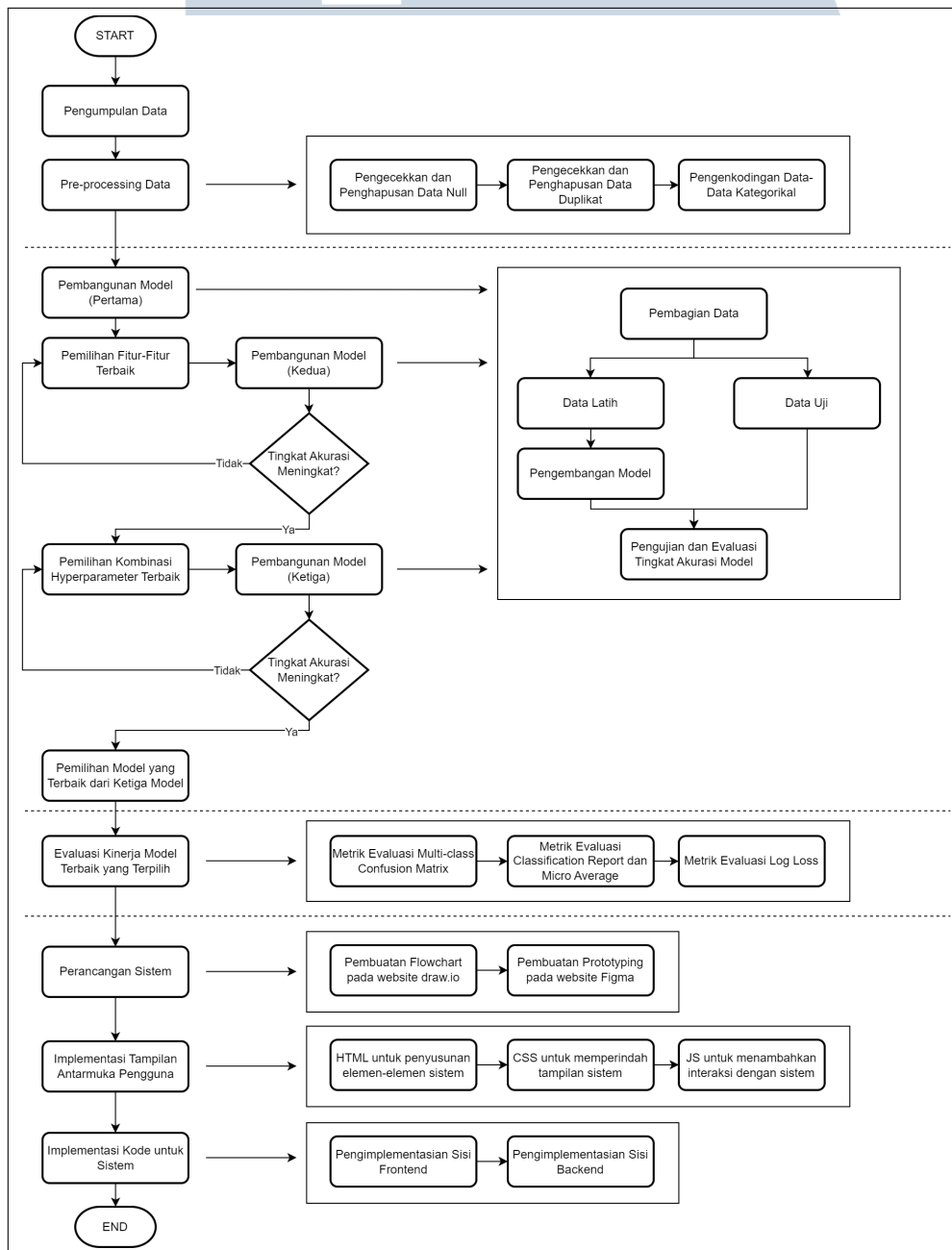


BAB 3 METODOLOGI PENELITIAN

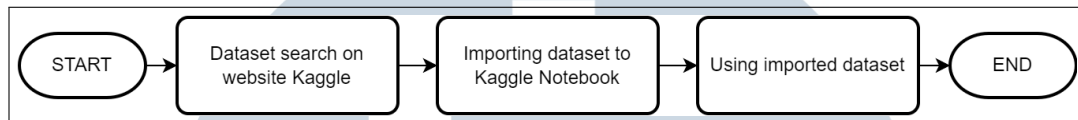
Gambar 3.1 merepresentasikan *framework* penelitian yang menjelaskan keseluruhan alur penelitian yang akan dilakukan, dari pengumpulan data hingga pengimplementasian kodingan.



Gambar 3.1. *Framework* Penelitian

3.1 Pengumpulan Data

Gambar 3.2 merepresentasikan *detail* tahapan untuk proses pengumpulan data.



Gambar 3.2. *Detail* Tahapan untuk Proses Pengumpulan Data

Pada penelitian ini, *dataset* yang digunakan adalah *dataset* yang berjudul "Estimation of obesity levels UCI" *dataset* yang didapatkan dari *website* kaggle.com yang berorisinil dari UCI *Machine Learning* yang selanjutnya di-*import* ke dalam *Kaggle Notebook* untuk pembangunan model. *Dataset* tersebut didapatkan dari hasil survei *online* pada *platform website* yang dapat diakses selama 30 hari [11]. Subjek dari penelitian tersebut adalah masyarakat Mexico, Peru, dan Columbia dengan rentang umur dari 14-61 tahun terhadap berbagai macam pola makan dan kondisi fisik [11]. Selain survei, data-data tersebut juga didapatkan secara sintesis menggunakan *SMOTE filter* untuk menyeimbangkan jumlah sampel antara kelas mayoritas dan minoritas [11]. *SMOTE* sendiri merupakan teknik *oversampling* yang menghasilkan sampel-sampel sintesis baru untuk kelas minoritas dengan melakukan interpolasi [31]. *SMOTE* pertama-tama akan memilih satu sampel secara acak dari kelas minoritas, lalu memilih satu sampel secara acak lagi dari tetangga-tetangga terdekatnya untuk melakukan interpolasi dengan memperkirakan nilai baru dari kedua sampel tersebut [31].

Dari seluruh proses tersebut, didapatkan sebuah *dataset* yang berjumlah 2111 data dan total 17 atribut, dengan 16 atribut merupakan fitur-fitur dari obesitas [11]. Fitur-fitur yang tersedia adalah jenis kelamin (*Gender*), umur (*Age*), tinggi badan (*Height*), berat badan (*Weight*), riwayat obesitas dalam keluarga (*family_history_with_overweight*), *Frequent consumption of high caloric food* (FAVC), *Frequency of consumption of vegetables* (FCVC), *Number of main meals* (NCP), *Consumption of food between meals* (CAEC), perokok (SMOKE), *Consumption of water daily* (CH2O), *Calories consumption monitoring* (SCC), *Physical activity frequency* (FAF), *Time using technology devies* (TUE), *Consumption of alcohol* (CALC), dan *Transportation used* (MTRANS). Lalu, 1 atribut lainnya adalah kelas "NObesidad" dengan *value* "Insufficient Weight", "Normal Weight", "Overweight Level I", "Overweight Level II", "Obesity Type I", "Obesity Type II", dan "Obesity Type III". Data-data untuk fitur didapatkan

dari hasil kuesioner dan SMOTE *filter*, sementara data untuk kelas didapatkan dari perhitungan BMI [11], sehingga *dataset* ini merupakan penggabungan dari nilai BMI dan fitur-fitur yang dapat mempengaruhi tingkat obesitas. Penggabungan ini dilakukan dengan tujuan untuk dapat menutupi kekurangan dari BMI sebagai indikator pengukuran tingkat obesitas. Tabel 3.1 menampilkan hasil survei untuk mendapatkan *dataset* yang terdiri dari pertanyaan dan kemungkinan jawaban yang diberikan responden [11].

Tabel 3.1. Pertanyaan dan Kemungkinan Jawaban dari Survei

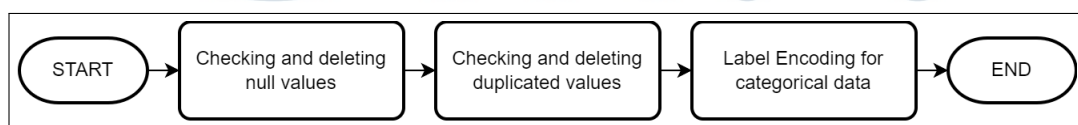
<i>Question</i>	<i>Possible Answers</i>
<i>What is your gender?</i>	- Male - Female
<i>What is your age?</i>	<i>Numerical value</i>
<i>What is your height?</i>	<i>Numerical value (m)</i>
<i>What is your weight?</i>	<i>Numerical value (kg)</i>
<i>Has a family member suffered/ suffers from overweight?</i>	- Yes - No
<i>Do you eat high caloric food frequently?</i>	- Yes - No
<i>Do you usually eat vegetables in your meals?</i>	- Never - Sometimes - Always
<i>How many main meals do you have daily?</i>	- Between 1 and 2 - 2 or 3 - More than 3
<i>Do you eat any food between meals?</i>	- No - Sometimes - Frequently - Always
<i>Do you smoke?</i>	- Yes - No
<i>How much water do you drink daily?</i>	- Less than 1 liter - Between 1 and 2 liter - More than 2 liter
Lanjut pada halaman berikutnya	

Tabel 3.1 Pertanyaan dan Kemungkinan Jawaban dari Survei (lanjutan)

<i>Question</i>	<i>Possible Answers</i>
<i>Do you monitor the calories you eat daily?</i>	- Yes - No
<i>How often do you have physical activity?</i>	- I don't have - 1 or 2 days - 2 or 4 days - 4 or 5 days
<i>How much time do you use technological devices, such as cellphone, video games, television, computer, and others?</i>	- 0 to 2 hours - 3 to 5 hours - More than 5 hours
<i>How often do you drink alcohol?</i>	- I don't drink - Sometimes - Frequently - Always
<i>Which transportation do you usually use?</i>	- Automobile - Motorbike - Bike - Public Transportation - Walking

3.2 Pre-processing

Gambar 3.3 merepresentasikan *detail* tahapan untuk proses *pre-processing*.



Gambar 3.3. *Detail* Tahapan untuk Proses *Pre-processing*

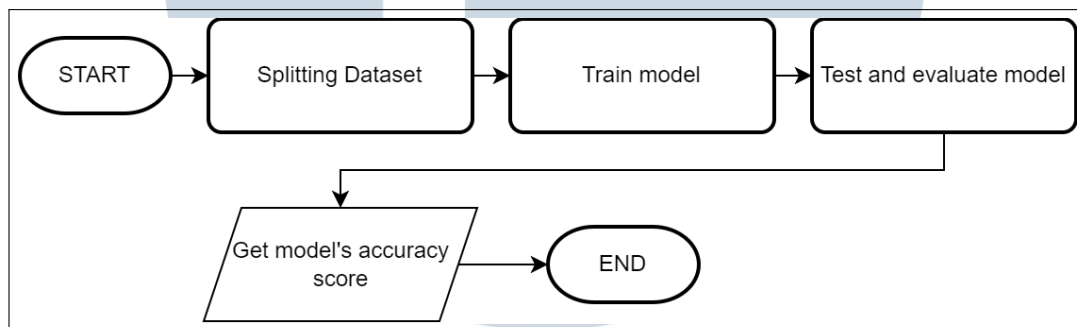
Pada tahapan ini, pertama-tama akan dilakukan pengecekan dan penghapusan data untuk seluruh data yang bernilai *null* pada *dataset* jika jumlahnya kecil. Tujuannya adalah untuk memastikan konsistensi data dan mencegah terjadinya bias dalam proses analisis untuk meningkatkan performa model. Selain itu, akan dilakukan pengecekan dan penghapusan data untuk seluruh data duplikat.

Tujuannya adalah untuk memastikan bahwa setiap *instance* dalam *dataset* bersifat unik untuk mencegah terjadinya bias dalam proses analisis.

Selain itu, pada tahapan ini, data yang telah diolah akan direpresentasikan ulang agar dapat dimengerti oleh komputer dengan melakukan *Label Encoding*. Hal ini dilakukan untuk mengubah seluruh data kategorikal data menjadi data numerikal karena diperlukan representasi numerik dari data-data kategorikal.

3.3 Pembangunan Model

Gambar 3.4 merepresentasikan *detail* tahapan untuk proses *feature selection* dan pembangunan model yang pertama.

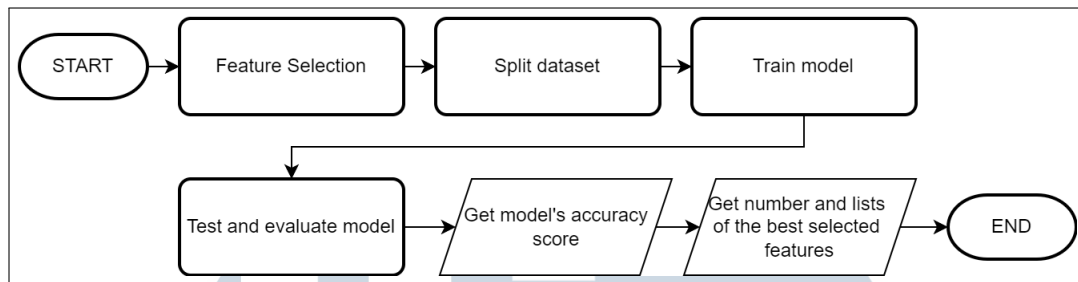


Gambar 3.4. *Detail* Tahapan untuk Proses Pembangunan Model

Tahapan ini merupakan tahapan awal dari proses *model selection* yang diawali dengan pembangunan dan pengecekan akurasi model awal terlebih dahulu menggunakan algoritma *Random Forest Classifier* untuk mengklasifikasikan tingkat obesitas dan mendapatkan tingkat akurasi awal model. Proses ini dilakukan dengan melakukan pembagian *dataset* menggunakan skenario *holdout* yang akan membagi *dataset* ke dalam 2 bagian, yaitu data *training* dan *testing* [32]. Pembagian *dataset* akan dilakukan dengan perbandingan 8:2, dimana sebanyak 80% data akan digunakan untuk melatih model dengan menggunakan data *training*, dan 20% sisanya akan digunakan untuk menguji seberapa baik model dalam melakukan klasifikasi dengan data *testing*. Setelah itu, akan dilakukan pengecekan akurasi model menggunakan fungsi "accuracy_score()" untuk mendapatkan tingkat akurasi awal model.

3.4 *Feature Selection* dan Pembangunan Model

Gambar 3.5 merepresentasikan *detail* tahapan untuk proses *hyperparameter tuning* dan pembangunan model yang kedua.

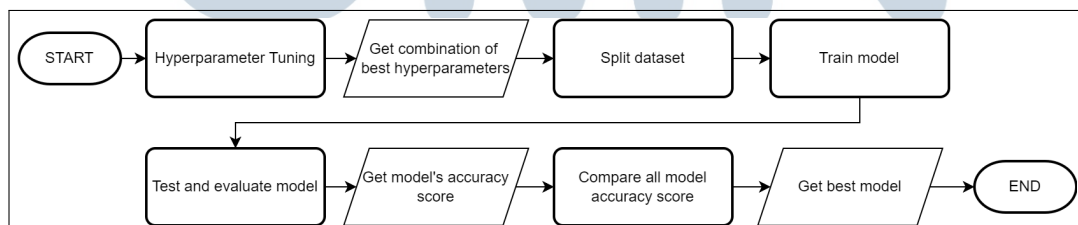


Gambar 3.5. *Detail* Tahapan untuk Proses *Feature Selection* dan Pembangunan Model

Tahapan ini merupakan tahapan kedua dari proses *model selection* yang diawali dengan dilakukannya proses seleksi fitur menggunakan metode ANOVA (*Analysis of Variance*) *F-score* dengan bantuan dari fungsi "SelectKBest()" dengan total fitur awal berjumlah 16 fitur. Proses seleksi fitur ini akan dilakukan sebanyak 16 kali, sesuai jumlah fitur dalam *dataset*. Pada setiap iterasi ke-*k*, data akan kembali dibagi menjadi 2 bagian menggunakan skenario *holdout* dengan perbandingan 8:2 untuk pembangunan model kedua yang akan dilatih menggunakan fitur-fitur terpilih untuk selanjutnya dilakukan pengecekan nilai akurasi pada model yang baru menggunakan fungsi "accuracy_score()". Jumlah fitur yang paling optimal dan fitur-fitur terbaik yang terpilih akan diambil dari model yang menghasilkan nilai akurasi yang tertinggi dari proses iterasi. Fitur-fitur terpilih tersebut nantinya akan dijadikan sebagai daftar pertanyaan pada sistem yang akan dibuat.

3.5 *Hyperparameter Tuning* dan Pembangunan Model

Gambar 3.6 merepresentasikan *detail* tahapan untuk proses *hyperparameter tuning* dan pembangunan model yang ketiga.



Gambar 3.6. *Detail* Tahapan untuk Proses *Hyperparameter Tuning* dan Pembangunan Model

Tahapan ini merupakan tahapan ketiga dari proses *model selection* yang diawali dengan dilakukannya proses *hyperparameter tuning* terlebih dahulu untuk mendapatkan kombinasi *hyperparameter values* terbaik yang

dapat menghasilkan kinerja klasifikasi model yang optimal. *Hyperparameter tuning* ini dilakukan menggunakan *Randomized Search* dengan bantuan dari fungsi "RandomizedSearchCV()" menggunakan *10-fold cross-validation* karena merupakan nilai *k-fold* yang paling optimum [26]. Tabel 3.2 merepresentasikan seluruh *hyperparameter* beserta *tuning values* yang akan digunakan.

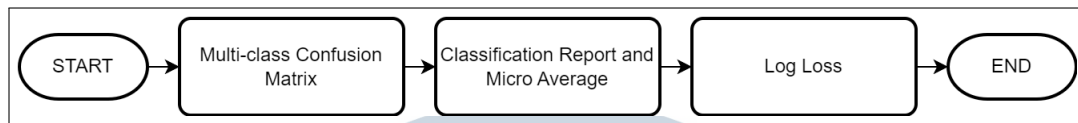
Tabel 3.2. *Hyperparameter tuning* untuk pembuatan model *Random Forest*

<i>Hyperparameter</i>	<i>Tuning Values</i>
<i>criterion</i>	"gini", "entropy", "log_loss"
<i>max_depth</i>	<i>Integer</i> 1-100
<i>n_estimators</i>	<i>Integer</i> 10-200
<i>max_features</i>	"sqrt", "log2", "None"
<i>min_samples_split</i>	<i>Integer</i> 2-5
<i>max_leaf_nodes</i>	<i>Integer</i> 10-500

Setelah itu, data akan kembali dibagi menjadi 2 bagian menggunakan skenario *holdout* dengan perbandingan 8:2 untuk pembangunan model ketiga yang akan dilatih menggunakan kombinasi *hyperparameter values* terbaik dari hasil *hyperparameter tuning*. Lalu, nilai akurasi dari model yang baru ini akan dicek kembali menggunakan fungsi "accuracy_score()". Lalu, karena tahapan ini merupakan tahapan terakhir dari proses *model selection*, maka setelah model ketiga dibangun, tahapan selanjutnya adalah membandingkan ketiga model yang telah dibangun untuk dapat mengetahui dan mendapatkan model yang menghasilkan tingkat akurasi yang tertinggi. Hal ini dikarenakan, model terbaik tersebut adalah model yang dipilih sebagai model klasifikasi pada penelitian ini untuk selanjutnya di evaluasi menggunakan beberapa metrik evaluasi dan diimplementasikan ke dalam sistem klasifikasi yang akan dibuat.

3.6 Evaluasi Kinerja Model

Gambar 3.7 merepresentasikan *detail* tahapan untuk proses evaluasi kinerja model.

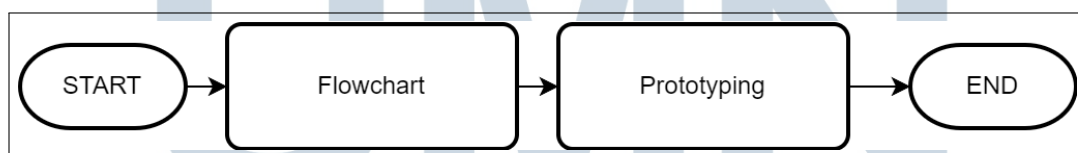


Gambar 3.7. *Detail* Tahapan untuk Proses Evaluasi Kinerja Model

Pada tahapan ini, pertama-tama akan digunakan *multi-class classification confusion matrix* untuk mengetahui sebaran kelas hasil klasifikasi yang dihasilkan oleh model yang terpilih terhadap kelas yang sebenarnya. Setelah itu, akan dilakukan pengecekan lanjutan untuk dapat mengevaluasi performa model secara keseluruhan untuk setiap kelas berdasarkan nilai *precision*, *recall*, dan *f1-score*, serta *macro average*, *weighted average*, dan *micro average* dari fungsi "classification_report()" dan perhitungan *micro average*. Terakhir, akan dilakukan juga pengecekan tingkat ketidakpastian atau kesalahan model dalam memprediksi probabilitas kelas dengan menggunakan metrik evaluasi "Log Loss" karena dapat digunakan untuk kasus *multi-class classification* dan sejalan dengan cara kerja dari algoritma *Random Forest* itu sendiri yang juga menghitung probabilitas kelas pada setiap *node* untuk dijadikan sebagai hasil klasifikasi pada *node* tersebut, sehingga cocok.

3.7 Perancangan Sistem

Gambar 3.8 merepresentasikan *detail* tahapan untuk proses perancangan sistem.

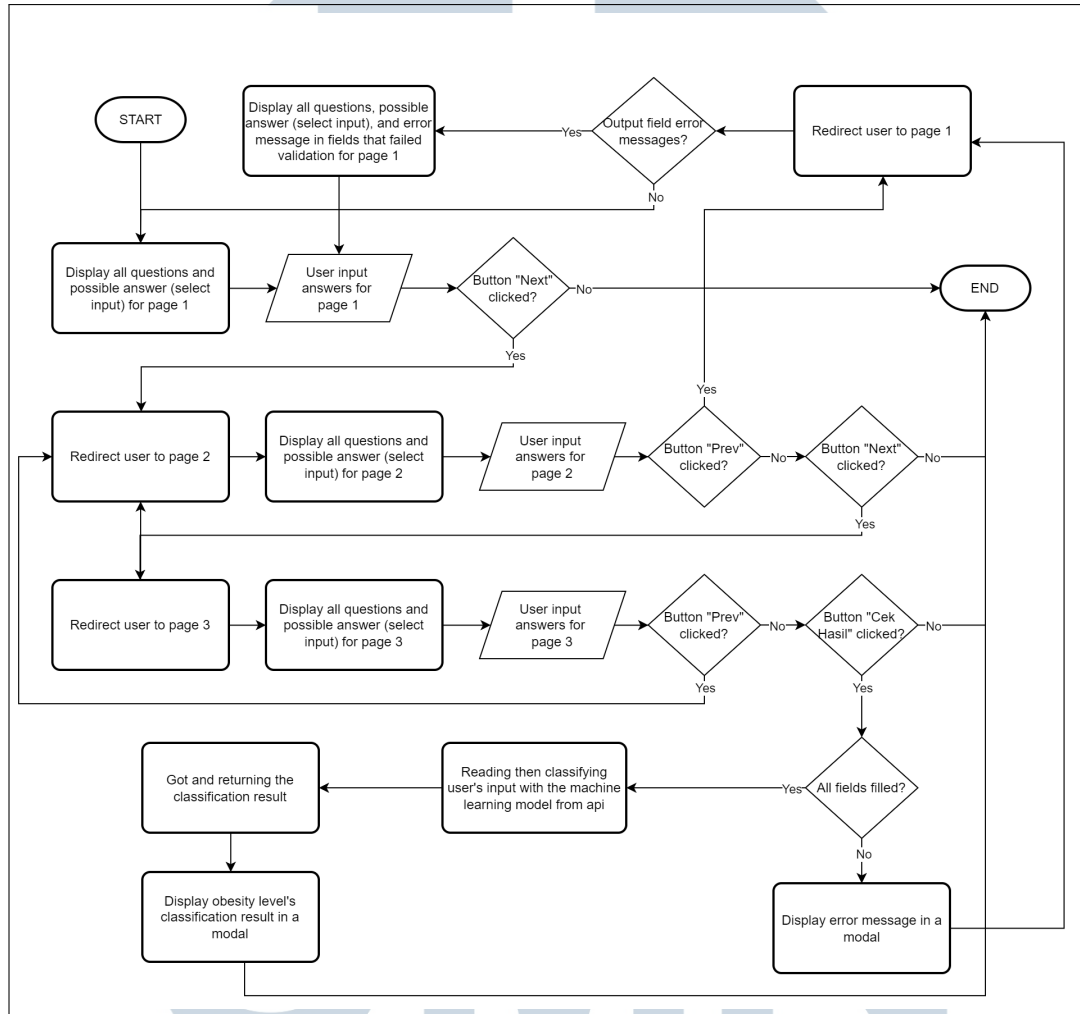


Gambar 3.8. *Detail* Tahapan untuk Proses Perancangan Sistem

Pada tahapan ini, pertama-tama akan dilakukan pembuatan *flowchart* terlebih dahulu pada *website* draw.io untuk mendapatkan gambaran terkait alur dari sistem yang akan dibuat. Setelah itu, akan dilakukan perancangan dari sistem yang akan dibuat dalam bentuk *prototype* pada Figma. Dalam proses *prototyping*, akan dirancang *user interface* beserta alur sistem hingga didapatkan hasil akhir klasifikasi untuk setiap pengguna. Berikut adalah penjelasan terkait *flowchart* dan hasil *prototyping* untuk setiap halaman dari sistem yang akan dibuat.

3.7.1 Flowchart

Gambar 3.9 merepresentasikan keseluruhan alur dari dari sistem klasifikasi tingkat obesitas yang digambarkan dalam bentuk *flowchart*.



Gambar 3.9. Alur Sistem

Saat sistem pertama kali diakses, pengguna akan langsung diarahkan ke halaman 1. Pada halaman ini, aplikasi akan menampilkan *list* pertanyaan beserta opsi jawaban untuk *select input*. *Action* yang dapat dilakukan adalah mengisi *fields* dan melakukan navigasi halaman dengan *button* "Next".

Pada halaman 2, aplikasi akan menampilkan *list* pertanyaan baru beserta opsi jawaban untuk *select input*. *Action* yang dapat dilakukan adalah mengisi *fields* dan melakukan navigasi halaman dengan *button* "Prev" dan "Next".

Pada halaman 3, aplikasi akan menampilkan *list* pertanyaan baru beserta opsi jawaban untuk *select input*. *Action* yang dapat dilakukan adalah mengisi

fields, melakukan navigasi halaman dengan *button* "Prev", serta mendapatkan hasil klasifikasi dengan *button* "Cek Hasil". Pada saat *button* "Cek Hasil" diklik, aplikasi pertama-tama akan melakukan validasi untuk seluruh *field* terlebih dahulu. Jika terdapat *field* yang kosong, maka aplikasi akan menampilkan pesan *error* dalam bentuk modalitas. Setelah itu, pengguna akan diarahkan kembali ke halaman pertama dengan pesan *error* yang ditampilkan di bawah setiap *field* yang masih kosong. Di sisi lain, jika seluruh *field* telah terisi, maka aplikasi akan memanggil API untuk melakukan klasifikasi dari hasil inputan pengguna. Setelah itu, API akan mengembalikan sebuah respon berupa hasil klasifikasi yang kemudian akan ditampilkan di dalam sebuah modalitas.

3.7.2 *Prototype*

Sistem klasifikasi tingkat obesitas yang akan dibuat akan berbentuk seperti sebuah *online form* dengan total 16 pertanyaan sesuai jumlah fitur awal. Dikarenakan jumlah pertanyaan yang cukup banyak, maka ke-16 pertanyaan tersebut akan dipisah ke dalam 3 halaman/ grup. Ketiga halaman tersebut akan dikelompokkan berdasarkan data diri, kesehatan keluarga dan pola makan, serta rutinitas. Selain itu, terdapat juga 2 jenis modalitas sebagai hasil *output* klasifikasi, yaitu modalitas *error* dan sukses. Berikut adalah penjelasan untuk keseluruhan *prototyping* tersebut.

A **Halaman Data Diri**

Pada halaman ini, pengguna akan diminta untuk mengisi 4 data yang berkaitan dengan informasi pribadinya, yaitu jenis kelamin, umur, tinggi badan, dan berat badan, serta navigasi halaman dengan *button* "Next". Gambar 3.10 merepresentasikan keseluruhan tampilan untuk halaman 1.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Gambar 3.10. Tampilan untuk Halaman Data Diri

Field untuk jenis kelamin merupakan *select field* yang terdiri dari 2 opsi/kemungkinan jawaban seperti pada Gambar 3.11. Sementara itu, 3 *field* lainnya merupakan *input field* yang dapat diisi secara manual melalui pengetikkan di *keyboard*.

Gambar 3.11. Opsi untuk Pertanyaan 1 pada Halaman 1

Pada halaman ini, terdapat validasi *input* untuk setiap *field* yang belum terisi saat ingin melakukan pengecekan tingkat obesitas. Gambar 3.12 merepresentasikan pesan *error* yang muncul akibat validasi *input*.

Prediksi Tingkat Obesitas dengan Algoritma Random Forest Classifier

Data Diri

Apa jenis kelamin anda?

Field tidak boleh kosong

Berapakah umur anda saat ini?

Field tidak boleh kosong

Berapakah tinggi badan anda dalam satuan centimeter?
 cm
Field tidak boleh kosong

Berapakah berat badan anda dalam satuan kilogram?
 kg
Field tidak boleh kosong

Next

Gambar 3.12. Tampilan untuk Validasi *Input* Halaman 1

B Halaman Kesehatan Keluarga dan Pola Makan

Pada halaman ini, pengguna akan diminta untuk mengisi 4 data yang berkaitan dengan riwayat kesehatan keluarga dan kebiasaan makan pribadi sehari-hari, yaitu riwayat *overweight* dalam keluarga, kebiasaan mengkonsumsi makanan berkalori tinggi, frekuensi konsumsi sayuran, dan jumlah makan berat per hari, serta navigasi halaman dengan *button* "Prev" dan "Next". Gambar 3.13 merepresentasikan keseluruhan tampilan untuk halaman 2.

U N I V E R S I T A S
 M U L T I M E D I A
 N U S A N T A R A

Gambar 3.13. Tampilan untuk Halaman Kesehatan Keluarga dan Pola Makan

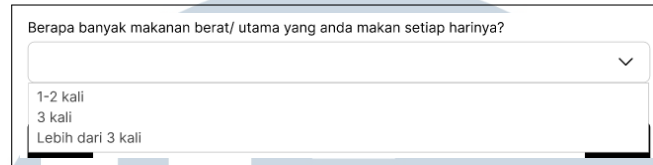
Seluruh *field* pada halaman ini merupakan *select field*. *Field* riwayat *overweight* dalam keluarga dan *field* konsumsi makanan berkalori tinggi terdiri dari 2 opsi jawaban seperti pada Gambar 3.14.

Gambar 3.14. Opsi untuk Pertanyaan 1 dan 2 pada Halaman 2

Field konsumsi sayuran saat makan berat terdiri dari 3 opsi jawaban seperti pada Gambar 3.15.

Gambar 3.15. Opsi untuk Pertanyaan 3 pada Halaman 2

Field jumlah makan berat per hari terdiri dari 3 opsi jawaban seperti pada Gambar 3.16.

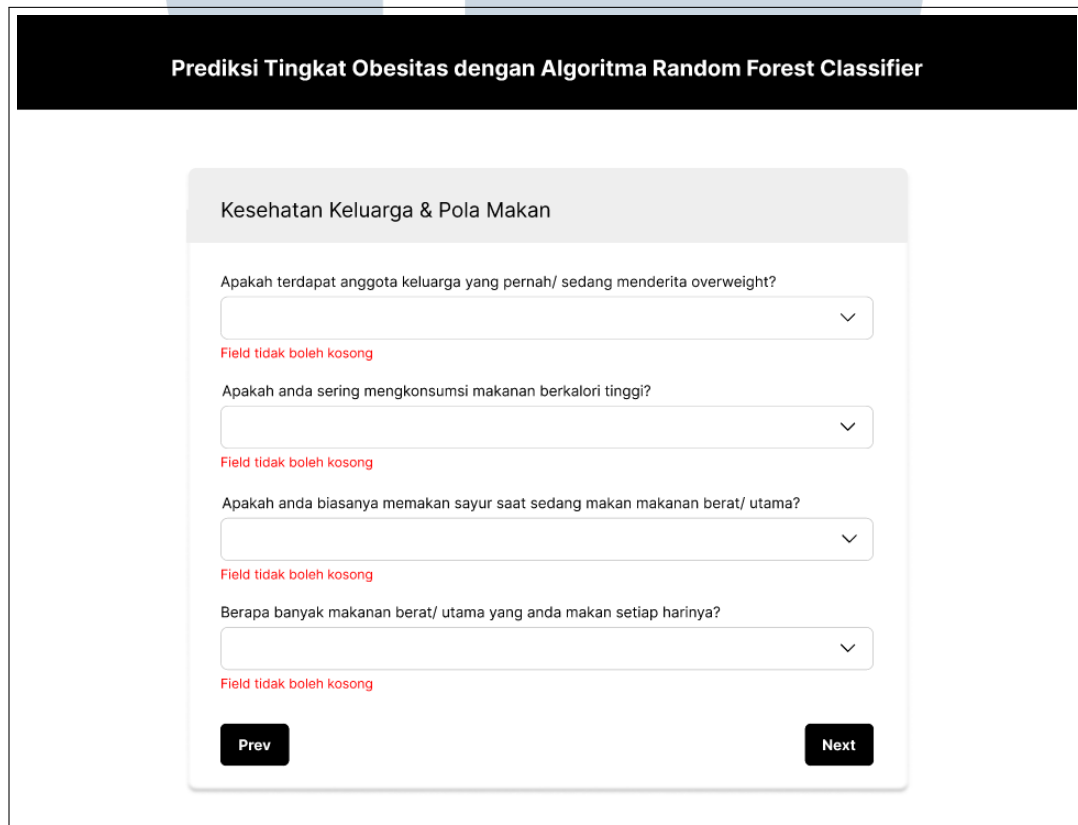


Berapa banyak makanan berat/ utama yang anda makan setiap harinya?

1-2 kali
3 kali
Lebih dari 3 kali

Gambar 3.16. Opsi untuk Pertanyaan 4 pada Halaman 2

Pada halaman ini, terdapat validasi *input* untuk setiap *field* yang belum terisi saat ingin melakukan pengecekan tingkat obesitas. Gambar 3.17 merepresentasikan pesan *error* yang muncul akibat validasi *input*.



Prediksi Tingkat Obesitas dengan Algoritma Random Forest Classifier

Kesehatan Keluarga & Pola Makan

Apakah terdapat anggota keluarga yang pernah/ sedang menderita overweight?

Field tidak boleh kosong

Apakah anda sering mengonsumsi makanan berkalori tinggi?

Field tidak boleh kosong

Apakah anda biasanya memakan sayur saat sedang makan makanan berat/ utama?

Field tidak boleh kosong

Berapa banyak makanan berat/ utama yang anda makan setiap harinya?

Field tidak boleh kosong

Prev Next

Gambar 3.17. Tampilan untuk Validasi *Input* Halaman 2

C Halaman Rutinitas

Pada halaman ini, pengguna akan diminta untuk mengisi 8 data yang berkaitan dengan rutinitas pribadi, yaitu kebiasaan mengemil, perokok, jumlah konsumsi air per hari, pemantauan jumlah kalori per hari, frekuensi beraktivitas fisik, frekuensi menggunakan perangkat elektronik, frekuensi meminum alkohol,

dan jenis transportasi yang biasanya digunakan, serta navigasi halaman dengan *button* "Prev" dan pengecekan hasil klasifikasi dengan *button* "Cek Hasil". Gambar 3.18 merepresentasikan keseluruhan tampilan untuk halaman 3.

Prediksi Tingkat Obesitas dengan Algoritma Random Forest Classifier

Rutinitas

Apakah anda mengemil diantara jam makan berat/ utama anda?

Apakah anda merokok?

Berapa liter air yang anda minum setiap hari nya?

Apakah anda memantau kalori yang anda makan setiap harinya?

Seberapa sering anda beraktivitas fisik?

Berapa lama anda biasanya menggunakan perangkat elektronik?

Seberapa sering anda minum minuman beralkohol?

Jenis transportasi apa yang biasanya anda gunakan?

Prev

Cek Hasil

Gambar 3.18. Tampilan untuk Halaman Rutinitas

Seluruh *field* pada halaman ini juga merupakan *select field*. *Field* kebiasaan mengemil dan *field* frekuensi meminum alkohol terdiri dari 4 opsi jawaban seperti pada Gambar 3.19.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Apakah anda mengemil diantara jam makan berat/ utama anda?

Tidak pernah
Kadang-kadang
Sering
Selalu

Seberapa sering anda minum minuman beralkohol?

Tidak pernah
Kadang-kadang
Sering
Selalu

Gambar 3.19. Opsi untuk Pertanyaan 1 dan 7 pada Halaman 3

Field perokok dan *field* pemantauan jumlah kalori per hari terdiri dari 2 opsi jawaban seperti pada Gambar 3.20.

Apakah anda merokok?

Ya
Tidak

Apakah anda memantau kalori yang anda makan setiap harinya?

Ya
Tidak

Gambar 3.20. Opsi untuk Pertanyaan 2 dan 4 pada Halaman 3

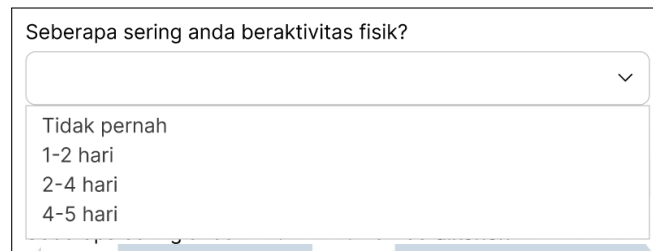
Field jumlah konsumsi air per hari terdiri dari 3 opsi jawaban seperti pada Gambar 3.21.

Berapa liter air yang anda minum setiap hari nya?

Kurang dari 1 liter
1-2 liter
Lebih dari 2 liter

Gambar 3.21. Opsi untuk Pertanyaan 3 pada Halaman 3

Field frekuensi beraktivitas fisik terdiri dari 4 opsi jawaban seperti pada Gambar 3.22.

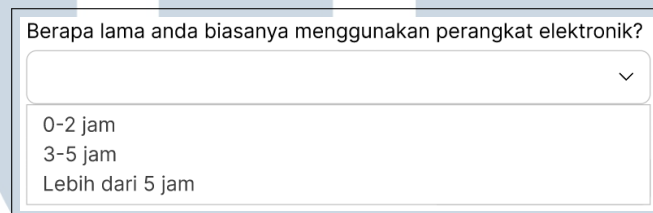


Seberapa sering anda beraktivitas fisik?

Tidak pernah
1-2 hari
2-4 hari
4-5 hari

Gambar 3.22. Opsi untuk Pertanyaan 5 pada Halaman 3

Field frekuensi menggunakan perangkat elektronik terdiri dari 3 opsi jawaban seperti pada Gambar 3.23.

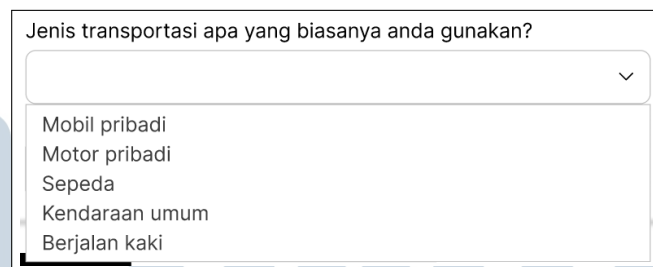


Berapa lama anda biasanya menggunakan perangkat elektronik?

0-2 jam
3-5 jam
Lebih dari 5 jam

Gambar 3.23. Opsi untuk Pertanyaan 6 pada Halaman 3

Field jenis transportasi yang biasanya digunakan terdiri dari 5 opsi jawaban seperti pada Gambar 3.24.



Jenis transportasi apa yang biasanya anda gunakan?

Mobil pribadi
Motor pribadi
Sepeda
Kendaraan umum
Berjalan kaki

Gambar 3.24. Opsi untuk Pertanyaan 8 pada Halaman 3

Pada halaman ini, terdapat validasi *input* untuk setiap *field* yang belum terisi saat ingin melakukan pengecekan tingkat obesitas. Gambar 3.25 menunjukkan pesan *error* yang muncul akibat validasi *input*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Prediksi Tingkat Obesitas dengan Algoritma Random Forest Classifier

Rutinitas

Apakah anda mengemil diantara jam makan berat/ utama anda?

Field tidak boleh kosong

Apakah anda merokok?

Field tidak boleh kosong

Berapa liter air yang anda minum setiap hari nya?

Field tidak boleh kosong

Apakah anda memantau kalori yang anda makan setiap harinya?

Field tidak boleh kosong

Seberapa sering anda beraktivitas fisik?

Field tidak boleh kosong

Berapa lama anda biasanya menggunakan perangkat elektronik?

Field tidak boleh kosong

Seberapa sering anda minum minuman beralkohol?

Field tidak boleh kosong

Jenis transportasi apa yang biasanya anda gunakan?

Field tidak boleh kosong

Gambar 3.25. Tampilan untuk Validasi *Input* Halaman 3

D Modalitas Hasil Klasifikasi

Modalitas hasil klasifikasi sendiri terbagi ke dalam 2 tipe. Tipe pertama adalah modalitas *error* yang akan ditampilkan ketika terdapat *field* yang masih belum terisi seperti yang dapat dilihat pada Gambar 3.26.

U I V E R S I T A S
 M U L T I M E D I A
 N U S A N T A R A

Prediksi Tingkat Obesitas dengan Algoritma Random Forest Classifier

Error

Terdapat field yang masih kosong, silahkan diisi terlebih dahulu.

Tutup

Berapa liter air yang anda minum setiap hari nya?

Apakah anda memantau kalori yang anda makan setiap harinya?

Seberapa sering anda beraktivitas fisik?

Berapa lama anda biasanya menggunakan perangkat elektronik?

Seberapa sering anda minum minuman beralkohol?

Jenis transportasi apa yang biasanya anda gunakan?

Prev

Cek Hasil

Gambar 3.26. Tampilan untuk Modalitas *Error*

Selanjutnya, tipe kedua adalah modalitas sukses yang akan ditampilkan hanya jika proses klasifikasi berhasil sampai aplikasi menerima hasil dari API. Pada modalitas ini, akan ditampilkan hasil klasifikasi dari inputan pengguna. Gambar 3.27 merepresentasikan modalitas sukses.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Prediksi Tingkat Obesitas dengan Algoritma Random Forest Classifier

Hasil Prediksi

Normal Weight

Tutup

Tidak

Berapa liter air yang anda minum setiap hari nya?

1-2 liter

Apakah anda memantau kalori yang anda makan setiap harinya?

Tidak

Seberapa sering anda beraktivitas fisik?

1-2 hari

Berapa lama anda biasanya menggunakan perangkat elektronik?

Lebih dari 5 jam

Seberapa sering anda minum minuman beralkohol?

Kadang-kadang

Jenis transportasi apa yang biasanya anda gunakan?

Mobil pribadi

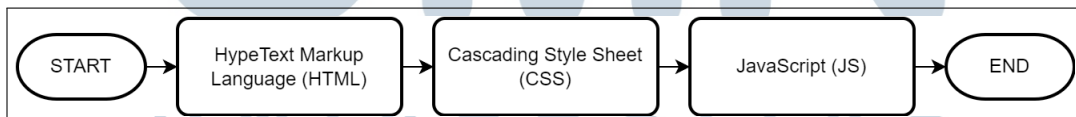
Prev

Cek Hasil

Gambar 3.27. Tampilan untuk Modalitas Sukses

3.8 Implementasi Tampilan Antarmuka Pengguna

Gambar 3.28 merepresentasikan *detail* tahapan untuk proses implementasi tampilan antarmuka pengguna.



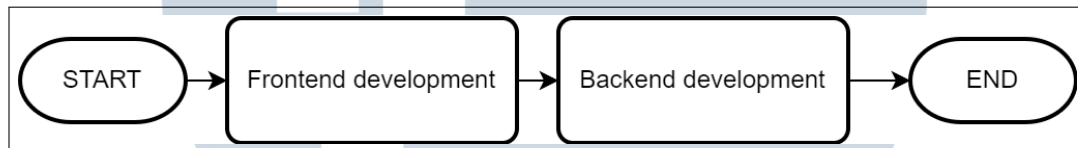
Gambar 3.28. *Detail* Tahapan untuk Proses Implementasi Tampilan Antarmuka Pengguna

Pada tahapan ini, gambaran tampilan dari proses *prototyping* akan diimplementasikan ke dalam sistem. Sistem ini akan berbentuk seperti sebuah *online form* yang berisikan sejumlah pertanyaan yang telah disesuaikan dengan fitur-fitur terbaik yang didapatkan dari hasil *feature selection*. Pengimplementasian ini akan dilakukan dengan penyusunan penyusunan elemen-elemen sistem terlebih dahulu menggunakan bahasa pemrograman HyperText Markup Language (HTML).

Selanjutnya, tampilan dari sistem akan diperindah menggunakan Cascading Style Sheet (CSS) terkait pengaturan warna, tulisan, dan *layout*. Tahapan terakhirnya adalah penggunaan JavaScript (JS) untuk mengatur logika tampilan dan menambahkan interaksi antara pengguna dengan sistem, seperti tombol navigasi dan validasi *input form*.

3.9 Implementasi Kode untuk Sistem

Gambar 3.29 merepresentasikan *detail* tahapan untuk proses implementasi kode untuk sistem.



Gambar 3.29. *Detail* Tahapan untuk Proses Implementasi Kode untuk Sistem

Pada tahapan ini, akan dibuat logika-logika pemrograman sistem dari sisi *frontend* maupun *backend* untuk dapat memberikan hasil klasifikasi. Proses ini dimulai dengan pembuatan kodingan dari sistem *frontend* untuk menerima inputan lalu mengirimkannya kepada API untuk melakukan klasifikasi. Selanjutnya, akan dilakukan penyimpanan model yang telah dibangun dengan bantuan modul "Pickle" yang akan dibuat menggunakan bahasa pemrograman Flask dari Python. Model ini nantinya akan tersimpan ke dalam sebuah file berekstensi ".pkl" yang dapat langsung dipanggil ketika aplikasi ingin melakukan klasifikasi. Dengan demikian, waktu yang diperlukan aplikasi untuk memberikan hasil prediksi pun akan lebih efisien karena sistem dapat langsung melakukan prediksi menggunakan model yang telah disimpan tanpa harus melatih ulang ataupun memuat model dari awal setiap kali melakukan prediksi. Setelah model tersimpan, langkah selanjutnya adalah pembuatan sistem dari sisi *backend* yang berfungsi untuk mengatur pengelolaan data untuk melakukan prediksi, *routing* untuk menangani permintaan dari *frontend*, melakukan prediksi dengan model yang telah disimpan, serta mengembalikan hasil prediksi.