

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Kebutuhan Fungsional Sistem

a. Pendaftaran Wajah

- Kemampuan untuk mendaftarkan wajah baru ke dalam sistem.
- Proses pendaftaran harus cepat dan efisien.

b. Pengenalan Wajah dan Verifikasi Identitas

- Sistem harus dapat mengenali wajah manusia dengan tingkat akurasi yang tinggi.
- Sistem harus dapat memverifikasi identitas seseorang berdasarkan wajah yang terdaftar.

c. Pengelolaan Basis Data Wajah

- Kemampuan untuk menyimpan dan mengelola basis data wajah dengan aman. Pembaruan dan penghapusan data wajah harus mudah dilakukan.

d. Pemantauan Wajah Secara Real-Time

- Pemantauan wajah secara real-time untuk mendeteksi perubahan atau ancaman potensial.

e. Keamanan Data

- Sistem harus memiliki lapisan keamanan yang kuat untuk melindungi data wajah pengguna.

3.2. Kebutuhan Non Fungsional Sistem

a. Pendaftaran Wajah

- Respons cepat terhadap permintaan pendaftaran dan identifikasi wajah.
- Kemampuan untuk menangani volume data wajah yang besar.

b. Ketersediaan & Kompabilitas

- Sistem harus tersedia sepanjang waktu dengan waktu downtime minimal.
- Kompabilitas dengan perangkat keras dan perangkat lunak yang umum digunakan.

3.3. Usecase Diagram Gambaran Umum Sistem

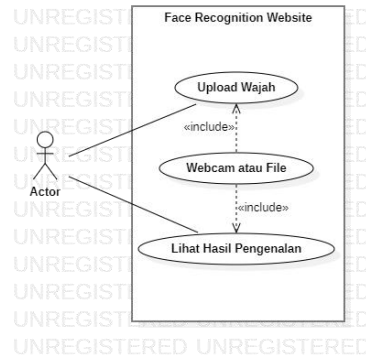


Fig. 20 Gambaran Umum Sistem

Pada gambaran umum sistem atau secara garis besar actor akan dihadapkan untuk memilih melakukan upload wajah (men-training) atau testing.

3.4. Class Diagram Perancangan Database

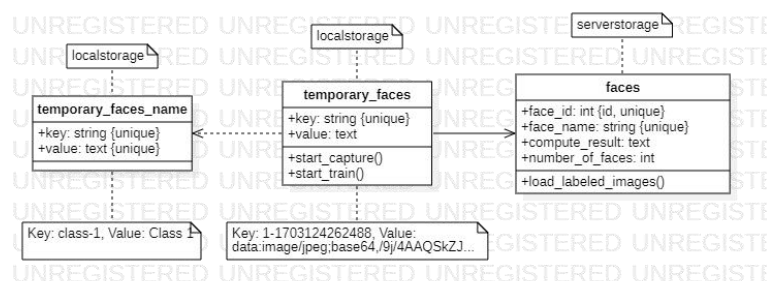


Fig. 21 Perangan Database Face Recognition

Pada perancangan database terdapat penggunaan secara local dan server, dimana local menggunakan localStorage bawaan dari browser dan server menggunakan MySQL. Sebelum data citra/gambar wajah dikirim ke server data wajah disimpan dulu di localStorage ini berguna untuk misalnya mengupload lebih dari 1 wajah individu, baik wajah individu yang berbeda atau satu individu namun berbeda variasi dan lain sebagainya. Setelah actor yakin untuk melakukan training pada gambar yang disimpan dengan menekan tombol “train” maka semua data yang tersimpan di localStorage akan diproses dengan dilakukan konvolusi

dan hasil konvolusi tersebut lah yang disimpan ke server (dengan nama field nya `compute_result`) jadi bukan gambar utuh yang disimpan. Sehingga, ketika melakukan testing nanti, sistem tidak perlu melakukan konvolusi kembali terhadap wajah yang tersimpan, hanya mengambil hasil konvolusi dari server dan sistem siap untuk mengenali data wajah yang diberikan.

3.5. Flowchart Diagram Simpan Gambar Sementara Pada Local Storage

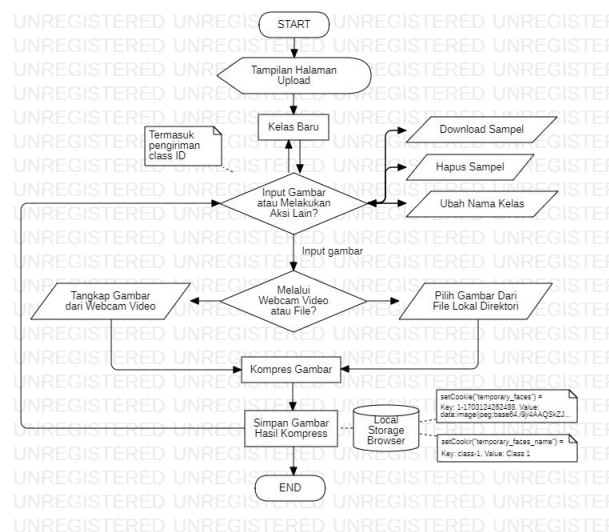


Fig. 22 Flow Gambaran Penyimpanan Sementara Data Gambar Pada Local Storage

Ini adalah alur dari process penyimpanan data citra atau gambar ke localstorage, di mana diawali dengan menampilkan halaman upload (training), kemudian secara sistem, sistem akan membuatkan class baru (untuk wajah siapa) sedangkan actor memilih aksi seperti: input wajah, download sample, hapus sample atau ubah nama class. Dan langsung saja misal actor memilih untuk upload data wajah maka sistem akan memberikan pilihan kembali seperti, ingin menggunakan video wabcam atau secara upload file/gambar. Lalu selanjutnya, baik upload ataupun melalui video webcam, gambar yang diupload, secara sistem, sistem akan melakukan kompresi gambar, hal ini sangat penting untuk mengurangi ukuran sehingga data gambar yang disimpan tidak begitu besar, seperti dilakukan cropping sedikit di bagian atas, kanan, bawah dan kiri dan dikurangi juga kualitas gambarnya, lalu setelah itu sistem akan menyimpan hasil kompres tersebut ke localstorage cookie yang diberi nama `temporary_faces` (untuk data

gambar) dan cookie temporary_faces_name (untuk nama class/wajah nya). Setelah itu balik lagi seperti yang disebutkan sebelumnya seperti actor ingin melakukan input wajah, download sample, hapus sample atau ubah nama class.

3.6. Flowchart Diagram Upload Citra Wajah

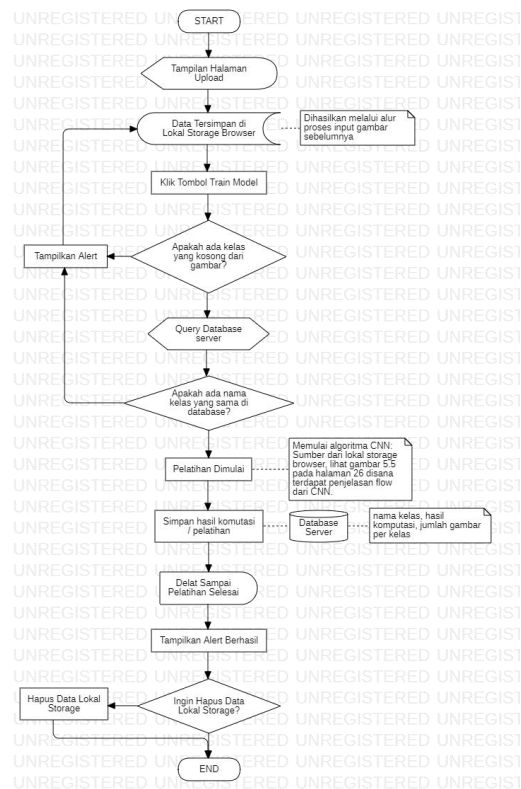


Fig. 23 Flow Upload Gambar / Citra ke Server

Ini adalah alur ketika actor sudah yakin untuk meng-upload (men-training sistem) pada data wajah yang disimpan sebelumnya, dimulai dari setelah klik tombol “train” sistem awal pertama kali akan mengecek apakah ada class yang kosong dari gambar atau tidak, jika ada maka sistem akan memberikan alert dan jika semua class suka terisi dengan gambar maka selanjutnya sistem akan melakukan query ke server untuk mengambil data nama class (nama wajah) yang sudah tersimpan sebelumnya, jika nama class sama seperti data nama wajah yang tersimpan maka sistem akan memberikan sebuah alert kembali untuk mengganti nama class tersebut. Lalu jika class sudah tidak ada yang kosong dan nama class pun sudah tidak ada yang sama dengan database nama wajah yang tersimpan

maka selanjutnya sistem akan melanjutkan proses nya ke “Pelatihan” dimana disini data wajah yang di *localStorage* akan dikirimkan ke model *library face-api.js* untuk dilakukan konvolusi atau library tersebut menamai sebagai mengambil deskripsi, contohnya menajdi seperti ini:

```
[
  [
    -0.07291821390390396,
    0.14813220500946045,
    0.07849273830652237,
    ...
  ],
  [
    -0.05436752736568451,
    0.13666945695877075,
    0.07281330227851868,
    ...
  ]
]
```

Kemudian hasil ini yang dilakukan oleh *library face-api.js*, hasil ini akan disimpan ke tabel *faces* dengan nama fieldnya *compute_result*. Setelah library berhasil melakukan tugasnya dan data hasilnya juga sistem berhasil menyimpan dan dengan mungkin akan delay beberapa detik atau mungkin menit sesuai seberapa banyaknya data yang dilatih dan oprasi computer actor itu sendiri, sistem kemudian akan memberikan alert success dan sistem akan memberikan juga tombol yang intinya “data ini semua ingin dihapus atau tidak”, jika ya maka sistem akan menghapus semua data yang berada di *localStorage*, dan selesai. Begitulah proses sistem dalam meyimpan data wajah ke server, untuk sistem dapat melakukan pengenalan nanti pada wajah yang diberikan.

3.7. Flowchart Halaman Pengenalan Wajah Pada Citra

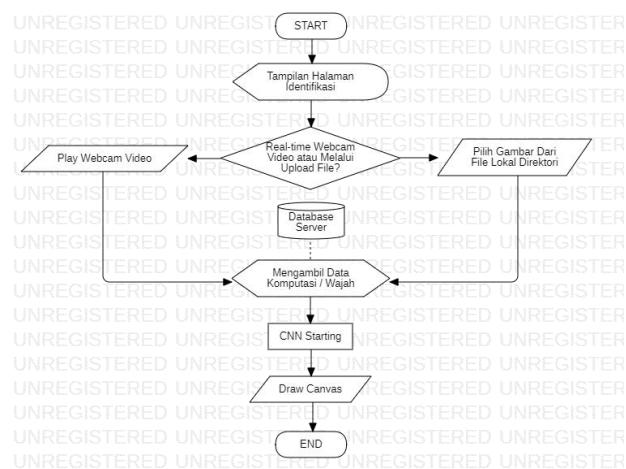


Fig. 24 Flow Sistem Melakukan Prediksi Nama Wajah Yang Diberikan

Sekarang ini adalah proses dimana sistem melakukan pengenalan wajah pada wajah yang diberikan, baik melalui video webcam atau file upload. Pertama-tama actor akan dihadapkan dengan pilihan melalui video atau file upload, keduanya saya saja prosesnya. Maka selanjutnya, setelah sistem mendapatkan data gambar, sistem akan melakukan query terlebih dahulu untuk mengambil “*compute_result*” dimana field tersebut adalah hasil konvolusi sebelumnya. Lalu, model *CNN* milik nya *library face-api.js* mulai bekerja, dimana *CNN* disini memiliki beberapa tugas seperti: memprediksi wajah, memprediksi landmark wajah dan terakhir menyamakan hasil *compute* dengan *compute* yang berada di database. Setelah proses tersebut selesai, sistem akan men-draw kotak dan nama wajah nya, baik yang terindikasi atau unknow, jika unknow maka sistem akan menulis unknow dan sebaliknya.

3.8. Flowchart Diagram CNN Dalam Kostumisasi Dataset

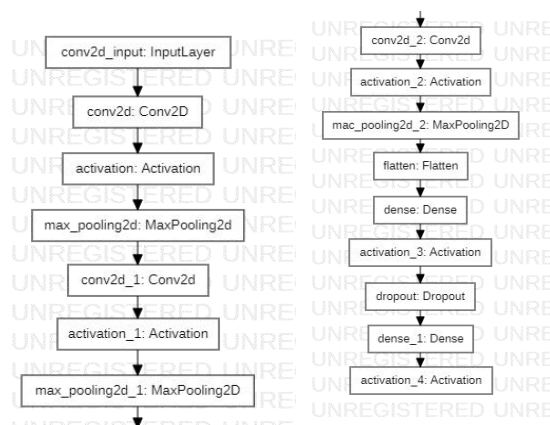


Fig. 25 Gambar Tambahan Untuk Flow CNN Bekerja

Ini adalah hanya contoh bagaimana proses CNN bekerja dan alur ini tidak ada hubungan dengan *library face-api.js*. Ini hanya permisalahan, intinya pada CNN itu ada conv, activation, pooling, flatten, dropout, lapisan dense dan lain sebagainya, hal-hal ini bisa diatur sesuai dengan kebutuhan yang ingin dipecahkan, jadi tidak harus sama seperti digambar.