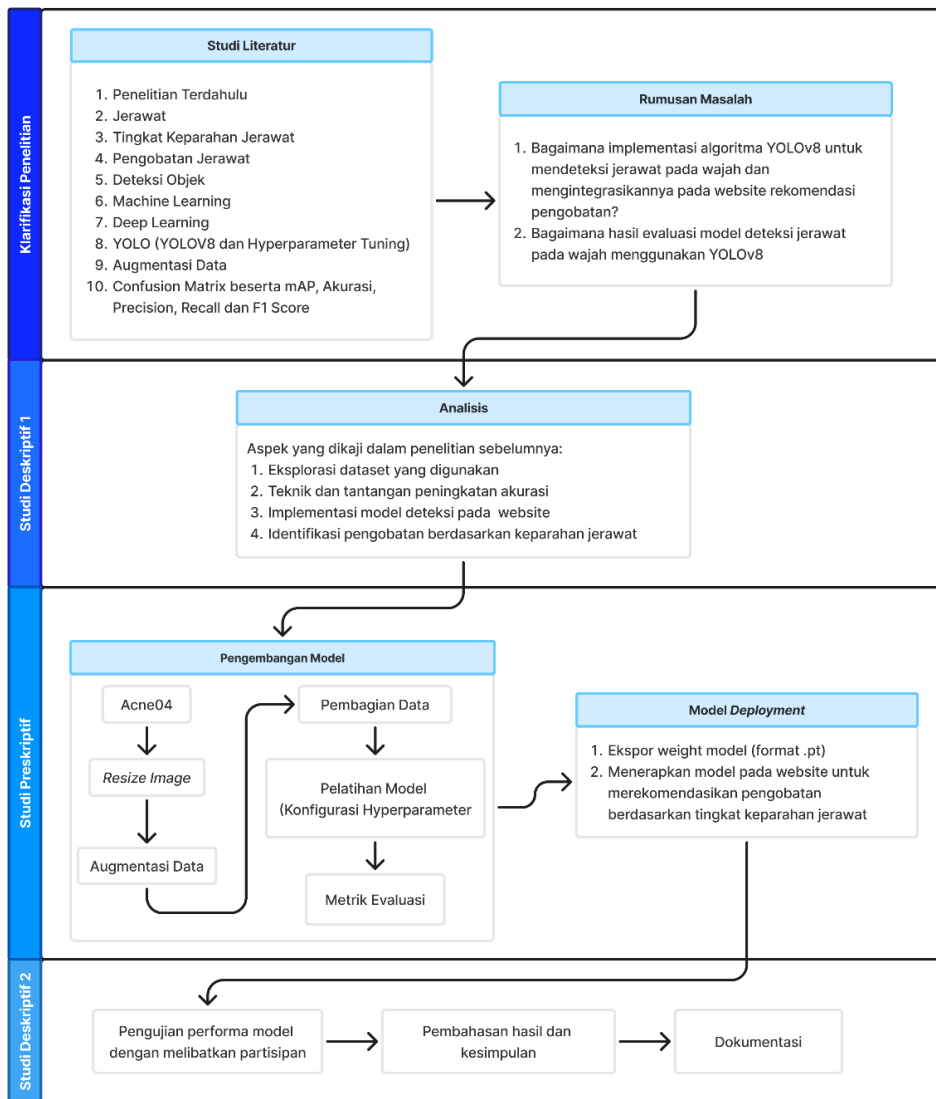


# BAB III

## METODOLOGI PENELITIAN

### 3.1 Desain Penelitian

Penelitian ini menerapkan pendekatan *Design Research Methodology* (DRM). Menurut Ebneyamini (2022), DRM adalah metode preskriptif yang bertujuan untuk mengembangkan atau merancang artefak guna memecahkan sejumlah masalah dalam domain pengetahuan tertentu. Metode ini terdiri atas empat tahapan yakni klarifikasi penelitian, studi deskriptif 1, studi preskriptif dan studi deskriptif 2. Gambar 3.1 menunjukkan skema penelitian yang akan dilakukan.



Gambar 3.1 Desain Penelitian

### 3.1.1 Klarifikasi Penelitian

Klarifikasi penelitian merupakan tahap awal pada DRM yang bertujuan untuk mengidentifikasi masalah penelitian serta memfokuskan tujuan penelitian yang akan dilakukan. Studi literatur dilakukan untuk menggali permasalahan dalam identifikasi jerawat dari berbagai metode berdasarkan penelitian-penelitian dalam jurnal artikel yang relevan. Selain itu dilakukan pemahaman mendalam terhadap permasalahan jerawat seperti tingkat keparahan jerawat serta pengobatan berdasarkan tingkatannya, teori deteksi objek yang melibatkan *machine learning* dan *deep learning* serta penggunaan algoritma YOLOv8 termasuk dengan metrik evaluasi yang digunakan dalam bidang deteksi objek. Hasil dari tahap ini adalah rumusan masalah dan tujuan penelitian yang dituangkan pada BAB I. Sementara untuk kajian literatur yang dianalisis dibahas pada BAB II.

### 3.1.2 Studi Deskriptif 1

Studi deskriptif 1 dilakukan untuk memperoleh pemahaman yang lebih mendalam terkait masalah yang diteliti. Pada tahap studi deskriptif 1 dilakukan kajian lebih lanjut dengan merujuk pada penelitian terdahulu yang telah melakukan pendeteksian objek khususnya pada deteksi jerawat. Kajian yang dilakukan meliputi metode, dataset, teknik dan tantangan peningkatan performa model hingga eksplorasi dataset yang dapat digunakan sebagai bahan penelitian. Melalui kajian ini fokus penelitian dipusatkan untuk mengidentifikasi jerawat menggunakan algoritma YOLOv8 untuk merekomendasikan pengobatan jerawat berdasarkan tingkat keparahan jerawat yang terdeteksi. Pengobatan yang direkomendasikan diperoleh melalui rujukan jurnal penelitian yang membahas mengenai pengobatan jerawat.

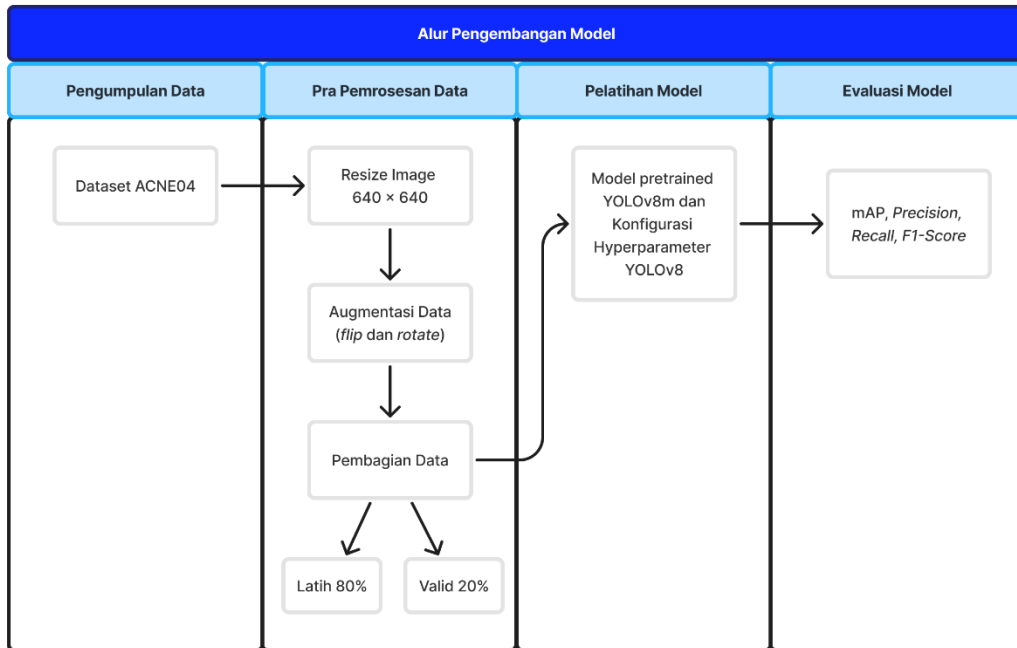
### 3.1.3 Studi Preskriptif

Tahapan studi preskriptif akan digunakan untuk melakukan pembuatan model pendeteksi jerawat menggunakan YOLOv8. Tahapan ini meliputi pengumpulan data, pra-pemrosesan data, pelatihan serta evaluasi model. Selanjutnya model terbaik akan diekspor untuk kemudian diintegrasikan kedalam

website yang akan digunakan sebagai alat untuk merekomendasikan pengobatan berdasarkan tingkat keparahan jerawat.

### 3.1.3.1 Pengembangan Model Deteksi Jerawat

Pengembangan model deteksi jerawat akan melewati beberapa tahapan yang digambarkan pada diagram berikut.



Gambar 3.2 Diagram Pengembangan Model Deteksi Jerawat

Diagram tersebut menjelaskan tahapan pengembangan model pendeteksi jerawat yang dilakukan. Dataset ACNE04 akan melalui tahap pra-pemrosesan sebelum pelatihan model dilakukan. Pelatihan model akan menggunakan *pretrained* YOLOv8m dan melakukan konfigurasi *hyperparameter* pada *optimizer*, *batch size* dan jumlah *epoch*. Selanjutnya model akan dievaluasi dengan metrik *mAP*, *precision*, *recall* dan *F1 Score*.

### 3.1.3.2 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini akan memanfaatkan dataset ACNE04 yang dikembangkan Wu dkk. (2019) untuk menilai tingkat keparahan jerawat mengikuti persyaratan kriteria penilaian Hayashi. Dataset ini juga pernah digunakan dalam penelitian Wen dkk. (2022); H. Zhang dan Ma (2022); serta Watef

dan Mahananto (2024). ACNE04 dipilih karena merupakan salah satu dataset yang terpublikasi dan sesuai dengan objek penelitian ini. Dataset ini diklaim memiliki jumlah sebanyak 1.457 gambar dengan total 18, 983 *bounding box* yang dianotasi langsung oleh pakar.

### 3.1.3.3 Pra-Pemrosesan Data

Prosedur ini fokus pada pengolahan dataset yang telah terkumpul. Citra akan melalui proses *resizing* dan augmentasi. Gambar diubah dan diseragamkan ukurannya menjadi 640 x 640 piksel. Ukuran tersebut dipilih dengan mempertimbangkan rasio dan dapat menghindari terjadinya *underfitting* pada model (Yanto dkk., 2023). Pemilihan ukuran tersebut juga dilakukan dengan memerhatikan algoritma YOLOv8 yang menggunakan ukuran piksel 640x640 piksel (Ibrahim dan Latifa, 2023). Proses augmentasi akan dilakukan melalui roboflow dan akan melalui beberapa modifikasi untuk menciptakan variasi data. Memilih komposisi data split yang tepat dapat membantu mencegah *overfitting* atau *underfitting*, sehingga model dapat berfungsi lebih efektif pada data yang sebelumnya tidak terlihat (Oktafiani dkk., 2023). Dalam penelitian ini pembagian data menggunakan rasio 80:20, yaitu 80% untuk data latih dan 20% sebagai data validasi. Dalam penelitian secara empiris yang dilakukan oleh Gholamy dkk. (2018) menunjukkan bahwa rasio 80:20 dianggap lebih baik untuk partisi data karena keseimbangan antara melatih model dan menguji kinerjanya pada subset yang representatif, sehingga mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi model.

### 3.1.3.4 Pelatihan Model

Implementasi model atau tahap pelatihan akan dilakukan menggunakan *tools* Google Colab Pro. *Hardware accelerator* yang digunakan ialah T4 GPU. Pelatihan model ini dilakukan dengan memanfaatkan model *pre-trained* YOLOv8. Penggunaan *pretrained* model ini memungkinkan pelatihan lebih cepat dan menghasilkan performa model yang cukup baik dalam melakukan deteksi objek (Ibrahim dan Latifa, 2023). Selain itu model yang telah dilatih sebelumnya dapat membantu generalisasi jaringan dan mempercepat konvergensi (Alzubaidi dkk.,

2021). Model *pre-trained* yang digunakan dalam penelitian ini ialah varian YOLOv8m (medium). Varian tersebut dipilih karena merupakan model berukuran menengah yang menawarkan keseimbangan yang baik antara kompleksitas dan performa model (Hussain, 2024).

Pelatihan model akan menggunakan jumlah epoch sebesar 120 epoch untuk memastikan bahwa rata-rata *loss* pada set pelatihan stabil (Wu dkk., 2019). Adam atau *Adaptive Moment Estimation* dan khususnya varian AdamW adalah salah satu *optimizer* yang paling umum digunakan dalam *deep learning* menurut Schmidt dkk. (dalam Wright dan Demeure, 2021). Memori yang lebih efisien dan daya komputasi yang lebih kecil adalah dua keunggulan Adam (Alzubaidi dkk., 2021). Dengan demikian *optimizer* yang digunakan pada penelitian ini ialah AdamW dengan learning rate 0.001 dan nilai yang diatur untuk *batch size* adalah 16.

Adapun berikut merupakan kepastakaan dan alat yang akan digunakan dalam mengembangkan model deteksi jerawat.

Tabel 3.1  
Pustaka Pengembangan Model

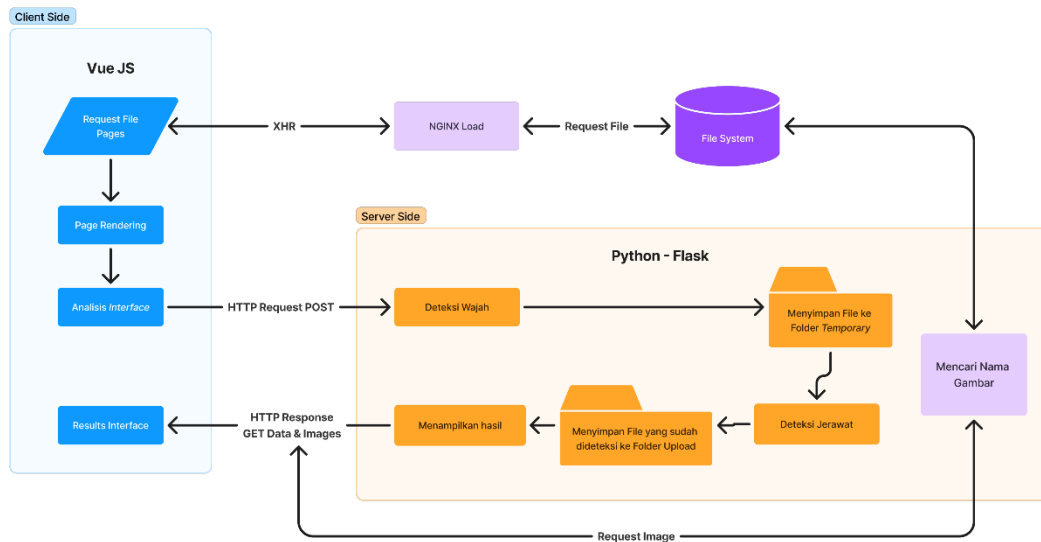
| <b>Kepustakaan</b> | <b>Deskripsi</b>  |
|--------------------|---|
| Ultralytics        | Untuk mengimplementasikan YOLOv8 untuk deteksi objek.                                     |
| Roboflow           | Untuk mengunduh dataset berformat YOLO menggunakan API dari Roboflow                      |
| Google Colab       | Untuk menghubungkan google colab dengan google drive.                                     |
| Shutil             | Untuk melakukan manajemen file, termasuk penyalinan dan pemindahan file.                  |
| Os                 | Digunakan untuk mengelola jalur file dan direktori.                                       |
| Matplotlib         | Digunakan untuk membuat visualisasi data.   |
| Pandas             | Untuk membaca file CSV untuk memudahkan dalam membaca hasil pelatihan dan validasi model. |

### 3.1.3.5 Pengembangan Website Rekomendasi (*Model Deployment*)

Dalam penelitian ini, website rekomendasi akan menjadi perantara bagi pengguna untuk melakukan pendeteksian jerawat.

### 3.1.3.5.1. Desain Arsitektur

Setelah memperoleh model terbaik, *file weight* model dengan format .pt akan diimplementasikan pada website.

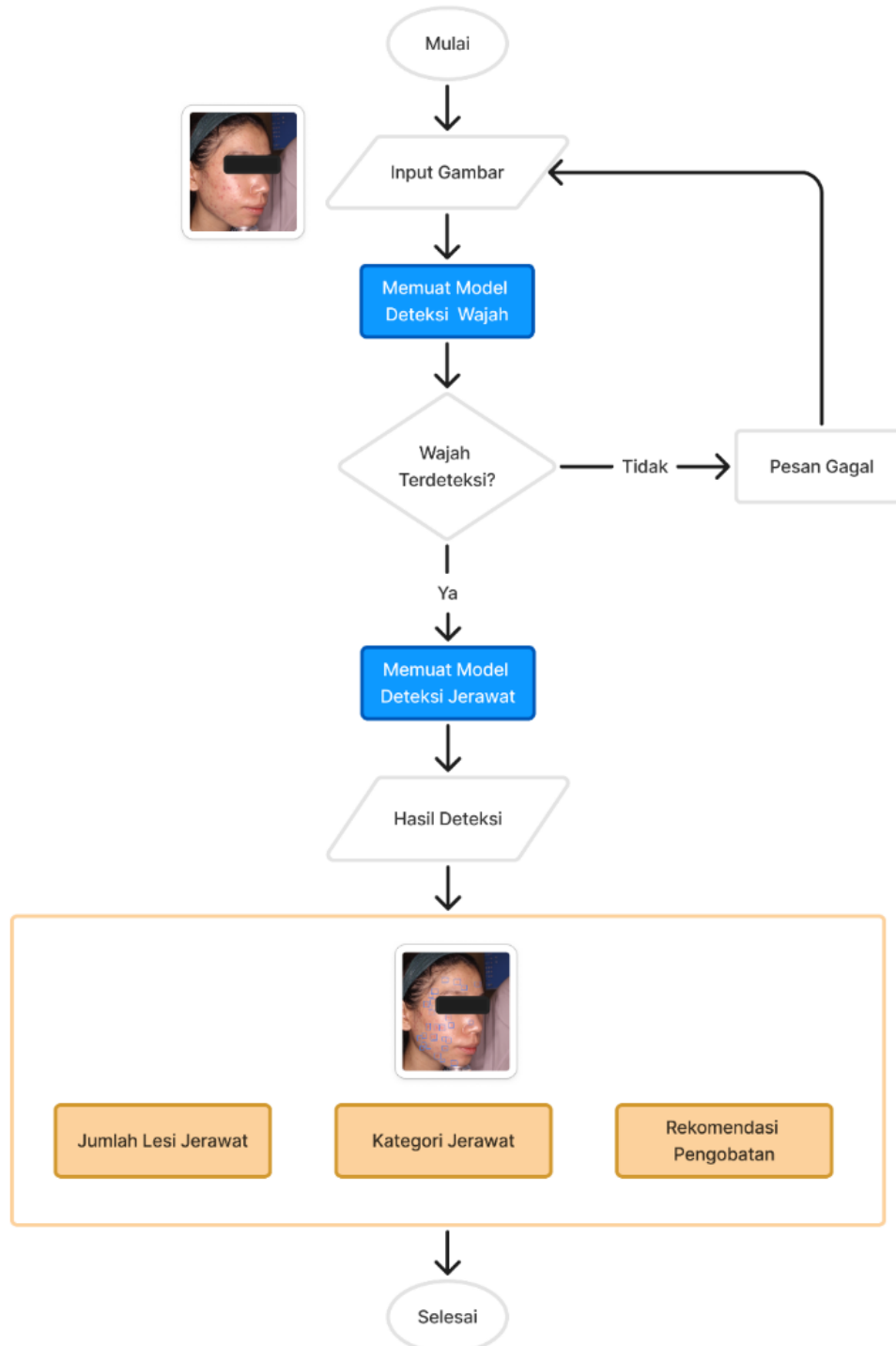


Gambar 3.3 Desain Arsitektur Website

Adapun gambar 3.3 merupakan desain arsitektur pada website. Diagram tersebut menggambarkan alur kerja sistem deteksi jerawat pada website yang dibagi menjadi dua bagian: *client-side* dan *server-side*. Pada *client side* antarmuka pengguna dibangun menggunakan Vue.js, yang bertanggung jawab untuk mengelola permintaan halaman, dan merender halaman. Ketika pengguna mengirimkan file gambar, Vue.js akan mengirimkan permintaan *file* ke *server* melalui Python, yang bertindak sebagai *server side* menggunakan metode API atau HTTP. File gambar tersebut kemudian disimpan dalam sistem *file* pada *server*. Sementara pada *server side*, proses dimulai dengan Python yang bertugas melakukan deteksi wajah pada gambar yang telah diunggah. Setelah wajah terdeteksi, *file* tersebut disimpan sementara di folder *temporary* dan melanjutkan proses dengan melakukan deteksi jerawat pada gambar tersebut. Setelah proses deteksi selesai, hasilnya disimpan di folder upload dan dikirim kembali ke sisi klien untuk ditampilkan kepada pengguna melalui halaman hasil deteksi. Adapun pustaka yang digunakan pada sisi *client side* dan *server side* beserta setiap fungsinya masing-masing dituliskan secara terperinci pada tabel 3.2.

### 3.1.3.5.2. User Flow Website Deteksi Jerawat

Adapun *user flow* proses deteksi jerawat pada website digambarkan pada gambar 3.4 sebagai berikut.



Gambar 3.4 User Flow Website Deteksi Jerawat

Pada saat website menerima *input* pengguna berupa gambar, maka model deteksi wajah akan dijalankan. Jika wajah terdeteksi maka gambar yang sama kemudian akan digunakan sebagai *input* untuk model selanjutnya, yang melakukan proses inferensi untuk mendeteksi jerawat pada wajah. Hasil inferensi dari model YOLO mencakup koordinat bounding box yang menunjukkan lokasi jerawat pada gambar, jumlah lesi, tingkat keparahan jerawat dan rekomendasi pengobatan yang sesuai. Adapun tabel 3.2 merupakan kepustakaan dan alat yang akan digunakan dalam pengembangan website rekomendasi pengobatan jerawat.

Tabel 3.2  
Pustaka Pengembangan Website

| <b>Kepustakaan/Alat</b> | <b>Deskripsi</b>   |
|-------------------------|--|
| Python                  | Bahasa pemrograman yang digunakan untuk mengembangkan website.   |
| Ultralytics             | Digunakan untuk mengimplementasikan YOLOv8 untuk deteksi objek.  |
| Flask                   | Digunakan untuk generate API yang akan di consume oleh web.  |
| NodeJS                  | Digunakan sebagai <i>package installer</i> pada web.   |
| VueJS                   | Digunakan sebagai pustaka <i>frontend</i> .  |
| Element Plus            | Pustaka UI yang menyediakan berbagai komponen siap pakai, seperti menu, form input, tombol, dan lain-lain. |
| Windy CSS               | Pustaka CSS yang menawarkan kelas-kelas siap pakai untuk mempercantik UI.                                  |
| OpenCV                  | Digunakan untuk memproses gambar agar bisa dideteksi oleh model.   |
| Axios                   | Digunakan untuk <i>consume</i> API dari service python.  |
| Roboflow                | Digunakan untuk mengambil model <i>existing</i> pada roboflow untuk melakukan deteksi wajah.               |
| Visual Studio Code      | Digunakan sebagai kode editor dalam pengembangan website rekomendasi pengobatan jerawat.                   |



### **3.1.4 Studi Deskriptif 2**

Tahap terakhir dari langkah DRM adalah studi deskriptif 2 yang mana pada tahap ini akan dilakukan evaluasi terhadap hasil pelatihan model yang telah diintegrasikan pada website rekomendasi pengobatan. Adapun pengujian yang dilakukan akan melibatkan partisipan untuk menguji coba langsung website deteksi untuk selanjutnya dilakukan evaluasi. Setelah melakukan pengujian maka selanjutnya adalah melakukan evaluasi hasil dengan menghitung nilai precision dan recall dari hasil deteksi model. Pembahasan dan analisis hasil serta simpulan menjawab pertanyaan dalam rumusan masalah akan didokumentasikan pada bab 4 dan 5. Penelitian ini ditutup dengan memberikan saran bagi peneliti selanjutnya untuk meningkatkan kualitas penelitian dan menyusun dokumen skripsi.

## **3.2 Partisipan**

Partisipan yang terlibat dalam penelitian ini akan diminta untuk menggunakan website yang sudah terintegrasi dengan model deteksi jerawat. Jumlah partisipan penelitian ini diadopsi dari penelitian Gold dkk. (2009) yang melibatkan 21 partisipan untuk mencoba perangkat untuk mengidentifikasi keparahan jerawat untuk perawatan kulit dengan kriteria memiliki jerawat berkategori ringan hingga sedang. Dengan jumlah partisipan sebanyak 21 orang dapat memberikan representasi yang cukup untuk mendapatkan pandangan awal tentang kinerja dan penerimaan sistem deteksi jerawat (Gold dkk., 2009). Sementara untuk karakteristik usia yang dipilih pada penelitian ini ialah usia 12-25 tahun. Hal tersebut karena menurut Isa dan Mangshor (2021) kelompok usia tersebut yang sering mengalami jerawat dan menurut Ip dkk (2021) biasanya memiliki sedikit pengetahuan tentang jerawat serta perawatannya.

## **3.3 Alat dan Bahan Penelitian**

Berikut merupakan alat penelitian yang digunakan dalam penelitian ini yang ditunjukkan dalam tabel 3.3.

Tabel 3.3  
Alat Penelitian

|                        |   |
|------------------------|---|
| <b>Perangkat Keras</b> | <ul style="list-style-type: none"> <li>a. Processor 11<sup>th</sup> Gen Intel Core i5-1135G7</li> <li>b. RAM 24 GB</li> <li>c. Intel Iris XE Graphics, NVIDIA GeForce MX330</li> <li>d. SSD 256 GB, HDD 1 TB</li> </ul> |
| <b>Perangkat Lunak</b> | <ul style="list-style-type: none"> <li>a. Sistem Operasi Windows 11</li> <li>b. Google Colaboratory dengan GPU T4</li> <li>c. Roboflow</li> <li>d. Visual Studio Code</li> <li>e. Figma</li> </ul>                      |

Adapun bahan penelitian yang dimanfaatkan dalam penelitian ini ialah jurnal, artikel ilmiah serta prosiding yang memiliki topik relevan. Sumber-sumber tersebut digunakan untuk menambah dan mendukung pemahaman penulis selama proses penelitian berlangsung.

### 3.4 Instrumen Penelitian

Instrumen penelitian adalah suatu alat yang dapat dipergunakan untuk mengukur suatu objek. Dalam hal ini instrumen tersebut digunakan untuk mengukur metrik evaluasi yang diidentifikasi dalam penelitian dengan cara yang terstandarisasi. Pada penelitian ini sebagai alat ukur pengembangan model, metrik yang digunakan ialah *mAP*, *precision*, *recall* dan *f1-score*.

#### 3.4.1 Mean Average Precision (mAP)

Metrik ini mengindikasikan kemampuan model untuk mendeteksi objek dari satu kelas dengan tingkat presisi yang tinggi di berbagai tingkat kepercayaan. mAP yang akan dihitung ialah mAP<sub>val</sub> yaitu rata-rata pada set validasi. *Mean average precision* (mAP) adalah nilai rata-rata dari AP. mAP dihitung dengan persamaan sebagai berikut.

$$AP = \int_0^1 p(r)dr \quad (1)$$

$$mAP = \frac{1}{N} \sum_{k=1}^{k=n} APk \times 100\% \quad (2)$$

### 3.4.2 Precision

*Precision* adalah metrik yang mengukur seberapa handal model pembelajaran mesin memprediksi kelas positif dengan tepat. Dengan kata lain presisi mencerminkan persentase prediksi positif yang benar. Presisi dihitung sebagai perbandingan antara jumlah sampel positif yang diklasifikasikan dengan benar dengan jumlah total sampel yang diklasifikasikan positif (baik benar atau salah). Perhitungan *precision* membutuhkan nilai TP yaitu jumlah *bounding box* yang diprediksi dengan benar, FP yaitu jumlah sampel positif yang dinilai salah, dan FN yaitu jumlah target yang tidak terdeteksi (Lou dkk., 2023). Adapun persamaan untuk mencari nilai *precision* adalah sebagai berikut

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{Seluruh\ Deteksi} \quad (3)$$

### 3.4.3 Recall

*Recall* menghitung persentase positif aktual yang diidentifikasi dengan benar oleh model. Dengan kata lain *recall* merupakan kemampuan model untuk menemukan semua kasus yang relevan (semua kotak pembatas yang sesuai dengan kebenaran). *Recall* dihitung menggunakan persamaan berikut.

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{Seluruh\ Ground\ Truth} \quad (4)$$

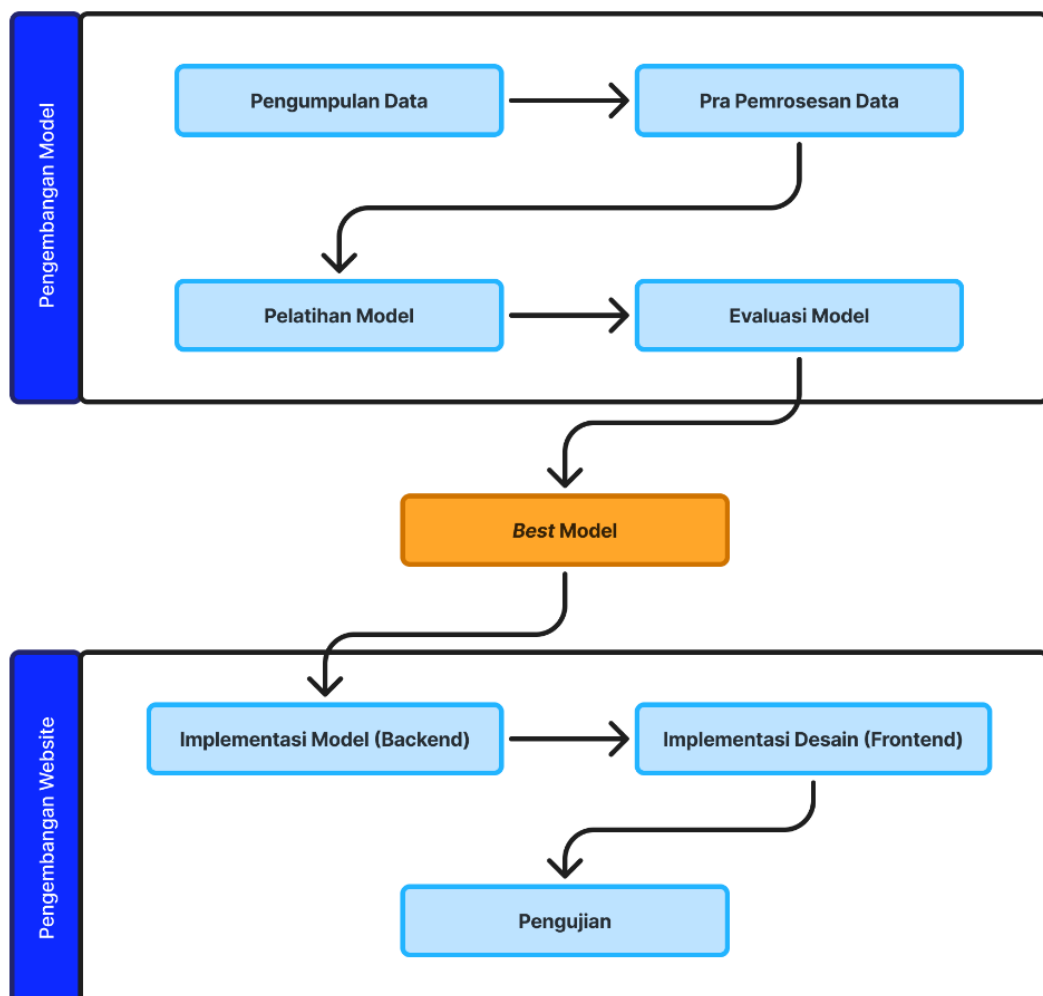
### 3.4.4 F1 Score

Metrik *f1-score* merupakan rata-rata dari *precision* dan *recall*. Skor yang sesuai dengan setiap kelas akan menunjukkan keakuratan pengklasifikasi dalam mengklasifikasikan titik data di kelas tertentu dibandingkan dengan semua kelas lainnya.

$$F1 \text{ Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

### 3.5 Prosedur Penelitian

Tujuan utama dari penelitian ini adalah untuk mengimplementasikan algoritma YOLOv8 untuk mendeteksi jerawat pada wajah. Untuk mencapai tujuan tersebut terdapat rangkaian proses yang harus dilakukan. Gambar 3.5 menunjukkan diagram prosedur penelitian.



Gambar 3.5 Prosedur Penelitian

Penelitian ini melibatkan pengembangan model dengan algoritma YOLOv8 dan pengembangan website rekomendasi pengobatan. Pengembangan model diawali dengan tahapan pengumpulan dataset, prapemrosesan data, pelatihan

hingga evaluasi model yang dilatih. Spesifikasi proses pengembangan model lebih lengkap dijelaskan pada gambar 3.2. Model yang telah dilatih akan digunakan untuk memberikan rekomendasi pengobatan berdasarkan tingkat keparahannya melalui website. Maka setelah memperoleh model terbaik akan dilakukan pengembangan website meliputi implementasi model pada *backend*, implementasi desain di sisi *frontend* serta melakukan pengujian pada website kepada partisipan penelitian dengan menghitung nilai *precision* dan *recall*. Desain arsitektur dan *user flow* website rekomendasi dituangkan lebih lengkap pada gambar 3.3 dan 3.4.