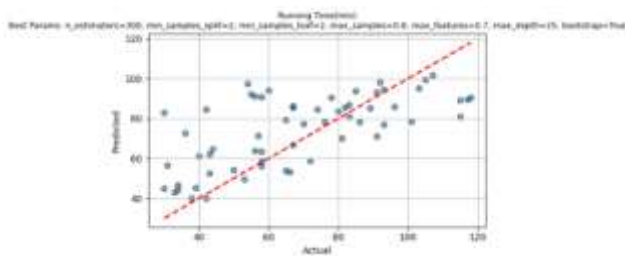
Hasil eksperimen menunjukkan bahwa Skenario 2 memberikan performa terbaik untuk dua target utama, yaitu *Running Time* dan *Running Speed*, baik dari segi nilai R^2 maupun MAE, meskipun menggunakan konfigurasi model yang paling kompleks.

A. Hasil Model

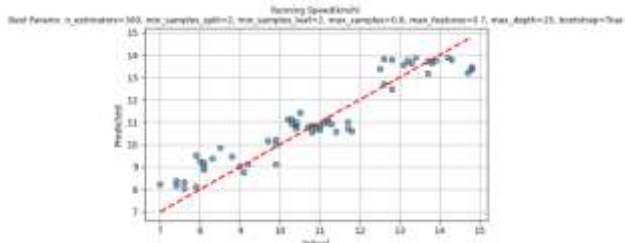
Model prediksi *Running Speed* secara konsisten memberikan hasil paling unggul dengan $R^2 = 0.8990$ dan MAE = 0.56 km/jam, yang telah memenuhi kriteria kelayakan model sebagaimana telah ditetapkan sebelumnya, yaitu $R^2 \geq 0.85$ dan $MAE \leq 0.60$ km/jam. Hal ini menandakan bahwa fitur-fitur yang digunakan sangat efektif dalam menjelaskan variasi kecepatan lari.

Sementara itu, model *Running Time* menunjukkan performa moderat dengan nilai $R^2 = 0.3799$ dan MAE = 14.81 menit pada Skenario 2, yang juga masih berada dalam batas toleransi kelayakan model, yaitu $R^2 \geq 0.35$ dan MAE ≤ 15 menit. Model *Running Time* menjadi salah satu yang sulit diprediksi karena memiliki aspek yang dapat mempengaruhi yaitu kondisi fisik, cuaca dan juga motivasi. Di sisi lain, model *Distance* menunjukkan performa terbaik pada Skenario 1 dengan $R^2 = 0.5497$ dan MAE = 1.96 km, yang telah melampaui ambang minimum kelayakan, yaitu $R^2 \geq 0.50$ dan MAE ≤ 2.00 km.

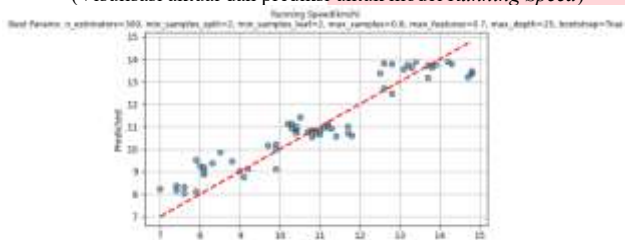
Secara keseluruhan, model prediksi *Running Speed* merupakan model paling akurat dan stabil di antara ketiga target yang dievaluasi, dengan $R^2 > 0.86$ pada seluruh skenario, menunjukkan bahwa fitur-fitur masukan yang digunakan memiliki relevansi tinggi dalam menjelaskan kecepatan lari, dibandingkan prediksi waktu atau jarak tempuh.



GAMBAR 5

(Visualisasi aktual dan prediksi untuk model *Running Time*)

GAMBAR 6

(Visualisasi aktual dan prediksi untuk model *Running Speed*)

GAMBAR 7

(Visualisasi aktual dan prediksi untuk model *Distance*)

- Performa Prediksi *Running Time*

Dari analisis data aktual dan prediksi, model *Running Time* menunjukkan variabilitas yang lebih tinggi pada selisih prediksi dengan parameter terbaik $n_estimator$ 300, $min_samples_split$ 2, $min_samples_leaf$ 2, $max_samples$ 0.8, $max_features$ 0.7, dan max_depth 25. Teridentifikasi adanya beberapa kasus dengan deviasi yang cukup signifikan, yang mengindikasikan tantangan dalam menangkap pola kompleks durasi latihan seperti yang divisualisasikan pada gambar 5. Hal ini sesuai dengan karakteristik *Running Time* yang dipengaruhi oleh banyak faktor eksternal seperti kondisi fisik, cuaca, dan motivasi, sehingga menjadi parameter yang paling sulit diprediksi.

- Performa Prediksi *Running Speed*

Model *Running Speed* menunjukkan konsistensi tinggi dalam prediksi dengan sebagian besar prediksi memiliki selisih minimal dengan nilai aktual dengan parameter terbaik $n_estimator$ 300, $min_samples_split$ 2, $min_samples_leaf$ 2, $max_samples$ 0.8, $max_features$ 0.7, dan max_depth 25. Hal ini mengkonfirmasi bahwa kecepatan lari memiliki pola yang lebih stabil dan dapat diprediksi dibandingkan parameter lainnya seperti yang divisualisasikan pada gambar 6. Konsistensi ini tercermin dalam nilai R^2 yang tinggi di seluruh skenario, menunjukkan kemampuan model yang excellent dalam menjelaskan variasi data kecepatan.

- Performa Prediksi *Distance*

Model *Distance* menunjukkan tingkat akurasi yang memadai dengan distribusi error yang relatif seimbang dengan parameter terbaik $n_estimator$ 300, $min_samples_split$ 2, $min_samples_leaf$ 2, $max_samples$ 0.8, $max_features$ 0.7, dan max_depth 25 seperti yang divisualisasikan pada gambar 7. Hal ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik untuk parameter jarak dan stabilitas yang cukup dalam berbagai kondisi pengujian.

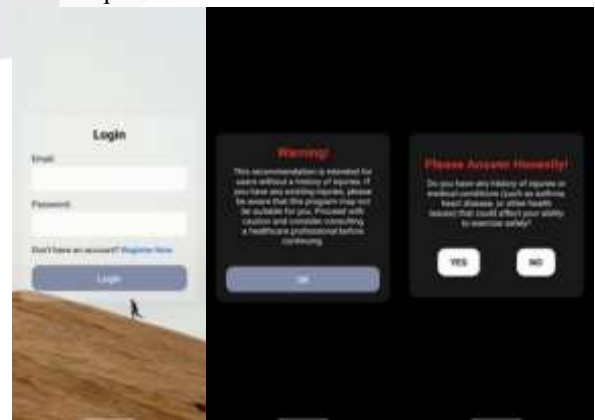
- Status Kelayakan Model untuk Deployment

Berdasarkan evaluasi komprehensif terhadap ketiga parameter melalui berbagai skenario eksperimen, model berada dalam status "Ready for Deployment" dengan catatan *monitoring* khusus untuk parameter *Running Time*. Analisis menunjukkan bahwa:

1. Parameter *Running Speed*: Memberikan performa *excellent* dengan memenuhi seluruh kriteria kelayakan. Model menunjukkan konsistensi tinggi dengan $R^2 > 0.86$ pada seluruh skenario dan mencapai performa optimal pada Skenario 2.
2. Parameter *Distance*: Memenuhi kriteria kelayakan dengan performa terbaik pada Skenario 1, menunjukkan kemampuan generalisasi yang solid dengan $R^2 = 0.5497$ dan $MAE = 1.96$ km.
3. Parameter *Running Time*: Berada pada ambang batas kriteria kelayakan namun masih dalam rentang yang dapat diterima. Kompleksitas prediksi yang tinggi dapat disebabkan oleh faktor-faktor eksternal seperti cuaca, niat dan motivasi.

Model menunjukkan kemampuan prediksi yang bervariasi sesuai dengan tingkat kompleksitas masing-masing parameter. Performa terbaik dicapai pada prediksi *Running Speed* yang menunjukkan stabilitas dan akurasi tinggi, sementara *Running Time* memerlukan perhatian khusus dalam implementasi. Secara keseluruhan, model telah memenuhi kriteria minimum untuk deployment dengan rekomendasi untuk monitoring berkelanjutan, terutama untuk parameter *Running Time* yang memiliki tingkat kompleksitas prediksi yang tinggi.

B. Hasil Aplikasi



GAMBAR 8

(Tampilan halaman login dan *Warning*)

Pada gambar 8 memperlihatkan apabila pengguna masuk menggunakan akun yang sudah dibuat dan setelahnya pengguna akan diberikan peringatan bahwa rekomendasi yang nantinya akan diberikan oleh aplikasi ditujukan untuk pengguna yang sedang tidak cedera atau tidak memiliki cedera serius sebelumnya. Lalu pengguna akan diminta untuk memilih ya dan tidak sesuai kondisi yang sebelumnya sudah disebutkan.



GAMBAR 9
(Tampilan halaman *form-data* dan *home*)

Pada tahap ini yang bisa dilihat pada gambar 9, pengguna diminta untuk mengisi data seperti usia, jenis kelamin, tinggi badan, berat badan, dan frekuensi lari. Data ini kemudian dikirimkan ke *backend* untuk diproses oleh model prediksi yang nantinya akan menjadi rekomendasi latihan. Setelah melengkapi form tersebut maka akan masuk ke halaman utama.



GAMBAR 10
(Tampilan halaman rekomendasi)

Pada gambar 10 menampilkan rekomendasi lari yang dipersonalisasi meliputi durasi, kecepatan, dan jarak tempuh yang dihasilkan langsung oleh model *Random Forest Regressor* berdasarkan input pengguna. Halaman ini juga berfungsi untuk melacak progres harian pengguna terhadap tantangan yang diberikan. Sesuai persetujuan pada halaman peringatan, menu tantangan ini akan tetap kosong jika pengguna memiliki riwayat cedera dan memilih untuk tidak menerima saran latihan.

C. Hasil Testing

Pengujian model dilakukan dengan pendekatan blackbox testing seperti yang bisa dilihat pada tabel 5, di mana proses evaluasi tidak melihat struktur internal dari model *Random Forest Regressor* yang dibangun, melainkan hanya fokus pada hasil output terhadap input yang diberikan. Pendekatan ini digunakan untuk memastikan bahwa model dapat menghasilkan output prediksi yang logis, stabil, dan sesuai ekspektasi pengguna berdasarkan skenario masukan yang telah ditentukan.

TABEL 5
(Hasil *Blackbox Testing*)

No	Fitur yang Diuji	Input yang Diberikan	Hasil Aktual	Status
1	Input Data Pengguna	Usia: 22, Berat: 60 kg, Tinggi: 170 cm, Gender: Male	Data terkirim berhasil	Sesuai
2	Prediksi Kecepatan	Berdasarkan data pengguna	Muncul rekomendasi kecepatan lari	Sesuai
3	Prediksi Durasi Lari	Berdasarkan data pengguna	Muncul rekomendasi durasi lari	Sesuai
4	Prediksi Jarak Lari	Berdasarkan data pengguna	Muncul rekomendasi jarak lari	Sesuai
5	Tampilan UI hasil prediksi	Setelah klik "Process"	Tampilan sesuai desain	Sesuai

Pengujian ini menguji bagaimana interaksi pengguna dengan aplikasi diuji tanpa melihat struktur kode internal. Fokus pengujian adalah memastikan bahwa setiap elemen antarmuka, seperti form input data pengguna, tombol prediksi, serta tampilan hasil, dapat merespons dengan benar dan menampilkan informasi yang sesuai. Hasil pengujian menunjukkan bahwa aplikasi dapat berjalan lancar dan setiap fitur inti bekerja sebagaimana mestinya.



GAMBAR 11
(Hasil Testing *Appium*)

Dari gambar 11 bisa diambil kesimpulan bahwa semua fitur dari *register* sampai dengan pengisian *form* dan pembuatan *challenges* oleh model sudah berjalan dengan baik sebagaimana mestinya dan dapat digunakan oleh pengguna untuk mendapatkan rekomendasi dari model yang sudah dipersonalisasi.

V. KESIMPULAN

Penelitian ini berhasil mengembangkan aplikasi *mobile* yang mampu memberikan rekomendasi lari yang dipersonalisasi menggunakan algoritma *Random Forest*

Regressor. Dengan memanfaatkan data masukan pengguna seperti usia, tinggi, dan berat badan, aplikasi ini memprediksi tiga target utama: Kecepatan Lari (*Running Speed*), Waktu Lari (*Running Time*), dan Jarak (*Distance*). Model prediktif ini dilatih menggunakan dataset "*Running Calorie Burn*" dari *Kaggle* melalui serangkaian proses yang mencakup *preprocessing*, *feature engineering*, dan *hyperparameter tuning* untuk memastikan akurasi dan relevansi hasil.

Dari hasil evaluasi, model *Running Speed* menunjukkan performa terbaik dan paling stabil ($R^2 = 0.8990$; MAE = 0.56), diikuti oleh model *Distance* ($R^2 = 0.5497$; MAE = 1.96) dan model *Running Time* ($R^2 = 0.3799$; MAE = 14.81). Seluruh model telah memenuhi kriteria kelayakan minimum untuk implementasi, sehingga dapat dinyatakan siap untuk digunakan dengan catatan *monitoring* berkala pada prediksi waktu lari. Aplikasi yang dibangun telah teruji secara fungsional, menunjukkan kinerja yang baik, dan memiliki antarmuka yang ramah pengguna, menjadikannya solusi awal yang efektif untuk membantu pelari dalam mengatur porsi latihan secara aman dan mengurangi risiko cedera.

REFERENSI

- [1] D. van Poppel *et al.*, "Risk factors for overuse injuries in short- and long-distance running: A systematic review," *J Sport Health Sci*, vol. 10, no. 1, pp. 14–28, Jan. 2021, doi: 10.1016/j.jshs.2020.06.006.
- [2] C. Damrongthai *et al.*, "Benefit of human moderate running boosting mood and executive function coinciding with bilateral prefrontal activation," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-01654-z.
- [3] M. García-Arrabé *et al.*, "Assessing the Impact of COVID-19 on Amateur Runners' Performance: An Analysis through Monitoring Devices," *Sensors*, vol. 24, no. 8, Apr. 2024, doi: 10.3390/s24082635.
- [4] A. F. DeJong, P. N. Fish, and J. Hertel, "Running behaviors, motivations, and injury risk during the COVID-19 pandemic: A survey of 1147 runners," *PLoS One*, vol. 16, no. 2 February, Feb. 2021, doi: 10.1371/journal.pone.0246300.
- [5] M. Straczekiewicz, P. James, and J. P. Onnela, "A systematic review of smartphone-based human activity recognition methods for health research," Dec. 01, 2021, *Nature Research*. doi: 10.1038/s41746-021-00514-4.
- [6] J. Martínez-Gramage, J. P. Albiach, I. N. Moltó, J. J. Amer-Cuenca, V. H. Moreno, and E. Segura-Ortí, "A random forest machine learning framework to reduce running injuries in young triathletes," *Sensors (Switzerland)*, vol. 20, no. 21, pp. 1–12, Nov. 2020, doi: 10.3390/s20216388.
- [7] A. Amendolara *et al.*, "An Overview of Machine Learning Applications in Sports Injury Prediction," *Cureus*, Sep. 2023, doi: 10.7759/cureus.46170.
- [8] H. Van Eetvelde, L. D. Mendonça, C. Ley, R. Seil, and T. Tischer, "Machine learning methods in sport injury prediction and prevention: a systematic review," Dec. 01, 2021, *Springer Science and Business Media Deutschland GmbH*. doi: 10.1186/s40634-021-00346-x.
- [9] H. L. Dimmick, C. R. van Rassel, M. J. MacInnis, and R. Ferber, "Use of subject-specific models to detect fatigue-related changes in running biomechanics: a random forest approach," *Front Sports Act Living*, vol. 5, 2023, doi: 10.3389/fspor.2023.1283316.
- [10] A. Fredette, J. S. Roy, K. Perreault, F. Dupuis, C. Napier, and J. F. Esculier, "The Association between Running Injuries and Training Parameters: A Systematic Review," Jul. 01, 2022, *National Athletic Trainers' Association Inc.* doi: 10.4085/1062-6050-0195.21.
- [11] H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen, "Decision trees: from efficient prediction to responsible AI," *Front Artif Intell*, vol. 6, Jun. 2023, doi: 10.3389/frai.2023.1124553.
- [12] Suci Amaliah, M. Nusrang, and A. Aswi, "Penerapan Metode Random Forest Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konjiwa Bantaeng," *VARIANSI: Journal of Statistics and Its application on Teaching and Research*, vol. 4, no. 3, pp. 121–127, Dec. 2022, doi: 10.35580/variensiunm31.
- [13] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *Stata Journal*, vol. 20, no. 1, pp. 3–29, Mar. 2020, doi: 10.1177/1536867X20909688.
- [14] D. Intern, "Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya," 2023. [Online]. Available: <https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/>
- [15] Z. Liang, Z. Liang, Y. Zheng, B. Liang, and L. Zheng, "Data analysis and visualization platform design for batteries using flask-based python web service," *World Electric Vehicle Journal*, vol. 12, no. 4, Dec. 2021, doi: 10.3390/wevj12040187.
- [16] G. Fantini, "Continuous integration for End-to-End testing of mobile applications," 2022.
- [17] A. Romdhana, A. Merlo, M. Ceccato, and P. Tonella, "Deep Reinforcement Learning for Black-box Testing of Android Apps," *ACM Transactions on Software Engineering and Methodology*, vol. 31, no. 4, Jul. 2022, doi: 10.1145/3502868.