

BAB III

PERANCANGAN ALAT DAN SISTEM

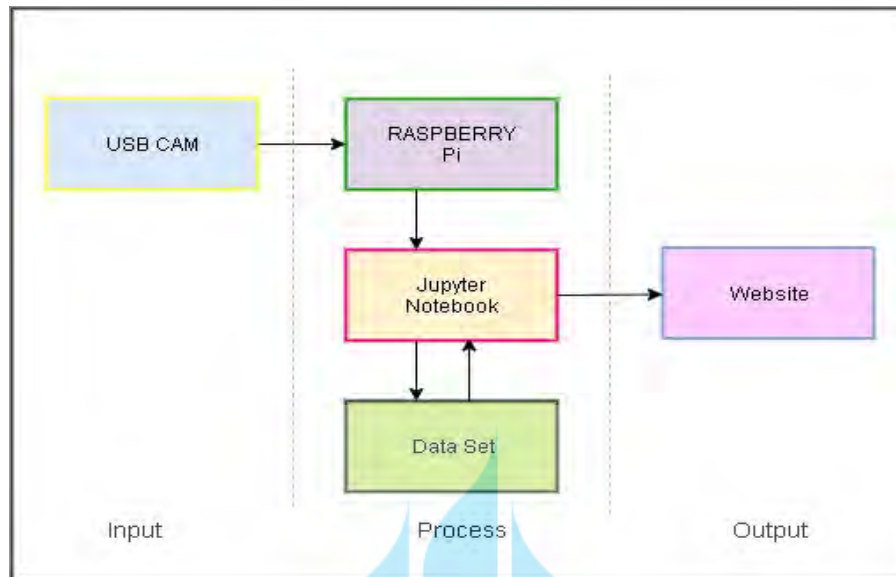
3.1 Gambaran Umum

Pada bagian ini akan dijelaskan mengenai perencanaan perangkat keras elektronik (*Hardware*). Sedangkan untuk pembuatan perangkat lunak dirancang menggunakan Bahasa pemograman Python dengan algoritma YOLO. Pada bab ini penulis akan membahas tentang:

1. Gambaran umum sistem.
2. Perancangan perangkat keras (*Hardware*).
3. Perancangan perangkat lunak (*Software*).

Secara umum sistem deteksi nomor plat ganjil genap terdiri dari Raspberry pi yang dihubungkan pada power. Kemudian *module* kamera yang terhubung dengan Raspberry pi diproses melalui *Jupyter Notebook*. Pada *Jupyter notebook*, algoritma mulai bekerja. Pertama-tama gambar akan di *resize* sesuai standar algoritma YOLO yaitu 320 x 320. Kemudian gambar baru akan diproses lebih lanjut. Yaitu plat nomor mobil akan diidentifikasi apakah platnya ganjil atau genap berdasarkan 2 angka pada nomor plat mobil. Kemudian di set tanggal apakah hari tersebut ganjil atau genap. Setelah itu hasil akan ditampilkan pada keluaran yaitu *website*.

3.2 Blok Diagram Sistem



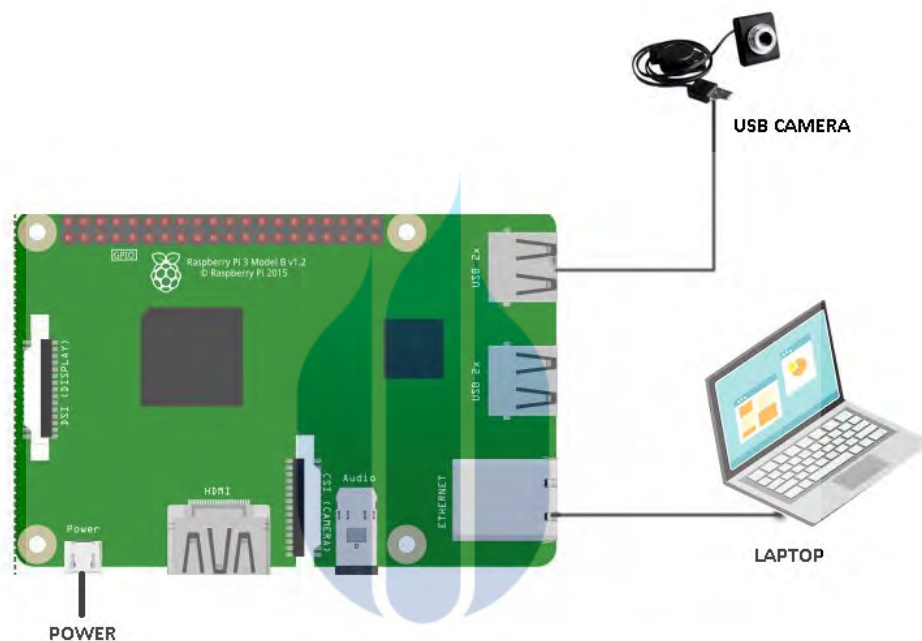
Gambar 3.1 Blok Diagram Sistem

Fungsi dari tiap blok dalam diagram adalah sebagai berikut:

1. USB Camera : *Module* yang akan meng *capture* citra atau gambar berupa video.
2. Raspberry Pi : Merupakan mikrokontroler yang akan menerima gambar dari module kamera kemudian diteruskan ke *Jupyter Notebook* melalui Wifi atau internet.
3. Jupyter Notebook : Pada *Jupyter Notebook* hasil gambar yang sudah di dapat akan diproses. Terlebih dahulu gambar akan di *Resize* menjadi 320 x 320 sesuai standar gambar dari algoritma YOLO kemudian *Jupyter Notebook* akan memproses gambar setelah mendapat *dataset*. Disini plat akan diidentifikasi apakah ganjil atau genap. Kemudian hasil dari pemrosesan gambar akan diteruskan kembali pada *website*.
4. Dataset : Berisi data-data yang menjadi model pre-trained data setelah dataset didapat, kemudian di kirim kembali ke *jupyter notebook*.
5. Website : berisi hasil keluaran dari segala pemrosesan pada *jupyter notebook*.

3.3 Perancangan Perangkat Keras (Hardware)

Pada perancangan perangkat keras meliputi perancangan Raspberry pi yang mendukung sistem kerja alat ini pada input dan output terdiri dari *module* kamera, Raspberry pi, dan laptop. Berikut tampilan skema rangkaian *hardware* sistem pendeteksi ganjil genap untuk elektronik tilang.



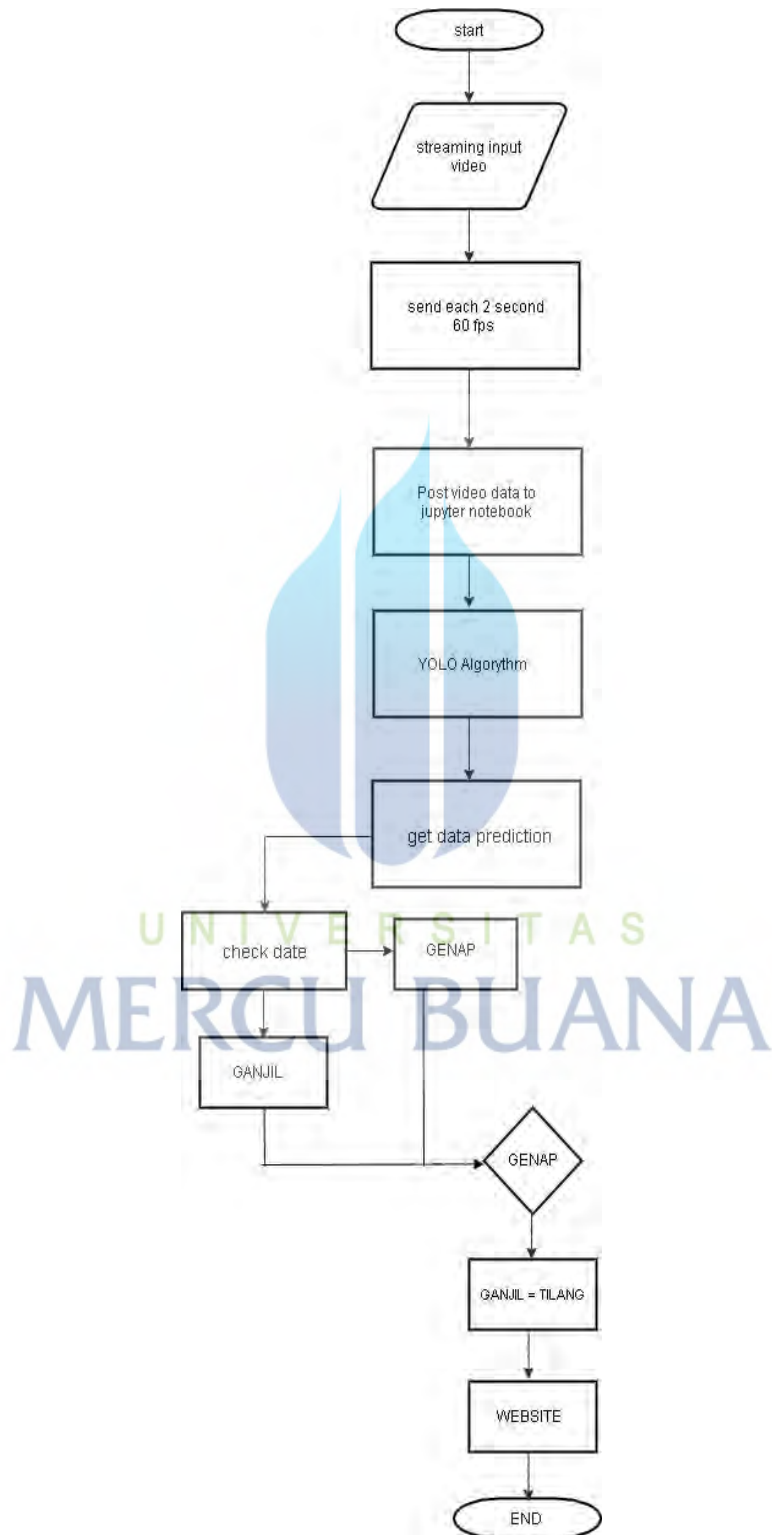
Gambar 3.2 Rangkaian sistem keseluruhan.

Dibawah ini merupakan tabel untuk pin-pin yang tersambung pada Raspberry Pi sebagai konektivitas data.

Tabel 3.1 Pin pada Raspberry pi

No	Nama	Pin
1	USB Kamera	USB
2	Laptop	Ethernet
3	Power	Power

3.4 Perancangan Perangkat Lunak (Software)



Gambar 3.3 Flowchart keseluruhan

Perancangan pada sisi *software* merupakan perancangan perangkat lunak yang akan diterapkan pada alat, ada beberapa tahapan dalam perancangan dan pembuatan sisi *software*. Untuk mengetahui proses cara kerja alar diperlukan *flowchart* sebagai garis ruang dari lingkup alat. *Flowchart* pada gambar diatas merupakan diagram proses aliran *input* hingga *output*. Mulai dari *module* kamera menangkap gambar berupa video hingga *output* pada *website*.

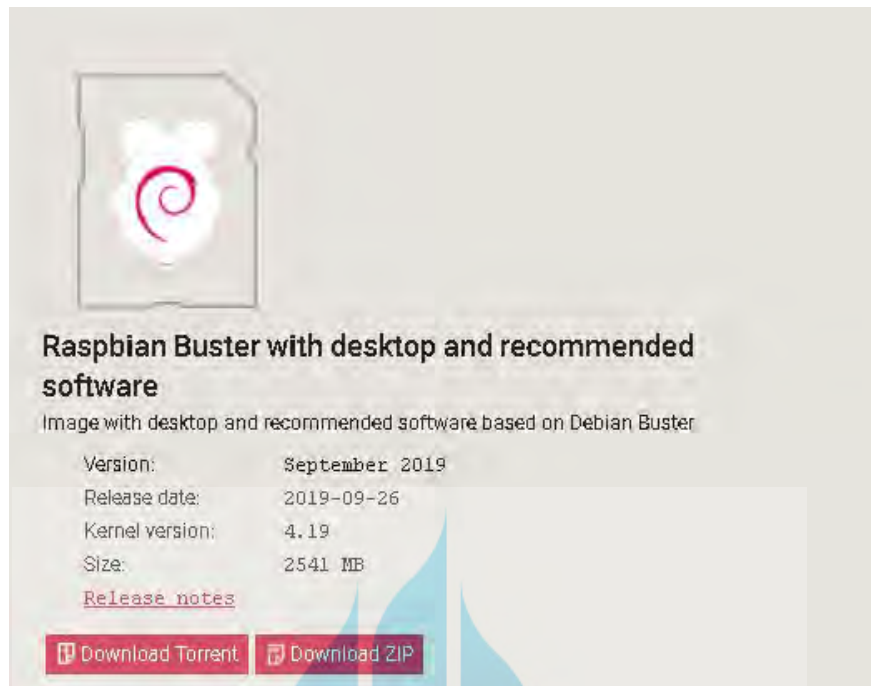
Dari *flow chart* diatas, jika diasumsikan sistem berjalan dengan baik maka proses kerja dari sistem adalah sebagai berikut:

1. Pertama *streaming input* video yaitu, kamera usb mengambil video.
2. Pada Raspberry akan diatur lama pengiriman sebuah video yaitu 60 fps untuk 2 detik atau 30 fps untuk 1 detik.
3. Setelah itu raspberry akan mengirimkan video pada *jupyter notebook*.
4. Pada *Jupyter Notebook* hasil gambar yang sudah di dapat akan diproses. Terlebih dahulu gambar akan di *Resize* menjadi 320 x 320 sesuai standar gambar dari algoritma YOLO.
5. Setelah itu YOLO akan mendapat dataset sebagai *model pre-trained*. Kemudian data diproses kembali apakah gambar plat ganjil atau genap.
6. Check date apakah tanggal ganjil atau genap.
7. Apabila tanggal genap dan plat ganjil maka akan kena tilang dan diteruskan hasilnya pada keluaran yaitu website.
8. Apabila tanggal genap tapi plat mobil terdeteksi genap maka tidak akan kena tilang.
9. Website akan menampilkan berupa keluaran hasil dari seluruh pemrosesan data yang telah terjadi.

3.4.1 Instal Raspberry

Untuk menginstal Raspberry dengan cara:

1. Buka website [raspberrypi.org/downloads](https://www.raspberrypi.org/downloads)
2. Pilih download zip seperti gambar dibawah ini



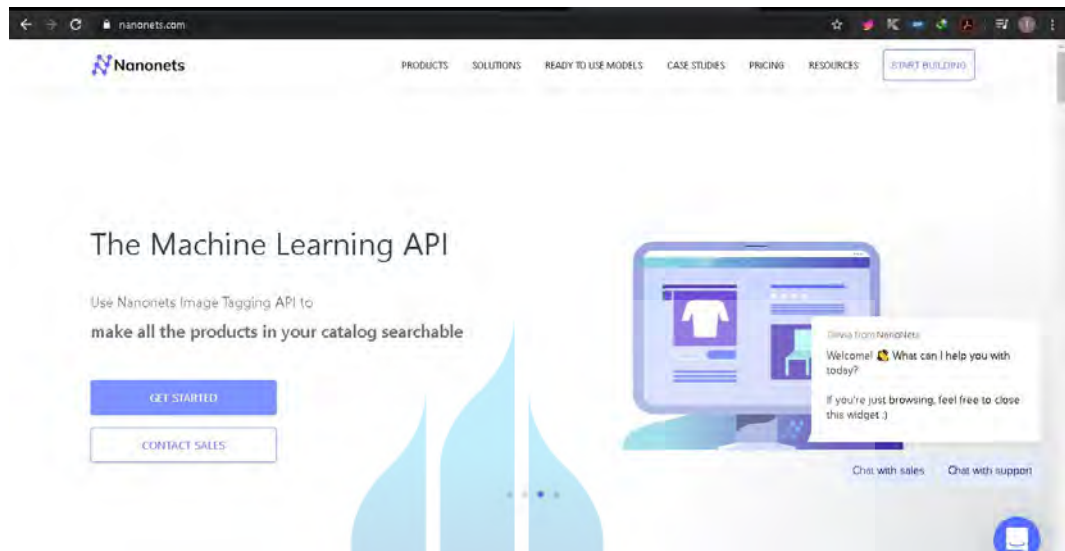
Gambar 3.4 Raspi Download

3. Download aplikasi etcher
4. Siapkan kartu sd dan hubungkan ke port usb. Kemudian buka etcher
5. Pilih OS Raspbian dan tekan flash. Tunggu proses flashing selesai.
6. Buka notepad dan setting wifi untuk raspi.
7. Hubungkan raspi (yang sudah dimasukkan kartu sd) dengan adaptor power dan tunggu 1-2 menit raspi booting.
8. Download aplikasi *angry ip scanner* dan hubungkan IP Raspberry.
9. Kemudian buka putty dan masukkan IP Raspi. Akan muncul notifikasi SSH klik *yes*.
10. Ketik user dan password raspberry (bawaan dari raspi).
11. Download VNC.
12. Ketikkan `sudo raspi-config` pada konsol putty, pilih *interfacing options*. Pilih VNC dan klik *yes*.
13. Buka aplikasi VNC, ketik IP Raspi dan isikan username dan password.

3.4.2 Memulai Nanonets

Untuk memulai training ada beberapa hal yang harus dilakukan, yaitu:

1. Pertama buka situs app.nanonets.com seperti gambar dibawah ini.



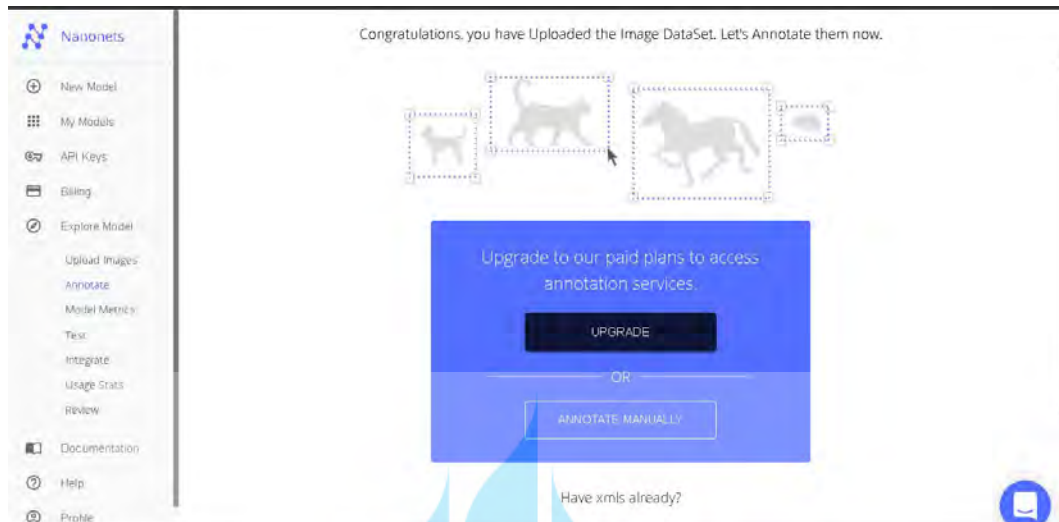
Gambar 3.5 Nanonets

2. Setelah membuka nanonets.com, beralih ke get started dan masuk ke akun anda.
3. Mulai upload images yang akan di *labelling* seperti gambar dibawah ini.



Gambar 3.6 Upload Images

4. Mulai *labelling* atau menganotasi gambar seperti yang terlihat pada gambar dibawah ini.



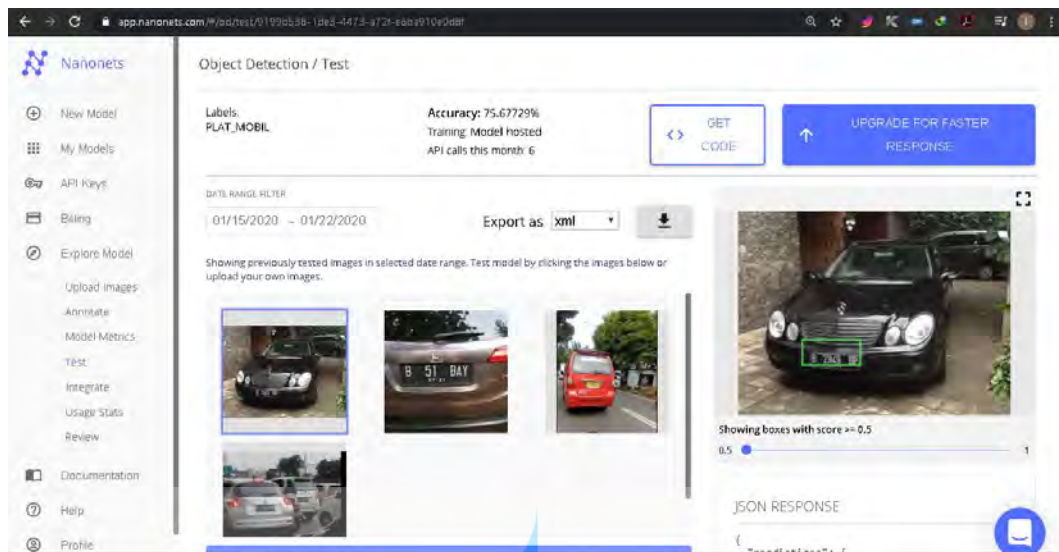
Gambar 3.7 Anotasi Gambar.

5. Setelah *labelling* dapat dilihat hasil dari *pre trained model* pada gambar dibawah ini.



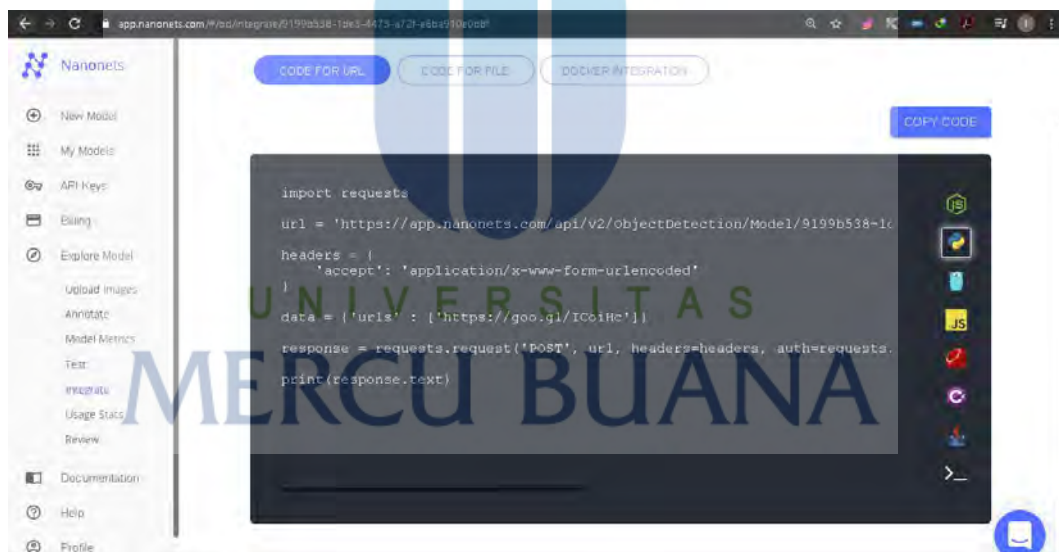
Gambar 3.8 Hasil Pre Trained.

6. *Step* selanjutnya yaitu memasukkan gambar yang akan di test ke dalam *pre trained model* seperti yang dapat dilihat pada gambar dibawah ini.



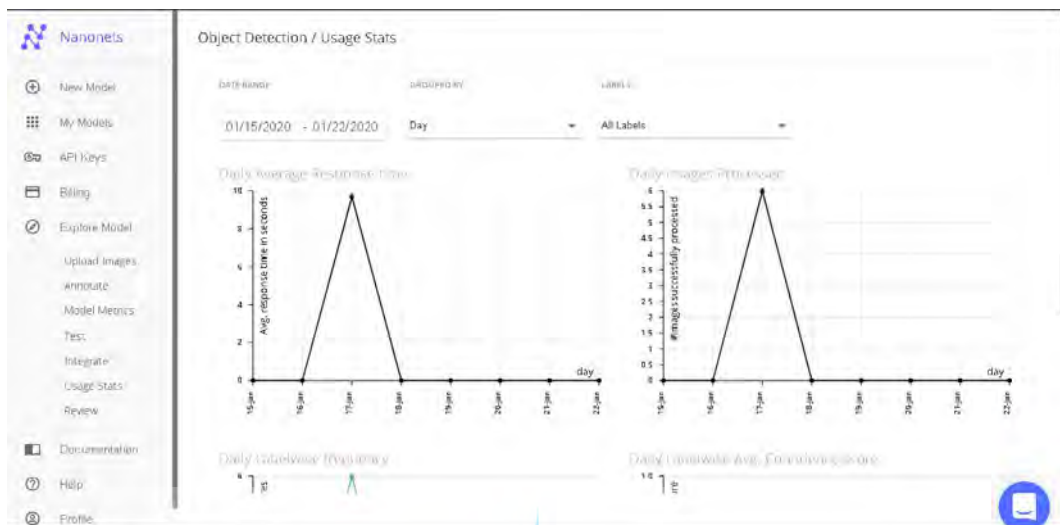
Gambar 3.9 Hasil Test Gambar

7. Setelah itu kita dapat mengintegrasikannya dengan raspberry yaitu dengan mendapatkan *python code* seperti pada gambar dibawah ini.



Gambar 3.10 Integrate Code

8. Selain *integrate code*, kita juga dapat melihat *usage stats* seperti gambar dibawah ini.



Gambar 3.11 Usage Stats

9. Selain *usage stats*, kita dapat melihat hasil *review* dari *pre trained model* kita. Yang dapat terlihat pada gambar dibawah ini.



Gambar 3.12 Review.