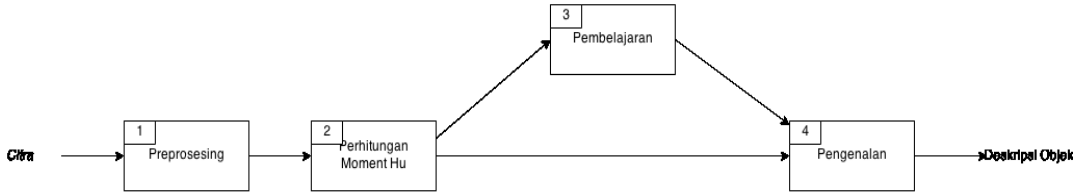
BAB 3
ANALISIS DAN KEBUTUHAN ALGORITMA

3.1. Analisis Masalah

Neural network backpropagation merupakan salah satu Algoritma *Machine Learning* yang dapat mengenal pola dan mengidentifikasi suatu objek citra berdasarkan pelatihan yang diberikan. Akan tetapi dalam sebuah citra digital objek tersebar memungkinkan memiliki posisi yang berbeda – beda, selain itu objek tersebar dalam sebuah citra digital memungkinkan memiliki beragam perbedaan translasi (*translation*), mempunyai beragam perbedaan rotasi (*rotation*), serta penskalaan citra (*scaling*). Implementasi neural network ketika digunakan untuk citra yang memiliki beragam translation, penskalaan dan rotasi tanpa dimanipulasi terlebih dahulu, memungkinkan terjadinya proses pembelajaran yang memerlukan jumlah sampel yang cukup banyak bahkan memungkinkan terjadinya kegagalan dalam pengambilan keputusan mengenali objek. Sebagai solusi perbaikan algoritma yang memungkinkan dari masalah tersebut yakni *Hu Invariant Moments*. Nilai-nilai yang dihasilkan dari *Hu Invariant Moments* ini terdiri dari tujuh nilai yang menidentifikasikan ciri dari sebuah objek citra digital. Nilai-nilai tersebut bersifat *independen* terhadap translasi, rotasi dan penskalaan. Nilai-nilai ini dapat dimanfaatkan sebagai satu sampel informasi tiap objek untuk inputan dari Machine Learning guna mengenali suatu objek.

3.2. Analisis Sistem

Dalam penyelesaian masalah yang telah dijabarkan pada analisis masalah, proses yang dilakukan dapat dikelompokan pada empat tahapan utama. Tahapan proses tersebut dapat dilihat pada Gambar 3.1.



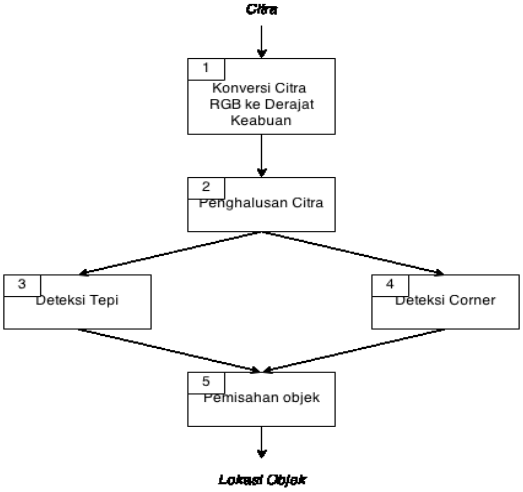
Gambar 3.1 Empat Proses Utama Sistem

3.2.1. Preprosesing

Citra penuh yang dijadikan masukan data diproses mulai dari dikonversi citra RGB ke citra keabuan, memperbaiki kualitas citra, deteksi tepi, deteksi corner, pemisahan objek-objek citra dari latar belakang.

Keluaran dari proses ini adalah lokasi objek-objek citra yang nantinya digunakan untuk informasi kepada user tentang objek citra mana yang akan diolah oleh sistem.

Untuk lebih jelasnya tahapan preprosesing dapat dilihat pada Gambar 3.2.



Gambar 3.2 Alur Proses Preprosesing Citra

1. RGB ke derajat keabuan

Citra digital inputan yang memiliki tiga chanel warna RGB dikonversi ke satu chanel warna keabuan bertujuan untuk mengurangi beban proses sistem. Proses ini menggunakan rumus (2.2).

2. Penghalusan Citra

Citra keabuan kemudian diperhalus untuk menanggulangi noise pada citra. Penghalusan citra dilakukan dengan menggunakan kernel gaussian.

3. Deteksi Tepi

Pendeteksian tepi pada citra menggunakan algoritma canny.

4. Deteksi *Corner*

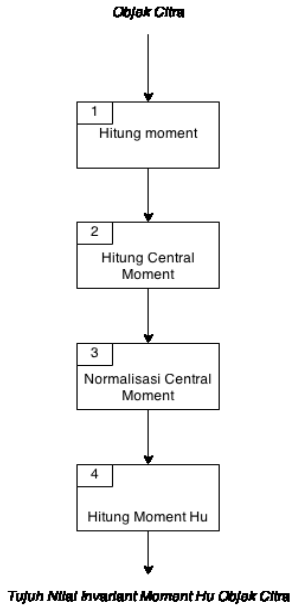
Pendeteksian *corner* pada citra menggunakan algoritma harris.

5. Pemisahan objek

Disini tepi dan *corner* yang telah didapat digunakan untuk ekstraksi ciri objek yang akan diolah. Objek-objek pada citra yang akan diolah hanya objek yang memiliki empat *corner* dan keempat *corner* saling terhubung oleh tepi objeknya (kotak).

3.2.2. Perhitungan Hu Invariant Moments

Setelah lokasi objek-objek ditemukan objek dihitung ketujuh nilai momentnya guna mendeskripsikan ciri-ciri objek citra. Tahapan proses ini dapat dilihat pada Gambar 3.3.



Gambar 3.3 Alur Proses Perhitungan Hu Invariant Moments

1. Hitung moment

   Hitung moment yang mentransformaskan fungsi citra pada sistem diskrit.

2. Hitung central moment

   Setelah tahap pertama dilakukan hitung moment pusat objek citra. Tahap ini menghasilkan tujuh nilai moment pusat citra.

3. Normalisasi central moment

   Setelah moment pusat ditemukan langkah selanjutnya moment pusat citra dinormalisasi.

4. Hitung ketujuh Moment Hu

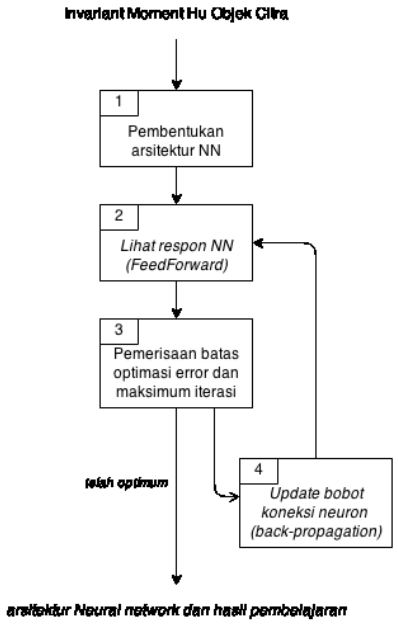
Setelah dinormalisasi ketujuh moment hu dapat dihitung.

3.2.3. Pembelajaran

Dalam tahapan ini user dapat memberikan label kesetiap objek selain itu dilakukan proses pembentukan *neural network,* proses pembelajaran neural network dan dalam tahapan ini memungkinkan untuk menyimpan arsitektur beserta hasil proses pembelajaran ke dalam file sehingga dapat digunakan dikemudian hari.

Keluaran dari proses ini adalah arsitektur beserta hasil proses pembelajaran neural network. Arsitektur beserta hasil proses pembelajaran neural network ini akan digunakan dalam proses selanjutnya yakni pengenalan objek.

Untuk lebih jelasnya tahapan pembelajaran dapat dilihat pada Gambar 3.4



Gambar 3.4 Alur Proses Pembelajaran

1. Pembuatan arsitektur neural network

Pembentukan *neural network* yang terdiri dari input layer yang beserta tujuh neuron, beberapa hidden layer beserta neuronnya sesuai inputan user dan hidden layer dengan jumlah neuron yang sesuai dengan jumlah objek citra yang akan di proses.

2. Lihat Respon NN *(Feed-Forward)*

Ketujuh moment Hu objek citra dimasukan kedalam arsitektur *neural network* yang telah dibuat guna dapat dilihat respon keluaran neural network terhadap masukannya.

3. Pemeriksaan batas optimasi error dan maksimum iterasi

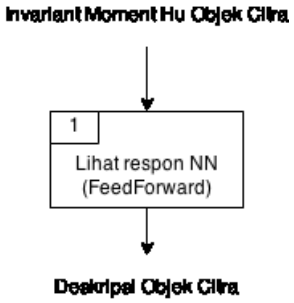
Dalam tahap ini respon *neura network* dihitung *error* responnya menggunakan fungsi *Root Mean Square Error* (RMS) serta jumlah iterasinya. Jika batas error ataupun batas maksimal iterasi belum terpenuhi maka lakukan tahap 4. Akan tetapi jika batas error ataupun batas maksimum iterasi sudah terpenuhi proses pembelajaran dihentikan.

4. Update bobot koneksi neuron *(Back-Propagation)*

Dalam proses ini neural network dilakukan proses pembelajaran menggunakan *backpropagation*. Proses pembelajaran disini berarti bobot koneksi akan diupdate berdasarkan respon selisih error yang didapatkan.

3.2.4. Pengenalan

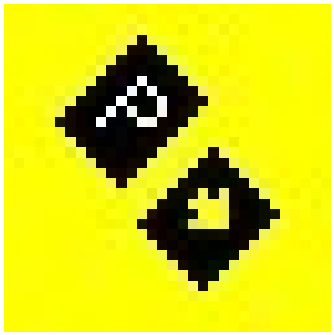
Dalam tahapan ini nilai invarian moment objek-objek dimasukan satu persatu dan dilihat respon dari neural network. Alur proses pengenalan dapat dilihat pada Gambar 3.5.



Gambar 3.5 Alur proses pengenalan

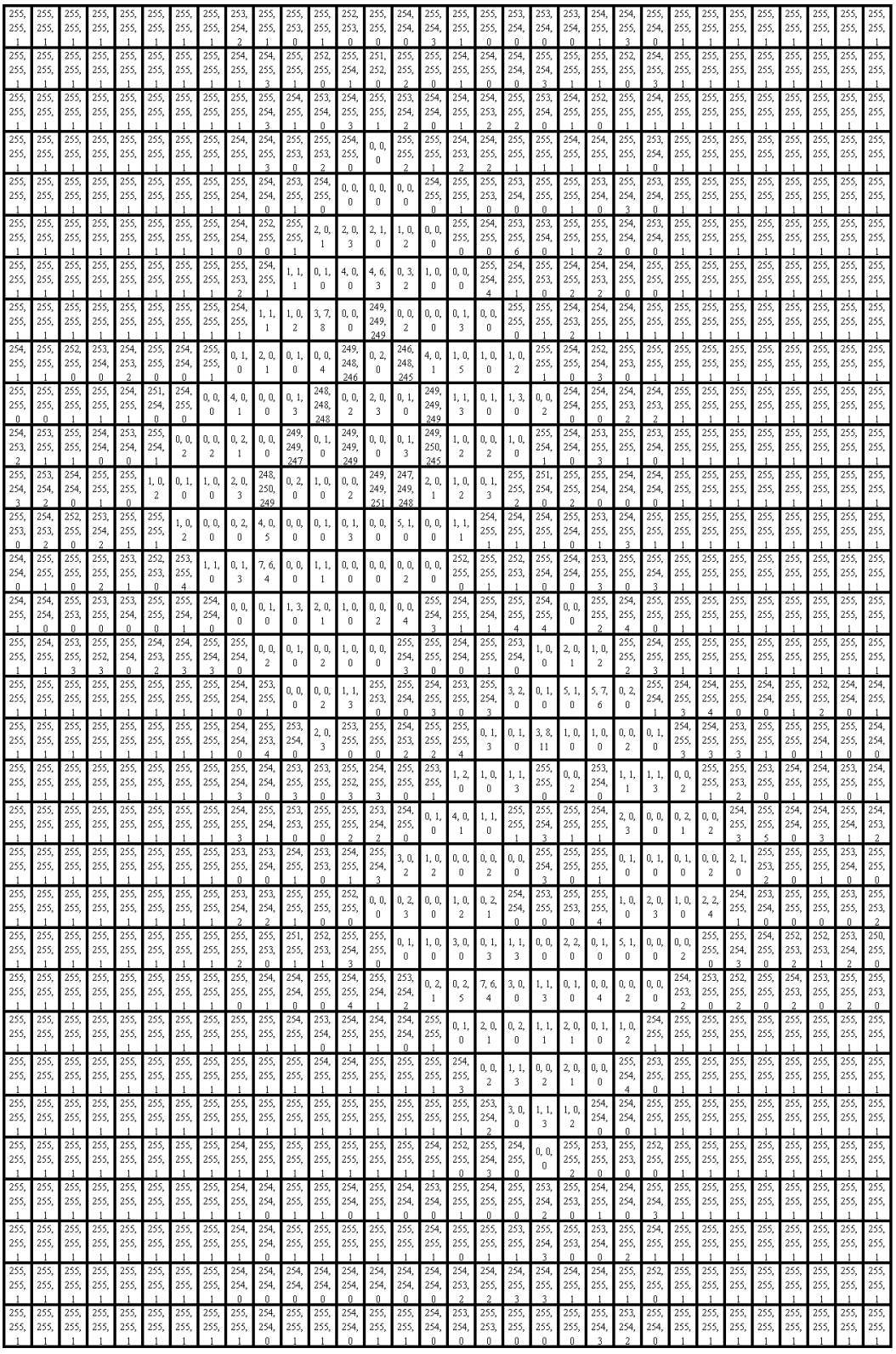
3.3. Analisis Data Masukan

Dalam penerapan algoritma canny untuk deteksi tepi dan harris untuk deteksi corner dalam preprosessing. Algoritma canny dan harris melakukan proses berdasarkan perubahaan intensitas cahaya dalam suatu data citra oleh karena itu data citra digital yang digunakan sebagai data masukan memerlukan citra objek yang memiliki border intensitas cahaya yang kontras sehigga algoritma canny dan harris dapat bekerja dengan baik. Contoh data citra yang dapat digunakan dapat dilihat pada Gambar 3.6.



Gambar 3.6 Citra RGB yang memiliki objek dengan warna border kontras

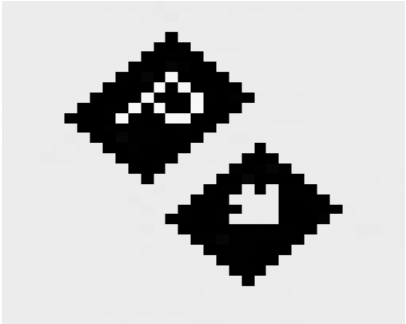
Gambar 3.6 memiliki ukuran 32x32 *pixel* dengan model warna RGB dan kedalaman warna 8bit. Nilai setiap *pixel* dapat dilihat pada Gambar 3.7.



Gambar 3.7 Nilai (R-G-B) pixel dari Citra

Gambar 3.6 memiliki 3 *chanel* warna RGB, guna mempermudah proses perhitungan 3 chanel warna di konversi ke citra keabuan menggunakan persamaan

(2.2). Hasil konversi dapat dilihat pada Gambar 3.8 dan daftar nilia per pixel nya dapat dilihat pada Gambar 3.9.

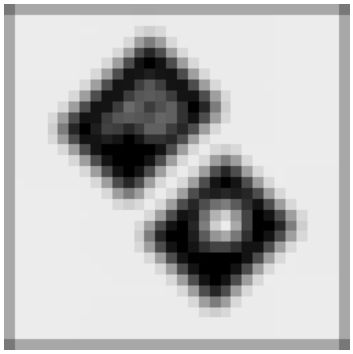


Gambar 3.8 Hasil konversi ke citra keabuan

236	236	236	236	236	236	236	236	235	236	235	236	234	236	235	236	236	236	235	235	235	236	236	235	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	236	236	236	235	235	233	236	236	236	236	235	236	236	236	235	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	236	236	236	235	236	236	235	235	236	235	236	235	236	235	236	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	236	235	235	236	0	236	236	235	236	236	236	236	236	236	235	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	235	235	236	236	0	0	0	236	236	235	235	236	236	235	236	235	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	235	235	236	0	0	1	0	0	236	236	236	235	236	236	235	235	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	235	236	1	0	0	5	2	0	0	236	236	235	236	235	236	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	236	1	0	6	0	248	0	0	0	0	236	236	235	236	236	236	236	236	236	236	236	236	236	236
236	236	235	235	235	236	235	236	0	0	0	0	248	1	247	0	0	0	0	236	236	235	235	236	236	236	236	236	236	236	236	236
236	236	236	236	236	235	236	0	0	0	0	247	0	0	0	248	1	0	2	0	235	236	235	235	236	236	236	236	236	236	236	236
235	236	236	235	235	235	0	0	1	0	248	0	248	0	0	249	0	0	0	235	235	236	236	235	236	236	236	236	236	236	236	236
236	235	235	236	236	0	0	0	0	249	1	0	0	249	248	0	0	0	236	235	236	235	235	235	236	236	236	236	236	236	236	236
235	235	235	235	236	236	0	0	1	1	0	0	0	0	1	0	1	236	236	236	235	236	236	236	236	236	236	236	236	236	236	236
235	236	236	236	236	234	236	0	0	6	0	1	0	0	0	0	235	236	234	235	235	236	236	236	236	236	236	236	236	236	236	236
236	235	235	236	235	236	235	235	0	0	2	0	0	0	0	236	236	235	236	236	0	236	236	236	236	236	236	236	236	236	236	236
236	236	236	234	235	235	236	236	235	0	0	0	0	0	236	236	235	236	235	0	0	0	236	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	235	236	0	0	1	235	235	236	236	236	2	0	1	6	1	235	236	236	236	235	236	236	235	236
236	236	236	236	236	236	236	236	235	235	235	0	236	236	235	236	236	0	0	7	0	0	0	0	236	236	236	236	236	235	236	235
236	236	236	236	236	236	236	236	236	235	236	236	234	236	236	236	1	0	1	236	0	235	1	1	0	236	235	236	236	236	236	236
236	236	236	236	236	236	236	236	236	235	234	236	236	235	236	0	0	0	236	236	236	236	0	0	1	0	236	236	235	236	235	235
236	236	236	236	236	236	236	236	236	235	236	234	236	236	0	0	0	0	0	236	236	236	0	0	0	0	1	235	236	236	235	236
236	236	236	236	236	236	236	236	235	235	236	236	235	0	1	0	0	1	235	236	235	236	0	0	0	2	236	235	236	236	236	235
236	236	236	236	236	236	236	236	236	235	235	234	236	236	0	0	0	0	1	0	1	0	1	0	0	236	236	236	234	234	235	235
236	236	236	236	236	236	236	236	236	236	235	236	236	235	235	1	1	6	0	1	0	0	0	0	235	236	236	236	235	236	236	235
236	236	236	236	236	236	236	236	236	236	236	235	236	236	235	236	0	0	1	1	0	0	0	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	0	1	0	0	0	236	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	235	0	1	0	235	235	236	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	235	236	236	0	236	236	235	235	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	236	235	236	236	235	236	235	235	236	236	236	235	235	236	235	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	236	235	236	236	236	236	236	236	236	236	236	236	235	235	236	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	235	235	235	235	235	235	235	235	235	236	236	236	236	236	236	235	236	236	236	236	236	236	236	236	236
236	236	236	236	236	236	236	236	236	235	236	236	235	236	236	235	235	235	236	236	236	236	235	235	236	236	236	236	236	236	236	236

Gambar 3.9 Nilai pixel Hasil konversi ke citra keabuan

Setelah citra keabuan didapatkan, citra dilakukan proses *bluring* sebagai pencegahan dan penanggulangan terhadap *noise* pada citra. Proses tersebut dilakukan dengan cara citra di konvolusi dengan kernel gaussian (2.6). Sebagai contoh kasus kernel gaussian dibangkitkan dengan nilai $\sigma = 1.1$ dan besar kernel 3x3. Hasil penghalusan dapat dilihat pada Gambar 3.10 dan Gambar 3.11



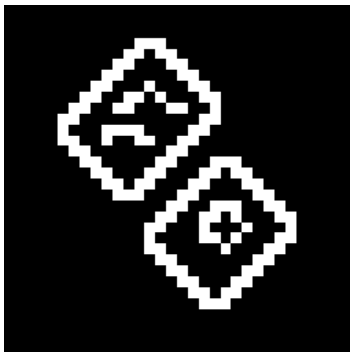
Gambar 3.10 Hasil penghalusan dengan gaussian

120	168	168	168	168	168	168	168	168	168	168	168	167	167	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	120
168	236	236	236	236	236	236	235	235	235	235	235	234	234	235	235	235	235	235	235	235	235	235	235	235	236	236	236	236	236	168
168	236	236	236	236	236	236	236	236	235	235	235	216	206	216	235	235	235	235	235	235	235	235	235	235	236	236	236	236	236	168
168	236	236	236	236	236	236	235	235	235	235	216	158	124	158	216	235	235	235	235	235	235	235	235	235	236	236	236	236	236	168
168	236	236	236	236	236	236	235	235	235	216	158	76	38	77	158	216	235	235	235	235	235	235	235	236	236	236	236	236	236	168
168	236	236	236	236	236	236	235	235	216	158	77	19	0	19	77	158	216	235	235	235	235	235	235	236	236	236	236	236	236	168
168	236	236	236	236	236	236	235	216	158	77	19	21	31	21	19	76	159	216	235	235	235	235	235	236	236	236	236	236	236	168
168	235	235	235	235	235	235	216	159	77	20	21	62	86	61	20	19	76	158	216	235	235	235	235	236	236	236	236	236	236	168
168	235	235	235	235	235	216	158	77	19	20	61	86	91	86	60	20	19	77	159	216	235	235	235	235	236	236	236	236	236	168
168	235	235	235	235	216	158	76	19	20	60	106	91	60	80	96	50	0	38	124	206	235	235	235	235	236	236	236	236	236	168
168	235	235	235	216	158	76	19	20	61	86	91	86	81	101	96	50	19	77	158	216	235	235	235	235	236	236	236	236	236	168
168	235	235	235	206	124	38	0	30	66	61	40	60	96	96	61	39	77	158	216	235	235	235	235	235	236	236	236	236	236	168
168	235	235	235	216	158	76	19	20	31	21	0	20	50	50	39	77	158	216	235	235	235	235	235	236	236	236	236	236	236	168
168	235	235	235	235	216	158	76	19	1	1	0	0	0	19	77	158	216	235	216	206	216	236	236	236	236	236	236	236	236	168
168	235	235	235	235	235	216	158	77	20	0	0	0	19	76	158	216	235	216	158	124	158	216	236	236	236	236	236	236	236	168
168	235	235	235	235	235	235	216	158	76	19	0	19	76	158	216	235	216	158	77	38	77	159	216	235	236	235	235	235	235	168
168	236	235	235	235	235	235	235	216	158	76	38	77	158	216	235	216	159	77	20	1	20	77	158	216	236	235	235	235	235	168
168	236	236	236	236	236	236	235	235	216	158	124	158	216	235	216	159	77	39	30	39	29	38	77	159	216	235	235	235	235	168
168	236	236	236	236	236	236	235	235	235	216	206	216	235	216	159	77	38	77	111	125	91	48	19	77	158	216	235	235	235	168
168	236	236	236	236	236	236	236	235	235	235	235	235	216	158	77	19	29	111	178	206	149	67	0	19	77	158	216	235	235	168
168	236	236	236	236	236	236	235	235	235	235	235	216	158	77	19	0	38	125	206	235	168	67	0	0	38	125	206	235	235	168
168	236	236	236	236	236	236	235	235	235	235	235	206	125	38	0	0	29	91	149	168	120	48	0	19	77	159	216	235	235	168
168	236	236	236	236	236	236	235	235	235	235	235	216	158	76	19	0	20	48	67	67	48	19	19	77	159	216	235	235	235	168
168	236	236	236	236	236	236	236	235	235	235	235	235	216	158	77	20	1	1	0	0	0	19	76	158	216	236	235	235	235	168
168	236	236	236	236	236	236	236	236	235	235	235	235	235	216	159	77	20	0	0	0	19	76	158	216	235	236	235	235	235	168
168	236	236	236	236	236	236	236	236	236	235	235	235	235	235	216	158	77	19	0	19	76	158	216	236	236	236	236	236	236	168
168	236	236	236	236	236	236	236	236	236	236	236	236	236	235	216	158	77	38	76	158	216	235	235	236	236	236	236	236	236	168
168	236	236	236	236	236	236	236	235	235	235	235	235	235	235	235	216	158	124	158	216	235	235	235	236	236	236	236	236	236	168
168	236	236	236	236	236	236	236	235	235	235	235	235	235	235	235	235	235	216	206	216	235	235	235	235	236	236	236	236	236	168
168	236	236	236	236	236	236	235	235	235	235	235	235	235	235	235	235	235	235	235	235	235	235	235	235	236	236	236	236	236	168
168	236	236	236	236	236	236	235	235	235	235	235	235	235	235	235	235	235	235	235	235	235	235	235	235	236	236	236	236	236	168
120	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	168	120

Gambar 3.11 Nilai pixel penghalusan dengan gaussian

Setelah citra diperhalus proses pendeteksian tepi dan *corner* dapat dilakukan. Sebagai contoh kasus parameter canny dan harris menggunakan $\sigma = 1.1$ untuk gaussian turunun pertama dan besar kernel gaussian $3x3$ serta

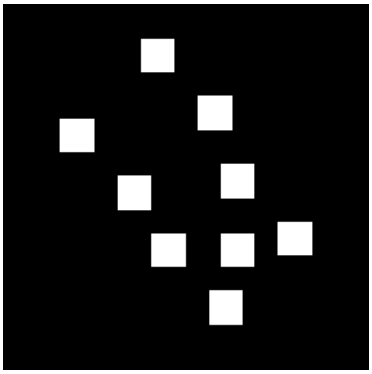
$minThresholdCanny = 3$, $maxThresholdCanny = 9$ dan $Threshold\ corner = 6000000$. Hasil pendeteksian tepi Canny dapat dilihat pada Gambar 3.12 dan Gambar 3.13, sedangkan hasil pendeteksian *corner* Harris dapat dilihat pada Gambar 3.14 dan Gambar 3.15.



Gambar 3.12 Hasil pendeteksian tepi canny

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	255	255	0	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	255	255	0	0	0	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	255	255	0	0	0	0	0	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	255	255	0	0	0	255	0	0	0	255	255	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	255	255	0	0	255	255	0	255	0	0	0	255	255	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	255	255	0	0	255	255	0	0	0	255	255	0	0	255	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	255	255	0	0	0	0	0	0	0	0	0	0	255	255	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	255	0	0	0	255	255	255	255	0	0	0	0	255	255	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	255	255	0	0	255	0	0	255	255	0	0	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	255	255	0	0	0	0	0	0	0	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	255	255	0	0	0	0	0	0	255	255	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	255	255	0	0	0	255	255	0	0	0	255	255	0	255	255	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	255	255	0	255	255	0	0	0	255	255	0	0	0	255	255	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	255	255	255	0	0	0	255	255	0	0	0	0	0	255	255	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0	255	255	255	0	0	255	255	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0	255	255	0	255	0	0	0	255	255	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0	255	255	0	255	0	0	0	255	255	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0	0	0	0	255	0	0	0	255	255	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0	0	0	0	0	0	255	255	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0	0	0	0	0	255	255	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0	0	0	0	255	255	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0	0	0	255	255	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0	0	255	255	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	0	255	255	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	0	255	255	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 3.13 Nilai pixel hasil deteksi tepi canny



Gambar 3.14 Hasil pendeteksian corner harris

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	3	3	3	0	0	0	0	0	0	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	5	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6	6	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	7	7	7	0	0	0	8	8	8	0	0	6	6	6	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	7	7	7	0	0	0	8	8	8	0	0	6	6	6	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	7	7	7	0	0	0	8	8	8	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	9	9	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	9	9	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	9	9	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 3.15 Nilai pixel hasil pendeteksian corner harris

Objek yang akan digunakan hanya objek yang memiliki ciri kotak. Kotak dalam hal ini objek citra yang memiliki empat corner dan saling terhubung oleh border tepinya. Pendeteksiaan ciri tersebut dilakukan dengan cara menelusuri setiap corner hasil algoritma harris, setiap corner diperiksa dan dilihat keterhubungan dengan tepi hasil algoritma canny. Setelah setiap lokasi objek kotak didapatkan objek dalam citra ditandai. Hasil penyeleksian dapat dilihat pada Gambar 3.16.

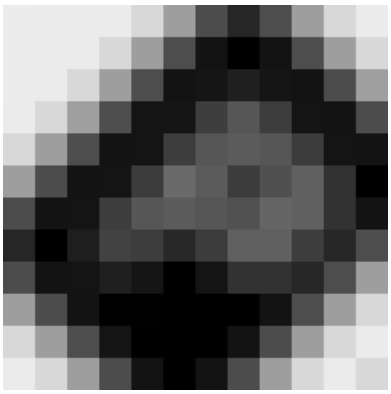
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 3.16 Hasil penyeleksian kontak pada citra

3.4. Analisis Metode / Algoritma

Setelah objek-objek citra didapatkan, objek citra dihitung ketujuh Hu Invariant Moments nya, ketujuh nilai dinormalisasi sehingga dapat digunakan

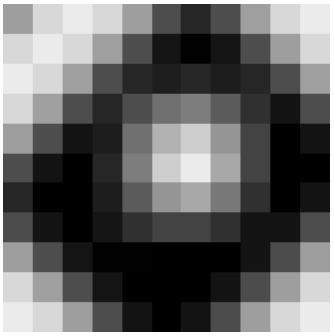
sebagai inputan neural network dalam poses pembelajaran dan rekognisi. Sebagai contoh kasus objek citra pada Gambar 3.6 digunakan sebagai data citra inputan dan lokasi objek dalam citra ditunjukan pada Gambar 3.16. serta data objek citra yang digunakan untuk perhitungan adalah citra satu chanel warna keabuan 8bit yang telah di *bluring* dengan gaussian. Sehingga objek pertama yang digunakan dapat dilihat pada Gambar 3.17 dan Tabel 3.1, serta objek kedua yang digunakan dapat dilihat pada Gambar 3.18 dan Tabel 3.2.



Gambar 3.17 Objek pertama

Tabel 3.1 Objek pertama

y\x	1	2	3	4	5	6	7	8	9	10	11	12	13
1								38	77				
2							19	0	19	77			
3						19	21	31	21	19	76		
4					20	21	62	86	61	20	19	76	
5				19	20	61	86	91	86	60	20	19	77
6			19	20	60	106	91	60	80	96	50	0	38
7		19	20	61	86	91	86	81	101	96	50	19	77
8	38	0	30	66	61	40	60	96	96	61	39	77	
9		19	20	31	21	0	20	50	50	39	77		
10			19	1	1	0	0	0	19	77			
11				20	0	0	0	19	76				
12					19	0	19	76					
13						38	77						



Gambar 3.18 Objek Kedua

Tabel 3.2 Objek kedua

y\x	1	2	3	4	5	6	7	8	9	10	11	12
1							38	77				
2						20	1	20	77			
3					39	30	39	29	38	77		
4				38	77	111	125	91	48	19	77	
5			19	29	111	178	206	149	67	0	19	77
6		19	0	38	125	206	235	168	67	0	0	38
7	38	0	0	29	91	149	168	120	48	0	19	77
8		19	0	20	48	67	67	48	19	19	77	
9			20	1	1	0	0	0	19	76		
10				20	0	0	0	19	76			
11					19	0	19	76				
12						38	76					

3.4.1. Perhitungan Invarian Moment Hu

Langkah pertama objek citra dihitung momen (m_{00}, m_{01}, m_{10}) pada sistem diskrit, menggunakan persamaan (2.17) :

$$m_{pq} = \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} x^p y^q f(x,y)$$

Dimana H dan W masing-masing merupakan tinggi dan lebar citra dan p = 0,1,2,… dan q = 0,1,2,… adalah integer.

1. Objek Pertama

   $m_{00} = 3873$

   $m_{10} = 30607$

   $m_{01} = 25196$

2. Objek Kedua

$m_{00} = 4099$

$m_{10} = 28867$

$m_{01} = 23652$

Selanjutnya, hitung momen pusat untuk suatu citra dengan persamaan (2.18):

$$\mu_{pq} = \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

Dimana :

$$\bar{x} = \frac{m_{10}}{m_{00}} \; dan \; \bar{y} = \frac{m_{01}}{m_{00}}$$

1. Objek Kesatu

$\bar{x} = \frac{30607}{3873} = 7.90266$

$\bar{y} = \frac{25196}{3873} = 6.50555$

$sehingga :$

μ_{11} =-3651.41

$\mu_{20} = 22690.3$

$\mu_{02} = 22490.1$

$\mu_{30} = -26801.4$

$\mu_{03} = -3438.17$

$\mu_{12} = 9445.5$

$\mu_{21} = 10199.5$

2. Objek Citra kedua

$\bar{x} = \frac{28867}{4099} = 7.04245$

$\bar{y} = \frac{23652}{4099} = 5.77019$

$sehingga:$

$\mu_{11} = -698.013$

$\mu_{20} = 14738.6$

$\mu_{02} = 17877.5$

$\mu_{30} = -4991.25$

$\mu_{03} = 6681.16$

$\mu_{12} = 10373.1$

$\mu_{21} = 5380.07$

Kemudian normalized central moments, menggunakan rumus (2.20):

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}{}^{\gamma}}$$

Dimana:

$$\gamma = \frac{p+q}{2} + 1$$

$$\mu_{00} = m_{00}$$

1. Objek Pertama

$$\eta_{11} = -\frac{3651.41}{3873^2} = -0.000243425$$

$$\eta_{20} = \frac{22690.3}{3873^2} = 0.00151267$$

$$\eta_{02} = \frac{22490.1}{3873^2} = 0.00149933$$

$$\eta_{30} = -\frac{26801.4}{3873^{2.5}} = -2.87103e-005$$

$$\eta_{03} = -\frac{3438.17}{3873^{2.5}} = -3.68306e-006$$

$$\eta_{12} = \frac{9445.5}{3873\text{^}2.5} = 1.01183e-005$$

$$\eta_{21} = \frac{10199.5}{3873\text{^}2.5} = 1.09259e-005$$

2. Objek Kedua

$$\eta_{11} = -\frac{698.013}{4099^2} = -4.15439e-005$$
$$\eta_{20} = \frac{14738.6}{4099^2} = 0.000877204$$
$$\eta_{02} = \frac{17877.5}{4099^2} = 0.00106402$$
$$\eta_{30} = -\frac{4991.25}{4099^{2.5}} = -4.63996e-006$$
$$\eta_{03} = \frac{6681.16}{4099^{2.5}} = 6.21094e-006$$
$$\eta_{12} = \frac{10373.1}{4099^{2.5}} = 9.64303e-006$$
$$\eta_{21} = \frac{5380.07}{4099\text{^}2.5} = 5.00142e-006$$

Sehingga ketujuh nilai invariant moments dapat diturunkan dari moment kedua dan ketiga sebagai berikut.

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4{\eta_{11}}^4$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - (3\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[(3\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\phi_6 = (\eta_{20} - 3\eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} - \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

1. Objek Pertama

$$\phi_1 = 0.00151267 + 0.00149933 = 0.003012$$

$$\phi_2 = (0.00151267 - 0.00149933)^2 + 4 * -0.000243425^2 = 2.37201e-007$$

$$\phi_3 = (-2.87103e-005 - 3 * 1.01183e-005)^2 + (3 * 1.09259e-005 - -3.68306e-006)^2 = 4.81809e-009$$

$$\phi_4 = (-2.87103e-005 + 1.01183e-005)^2 + (1.09259e-005 + -3.68306e-006)^2 = 3.98124e-010$$

$$\phi_5 = (-2.87103e-005 - 3 * 1.01183e-005) * (-2.87103e-005 + 1.01183e-005) * [(-2.87103e-005 + 1.01183e-005)^2 - 3 * (1.09259e-005 + -3.68306e-006)^2] + (3 * 1.09259e-005 - -3.68306e-006) * (1.09259e-005 + -3.68306e-006) * [3 * (-2.87103e-005 + 1.01183e-005)^2 - (1.09259e-005 + -3.68306e-006)^2] = 4.66763e-019$$

$$\phi_6 = (0.00151267 - 0.00149933) * [(-2.87103e-005 + 1.01183e-005)^2 - (1.09259e-005 + -3.68306e-006)^2] + 4 * -0.000243425 * (-2.87103e-005 + 1.01183e-005) * (1.09259e-005 + -3.68306e-006) = 1.35031e-013$$

$$\phi_7 = (3 * 1.09259\text{e} - 005 - -3.68306\text{e} - 006) * (-2.87103\text{e} - 005 + 1.01183\text{e} - 005) * [(-2.87103\text{e} - 005 + 1.01183\text{e} - 005)^2 - 3 * (1.09259\text{e} - 005 + -3.68306\text{e} - 006)^2] + (3 * 1.01183\text{e} - 005 - -2.87103\text{e} - 005) * (1.09259\text{e} - 005 + -3.68306\text{e} - 006) * [3 * (-2.87103\text{e} - 005 + 1.01183\text{e} - 005)^2 - (1.09259\text{e} - 005 + -3.68306\text{e} - 006)^2] = 2.9355\text{e} - 019$$

2. Objek Kedua

$$\phi_1 = 0.000877204 + 0.00106402 = 0.00194123$$

$$\phi_2 = (0.000877204 - 0.00106402)^2 + 4 * -4.15439e - 005^2 = 4.18051e - 008$$

$$\phi_3 = (-4.63996e - 006 - 3 * 9.64303e - 006)^2 + (3 * 5.00142e - 006 - 6.21094e - 006)^2 = 1.2042e - 009$$

$$\phi_4 = (-4.63996\text{e} - 006 + 9.64303\text{e} - 006)^2 + (5.00142\text{e} - 006 + 6.21094\text{e} - 006)^2 = 1.50748\text{e} - 010$$

$$\phi_5 = (-4.63996e - 006 - 3 * 9.64303e - 006) * (-4.63996e - 006 + 9.64303e - 006) * [(-4.63996e - 006 + 9.64303e - 006)^2 - 3 * (5.00142e - 006 + 6.21094e - 006)^2] + (3 * 5.00142e - 006 - 6.21094e - 006) * (5.00142e - 006 + 6.21094e - 006) * [3 * (-4.63996e - 006 + 9.64303e - 006)^2 - (5.00142e - 006 + 6.21094e - 006)^2] = 5.41467e - 020$$

$$\phi_6 = (0.000877204 - 0.00106402) * [(-4.63996e-006 + 9.64303e-006)^2 - (5.00142e-006 + 6.21094e-006)^2] + 4 * -4.15439e-005 * (-4.63996e-006 + 9.64303e-006) * (5.00142e-006 + 6.21094e-006) = 9.48832e-015$$

$$\phi_7 = (3 * 5.00142e-006 - 6.21094e-006) * (-4.63996e-006 + 9.64303e-006) * [(-4.63996e-006 + 9.64303e-006)^2 - 3 * (5.00142e-006 + 6.21094e-006)^2] + (3 * 9.64303e-006 - -4.63996e-006) * (5.00142e-006 + 6.21094e-006) * [3 * (-4.63996e-006 + 9.64303e-006)^2 - (5.00142e-006 + 6.21094e-006)^2] = -3.45456e-020$$

Sehingga invarian moment hu ketiga citra dapat dilihat pada Tabel 3.3. ϕ_1 sampai dengan ϕ_7 memiliki nilai yang sangat kecil, sehingga untuk melihat perbedaan nilai setiap objek citra nilai $\phi_1 sampai\ dengan\ \phi_7$. Ditampilkan dengan definisi $|\log(|\phi|)|$.

Tabel 3.3 Invarian moment Hu objek pertama dan kedua

	Pertama	**Kedua**
ϕ_1	5.80515	6.24443
ϕ_2	15.2544	16.9902
ϕ_3	19.1509	20.5374
ϕ_4	21.6443	22.6154
ϕ_5	42.2085	44.3626
ϕ_6	29.6333	32.2887
ϕ_7	42.6722	44.812

3.4.2. Normalisasi Data Input

Nilai-nilai yang didapatkan dari Invarian Moment hu belum memenuhi persyaratan data masukan neural network, oleh sebab itu setiap nilai ciri objek perlu dinormalisasi. Normalisasi nilai ciri objek menggunakan persamaan :

$$\gamma_i = \frac{X_i - \bar{X}}{\sigma}$$

Dimana :

$X_i = data\ ke\ i$

$\bar{X} = rata - rata\ data$

$\sigma = standar\ deviasi\ data$

Berdasarkan data yang telah didapat dan dapat dilihat pada Tabel 3.3. Normalisasi dapat dilakukan sebagai berikut :

1. Objek Pertama

$$\gamma_1 = \frac{5.80515 - 25.1955}{12.7533} = -1.52042$$

$$\gamma_2 = \frac{15.2544 - 25.1955}{12.7533} = -0.779495$$

$$\gamma_3 = \frac{19.1509 - 25.1955}{12.7533} = -0.473965$$

$$\gamma_4 = \frac{21.6443 - 25.1955}{12.7533} = -0.278458$$

$$\gamma_5 = \frac{42.2085 - 25.1955}{12.7533} = 1.334$$

$$\gamma_6 = \frac{29.6333 - 25.1955}{12.7533} = 0.347968$$

$$\gamma_7 = \frac{42.6722 - 25.1955}{12.7533} = 1.37037$$

2. Objek Kedua

$$\gamma_1 = \frac{6.24443 - 26.8358}{13.3014} = -1.54807$$

$$\gamma_2 = \frac{16.9902 - 26.8358}{13.3014} = -0.740193$$

$$\gamma_3 = \frac{20.5374 - 26.8358}{13.3014} = -0.473514$$

$$\gamma_4 = \frac{22.6154 - 26.8358}{13.3014} = -0.317292$$

$$\gamma_5 = \frac{44.3626 - 26.8358}{13.3014} = 1.31766$$

$$\gamma_6 = \frac{32.2887 - 26.8358}{13.3014} = 0.409948$$

$$\gamma_7 = \frac{44.812 - 26.8358}{13.3014} = 1.35145$$

Setelah semua nilai dinormalisasi, semua nilai telah memenuhi syarat sebagai masukan neuran network.

3.4.3. Pembentukan Target Output

Nilai target output akan digunakan sebagai patokan keberhasilan respon neural network. Banyak nilai target output sesuai dengan sejumlah objek yang akan digunakan. Sebagai contoh kasus objek citra yang akan di deteksi adalah citra objek pertama dan kedua yang telah dihitung semua ciri nya di pembahasan sebelumnya. Sehingga nilai target output dapat dilihat pada Tabel 3.4.

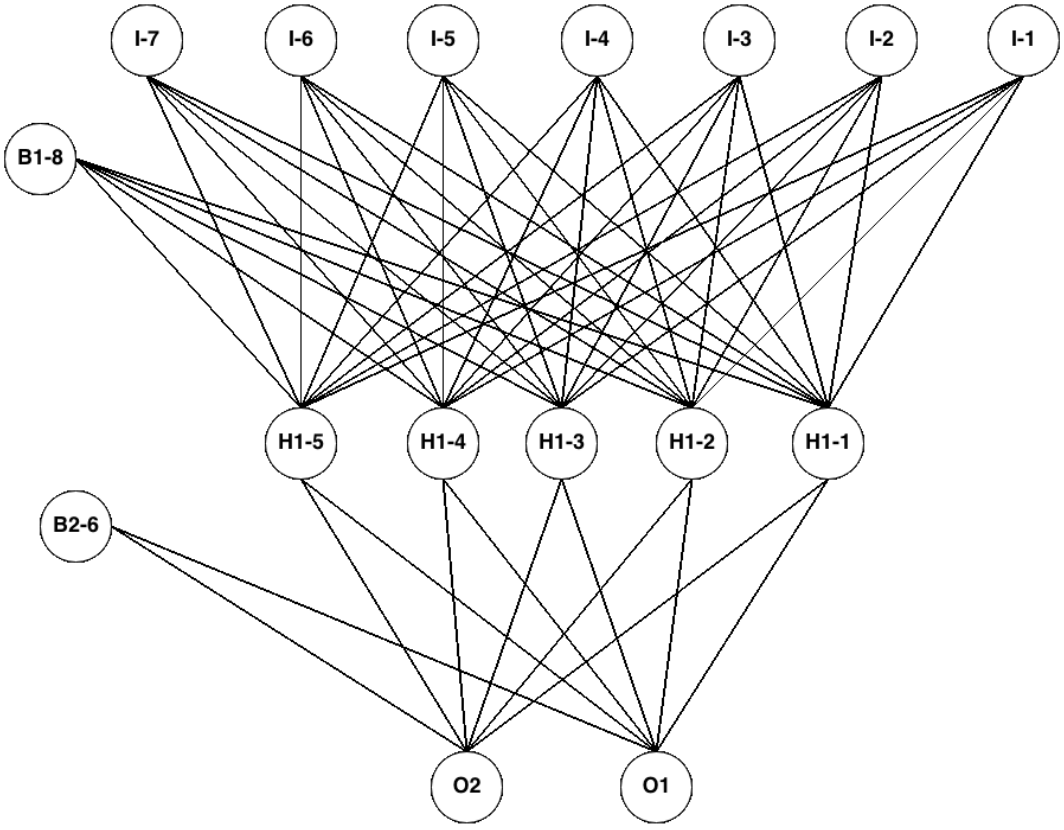
Tabel 3.4 Target Nilai output

Nama Objek	Nilai Target	
Pertama	1	0
Kedua	0	1

3.4.4. Arsitektur Neural Network

Arsitektur Neural Network yang digunakan terdiri dari tiga kategori layer yakni input layer, hidden layer dan output layer. Input layer memiliki tujuh neuron dan satu bias, hidden layer terdiri dari satu atau lebih lapis yang memiliki satu atau lebih neuron dan satu bias di setiap lapisnya, sedangkan output layer memiliki jumlah neuron sesuai jumlah objek citra.

Sebagai contoh kasus arsitektur neural network dibuat dengan input layer terdiri dari tujuh neuron (I) dan satu bias (B1), satu lapis hidden layer terdiri dari lima neuron (H1) dan satu bias (B2), serta output layer terdiri dari dua neuron (O). Semua neuron terhubung penuh. Arsitektur dapat dilihat pada Gambar 3.19.



3.4 *Gambar 3.19 Arsitektur Neural Network*

Setelah arsitektur dibuat proses pembelajaran neural network dilakukan. Langkah proses pembelajaran adalah sebagai berikut :

1. Inisialisasi bobot w dengan nilai acak

Tabel 3.5 Inisialisasi w dengan nilai acak

w	nilai	w	nilai	w	nilai
(I1, H1-1)	0,734508	(I3, H1-3)	0,749349	(I5, H1-5)	0,512095
(I2, H1-1)	0,665487	(I4, H1-3)	0,450334	(I6, H1-5)	0,199195
(I3, H1-1)	0,712425	(I5, H1-3)	0,124474	(I7, H1-5)	0,391401
(I4, H1-1)	0,995045	(I6, H1-3)	0,450328	(B1-8, H1-5)	0,102384
(I5, H1-1)	0,51333	(I7, H1-3)	0,025615	(H1-1,O1)	0,884172
(I6, H1-1)	0,360537	(B1-8, H1-3)	0,743469	(H1-2,O1)	0,446872
(I7, H1-1)	0,125333	(I1, H1-4)	0,788165	(H1-3,O1)	0,841721
(B1-8, H1-1)	0,835721	(I2, H1-4)	0,000604	(H1-4,O1)	0,779444
(I1, H1-2)	0,781151	(I3, H1-4)	0,238723	(H1-5,O1)	0,430308
(I2, H1-2)	0,639882	(I4, H1-4)	0,846913	(B2-6,O1)	0,908627
(I3, H1-2)	0,262325	(I5, H1-4)	0,303105	(H1-1,O2)	0,879888
(I4, H1-2)	0,644754	(I6, H1-4)	0,0374	(H1-2,O2)	0,434968

(I5, H1-2)	0,802046	(I7, H1-4)	0,477555	(H1-3,O2)	0,769047
(I6, H1-2)	0,073296	(B1-8, H1-4)	0,389626	(H1-4,O2)	0,31546
(I7, H1-2)	0,192204	(I1, H1-5)	0,968133	(H1-5,O2)	0,602273
(B1-8, H1-2)	0,817824	(I2, H1-5)	0,437176	(B2-6,O2)	0,553591
(I1, H1-3)	0,692875	(I3, H1-5)	0,976811		
(I2, H1-3)	0,203867	(I4, H1-5)	0,012421		

2. Masukan pola input di mulai dari objek pertama “Logo Sadaya”

Tabel 3.6 Inisialisasi neuron input

	nilai
I1	-1,56026000000
I2	-0,78621600000
I3	-0,28303200000
I4	-0,35404600000
I5	1,31368000000
I6	0,27726600000
I7	1,39261000000

3. Hitung respons jaringan neural network. Untuk setiap neuron mengunakan rumus (2.22) :

$$output = f\left(\sum_i i_i w_i\right)$$

Dimana $f(x)$ adalah fungsi sigmoid :

$$f(x) = \frac{1}{(1 + e^{-x})}$$

Sehingga perhitungan hidden layer pertama adalah sebagai berikut :

Tabel 3.7 Perhitungan Respon H1-1

H1-1				
input		**w**		**i x w**
I1	-1,520420	(I1, H1-1)	0,734508211	-1,11676097
I2	-0,779495	(I2, H1-1)	0,665486885	-0,5187437
I3	-0,473965	(I3, H1-1)	0,712425322	-0,33766467
I4	-0,278458	(I4, H1-1)	0,995044891	-0,27707821
I5	1,334000	(I5, H1-1)	0,513329855	0,684782027
I6	0,347968	(I6, H1-1)	0,360536991	0,125455336
I7	1,370370	(I7, H1-1)	0,12533251	0,171751912
B1-8	1,000000	(B1-8, H1-1)	0,835721316	0,835721316
$\sum_i i_i w_i$				-0,43253696
$\mathbf{H1-1} = \boldsymbol{f}\left(\sum_i \boldsymbol{i_i w_i}\right)$				0,393520692

Tabel 3.8 Perhitungan Respon H1-2

H1-2				
input		**w**		**i x w**
I1	-1,520420	-1,1876779	-1,1876779	-1,1876779
I2	-0,779495	-0,49878466	-0,49878466	-0,49878466
I3	-0,473965	-0,12433286	-0,12433286	-0,12433286
I4	-0,278458	-0,17953696	-0,17953696	-0,17953696
I5	1,334000	1,069929746	1,069929746	1,069929746
I6	0,347968	0,025504707	0,025504707	0,025504707
I7	1,370370	0,263389953	0,263389953	0,263389953
B1-8	1,000000	0,817824025	0,817824025	0,817824025
$\sum_i \boldsymbol{i_i w_i}$				0,186316044
$\mathbf{H1-2} = \boldsymbol{f}\left(\sum_i \boldsymbol{i_i w_i}\right)$				0,546444733

Tabel 3.9 Perhitungan Respon H1-3

H1-3				
input		**w**		**i x w**
I1	-1,0534609	-1,0534609	-1,0534609	-1,0534609
I2	-0,15891315	-0,15891315	-0,15891315	-0,15891315
I3	-0,35516506	-0,35516506	-0,35516506	-0,35516506
I4	-0,12539902	-0,12539902	-0,12539902	-0,12539902
I5	0,166048281	0,166048281	0,166048281	0,166048281
I6	0,156699704	0,156699704	0,156699704	0,156699704
I7	0,035102214	0,035102214	0,035102214	0,035102214
B1-8	0,743469092	0,743469092	0,743469092	0,743469092
$\sum_i i_i w_i$				-0,59161884
$\mathbf{H1-1} = f\left(\sum_i i_i w_i\right)$				0,356263503

Tabel 3.10 Perhitungan Respon H1-4

H1-4				
input		**w**		**i x w**
I1	-1,19834136	-1,19834136	-1,19834136	-1,19834136
I2	-0,00047045	-0,00047045	-0,00047045	-0,00047045
I3	-0,1131462	-0,1131462	-0,1131462	-0,1131462
I4	-0,23582977	-0,23582977	-0,23582977	-0,23582977
I5	0,404342436	0,404342436	0,404342436	0,404342436
I6	0,0130141	0,0130141	0,0130141	0,0130141
I7	0,654427129	0,654427129	0,654427129	0,654427129
B1-8	0,389626088	0,389626088	0,389626088	0,389626088
$\sum_i i_i w_i$				-0,08637802
$\mathbf{H1-4} = f\left(\sum_i i_i w_i\right)$				0,478418911

Tabel 3.11 Perhitungan Respon H1-5

H1-5				
input		**w**		**i x w**
I1	-1,4719688	-1,4719688	-1,4719688	-1,4719688
I2	-0,34077638	-0,34077638	-0,34077638	-0,34077638
I3	-0,46297428	-0,46297428	-0,46297428	-0,46297428
I4	-0,0034587	-0,0034587	-0,0034587	-0,0034587
I5	0,683134234	0,683134234	0,683134234	0,683134234
I6	0,069313575	0,069313575	0,069313575	0,069313575
I7	0,536364208	0,536364208	0,536364208	0,536364208
B1-8	0,102383856	0,102383856	0,102383856	0,102383856
$\sum_i i_i w_i$				-0,8879823
$\mathbf{H1-5} = f\left(\sum_i i_i w_i\right)$				0,291526385

Sehingga respon hidden layer adalah sebagai berikut :

H1-1	0,393520692
H1-2	0,546444733
H1-3	0,356263503
H1-4	0,478418911
H1-5	0,291526385

Sehingga respon neural network dapat dihitung, perhitungannya adalah sebagai berikut :

Tabel 3.12 Perhitungan Respon O1

O1				
input		**w**		**i x w**
H1-1	0,393520692	0,347939991	0,347939991	0,347939991
H1-2	0,546444733	0,244190704	0,244190704	0,244190704
H1-3	0,356263503	0,29987448	0,29987448	0,29987448
H1-4	0,478418911	0,372900677	0,372900677	0,372900677
H1-5	0,291526385	0,125446064	0,125446064	0,125446064
B2-6	1	0,908627227	0,908627227	0,908627227
$\sum_i \boldsymbol{i_i w_i}$				2,298979143
$\mathbf{O1} = \boldsymbol{f}\left(\sum_i \boldsymbol{i_i w_i}\right)$				0,908792457

Tabel 3.13 Perhitungan Respon O2

O2				
input		**w**		**i x w**
H1-1	0,34625431	0,34625431	0,34625431	0,34625431
H1-2	0,237686149	0,237686149	0,237686149	0,237686149
H1-3	0,273983364	0,273983364	0,273983364	0,273983364
H1-4	0,150922083	0,150922083	0,150922083	0,150922083
H1-5	0,175578472	0,175578472	0,175578472	0,175578472
B2-6	0,553590652	0,553590652	0,553590652	0,553590652
$\sum_i \boldsymbol{i_i w_i}$				1,73801503
$\mathbf{O2} = \boldsymbol{f}\left(\sum_i \boldsymbol{i_i w_i}\right)$				0,850434762

Sehingga didapatlah respon neural network sebagai berikut :

Tabel 3.14Hasil Perhitungan respon neural network

O1	0,908792457
O2	0,850434762

4. Hitung nilai error output menggunakan Root Mean Square Error (2.24) :

$$rms = \sqrt{\frac{1}{n}\sum_{i}^{n}(target_i - output_i)^2}$$

Sehingga nilai error output didapatkan sebagai berikut :

Tabel 3.15 Nilai error output Neural Network

output	target	$\boldsymbol{target_i - output_i}$	$\boldsymbol{(target_i - output_i)^2}$
0,908792457	1	-0,091207543	0,00831882
0,850434762	0	0,850434762	0,72323928
$\boldsymbol{\sum_{i}^{n}(target_i - output_i)^2}$			0,7315581
$\boldsymbol{\frac{1}{n}\sum_{i}^{n}(target_i - output_i)^2}$			0,2438527
$\boldsymbol{\sqrt{\frac{1}{n}\sum_{i}^{n}(target_i - output_i)^2}}$			0,49381444

5. Hitung gradient ouput layer (2.25):

$$\delta = (target - ouput) \,.\, f'(output)$$

Dimana :

$f'(output)$ = turunan fungsi sigmoid.

$$f'(x) = f(x)\ (1 - f(x))$$

$$f'(x) = \left(\frac{1}{(1 + e^{-x})}\right)\left(1 - \left(\frac{1}{(1 + e^{-x})}\right)\right)$$

Sehingga gradient output layer adalah :

Tabel 3.16 Gradient Output Layer

	O1	**O2**
output	0,908792457	0,850434762
target	1	0
target-output	-0,091207543	0,850434762
$\boldsymbol{f(output)}$	0,712752999	0,700658336
$\boldsymbol{1 - f(output)}$	0,287247001	0,299341664
$\boldsymbol{f'(output)}$	0,204736162	0,209736232
$\boldsymbol{\delta}$	-0,018673482	0,178366983

6. Hitung gradient hidden layer (2.26):

$$\delta = \left(\sum_i (w_i * \delta_{n_i})\right).f'(output)$$

Dimana :

δ_{n_i} = keluaran dari neuron selanjutnya

$f'(output)$ = turunan fungsi sigmoid.

$$f'(x) = f(x)\ (1 - f(x))$$

$$f'(x) = \left(\frac{1}{(1 + e^{-x})}\right)\left(1 - \left(\frac{1}{(1 + e^{-x})}\right)\right)$$

Sehingga gradient hidden layer dapat dihitung:

Tabel 3.17 Gradient Hidden Layer H-1

H1-1			
w_i		δ_{n_i}	$w_i * \delta_{n_i}$
H1-1,O1	0,884172035	-0,018673482	-0,016510571
H1-1,O2	0,879888448	0,178366983	0,156943048
$\sum_i (\boldsymbol{w_i * \delta_{n_i}})$			0,140432477
$output$			0,607924103
$f(output)$			0,647467117
$1 - f(output)$			0,352532883
$\boldsymbol{f'(output)}$			0,228253449
$\boldsymbol{\delta}$			0,032054197

Tabel 3.18 Gradient Hidden Layer H1-2

H1-2			
w_i		δ_{n_i}	$w_i * \delta_{n_i}$
H1-2,O1	0,446871731	-0,018673482	-0,008344651
H1-2,O2	0,434968322	0,178366983	0,077583987
$\sum_i (\boldsymbol{w_i} * \boldsymbol{\delta_{n_i}})$			0,069239336
$output$			0,607924103
$f(output)$			0,647467117
$1 - f(output)$			0,352532883
$\boldsymbol{f'(output)}$			0,228253449
$\boldsymbol{\delta}$			0,015804117

Tabel 3.19 Gradient Hidden Layer H1-3

H1-3			
w_i		δ_{n_i}	$w_i * \delta_{n_i}$
H1-3,O1	0,841721023	-0,018673482	-0,015717863
H1-3,O2	0,769046960	0,178366983	0,137172586
$\sum_i (\boldsymbol{w_i} * \boldsymbol{\delta_{n_i}})$			0,121454723
$output$			0,607924103
$f(output)$			0,647467117
$1 - f(output)$			0,352532883
$\boldsymbol{f'(output)}$			0,228253449
$\boldsymbol{\delta}$			0,02772246

Tabel 3.20 Gradient Hidden Layer H1-4

H1-4			
w_i		δ_{n_i}	$w_i * \delta_{n_i}$
H1-4,O1	0,779443848	-0,018673482	-0,014554931
H1-4,O2	0,315460111	0,178366983	0,056267668
$\sum_i (\boldsymbol{w_i} * \boldsymbol{\delta_{n_i}})$			0,041712737
$output$			0,607924103
$f(output)$			0,647467117
$1 - f(output)$			0,352532883
$\boldsymbol{f'(output)}$			0,228253449
$\boldsymbol{\delta}$			0,009521076

Tabel 3.21 Gradient Hidden Layer H1-5

H1-5			
w_i		δ_{n_i}	$w_i * \delta_{n_i}$
H1-5,O1	0,430307755	-0,018673482	-0,008035344
H1-5,O2	0,602273006	0,178366983	0,107425619
$\sum_i (\boldsymbol{w_i} * \boldsymbol{\delta_{n_i}})$			0,099390275
$output$			0,607924103
$f(output)$			0,647467117
$1 - f(output)$			0,352532883
$\boldsymbol{f'(output)}$			0,228253449
$\boldsymbol{\delta}$			0,022686173

7. Update bobot $w(i,j)$ dengan persamaan(2.27):

$\Delta w_{baru} = \beta \,.\delta\,.output + \alpha.\Delta w_{lama}$

$\Delta w_{lama} = \Delta w_{baru}$

$w = w + \Delta w_{baru}$

Dimana :

β = laju pembelajaran (learning rate)

α = parameter momentum

Sebagai contoh kasus, laju pembelajaran dan momentum yang digunakan adalah β = 0.25 dan $\alpha = 0.5$.

Tabel 3.22 Perhitungan Update Nilai w

$\boldsymbol{output}$	$\boldsymbol{w}$ (**saat ini**)		$\boldsymbol{\delta}$	$\Delta \boldsymbol{w_{lama}}$	$\Delta \boldsymbol{w_{baru}}$	$\boldsymbol{w}$ (**baru**)
-1,520420	(I1, H1-1)	0,734508211	0,032054197	0	-0,012183961	0,72232425
-0,779495	(I2, H1-1)	0,665486885	0,032054197	0	-0,006246522	0,659240363
-0,473965	(I3, H1-1)	0,712425322	0,032054197	0	-0,003798142	0,70862718
-0,278458	(I4, H1-1)	0,995044891	0,032054197	0	-0,002231437	0,992813454
1,334000	(I5, H1-1)	0,513329855	0,032054197	0	0,010690075	0,52401993
0,347968	(I6, H1-1)	0,360536991	0,032054197	0	0,002788459	0,36332545
1,370370	(I7, H1-1)	0,12533251	0,032054197	0	0,010981528	0,136314038
1,00000000000	(B1-8, H1-1)	0,835721316	0,032054197	0	0,008013549	0,843734865
-1,520420	(I1, H1-2)	0,781151194	0,015804117	0	-0,006007224	0,77514397
-0,779495	(I2, H1-2)	0,639881796	0,015804117	0	-0,003079808	0,636801988
-0,473965	(I3, H1-2)	0,26232499	0,015804117	0	-0,00187265	0,26045234

$output$	$w\ (saat\ ini)$		δ	Δw_{lama}	Δw_{baru}	$w\ (baru)$
-0,278458	(I4, H1-2)	0,644754193	0,015804117	0	-0,001100196	0,643653997
1,334000	(I5, H1-2)	0,802046286	0,015804117	0	0,005270673	0,807316959
0,347968	(I6, H1-2)	0,073296127	0,015804117	0	0,001374832	0,074670959
1,370370	(I7, H1-2)	0,192203531	0,015804117	0	0,005414372	0,197617903
1,00000000000	(B1-8, H1-2)	0,817824025	0,015804117	0	0,003951029	0,821775054
-1,520420	(I1, H1-3)	0,69287493	0,02772246	0	-0,010537445	0,682337485
-0,779495	(I2, H1-3)	0,2038668	0,02772246	0	-0,00540238	0,19846442
-0,473965	(I3, H1-3)	0,749348715	0,02772246	0	-0,003284869	0,746063846
-0,278458	(I4, H1-3)	0,450333689	0,02772246	0	-0,001929885	0,448403804
1,334000	(I5, H1-3)	0,124473974	0,02772246	0	0,00924544	0,133719414
0,347968	(I6, H1-3)	0,450327915	0,02772246	0	0,002411632	0,452739547
1,370370	(I7, H1-3)	0,025615136	0,02772246	0	0,009497507	0,035112643
1,00000000000	(B1-8, H1-3)	0,743469092	0,02772246	0	0,006930615	0,750399707
-1,520420	(I1, H1-4)	0,788164689	0,009521076	0	-0,003619009	0,78454568
-0,779495	(I2, H1-4)	0,000603531	0,009521076	0	-0,001855408	-0,001251877
-0,473965	(I3, H1-4)	0,238722685	0,009521076	0	-0,001128164	0,237594521
-0,278458	(I4, H1-4)	0,846913258	0,009521076	0	-0,000662805	0,846250453
1,334000	(I5, H1-4)	0,303105274	0,009521076	0	0,003175279	0,306280553
0,347968	(I6, H1-4)	0,037400278	0,009521076	0	0,000828257	0,038228535
1,370370	(I7, H1-4)	0,477555061	0,009521076	0	0,003261849	0,48081691
1,00000000000	(B1-8, H1-4)	0,389626088	0,009521076	0	0,002380269	0,392006357
-1,520420	(I1, H1-5)	0,968133019	0,022686173	0	-0,008623128	0,959509891
-0,779495	(I2, H1-5)	0,43717584	0,022686173	0	-0,00442094	0,4327549
-0,473965	(I3, H1-5)	0,976811124	0,022686173	0	-0,002688113	0,974123011
-0,278458	(I4, H1-5)	0,012420912	0,022686173	0	-0,001579287	0,010841625
1,334000	(I5, H1-5)	0,512094628	0,022686173	0	0,007565839	0,519660467
0,347968	(I6, H1-5)	0,199195256	0,022686173	0	0,001973516	0,201168772
1,370370	(I7, H1-5)	0,391401014	0,022686173	0	0,007772113	0,399173127
1,00000000000	(B1-8, H1-5)	0,102383856	0,022686173	0	0,005671543	0,108055399
0,393520692	(H1-1,O1)	0,884172035	-0,018673482	0	-0,0018371	0,882334935
0,546444733	(H1-2,O1)	0,446871731	-0,018673482	0	-0,002551007	0,444320725
0,356263503	(H1-3,O1)	0,841721023	-0,018673482	0	-0,00166317	0,840057853
0,478418911	(H1-4,O1)	0,779443848	-0,018673482	0	-0,002233437	0,777210412
0,291526385	(H1-5,O1)	0,430307755	-0,018673482	0	-0,001360953	0,428946802
1	(B2-6,O1)	0,908627227	-0,018673482	0	-0,004668371	0,903958856
0,393520692	(H1-1,O2)	0,879888448	0,178366983	0	0,017547775	0,897436222
0,546444733	(H1-2,O2)	0,434968322	0,178366983	0	0,024366925	0,459335247
0,356263503	(H1-3,O2)	0,76904696	0,178366983	0	0,015886412	0,784933371
0,478418911	(H1-4,O2)	0,315460111	0,178366983	0	0,021333534	0,336793645
0,291526385	(H1-5,O2)	0,602273006	0,178366983	0	0,01299967	0,615272676
1	(B2-6,O2)	0,553590652	0,178366983	0	0,044591746	0,598182398

Proses pembelajaran tersebut dilakukan secara berulang dengan objek citra bergantian hingga nilai minimum error respon pada tahap ke-4 terpenuhi. Untuk tahap pengujian atau rekognisi lakukan proses perhitungan respon neural network seperti tahap ke-2 dan ke-3.

3.5. Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dalam implementasi *Neural Network Menggunakan Hu Invariant Moment Untuk Rekognisi Objek Tersebar* ini meliputi beberapa kebutuhan, diantaranya :

3.5.1. Analisis Kebutuhan Non Fungsional

Kebutuhan non fungsional yang dibutuhkan untuk implementasi kedalam sistem mencakup dua hal yakni kebutuhan perangkat keras dan perangkat lunak.

Kebutuhan Perangkat Keras

Adapun spesifikasi perangkat keras yang dibutuhkan untuk implementasi kedalam sistem antara lain :

1. Processor minimal memiliki kecepatan 2 GHz.
2. RAM berkapasitas minimal 2 GB.
3. Hard disk (HDD) berkapasitas minimal 20 GB atau disesuaikan dengan sistem operasi yang digunakan.
4. *Web-cam* minimal 2 MP.

3.5.1.1. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk implementasi kedalam sistem antara lain :

1. Sistem Operasi Windows XP, Windows 7 atau Windows 8.
2. MinGW (gcc) compiler
3. OpenCv 2.4.9
4. Qt 5.2.0

3.5.2. Analisis Kebutuhan Fungsional

fungsional merupakan tindakan mengidentifikasi proses yang akan berjalan didalam sistem, yang akan dikembangkan dan menjelaskan kebutuhan yang diperlukan. Sehingga sistem dapat berjalan seperti yang diharapkan.

Analisis fungsional dimodelkan dengan menggunakan UML (Unified Modeling Language). Tahapan pemodelan dalam analisis tersebut antara lain mengidentifikasi aktor, pembuatan diagram use case, scenario use case, diagram sequence, diagram class dan diagram package.

3.5.2.1. Identifikasi Aktor

Aktor yang teridentifikasi dan terlibat dalam sistem hanya ada satu yakni user.

3.5.2.2. Diagram Use Case

Use Case adalah interaksi yang terjadi antara sistem dan aktor yang terlibat, didalamnya termasuk pertukaran pesan dan tindakan yang dilakukan oleh sistem. Use Case diagram menggambarkan setiap aktivitas yang dilakukan oleh sistem dari sudut pengamatan tertentu.

Use case diagram dalam sistem ini dapat dilihat pada Lampran A-1 menunjukan diagram usecase untuk sistem ini.

3.5.2.3. Skenario Use Case

Skenario use case merupakan skenario (flow of event) interaksi aktor yang terlibat dengan beberapa usecase utama. Skenario usecase menggambarkan urutan interaksi actor dengan use case tersebut, dari awal sampai akhir. Skenario usecase utama untuk implementasi sistem dapat dilihat pada Lampiran A-2.

3.5.2.4. Activity diagram

Activity diagram merupakan teknik untuk menggunakan logika prosedural, proses bisnis, dan jalur kerja. Dari skenario yang dibuat, berikut akan digambarkan activity diagramnya. Activity diagram untuk implementasi sistem dapat dilihat pada Lampiran A-3.

3.5.2.5. Sequence diagram

Sequence diagram adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Diagram sequence untuk implemenasi sistem dapat dilihat pada Lampiran A-4

3.5.2.6. Class Diagram

Class diagram mendeskripsikan jenis–jenis objek dalam sistem dan berbagai macam hubungan statis yang ada. Class diagram beserta deskripsinya untuk implementasi sistem ini dapat dilihat pada Lampiran A-5.

3.5.2.7. Package Diagram

Package diagram merupakan pengelompokan kesamaan jenis class untuk memudahkan dalam pengorganisasian file. Package diagram untuk sistem ini dapat dilihat pada Lampiran A-6.

3.5.2.8. Deployment Diagram

Deployment diagram untuk sistem ini dapat dilihat pada Lampiran A-7.

3.6. Perancangan Sistem

Perancangan sistem merupakan penggambaran dan perencanaan dari beberapa elemen sistem yang terpisah ke dalam suatu kesatuan model yang utuh. Tahapan ini meliputi perancangan antarmuka dan jaringan semantik.

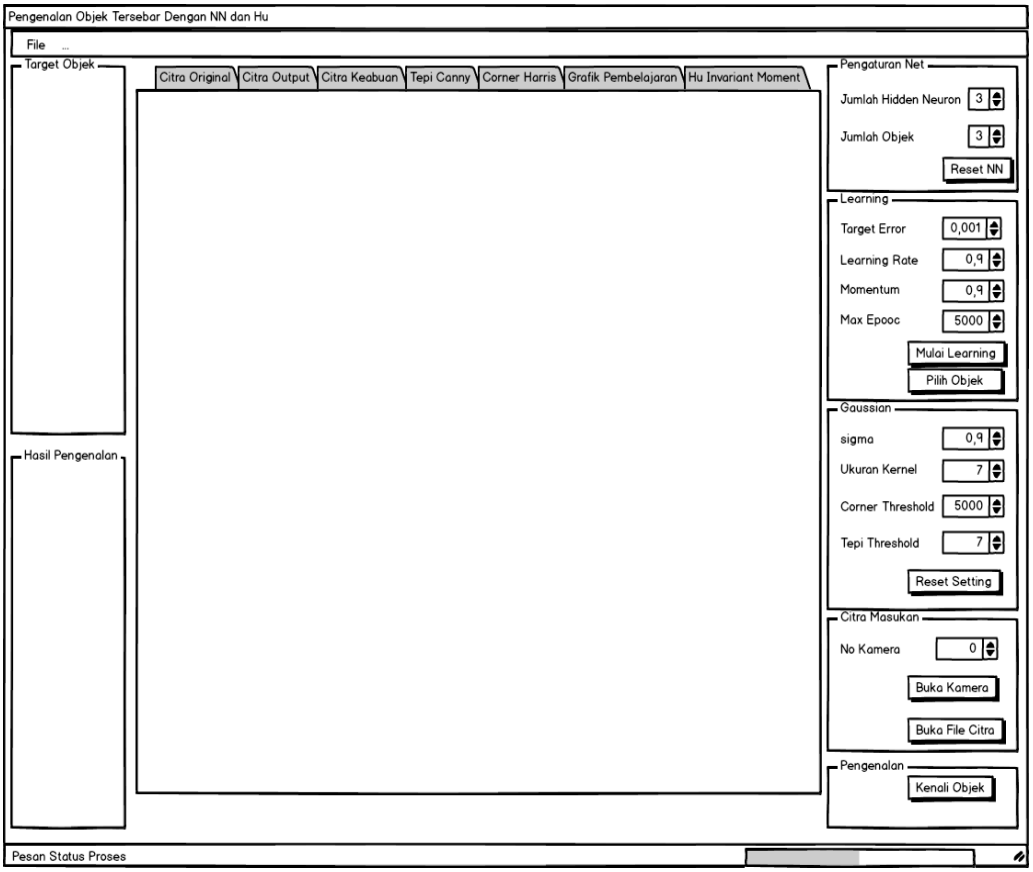
3.6.1. Perancangan Antar Muka

Perancangan antarmuka mendeskripsikan rencana tampilan dari setiap tampilan yang akan digunakan. Perancangan terdiri dari perancangan *form* dan pesan.

3.6.1.1. Perancangan Form

1. Perancangan Form Antar Muka Utama

No. F01



* Klik menu *File* untuk menyimpan dan membuka NN serta citra.

* Klik tombol **Reset NN** untuk mebuat ulang arsitektur NN.

* Klik tombol **Mulai Learning** untuk memulai atau menghentikan proses learning.

* Klik tombol **Pilih Objek** untuk memilih Objek *learning* yang digunakan dalam citra dan menuju ke **No 02**.

* Klik tombol **Reset Setting** untuk membuat ulang kernel gaussian dan mereset parameter algoritma haris dan canny.

* Klik tombol **Buka Kamera** untuk mengambil masukan citra dari device kamera terpilih.

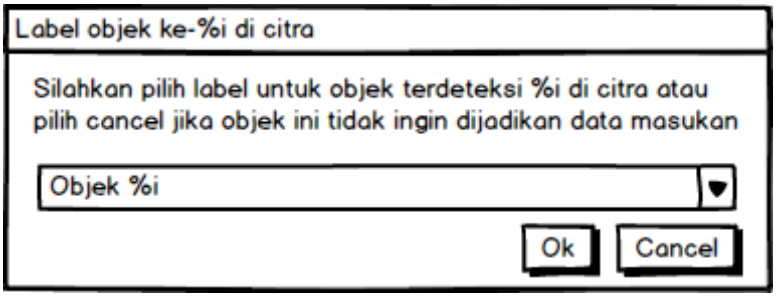
* Klik tombol **Buka File Citra** untuk mengambil masukan citra dari memory sekunder.

* Klik tombol **Kenali Objek** untuk mengetahui respon NN dari semua objek yang ada citra.

Ukuran 1090 x 591, warna disesuaikan

Font 8, Ms Shell Dlg 2, warna hitam

2. Perancangan Form Pilih Objek Learning

No F02	
	* Klik **Ok** untuk memberi menyetujuai nama objek terpilih dan kembali ke **No 01** * Klik **Cancel** untuk tidak memberikan nama objek yang terpilih dan kembali ke **No 01**
Ukuran 64x27 warna disesuaikan Font 8, Ms Shell Dlg 2, warna hitam	

3.6.1.2. Perancangan Pesan

1. Perancangan Pesan Saat Gagal Membuka File Citra

No P01
Gagal membuka file, periksa file anda
Font 8, Ms Shell Dlg 2, warna hitam

2. Perancangan Pesan Saat Gagal Membuka Device Kamera

No P02
Kamera dengan no kamera terpilih, tidak bisa dibuka. Periksa koneksi device anda.
Font 8, Ms Shell Dlg 2, warna hitam

3. Perancangan Pesan Saat Akan Memulai Preprosessing

No P03
Mohon tunggu sedang memulai preprosessing
Font 8, Ms Shell Dlg 2, warna hitam

4. Perancangan Pesan Saat Akan Mengkonversikan Citra Kederajat Keabuan

No P04
Mohon tunggu, sedang mengkonversi citra ke derajat keabuan
Font 8, Ms Shell Dlg 2, warna hitam

5. Perancangan Pesan Saat Akan Memfilter Citra

No P05
Mohon tunggu, sedang memfilter citra
Font 8, Ms Shell Dlg 2, warna hitam

6. Perancangan Pesan Saat Akan Mendeteksi Tepi Citra

No P06
Mohon tunggu, sedang mendeteksi TEPI citra
Font 8, Ms Shell Dlg 2, warna hitam

7. Perancangan Pesan Saat Akan Mendeteksi Corner Citra

No P07
Mohon tunggu, sedang mendeteksi CORNER citra
Font 8, Ms Shell Dlg 2, warna hitam

8. Perancangan Pesan Saat Akan Mendeteksi Objek Kotak di Citra

No P08
Mohon tunggu, sedang mendeteksi objek di citra
Font 8, Ms Shell Dlg 2, warna hitam

9. Perancangan Pesan Saat Akan Menghitung Hu Invariant Moment

No P09
Mohon tunggu, sedang menghitung HU INVARIANT MOMENT dari %i objek citra
Font 8, Ms Shell Dlg 2, warna hitam

10. Perancangan Pesan Saat Preprosessing Telah Selesai dan Menemukan Objek

No P10
%d objek kotak di temukan dan jumlah sampel NN %d
Font 8, Ms Shell Dlg 2, warna hitam

11. Perancangan Pesan Saat Preprosessing Telah Selesai dan Tidak Menemukan Objek

No P11
tidak menemukan objek kotak
Font 8, Ms Shell Dlg 2, warna hitam

12. Perancangan Pesan Saat Prosses Learning

No P12
Learning masih proses dengan error %+06.8f di iterasi %d
Font 8, Ms Shell Dlg 2, warna hitam

13. Perancangan Pesan Saat Prosses Learning Berhenti atau Dihentikan

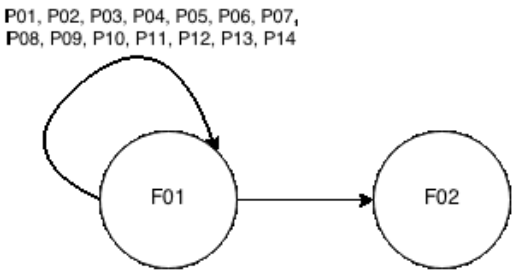
No P13
Learning berhenti dengan error %+06.8f dan iterasi ke %d
Font 8, Ms Shell Dlg 2, warna hitam

14. Perancangan Pesan Saat Gagal Membuka File NN

No P14
Terjadi kesalahan saat membuka file NN
Font 8, Ms Shell Dlg 2, warna hitam

3.6.2. Jaringan Semantik

Jaringan Semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek. Jaringan semantik terdiri dari lingkaran-lingkaran yang menunjukkan objek dan informasi tentang objek-objek tertentu. Jaringan semantik pada sistem ini dapat dilihat pada Gambar 3.20.



Gambar 3.20 Jaringan Semantik