

BAB 3

METODE PENELITIAN

Penelitian ini menggunakan algoritma SSD MobileNet V2 untuk mendeteksi penyakit pada kulit wajah. Proses penelitian diawali dengan mengidentifikasi latar belakang, memetakan rangkaian langkah-langkah yang terlibat, mencari literasi yang relevan dan menghasilkan sistem yang dirancang dan dikembangkan untuk menganalisis penyakit pada kulit wajah. Penelitian ini meliputi bahan, alat dan metode pengembangan serta tahapan penerapan deteksi objek untuk menganalisis penyakit pada wajah dengan menggunakan algoritma SSD MobileNet V2 adalah sebagai berikut:

3.1 BAHAN DAN ALAT PENELITIAN

Penelitian ini menggunakan dataset penyakit kulit wajah yang diperoleh dari Kaggle. Data yang terkumpul meliputi gambar-gambar wajah yang terdapat penyakit wajah seperti jerawat, kutil, komedo, herpes, panu, milia, flek hitam, eksim dengan jumlah keseluruhan gambar sebanyak 800 gambar.

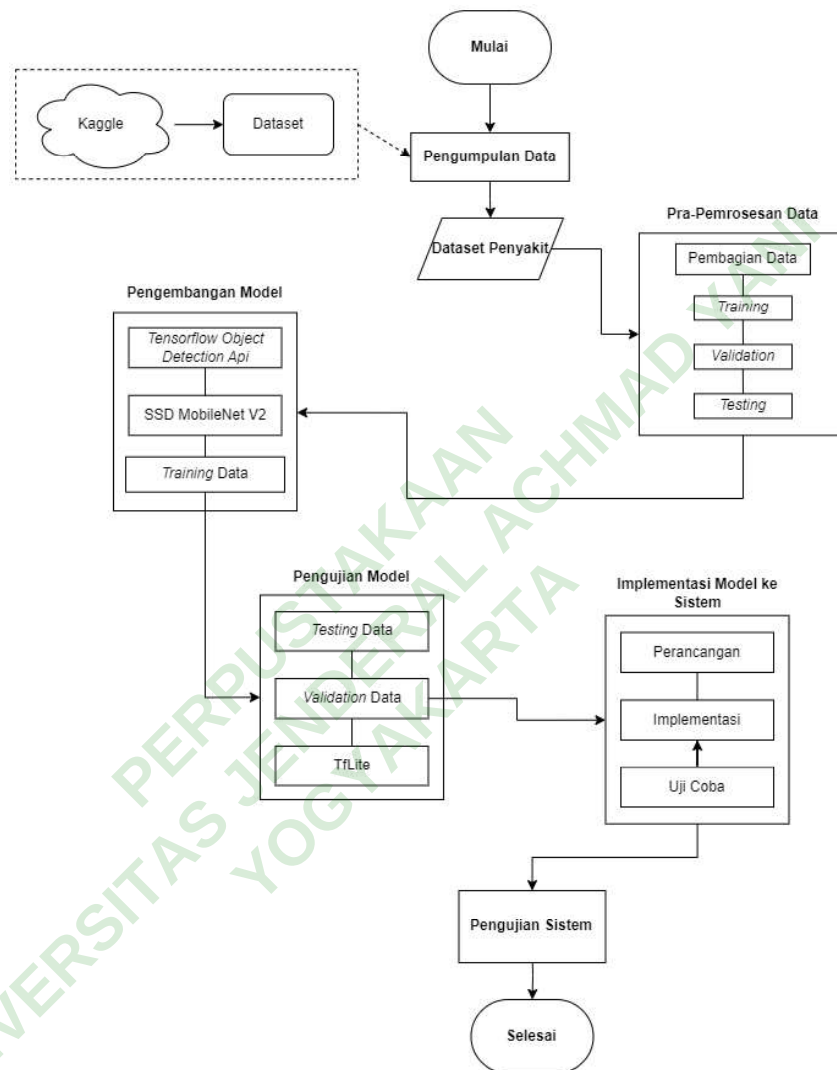
Alat dalam penelitian ini memerlukan seperangkat komputer dengan spesifikasi yang mumpuni untuk menjalankan sistem operasi dan beberapa program aplikasi serta dapat terkoneksi dengan internet.

Berikut Sistem Operasi serta aplikasi yang akan digunakan :

1. Sistem Operasi: Windows 11.
2. Bahasa Pemrograman: Kotlin ver 1.8
3. Android Studio ver 2023.3.1
4. Google Colab
5. Google Drive
6. Database engine: Firestore
7. Roboflow
8. TensorFlow ver 2.8.0

3.2 JALAN PENELITIAN

Berikut merupakan alur penelitian dan tahapan dari deteksi objek menggunakan algoritma SSD MobileNet V2 yang ditampilkan pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

3.2.1 Pengumpulan Data

Pada tahap ini data gambar diambil dari Kaggle dengan label penyakit yang sudah terdefinisi sebelumnya. Gambar-gambar tersebut berisi kumpulan penyakit yang akan menjadi dasar dalam pengembangan model. Penyakit yang terdefinisi mencakup 8 jenis penyakit, dengan total gambar sebanyak 800 gambar. Hasil pada tahap ini mencakup kumpulan gambar yang siap untuk diolah pada tahap selanjutnya.

3.2.2 Pra-pemrosesan Data

Pada tahap ini langkah detail dilakukan untuk mempersiapkan dataset sebelum dimasukkan kedalam model. Berikut langkah-langkah dalam pra-pemrosesan data:

1. Anotasi gambar, data gambar akan melalui tahap anotasi terlebih dahulu untuk memberikan tambahan informasi berupa *bounding box* serta kelas keseluruhan label.
2. Pembagian data, dataset dibagi menjadi tiga bagian: data pelatihan, data validasi, dan pengujian. Dari 800 gambar, 80% digunakan untuk data training, 10% untuk data validasi, dan 10% untuk data pengujian. Pembagian dataset dalam penelitian ini dijelaskan pada Tabel 3.1, Tabel 3.2, Tabel 3.3.

Tabel 3.1 Data Pelatihan

No	Jenis Penyakit	Data	Jumlah
1	Jerawat	Training	80
2	Flek Hitam	Training	80
3	Eksim	Training	80
4	Herpes	Training	80
5	Milia	Training	80
6	Panu	Training	80
7	Komedo	Training	80
8	Kutil	Training	80

Tabel 3.2 Data Validasi

No	Jenis Penyakit	Data	Jumlah
1	Jerawat	Validasi	10
2	Flek Hitam	Validasi	10
3	Eksim	Validasi	10
4	Herpes	Validasi	10
5	Milia	Validasi	10
6	Panu	Validasi	10
7	Komedo	Validasi	10

8	Kutil	Validasi	10
---	-------	----------	----

Tabel 3.3 Data Pengujian

No	Jenis Penyakit	Data	Jumlah
1	Jerawat	Pengujian	10
2	Flek Hitam	Pengujian	10
3	Eksim	Pengujian	10
4	Herpes	Pengujian	10
5	Milia	Pengujian	10
6	Panu	Pengujian	10
7	Komedo	Pengujian	10
8	Kutil	Pengujian	10

3. *Resizing* gambar, yaitu semua gambar dalam dataset diubah ukurannya menjadi 320 x 320 piksel agar memiliki dimensi yang konsisten dan memudahkan proses serta mengurangi beban komputasi.

3.2.3 Pengembangan Model

Pada tahap ini model dikembangkan dengan menggunakan algoritma SSD MobileNet V2 yang meliputi beberapa Langkah antara lain:

1. Instalasi *framework* dan *package*, dalam hal ini kita menggunakan *TensorFlow Object Detection API* untuk membangun model serta menggunakan CUDA dan memungkinkan pelatihan model menggunakan GPU, yang secara drastis mengurangi waktu pelatihan dibandingkan dengan hanya menggunakan CPU.
2. Model *training*, bertujuan untuk memastikan bahwa model dapat belajar dengan efektif yang meliputi konfigurasi dan penentuan parameter seperti *learning rate*, *batch size*, *epoch*. *Learning rate* yang digunakan adalah 0.08, ini untuk mengontrol seberapa besar perubahan bobot pada setiap langkah pembaruan. *Batch size* yang digunakan adalah 16 dimana *batch size* yang besar memanfaatkan paralelisme komputasi GPU secara lebih efisien, tetapi memerlukan lebih banyak memori. *epoch* yang digunakan adalah 20.000 dimana satu *epoch* adalah satu siklus penuh melalui seluruh dataset.

Menentukan jumlah *epoch* yang tepat penting untuk memastikan model terlatih dengan baik tanpa overfitting.

3. Model *validation*, bertujuan untuk mengidentifikasi dan mencegah overfitting serta memastikan konsistensi kinerja model. Setelah setiap *epoch*, model dievaluasi menggunakan dataset validasi yang tidak terlibat dalam proses pelatihan. Ini bertujuan untuk memastikan bahwa model dapat menggeneralisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya.

3.2.4 Pengujian Model

Model yang telah dilatih dan divalidasi akan diuji untuk mengevaluasi kinerjanya terhadap data yang belum pernah dilihat sebelumnya. Pengujian model akan mengambil data *testing* yang akan memberikan hasil apakah model berfungsi dengan baik atau perlu adanya evaluasi lebih lanjut serta dengan menambahkan parameter pada *threshold*.

3.2.5 Implementasi Model Ke Sistem

Pada tahap ini, model yang telah berhasil dilatih dan disimpan dalam format tflite akan dimasukkan kedalam direktori asset dalam base aplikasi. Format tflite dimaksudkan agar model dapat diimplementasikan dengan baik dalam aplikasi *mobile*. Bahasa pemrograman Kotlin dipilih untuk mengembangkan aplikasi ini karena Kotlin telah diadopsi secara resmi oleh Google sebagai bahasa utama untuk pengembangan aplikasi Android.

Setelah dilakukan konfigurasi maka saat kamera dijalankan, aplikasi akan memproses gambar secara real-time dengan memanfaatkan model SSD MobileNet V2 yang nantinya akan menampilkan *bounding box* pada objek yang berhasil terdeteksi. Saat ada objek penyakit wajah yang berhasil terdeteksi maka akan otomatis melakukan *capture* gambar dan akan menuju ke halaman hasil dimana akan menampilkan informasi mengenai penyakit yang terdeteksi, penyebab dan saran penanganan. Informasi tersebut akan diambil dari database Firestore yang telah disimpan di firebase. Dengan demikian, implementasi ini tidak hanya

memberikan prediksi yang akurat tetapi juga memberikan solusi yang dapat membantu pengguna dalam mengatasi masalah pada wajah mereka.

3.2.6 Pengujian Sistem

Setelah sistem dibangun, tahap berikutnya adalah pengujian. Pengujian ini bertujuan untuk memastikan bahwa logika internal dan fungsi eksternal perangkat lunak berfungsi dengan baik. Pengujian dilakukan dengan mengevaluasi semua fungsi untuk menemukan kesalahan dan memverifikasi bahwa *input* yang diberikan menghasilkan *output* sesuai yang diharapkan. Sistem diuji menggunakan metode *black box* untuk memastikan bahwa fungsionalitas perangkat lunak bekerja dengan benar tanpa melihat ke dalam struktur internal atau kode sumbernya. Metode ini menekankan pada spesifikasi dan persyaratan perangkat lunak untuk memastikan bahwa sistem memenuhi kebutuhan yang telah ditentukan.

Tabel 3.4 Rencana Pengujian

No	Gambar	Jenis Penyakit	Hasil Deteksi	Kesimpulan
1		Jerawat		
2		Komedo		
3		Kutil		
4		Eksim		
5		Herpes		
6		Panu		
7		Milia		
8		Flek Hitam		

Tabel 3.2 merupakan rencana pengujian untuk memastikan apakah model yang telah diimplementasikan ke dalam aplikasi bekerja dengan baik atau tidak.