

BAB III

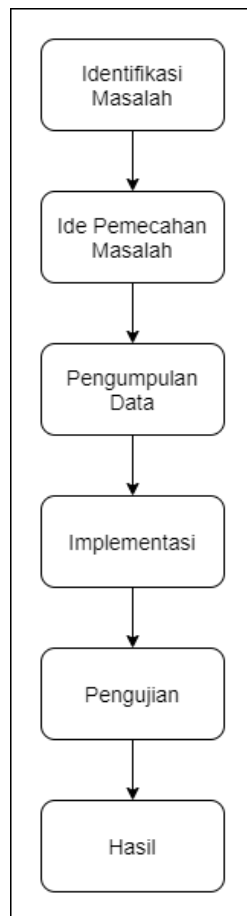
METODOLOGI PENELITIAN

3.1 Analisis Masalah

Pada permasalahan yang telah dijelaskan pada BAB 1 maka penelitian ini bertujuan untuk membuat sebuah program yang mampu mengenali wajah pemilik laptop dengan memanfaatkan *webcam* yang tersedia sehingga fitur *privacy guard* dapat ditambahkan seperti yang ada pada laptop Lenovo. Dengan memanfaatkan MobileNet-SSD diharapkan mampu melakukan identifikasi dan mengenali wajah pemilik laptop dengan akurat. Selain itu penelitian ini bertujuan untuk mencari *dataset* minimum yang ideal untuk digunakan dalam *face recognition*.

3.2 Rancangan Penelitian

Pada Gambar 3.1 dijelaskan bagaimana langkah-langkah yang akan dilakukan dalam penelitian ini.



Gambar 3.1 Alur rancangan penelitian

1. Identifikasi Masalah

Orang lain dapat melihat apa yang sedang kita lakukan saat melakukan kegiatan menggunakan laptop sehingga privasi dinilai kurang.

2. Ide Pemecahan Masalah

Memanfaatkan *webcam* supaya dapat mengenali wajah dari pemilik laptop sehingga apabila terdapat wajah lain yang tidak dikenali oleh *wabcam* maka laptop akan me-*minimize* semua aplikasi yang sedang terbuka dan metode yang digunakan yaitu MobileNet-SSD

3. Pengumpulan Data

Mengumpulkan foto wajah dari pemilik laptop dengan tujuan dijadikan *dataset* untuk proses *training* dan *testing*.

4. Implementasi

Membuat program untuk mengenali wajah dari pemilik laptop serta melakukan *training* dan *testing* menggunakan *dataset* yang telah dibuat sebelumnya

5. Pengujian

Pengujian akan dilakukan untuk mengetahui apakah program berjalan sesuai yang diharapkan atau tidak.

6. Hasil

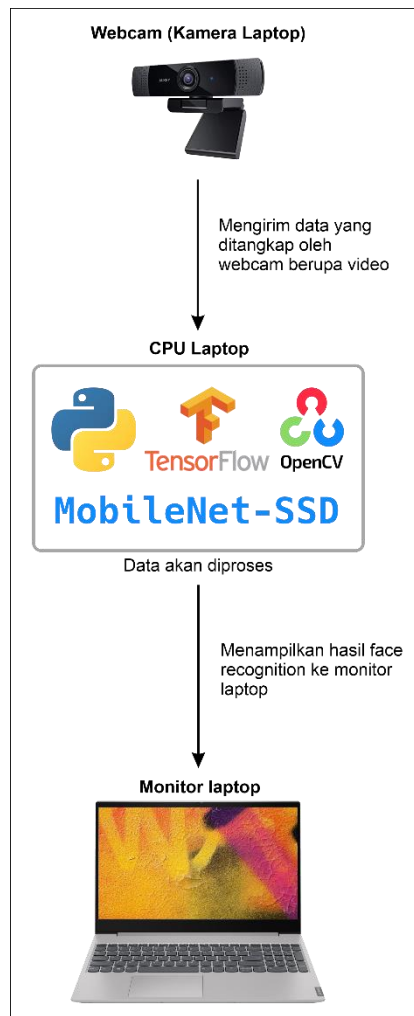
Webcam dapat mengenali wajah dari pemilik laptop dan me-*minimize* semua aplikasi yang terbuka jika mendeteksi wajah lain.

3.2.2 Arsitektur Sistem

Sedangkan proses deteksi dan pengenalan wajah orang yang di inginkan tersebut dilakukan oleh kamera yang kemudian data berupa citra gambar akan diproses oleh *computer vision*. Arsitektur sistem dapat dilihat pada Gambar 3.2.

Input-an dari *webcam* akan diproses oleh CPU laptop dengan *image processing* menggunakan Python, Tensorflow, Keras, dan OpenCV dengan arsitektur model MobileNet dan metode SSD dimana metode ini berfungsi untuk mendeteksi atau mengenali objek. Proses *face detection* dilakukan pertama kali dan mengabaikan semua objek yang bukan wajah sehingga akan memperingan komputasi setelah itu proses *face recognition* dimana menggunakan model hasil *training* terhadap wajah

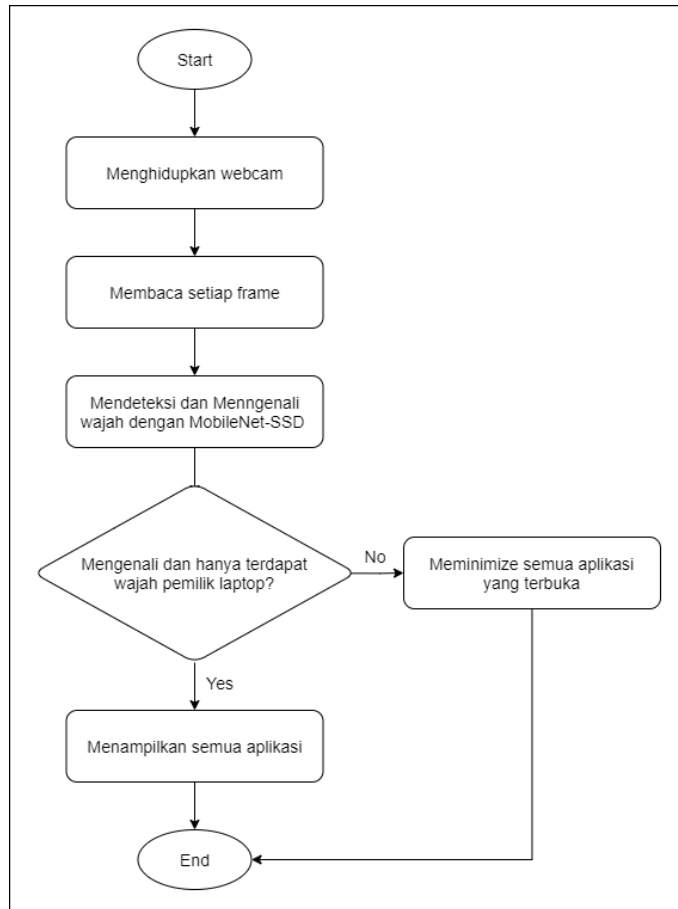
pemilik laptop sehingga *webcam* dapat mengenali atau tidak terhadap wajah yang didapat. Kemudian hasil dari *face recognition* akan di tampilkan ke dalam monitor.



Gambar 3.2 Arsitektur sistem

3.2.3 Flowchart Program

Perancangan program pada penelitian ini merupakan implementasi dari metode MobileNet-SSD untuk *face detection* dan MobileNet untuk *face recognition*. Secara sederhana cara kerja program *privacy guard* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Flowchart program *privacy guard*

Secara garis besar tahapan dalam *face detection* adalah dengan mengambil atau membaca setiap *frame* citra dari *wabcam*. *Frame* tersebut berupa foto yang akan diolah oleh SSD untuk mendeteksi wajah dengan memanfaatkan *model pretrained* SSD yang sudah ada khusus untuk *face detection*. Proses *face detection* ini akan dilakukan menggunakan pengenalan citra dari ekstraksi ciri kedua mata, apabila terdeteksi maka akan dilanjutkan ke tahap normalisasi bentuk citra dan orientasi wajah. Setelah ditemukannya wajah maka bagian sekitar wajah tersebut akan di-*crop* untuk selanjutnya dikenali atau tidak dengan menggunakan model yang akan dibuat pada penelitian ini.

3.3 Bahan dan Alat

3.3.1 Bahan

Bahan merupakan sesuatu yang akan diolah sehingga menghasilkan sesuatu yang lainnya, di dalam penelitian ini bahan yang digunakan yaitu berupa kumpulan foto

sehingga membentuk sebuah *dataset*. *Dataset* berupa foto penulis dan foto orang lain sehingga akan terbentuk menjadi 2 kategori yaitu *edi* dan *unknown*.

3.3.2 Alat

Alat adalah sesuatu yang digunakan untuk mengolah bahan tersebut, dalam penelitian ini alat tersebut ialah berupa laptop. Laptop tersebut tentunya memiliki perangkat keras dan perangkat lunak yang digunakan dan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Perangkat keras dan lunak yang digunakan

Perangkat Keras	Perangkat Lunak
Intel® Core™ i5-6300U CPU @2.40GHz (4 CPUs), ~2.5GHz	Windows 10 Professional
RAM 16GB	Python
Graphics Display Intel® HD Graphics 520	OpenCV
Storage SSD 256GB	TensorFlow
Webcam 720p	Keras
	Pynput
	Pycharm Community Edition

3.4 Metode Penelitian

3.4.1 Pengambilan Data

Pada tahap ini pengambilan data dilakukan otomatis menggunakan program yang telah dibuat. Dalam hal ini penulis memanfaatkan *model pretrained SSD* yang sudah ada untuk *face detection*. Foto tersebut didapatkan menggunakan program yang secara otomatis akan mendeteksi bagian wajah dan meng-*crop* sekitar area wajahnya. Program akan membaca setiap *frame* dari *webcam* dan penulis dapat memosisikan wajah dengan tepat, dan tombol spasi digunakan untuk meng-*capture* dan menyimpannya ke dalam direktori yang telah ditentukan.

Pada penelitian ini juga bertujuan untuk mencari *dataset* minimum untuk *face recognition*, sehingga penulis membagi *dataset* menjadi 4. *Dataset* pertama berisi

5 foto pada masing-masing kategori, *dataset* berikutnya berisi 10, 15, dan 20 foto pada masing-masing kategori.

3.4.2 Implementasi program

Ada beberapa program yang akan dibuat pada penelitian ini. Program pertama digunakan untuk mengambil foto wajah penulis menggunakan *webcam* secara otomatis. Program kedua yaitu digunakan untuk *training* dari *dataset* yang telah dibuat. Program ketiga digunakan untuk menguji *face detection*, sedangkan program keempat digunakan untuk implementasi dari fitur *privacy guard*. Dalam implementasi ini penulis memanfaatkan *model pretrained SSD* untuk *face detection* sehingga penulis membuat model untuk *face recognition* menggunakan arsitektur MobileNet dan menggunakan metode *transfer learning*. Implementasi program ini mengimplementasikan dari beberapa *library* Python yaitu:

1. OpenCV adalah *library* yang berfungsi mengolah gambar maupun video . OpenCV dapat digunakan untuk melakukan pengenalan citra wajah dan beragam objek lainnya.
2. Tensorflow adalah *library* yang berfungsi untuk membuat model *machine learning*.
3. Keras adalah *library* yang berfungsi untuk membuat penerapan model pembelajaran yang mendalam secepat dan semudah mungkin.
4. Google Colaboratory adalah sebuah layanan yang bertujuan untuk mengeksekusi atau menjalankan program pada sistem *cloud*. Pada penelitian ini, Google Colab berfungsi untuk *training* dan membuat model prediksi.
5. Pynput adalah sebuah *library* yang berfungsi untuk mengontrol dan *monitoring device input* berupa *keyboard* dan *mouse*.

Selanjutnya setelah seluruh data dan program sudah siap, maka proses *training* sudah dapat dilakukan. Foto yang akan diproses untuk *training* adalah sebesar 80% dari jumlah seluruhnya sedangkan sisanya digunakan untuk validasi yaitu sebesar 20% dengan menggunakan layanan Google Colaboratory. Hasil dari proses *training* ini adalah sebuah *file* dengan ekstensi *.h5* yang nantinya akan dapat digunakan untuk mendeteksi kategori suatu objek pada sebuah gambar maupun video.

3.4.3 Pengujian

Pengujian ini dilakukan untuk mengetahui seberapa bagus model yang telah dihasilkan dan juga mengetahui apakah program berjalan sesuai yang diinginkan atau tidak. Pada *face detection*, *model pretrained* SSD akan diuji dengan 20 foto dengan kategori *face* dan 20 foto dengan kategori *not face*, selain itu diuji dengan menggunakan *webcam* dengan parameter jarak dan posisi wajah. Selanjutnya untuk *face recognition*, akan ada 4 model yang dihasilkan dari masing-masing *dataset* di mana setiap model tersebut akan diuji dengan 20 foto kategori *edi* dan 20 foto kategori *unknown*, selain itu model yang memiliki akurasi terbaik berdasarkan nilai *accuracy*, *precision* dan *recall* akan digunakan untuk selanjutnya diuji menggunakan *webcam* dengan parameter jarak dan posisi wajah. Model tersebut juga yang akan digunakan untuk membuat program *privacy guard*.

Pada pengujian program *privacy guard* ada beberapa kondisi yang harus dipenuhi sehingga memiliki fungsi yang hampir sama dengan aplikasi yang sesungguhnya. Beberapa kondisi tersebut dapat diuraikan sebagai berikut:

1. Terdapat hanya wajah penulis maka program akan tetap menampilkan semua aplikasi.
2. Tidak terdapat wajah pemilik laptop maka program akan *me-minimize* semua aplikasi yang terbuka.
3. Terdapat wajah pemilik laptop dan wajah orang lain maka program akan *me-minimize* semua aplikasi yang terbuka.