BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN

3.1 Metodologi Penelitian

Metodologi penelitian dalam penyusunan tugas akhir ini dibagi menjadi tujuh tahap, yaitu tahap wawancara, pengumpulan data, tahap perancangan, tahap pembangunan *software*, tahap uji coba beserta perbaikan, evaluasi, dan penulisan laporan.

a. Telaah Literatur

Tahap ini dilakukan dengan mempelajari teori-teori dengan mencari literatur, jurnal, *paper*, dan bacaan-bacaan yang berhubungan dengan aplikasi yang akan dirancang dan dibangun.

b. Tahap Perancangan

Tahap ini dimulai dengan analisis kebutuhan yang akan diperlukan dalam pembangunan aplikasi. Pada tahap ini dilakukan perancangan aplikasi. Perancangan aplikasi yang dimaksud, yaitu perancangan *flowchart* dan desain antarmuka.

c. Tahap Implementasi

Dari hasil perancangan yang telah dibuat, diimplementasikan pada aplikasi yang telah dibangun yang berbasis *website*, adapun pemrograman Faster Region-based Convolutional Neural Network dilakukan dengan memanfaatkan Object Detection API dari Tensorflow. *Dataset* yang digunakan enam ratus *dataset* yang telah *grayscale*, dan enam ratus *dataset* yang telah *grayscale* dan *noise*. Data *training* dan *testing* yang digunakan yaitu 80:20 dari keseluruhan *dataset* yang ada.

d. Tahap Uji Coba dan Perbaikan

Setelah pembangunan aplikasi selesai, maka akan dilaksanakan pengujian terhadap aplikasi yang telah dibuat. Sistem diuji menggunakan data test yang telah disediakan pada penelitian. Dari hasil *test* menggunakan *confusion matrix* dengan menghitung nilai akurasi, presisi, recall yang telah dilakukan maka akan dilakukan perbaikan agar mendapatkan hasil yang maksimal. Skenario *testing* menggunakan data yang berbeda dari *dataset training* dari perbandingan 80:20 untuk data *training* dan *testing*.

e. Tahap Penulisan Laporan

Setelah aplikasi dikira memenuhi kriteria, maka akan melakukan pencatatan dari hasil uji coba. Seluruh hasil akan dicatat dan dihitung nilai keberhasilannya mengenai akurasi, presisi, *recall*, dan F-measure dari algoritma Faster Region-based Convolutional Neural Network pada aplikasi klasifikasi bentuk pada iStar 2.0 menggunakan *confusion matrix*.

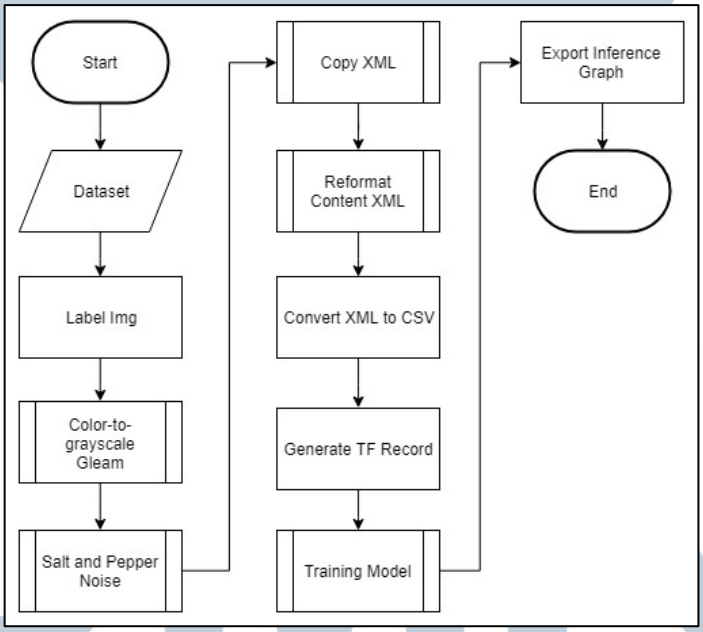
3.2 Perancangan Aplikasi

Perancangan aplikasi yang dilakukan menghasilkan model berupa *flowchart* dan perancangan antarmuka.

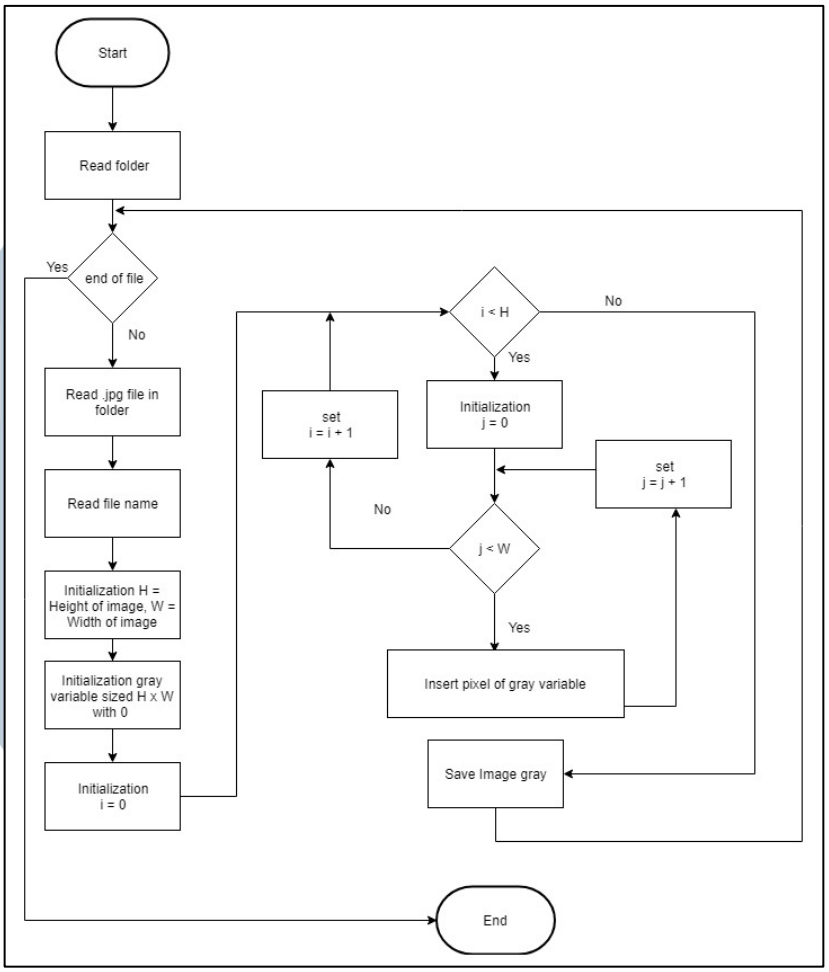
3.2.1 Flowchart

Flowchart merupakan diagram yang menunjukkan alur kerja program. Gambar 3.1 merupakan *flowchart* aktivitas training yang dilakukan untuk mendapatkan model. Pada langkah pertama dilakukan pengambilan dataset sebagai alat pembelajaran dari sebuah model. Dataset yang diambil sebanyak 600 data yang terdiri dari 5 kelas yaitu *goal*, *quality*, *actor*, *task*, dan *resource* yang digambar oleh tangan. Terdapat 2 jenis alat penulisan yaitu bolpoin dan spidol. Langkah

selanjutnya yang dilakukan adalah melabelkan setiap gambar sesuai dengan kelas dari gambar tersebut. *Tools* yang digunakan yaitu LabelImg. pada program tersebut dilakukan peletakan posisi dimana object dari gambar tersebut berada dan memberikan nama kelas dari object tersebut lalu hasil dari LabelImg berupa XML yang berisikan nama file, *path*, dan detail dari posisi object tersebut berada. Pada proses Convert XML to CSV adalah proses dimana setiap XML yang ada dicatat dalam ke dalam bentuk CSV. Pada proses Generate TF Record adalah proses yang membutuhkan CSV yang telah didapatkan dari proses Convert XML to CSV dan folder gambar yang sesuai. Pada proses Export Inference Graph adalah proses dimana pembuatan model dilakukan dari hasil Training Model.



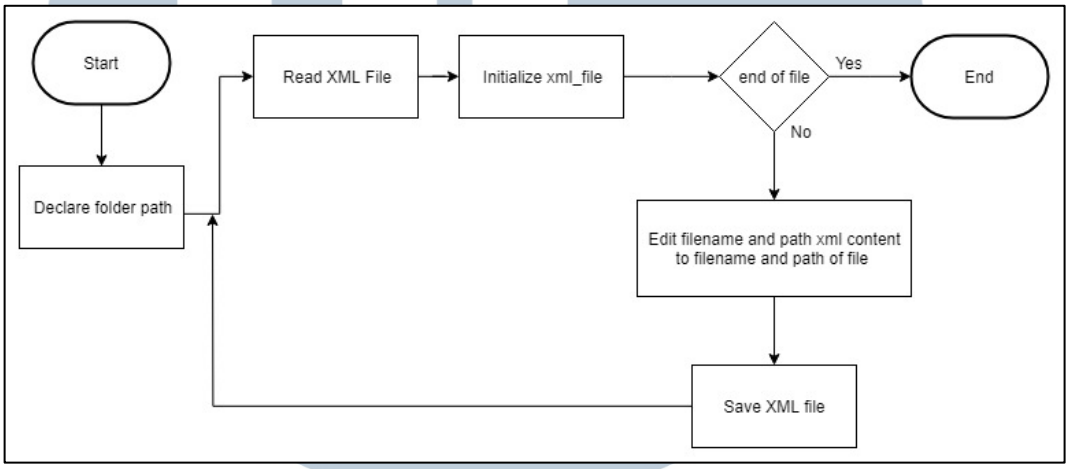
Gambar 3.1 Flowchart Aktivitas Training



Gambar 3.2 Flowchart color-to-grayscale Gleam

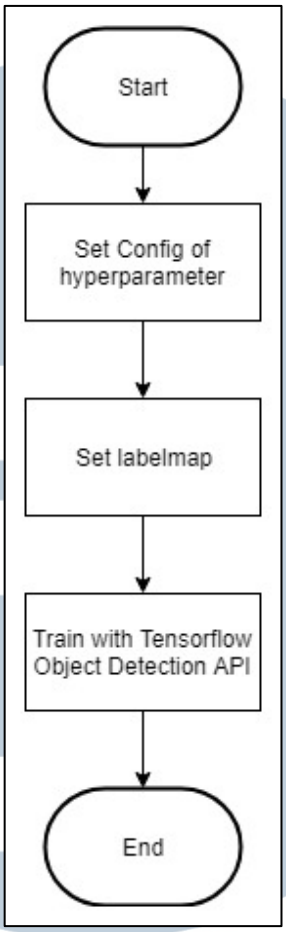
Gambar 3.2 menggambarkan *flowchart color-to-grayscale* Gleam. Proses ini akan memproses gambar dataset menjadi warna abu-abu. Pada proses ini secara otomatis menambahkan gambar yang telah di grayscale terhadap semua gambar yang terdapat dalam folder. Pada proses Insert pixel of gray variable dilakukan pengisian pixel pada variable gray yang telah di inisialisasi dengan ukuran sesuai gambar asli sesuai dengan yang terdapat pada proses Initialization gray variable sized H x W x C with 0. Metode grayscale yang digunakan yaitu Gleam. Pada pengisian pixel variable gray diisi dengan hasil pixel (red'[i][j] + green'[i][j] + blue'[i][j]) / 3 dan disesuaikan dengan channel dari gambar asli. Arti dari red', green', dan blue' adalah telah dilakukannya *gamma correction* pada masing-masing

variable. Setelah dilakukan pengisian pixel pada variable gray hingga pixel pada variable gray sesuai dengan panjang dan lebar dari gambar asli yang sedang dibaca maka gambar akan disimpan dengan nama sesuai dengan nama dari gambar asli ditambah "_grayscale". Setelah semua gambar asli pada folder telah dilakukan *color-to-grayscale* maka proses ini telah selesai.



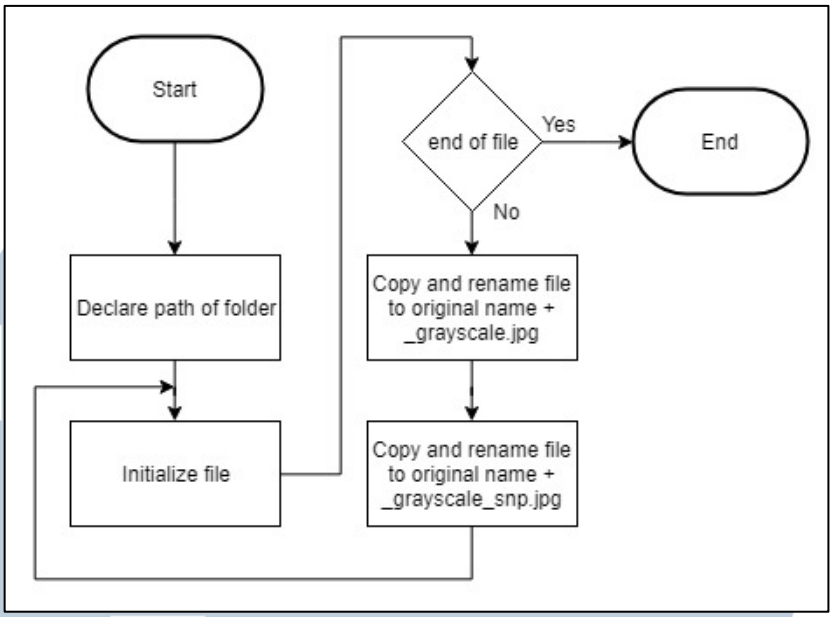
Gambar 3.3 Flowchart Reformat Content XML

Gambar 3.3 menggambarkan *flowchart reformat content* XML. Pada proses ini semua file XML pada folder yang telah ditentukan akan dibaca dan diproses. Proses yang dilakukan yaitu mengubah isi dari file XML yang telah dibaca. Pengubahan isi dari file XML yang telah dibaca adalah pada bagian filename dan path. Dilakukan penyesuaian pada setiap filename dari file XML sesuai dengan nama dari file XML. Setelah dilakukan penyesuaian maka file XML akan disimpan kembali. Setelah semua file XML dalam folder yang telah dilakukan selesai maka proses ini telah selesai.



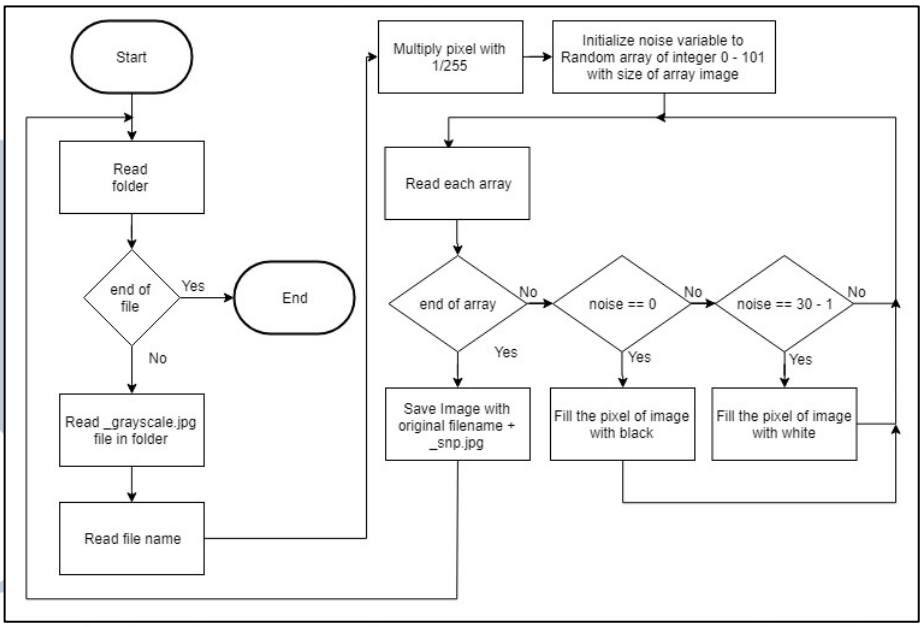
Gambar 3.4 Flowchart Training Model

Gambar 3.4 menggambarkan *flowchart training model*. Pada proses ini akan dilakukan *training* sesuai dengan konfigurasi yang telah diatur. Lalu labelmap berisikan nama-nama kelas yang ditentukan untuk proses pembelajaran. Proses training dilakukan oleh Tensorflow Object Detection API dengan konfigurasi dan labelmap yang telah ditentukan.



Gambar 3.5 Flowchart Copy XML

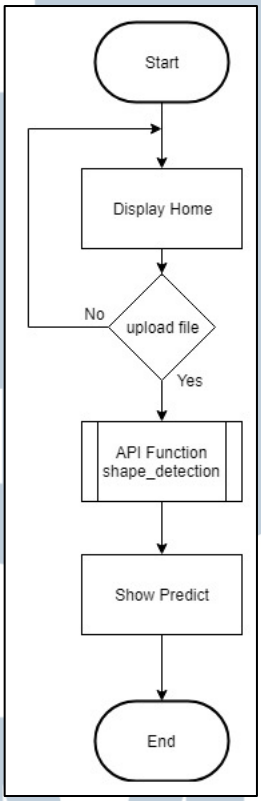
Gambar 3.5 menggambarkan *flowchart copy* XML. Pada proses ini akan dilakukan duplikasi pada file XML yang ada dan terdapat perbedaan nama menjadi nama asli dari file ditambah dengan "_grayscale.jpg" dan "_grayscale_snp.jpg".



Gambar 3.6 Flowchart Salt and Pepper Noise

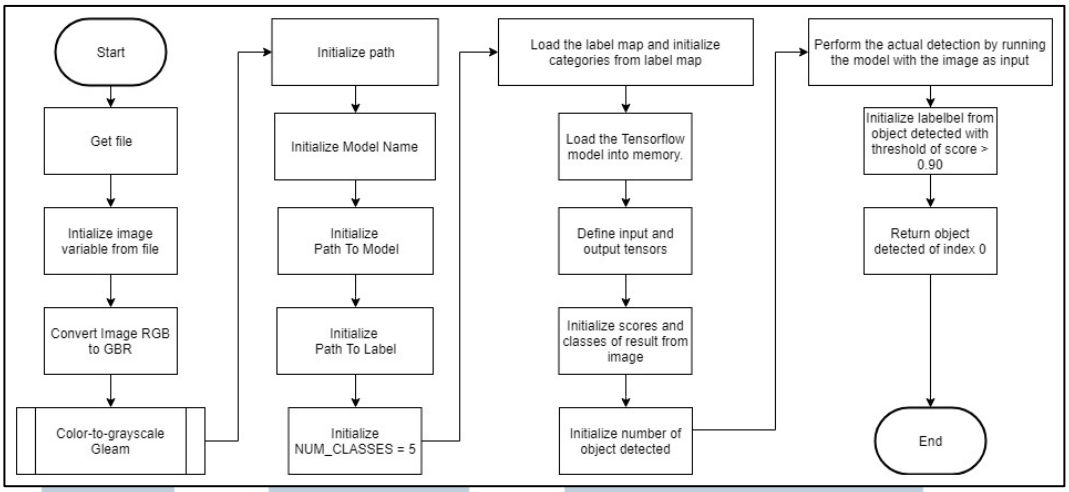
Gambar 3.6 menggambarkan *flowchart* Salt and Pepper *noise.* Pada proses ini dilakukan penggantian beberapa pixel dari angka acak menjadi warna hitam dan

putih. Proses ini dilakukan dengan menggunakan sebuah array yang serupa dengan array dari gambar dan diberikan angka acak. Jika angka acak tersebut adalah minimum dari angka acak yang telah ditentukan maka pixel itu akan diberikan warna putih. Sedangkan jiga angka itu berupakan angka maximal dikurang 1 dari angka yang telah diacak maka pixel pada gambar akan diberikan warna hitam.



Gambar 3.7 Flowchart Testing

Gambar 3.7 menggambarkan *flowchart testing.* Pada proses ini tampilan utama akan memiliki tempat untuk menunggah file gambar. Setelah *user* mengunggah gambar yang ingin diklasifikasikan, proses langsung dilakukan oleh API yang berfungsi untuk mengklasifikasikan gambar. Setelah diproses oleh API maka hasil akan ditampilkan.

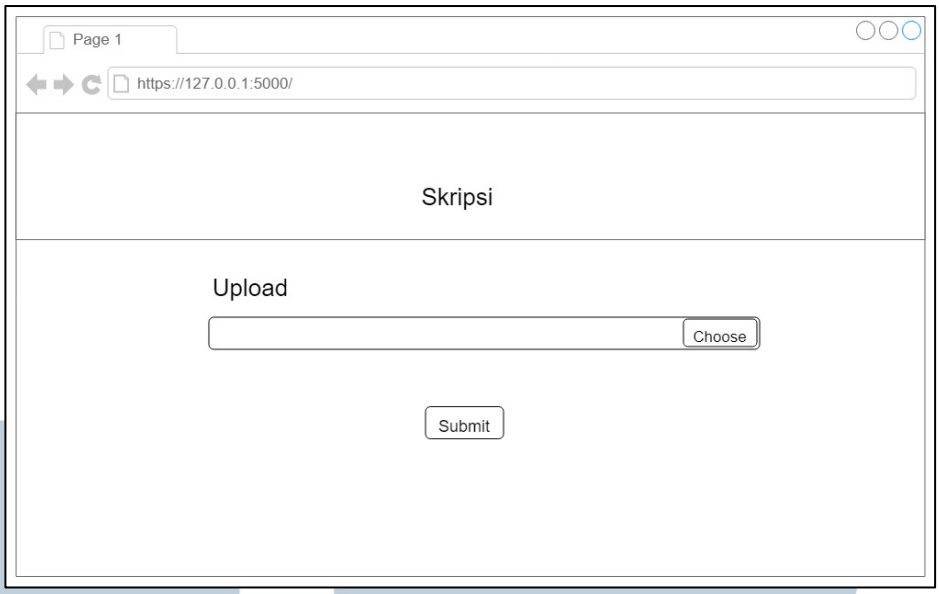


Gambar 3.8 Flowchart API Function shape_detection

Gambar 3.8 menggambarkan *flowchart* API *function shape_detection*. Pada proses ini gambar dikirim dari *website* yang telah dibuat lalu file yang dikirim akan dibaca sebagai gambar oleh API. Gambar yang dikirim akan dikonversikan dari tipe gambar RGB menjadi GBR. Lalu gambar diproses untuk dilakukan *color-to-grayscale* terlebih dahulu. Selanjutnya inisialisasi *path* akan dilakukan agar API mengetahui keberadaan dari file yang dibutuhkan seperti *path* dari model dan *path* dari label. NUM_CLASSES menunjukkan ada berapa kelas pada objek yang akan ditebak. Kelas dari objek yang ditebak akan diambil yang memiliki nilai lebih dari 0,9. Setelah didapatkan nilai yang berhasil ditebak lalu index ke 0 akan diberikan sebagai nilai yang ditebak.

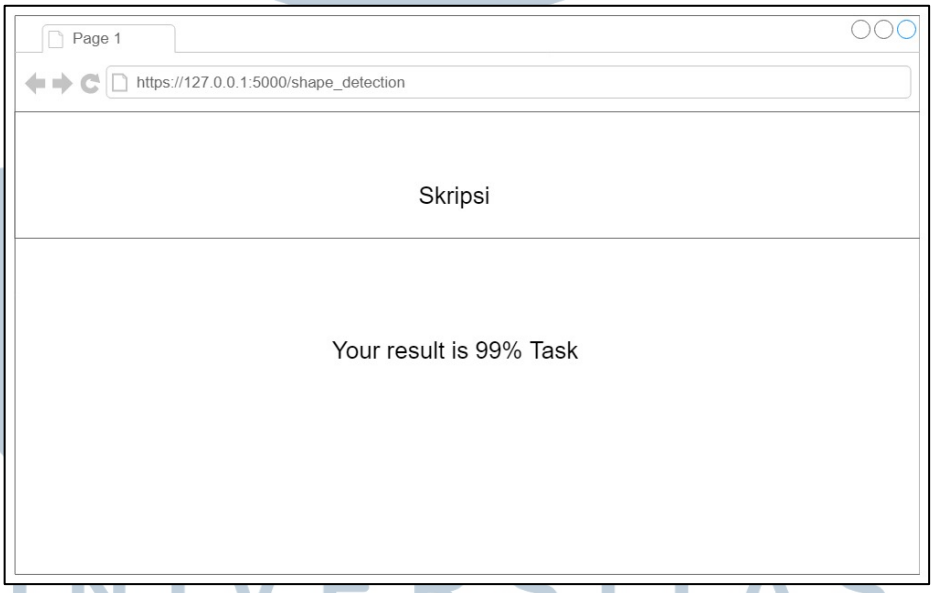
3.2.2 Perancangan Antamuka Pengguna

Perancangan antar muka *website* klasifikasi bentuk diagram iStar 2.0.



Gambar 3.9 Rancangan antarmuka home

Gambar 3.9 munjukan rancangan antarmuka home. Dalam rancangan antarmuka ini memiliki form untuk upload gambar. Ketika tombol submit ditekan maka gambar akan langsung diklasifikasikan.



Gambar 3.10 Rancangan antarmuka result

Gambar 3.10 rancangan antarmuka *result*. Rancangan antarmuka ini memiliki hasil pemrosesan klasifikasi yang dilakukan oleh API.