

**ANALISIS DAN PERANCANGAN APLIKASI PENDAMPING
OLAHRAGA DENGAN ALGORITMA DECISION TREE C4.5**

TUGAS AKHIR



UNIVERSITAS
MA CHUNG

TJIA, TERRY FERDINAND CAHYONO
NIM: 312010012

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG
2024

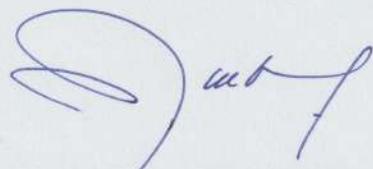
LEMBAR PENGESAHAN
TUGAS AKHIR

**ANALISIS DAN PERANCANGAN APLIKASI PENDAMPING
OLAHRAGA DENGAN ALGORITMA DECISION TREE C4.5**

Oleh:
TJIA, TERRY FERDINAND CAHYONO
NIM. 312010012

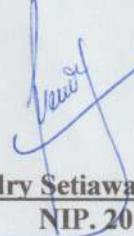
dari:
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG

Dosen Pembimbing 1



Paulus Lucky Tirma Irawan, S.Kom., MT.
NIP. 20100005

Dosen Pembimbing 2



Hendry Setiawan, ST, M.Kom.
NIP. 20100006

Dekan Fakultas Teknologi dan Desain



Prof. Dr. Eng. Romy Budhi, ST., MT.
NIP. 20070035

LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama : Tjia, Terry Ferdinand Cahyono
NIM : 312010012
Fakultas : Fakultas Teknologi dan Desain
Program Studi : Teknik Informatika
Judul Tugas Akhir : Analisis dan Perancangan Aplikasi Pendamping Olahraga dengan Algoritma Decision Tree C4.5

Menyatakan dengan sebenarnya bahwa tugas akhir atau skripsi yang ditulis ini tidak mempunyai persamaan dengan skripsi lain.

Dengan pernyataan ini dibuat tanpa adanya paksaan dari pihak manapun. Apabila pernyataan ini tidak benar, maka akan diberikan sanksi oleh pimpinan fakultas.

Malang, 21 Agustus 2024



Tjia, Terry Ferdinand Cahyono

NIM. 312010012

ABSTRAK

Penelitian ini menyoroti pentingnya asupan nutrisi dalam mendukung performa kesehatan pasca olahraga, terutama dalam pemulihan energi, keseimbangan cairan, dan perbaikan otot. Dengan menggunakan algoritma C4.5, penelitian ini bertujuan untuk mengembangkan model klasifikasi yang menganalisis pengaruh intensitas latihan terhadap kebutuhan nutrisi. Algoritma C4.5 dipilih karena akurasinya yang tinggi dalam menghasilkan model *decision tree*, yang dapat mengidentifikasi pola asupan nutrisi berdasarkan intensitas olahraga. Model yang dihasilkan dibuat dengan data training. Untuk pembuatan model, digunakan 500 data *training* untuk *training* model. Untuk uji akurasi, model diuji dengan menggunakan metode *Confusion Matrix* dengan hasil akurasi yang baik yaitu 100%, diuji dengan menggunakan data *testing* sebanyak 93 data. Hasil penelitian ini diwujudkan dalam bentuk aplikasi berbasis *Progressive Web App* (PWA) yang memberikan rekomendasi latihan dan nutrisi yang dipersonalisasi, memudahkan pengguna dalam merancang program olahraga dan nutrisi tanpa harus menggunakan jasa profesional. Dengan demikian, aplikasi ini diharapkan mampu meningkatkan efisiensi dan efektivitas program latihan dan nutrisi bagi penggunanya. Selain itu, penelitian ini juga berkontribusi dalam menghasilkan model *decision tree* yang akurat dan relevan, yang diintegrasikan dalam aplikasi untuk memberikan asupan nutrisi dan rekomendasi latihan yang tepat. Luaran penelitian ini diharapkan dapat dipublikasikan dalam jurnal nasional terakreditasi, serta menyediakan aplikasi pendamping olahraga yang bermanfaat bagi pengguna.

Kata Kunci: Algoritma C4.5, Asupan nutrisi, *Decision tree*, Intensitas latihan, *Progressive Web App* (PWA).

ABSTRACT

This research highlights the importance of nutritional intake in supporting post-exercise health performance, particularly in energy recovery, fluid balance, and muscle repair. Using the C4.5 algorithm, this study aims to develop a classification model that analyzes the impact of exercise intensity on nutritional needs. The C4.5 algorithm was chosen due to its high accuracy in generating decision tree models, which can identify nutritional intake patterns based on exercise intensity. The model was created using training data, with 500 data points utilized for training the model. For accuracy testing, the model was evaluated using the Confusion Matrix method, resulting in an excellent accuracy rate of 100%, tested with 93 testing data points. The outcomes of this research are realized in the form of a Progressive Web App (PWA) that provides personalized exercise and nutrition recommendations, making it easier for users to design exercise and nutrition programs without professional assistance. Thus, this application is expected to enhance the efficiency and effectiveness of exercise and nutrition programs for its users. Additionally, this research contributes to the production of an accurate and relevant decision tree model, which is integrated into the application to provide precise nutritional intake and exercise recommendations. The outcomes of this study are expected to be published in an accredited national journal and to offer a valuable fitness companion application for users.

Keywords: *C4.5 algorithm, Decision tree, Exercise intensity, Nutritional intake, Progressive Web App (PWA).*

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa yang telah memberikan kesehatan dan kekuatan kepada penulis sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini. Laporan penelitian mengenai Tugas Akhir (TA) ini berjudul “Analisis dan Perancangan Aplikasi Pendamping Olahraga dengan Algoritma Decision Tree C4.5”.

Tujuan dari penulisan laporan ini adalah untuk mengimplementasikan teori-teori yang telah dipelajari selama perkuliahan ke dalam praktik agar dapat digunakan secara nyata. Tidak hanya itu, penulis melakukan kegiatan Tugas Akhir sebagai salah satu syarat kelulusan. Dalam kesempatan ini, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah turut serta dalam penyusunan laporan ini serta pelaksanaan Tugas Akhir hingga selesai, khususnya kepada:

1. Prof. Dr. Eng. Romy Budhi, ST., MT., selaku Dekan Fakultas Teknologi dan Desain Universitas Ma Chung
2. Hendry Setiawan, ST, M.Kom., selaku Ketua Program Studi Teknik Informatika
3. Paulus Lucky Tirma Irawan, S.Kom., MT., selaku Dosen Pembimbing 1 yang telah memberikan asistensi dalam penyusunan Laporan Tugas Akhir
4. Hendry Setiawan, ST, M.Kom., selaku Dosen Pembimbing 2 yang telah memberikan asistensi dalam penyusunan Laporan Tugas Akhir
5. Mochamad Subianto, S.Kom., M.Cs., selaku Ketua Penguji Laporan Tugas Akhir
6. Juniarko Yumana, S.T., sebagai narasumber *personal trainer* yang telah memberikan ilmunya sebagai landasan dalam penyusunan Laporan Tugas Akhir
7. Julius Ronny, sebagai narasumber *nutritionist expert* yang telah memberikan ilmunya sebagai landasan dalam penyusunan Laporan Tugas Akhir

8. Semua rekan dan keluarga yang telah memberikan dukungan kepada penulis dalam menjalankan penelitian Tugas Akhir

Malang, 7 November 2023
Tjia, Terry Ferdinand Cahyono

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
DAFTAR LAMPIRAN	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	2
1.4 Perumusan Masalah	3
1.5 Tujuan Penelitian	3
1.6 Luaran Penelitian	4
1.7 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA	5
2.1 <i>Decision Tree</i>	5
2.2 Algoritma <i>C4.5</i>	6
2.3 Pengertian dan Manfaat Nutrisi	7
2.4 Kekuatan Otot	9
2.5 <i>Body Mass Index (BMI)</i>	10
2.6 <i>Hypertext Preprocessor (PHP)</i>	10
2.7 <i>Oracle</i>	11
2.8 <i>Laravel</i>	12
2.9 <i>Model-View-Controller (MVC)</i>	13
2.10 <i>Hypertext Markup Language (HTML)</i>	13

2.11 <i>Cascading Style Sheet (CSS)</i>	14
2.12 <i>JavaScript</i>	14
2.13 <i>Bootstrap</i>	15
2.14 <i>jQuery</i>	15
2.15 Metode <i>Waterfall Model</i>	16
2.16 <i>Progressive Web App (PWA)</i>	17
2.17 <i>Service Worker</i>	18
2.18 <i>Scikit-learn 1.5.1</i>	18
2.19 <i>Confusion Matrix</i>	19
2.20 Penelitian Terdahulu	20
BAB III RANCANGAN SISTEM	23
3.1 Tahapan Penelitian	23
3.2 Studi Pustaka	24
3.3 Pengumpulan Data	26
3.4 Penentuan <i>Root</i> dan <i>Node</i>	28
3.5 Pembuatan Model <i>Decision Tree</i> dan <i>Rules</i>	30
3.6 <i>Training Model</i>	31
3.7 <i>System Design</i>	31
3.8.1 <i>Requirements</i>	31
3.8.2 <i>System Design</i>	32
3.8.3 <i>Implementation and Development</i>	33
3.8.4 <i>Testing</i>	33
BAB IV HASIL DAN PEMBAHASAN	34
4.1 Perhitungan Nilai <i>Entropy</i> dan <i>Gain</i>	34
4.2 Pembuatan Model <i>Decision Tree</i>	35
4.3 Penyederhanaan Model <i>Decision Tree</i>	37

4.4 <i>Testing Model</i>	40
4.5 <i>Rules</i> dari Model <i>Decision Tree</i>	41
4.6 Hasil Wawancara Narasumber	43
4.7 Arsitektur Sistem	47
4.8 Diagram IPO (<i>input, process, output</i>)	48
4.9 Integrasi Model ke Dalam Aplikasi	49
4.10 Implementasi pada Aplikasi	49
4.11 Pengujian Aplikasi dengan Contoh <i>Case</i>	54
BAB V KESIMPULAN DAN SARAN	57
5.1 Kesimpulan	57
5.2 Saran	58
DAFTAR PUSTAKA	59
LAMPIRAN	62

DAFTAR TABEL

Tabel 3. 1 Rumus Perhitungan Nutrisi Sebelum <i>Workout</i>	26
Tabel 3. 2 Rumus Perhitungan Nutrisi Sedang <i>Workout</i>	26
Tabel 3. 3 Rumus Perhitungan Nutrisi Setelah <i>Workout</i>	26
Tabel 3. 4 Contoh Data untuk <i>Training</i> dan <i>Testing</i> Model	28

DAFTAR GAMBAR

Gambar 2. 1 Konsep Alur <i>Decision Tree</i>	5
Gambar 2. 2 Konsep Dasar Pembentukan <i>Decision Tree</i>	5
Gambar 2. 3 Diagram Relasi antara <i>Model, View, Controller</i>	13
Gambar 2. 4 Metode <i>Waterfall</i>	17
Gambar 2. 5 <i>Service Worker</i>	18
Gambar 3. 1 Tahapan Penelitian	23
Gambar 3. 3 <i>Flowchart</i> Proses Pencarian <i>Root</i>	29
Gambar 3. 4 <i>Flowchart</i> Proses Perhitungan <i>Information Gain</i>	30
Gambar 3. 5 Pembentukan <i>Node</i>	30
Gambar 3. 6 <i>Design UI Input</i>	32
Gambar 3. 7 Design <i>UI Output</i>	32
Gambar 4. 1 Hasil Model <i>Decision Tree</i>	35
Gambar 4. 2 Hasil Penyederhanaan Model <i>Decision Tree</i>	37
Gambar 4. 3 Hasil <i>Confusion Matrix</i>	40
Gambar 4. 4 Diagram IPO	48
Gambar 4. 5 Tampilan Aplikasi pada <i>Desktop</i>	50
Gambar 4. 6 Tampilan Aplikasi pada <i>Mobile</i>	52
Gambar 4. 8 Tampilan Luaran pada <i>Desktop</i>	52
Gambar 4. 7 Tampilan Luaran pada <i>Mobile</i>	54
Gambar 4. 9 Hasil <i>Overview</i> dari <i>Input User</i>	55
Gambar 4. 10 Hasil Perhitungan Nutrisi	55
Gambar 4. 11 Program Latihan <i>Upper Body</i>	56
Gambar 4. 12 Program Latihan <i>Lower Body</i>	56

DAFTAR LAMPIRAN

Lampiran 1. Model <i>Decision Tree</i>	62
Lampiran 2. Penyederhanaan Model <i>Decision Tree</i>	63
Lampiran 3. Potongan <i>Code</i> untuk <i>Model Generation</i>	64
Lampiran 4. Potongan <i>Code</i> untuk <i>Input User</i>	65
Lampiran 5. Potongan <i>Code</i> untuk <i>Output User</i>	66

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada prinsipnya, untuk mencapai performa kesehatan terbaik seseorang, sangat penting untuk memperhatikan asupan nutrisinya (Muhlisin, 2020). Setelah selesai berolahraga, tubuh kehilangan banyak energi yang perlu digantikan dengan asupan makanan untuk mempercepat pemulihan. Makanan juga membantu mengisi kembali cadangan energi, mengatur keseimbangan cairan dan mineral dalam tubuh, serta memperbaiki kerusakan otot yang disebabkan oleh olahraga. Selain istirahat, nutrisi juga memainkan peran penting sebelum dan sesudah olahraga.

Fokus pada pemberian nutrisi berkualitas tinggi terletak pada hubungan antara asupan zat gizi dan peningkatan fungsi organ tubuh. Sebagai sumber energi, karbohidrat memainkan peran penting dalam tubuh. Hampir 40% dari total energi tubuh saat istirahat disuplai oleh karbohidrat, dengan 15-20% digunakan oleh otot (Rahman, 2021). Sehingga, pengaruh asupan nutrisi yang masuk ke dalam tubuh sangatlah penting untuk performa tubuh yang baik.

Dalam hal ini, pemenuhan nutrisi saat berolahraga tidak dapat diabaikan. Sebuah diet seimbang yang mengandung nutrisi lengkap harus menyediakan jumlah yang memadai dari energi, karbohidrat, dan protein untuk memastikan nutrisi optimal yang dibutuhkan untuk mendukung kinerja olahraga (Kostrakiewicz-Gieralt, 2021). Pengaturan nutrisi yang seimbang baik karbohidrat, protein, maupun serat bisa menjadi sangat rumit tanpa bantuan profesional.

Algoritma C4.5 memainkan peran penting dalam menghasilkan model klasifikasi (*decision tree*) yang akurat untuk memahami dan menjelaskan pengaruh intensitas latihan terhadap kebutuhan nutrisi (Bulqini, 2022). Dengan memanfaatkan algoritma ini, kita dapat mengidentifikasi pola dan hubungan antara berbagai faktor yang mempengaruhi asupan nutrisi berdasarkan intensitas latihan (Rosenbloom, 2012). Alasan memilih algoritma C4.5 adalah karena tingkat akurasinya jauh lebih tinggi dibandingkan dengan algoritma *machine learning* lainnya (Divya Mogaveera, 2021). Algoritma C4.5 mampu menangani berbagai

jenis data dan menghasilkan aturan yang dapat digunakan untuk memberikan rekomendasi yang tepat dan dipersonalisasi.

Oleh karena itu, sebuah aplikasi berbasis web dirancang untuk mengatur jumlah nutrisi yang diperlukan sesuai dengan intensitas olahraga yang dilakukan, disertai dengan program yang sekiranya cocok dengan pengguna aplikasi. Aplikasi ini akan mempermudah kegiatan olahraga dari pengguna, baik program olahraganya maupun asupan nutrisi yang diperlukan, tanpa harus mengeluarkan biaya mahal untuk mendapatkan jasa *personal trainer* dan *food nutrionist*.

1.2 Identifikasi Masalah

Berdasarkan hal – hal yang telah disampaikan dalam latar belakang, permasalahan yang dapat diidentifikasi adalah tingginya biaya *personal trainer*, sehingga kesulitan mencapai tujuan latihan karena kurangnya bimbingan dan ketidaksesuaian program latihan dengan kebutuhan individu, serta konsumsi nutrisi yang tidak teratur akibat kurangnya arahan yang berdampak pada hasil latihan yang kurang maksimal.

1.3 Batasan Masalah

Keterkaitan aplikasi pendamping olahraga ini erat dengan aktivitas olahraga lainnya. Karena itu, penelitian ini difokuskan pada:

1. Data rekomendasi latihan (bagian otot yang dilatih) yang didapatkan dari hasil wawancara dengan narasumber Juniarko Yumana yang akan digunakan pada aplikasi pendamping olahraga berdasarkan hasil dari observasi dan wawancara di *gym*.
2. Data asupan nutrisi yang didapatkan dari hasil wawancara dengan narasumber Yulius Ronny dan penelitian terdahulu oleh (Rosenbloom, 2012) yang akan digunakan pada aplikasi pendamping olahraga berdasarkan hasil dari observasi dan wawancara di *gym*.
3. Perancangan aplikasi pendamping olahraga berbasis *Progressive Web App* (PWA) berdasarkan kebutuhan *user* yang berolahraga di *gym*.

1.4 Perumusan Masalah

Berdasarkan beberapa penjelasan yang telah disampaikan pada bagian latar belakang, dirumuskan permasalahan sebagai berikut.

1. Bagaimana mengembangkan model klasifikasi dengan algoritma *C4.5* dan *decision tree* untuk menganalisis pengaruh *workout intensity* terhadap asupan nutrisi?
2. Bagaimana menyusun dan mengintegrasikan informasi asupan nutrisi yang tepat untuk berbagai jenis latihan dalam panduan komprehensif bagi pengguna aplikasi?
3. Bagaimana merancang dan mengembangkan aplikasi pendamping olahraga berbasis PWA yang mengimplementasikan model *C4.5* dan *decision tree* untuk memberikan rekomendasi latihan dan asupan nutrisi yang dipersonalisasi?

1.5 Tujuan Penelitian

Dengan mengacu pada rumusan masalah tersebut, tujuan penelitian dapat diidentifikasi sebagai berikut.

1. Membuat model klasifikasi menggunakan algoritma *C4.5* dan *decision tree* untuk menganalisis dan menjelaskan pengaruh atribut *workout intensity* terhadap asupan nutrisi.
2. Menyusun dan mengintegrasikan informasi mengenai asupan nutrisi yang tepat untuk mendukung masing-masing jenis latihan, guna menyediakan panduan yang komprehensif bagi pengguna aplikasi.
3. Merancang dan mengembangkan aplikasi pendamping olahraga berbasis *Progressive Web App* (PWA) dengan mengimplementasikan model *C4.5* dan *decision tree* dalam aplikasi untuk memberikan rekomendasi latihan dan asupan nutrisi yang dipersonalisasi berdasarkan data pengguna dan intensitas latihan.

1.6 Luaran Penelitian

Luaran yang diharapkan dari penelitian ini adalah:

1. Aplikasi pendamping olahraga untuk memberikan program *workout* dan nutrisi berbasis *web*.
2. Hasil dari model *decision tree* yang akurat sesuai dengan kebutuhan *user*.
3. Publikasi jurnal nasional terakreditasi

1.7 Manfaat Penelitian

Berikut ini adalah beberapa manfaat yang dapat diperoleh dari penelitian ini:

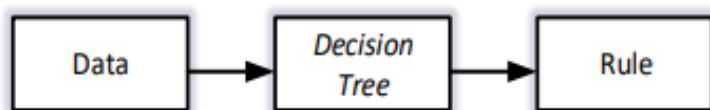
1. Manfaat Teoritis
 - a. Penelitian ini diharapkan dapat melengkapi dan memperluas wawasan, informasi, pemikiran, dan pengetahuan bagi para pengguna.
 - b. Sebagai acuan dan pertimbangan untuk penelitian selanjutnya, terutama yang terkait dengan pengembangan aplikasi pendamping olahraga.
2. Manfaat Praktis
 - a. Bagi pengguna, diharapkan menjadi suatu alat bantu untuk mendampingi olahraga di *gym* agar dapat tercapainya tujuan tanpa mengeluarkan banyak biaya, dan kontribusi yang bermanfaat untuk kemajuan ilmu pengetahuan, terutama di bidang olahraga.

BAB II

TINJAUAN PUSTAKA

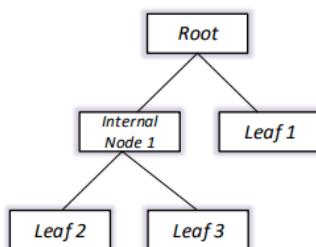
2.1 *Decision Tree*

Decision Tree merupakan algoritma *clustering* yang memiliki struktur seperti *flowchart* di mana masing-masing *node* mewakili nilai atribut dan masing-masing cabang merepresentasikan hasil pengujian, dan masing-masing daun merepresentasikan kelas atau distribusi kelas (Supangat, 2018). *Decision Tree* memiliki beberapa algoritma yang dapat digunakan, di antaranya ID3 (*Iterative Dichotomiser 3*), C4.5, C5.0, CHAID (*Chi-squared Automatic Interaction Detection*), dan MARS (*Multivariate Adaptive Regression Splines*). Seperti dapat dilihat gambar 2.1, di Alur pembuatan model *decision tree* pertama-tama mengumpulkan data, kemudian membuat model *decision tree*, dan dari model tersebut akan muncul *Rules* yang akan digunakan.



Gambar 2. 1 Konsep Alur *Decision Tree*

Decision Tree C4.5 merupakan bentuk pengembangan dari algoritma ID3 yang mengadopsi pendekatan *greedy* dengan pengambilan keputusan berdasarkan *tree* yang terbentuk menggunakan pendekatan rekursif *top down* dan sistem bagi serang (Supangat, 2018). Gambar 2.2 menjelaskan cara algoritma C4.5 untuk membangun model *decision tree* yaitu, pilih atribut yang akan digunakan sebagai akar, kemudian buatlah sebuah cabang untuk setiap nilai, lalu bagilah kasus dalam sebuah cabang, lalu ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama (Prasetio, 2021).



Gambar 2. 2 Konsep Dasar Pembentukan *Decision Tree*

2.2 Algoritma C4.5

Algoritma C4.5 memiliki beberapa tahapan untuk membuat sebuah model *decision tree* (Larose, 2006), yaitu terdiri dari beberapa tahapan berikut.

1. Mempersiapkan data *training*. Data diambil dari histori yang sudah pernah terjadi sebelumnya, dan dikelompokkan ke kelas-kelas tertentu.
2. Menghitung nilai *entropy*. Nilai ini mengindikasikan perbedaan keputusan terhadap nilai atribut tersebut. Semakin tinggi nilainya, semakin tinggi juga perbedaannya (ketidakpastian). Rumus *entropy* adalah sebagai berikut:

$$Entropy = - \sum_{i=1}^k P_i \times \log_2 P_i \quad (2-1)$$

Keterangan:

S = Himpunan Kasus

k = Jumlah Partisi S

p_i = Proporsi S_i terhadap S

3. Menghitung nilai *Gain*. Nilai *Gain* ini nanti akan menentukan seberapa pengaruhnya suatu atribut terhadap pohon keputusan tersebut, atau bisa disebut ukuran efektifitas suatu variabel dalam mengklasifikasikan data. Atribut yang memiliki nilai *Gain* paling tinggi yang akan dijadikan *node* akar pertama. Nilai *Gain* didapatkan dari rumus berikut:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^k \frac{|S_i|}{|S|} \times Entropy(S_i) \quad (2-2)$$

Keterangan:

S = Himpunan Kasus

A = Fitur

k = Jumlah Partisi Atribut A

$|S_i|$ = Proporsi S_i terhadap S

$|S|$ = Jumlah kasus dalam S

4. Mengulangi kembali langkah ke-2 dan ke-3 hingga semua terpartisi.
5. Proses akan berhenti saat semua isi dalam *node* N mendapatkan kelas yang sama dan/atau tidak ada data atau variabel yang dipartisi lagi.

2.3 Pengertian dan Manfaat Nutrisi

Dalam hal ini, perspektif tersebut memberikan keuntungan tambahan karena dapat memberikan fleksibilitas yang lebih besar bagi pelatih atau atlet dalam mempertimbangkan jenis bahan bakar yang dapat digunakan (Kerksick, 2017). Ketika tubuh sedang istirahat, sekitar 40% dari total energinya dipasok oleh karbohidrat, dengan sekitar 15-20% di antaranya digunakan oleh otot (Arimbi, 2018).

Dengan memuat karbohidrat, glikogen dalam jumlah besar akan tersimpan di otot yang selanjutnya akan meningkatkan performa fisik dan daya tahan dalam jangka panjang (Mueller, 2016). Konsumsi nutrisi selama olahraga terbagi menjadi tiga fase penting, yaitu sebelum latihan, saat latihan, dan setelah selesai latihan. Setiap fase memiliki kebutuhan nutrisi spesifik untuk mendukung performa dan pemulihan tubuh. Sebelum latihan, terutama jika durasi olahraga lebih dari 90 menit, perhitungan nutrisi dilakukan berdasarkan jarak waktu antara konsumsi dan waktu latihan. Jika jarak waktu kurang dari 1 jam, diperlukan sekitar 30-gram karbohidrat. Jika jaraknya 1 jam, dibutuhkan sekitar 0,45-gram karbohidrat per pon berat badan untuk pemulihan glikogen. Untuk jarak waktu 2 jam, diperlukan sekitar 0,9-gram karbohidrat per pon berat badan. Jika lebih dari 2 jam, cukup konsumsi makanan yang mengandung campuran karbohidrat, protein, dan lemak dalam jumlah sedang. Untuk intensitas olahraga yang tidak lebih dari 1 jam, hanya diperlukan konsumsi nutrisi sebanyak 30-gram karbohidrat. Dengan mengikuti panduan ini, kebutuhan energi dan nutrisi tubuh selama latihan dapat terpenuhi dengan optimal (Rosenbloom, 2012).

Selama latihan berlangsung, konsumsi nutrisi dibagi menjadi dua bagian. Jika intensitas latihan sedang hingga tinggi berlangsung selama 1 hingga 2 jam, diperlukan 30 hingga 60-gram karbohidrat yang mudah dicerna. Jika tidak, tidak ada makanan ekstra yang dibutuhkan selama latihan berlangsung. Setelah latihan, jika intensitas latihan sedang hingga tinggi berlangsung lebih dari 90 menit, diperlukan sekitar 0,45-gram karbohidrat per pon berat badan dan 10 hingga 20 gram protein untuk perbaikan dan sintesis protein otot. Dengan mengikuti panduan ini, kebutuhan energi dan nutrisi tubuh selama latihan dapat terpenuhi dengan

optimal, mendukung performa yang baik serta pemulihan yang cepat dan efisien (Rosenbloom, 2012).

Penekanan untuk memberikan kualitas nutrisi yang baik adalah pada interaksi antara asupan zat gizi dengan peningkatan fungsi alat-alat tubuh (Bulqini, 2022). Tujuan utamanya adalah optimalisasi kinerja dan pencegahan kelelahan (Leonarda, 2018). Untuk mendukung kinerja olahraga yang optimal, diet yang seimbang harus menyediakan asupan energi, karbohidrat, dan protein yang memadai. Ini bertujuan untuk memastikan bahwa nutrisi yang diperlukan terpenuhi dengan baik (Chowdhury, 2020).

Untuk memulihkan tubuh setelah berolahraga, strategi nutrisi yang direkomendasikan harus mencakup asupan cairan, elektrolit (seperti natrium), karbohidrat, dan protein. Hal ini penting untuk memastikan tubuh mendapatkan nutrisi yang dibutuhkan untuk memperbaiki dan mengembalikan energi serta kekuatan setelah aktivitas fisik yang intens (Meyer, 2012). Penipisan glikogen otot dan dehidrasi merupakan faktor utama yang menyebabkan kelelahan (Al-Okbi, 2014).

Asupan nutrisi yang tepat dapat sangat membantu atlet dalam meningkatkan performa mereka. Ini dilakukan melalui produksi energi yang optimal, penundaan terhadap kelelahan (*fatigue*), peningkatan kekuatan dan fokus, serta mempercepat proses pemulihan (*recovery*). Oleh karena itu, sangat penting bagi atlet untuk memperhatikan asupan nutrisi mereka agar dapat mencapai performa terbaik mereka (Zahra, 2020).

Untuk meningkatkan performa atlet, terdapat beberapa faktor penting yang perlu diperhatikan. Faktor-faktor tersebut meliputi fondasi yang kuat dalam kondisi fisik, pengalaman, program latihan yang terstruktur dan efektif, serta asupan nutrisi yang tepat dan teratur. Dengan memiliki fondasi yang kuat dalam kondisi fisik, atlet dapat membangun kekuatan dan daya tahan yang dibutuhkan dalam berbagai jenis olahraga. Pengalaman juga memainkan peran penting dalam meningkatkan keterampilan dan konsistensi dalam berkompetisi. Program latihan yang terstruktur dan efektif membantu atlet untuk memperbaiki teknik, memperkuat otot, dan meningkatkan kondisi fisik secara keseluruhan. Selain itu, asupan nutrisi yang tepat dan teratur juga sangat penting untuk memastikan tubuh mendapatkan energi dan

nutrisi yang dibutuhkan untuk mempertahankan dan meningkatkan performa atlet (Peeling, 2018).

Untuk mencapai kesehatan jasmani yang optimal, atlet harus memenuhi kebutuhan gizi yang sesuai dengan karakteristik olahraga yang dilakukannya. Asupan nutrisi yang tepat dan sesuai dengan jenis olahraga yang dilakukan oleh atlet dapat membantu meningkatkan kesehatan jasmani dan performa atlet. Oleh karena itu, penting bagi atlet untuk memahami kebutuhan nutrisi mereka dan memenuhinya dengan tepat (Sasmarianto, 2021).

Makanan yang tinggi gula dapat mengakibatkan peningkatan kadar gula darah yang cepat, yang diikuti oleh penurunan kadar gula darah dan energi yang rendah. Selain itu, makanan yang mengandung gula kental seperti permen dapat menarik cairan ke dalam saluran pencernaan dan menyebabkan dehidrasi, kram, mual, dan diare (Zanecosky, 1986).

2.4 Kekuatan Otot

Kekuatan otot mengacu pada kemampuan otot untuk menghasilkan gaya atau tenaga kontraktif yang diperlukan dalam pelaksanaan aktivitas fisik, seperti mengangkat beban berat, bergerak, atau menjalankan fungsi tubuh lainnya. Pengukuran kekuatan otot dapat dilakukan dengan beragam metode, termasuk melalui evaluasi fisik atau menggunakan peralatan khusus.

Ada beberapa faktor yang mempengaruhi kekuatan otot. Menurut (Sharkey, 2003) beberapa faktor yang dapat mempengaruhi kekuatan otot adalah:

1. Ukuran otot. Ukuran otot memiliki dampak signifikan pada kekuatan otot. Semakin besar serabut otot seseorang, semakin besar kemampuannya dalam mengangkat beban. Ukuran otot yang besar dan panjang bisa dipengaruhi oleh faktor genetik, tetapi juga bisa diperbesar melalui latihan yang memperluas serabut otot.
2. Jenis Kelamin. Latihan kekuatan lebih bermanfaat bagi pria dibandingkan wanita karena perbedaan perkembangan otot berdasarkan jenis kelamin. Pada masa pubertas, pria mulai mengalami pertumbuhan otot yang lebih besar, terutama karena tingginya hormon testosteron yang membantu otot tumbuh membesar. Sebelum usia 12-14 tahun,

kekuatan antara pria dan wanita relatif seimbang, tetapi setelah itu, pria cenderung memiliki kekuatan yang lebih besar.

3. Umur manusia. Kekuatan otot pada pria mencapai puncaknya di usia 20-an, kemudian mulai mengalami penurunan yang lambat hingga mencapai usia 60 tahun. Setelah itu, penurunannya menjadi lebih signifikan. Namun, penurunan ini dapat diperlambat dengan menjaga kekuatan otot melalui latihan teratur. Dengan latihan yang konsisten, kekuatan otot dapat dipertahankan hingga usia 60-an, bahkan beberapa atlet seperti petinju kelas berat George Foreman dapat mencapai prestasi puncak pada usia 40-an. Penelitian juga menunjukkan bahwa latihan beban pada usia 72-98 tahun dapat meningkatkan kekuatan, massa otot, dan mobilitas.

2.5 Body Mass Index (BMI)

Pengukuran Body Mass Index (BMI) merupakan nilai yang diperoleh dari perhitungan dua komponen utama, yaitu tinggi badan (TB) dan berat badan (BB) (Santi Dhara, 2015). Kedua komponen ini dapat diukur dengan akurat oleh seseorang dengan sedikit latihan. Namun, BMI memiliki keterbatasan dan membutuhkan indikator penilaian tambahan jika digunakan secara individu. BMI banyak digunakan di masyarakat karena kemudahannya. Untuk mengetahui nilai BMI, dapat dihitung dengan rumus berikut:

$$\text{Body Mass Index (BMI)} = \frac{\text{Berat Badan (Kg)}}{\text{Tinggi Badan (m)}^2} \quad (2-3)$$

2.6 Hypertext Preprocessor (PHP)

PHP merupakan salah satu bahasa pemrograman yang digunakan khusus dalam pengembangan *web*, dan dapat digunakan bersamaan dengan *script* HTML. Bahasa pemrograman PHP dapat menggambarkan beberapa bahasa pemrograman lainnya, seperti contohnya *Java* dan *C*, serta mudah untuk dipelajari dan diimplementasikan.

PHP disebut sebagai bahasa pemrograman *server-side* karena diproses pada komputer *server* bukan pada *web browser*-nya (Jannah, 2019). Artinya, pada saat menerjemahkan *script* program dilakukan di *server*, baru kemudian hasil dari terjemahan tersebut akan dikirim balik ke *client* yang melakukan *request*. PHP banyak digunakan untuk membuat banyak proyek, sebagai contohnya GUI dan *website* yang dinamis. Pada dasarnya, *server* akan melakukan proses bekerja jika menerima permintaan dari *client*. Dalam hal ini, *client* melakukan permintaan dengan menggunakan kode skrip PHP untuk mengirimkan permintaan ke *server*.

Alur kerja PHP diawali dengan permintaan dari *user* yang berasal dari *website* di *browser*. Berdasarkan URL yang tertera pada *browser*, *browser* akan mencari dan mengidentifikasi alamat dari *webserver* yang dituju, kemudian mengirim semua data yang akan dipakai oleh *webserver*. Selanjutnya, *webserver* akan mencari dan menampilkan data yang diminta di *browser*. *Browser* akan otomatis menerjemahkan skrip kode dan menampilkannya kepada *user*.

2.7 *Oracle*

Oracle merupakan salah satu aplikasi basis data yang digunakan dalam suatu sistem manajemen basis data RDBMS (*Relational Data Base Management System*) (Ali, 2014). *Oracle* memiliki satu atau beberapa jumlah tabel, yang tiap tabelnya terdiri dari sejumlah baris dan kolom. Setiap tabel menyimpan data – data yang disimpan oleh *user* ke dalam *database*.

Oracle digunakan untuk pembuatan *query* atau perintah untuk mengambil dan menampilkan data yang diambil dari tabel – tabel yang ada (Sinuraya, 2018). *Oracle* menawarkan berbagai macam produk, yang dapat dijalankan di berbagai macam dan merk perangkat komputer, sehingga dapat dengan mudah digunakan untuk berbagai kebutuhan menyimpan data.

Dengan berbagai keunggulan yang dimiliki *Oracle*, *database* ini banyak digunakan oleh praktisi untuk membangun suatu proyek. Adanya fasilitas API (*Application Programming Interface*) yang dimiliki oleh *Oracle*, memudahkan berbagai macam aplikasi pada komputer yang ditulis dengan menggunakan bahasa pemrograman yang berbeda – beda dapat menggunakan *Oracle* sebagai basis datanya.

2.8 Laravel

Laravel merupakan sebuah *framework open-source* berbasis MVC (*Model-View-Controller*) yang digunakan untuk mengembangkan dan memaksimalkan penggunaan PHP dalam pembuatan *web*. *Laravel* memiliki beberapa fitur yang mempermudah *developer*, seperti *template engine*, *routing*, serta *modularity* (Alpina, 2022).

Fokus penggunaan *Laravel* terletak pada *end-user*. Dengan kata lain, *Laravel* fokus pada kesederhanaan serta kejelasan, baik dari segi penulisan maupun tampilan, sehingga dapat menghasilkan aplikasi *web* yang dapat bekerja sesuai dengan fungsionalitasnya. Dengan adanya *Laravel*, bahasa pemrograman PHP dapat digunakan lebih efisien, aman, dan simpel berkat banyaknya fitur yang ditawarkan oleh *Laravel*. Tidak hanya membuat pengembangan *website* menjadi lebih ekspresif, *Laravel* juga mempermudah pengembangan *website*.

Fitur *blade engine template* pada *Laravel* memberikan *developer* kemudahan untuk mendesain *layout* yang unik dan struktur yang konsisten selama proses pengembangan. Kelebihan lainnya adalah tidak membatasi *developer* dalam menggunakan skrip kode PHP biasa dalam tampilan, sehingga lebih fleksibel dibandingkan *template engine* lainnya.

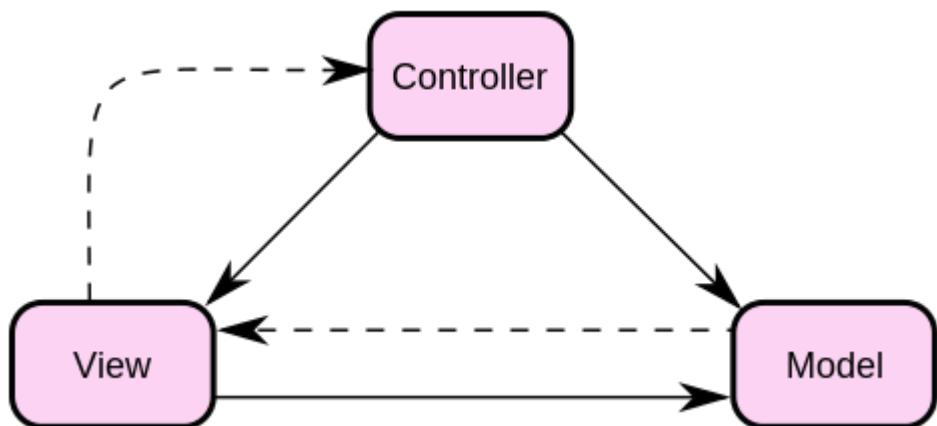
Selain itu, *Laravel* mempunyai kumpulan modul dan *library* yang dapat digunakan oleh pengembang dengan menggunakan *Composer*. Fitur *Composer* sangat membantu menyempurnakan fungsionalitas dari aplikasi *website* yang sedang dikembangkan. *Modularity* ini cocok untuk mengembangkan *website* dalam skala besar, karena mudahnya proses *update*, sehingga *error* dan *bug* yang terdapat dalam program dapat diperbaiki dengan baik.

Seperti yang disebutkan di paragraf sebelumnya, *Laravel* memiliki *tool* unggulan yang disebut *Composer*. *Composer* ini memiliki berbagai macam *dependencies* dan kumpulan *library* yang dapat digunakan oleh *developer*. Jadi *developer* cukup memanggil *Composer* untuk memasang *library* apapun, sesuai dengan kebutuhan *developer*.

2.9 Model-View-Controller (MVC)

MVC adalah suatu arsitektur dalam pengembangan aplikasi yang memiliki 3 komponen penting. Ketiga komponen itu (*Model*, *View*, dan *Controller*) memiliki fokus dan logika sendiri, sehingga dapat mempercepat kinerjanya (Harianja, 2010). MVC memisahkan komponen-komponen utama dalam pengembangan aplikasi seperti proses olah data, kontrol dari sebuah *web*, dan *user interface*-nya.

Tiap komponen memiliki fungsi masing-masing. Pada gambar 2.3, dapat terlihat relasi antara masing-masing atribut, di antaranya *Controller* berfungsi sebagai penghubung antara *model* dan *view*, yang tugasnya memproses data dan permintaan masuk yang kemudian dikirimkan ke halaman *web*, kemudian komponen *view* bertanggung jawab untuk *user interface* pada aplikasi tersebut, dan yang terakhir *model* berhubungan dengan data-data logis yang akan digunakan oleh *user*, seperti memasukkan data, menghapus data, maupun pembaruan data. Dengan menggunakan arsitektur MVC, maka aplikasi yang dibuat dapat lebih mudah dikembangkan.



Gambar 2. 3 Diagram Relasi antara *Model*, *View*, *Controller*

2.10 Hypertext Markup Language (HTML)

HTML merupakan bahasa pemrograman yang digunakan untuk membuat struktur *web* dan mempublikasikannya secara *online* (Henderson, 2009). Bahasa pemrograman ini menggunakan *tag* tertentu, untuk mendeklarasikan baris kode

mana yang harus diimplementasikan agar halaman tersebut dapat tampil dengan benar.

HTML juga menyediakan struktur dasar halaman (*template*) yang disebut CSS (*Cascading Style Sheet*), untuk dapat mengubah tampilannya. HTML sendiri memiliki beragam fungsi yang dapat digunakan. Salah satu fungsi utamanya ialah menjadi dasar dari sebuah *website*, yang nantinya bisa diimplementasikan dengan bahasa pemrograman PHP, dan didesain ulang dengan menggunakan CSS sesuai kebutuhan. HTML juga berfungsi untuk menandai bagian-bagian dari *website*, di antaranya *header*, *main*, *footer*, dan *navigation*, dengan menggunakan *tag*.

2.11 *Cascading Style Sheet (CSS)*

CSS merupakan bahasa pemrograman yang digunakan untuk desain tampilan halaman *web*, seperti *layout*, warna, dan *font*. Pembuatan CSS juga digunakan untuk membuat *adaptive web*, yang bisa membuat tampilan beradaptasi dengan berbagai ukuran *layer* (Henderson, 2009).

Penggunaan CSS dapat mempermudah pengembangan *website*. Karena apabila hanya menggunakan HTML saja, tiap halaman *web* yang berbeda akan terus menulis *tag* baru untuk elemen baru tersebut. Apabila menggunakan CSS, hanya perlu menulis skrip CSS satu kali saja dan dapat diterapkan pada semua halaman *web*. Jika ingin mengubah, hanya perlu mengubah skrip CSS saja nantinya semua halaman *web* akan mengikuti perubahannya. Oleh karena itu, CSS disebut juga sebagai bahasa *style sheet* yang digunakan secara bersamaan dengan *JavaScript*.

Karena penggunaan CSS hanya memerlukan 1 skrip yang dapat digunakan di berbagai halaman *web*, kecepatan dari *loading website* bisa meningkat karena beban pada saat memuat halaman *web* lebih kecil. CSS juga menawarkan banyak tampilan halaman *web*, sehingga *developer* akan lebih mudah mengembangkan *website* dengan tampilan yang rapi dan adaptif.

2.12 *JavaScript*

Berbeda dengan PHP, *JavaScript* merupakan bahasa pemrograman yang bersifat *Client-Side Programming*, yang prosesnya bukan dilakukan di komputer

server, melainkan diproses langsung di *client*. *JavaScript* dirancang untuk mengatur bagaimana halaman *web* berinteraksi dengan *user*-nya [Henderson, 2009]. *JavaScript* juga membuat halaman *web* lebih interaktif saat digunakan oleh *user*.

JavaScript sendiri biasanya digunakan dengan HTML dan CSS untuk menciptakan halaman *web* yang dinamis dan interaktif bagi pengguna. Contohnya, ketika pengguna mengakses dan melakukan suatu permintaan pada *web*, pembaruan data dapat terjadi tanpa proses *reload*, sehingga dapat memberikan *user experience* yang sangat baik terhadap pengguna.

2.13 Bootstrap

Bootstrap merupakan sebuah *framework* untuk CSS, yang berisi banyak *plugins* dan ikon yang dapat dipakai untuk mempermudah *developer* mengembangkan halaman *web* (SPURLOCK, 2013). Dalam *bootstrap* terdapat banyak *template* yang dapat digunakan, seperti tombol, tipografi, *form*, *navbar*, dan komponen *interface* lainnya. *Bootstrap* dirancang untuk mempercepat pengembangan *web* dengan kemudahan yang ditawarkannya secara gratis dan berisfat *open source*.

Bootstrap juga memiliki banyak *template styling* yang lebih kaya dibandingkan dengan CSS dan lebih mudah dimodifikasi. Sehingga, *developer* tidak perlu membuat dari awal, tapi *Bootstrap* sudah mempunyai sistem *grid* yang siap pakai dan mudah digunakan.

2.14 jQuery

jQuery adalah sebuah *add-on library* dari *JavaScript* yang digunakan dalam interaksi antara *JavaScript* dan HTML. *jQuery* sendiri merupakan kode dari *JavaScript* yang telah ada dan tinggal menambahkan beberapa baris kode untuk memanggilnya (Beighley, 2010).

2.15 Metode *Waterfall Model*

Metode *Waterfall* merupakan metode pengembangan perangkat lunak yang bersifat sistematis dan sekuensial (Pressman, 2002). Metode *Waterfall* (Gambar 2.4) memiliki langkah – langkah sebagai berikut (Sommerville, 2011).

1. *Requirements Definition*

Pengerjaan sistem, masalah, dan tujuan akan ditetapkan dari hasil diskusi dengan *user* yang dijelaskan secara rinci dan fungsionalitas spesifikasi sistem.

2. *System and Software Design*

Tahapan ini bertujuan untuk merancang arsitektur sistem secara keseluruhan untuk alokasi kebutuhan sistem, baik perangkat lunak maupun perangkat keras.

3. *Implementation and Unit Testing*

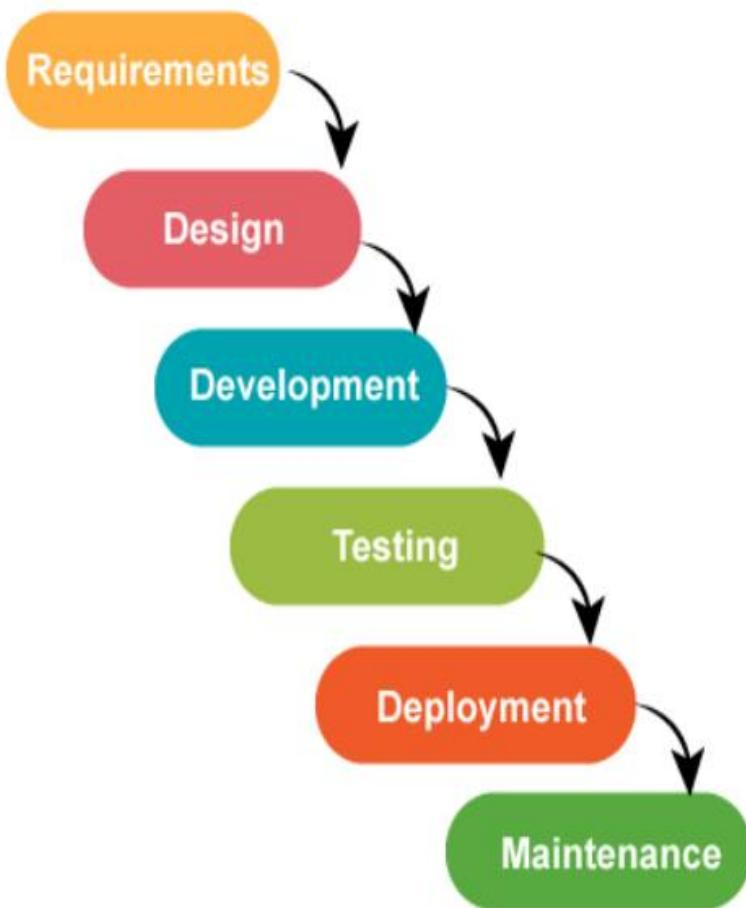
Pada tahap ini, rancangan perangkat lunak diimplementasikan dengan membuat serangkaian program. Setelah rangkaian program selesai, akan dicek kembali bahwa apakah tiap program sudah memenuhi spesifikasi permintaan.

4. *Integration and System Testing*

Rangkaian program tersebut digabung menjadi satu program utuh, kemudian diuji sebagai sebuah sistem lengkap apakah sudah memenuhi kebutuhan dan permintaan atau belum. Setelah pengujian selesai dan dinyatakan sesuai, perangkat lunak dikirimkan ke *user*.

5. *Operation and Maintenance*

Sistem dipasang dan digunakan langsung di lapangan. *Maintenance* yang dilakukan adalah untuk memperbaiki kesalahan – kesalahan yang sebelumnya tidak ditemukan, untuk meningkatkan performa dari implementasi unit sistem, dan menambahkan layanan sistem sebagai kebutuhan baru berdasarkan permintaan.



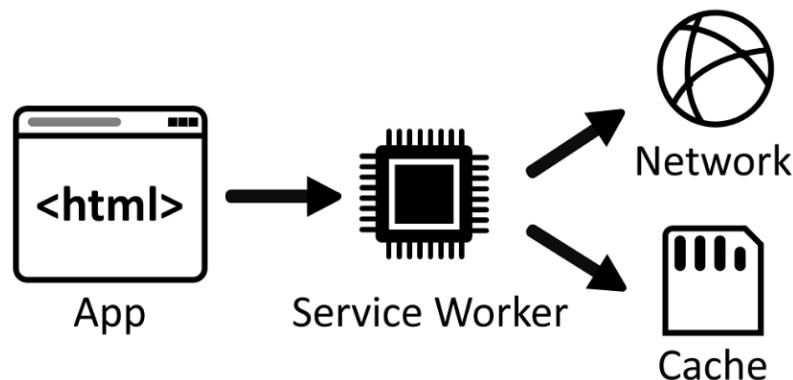
Gambar 2. 4 Metode *Waterfall*

2.16 *Progressive Web App (PWA)*

Progressive Web App adalah jenis situs *web* yang dikembangkan memiliki kemampuan untuk menyajikan konten seperti halaman *web* biasa, namun juga mampu memberikan fitur kepada pengguna seperti tetap dapat bekerja tanpa koneksi internet, memberikan notifikasi, dan mengakses perangkat keras perangkat *mobile*. Singkatnya, teknologi PWA adalah situs *mobile* yang dapat diakses seperti aplikasi *mobile*. Ketika pengguna mengunjungi situs *web*, mereka akan menerima pemberitahuan "Tambahkan ke Layar Beranda" atau "Add to Your Home Screen" yang menawarkan fitur kepada pengguna bahwa ada opsi untuk membuat akses cepat ke aplikasi di layar beranda *smartphone* pengguna (Dewi, 2020).

2.17 Service Worker

Service worker, pada dasarnya, merupakan sebuah berkas *JavaScript* yang berjalan di *thread* terpisah dari *thread* utama *browser*. Fungsinya adalah untuk mengelola permintaan jaringan, penyimpanan *cache*, mengambil *resource* dari *cache*, serta mampu mengirimkan pesan (*push notification*) (Adi, 2018). Pada gambar 2.5, dapat terlihat bahwa fungsi dari *service worker* untuk menjembatani *request* dari *user* yang terhubung langsung ke internet, dan juga dapat mengambil serta menyimpan *cache*.



Gambar 2. 5 *Service Worker*

2.18 Scikit-learn 1.5.1

Scikit-learn adalah salah satu *library machine learning* yang paling populer di *Python*. Versi terbaru, *Scikit-learn 1.5.1*, memiliki berbagai algoritma dan alat yang memudahkan implementasi dan eksperimen dalam penggunaan *machine learning*. *Library* ini sangat komprehensif, mudah digunakan, dan memiliki dokumentasi yang luas, membuatnya menjadi pilihan utama bagi banyak praktisi dan peneliti (Pedregosa, 2011).

Salah satu algoritma yang diimplementasikan dalam *Scikit-learn* adalah *C4.5*. Algoritma *C4.5* membangun model *decision tree* dengan menggunakan fitur dan *threshold* yang memaksimalkan *Gain Ratio* pada setiap *node*. Proses ini dilakukan secara rekursif hingga tercapai kondisi penghentian tertentu, seperti kedalaman maksimal pohon atau jumlah minimal sampel pada sebuah *node*.

Scikit-learn juga dikenal karena kemampuannya dalam menangani variasi data yang besar dan kompleks. *Library* ini memiliki berbagai alat dan teknik untuk menangani data skala besar, termasuk kemampuan untuk melakukan pemrosesan data secara paralel dan normalisasi data, serta dukungan untuk integrasi dengan alat *big data* lainnya seperti *Dask* dan *joblib*. Hal ini memungkinkan *Scikit-learn* untuk digunakan pada *dataset* yang sangat besar, memberikan efisiensi dan skalabilitas yang tinggi.

Namun, algoritma *C4.5* yang digunakan dalam *Scikit-learn* masih memiliki keterbatasan, yaitu belum mendukung data kategorikal secara langsung. Oleh karena itu, data kategorikal perlu diubah menjadi data numerikal terlebih dahulu sebelum dapat digunakan oleh algoritma ini melalui *preprocessing*. Meskipun demikian, versi optimasi dari algoritma *C4.5* yang diimplementasikan dalam *Scikit-learn* tetap menawarkan performa yang efisien dan efektif untuk berbagai aplikasi *machine learning* (Pedregosa, 2011).

2.19 *Confusion Matrix*

Confusion matrix adalah alat yang sangat penting dalam evaluasi kinerja model klasifikasi. Matriks ini membantu dalam menilai performa model dengan memberikan gambaran yang jelas tentang prediksi yang benar dan salah yang dibuat oleh model. Matriks ini berbentuk tabel yang menunjukkan perbandingan antara label sebenarnya dan prediksi yang dilakukan oleh model (B. Firmanto, 2016).

Dalam sebuah *confusion matrix*, terdapat empat elemen utama:

1. *True Positive* (TP): Jumlah kasus positif yang benar-benar diprediksi sebagai positif oleh model.
2. *True Negative* (TN): Jumlah kasus negatif yang benar-benar diprediksi sebagai negatif oleh model.
3. *False Positive* (FP): Jumlah kasus negatif yang salah diprediksi sebagai positif oleh model (juga dikenal sebagai tipe I error).
4. *False Negative* (FN): Jumlah kasus positif yang salah diprediksi sebagai negatif oleh model (juga dikenal sebagai tipe II error).

Dengan menggunakan *confusion matrix*, beberapa metrik evaluasi penting dapat dihitung yang terdiri dari:

1. *Precision*: Proporsi prediksi positif yang benar dari total prediksi positif

$$\frac{TP}{TP + FP} \quad (2-4)$$

2. *Recall (Sensitivity)*: Proporsi kasus positif yang benar-benar diidentifikasi dari total kasus positif sebenarnya

$$\frac{TP}{TP + FN} \quad (2-5)$$

3. *Accuracy*: Proporsi prediksi yang benar dari total prediksi

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (2-6)$$

Nilai-nilai dalam *confusion matrix* biasanya ditunjukkan dalam satuan persen (%) untuk memudahkan interpretasi. Misalnya, jika sebuah model memiliki akurasi sebesar 90%, ini berarti 90% dari seluruh prediksi yang dibuat oleh model adalah benar (B. Firmanto, 2016).

Confusion matrix juga sangat bermanfaat dalam memahami jenis kesalahan yang dilakukan oleh model dan memberikan panduan untuk perbaikan model. Misalnya, jika model memiliki banyak *False Positives*, ini bisa menunjukkan bahwa model terlalu agresif dalam mengklasifikasikan kasus sebagai positif. Sebaliknya, banyaknya *False Negatives* bisa menunjukkan bahwa model terlalu konservatif.

2.20 Penelitian Terdahulu

Penelitian mengenai kebutuhan gizi untuk atlet atau olahragawan telah dilakukan oleh (Bulqini, 2022) dalam penelitiannya yang berjudul “Nutrisi untuk Peak Performance bagi Atlet Profesional Sebelum, Selama dan Setelah Kompetisi”, dan penelitian yang dilakukan oleh (Rosenbloom, 2012) yang berjudul “Food and Fluid Guidelines Before, During, and After Exercise”.

Adapun persamaan penelitian ini dengan penelitian yang terdahulu adalah:

1. Klasifikasi asupan nutrisi berdasarkan aspek berat badan
2. Klasifikasi asupan nutrisi berdasarkan intensitas latihan

3. Klasifikasi asupan nutrisi berdasarkan lama *pre-workout* dan *post-workout*
4. Klasifikasi menggunakan algoritma *C4.5* dan model *decision tree*

Sedangkan, perbedaan dari penelitian ini dengan penelitian terdahulu adalah:

1. Batasan klasifikasi asupan nutrisi hanya sebatas berat badan pada penelitian terdahulu, sedangkan dalam penelitian ini menggunakan batasan *BMI (Body Mass Index)* yang akan menghasilkan luaran yang lebih akurat.
2. Penambahan batasan klasifikasi asupan nutrisi dengan jenis kelamin, agar dapat lebih akurat menyesuaikan kebutuhan *user*.
3. Penambahan batasan klasifikasi berupa bagian tubuh mana yang akan dilatih (*program workout*).
4. Luaran yang ada di penelitian dahulu hanya berupa asupan nutrisi yang diperlukan saja, dan hanya berpusat pada berat badan. Namun dalam penelitian ini, luaran yang dihasilkan dapat lebih banyak, yaitu program latihan, nutrisi yang diperlukan, dan rekomendasi makanan, berdasarkan beberapa aspek yang sudah disebutkan di atas.

Penelitian mengenai kebutuhan gizi juga telah dilakukan oleh (Azmi, 2022) dengan penelitiannya yang berjudul “Penerapan Expert System Pada Anak Balita untuk Mendeteksi Obesitas Menggunakan Metode Case Based Reasoning”.

Adapun persamaan dengan penelitian ini adalah klasifikasi asupan nutrisi berdasarkan faktor berat badan. Sedangkan perbedaan dengan penelitian ini adalah pada penelitian terdahulu menggunakan *Case Based Reasoning* (CBR), sedangkan penelitian ini menggunakan algoritma *Decision Tree C4.5*. Dikarenakan adanya atribut-atribut kelas berbeda, yang kurang efektif apabila menggunakan CBR, dan dapat mempengaruhi akurasi dari hasil luaran.

Penelitian lain yang sangat relevan dengan penelitian ini adalah penelitian yang dilakukan oleh (Divya Mogaveera, 2021) yang berjudul “e-Health Monitoring System with Diet and Fitness Recommendation using Machine Learning”. Penelitian ini mengembangkan sistem pemantauan kesehatan elektronik yang

memberikan rekomendasi diet dan kebugaran menggunakan teknologi *machine learning*. Adapun persamaan dengan penelitian ini dengan penelitian yang dilakukan (Divya Mogaveera, 2021) adalah:

1. Penggunaan algoritma *machine learning* (C4.5) untuk rekomendasi yang dipersonalisasi berdasarkan pengguna.
2. Bertujuan untuk membantu pengguna dalam mengelola kegiatan kebugaran dan kesehatan dengan lebih efektif.

Sedangkan perbedaannya dari penelitian ini adalah:

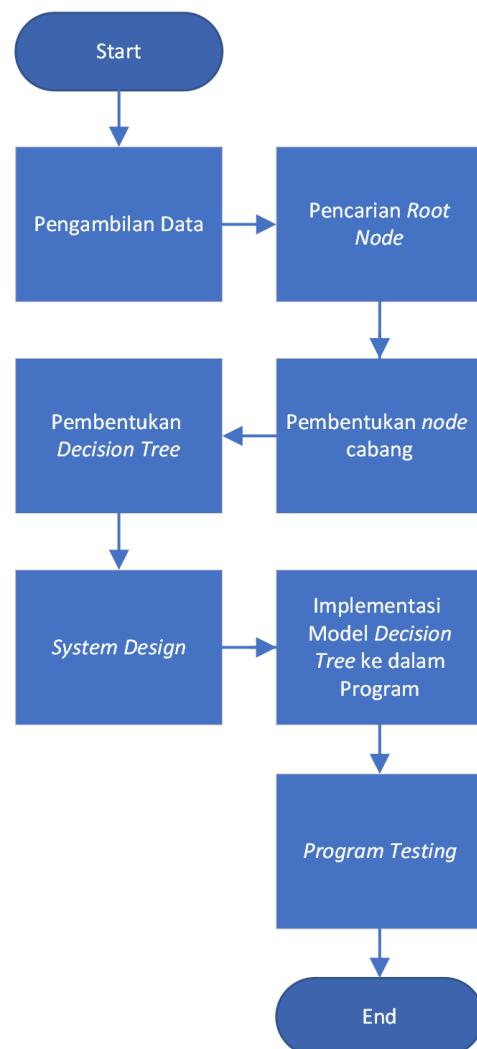
1. Penelitian ini lebih spesifik dan berfokus kepada pengguna yang berolahraga di *gym*, dengan penambahan batasan klasifikasi seperti bagian tubuh yang dilatih dan jenis kelamin.
2. Luaran penelitian ini mencakup program latihan, asupan nutrisi, dan rekomendasi nutrisi yang lebih terperinci dan disesuaikan dengan kebutuhan individu berdasarkan aspek-aspek yang sudah disebutkan di atas.

BAB III

RANCANGAN SISTEM

3.1 Tahapan Penelitian

Penelitian ini bertujuan untuk mengembangkan suatu prototipe berupa sebuah aplikasi berbasis *web*, yang dibuat dengan menggunakan *framework Laravel* untuk menghasilkan aplikasi yang memberikan panduan kebutuhan nutrisi dan program latihan yang disesuaikan dengan *user*, berdasarkan hasil dari model yang dibuat menggunakan algoritma *decision tree C4.5*. Adapun susunan tahapan penelitian dapat dilihat pada gambar 3.1.



Gambar 3. 1 Tahapan Penelitian

3.2 Studi Pustaka

Penelitian mengenai penggunaan algoritma *decision tree*, khususnya algoritma *C4.5*, dalam bidang kebugaran dan *gym* telah menunjukkan bahwa sangat efektif dalam memberikan rekomendasi yang akurat dan dapat dipersonalisasi. Berikut adalah ringkasan dari beberapa studi yang relevan dengan penelitian ini.

1. Algoritma *decision tree* (*C4.5*), telah banyak digunakan untuk mengembangkan sistem rekomendasi latihan di *gym*. Penelitian oleh (Divya Mogaveera, 2021) dalam “e-Health Monitoring System with Diet and Fitness Recommendation using Machine Learning” menunjukkan bahwa *decision tree* dapat digunakan untuk mempersonalisasi program latihan berdasarkan data pengguna seperti usia, berat badan, tingkat kebugaran, dan tujuan dari latihan tersebut. Algoritma ini membantu dalam mengklasifikasikan pengguna ke dalam kategori yang tepat dan memberikan rekomendasi latihan yang sesuai dengan personalisasi pengguna.
2. Konsumsi nutrisi selama olahraga terbagi menjadi tiga fase penting: sebelum latihan, saat latihan, dan setelah selesai latihan. Setiap fase memiliki kebutuhan nutrisi spesifik untuk mendukung performa dan pemulihan tubuh. Sebelum latihan, terutama jika durasi olahraga lebih dari 90 menit, perhitungan nutrisi dilakukan berdasarkan jarak waktu antara konsumsi dan waktu latihan. Jika jarak waktu kurang dari 1 jam, diperlukan sekitar 30-gram karbohidrat. Jika jaraknya 1 jam, dibutuhkan sekitar 0,45-gram karbohidrat per pon berat badan untuk pemulihan glikogen. Untuk jarak waktu 2 jam, diperlukan sekitar 0,9-gram karbohidrat per pon berat badan. Jika lebih dari 2 jam, cukup konsumsi makanan yang mengandung campuran karbohidrat, protein, dan lemak dalam jumlah sedang. Untuk intensitas olahraga yang tidak lebih dari 1 jam, hanya diperlukan konsumsi nutrisi sebanyak 30-gram karbohidrat. Selama latihan berlangsung, konsumsi nutrisi dibagi menjadi dua bagian. Jika intensitas latihan sedang hingga tinggi berlangsung selama 1 hingga 2 jam, diperlukan 30 hingga 60-gram karbohidrat yang mudah dicerna. Jika tidak, tidak ada makanan ekstra yang dibutuhkan selama latihan

berlangsung. Setelah latihan, terutama jika intensitas latihan sedang hingga tinggi berlangsung lebih dari 90 menit, diperlukan sekitar 0,45-gram karbohidrat per pon berat badan dan 10 hingga 20-gram protein untuk perbaikan dan sintesis protein otot. Dengan mengikuti panduan ini, kebutuhan energi dan nutrisi tubuh selama latihan dapat terpenuhi dengan optimal, mendukung performa yang baik serta pemulihan yang cepat dan efisien. Selain rekomendasi latihan, algoritma *C4.5* juga digunakan untuk mengklasifikasikan kebutuhan asupan nutrisi pengguna *gym*. Penelitian oleh (Bulqini, 2022) dan (Rosenbloom, 2012) telah menunjukkan pentingnya asupan nutrisi sesuai dengan intensitas dan jenis latihan yang dilakukan. Dengan menggunakan *decision tree*, sistem dapat mengkategorikan pengguna berdasarkan BMI, jenis kelamin, dan tujuan kebugaran, lalu memberikan rekomendasi nutrisi yang sesuai dengan kebutuhan masing-masing pengguna.

3. Penggunaan algoritma *C4.5* juga ditemukan dalam pengembangan aplikasi kebugaran berbasis *Progressive Web App* (PWA). (Azmi, 2022) dalam penelitiannya mengenai “Penerapan Expert System Pada Anak Balita untuk Mendeteksi Obesitas Menggunakan Metode Case Based Reasoning” menunjukkan bahwa algoritma *decision tree* (*C.45*) lebih efektif dibandingkan metode *CBR* untuk klasifikasi yang memerlukan banyak atribut. Dalam konteks aplikasi *gym*, *C4.5* digunakan untuk mempersonalisasi rekomendasi tidak hanya berdasarkan berat badan, tetapi juga intensitas latihan, jenis kelamin, dan bagian tubuh yang dilatih, sehingga dapat menghasilkan aplikasi yang lebih responsif dan relevan bagi pengguna.

Tabel 3. 1 Rumus Perhitungan Nutrisi Sebelum Workout

Intensitas Latihan	Jeda Makan Sebelum Workout	Nutrisi (/pon Berat Badan)
Tinggi	< 1 jam	30-gram karbohidrat
	1 jam	0.45-gram karbohidrat/pon berat badan
	2 jam	0.9-gram karbohidrat/pon berat badan
	3-4 jam	Konsumsi bebas
Rendah	Tidak Dipert	30-gram karbohidrat

Tabel 3. 2 Rumus Perhitungan Nutrisi Sedang Workout

Intensitas Latihan	Nutrisi
Tinggi	30 – 60-gram karbohidrat
Rendah	-

Tabel 3. 3 Rumus Perhitungan Nutrisi Setelah Workout

Intensitas Latihan	Nutrisi	
	Protein	Karbohidrat
Tinggi	20 gram/pon	0.45 gram/pon
Rendah	-	-

3.3 Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan dengan wawancara terhadap *nutritionist expert* dan *personal trainer*. Data ini nantinya akan dihitung menggunakan algoritma *C4.5* untuk menghasilkan sebuah model *decision tree*. Dari

model tersebut, didapatkan *Rules* yang akan diimplementasikan ke dalam aplikasi. Data yang dihimpun dari *nutritionist expert* dan *personal trainer* akan digabungkan dan diperkaya dengan beberapa data dari penelitian terdahulu. Adapun data yang digunakan sebagai berikut:

1. *BMI (Body Mass Index):*
 - a. Tinggi Badan (m)
 - b. Berat Badan (kg): *Input* berat badan dalam satuan kilogram, yang kemudian diubah ke satuan pon dengan rumus:
$$\text{Berat Badan(pon)} = \text{Berat Badan(kg)} \times 2.2046 \quad (3-1)$$
2. Jenis Kelamin (Perempuan / Laki-laki)
3. Umur
4. Intensitas Latihan (Tinggi / Rendah)
5. Durasi *pre-workout* (Jam)
6. Durasi Latihan (Jam)
7. Bagian Tubuh yang Dilatih (*Upper Body / Lower Body*)

Pengumpulan data yang dilakukan terbagi menjadi tiga, menyesuaikan kebutuhan dari aplikasi yang dibuat:

1. Data untuk *training* model
 - a. Data yang digunakan untuk *training* model sebanyak 80% dari 100 sampel data (Kaggle, 2021).
2. Data untuk *testing* model
 - a. Data yang digunakan untuk *testing* model sebanyak 20% dari 100 sampel data yang ada (Kaggle, 2021).
3. Data untuk *testing* aplikasi
 - a. Data yang digunakan adalah data *input* langsung dari *user*

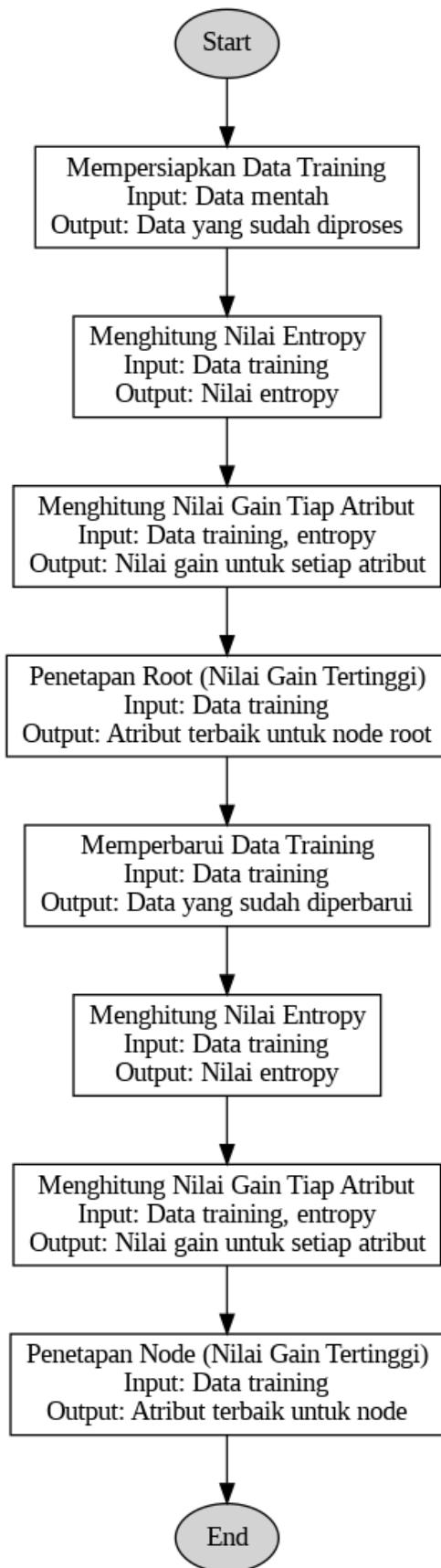
Tabel 3. 4 Contoh Data untuk *Training* dan *Testing* Model

Weight	Height	BMI	F/M	Age	WorkoutDuration	WorkoutI
92.08	1.76	29.71	F	59	45	High
61.08	1.59	23.99	F	25	30	Low
82.45	1.81	24.98	F	50	60	High
101.7	1.79	31.72	F	62	40	Low
99.60	1.96	25.67	M	57	50	High
99.04	1.85	28.85	M	46	35	Low

3.4 Penentuan *Root* dan *Node*

1. Pencarian *Root*

Proses dilakukan dengan pertama menghitung nilai *entropy* dari tiap kategori. Selanjutnya setelah didapatkan nilai *entropy*, dilakukan perhitungan nilai *gain*. Nilai *gain* yang paling tinggi dari masing-masing kategori akan dijadikan sebagai *root*. Nilai *gain* dan *entropy* didapatkan dari rumus yang sudah dijelaskan pada Bab 2. Pada gambar 3.2, dapat terlihat proses dimulai dengan menghitung nilai *entropy* dari *data training*. *Entropy* digunakan untuk mengukur tingkat ketidakpastian dalam *data training*. Setelah nilai *entropy* dihitung, langkah selanjutnya adalah menghitung nilai *gain* untuk setiap atribut dalam *data training*. *Gain* menunjukkan seberapa banyak informasi yang diperoleh dengan membagi data berdasarkan atribut tertentu. Setelah nilai *gain* dihitung untuk setiap atribut, atribut dengan nilai *gain* tertinggi dipilih sebagai *root* dari *decision tree*. *Root* ini merupakan atribut yang memberikan pemisahan terbaik dalam hal pengurangan ketidakpastian dalam data. Atribut dengan nilai *gain* tertinggi kemudian dipilih sebagai *node* berikutnya dalam pohon keputusan, yang akan menjadi cabang berikutnya dari *root* atau *node* sebelumnya.



Gambar 3. 2 Flowchart Proses Pencarian Root

2. Pembentukan *Node*

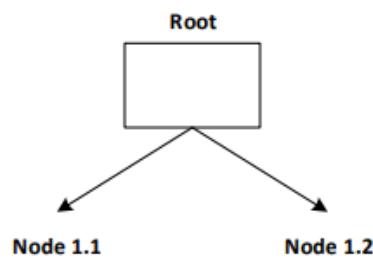
Root yang telah didapatkan akan dijadikan dasar pembentukan *node* atau cabang. Kategori dengan nilai *gain* yang tertinggi yang akan dijadikan *node* selanjutnya.



Gambar 3. 3 *Flowchart Proses Perhitungan Information Gain*

3.5 Pembuatan Model *Decision Tree* dan *Rules*

Setelah *root* ditentukan, langkah selanjutnya melibatkan perhitungan *entropy* dan *gain* untuk menentukan simpul berikutnya. Penentuan simpul berikutnya dilakukan dengan menghitung nilai *entropy* dan *gain* untuk semua atribut berdasarkan *root* yang telah ditentukan sebelumnya. Proses ini akan berjalan secara rekursif untuk menemukan simpul berikutnya dan akan berhenti jika atribut sudah termasuk dalam kelas yang sama atau jika *entropy* kelas mencapai nol. Dalam kondisi tersebut, simpul daun akan terbentuk. Hasil dari proses ini akan membentuk sebuah model *decision tree* dan *rules* yang akan diimplementasikan ke dalam aplikasi.



Gambar 3. 4 Pembentukan *Node*

3.6 Training Model

Setelah data siap, langkah berikutnya adalah melatih model menggunakan algoritma *C4.5* dan *decision tree*. Langkah-langkahnya adalah sebagai berikut.

1. *Preprocessing Data*: Mengubah data kategorikal menjadi data numerik, karena algoritma *C4.5* memerlukan *input* numerik.
2. Pemilihan Fitur: Memilih fitur yang relevan yang akan digunakan dalam model, yaitu intensitas latihan.
3. Pembagian Data: Membagi data menjadi set pelatihan dan set pengujian, dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian.

3.7 System Design

Aplikasi yang dikerjakan menggunakan metode perancangan perangkat lunak *Waterfall*.

3.8.1 Requirements

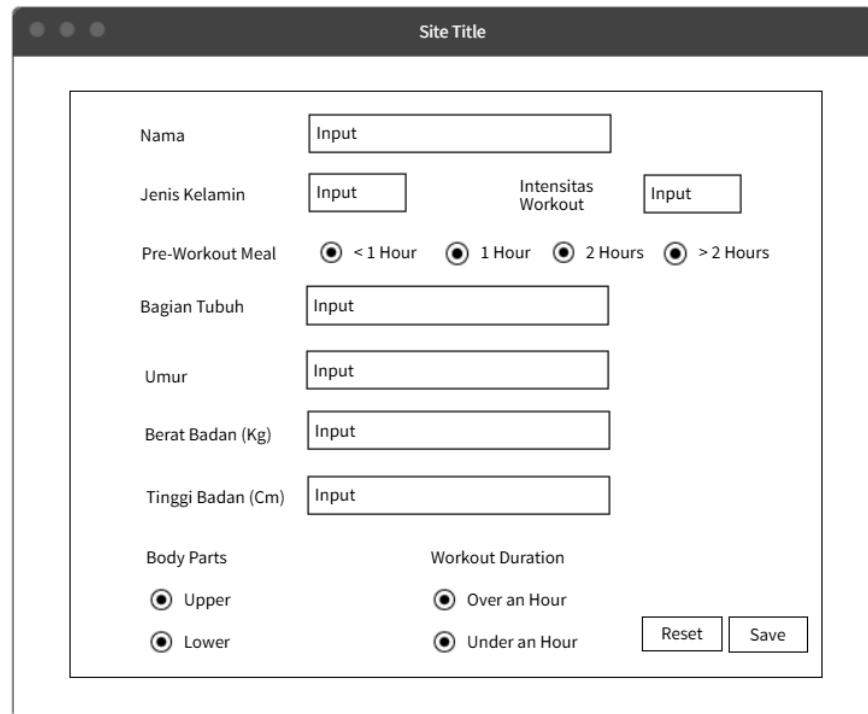
Aplikasi yang akan dibuat menerima inputan sebagai berikut:

1. Jenis kelamin (Perempuan / laki-laki)
2. Umur (Tahun)
3. *BMI (Body Mass Index)*, *BMI* akan memerlukan tinggi badan (cm) dan berat badan (kg)
4. Bagian tubuh yang dilatih (*upper body* dan *lower body*)
5. Intensitas latihan (*High* dan *Low*)
6. Durasi latihan
8. Durasi *pre-workout*

Dengan menghasilkan luaran sebagai berikut:

1. Asupan nutrisi yang diperlukan (dalam satuan gram)
2. Program latihan
3. Rekomendasi makanan

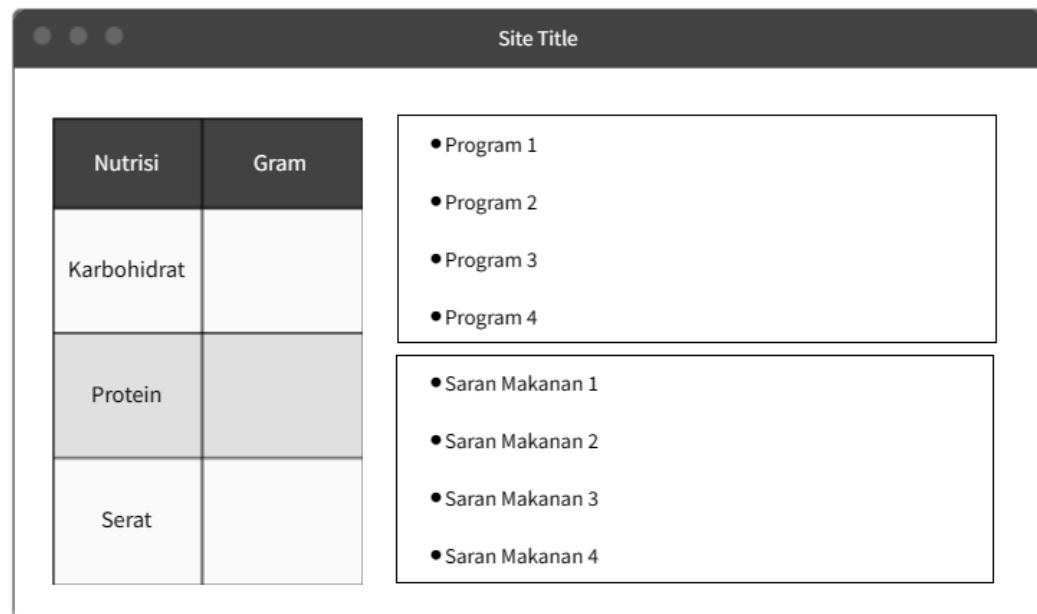
3.8.2 System Design



The image shows a user interface for inputting personal information. The form is titled 'Site Title' at the top. It contains the following fields:

- Nama: Input field
- Jenis Kelamin: Input field
- Intensitas Workout: Input field
- Pre-Workout Meal: Radio buttons for <1 Hour, 1 Hour, 2 Hours, and >2 Hours. The '1 Hour' option is selected.
- Bagian Tubuh: Input field
- Umur: Input field
- Berat Badan (Kg): Input field
- Tinggi Badan (Cm): Input field
- Body Parts: Radio buttons for Upper and Lower. The 'Upper' option is selected.
- Workout Duration: Radio buttons for Over an Hour and Under an Hour. The 'Under an Hour' option is selected.
- Buttons: 'Reset' and 'Save'

Gambar 3. 5 Design UI Input



The image shows a user interface for outputting nutritional information. The form is titled 'Site Title' at the top. It features a table on the left and two lists on the right:

Nutrisi	Gram
Karbohidrat	
Protein	
Serat	

- Program 1
- Program 2
- Program 3
- Program 4

- Saran Makanan 1
- Saran Makanan 2
- Saran Makanan 3
- Saran Makanan 4

Gambar 3. 6 Design UI Output

3.8.3 Implementation and Development

Dari *design* yang telah dibuat, maka diimplementasikan ke program dengan alur sebagai berikut.

1. *User* memasukkan data jenis kelamin, bagian tubuh yang akan dilatih, umur, berat badan, dan tinggi badan.
2. *User* menekan tombol *save* untuk menyimpan dan memproses data masukkan, atau dapat menekan tombol *reset* untuk mengisi ulang.
3. Tampilan *web* akan pindah halaman, dengan menampilkan hasil proses dari inputan *user* yang sebelumnya, lengkap dengan asupan nutrisi yang diperlukan, program latihan, dan saran makanan untuk sumber nutrisi.

3.8.4 Testing

Setelah pembuatan dan pengembangan program dan aplikasi selesai, berikutnya masuk ke tahap *testing*. Simulasi aplikasi dilakukan dengan tahapan – tahapan berikut:

1. Melakukan *input* data *training* (Kaggle, 2021)
2. Memeriksa keakuratan model *decision tree* yang digunakan dalam *logic* program dengan menggunakan *confusion matrix*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Perhitungan Nilai *Entropy* dan *Gain*

1. Perhitungan Nilai Entropy

Proses dilakukan dengan pertama menghitung nilai *entropy* dari tiap kategori. Perhitungan menggunakan data *training* (Kaggle, 2021). Perhitungan nilai *entropy* dilakukan dengan menggunakan sampel data sebanyak 93 sampel dari *dataset*. Variabel / atribut yang dihitung nilai *entropy* adalah umur, jenis kelamin, *BMI Index*, durasi latihan, dan intensitas latihan. Berikut hasil dari perhitungan nilai *entropy* masing-masing atribut:

1. Nilai *entropy* umur: 5.214873698247048
2. Nilai *entropy* jenis kelamin: 0.9286751810936653
3. Nilai *entropy BMI Index*: 6.539158811108031
4. Nilai *entropy* durasi latihan: 2.7517278935135105
5. Nilai *entropy* intensitas latihan: 0.9999165959739288

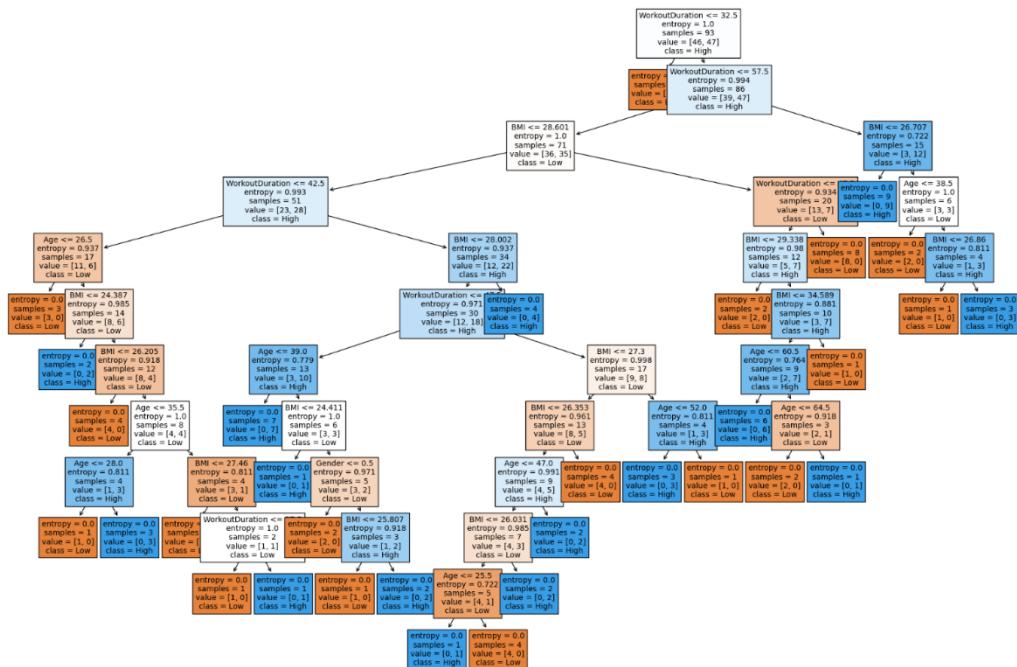
2. Perhitungan Nilai *Gain*

Proses berikutnya adalah menghitung nilai *gain* dari nilai *entropy* yang didapatkan sebelumnya. Nilai *gain* ini nanti akan menentukan seberapa pengaruhnya atribut terhadap model *decision tree*. Atribut dengan nilai *gain* paling tinggi akan menjadi *node* akar pertama. Berikut hasil dari perhitungan nilai *gain* dari masing-masing atribut:

1. Nilai *gain* umur: 0.45024546179367353
2. Nilai *gain* jenis kelamin: 0.0010139415039952526
3. Nilai *gain BMI Index*: 0.9999165959739288
4. Nilai *gain* durasi latihan: 0.20591976560136416

4.2 Pembuatan Model *Decision Tree*

Setelah penghitungan nilai *entropy* dan nilai *gain*, proses selanjutnya mengimplementasikan nilai-nilai tersebut ke dalam model *decision tree*. Meskipun *BMI Index* mungkin memiliki *gain* informasi yang lebih tinggi secara keseluruhan saat mempertimbangkan keseluruhan pohon, durasi latihan kemungkinan besar memberikan pembagian awal yang terbaik berdasarkan pengurangan *impurity* atau *gain* informasi pada tingkat *node root*. Algoritma *decision tree*, seperti CART (*Classification and Regression Trees*), biasanya memilih fitur yang memberikan *gain* tertinggi pada setiap langkah, dimulai dari *root* dan bergerak ke bawah. Proses ini memastikan bahwa setiap pembagian optimal berdasarkan keadaan pohon saat ini, meskipun fitur lain mungkin tampak memiliki *gain* yang lebih tinggi secara keseluruhan ketika mempertimbangkan beberapa pembagian. Berikut hasil implementasi nilai *gain* dalam model *decision tree*:



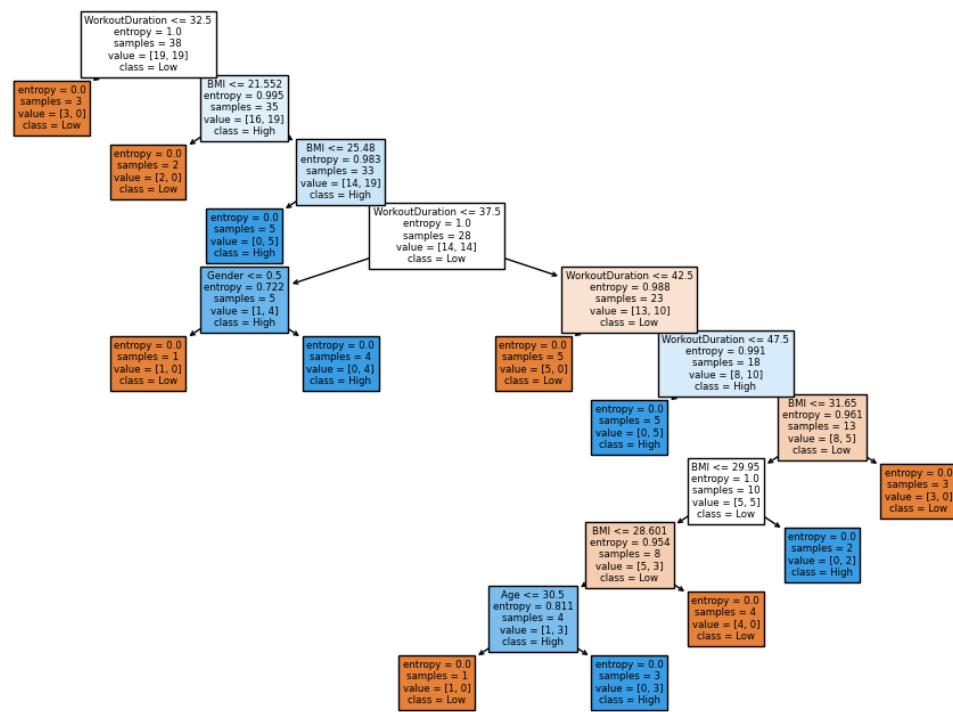
Gambar 4. 1 Hasil Model *Decision Tree*

1. *Root Node*
 - a. *Feature: WorkoutDuration <= 42.5*
 - b. *Entropy: 0.993*
 - c. *Samples: 51*
2. *First Level Nodes*
 - a. *Left Node (WorkoutDuration <= 42.5)*
 - i. *Feature: Age <= 26.5*
 - ii. *Entropy: 0.937*
 - iii. *Samples: 17*
 - b. *Right Node (WorkoutDuration > 42.5)*
 - i. *Feature: BMI <= 28.002*
 - ii. *Entropy: 0.937*
 - iii. *Samples: 34*
3. *Second Level Nodes*
 - a. *Left Node (Age <= 26.5)*
 - i. *Feature: BMI <= 24.387*
 - ii. *Entropy: 0.985*
 - iii. *Samples: 14*
 - b. *Right Node (Age > 26.5)*
 - i. *Entropy: 0.0*
 - ii. *Samples: 3*
4. *Third Level Nodes (Left Node of Age <= 26.5)*
 - a. *Left Node (BMI <= 24.387)*
 - i. *Feature: BMI <= 26.205*
 - ii. *Entropy: 0.918*
 - iii. *Samples: 12*
 - b. *Right Node (BMI > 24.387):*
 - i. *Entropy: 1.0*
 - ii. *Samples: 2*

4.3 Penyederhanaan Model Decision Tree

Model *decision tree* sebelumnya memiliki tingkat kompleksitas yang sangat tinggi dan sulit untuk dipahami oleh pengguna. Oleh karena itu, diperlukan pengembangan lebih lanjut untuk menyederhanakan model tersebut. Proses ini melibatkan evaluasi dan analisis mendalam terhadap setiap cabang dan *node* dalam model keputusan awal.

Langkah-langkah yang diambil meliputi penghapusan elemen yang tidak relevan, restrukturisasi keputusan, serta penerapan algoritma yang lebih efisien. Tujuannya adalah agar model lebih mudah dipahami dan diimplementasikan. Diharapkan dengan pendekatan ini, model keputusan yang baru dapat memberikan hasil lebih cepat, serta dapat diterapkan dalam berbagai situasi dengan lebih baik tanpa mengurangi akurasi.



Gambar 4. 2 Hasil Penyederhanaan Model *Decision Tree*

Gambar 4.2 adalah hasil dari model *decision tree* yang telah disederhanakan. Dalam konteks ini, digunakan untuk memprediksi kelas (*Low* atau *High*) berdasarkan beberapa fitur yaitu *WorkoutDuration*, *BMI*, dan *Gender*, dengan rincian sebagai berikut:

1. *Root Node*

- a. *WorkoutDuration* ≤ 32.5 (Pada *root node* ini, model memisahkan data berdasarkan *WorkoutDuration*, kemudian menghasilkan cabang, dengan jika durasi latihan kurang dari 32.5 atau sama maka masuk ke cabang kiri, sedangkan jika lebih akan masuk ke cabang kanan).
- b. *Entropy*: 1 (Ketidakpastian tinggi)
- c. *Samples*: 38

2. Cabang Kiri (*WorkoutDuration* ≤ 32.5):

- a. *BMI* ≤ 21.552
 - i. *Entropy*: 0.995
 - ii. *Samples*: 9
 - iii. *Node* ini memisahkan data berdasarkan *BMI*. Jika *BMI* kurang dari atau sama dengan 21.552, maka data masuk ke cabang kiri.
 - iv. Cabang Kiri:
 1. *Entropy*: 0.0
 2. *Samples*: 3
 - v. Cabang Kanan:
 1. *Entropy*: 0.0
 2. *Samples*: 6
- b. *BMI* > 21.552
 - i. *BMI* ≤ 25.48
 1. *Entropy*: 0.983
 2. *Samples*: 33

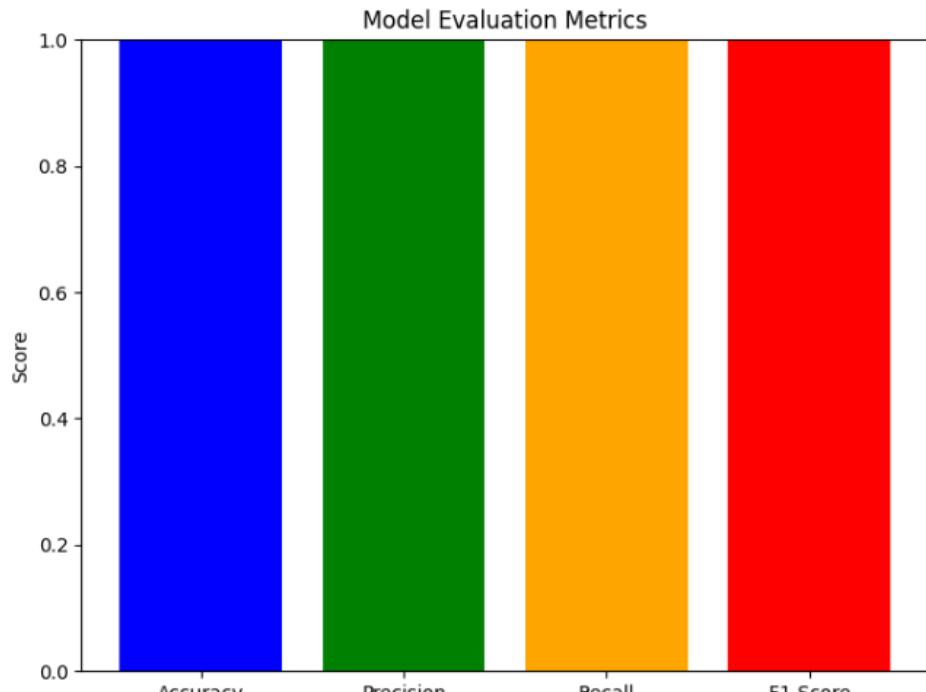
3. Cabang ini memisahkan data berdasarkan *BMI*, jika *BMI* kurang dari atau sama dengan 25.48 masuk ke cabang kiri.
4. Cabang Kiri:
 - a. *Entropy*: 0.0
 - b. *Samples*: 5
5. Cabang Kanan:
 - a. *Gender* ≤ 0.5
 - i. *Entropy*: 0.722
 - ii. *Samples*: 5
 - iii. Cabang Kiri:
 1. *Entropy*: 0.0
 2. *Samples*: 4
 - iv. Cabang Kanan:
 1. *Entropy*: 0.0
 2. *Samples*: 1
 - b. *Workout Duration* ≤ 37.5
 - i. *Entropy*: 1.0
 - ii. *Samples*: 28
 - iii. Cabang Kiri:
 1. *Entropy*: 0.0
 2. *Samples*: 5
 - iv. Cabang Kanan:
 1. *Workout Duration* ≤ 42.5
 - a. *Entropy*: 0.998
 - b. *Samples*: 23
 - c. Cabang Kiri:
 - i. *Entropy*: 0.0
 - ii. *Samples*: 5
 - d. Cabang Kanan:

- i. $WorkoutD$
 $uration \leq$
47.5

Setiap *node* dalam *decision tree* memisahkan data berdasarkan nilai *threshold* dari fitur yang dipilih. Pemisahan data akan terus berlanjut hingga tidak ada lagi data yang dapat dipisah lagi. Nilai entropi, jumlah sampel, dan distribusi kelas pada setiap node digunakan untuk mengevaluasi efektivitas pemisahan tersebut.

4.4 Testing Model

Model *C.45* yang sudah jadi dievaluasi kinerjanya menggunakan *confusion matrix*. *Confusion matrix* membantu memahami performa model dengan mengukur akurasi, *precision*, *recall*, dan *F1 Score*. Perhitungan *confusion matrix* menggunakan *library scikit-learn* dengan hasil sebagai berikut:



Gambar 4. 3 Hasil *Confusion Matrix*

Berikut penjelasan dari masing-masing metrik evaluasi yang ditampilkan dalam grafik pada gambar 4.3:

1. *Accuracy*: Metrik ini menunjukkan proporsi prediksi yang benar dari total prediksi yang dilakukan oleh model. Nilai akurasi yang mendekati 1 (atau 100%) menunjukkan bahwa model memiliki performa yang sangat baik dalam mengklasifikasikan data dengan benar. Dalam hal ini, model memiliki akurasi sekitar 1.0, yang menunjukkan performa sangat baik.
2. *Precision*: *Precision* mengukur proporsi prediksi positif yang benar dari total prediksi positif yang dilakukan oleh model. Nilai *precision* yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan yang rendah dalam memprediksi kelas positif. Grafik menunjukkan *precision* yang cukup tinggi, mendekati 1.0, menunjukkan bahwa sebagian besar prediksi positif oleh model adalah benar.
3. *Recall*: *Recall* mengukur proporsi kasus positif yang benar-benar diidentifikasi oleh model dari total kasus positif sebenarnya. Nilai *recall* yang tinggi menunjukkan bahwa model mampu menangkap sebagian besar kasus positif. Grafik menunjukkan *recall* yang juga mendekati 1.0, yang berarti model berhasil mengidentifikasi sebagian besar kasus positif dengan benar.
4. *F1 Score*: *F1 Score* adalah metrik yang mengombinasikan *precision* dan *recall* untuk memberikan gambaran keseimbangan antara keduanya. Nilai *F1 Score* yang tinggi menunjukkan bahwa model memiliki keseimbangan yang baik antara *precision* dan *recall*. Dalam grafik, *F1 Score* yang mendekati 1.0 menunjukkan bahwa model memiliki performa yang sangat baik dalam hal *precision* dan *recall*.

4.5 Rules dari Model *Decision Tree*

1. *Rule 1*:
 - a. *Condition*: $WorkoutDuration \leq 32.5$
 - b. *Output*:
 - i. *Workout Intensity*: *Low*

- ii. *Nutrition Plan: Moderate*
- iii. *Workout Program: Low Intensity*

2. *Rule 2:*

- a. *Condition: WorkoutDuration > 32.5 and BMI ≤ 21.552*
- b. *Output:*
 - i. *Workout Intensity: High*
 - ii. *Nutrition Plan: High Protein*
 - iii. *Workout Program: High Intensity*

3. *Rule 3:*

- a. *Condition: WorkoutDuration > 32.5 and BMI > 21.552 and BMI ≤ 25.48*
- b. *Output:*
 - i. *Workout Intensity: High*
 - ii. *Nutrition Plan: Balanced*
 - iii. *Workout Program: Moderate Intensity*

4. *Rule 4:*

- a. *Condition: WorkoutDuration > 32.5 and BMI > 25.48 and Gender = 0*
- b. *Output:*
 - i. *Workout Intensity: High*
 - ii. *Nutrition Plan: Low Carb*
 - iii. *Workout Program: Weight Training*

5. *Rule 5:*

- a. *Condition: WorkoutDuration > 32.5 and BMI > 25.48 and Gender = 1*
- b. *Output:*
 - i. *Workout Intensity: High*
 - ii. *Nutrition Plan: High Protein*
 - iii. *Workout Program: High Intensity*

6. *Rule 6:*

- a. *Condition: WorkoutDuration > 32.5 and BMI > 25.48 and BMI ≤ 29.95*

b. *Output:*

- i. *Workout Intensity: Low*
- ii. *Nutrition Plan: Balanced*
- iii. *Workout Program: Moderate Intensity*

7. *Rule 7:*

- a. *Condition: WorkoutDuration > 32.5 and BMI > 29.95 and Age ≤ 30.5*

b. *Output:*

- i. *Workout Intensity: High*
- ii. *Nutrition Plan: High Protein*
- iii. *Workout Program: High Intensity*

8. *Rule 8:*

- a. *Condition: WorkoutDuration > 32.5 and BMI > 29.95 and Age > 30.5*

b. *Output:*

- i. *Workout Intensity: Low*
- ii. *Nutrition Plan: Moderate*
- iii. *Workout Program: Low Intensity*

4.6 Hasil Wawancara Narasumber

Berikut adalah hasil wawancara dengan narasumber yang bersangkutan dan ahli di bidangnya. Dalam sesi ini, penulis menggali berbagai pandangan, wawasan, serta pengalaman narasumber untuk mendapatkan pemahaman yang lebih mendalam mengenai topik yang sedang dikaji. Berikut rincian hasil wawancara.

1. Latihan untuk *Upper Body*

a. *Chest*

- i. *Incline Bench Press:* Latihan untuk membentuk dan memperkuat otot dada bagian atas. Latihan ini dilakukan dengan berbaring di bangku yang dimiringkan (*incline*) dan mengangkat beban ke atas. Latihan ini menargetkan otot *pectoralis major* bagian atas, berfungsi untuk memberikan bentuk yang lebih penuh pada dada.

- ii. *Bench Press*: Latihan untuk mengembangkan kekuatan dan massa otot dada. Latihan ini dilakukan dengan berbaring di bangku datar dan mengangkat barbel ke atas. Fokus utama dari latihan ini adalah otot *pectoralis major*, tetapi juga melatih otot bahu dan trisep sebagai pendukung.
- iii. *Decline Bench Press*: Latihan yang menargetkan otot dada bagian bawah. Latihan ini dilakukan dengan berbaring di bangku yang dimiringkan ke bawah (*decline*) dan mengangkat barbel. Dengan variasi ini, otot *pectoralis major* bagian bawah lebih banyak terlatih.

b. *Shoulder*

- i. *Incline Bench Pull*: Latihan memperkuat otot bahu dan punggung bagian atas. Latihan ini dilakukan dengan berbaring di bangku yang dimiringkan ke depan dan menarik beban ke arah dada. Latihan ini menargetkan otot *deltoid* dan *trapezius*, membantu dalam membentuk bahu yang kuat dan bervolume.
- ii. *Lateral Rise*: Latihan isolasi yang menargetkan otot *deltoid* bagian tengah. Latihan ini dilakukan dengan berdiri tegak dan mengangkat *dumbbell* ke samping hingga sejajar dengan bahu.
- iii. *Overhead Press*: Latihan komprehensif untuk memperkuat seluruh otot bahu. Latihan ini dilakukan dengan mengangkat barbel atau *dumbbell* dari posisi bahu ke atas kepala. Selain otot *deltoid*, latihan ini juga melibatkan otot *trapezius* dan trisep.

c. *Wings*

- i. *Pull Up*: Latihan berat badan untuk memperkuat otot punggung, terutama otot *latissimus dorsi* (sayap). Latihan ini dilakukan dengan menarik tubuh ke atas

menggunakan palang. *Pull Up* juga melibatkan otot *biceps* dan otot punggung bagian atas.

- ii. *Dumbbell Pullover*: Latihan yang menargetkan otot punggung bagian atas dan otot dada. Latihan ini dilakukan dengan berbaring di bangku dan mengangkat *dumbbell* dari belakang kepala ke atas dada. Latihan ini membantu dalam memperluas dada dan memperkuat otot *latissimus dorsi*.
- iii. *Lateral Pulldown*: Latihan mesin yang menargetkan otot punggung, khususnya otot *latissimus dorsi*. Latihan ini dilakukan dengan menarik beban dari atas ke bawah menuju dada.

d. Tangan

- i. *Biceps*: Latihan *biceps* melibatkan berbagai variasi *curl*, seperti *Barbell Curl*, *Dumbbell Curl*, dan *Hammer Curl*. Latihan ini fokus pada otot *biceps brachii*, untuk membentuk dan memperkuat otot lengan bagian depan.
- ii. *Triceps*: Latihan *triceps* meliputi *Tricep Dips*, *Tricep Pushdown*, dan *Skull Crushers*. Latihan ini menargetkan otot *triceps brachii*, yang berada di belakang lengan atas.
- iii. *Forearms*: Latihan *forearms* meliputi *Wrist Curls*, *Reverse Wrist Curls*, dan *Farmer's Walk*. Latihan ini fokus pada otot-otot lengan bawah, untuk meningkatkan kekuatan cengkeraman dan stabilitas lengan.

2. Latihan untuk *Lower Body*:

a. *Glutes*:

- i. *Hip Thrust*: Latihan untuk memperkuat dan membentuk otot *glutes*. Latihan ini dilakukan dengan berbaring dan mengangkat pinggul ke atas dengan beban di pinggul. Latihan ini menargetkan otot *gluteus maximus*, memberikan kekuatan dan volume pada otot pantat.

- ii. *Low Bar Squats*: Latihan ini adalah variasi *squat* yang menargetkan otot *glutes* dan otot paha. Latihan ini dilakukan dengan memposisikan barbel lebih rendah di punggung dan melakukan gerakan *squat*.
- iii. *Crusty Lunge*: Latihan ini adalah variasi *lunge* yang menargetkan otot *glutes* dan paha. Latihan ini dilakukan dengan menyilangkan satu kaki ke belakang dan menekuk lutut seperti posisi *curtsy*.

b. *Hamstrings*:

- i. *Deadlift*: Latihan komprehensif yang menargetkan otot *hamstrings* dan punggung bawah. Latihan ini dilakukan dengan mengangkat beban dari lantai ke posisi berdiri. Latihan ini sangat efektif dalam meningkatkan kekuatan dan massa otot *hamstrings* serta otot punggung.
- ii. *Leg Curl*: Latihan isolasi yang menargetkan otot *hamstrings*. Latihan ini dilakukan dengan menekuk lutut melawan beban pada mesin *leg curl*.
- iii. *Dumbbell Hip Thrust*: Latihan ini adalah variasi *hip thrust* yang menggunakan *dumbbell* sebagai beban. Latihan ini menargetkan otot *hamstrings* dan *glutes*.

c. *Calves*:

- i. *Calf Press*: Latihan yang menargetkan otot betis (*calves*). Latihan ini dilakukan dengan menekan beban menggunakan kaki pada mesin *calf press*.
- ii. *Calf Raise*: Latihan sederhana namun efektif untuk melatih otot betis. Latihan ini dilakukan dengan berdiri dan mengangkat tumit ke atas.
- iii. *Jump Rope*: Latihan kardio yang efektif untuk memperkuat otot betis. Latihan ini dilakukan dengan melompat tali secara berulang-ulang. Latihan ini membantu dalam meningkatkan kekuatan otot betis serta daya tahan kardiovaskular.

d. *Thighs*:

- i. *Squats*: Latihan ini dilakukan dengan menekuk lutut dan menurunkan tubuh seperti duduk di kursi. *Squats* membantu dalam melatih otot paha (*thighs*) dan *glutes*.
- ii. *Sumo Squat*: Latihan ini adalah variasi *squat* yang menargetkan otot paha bagian dalam dan *glutes*. Latihan ini dilakukan dengan posisi kaki yang lebih lebar dan jari kaki mengarah ke luar.
- iii. *Leg Press*: Latihan dengan menggunakan mesin yang menargetkan otot paha dan *glutes*. Latihan ini dilakukan dengan menekan beban menggunakan kaki pada mesin *leg press*.

3. Latihan untuk *Abdominal*:

- a. *Upper Abs* (Bagian perut atas): Latihan untuk melatih otot perut bagian atas meliputi *Crunches*, *Sit-Ups*, dan *Leg Raises*. Latihan ini fokus pada otot *rectus abdominis* bagian atas.
- b. *Obliques Abs* (Bagian perut samping): Latihan untuk otot perut samping meliputi *Russian Twists*, *Side Planks*, dan *Bicycle Crunches*. Latihan ini menargetkan otot *obliques*, yang berada di sisi perut.
- c. *Lower Abas* (Bagian perut bawah): Latihan untuk otot perut bawah meliputi *Leg Raises*, *Reverse Crunches*, dan *Mountain Climbers*. Latihan ini fokus pada otot *rectus abdominis* bagian bawah.

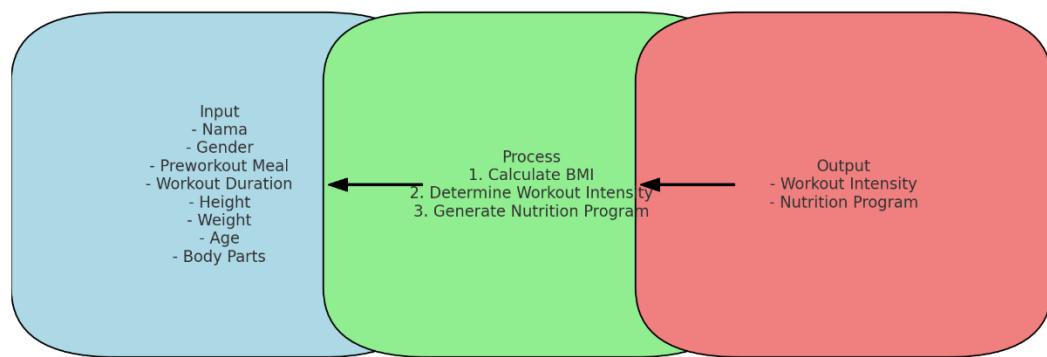
4.7 Arsitektur Sistem

Arsitektur sistem aplikasi terdiri dari beberapa komponen utama yaitu input pengguna, pemrosesan data menggunakan model *C4.5*, dan *output* berupa rekomendasi program nutrisi dan latihan.

1. *Input* Pengguna: Pengguna memasukkan data terkait latihan seperti tinggi badan, berat badan, jenis kelamin, dan umur.

2. Pemrosesan Data: Data *input* pengguna diproses menggunakan model *C4.5* untuk mengklasifikasikan intensitas latihan sebagai tinggi atau rendah. Model ini telah dilatih dan diuji untuk memastikan akurasi klasifikasinya.
3. *Output*: Berdasarkan hasil dari penggabungan klasifikasi intensitas latihan dan data dari narasumber, serta penelitian terdahulu (Rosenbloom, 2012), sistem kemudian menghasilkan program nutrisi dan latihan yang disesuaikan dengan kebutuhan pengguna.

4.8 Diagram IPO (*input, process, output*)



Gambar 4. 4 Diagram IPO

Input:

1. Nama
2. *Gender* (Perempuan / laki-laki)
3. *Preworkout Meal* (Jam)
4. *Workout Duration* (Jam)
5. *Height* (m)
6. *Weight* (kg)
7. *Age* (Tahun)
8. *Body Parts*

Process:

1. Menghitung *BMI*:

- a. Menggunakan tinggi (Height) dan berat badan (Weight) untuk menghitung nilai *BMI*.
2. Menentukan *Workout Intensity*:
 - a. Menggunakan nilai *BMI*, umur (Age), dan durasi latihan (Workout Duration) untuk menentukan intensitas latihan (Workout Intensity).
3. Menghasilkan Program Nutrisi:
 - a. Menggunakan nilai intensitas latihan (Workout Intensity) dan *preworkout meal* untuk menghasilkan program nutrisi dengan rumus yang sesuai.

Output:

1. *Workout Intensity*: Intensitas latihan yang dipersonalisasi berdasarkan *BMI*, umur, dan durasi latihan.
2. Program Nutrisi: Rekomendasi nutrisi yang disesuaikan dengan intensitas latihan dan makanan pra-latihan (*preworkout meal*).

4.9 Integrasi Model ke Dalam Aplikasi

1. Pengembangan *Backend*: Model *C4.5* diintegrasikan ke dalam *backend* aplikasi menggunakan bahasa pemrograman PHP dengan *framework Laravel*. Model ini diimplementasikan sebagai layanan yang dapat menerima data *input* dan mengembalikan hasil klasifikasi berupa intensitas latihan.
2. Antarmuka Pengguna (UI): Aplikasi menyediakan antarmuka yang intuitif untuk pengguna memasukkan data latihan mereka. UI ini juga menampilkan hasil klasifikasi yang sudah diimplementasikan ke dalam *logic* program berupa program nutrisi dan latihan.

4.10 Implementasi pada Aplikasi

Model *C4.5* dan *decision tree* yang telah dibuat, untuk memproses apakah *user* melakukan *workout* dengan intensitas tinggi atau rendah. Sehingga tidak mempersusah user untuk mengetahui apakah intensitas latihannya tinggi atau rendah. Dengan model yang sudah didapat dari hasil *training* model, dan hasil

akurasi yang baik dari hasil *testing* model, dapat menghasilkan intensitas latihan langsung dari inputan *user*. Dengan begitu, hasil dari data *input user* yang diproses di model (intensitas latihan), akan diproses kembali dengan input lainnya untuk menghasilkan program nutrisi dan program latihan yang sesuai.

The screenshot shows a user interface for data entry. At the top, a title 'Insert Data' is displayed. Below it, a section titled 'Personal Data' contains fields for 'Name' (a text input box), 'Gender' (radio buttons for 'Male' and 'Female'), 'Height (cm)' (a text input box), 'Weight (kg)' (a text input box), and 'Age' (a text input box). To the right of the gender section are three radio buttons for 'Pre-Workout Meal' duration: '< 1 hour', '1 hour', '2 hours', and '> 2 hours'. To the right of the meal section are three radio buttons for 'Workout Duration': 'Over an Hour', 'Under an Hour', and an unlabeled option. Below these are two radio buttons for 'Body Parts': 'Upper Body' and 'Lower Body'. At the bottom of the form are two buttons: a blue 'Submit' button and a dark grey 'Reset' button.

Gambar 4. 5 Tampilan Aplikasi pada *Desktop*

Gambar 4.5 menunjukkan tampilan aplikasi yang dirancang untuk memastikan pengguna dapat dengan mudah memasukkan dan mengirimkan data. Data-data yang dimasukkan terdiri dari beberapa bagian yang berbeda, dengan rincian sebagai berikut:

1. Judul *Form*
 - a. Terdapat pada bagian atas dengan judul “Insert Data” untuk memberikan informasi pada pengguna.
2. Bagian Data Pribadi
 - a. Nama: Kolom teks untuk pengguna memasukkan nama.
 - b. Jenis Kelamin: Opsi pilihan dengan dua pilihan “Male” untuk laki – laki dan “Female” untuk Perempuan.
 - c. Usia: Kolom teks untuk pengguna memasukkan umur.

- d. Tinggi Badan (cm): Kolom teks untuk memasukkan tinggi badan pengguna dalam satuan sentimeter.
- e. Berat Badan (kg): Kolom teks untuk memasukkan berat badan pengguna dalam satuan kilogram.

3. Bagian Kebiasaan Olahraga

- a. Intensitas Olahraga: Opsi pilihan dengan dua pilihan, “*High*” untuk intensitas tinggi dan “*Low*” untuk intensitas rendah.
- b. Makan Sebelum Olahraga: Opsi pilihan dengan empat pilihan, “< 1 hour” untuk makan kurang dari 1 jam sebelum, “1 hour” untuk 1 jam sebelum, “2 hours” untuk 2 jam sebelum, “> 2 hours” untuk lebih dari 2 jam sebelum.
- c. Durasi Olahraga: Opsi pilihan dengan dua pilihan, “Over an Hour” untuk durasi olahraga lebih dari 1 jam, “Under an Hour” untuk durasi olahraga kurang dari 1 jam.
- d. Bagian Tubuh yang Dilatih: Opsi pilihan dengan dua pilihan, “Upper Body” untuk bagian atas tubuh, “Lower Body” untuk bagian bawah tubuh.

4. Tombol Aksi:

- a. *Submit*: Tombol berwarna biru untuk mengirimkan data yang telah diisi oleh pengguna
- b. *Reset*: Tombol berwarna abu-abu untuk menghapus semua data yang telah diisi dan mengembalikan ke kondisi awal (kosong).

Insert Data

Personal Data

Name

Gender
 Male
 Female

Pre-Workout Meal
 < 1 hour
 1 hour
 2 hours
 > 2 hours

Workout Duration
 Over an Hour
 Under an Hour

Height (cm)

Weight (kg)

Gambar 4. 6 Tampilan Aplikasi pada *Mobile*

Gambar 4.6 menunjukkan sebuah tampilan entri data yang serupa dengan versi *desktop* tetapi dioptimalkan untuk tampilan *mobile*.

MyGym ≡

Overview

Profile Details

Name	JOHN
Height	80
Weight	90
Workout Duration	UPONE
Intensity	HIGH
BMI	140.63
BMI Index	OVERWEIGHT
Meal Before Workout	MINONE HOURS

Nutrition Needs [Workout Program Upper Body](#) [Workout Program Lower Body](#)

Nutrition	Sebelum	Sedang	Setelah
Karbohidrat	30	60	89.29
Protein	Secukupnya	Secukupnya	20

Gambar 4. 7 Tampilan Luaran pada *Desktop*

Gambar 4.7 menunjukkan tampilan bagian dari sistem yang digunakan oleh pengguna untuk memantau dan mengelola program latihan dan kebutuhan nutrisi mereka di *platform* "MyGym". Tampilan ini dirancang untuk membantu pengguna dalam mengelola dan memantau program kebugaran mereka secara efektif. Data-data yang ada pada luaran terbagi menjadi beberapa bagian yang berbeda dengan rincian sebagai berikut.

1. *Header*

- a. Logo: Terletak di bagian kiri atas halaman untuk menunjukkan nama *platform*.
- b. Menu Navigasi: Simbol tiga garis horizontal menunjukkan menu yang dapat diakses.

2. *Tab Overview*

a. *Profile Details:*

- i. *Name:* Nama pengguna.
- ii. *Height:* Tinggi badan pengguna.
- iii. *Weight:* Berat badan pengguna.
- iv. *Workout Duration:* Durasi latihan.
- v. *Intensity:* Intensitas latihan.
- vi. *BMI:* Indeks masa tubuh.
- vii. *BMI Index:* Kategori *BMI*.
- viii. *Meal Before Workout:* Waktu makan sebelum latihan.

3. *Tab Nutrisi dan Program Latihan*

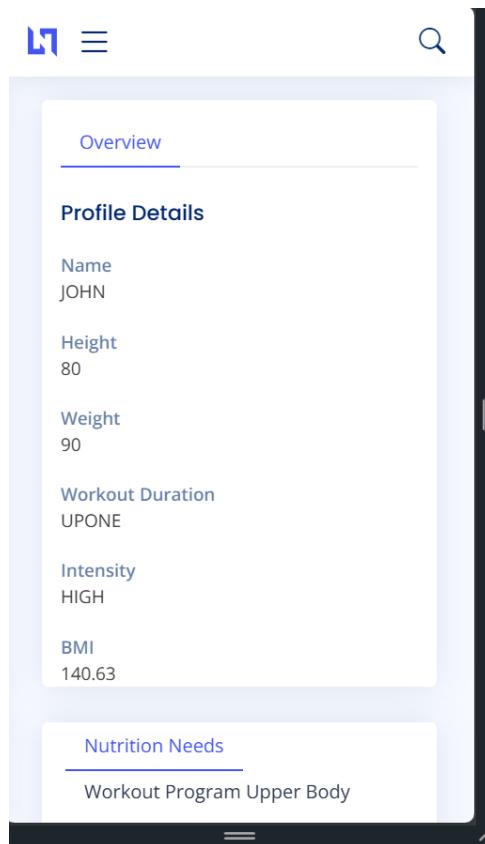
a. *Tab* ini berisi informasi kebutuhan nutrisi dan program latihan yang dibagi menjadi tiga bagian:

- i. *Nutrition Needs:* Menampilkan kebutuhan nutrisi sebelum, selama, dan sesudah latihan.
- ii. *Workout Program Upper Body:* Program latihan untuk bagian atas tubuh.
- iii. *Workout Program Lower Body:* Program latihan untuk bagian bawah tubuh.

4. Tabel Kebutuhan Nutrisi:

a. *Nutrition*: Menampilkan jenis nutrisi yang diperlukan, yang mencakup:

- Karbohidrat
- Protein



Gambar 4. 8 Tampilan Luaran pada *Mobile*

Gambar 4.8 menunjukkan tampilan entri data yang serupa dengan versi *desktop* tetapi dioptimalkan untuk tampilan *mobile*.

4.11 Pengujian Aplikasi dengan Contoh Case

Pengujian aplikasi menggunakan contoh kasus yang relevan. Melalui contoh kasus yang disajikan, kita akan melihat bagaimana pengujian dapat diterapkan secara efektif.

1. *Case 1*
 - Nama: John
 - Tinggi Badan: 180 cm

- c. Berat Badan: 90 kg
- d. Durasi Latihan: Di atas 1 jam
- e. Konsumsi Makanan: Kurang dari 1 jam sebelum latihan

Overview	
Profile Details	
Name	JOHN
Height	180
Weight	90
Workout Duration	UPONE
Intensity	HIGH
BMI	27.78
BMI Index	OVERWEIGHT
Meal Before Workout	MINONE HOURS

Gambar 4. 9 Hasil *Overview* dari *Input User*

Gambar 4.9 menunjukkan tampilan *overview* dengan detail profil sebagai berikut:

1. Nama: John
2. Tinggi: 180 cm
3. Berat: 90 kg
4. Durasi Latihan: Di atas 1 jam (UPONE)
5. Intensitas Latihan: Tinggi
6. *BMI*: 27,78
7. Klasifikasi *BMI*: Kelebihan Berat Badan
8. Makan Sebelum Latihan: Satu jam sebelum (MINONE HOURS)

Nutrition Needs	Workout Program Upper Body	Workout Program Lower Body	
Nutrition	Sebelum workout	Sedang Workout	Setelah Workout
Karbohidrat	30 gram	60 gram	89.29 gram
Protein	Secukupnya	Secukupnya	20 gram

Karbohidrat dan protein sangat penting dalam diet harian Anda. Pastikan untuk mengkonsumsi karbohidrat sebelum berolahraga untuk energi, dan protein setelah berolahraga untuk pemulihan otot.

Gambar 4. 10 Hasil Perhitungan Nutrisi

Gambar 4.10 menampilkan hasil perhitungan nutrisi berdasarkan data yang telah dimasukkan oleh pengguna. Data ini kemudian diproses menggunakan model

dan rumus perhitungan nutrisi untuk memberikan informasi yang lebih rinci dengan hasil sebagai berikut:

1. Kebutuhan Nutrisi

a. Sebelum *Workout*:

- i. Karbohidrat: 30-gram
- ii. Protein: Secukupnya

b. Sedang *Workout*:

- i. Karbohidrat: 60-gram
- ii. Protein: Secukupnya

c. Setelah *Workout*:

- i. Karbohidrat: 89.29-gram
- ii. Protein: 20-gram

Nutrition Needs		Workout Program Upper Body		Workout Program Lower Body	
No	Chest	Shoulder	Wings	Hand	Abs
1	Incline Bench Press	Incline Bench Pull	Pull Up	Biceps	Upper Abs
2	Bench Press	Lateral Rise	Dumbbell Pullover	Triceps	Obliques Abs
3	Decline Bench Press	Overhead Press	Lateral Pulldown	Forearm	Lower Abs

Gambar 4. 11 Program Latihan *Upper Body*

Gambar 4.11 menunjukkan program latihan yang dirancang khusus untuk melatih bagian atas tubuh. Program ini mencakup berbagai latihan yang ditujukan untuk mengembangkan kekuatan dan massa otot di area dada, punggung, perut, bahu, dan lengan.

Nutrition Needs		Workout Program Upper Body		Workout Program Lower Body	
No	Glutes	Hamstrings	Calves	Thighs	
1	Hip Thrust	Deadlift	Calf Press	Squats	
2	Low Bar Squats	Leg Curl	Calf Raises	Sumo Squat	
3	Crusty Lunge	Dumbbell Hip Thrust	Jump Rope	Leg Press	

Gambar 4. 12 Program Latihan *Lower Body*

Gambar 4.12 menampilkan program latihan yang dirancang khusus untuk melatih bagian bawah tubuh. Program ini mencakup berbagai latihan yang ditujukan untuk mengembangkan kekuatan dan massa otot di area kaki, paha, dan pinggul.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan implementasi model *decision tree* pada aplikasi untuk menentukan pengaruh atribut-atribut terhadap kategori tertentu, beberapa kesimpulan dapat diambil:

1. Efektivitas model klasifikasi dalam menganalisis pengaruh *workout intensity* terhadap asupan nutrisi:
 - a. Dengan menggunakan algoritma *C4.5* dan *decision tree*, model klasifikasi yang dikembangkan mampu menganalisis dan menjelaskan pengaruh intensitas latihan terhadap kebutuhan asupan nutrisi. Model ini membantu dalam mengidentifikasi pola-pola penting dan menentukan atribut-atribut yang paling berpengaruh terhadap kebutuhan nutrisi berdasarkan intensitas *workout*, yang dapat digunakan untuk memberikan rekomendasi nutrisi yang lebih akurat.
2. Panduan nutrisi komprehensif untuk berbagai jenis latihan:
 - a. Informasi yang disusun mengenai asupan nutrisi yang tepat untuk setiap jenis latihan memberikan panduan yang komprehensif bagi pengguna. Integrasi informasi ini memastikan bahwa setiap pengguna mendapatkan rekomendasi nutrisi yang sesuai dengan jenis dan intensitas latihannya, sehingga mendukung performa latihan yang optimal dan membantu dalam mencapai tujuan kebugaran dengan lebih efektif.
3. Pengembangan aplikasi pendamping olahraga berbasis PWA dengan rekomendasi yang dipersonalisasi:
 - a. Aplikasi pendamping olahraga berbasis Progressive Web App (PWA) dengan implementasi model *C4.5* dan *decision tree* memungkinkan penyediaan rekomendasi latihan dan nutrisi

yang dipersonalisasi. Aplikasi ini memanfaatkan data pengguna dan intensitas latihan untuk memberikan saran yang tepat, menjadikan pengalaman latihan lebih terarah dan membantu pengguna mencapai tujuan kebugaran mereka dengan cara yang lebih efisien dan tepat.

5.2 Saran

Berdasarkan hasil penelitian dan pengembangan aplikasi dari model *decision tree*, berikut adalah beberapa saran yang dapat diberikan untuk penelitian dan implementasi lebih lanjut.

1. Pengembangan Algoritma
 - a. Penelitian lebih lanjut dapat dilakukan untuk mengembangkan algoritma yang lebih efisien dalam menangani kompleksitas model *decision tree*. Ini dapat dilakukan penerapan teknik *pruning* atau metode lain yang dapat mengurangi kompleksitas tanpa mengurangi akurasi.
2. Antarmuka Aplikasi
 - a. Antarmuka aplikasi harus terus disederhanakan dan ditingkatkan berdasarkan pengalaman pengguna. Fokus pada kemudahan penggunaan dan pengalaman pengguna sangat penting untuk pengembangan aplikasi agar dapat digunakan oleh berbagai kalangan.

DAFTAR PUSTAKA

Adi, L. A. (2018). Platform e-learning untuk pembelajaran pemrograman web menggunakan konsep progressive web apps. *Jurnal Teknik ITS*, A781-A786.

Ali, E. Y. (2014). Penerapan Data Warehouse Menggunakan Teknologi Oracle Sebagai Solusi Big Data Pada Database Client Server. *JURNAL DIGIT*, 4(1), 12-21.

Al-Okbi. (2014). Urgent Need of Nutritional Strategy and Innovated Functional Foods for Athletes Health and Fitness. *Medical Journal of Islamic World Academy of Sciences*.

Alpina. (2022). Pemanfaatan Framework Laravel Dan Framework Bootstrap Pada Pembangunan Aplikasi Penjualan Hijab Berbasis Web. *Jurnal Media Infotama*, 18(1), 36-42.

Arimbi. (2018). Pengaturan Nutrisi Tepat . *Prosiding Seminar* , 442–444.

Azmi, M. F. (2022). Sistem Pakar Mendeteksi Gizi Buruk Pada Balita Menggunakan Metode Case Based Reasoning. *Jurnal Cyber Tech*, 1(3).

B. Firmanto, .. H. (2016). Perbandingan Kinerja Algoritma Promethee dan Topsis untuk Pemilihan Guru Teladan. *Jurnal Penelitian Pendidikan IPA*.

Beighley. (2010). *jQuery for Dummies*. John Wiley & Sons.

Bulqini, A. (2022). Nutrisi untuk Peak Performance bagi Atlet Profesional Sebelum, Selama, dan Setelah Kompetisi. *Jurnal Pendidikan Kesehatan Rekreasi Vol. 8, No. 2*.

Chowdhury. (2020). Nutrition for Athletes for Enhancement of Their Performance. *March*.

Dewi, G. L. (2020). Pemanfaatan Progressive Web Apps Pada Web Akuntansi. *Teknika*, 38-47.

Divya Mogaveera, V. M. (2021). e-Health Monitoring System with Diet and Fitness Recommendation using Machine Learning. *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, 694-700.

Harianja, J. P. (2010). Sistem Ujian Adaptif untuk Seleksi Penerimaan Mahasiswa Baru pada Perguruan Tinggi Berbasis Web melalui Penerapan Teknik MVC

dengan menggunakan PHP Framework Codeignitier. *Universitas Maritim Raja Ali Haji, Jurnal*.

Henderson. (2009). Encyclopedia of computer science and technology. *Infobase Publishing*.

Jannah, M. (2019). Mahir Bahasa Pemrograman PHP. *Elex Media Komputindo*.

Kaggle. (2021). Kaggle. Diambil kembali dari Kaggle: <https://www.kaggle.com/datasets/yassserh/bmidataset/>

Kerksick. (2017). International society of sports nutrition. *Journal of the International Society of Sports Nutrition*.

Kostrakiewicz-Gieralt. (2021). A systematic review of sweet potato-derived nutritional products for athletes. *Movement & Sport Sciences-Science & Motricité*.

Larose, D. T. (2006). Data mining methods & models. *John Wiley & Sons*.

Leonarda. (2018). Healthy athlete's nutrition. *Medicina Sportiva: Journal of Romanian Sports Medicine Society*.

Meyer. (2012). Fueling for fitness: Food and fluid recommendations for before, during, and after exercise. *ACSM's Health & Fitness Journal*.

Mueller, A. R. (2016). Effects of Carbohydrate Loading on High Performance Athletics. *Lorem Ipsum Solor Spring*.

Muhlisin, Z. &. (2020). Nutrisi bagi atlet remaja. *JTIKOR (Jurnal Terapan Ilmu Keolahragaan)*.

Pedregosa, F. V. (2011). Scikit-learn: Machine learning in python journal of machine learning research. *Journal of machine learning research*, 2825-2830.

Peeling. (2018). Evidence-based supplements for the enhancement of athletic performance. *International journal of sport nutrition and exercise metabolism*.

Prasetio. (2021). Simulasi Penerapan Metode Decision Tree (C4.5) Pada Penentuan Status Gizi Balita. *Jurnal Nasional Komputasi dan Teknologi Informasi Vol.4 No.3*, 210.

Pressman, R. S. (2002). Rekayasa perangkat lunak pendekatan praktisi (buku satu). *Yogyakarta: Andi*.

Rahman. (2021). Pengaturan nutrisi tepat bagi atlet. *Prosiding Seminar Nasional Lembaga Pengabdian Kepada Masyarakat Universitas Negeri Makassar ISBN*.

Rosenbloom. (2012). Food and fluid guidelines before, during, and after exercise. *Nutrition Today. Nutrition Today*.

Santi Dhara, K. C. (2015). A Study of VO2 max in Relation with Body Mass Index (BMI) of Physical. *Research Journal of Physical Education Sciences*, 9-12.

Sasmarianto. (2021). Understanding the needs of nutrition intake on athletes. *Journal Sport Area*.

Sharkey, B. J. (2003). Kebugaran dan kesehatan. *Jakarta: PT Raja Grafindo Persada*.

Sinuraya, J. Z. (2018). Perbandingan Pencarian Data Menggunakan Query Hash Join dan Query Nested Join. *Teknovasi*, 1(2), 71-93.

Sommerville. (2011). Software engineering 9th Edition. *ISBN-10, 137035152*.

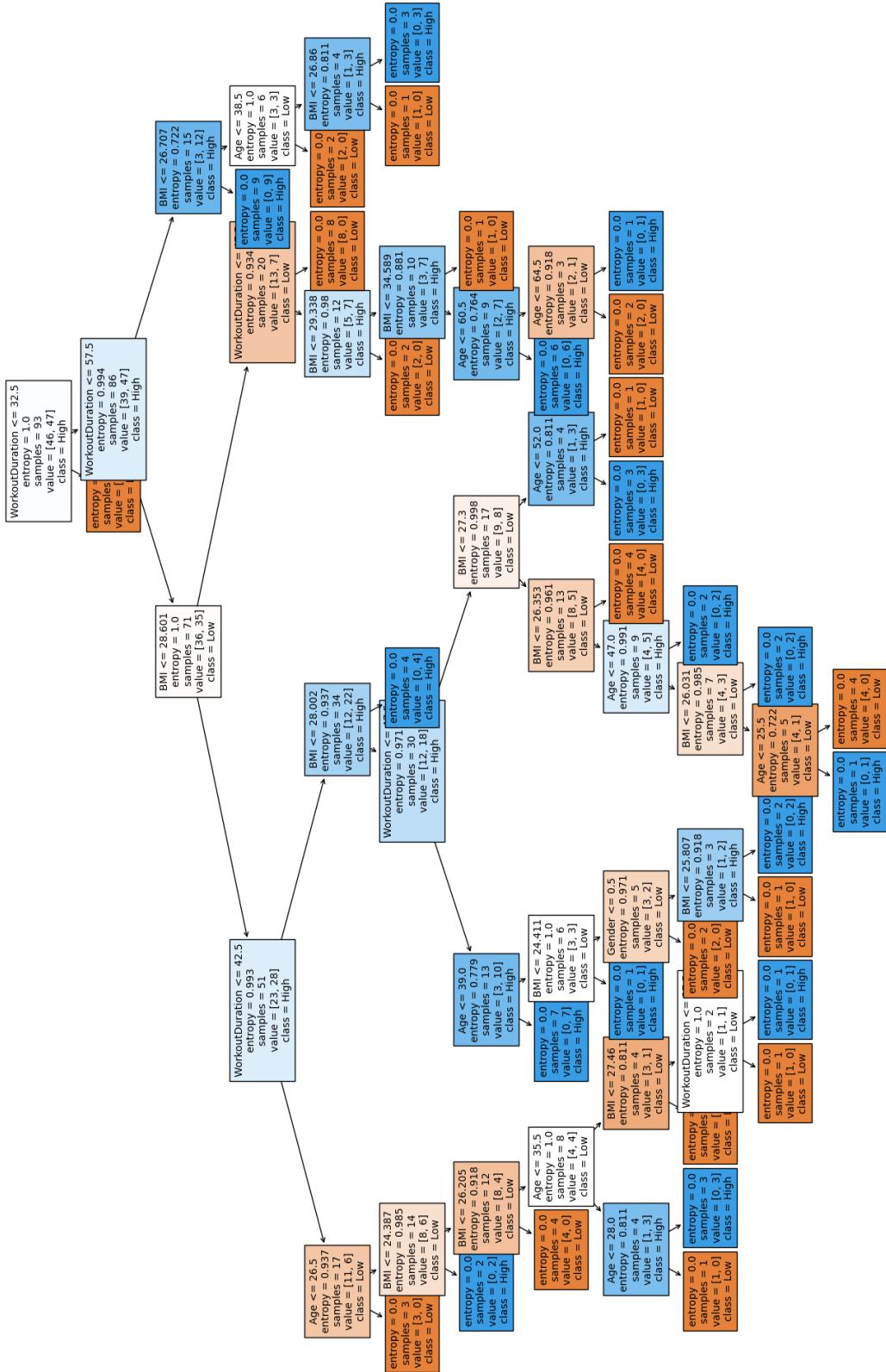
SPURLOCK. (2013). Responsive Web Development Bootstrap.

Supangat. (2018). Implementasi Decision Tree C4.5 UntukMenentukan Status Berat Badan dan Kebutuhan Energi Pada Anak Usia 7-12 Tahun. *Teknika*, 74.

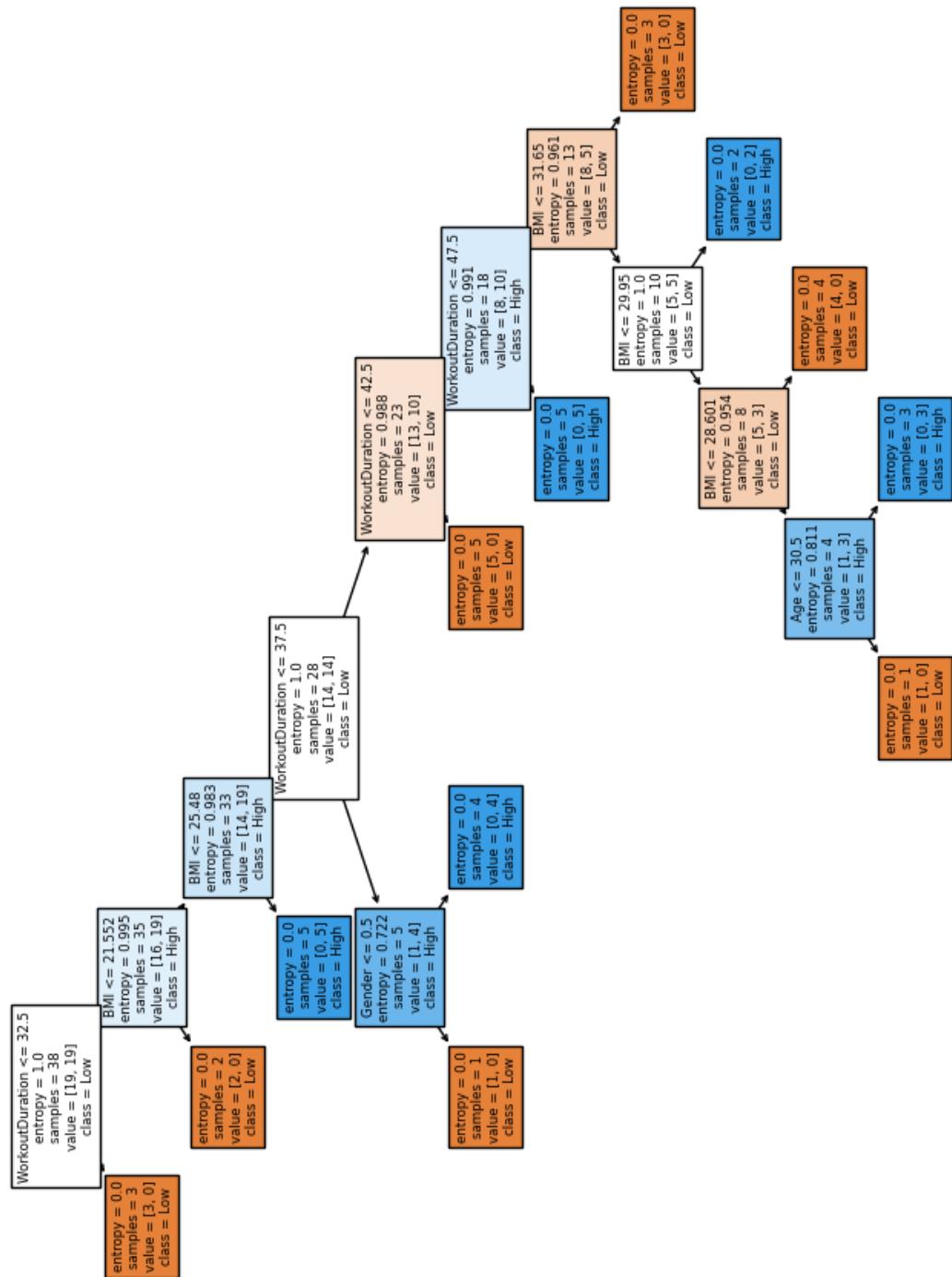
Zahra. (2020). Nutrisi bagi atlet remaja. *JTIKOR (Jurnal Terapan Ilmu Keolahragaan)*.

Zanecosky. (1986). Nutrition for athletes. *Clinics in podiatric medicine and surgery*.

LAMPIRAN



Lampiran 1. Model *Decision Tree*



Lampiran 2. Penyederhanaan Model *Decision Tree*

```

private function calculateIntensity($workoutDuration, $bmi, $gender, $age)
{
    if ($workoutDuration <= 32.5) {
        return 'high';
    } else {
        if ($workoutDuration <= 57.5) {
            if ($bmi <= 28.601) {
                if ($bmi <= 26.707) {
                    if ($age <= 38.5) {
                        return 'low';
                    } else {
                        return 'high';
                    }
                } else {
                    if ($age <= 26.86) {
                        return 'low';
                    } else {
                        return 'high';
                    }
                }
            } else {
                if ($workoutDuration <= 42.5) {
                    if ($age <= 26.5) {
                        return 'high';
                    } else {
                        if ($bmi <= 24.387) {
                            return 'low';
                        } else {
                            if ($bmi <= 26.205) {
                                if ($age <= 35.5) {
                                    return 'low';
                                } else {
                                    return 'high';
                                }
                            } else {
                                return 'high';
                            }
                        }
                    }
                } else {
                    return 'high';
                }
            }
        }
    }
}

```

Lampiran 3. Potongan Code untuk *Model Generation*

```

<div class="card-body">
    <h5 class="card-title">Personal Data</h5>
    <!-- Multi Columns Form -->
    <form id="userForm" class="row g-3">
        <div class="col-md-12">
            <label for="name" class="form-label">Name</label>
            <input type="text" class="form-control" id="name" required>
        </div>
        <div class="col-md-4">
            <label class="form-label">Gender</label>
            <div>
                <input type="radio" id="male" name="gender" value="male" required>
                    <label for="male">Male</label>
                </div>
                <div>
                    <input type="radio" id="female" name="gender" value="female" required>
                        <label for="female">Female</label>
                    </div>
                </div>
                <!-- <div class="col-md-4">
                    <label class="form-label">Workout Intensity</label>
                    <div>
                        <input type="radio" id="high" name="intensity" value="high" required>
                            <label for="high">High</label>
                        </div>
                        <div>
                            <input type="radio" id="low" name="intensity" value="low" required>
                                <label for="low">Low</label>
                            </div>
                        </div> -->
                    <div class="col-md-4">
                        <label class="form-label">Pre-Workout

```

Lampiran 4. Potongan *Code* untuk *Input User*

```

<div class="tab-content pt-2">
    <div class="tab-pane fade show active profile-overview" id="nutrition-needs">
        <table class="table table-bordered mb-4">
            <thead>
                <tr>
                    <th scope="col">Nutrition</th>
                    <th scope="col">Sebelum workout</th>
                    <th scope="col">Sedang Workout</th>
                    <th scope="col">Setelah Workout</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <th scope="row">Karbohidrat</th>
                    <td>{{ $formData->FBefore }} gram</td>
                    <td>{{ $formData->FDuring }} gram</td>
                    <td>{{ $formData->FCarbAfter }} gram</td>
                </tr>
                <tr>
                    <th scope="row">Protein</th>
                    <td>Secukupnya</td>
                    <td>Secukupnya</td>
                    <td>{{ $formData->FProAfter }} gram</td>
                </tr>
            </tbody>
        </table>
        <p class="explanation">
            Karbohidrat dan protein sangat penting dalam diet harian Anda. Pastikan untuk mengkonsumsi karbohidrat sebelum
        </p>
    </div>

```

Lampiran 5. Potongan *Code* untuk *Output User*