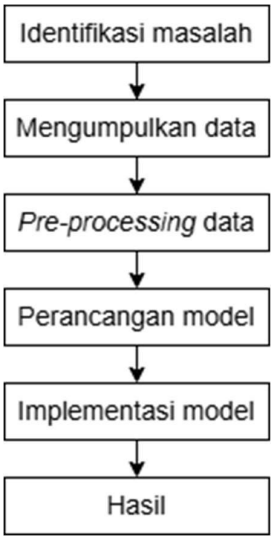
BAB III

METODOLOGI PENELITIAN

3.1 Desain Penelitian

Desain penelitian ini menggunakan pendekatan eksperimen untuk menguji keakuratan sistem prediksi kalori makanan dengan metode *Single Shot MultiBox Detector* (SSD). Penelitian dilakukan dengan membangun model deteksi objek, mengumpulkan dataset citra makanan, dan menerapkan SSD untuk mendeteksi serta mengenali jenis makanan secara otomatis, guna memperkirakan kalori dalam setiap porsi.



Gambar 3. 1 Desain Penelitian
(Sumber: Data Penelitian, 2025)

Berikut adalah penjelasan Langkah-langkah yang dilakukan dalam penelitian:

1. Identifikasi Masalah

Mengidentifikasi masalah terkait kelebihan kalori yang dapat menyebabkan gangguan kesehatan, dimana banyak orang kesulitan mengelola asupan kalori

karena kurangnya informasi. Sistem prediksi kalori berbasis citra digital dapat membantu memberikan estimasi yang lebih akurat dan memudahkan pengguna dalam mengontrol kalori mereka.

2. Mengumpulkan Data

Pengumpulan data citra makanan melibatkan pengambilan foto makanan menggunakan kamera digital atau ponsel cerdas, penggunaan katalog makanan yang berisi gambar dari berbagai sumber, pengumpulan data online dari situs *web* atau aplikasi terkait makanan. Dalam proses pengumpulan data sendiri, terdapat dua tahapan yang terbagi menjadi pengumpulan data mengenai kalori dan pengumpulan data citra

3. *Pre-processing* Data

Pada tahap ini, dilakukan pengolahan data citra makanan, termasuk anotasi, sebelum digunakan untuk melatih model. Hasil dari tahap ini adalah gambar yang telah diproses untuk memastikan data yang digunakan dalam model SSD memiliki format yang tepat dan kualitas yang optimal, sehingga dapat meningkatkan akurasi dalam prediksi kalori.

4. Perancangan Model

Pembuatan model SSD (*Single Shot MultiBox Detector*) untuk memprediksi dua hal utama untuk setiap objek yang terdeteksi dalam gambar: pertama, kelas objek, yang merujuk pada jenis atau kategori objek yang ada dalam gambar, seperti jenis makanan dan kedua, lokasi objek, yang menunjukkan posisi objek tersebut dalam gambar, yang diwakili dalam bentuk *bounding box*.

5. Implementasi Model

Langkah di mana model yang telah dilatih diterapkan untuk menyelesaikan masalah di dunia nyata. Pada tahap ini, model SSD yang telah dipelajari untuk mendeteksi objek misalnya, jenis makanan dan memprediksi lokasi objek (*bounding box*) dalam gambar akan diintegrasikan ke dalam sistem yang dapat digunakan dalam aplikasi nyata.

6. Hasil

Pada tahap ini, penulis menyimpulkan bahwa model SSD yang dikembangkan efektif dalam mendeteksi dan mengenali objek makanan dengan akurasi yang memadai untuk prediksi kalori. Sistem ini mampu memberikan estimasi kalori secara otomatis, mempermudah pengguna dalam mengelola asupan kalori harian. Selain itu, implementasi berbasis web memberikan aksesibilitas yang tinggi, memungkinkan pengguna mengupload gambar makanan dan mendapatkan estimasi kalori dengan cepat. Penelitian ini berkontribusi dalam mengintegrasikan teknologi pengolahan citra digital dengan kebutuhan masyarakat untuk mendukung pengelolaan kalori dan peningkatan kesehatan.

3.2 Metode Pengumpulan Data

Dalam penelitian ini, pengumpulan data dilakukan melalui beberapa teknik untuk memastikan data yang diperoleh memiliki keberagaman dan relevansi yang tinggi terhadap tujuan penelitian. Teknik-teknik yang digunakan meliputi:

1. Observasi Tidak Langsung

Teknik ini dilakukan dengan mengumpulkan data citra makanan dari media sosial dan sumber internet lainnya. Data yang dikumpulkan dipilih berdasarkan relevansi dengan objek penelitian untuk memastikan kualitas serta keberagamannya. Dalam proses pengumpulan data gambar, penulis memanfaatkan teknologi *search engine* dan media sosial. Rincian hasil pengumpulan data dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Daftar Gambar

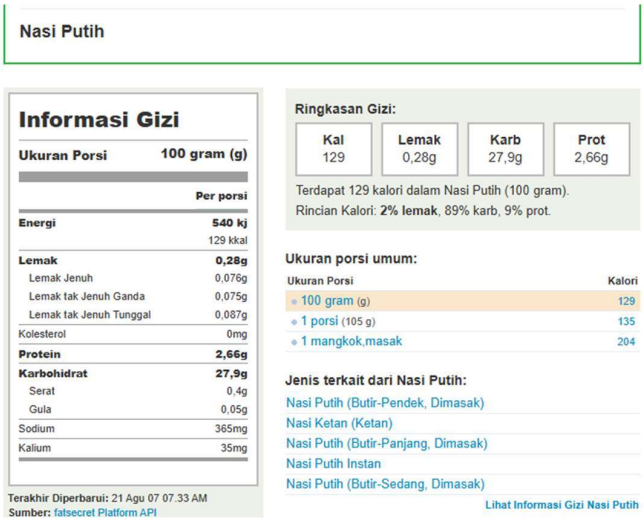
Nama objek	Jumlah Gambar
Nasi putih	99
Tempe goreng tanpa tepung	99
Tahu goreng tanpa tepung	68
Tumis kangkung	96
Telur goreng	99
Ayam Goreng	50
Kerupuk putih	54
Kentang goreng	102
Nasi goreng	99
Rendang	61
Terong balado	50

(Sumber: Data Penelitian, 2025)

2. Wawancara

Teknik ini dilakukan dengan mewawancarai ahli gizi untuk memverifikasi data kalori yang diperoleh dari FatSecret Indonesia. Informasi yang diperoleh dari ahli gizi digunakan untuk memastikan bahwa estimasi kalori pada berbagai jenis makanan yang tercantum di FatSecret Indonesia akurat dan sesuai dengan standar kebutuhan kalori harian. Data yang telah diverifikasi kemudian digunakan dalam pengembangan sistem prediksi kalori.

Data kalori yang digunakan dalam penelitian ini didasarkan pada asumsi porsi makanan khas Indonesia. Oleh karena itu, penelitian ini merujuk pada data yang disediakan oleh FatSecret Indonesia**.** Informasi mengenai kalori tersebut dapat diakses melalui halaman website www.fatsecret.co.id. Tampilan informasi yang tersedia di website FatSecret Indonesia ditunjukkan pada gambar berikut.



Gambar 3. 2 *Fat Secret* Indonesia
(Sumber: www.fatsecret.co.id)

Dengan demikian, diperoleh data kalori makanan yang sesuai dengan jenis makanan yang telah ditentukan, yang kemudian ditampilkan pada Tabel 3.2.

Tabel 3. 2 Data Kalori

Nama makanan	Porsi	Estimasi kalori	Lemak	Karbohidrat	Protein
Nasi putih	100g	130 kkal	0,21 g	28,59 g	2,38 g
Tempe goreng tanpa tepung	85g	192 kkal	12,93 g	10,15 g	11,31 g
Tahu goreng tanpa tepung	100g	271 kkal	20,18 g	10,49 g	17,19 g
Tumis kangkung	100g	98 kkal	8,71 g	3,99 g	2,56 g

Lanjutan Tabel 3.2

Telur goreng	46g	89 kkal	6,76 g	0,43 g	6,24 g
Ayam goreng	49g	119 kkal	6,79 g	0 g	13,6 g
Kerupuk putih	45g	194 kkal	6,32 g	31,19 g	2,48 g
Kentang goreng	100g	274 kkal	14,06 g	35,66 g	3,48 g
Nasi goreng	100g	168 kkal	6,23 g	21,06 g	6,3 g
Rendang	75g	200 kkal	14 g	11 g	7 g
Terong balado	100g	97 kkal	7,99 g	7,09 g	1,16 g

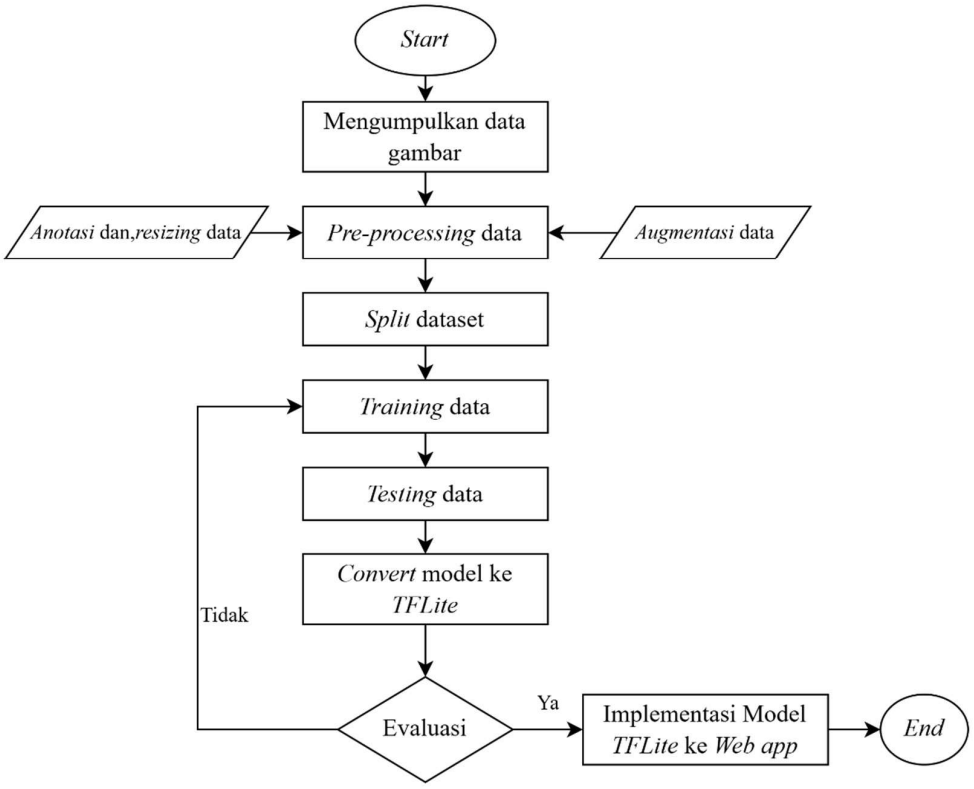
(Sumber: Data Penelitian, 2025)

3. Studi Literatur

Teknik ini dilakukan dengan mengkaji berbagai literatur, jurnal, dan penelitian terdahulu yang relevan. Penulis mempelajari metode pengolahan citra digital, algoritma SSD, serta penelitian terkait prediksi kalori pada makanan untuk memperkuat dasar teori dan metodologi penelitian.

3.3 Perancangan SSD

Perancangan model SSD untuk prediksi kalori pada makanan dimulai dengan Pengumpulan Citra Makanan, Pengolahan dan Persiapan Data, Anotasi serta Pelabelan Gambar, Desain Model SSD, Pelatihan Model menggunakan Data Latih, Pengujian Model dengan Data Uji, Evaluasi Model, Interpretasi Hasil, dan diakhiri dengan penyelesaian desain sebagai berikut:



Gambar 3. 3 Perancangan SSD
(Sumber: Data Penelitian, 2025)

3.3.1 Mengumpulkan Data Gambar

Mengumpulkan gambar berbagai jenis makanan dengan label yang sesuai, seperti nasi putih, tempe goreng, tahu goreng, tumis kangkung, telur goreng, Ayam goreng (tanpa pelapis), kerupuk putih, kentang goreng, nasi goreng, dan rendang. Gambar-gambar ini diperoleh dari sumber internet dan media sosial, seperti Instagram.

3.3.2 Pre-processing Data

Tahap ini bertujuan untuk mempersiapkan data gambar agar siap digunakan dalam pelatihan model SSD. Proses *pre-processing* sangat penting untuk memastikan data yang digunakan memiliki format, kualitas, dan keragaman yang

memadai, sehingga model dapat dilatih dengan lebih efektif. Tahapan dalam *pre-processing* meliputi:

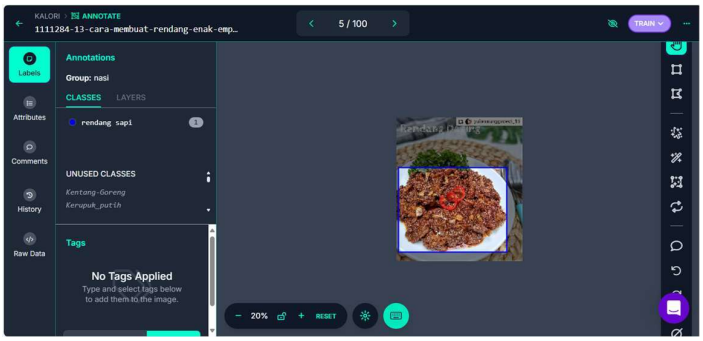
Resizing

Mengubah ukuran gambar menjadi 600x600 piksel agar seragam sesuai spesifikasi model SSD, tanpa mengurangi proporsi atau informasi penting. Gambar juga dikonversi ke format JPG untuk standarisasi dan efisiensi ukuran file. Untuk mempercepat proses ini, digunakan skrip Python yang otomatis melakukan konversi dan pengubahan ukuran gambar secara efisien.

Anotasi

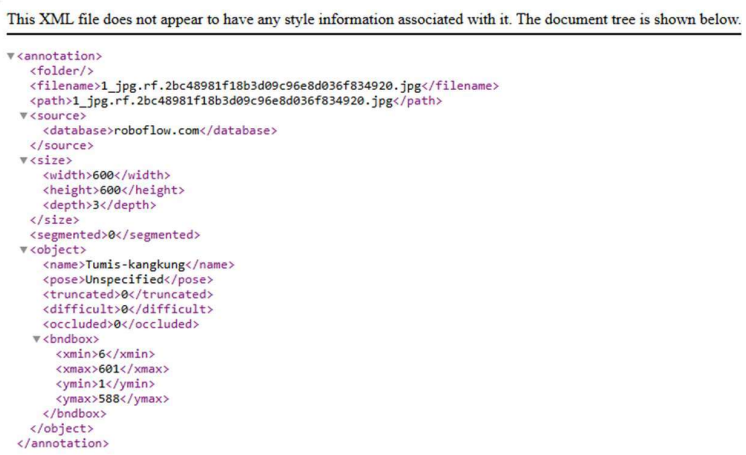
Setelah gambar dari sepuluh jenis makanan (kelas) berhasil dikumpulkan dan format file disesuaikan, langkah berikutnya adalah melakukan pelabelan atau anotasi gambar. Proses anotasi ini bertujuan untuk memberikan label pada tiap jenis makanan dengan *bounding box*. Dalam penelitian ini, anotasi dilakukan menggunakan platform Roboflow, dengan output dalam format PASCAL VOC.

Roboflow memudahkan dalam menandai objek-objek yang ada pada gambar, seperti yang ditunjukkan pada Gambar 3.4.



Gambar 3. 4 *Anotasi* Dengan Roboflow
(Sumber: Data Penelitian,2025)

Setelah proses anotasi selesai, langkah selanjutnya adalah menyimpan hasil anotasi dalam format Pascal VOC, yang merupakan format standar untuk menyimpan data anotasi objek dalam gambar. Format ini memungkinkan untuk menyimpan informasi mengenai koordinat bounding box serta label objek yang terdeteksi, seperti yang ditunjukkan pada Gambar di bawah.



Gambar 3. 5 Hasil Anotasi Disimpan Dalam Format Pascal VOC
(Sumber: Data Penelitian, 2025)

Format PASCAL VOC menghasilkan file XML untuk setiap gambar yang dianotasi. File XML tersebut memuat informasi penting, seperti:

1. Nama kelas: Jenis objek yang dianotasi (contoh: "nasi", "ayam goreng").
2. Koordinat bounding box: Lokasi *bounding box* dalam bentuk koordinat (*xmin, ymin, xmax, ymax*).
3. Dimensi gambar: Lebar dan tinggi gambar yang dianotasi.

File XML ini kompatibel dengan banyak *framework deep learning* popular salah satunya TensorFlow, sehingga dapat langsung digunakan dalam proses pelatihan model deteksi objek. Dengan menggunakan format ini, model dapat

mempelajari lokasi dan klasifikasi objek secara akurat dari dataset yang telah dianotasi.

Augmentasi Data

Gambar yang sudah dikumpulkan dan sudah dianotasi bounding box untuk tiap kelasnya akan diaugmentasi untuk memperbanyak variasi data gambar, tujuannya agar model bisa lebih baik lagi untuk mengenali kelas nantinya dalam berbagai kondisi. Pada tahapan ini, akan menggunakan beberapa library berbahasa *Python* seperti os, random, *xml.etree.ElementTree* (ET), PIL (*Python Imaging Library*), *numpy*, dan *scipy.ndimage.*

Library ini mempermudah tahapan augmentasi dengan berbagai operasi seperti pengubahan ukuran gambar, peningkatan kontras, penambahan *noise*, dan transformasi lainnya. Selain itu, PIL mempermudah penerapan filter pada gambar, sedangkan *xml.etree.ElementTree* digunakan untuk menyesuaikan anotasi bounding-box dari gambar sebelumnya dengan gambar yang sudah diaugmentasi. Berikut adalah daftar teknik augmentasi yang digunakan dalam kode ini beserta parameter atau nilai yang digunakan:

Tabel 3. 3 Teknik Augmentasi

No.	Teknik *Augmentasi*	Parameter/Value	Deskripsi
1	*Gamma Contrast*	Gamma: 0,5	Mengatur tingkat kontras gambar dengan mengurangi kontras.
2	*Average Blur*	*Kernel size*: 3	Mengaburkan gambar menggunakan *BoxBlur* dengan ukuran kernel 3.

Lanjutan Tabel 3.3

3	*Horizontal Flip*	1	Membalik gambar secara horizontal; bounding box disesuaikan dengan membalik posisi xmin dan xmax.
4	*Brightness Adjustment*	Faktor kecerahan: 1,1	Meningkatkan kecerahan gambar sebesar 10%.
5	*Perspective Transformation*	(*Placeholder*)	Belum ada logika khusus yang diterapkan untuk perspektif transformasi.
6	*Zoom*	*Zoom factor*: 1.1	Memperbesar gambar sebesar 10%; *bounding box* disesuaikan dengan skala zoom.
7	*Additive Poisson Noise*	Intensitas *noise*: 15	Menambahkan *noise Poisson* pada gambar untuk membuat variasi warna piksel secara acak.
8	*Affine Transformation*	Skala: [0.6, 0.9]	Mengubah ukuran gambar menjadi 60% dan 90% dari ukuran aslinya; *bounding box* disesuaikan.
9	*Vertical Flip*	1	Membalik gambar secara *vertikal*; *bounding box* disesuaikan dengan membalik posisi *ymin* dan *ymax*.
10	*Multiply Pixel (Contrast + Brightness)*	Kontras: 0,5, Kecerahan: 0,7	Mengatur kontras dan kecerahan secara bersamaan untuk menambah variasi.
11	Rotasi 180 Derajat	Sudut *rotasi*: 180°	Memutar gambar 180°, *bounding box* disesuaikan untuk rotasi penuh.
12	Rotasi 90 Derajat	Sudut *rotasi*: 90°	Memutar gambar 90°, *bounding box* disesuaikan untuk rotasi perempat putaran.

(Sumber: Data Penelitian, 2025)

Setiap gambar akan melalui proses augmentasi yang menghasilkan variasi baru dari dataset, seperti yang ditunjukkan pada gambar di bawah, yang memperlihatkan contoh hasil dari proses augmentasi

Gambar 3. 6 Contoh Hasil Augmentasi
(Sumber: Data Penelitian, 2025)

Jumlah total data gambar yang telah dikumpulkan, termasuk hasil augmentasi, adalah 12.278 gambar. Dataset ini mencakup gambar asli yang telah diproses dan variasi baru yang dihasilkan melalui teknik augmentasi, yang memungkinkan peningkatan jumlah data untuk keperluan pelatihan model.

Tabel 3. 4 Jumlah Dataset

Nama objek	Jumlah Gambar
Nasi putih	1386
Tempe goreng	1386
Tahu goreng	952
Tumis kangkung	1344
Telur goreng	1386
Ayam goreng	700
Kerupuk putih	756
Kentang goreng	1428
Nasi goreng	1386
Rendang	854
Terong balado	700

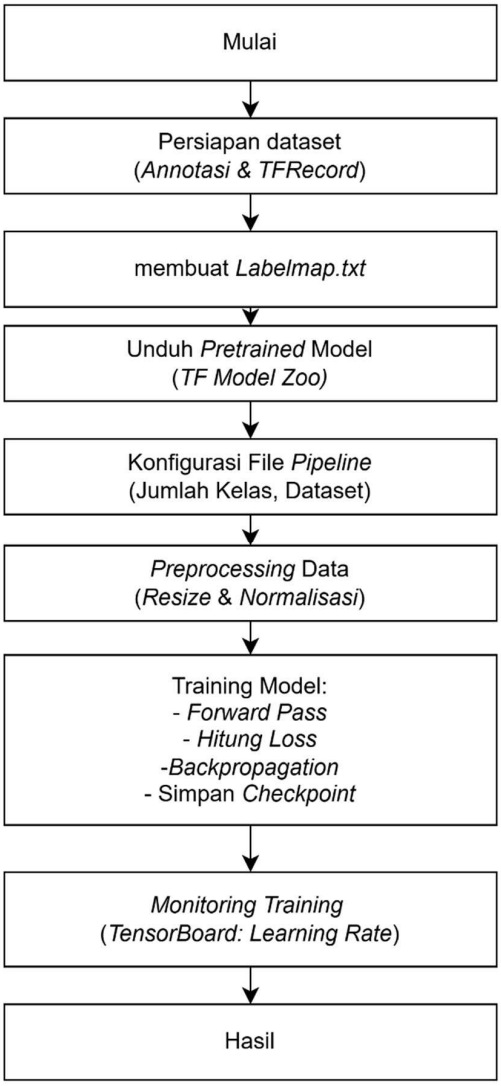
(Sumber: Data Penelitian, 2025)

3.3.3 *Split* Dataset

Setelah seluruh gambar diaugmentasi, tahapan selanjutnya adalah melakukan pembagian dataset. Dataset gambar yang telah dikumpulkan dan diaugmentasi akan dibagi menjadi 3 bagian: Train set, Test set dan validation set. Pembagian dataset ini dilakukan dengan menggunakan beberapa library berbahasa *Python* seperti os, random, dan *shutil*. *Library-library* ini memungkinkan proses pemindahan file gambar dan file XML (anotasi *bounding-box*) ke folder Train, Validation, dan Test dengan cara yang efisien dan terstruktur. Total gambar setelah augmentasi akan dibagi menjadi dataset dengan perbandingan 80:10:10, yaitu 80% untuk data latih (train), 10% untuk data validasi (validation), dan 10% untuk data uji (test). Proses pembagian dataset ini akan dilakukan menggunakan skrip Python yang dijalankan melalui *Google Colab* Sehingga, data menjadi 9822 untuk train, 1228 untuk test, dan 1228 untuk validation.

3.3.4 Training

Alur pelatihan model dijelaskan pada flowchart pada Gambar 3.7 di bawah ini.

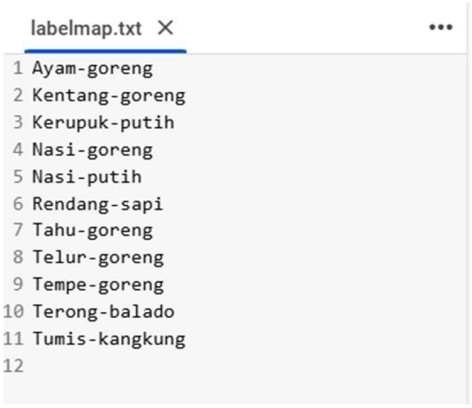


Gambar 3. 7 Alur *Training*
(Sumber: Data Penelitian, 2025)

Arsitektur yang digunakan dalam penelitian ini adalah *SSD-MobileNetV2 FPNLite*, yang diimplementasikan berdasarkan konfigurasi *ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8*. *MobileNetV2* digunakan hingga tahapan tertentu untuk mengekstraksi ciri-ciri utama dari gambar, sedangkan fitur *FPNLite* berfungsi untuk memperkuat kemampuan deteksi objek pada skala yang berbeda. Setiap *feature map* yang dihasilkan oleh *backbone* ini

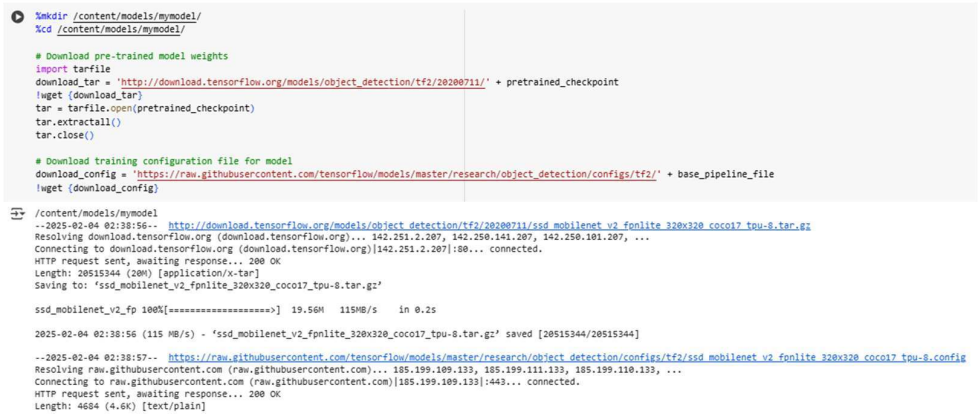
dihubungkan langsung ke layer deteksi terakhir, memungkinkan model mengenali dan melokalisasi objek secara efisien.

Proses dimulai dengan persiapan dataset, di mana data anotasi dikonversi ke format *TFRecord* agar dapat digunakan dalam pelatihan. Setelah itu, membuat file Bernama labelmap dengan format txt yang berisi dafar kelas .



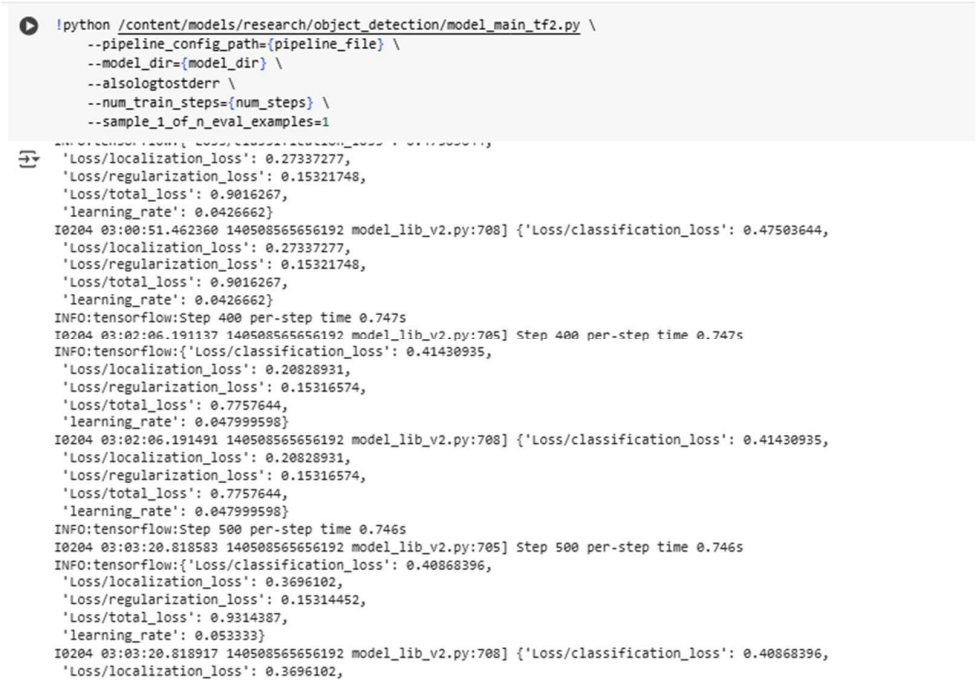
Gambar 3. 7 File labelmap
(Sumber: Data Penelitian, 2025)

Selanjutnya dilakukan pengunduhan model *pretrained* dari TensorFlow Model Zoo untuk *transfer learning*, sehingga model tidak perlu dilatih dari awal.



Gambar 3. 8 Proses Unduh Model *pretrained*
(Sumber: Data Penelitian, 2025)

Selanjutnya, dilakukan konfigurasi file *pipeline*, termasuk pengaturan jumlah kelas dan dataset yang digunakan. Data kemudian diproses melalui *preprocessing*, seperti resize dan *normalisasi*, untuk memastikan konsistensi input model.



Gambar 3. 9 Proses Training
(Sumber: Data Penelitian, 2025)

Pada tahap training, model melakukan forward pass untuk menghasilkan prediksi, kemudian menghitung *loss*, melakukan *backpropagation* untuk memperbarui bobot, dan menyimpan *checkpoint* untuk menjaga progress pelatihan. Proses pelatihan ini dimonitor menggunakan *TensorBoard*, yang memberikan visualisasi terhadap *learning rate, loss*, dan lainnya. Setelah semua tahap selesai, model siap digunakan untuk inferensi dan menghasilkan output yang diinginkan.

3.3.5 Testing

Pada tahap ini, model SSD yang telah dilatih digunakan untuk mendeteksi objek dalam data uji. Model akan menghasilkan beberapa output utama, yaitu *bounding box*, yang menunjukkan lokasi objek yang terdeteksi dalam gambar, label kelas, yang mengidentifikasi kategori objek, serta *confidence score*, yang menunjukkan tingkat keyakinan model terhadap setiap prediksi. Setelah hasil prediksi diperoleh, dilakukan proses *Non-Maximum Suppression* (NMS) untuk mengeliminasi deteksi duplikat, sehingga hanya bounding box dengan skor tertinggi yang dipertahankan. Langkah ini bertujuan untuk meningkatkan akurasi deteksi dengan memastikan bahwa setiap objek hanya memiliki satu bounding box yang mewakili hasil prediksi terbaik.

3.3.6 Convert Model

Setelah model dinilai memiliki performa yang cukup baik berdasarkan monitoring tensorboard, langkah selanjutnya adalah mengekspor model ke dalam bentuk *TFLite Graph* menggunakan skrip *export_tflite_graph_tf2.py.* Proses ini memanfaatkan API *TensorFlow Object Detection* untuk mengonversi model yang telah dilatih ke format *frozen graph*.

Hasilnya berupa beberapa file, termasuk *tflite_graph.pb* dan folder saved_model/, yang akan digunakan untuk proses konversi lebih lanjut. File-file tersebut kemudian dikonversi ke dalam format *TensorFlow Lite* (*TFLite*), menghasilkan file berekstensi *.tflite*. File ini dirancang agar dapat dijalankan secara efisien di perangkat mobile atau sistem dengan sumber daya terbatas, seperti *IoT* atau *edge devices*.

Dengan menggunakan API dari TensorFlow, proses ini menjadi lebih terstruktur dan sederhana, sehingga memudahkan transisi model dari tahap pelatihan hingga siap digunakan dalam aplikasi mobile atau perangkat lainnya.

3.3.7 Evaluasi

Setelah model melakukan prediksi pada data uji, langkah selanjutnya adalah proses evaluasi dengan membandingkan hasil prediksi model terhadap ground truth atau label asli dari dataset. Pada tahap ini, dilakukan perhitungan metrik untuk mengukur performa model. Salah satu metrik utama yang digunakan adalah *confusion matrix*, yang membantu mengidentifikasi jumlah *True Positive* (TP), yaitu objek yang terdeteksi dengan benar, *False Positive* (FP), yaitu objek yang salah terdeteksi atau diklasifikasikan, serta *False Negative* (FN), yaitu objek yang ada tetapi gagal dideteksi oleh model. Dari hasil confusion matrix ini, selanjutnya dihitung metrik evaluasi utama seperti *Precision*, yang menunjukkan tingkat akurasi prediksi model terhadap semua deteksi yang dilakukan, *Recall*, yang mengukur sejauh mana model berhasil mendeteksi objek yang benar-benar ada, serta *F1-Score*, yang merupakan rata-rata harmonik antara *Precision* dan *Recall*. Evaluasi ini penting untuk memahami keandalan model dalam mendeteksi objek dengan tingkat akurasi yang optimal.

Tabel 3. 5 Format Tabel Confusion Matrix Yang Digunakan

		True Class										
		A	B	C	D	E	F	G	H	I	J	K
Predicted class	A	TP_1	FP_{21}	FP_{31}	FP_{41}	FP_{51}	FP_{61}	FP_{71}	FP_{81}	FP_{91}	FP_{101}	FP_{111}
	B	FP_{12}	TP_2	FP_{32}	FP_{42}	FP_{52}	FP_{62}	FP_{72}	FP_{82}	FP_{92}	FP_{102}	FP_{112}
	C	FP_{13}	FP_{23}	TP_3	FP_{43}	FP_{53}	FP_{63}	FP_{73}	FP_{83}	FP_{93}	FP_{103}	FP_{113}
	D	FP_{14}	FP_{24}	FP_{34}	TP_4	FP_{54}	FP_{64}	FP_{74}	FP_{84}	FP_{94}	FP_{104}	FP_{114}
	E	FP_{15}	FP_{25}	FP_{35}	FP_{45}	TP_5	FP_{65}	FP_{75}	FP_{85}	FP_{95}	FP_{105}	FP_{115}
	F	FP_{16}	FP_{26}	FP_{36}	FP_{46}	FP_{56}	TP_6	FP_{76}	FP_{86}	FP_{96}	FP_{106}	FP_{116}
	G	FP_{17}	FP_{27}	FP_{37}	FP_{47}	FP_{57}	FP_{67}	TP_7	FP_{87}	FP_{97}	FP_{107}	FP_{117}
	H	FP_{18}	FP_{28}	FP_{38}	FP_{48}	FP_{58}	FP_{68}	FP_{78}	TP_8	FP_{98}	FP_{108}	FP_{118}
	I	FP_{19}	FP_{29}	FP_{39}	FP_{49}	FP_{59}	FP_{69}	FP_{79}	FP_{89}	TP_9	FP_{109}	FP_{119}
	J	FP_{110}	FP_{210}	FP_{310}	FP_{410}	FP_{510}	FP_{610}	FP_{710}	FP_{810}	FP_{910}	TP_{10}	FP_{1110}
	K	FP_{111}	FP_{211}	FP_{311}	FP_{411}	FP_{511}	FP_{611}	FP_{711}	FP_{811}	FP_{911}	FP_{1011}	TP_{11}
FN		FN_A	FN_B	FN_C	FN_D	FN_E	FN_F	FN_G	FN_H	FN_I	FN_J	FN_K

(Sumber: Data Penelitian, 2025)

Berikut ini adalah keterangan kelas dari Tabel 3.5 Format Tabel *Confusion Matrix* Yang Digunakan di atas:

Tabel 3. 6 Keterangan Tabel

Abjad	Nama Kelas
A	Nasi putih
B	Tempe goreng
C	Tahu goreng
D	Tumis kangkung

Lanjutan Tabel 3.6

E	Telur goreng
F	Ayam goreng
G	Kerupuk putih
H	Kentang goreng
I	Nasi goreng
J	Rendang
K	Terong balado

(Sumber: Data Penelitian, 2025)

Tabel 3.5 *confusion matrix* ini berfungsi untuk menilai kinerja model saat diterapkan dalam aplikasi *mobile*. Seperti yang ditampilkan pada tabel terdapat beberapa kolom, yaitu TP1, TP2, TP3, TP4, TP5, TP6, TP7, TP8, TP9, TP10, dan TP11 yang masing-masing akan diisi dengan jumlah data *true positive*. Misalnya, pada kolom TP1, nilai yang dimasukkan merupakan jumlah gambar nasi putih (*true object*) yang berhasil diprediksi oleh model sebagai nasi putih (*prediction*), yang termasuk dalam kategori true *positive*. Selain data *true positive*, terdapat pula kolom lain yang tergolong sebagai *false*, seperti kolom FP14, yang berisi jumlah gambar nasi putih (true object) yang diprediksi oleh model sebagai tumis kangkung (*prediction*), Sebagai contoh FN (*false negative*), terdapat kasus di mana objek nasi putih tidak berhasil terdeteksi oleh model.

Dalam menilai performa model, penelitian ini akan menggunakan beberapa metrik *evaluasi*, seperti *Presisi*, *Recall*, dan *F1-Score* yang dijabarkan dalam format Tabel 3.7 untuk setiap kelas yang dikenali oleh model. Selain itu, perhitungan *Accuracy*, *Macro Average*, dan Weighted Average juga dilakukan untuk memberikan gambaran menyeluruh mengenai efektivitas model.

Tabel 3. 7 Format Tabel performa model

Kelas	*Presisi*	*Recall*	*F1-Score*	*Support*
Nasi putih				
Tempe goreng tanpa tepung				
Tahu goreng tanpa tepung				
Tumis kangkung				
Telur goreng				
Ayam goreng				
Kerupuk putih				
Kentang goreng				
Nasi goreng				
Rendang				
Terong balado				
Macro Average				
Weighted Average				

(Sumber: Data Penelitian, 2025)

Presisi

Presisi adalah metrik yang digunakan untuk menilai sejauh mana model mampu memprediksi suatu objek dengan benar dibandingkan dengan total prediksi yang dibuat untuk objek tersebut. Sebagai ilustrasi, jika model mengidentifikasi sepuluh gambar sebagai ayam goreng, tetapi hanya lima di antaranya yang benar-benar ayam goreng sesuai dengan data sebenarnya, maka presisi model untuk objek tersebut adalah 50%. Dalam penelitian ini, perhitungan presisi ayam goreng yang akan dilakukan menggunakan Rumus 2.1 yang nilainya berasal dari Tabel 3.5, sebagaimana ditampilkan hasilnya ditampilkan dibawah ini.

$$Presisi_A = \frac{TP_1}{TP_1 + FP_{12} + FP_{13} + FP_{14} + FP_{15} + FP_{16} + FP_{17} + FP_{18} + FP_{19} + FP_{110} + FP_{11}}$$

Recall

Recall mengukur seberapa banyak objek yang benar-benar ada yang berhasil dideteksi. Dalam penelitian ini, perhitungan presisi nasi putih yang akan dilakukan menggunakan Rumus 2.2 yang nilainya berasal dari Tabel 3.5, sebagaimana ditampilkan hasilnya ditampilkan dibawah ini.

$$Recall_A = \frac{TP_1}{TP_1 + FN_A}$$

F1-Score

F1-Score tetap dihitung sebagai rata-rata harmonik dari Precision dan Recall, Dalam penelitian ini, perhitungan presisi ayam goreng yang akan dilakukan menggunakan Rumus 2.3 yang nilainya berasal $Presision_A$ dan $Recall_A$:

$$F1 - Score = 2\,X\,\frac{Precision.Recall}{Precision + Recall}$$

Macro Average

Macro Average menghitung rata-rata *Precision, Recall, dan F1-Score* dengan memberi bobot yang sama ke setiap kelas, tanpa mempertimbangkan jumlah sampel di setiap kelas.

Weighted Average

Weighted Average menghitung rata-rata *Precision, Recall, dan F1-Score* dengan mempertimbangkan jumlah objek di setiap kelas.

$$Weighted\ Precision = \frac{(Support \times presisi)}{Total\ support}$$

$$Weighted\ recall = \frac{(Support \times recall)}{Total\ support}$$

$$Weighted\ F1 - Score = \frac{(Support \times F2 - Score)}{Total\ support}$$

3.3.8 Implementasi *TFLite* Ke *Web App*

Di tahap ini, untuk mengimplementasikan model *TensorFlow* Lite ke dalam aplikasi *web*, pertama-tama perlu membangun aplikasi menggunakan IDE (*Integrated Development Environment*) seperti Visual Studio Code atau IDE lain yang sesuai dengan pengembangan aplikasi *web*. Bahasa pemrograman yang digunakan untuk aplikasi *web* ini adalah JavaScript, python tampilannya menggunakan HTML dan juga CSS.

Hasil deteksi akan disimpan dalam file JSON yang mencakup informasi tentang klasifikasi objek dan koordinat bounding box. Berdasarkan hasil deteksi ini, nilai kalori makanan akan ditampilkan pada halaman *web*.

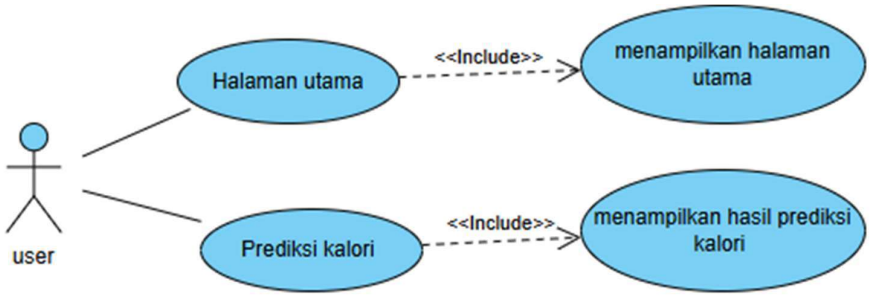
3.4 Design Interface

Dalam metode perancangan penelitian ini, algoritma SSD (*Single Shot MultiBox Detector*) digunakan sebagai pendekatan utama untuk prediksi kalori pada makanan dengan pengolahan citra digital. Untuk mempermudah proses perancangan, penelitian ini juga memanfaatkan UML (*Unified Modeling Language*) sebagai ilustrasi model perancangan. UML dipilih karena merupakan alat yang umum digunakan pada tahap awal pengembangan sistem, sehingga dengan menggunakan UML, penelitian ini dapat memiliki gambaran awal yang lebih terstruktur mengenai alur dan rancangan implementasi sistem.

3.4.1 UML *(Unified Modeling Language)*

1. *Use Case* Diagram

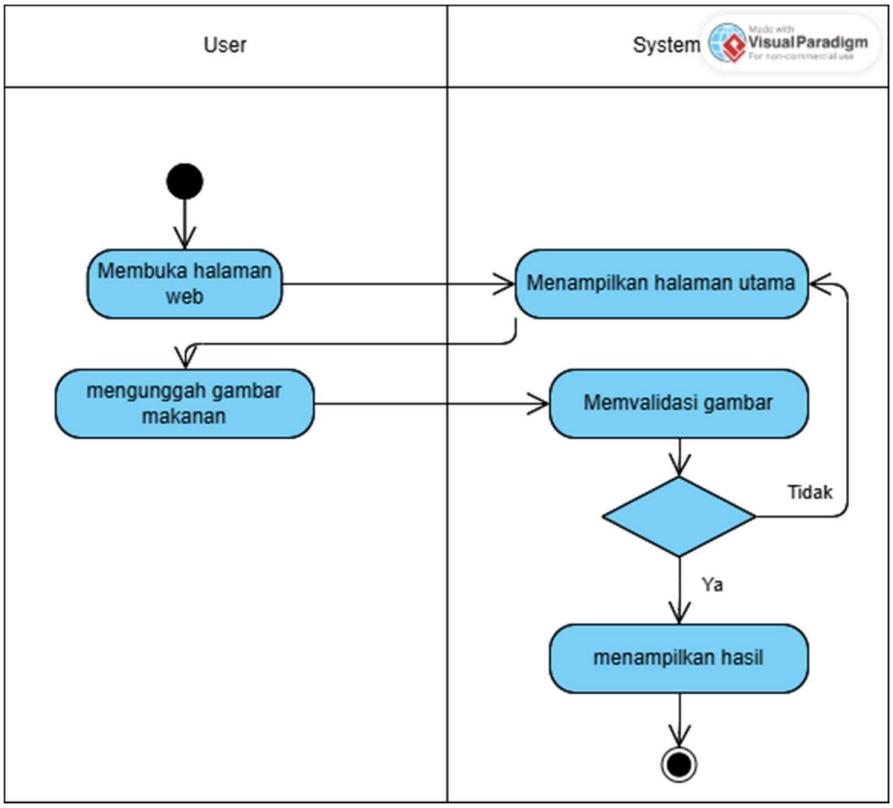
Use case diagram dapat diperjelas dengan merujuk pada gambar yang tersedia di bawah ini.



Gambar 3. 10 *Use case* diagram
(Sumber: Data Penelitian, 2025)

2. Activity Diagram

Activity Diagram Prediksi Kalori pada Makanan diawali pengguna membuka sistem yang sama, sistem kemungkinan akan menampilkan halaman utama yang langsung terdapat halaman untuk predksi kalori. Selanjutnya, pengguna dapat menambahkan gambar ke dalam system melalui tombol unggah gambar, dan sistem akan menampilkan halaman unggah gambar selanjutnya memulai proses input gambar, user dapat memulai proses prediksi dengan menekan tombol pediksi lalu system akan memproses gambar dan menyajikan hasil jika gambar yang dimasukkan benar adalah gambar makanan. Flowchart kegiatan berikut ini menggambarkan proses pengenalan prediksi makanan.

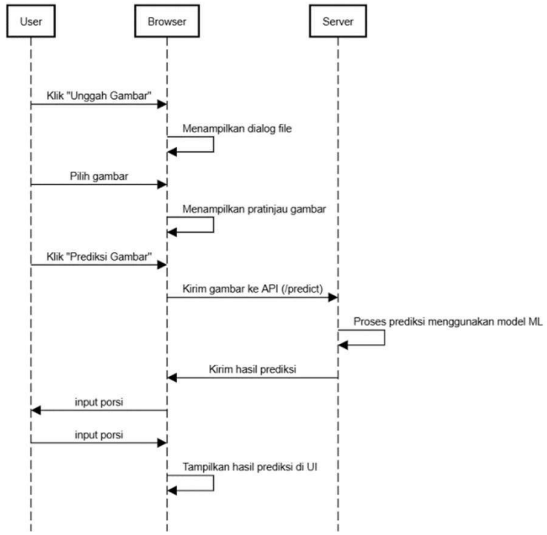


Gambar 3. 11 *Activity* diagram prediksi kalori pada makanan
(Sumber: Data Penelitian, 2025)

3. Sequence Diagram

Pengguna memulai proses dengan mengklik tombol "Unggah Gambar", yang memicu browser untuk menampilkan dialog pemilihan file. Setelah pengguna memilih gambar makanan, browser menampilkan pratinjau gambar yang telah diunggah. Selanjutnya, pengguna menekan tombol "Prediksi Gambar", sehingga browser mengirimkan gambar ke server melalui API (*/predict*). Server kemudian memproses gambar menggunakan model SSD untuk mengenali jenis makanan dan memperkirakan jumlah kalorinya. Setelah analisis selesai, server mengembalikan hasil prediksi ke *browser*. *Browser* kemudian meminta pengguna untuk memasukkan jumlah porsi makanan yang dikonsumsi. Setelah pengguna

menginput jumlah porsi, browser memperbarui informasi dan menampilkan hasil akhir prediksi kalori yang telah disesuaikan dengan jumlah porsi yang dimasukkan..



Gambar 3. 12 *Sequence* diagram
(Sumber: Data Penelitian, 2025)

3.4.2 Perancangan Antarmuka Pengguna

Tujuan dari Perancangan Antarmuka Pengguna adalah menciptakan antarmuka pengguna yang mudah digunakan untuk program perangkat lunak. Tujuan ini mencakup kemudahan penggunaan, kecepatan dalam mencapai hasil yang diinginkan, dan siap pakai. Fokus utamanya adalah memenuhi kebutuhan pengguna. Evaluasi pengguna terhadap kegunaan sistem sering kali terutama berdasarkan antarmuka pengguna, bukan hanya fungsionalitasnya. Ketika antarmuka pengguna dirancang dengan buruk, pengguna cenderung menghindari penggunaan perangkat lunak tersebut. Selain itu, antarmuka pengguna yang tidak memadai dapat menyebabkan masalah serius.

Gambar 3. 13 Perancangan Halaman Prediksi
(Sumber: Data Penelitian, 2025)

Keterangan:

1. Judul sistem
2. Gambar Makanan yang akan diprediksi
3. *Button*: unggah gambar dan prediksi
4. Table informasi hasil prediksi

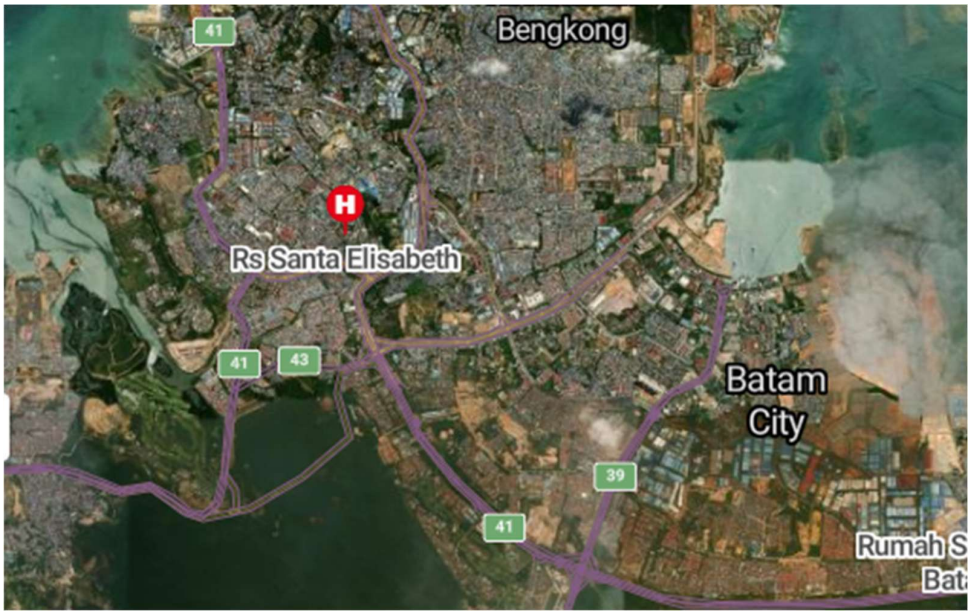
3.5 Lokasi Dan Jadwal Penelitian

Dalam persiapan proyek ini, akan dibahas tentang penentuan lokasi dan waktu penelitian yang penting agar peneliti dapat menyelesaikan proyek studi dengan tepat waktu dan sukses.

3.5.1 Lokasi Penelitian

Lokasi penelitian mencakup akses online melalui basis data akademik, perpustakaan digital, dan sumber-sumber literatur elektronik untuk studi literatur.

Selain itu, wawancara dengan dokter gizi dilakukan di Rumah Sakit Elisabeth Batam Kota.



Gambar 3. 14 Lokasi Penelitian
(Sumber: Data Penelitian, 2025)

3.5.2 Jadwal Penelitian

Peneliti telah merencanakan jadwal untuk melakukan investigasi dalam penelitian yang sedang berlangsung. Tujuan dari jadwal penelitian ini adalah untuk memperoleh pemahaman tentang perkembangan penelitian yang terjadi dalam beberapa bulan terakhir.

Tabel 3. 8 Jadwal Penelitian

JADWAL PENELITIAN																					
No	Kegiatan	Bulan																			
		September				October				November				Desember				Januari				Februari
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1
1	Pengajuan Judul																					
2	penyusunan Bab I																					
3	penyusunan Bab II																					
4	penyusunan Bab III																					
5	penyusunan Bab IV																					
6	penyusunan Bab V																					
7	Revisi																					
8	Pengumpulan skripsi																					

(Sumber: Data Penelitian, 2025)