

LAPORAN TUGAS AKHIR

DETEKSI KENDARAAN MOBIL BERDASARKAN
HISTOGRAM OF ORIENTED GRADIENTS (HOG) DENGAN
METODE SUPPORT VECTOR MACHINE (SVM)

Mata Kuliah : Pengenalan Pola



Muhammad Luqman Aristio

2008561022

PROGRAM STUDI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

2023

BAB 1

MANUAL BOOK

1.1. Pendahuluan

Aplikasi yang dikembangkan merupakan aplikasi pendeteksi kendaraan mobil otomatis berdasarkan gambar ataupun video. Metode yang digunakan dalam pendeteksian yakni *histogram of gradients* (HOG). HOG adalah sebuah teknik dalam pengolahan citra dan visi komputer yang digunakan untuk mengekstrak fitur-fitur penting dari suatu gambar. HOG digunakan secara luas dalam aplikasi deteksi objek, terutama dalam konteks deteksi manusia dan objek pada citra. Hasil dari HOG akan digunakan dalam algoritma pembelajaran mesin yakni *support vector machine* (SVM). Pada SVM inilah akan dilakukan pelatihan.

Hasil dari model pelatihan SVM ini akan digunakan dalam aplikasi yang dikembangkan. Aplikasi dikembangkan menggunakan framework flask. Pada aplikasi, pengguna perlu memasukkan sebuah gambar ataupun video yang mengandung objek mobil. Kemudian aplikasi akan memproses input tersebut menggunakan model yang dimiliki dan menghasilkan output yang berisi hasil deteksi objek tersebut.

1.2. Kebutuhan Instalasi

1.2.1. Model

Pada proses pelatihan model, digunakan google colab sebagai platform atau environmentnya. Untuk file program pelatihan model dapat dilihat pada repositori github pada folder **training model** berikut :

<https://github.com/LuqmanAristio/car-detection-svm-hog>

File program tersebut dapat diupload menuju google colab. Untuk dapat menjalankannya, pengguna perlu mempersiapkan dataset yang digunakan. Pada program tersebut, digunakan google drive sebagai platform penyimpanan dataset.

1.2.2. Dataset

Dataset yang digunakan berupa citra dengan 2 label yakni label mobil dan non mobil. Jumlah dataset kurang lebih 16 ribu secara keseluruhan. Dataset dapat diakses pada link berikut :

https://drive.google.com/file/d/1Vkt3LtdNi6SxmhYG7Rj2kxr_PCdKNMSE/view?usp=sharing

1.2.3. Aplikasi

Aplikasi yang dikembangkan berbasis website. Sebelum menjalankan aplikasi, terdapat beberapa hal yang perlu dipersiapkan sebagai berikut :

- Python : Pastikan python sudah terinstall secara global dalam sistem yang digunakan
- PIP (Package Installer for Python): PIP digunakan untuk menginstal dan mengelola paket Python. Sebagian besar distribusi Python menyertakan PIP secara default.
- Text Editor atau IDE : Contoh IDE populer untuk pengembangan Flask termasuk Visual Studio Code, PyCharm, atau Sublime Text.

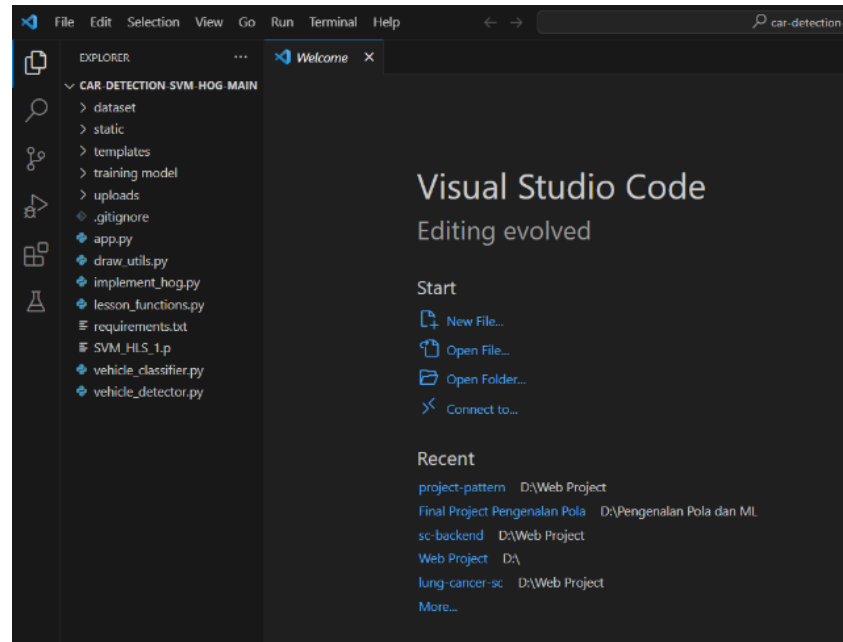
1.3. Menjalankan Program

Untuk melakukan **installasi** sebelum menggunakan program, ikuti langkah-langkah dibawah ini :

1. Unduh file program secara keseluruhan pada link repositori github dibawah ini :

<https://github.com/LuqmanAristio/car-detection-svm-hog>

2. Sesudah file program berhasil diunduh, lakukan ekstraksi jika mendownloadnya dalam format .zip. Kemudian setelah melakukan ekstraksi, masuk kedalam folder hasilnya.
3. Setelah itu buka isi folder tersebut menggunakan IDE yang digunakan. Berikut merupakan contohnya menggunakan IDE Visual Studio Code :



4. Buka terminal pada Visual Studio Code atau gunakan shortcut **CTRL + SHIFT + `** pada **keyboard**. Kemudian kita akan membuat environment untuk menjalankan aplikasi. Jalankan perintah berikut pada terminal :

- Untuk Windows

```
py -3 -m venv .venv
```

- Untuk macOS/Linux

```
python3 -m venv .venv
```

5. Setelah berhasil, berikut yakni mengaktifkan environment yang telah dibuat dengan perintah berikut :

- Untuk Windows

```
.venv\Scripts\activate
```

- Untuk macOS/Linux

```
..venv/bin/activate
```

6. Jika berhasil, maka seharusnya tampilan terminal akan berada di posisi environment yang dibuat. Berikut gambarnya :

```
PS D:\Pengenalan Pola dan ML\car-detection-svm-hog-main> .venv\Scripts\activate  
(.venv) PS D:\Pengenalan Pola dan ML\car-detection-svm-hog-main> █
```

- Setelah berada dalam environment yang dibuat, berikut yakni kita perlu melakukan instalasi terhadap framework flask. Jalankan perintah berikut pada terminal :

```
pip install flask
```

- Kemudian jalankan perintah berikut juga untuk melakukan instalasi terhadap seluruh kebutuhan library yang digunakan program :

```
pip install -r requirements.txt
```

Jika berhasil, maka sistem akan mengunduh seluruh library yang digunakan dalam aplikasi. Tunggu prosesnya hingga selesai.

- Terakhir, jika instalasi library sebelumnya selesai. Berikut yakni tinggal menjalankan program dengan perintah berikut :

```
flask --app app run
```

Tunggu prosesnya sejenak maka kemudian akan tampil seperti berikut pada terminal :

```
(.venv) PS D:\Pengenalan Pola dan ML\car-detection-svm-hog-main> flask --app app run
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [24/Dec/2023 17:20:41] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Dec/2023 17:20:41] "GET /static/image/vehicle-icon-1-300x300-1.png HTTP/1.1" 200 -
127.0.0.1 - - [24/Dec/2023 17:20:41] "GET /static/styles/index.css HTTP/1.1" 200 -
127.0.0.1 - - [24/Dec/2023 17:20:41] "GET /favicon.ico HTTP/1.1" 404 -
```

Buka url pada tampilan tertera, sebagai contoh pada gambar yakni <http://127.0.0.1:5000>. Buka pada browser maka akan tampil aplikasi yang sudah berjalan.

Berikut merupakan langkah-langkah dalam penggunaan aplikasi untuk melakukan deteksi mobil pada gambar atau video :

- Persiapkan gambar atau video yang akan digunakan, bisa menggunakan data tes yang tersedia pada link berikut :

<https://drive.google.com/drive/folders/1k-Q9T3exmiX0lon7MH4Hk777f3zRiXHF?usp=sharing>

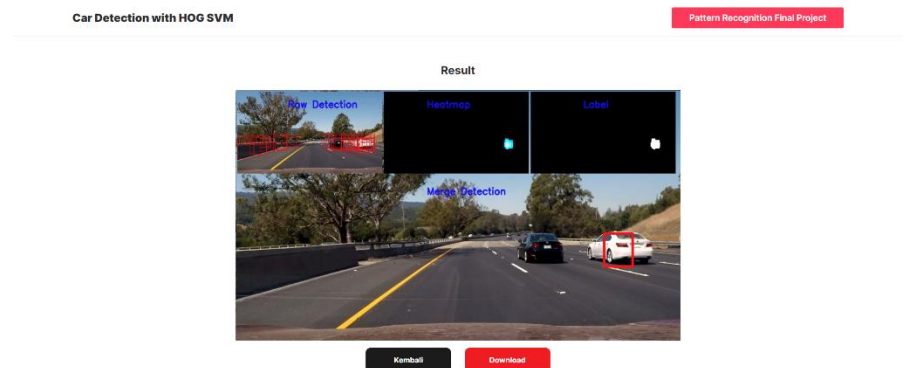
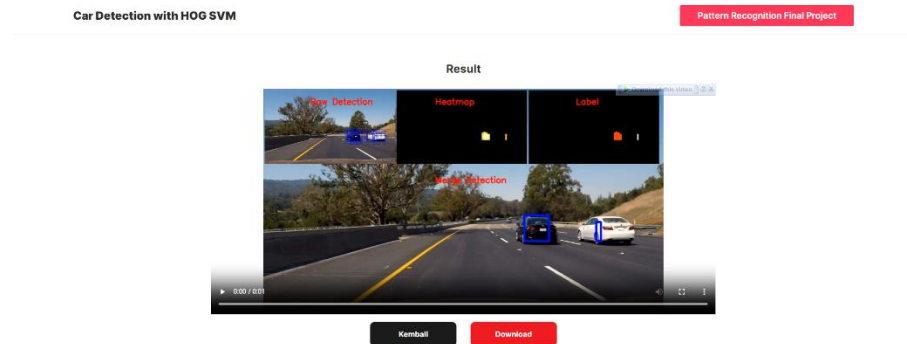
- Kemudian tekan tombol upload pada tombol **choose file** untuk memilih file yang akan dimasukan. Kemudian jika sudah maka tinggal klik upload untuk memprosesnya.
- Jikalau file input merupakan gambar, maka proses tidak akan berlangsung lama namun jika file berupa video maka biasanya akan

memakan waktu beberapa menit tergantung durasi video. Kita bisa melihat prosesnya pada terminal seperti berikut :

```
Moviepy - Building video static/result_video.mp4
Moviepy - Writing video static/result_video.mp4
t: 18% | 7/38 [00:04:00:23, 1.30it/s, now=None]
```

Disana terdapat tampilan durasi estimasi prosesnya.

4. Kemudian jika sudah selesai, maka akan tampil hasilnya sebagai berikut :

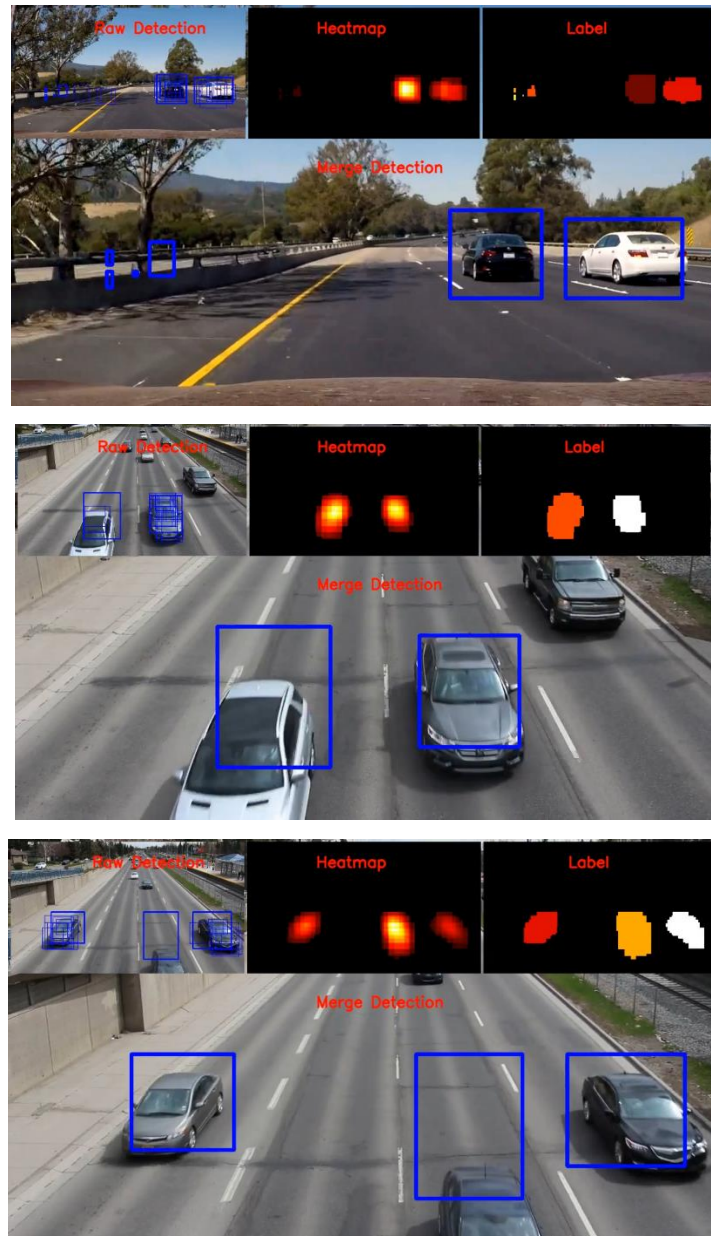


BAB 2

HASIL DAN ANALISIS

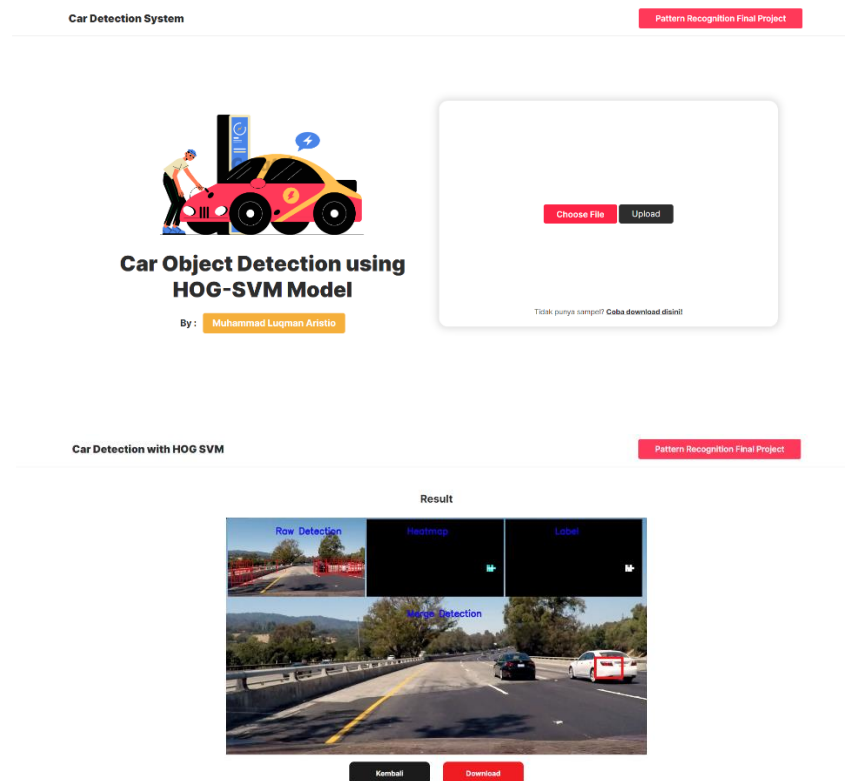
2.1 Hasil

Berikut merupakan beberapa hasil gambar dan video hasil deteksi mobil menggunakan metode HOG-SVM :



Bisa dilihat untuk hasilnya, model mampu mendeteksi mobil yang ada pada video dengan membuat kotak warna biru sebagai penandanya. Selain itu juga bisa dilihat hasil deteksi secara raw, heatmap serta labelingnya

Berikut merupakan beberapa hasil tangkapan layer dari aplikasi yang dikembangkan :



Terdapat 2 halaman utama yakni halaman untuk user menginputkan gambar yang digunakan serta halaman untuk menampilkan hasil deteksinya. File tersebut dapat didownload oleh pengguna jika diinginkan

2.2 Analisis

2.2.1. Model

Pada pengembangan model menggunakan metode HOG SVM untuk mendeteksi kendaraan mobil, diperoleh akurasi sebesar 98 % pada kedua label. Walaupun akurasi yang diperoleh sangat tinggi, namun generalisasi model masih sangat rendah terhadap berbagai jenis tipe mobil. Hal ini menyebabkan sering sekali kendaraan mobil yang jenis sangat berbeda dengan dataset yang

dimiliki tidak terdeteksi saat diterapkan. Berikut contohnya :



Pada gambar tersebut, mobil tidak terdeteksi sama sekali padahal gambar sudah sangat jelas. Hal ini dikarenakan kurang bervariasinya dataset yang digunakan.

Selain itu juga metode histogram of gradients(HOG) memiliki beberapa kekurangan yang mempengaruhi performa model. Dalam kasus ini, beberapa kekurangan dari HOG yakni sebagai berikut :

1. HOG tidak secara alami invariant terhadap transformasi ruang, seperti rotasi dan perubahan ukuran.
2. Terjadinya overfitting seperti penjelasan sebelumnya karena HOG menghasilkan fitur yang terlalu rinci sehingga tidak beradaptasi dengan baik dengan perbedaan yang signifikan.
3. Komputasi yang sangat tinggi karena melibatkan perhitungan gradien di seluruh citra dan penghitungan histogram di seluruh sel dan blok

2.2.2. Aplikasi

Pada aplikasi yang dikembangkan dengan framework flask, mampu berjalan sebagaimana mestinya. Aplikasi mampu menerima inputan dari pengguna, kemudian memproses inputan tersebut menggunakan model yang telah dibuat dan terakhir yakni menampilkannya kepada pengguna sesuai hasil deteksi.

Namun masih terdapat beberapa kekurangan dalam aplikasi sebagai berikut :

1. Pemrosesan untuk file audio masih sangat lama, hal ini disebabkan karena proses dilakukan per frame sehingga sangat dipengaruhi durasi videonya
2. Tidak dapat melakukan pemrosesan secara real-time sesuai kebutuhan.
3. Butuh peningkatan dibagian pengolahan citra dengan memanfaatkan library yang lebih efektif.

BAB 3

KESIMPULAN

Berdasarkan beberapa pembahasan di bab sebelumnya, diperoleh beberapa kesimpulan dari laporan ini :

1. Penerapan metode HOG-SVM untuk melakukan proses deteksi kendaraan mobil dapat dilakukan dengan baik, namun masih memerlukan beberapa peningkatan khususnya dibidang ekstraksi fitur HOG.
2. Model yang dihasilkan sudah dapat melakukan tugasnya dengan baik namun memiliki kekurangan besar dalam masalah komputasi yang lama.
3. Aplikasi yang dikembangkan sudah mampu berjalan sesuai tujuan yakni membantu pengguna mendeteksi kendaraan mobil berdasarkan gambar ataupun video.
4. Kedepannya, penerapan deteksi dapat dilakukan menggunakan metode yang lebih efektif dan efisien seperti jaringan syaraf tiruan sehingga mempersingkat waktu komputasi.