BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Beras merupakan makanan pokok penduduk dunia salah satunya indonesia sebagai negara yang tingkat konsumsi beras mencapai 139 kilogram perkapita per tahun, kualitas beras menjadi hal yang sangat penting yang akan berpengaruh pada kandungan dan jumlah gizi, pada penelitian sebelumnya kualitas fisik beras telah diuji menggunakan pengolahan citra dan jaringan syaraf tiruan[3] dimana jaringan syaraf tiruan tidak dapat dikombinasikan dengan analisis tekstur sehingga tingkat akurasi yang dihasilkan kurang optimal.

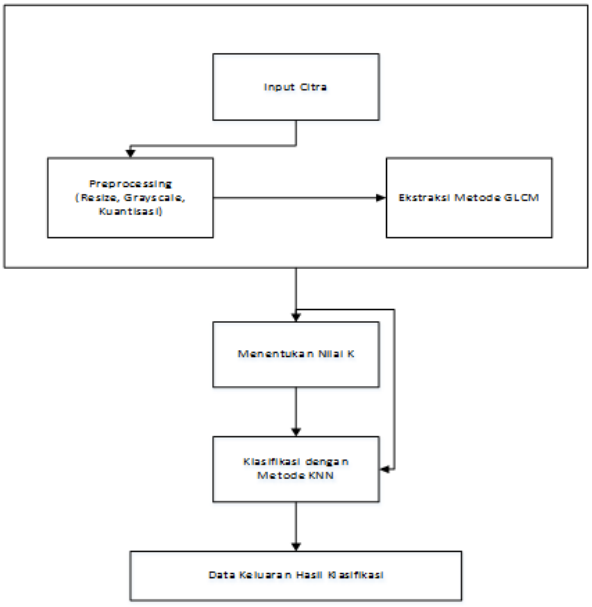
Keakurasian dan kecepatan pendeteksi merupakan hal yang sangat penting dalam pengolahan citra. Sehingga metode yang digunakan untuk mendeteksi citra jadi salah satu pertimbangan yang sangat penting agar menghasilkan data yang sesuai dan optimal. Algoritma yang digunakan pada penelitian ini adalah algoritam *grey level co-occurrence matrix* dan metode klasifikasi *k-nearest neighbor*.

Pada penelitian sebelumnya yang dilakukan oleh [1] dan [2], telah membuktikan bahwa algoritma GLCM dapat mengekstraksi ciri tekstur dengan baik, GLCM menghasilkan hasil ekstraksi ciri berupa data angka, hasil pengukuran atau disebut data *continues*, berupa kontras, *homogenitas*, *dissimilarity*, *energy* dan *entropy*. selain itu pada penelitian yang dilakukan oleh [4] dengan menggunakan metode KNN telah menghasilkan tingkat keakurasian.

Dari yang telah dijelaskan, maka dalam penelitian ini akan melakukan analisis, dengan mengkombinasikan algoritma ekstraksi ciri *grey level co-occurrence matrix* dengan metode klasifikasi *k-nearest neighbor* untuk mengklasifikasikan citra. Dan menguji tingkat keakuratan dalam pengujian dengan berbagai skenario yang telah dipersiapkan.

3.2 Analisis Proses

Analisis proses yang akan dilakukan pada penelitian kali ini dibagi menjadi beberapa tahapan seperti: input citra beras, proses pengolahan citra menentukan nilai k pada metode KNN, pengujian dengan metode KNN hingga menghasilkan data keluaran hasil klasifikasi. Berikut adalah tahapan analisis proses yang akan dilakukan dapat dilihat sebagai berikut:



Gambar 0.1 Diagram Alir Proses Sistem

3.3 Analisis Data Masukan

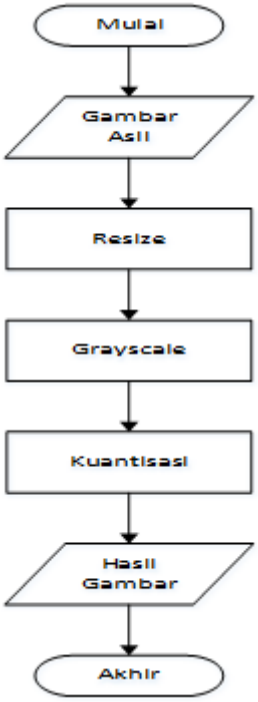
Pada penelitian ini yang akan dilakukan pertama adalah analisis data masukan. Data masukan berupa citra beras, kemudian citra akan di resize menjadi ukuran 32 x 32 piksel dengan *true color* yakni tiap piksel terdapat 3 nilai yaitu RGB, kemudian citra akan di konversi menjadi grayscale. Hal ini dilakukan untuk menyederhanakan dan memudahkan proses selanjutnya, dengan tujuan menjadikan tiap piksel memiliki satu nilai yaitu nilai keabuan (*gray value*). Setelah citra di konversi menjadi grayscale akan didapatkan matriks grayscale tersebut , matriks ini akan digunakan sebagai proses ekstraksi citra kookurensi.

Tahapan yang akan dilakukan pada analisis data masukan adalah preprocessing yaitu dengan melakukan resize, grayscale dan kuantisasi citra setelah melakukan

preprocessing, selanjutnya adalah ekstraksi ciri dengan metode GLCM untuk mendapatkan nilai fitur dari citra tersebut.

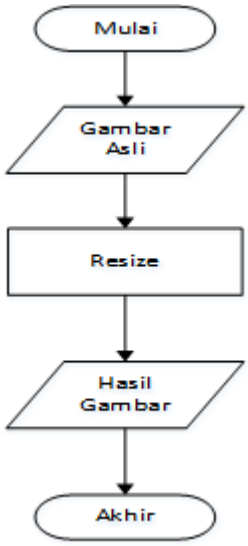
3.3.1 Preprosessing

Pada penelitian ini, preporcessing dilakukan untuk memudahkan mendapatkan nilai ekstraksi ciri, preprocessing yang akan dilakukan adalah resize, grayscale dan kuantisasi citra. Berikut alur proses preprocessing.



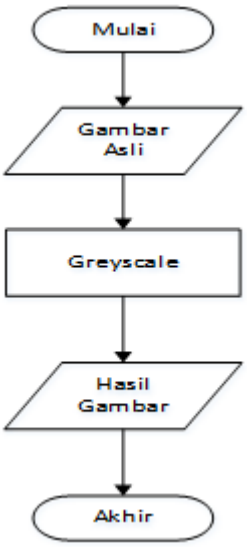
Gambar 0.2 Alir Preprosessing

1. Resize adalah tahap pertama dari preprocessing. Resize dilakukan untuk mempercepat dan memudahkan proses perhitungan. Berikut merupakan alur resizing:



Gambar 0.3 Alir Resize

2. Grayscale merupakan proses untuk mengubah warna menjadi keabu-abuan. Dengan mengubah nilai RGB setiap piksel gambar menjadi satu nilai yang sama sehingga setiap piksel memiliki nilai yang sama untuk ketiga unsur warna serta didapatkan nilai matriks grayscale. Berikut alur grayscale :

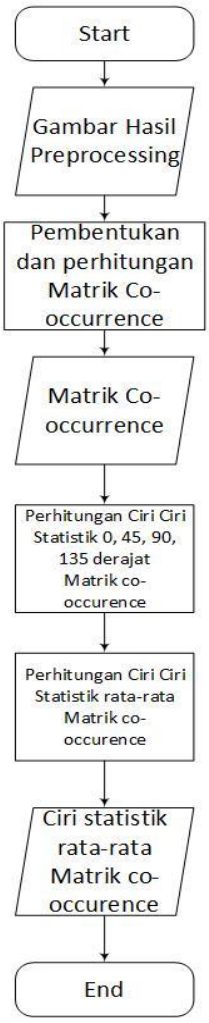


Gambar 0.4 DiagramAlir Greyscale

3. Kuantisasi citra merupakan proses untuk mengubah nilai range derajat keabuan dari citra greyscale yang awalnya memiliki range derajat keabuan 0-255 menjadi range derajat keabuan 0-15. Berikut alur Kuantisasi :

3.3.2 Ekstraksi Ciri

Ekstraksi ciri adalah proses untuk mendapatkan ciri utama yang terdapat pada citra, citra yang telah diubah ke dalam bentuk grayscale akan menghasilkan matriks grayscale yang telah di kuantisasi, matriks tersebutlah yang akan digunakan pada tahap ini, tahap ini akan mengitung 5 nilai statistik dari kookurensi yaitu kontras, energi, entopi, homogenitas dan dissimilarity dengan sudut 0^0, 45^0, 90^0,135^0. Setelah didapatkan semua nilai tersebut maka akan dirata-ratakan. Berikut alur ekstraksi ciri :

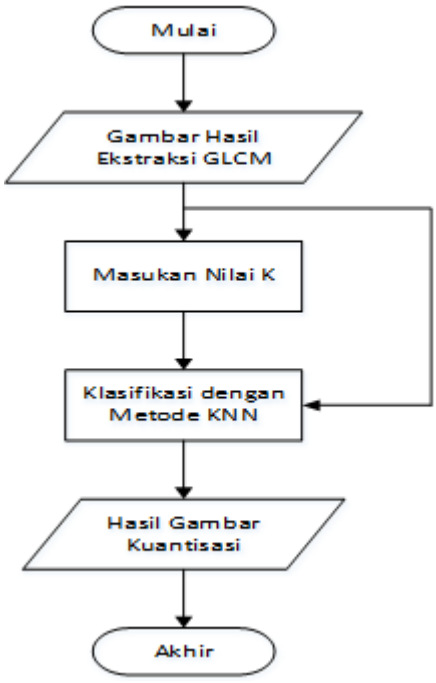


Gambar 0.5 Diagram Alir Ekstraksi Cri

3.4 Analisis Data Pelatihan dan Pengujian

3.4.1 Analisis Pengujian

Pengujian merupakan proses klasifikasi citra berdasarkan tekstur, pada proses ini citra yang dimasukan nilai probabilitasnya untuk dibandingkan. Berikut alur pengujian dengan metode k-nearest neighbor.



Gambar 0.6 Diagram alir pengujian

3.5 Analisis Perhitungan

3.5.1 Analisis Data Masukan

Data masukan berupa citra beras dengan berbagai kualitas dan akan melalui beberapa tahapan proses pengolahan citra, diantaranya adalah proses resizing, greyscaling, kuantisasi citra grayscale dan ekstraksi ciri. Berikut data masukan :



Gambar 0.7 Citra Beras Asli

1. Resizing

Untuk proses yang lebih cepat dalam mencari parameter pada citra data masukan, citra masukan awal akan di ubah ukurannya dengan proses resizing dari ukuran citra asli menjadi citra ukuran 32 x 32 pixel dengan mode warna RGB (Red, Green, Blue) yang artinya dalam setiap pixel terdapat 3 nilai warna.

Berikut adalah gambar citra asli(kiri) dan citra hasil resize(kanan) :



Gambar 0.8 Citra Asli (Kiri) dan Citra Resize(Kanan)

2. Greyscale

Setelah melalui tahap resize, maka tahap selanjutnya adalah mengubah citra mode RGB, menjadi Greyscale. Proses ini bertujuan untuk menyederhanakan nilai pixel pada sebuah citra. Yang awalnya pada setiap pixel memiliki 3 nilai yaitu RGB menjadi hanya 1 nilai keabuan.

Rumus yang digunakan untuk proses greyscale adalah sebagai berikut :

$X = 0.21*R + 0.72*G + 0.07*B$ (Persamaan 1)

Dimana

X = Nilai Greyscale

R = Nilai Red

G = Nilai Green

B = Nilai Blue

Analisis perhitungan :

Agar dapat melaukan tahap perhitungan greyscale maka dibutuhkan citra masukan, berikut ini merupakan citra masukan data 1, yaitu citra beras yang telah dilakukan proses resizing:



Gambar 0.9 Citra Beras melalui Resize

A=,255, R=,220, G=,204, B=,143	A=,255, R=,219, G=,205, B=,144	A=,255, R=,208, G=,189, B=,130
A=,255, R=,214, G=,198, B=,136	A=,255, R=,216, G=,202, B=,141	A=,255, R=,240, G=,226, B=,165
A=,255, R=,224, G=,208, B=,146	A=,255, R=,209, G=,193, B=,133	A=,255, R=,253, G=,243, B=,181

Gambar 0.10 Nilai RGB beras 3x3

Perhitungan nilai Matriks :

Perhitungan piksel (0.0) :

X = 0,21 * 220 + 0,72 * 204 + 0.07 * 143

X = 46,2 + 146,88 + 10,1

X = 203.18 = 203

Perhitungan piksel (0.1) :

X = 0,21 * 219 + 0,72 * 205 + 0.07 * 144

X = 45,99 + 147,6 + 10,8

X = 203,67 = 204

Perhitungan piksel (0.2) :

X = 0,21 * 208 + 0,72 * 189 + 0.07 * 130

X = 43,68 + 136,8 + 9,1

X = 188,86 = 189

Dengan menggunakan rumus yang sama pada semua piksel maka akan di dapatkan hasil matriks sebagai berikut :

203	204	189
197	201	220
207	192	241

Gambar 0.11 Matriks Greyscale Citra Beras 3x3

3. Kuantisasi

Setelah dilakukan konversi dari mode RGB ke greyscale, maka akan didapatkan 1 nilai derajat keabuan dengan rentang nilai keabuan 0-255, selanjutnya akan dilakukan proses kuantisasi citra yang sudah melalui tahap greyscale dengan rentang nilai 0-255 menjadi 0-15. Proses ini bertujuan agar menghemat waktu pemrosesan komputasi.

Rumus yang digunakan untuk proses penskalaan citra greyscale nilai keabuan 0-255 menjadi 0-15 adalah dengan persamaan berikut:

$Xb=int(X*(2^n -1) / 255)$

Dimana :

Xb = Nilai GreyScale Baru

X = Nilai Greyscale Lama

2^n = Nilai Derajat Keabuan Baru

Perhitungan piksel (0,0):

Xb = int(203*15/255) = 11,94

Xb = 12

Perhitungan piksel (0,1):

Xb = int(204*15/255) = 12

Xb = 12

Perhitungan piksel (0,1):

Xb = int(189*15/255) = 12,29

Xb = 11

M(G)	0	1	2
0	12	12	11
1	12	12	4
2	13	11	14

Gambar 0.12 Matriks Greyscale citra beras range 0-15

3.5.1.1 Ekstraksi Co-occurrence Matrix

Metode Matrix Co-occurrence merupakan metode untuk ekstraksi ciri, dimana nilai ekstraksi ciri yang akan dicari adalah nilai *Contras*, *Homogenitas*, *energy*, dan *dissimilarity*. Sebagai contoh, citra kacang hijau dibawah ini telah melalui

preprocesssing dengan ukuran 3x3 pixel yang memiliki 16 derajat keabuan dengan rentang dari 0-15.



Gambar 0.13 Citra Greyscale range 0-15

Langkah-langkah untuk ekstraksi ciri dengan metode co-occurrence matrix adalah:

1. Membuat area kerja / framework

Pada gambar 3.3 memiliki 16 derajat keabuan, maka jumlah piksel ketetanggan dan nilai pixel referensi pada area kerja matriks berjumlah 16, seperti pada tabel 3.4 berikut:

Tabel 0.1 Framework rentang 0-15

P	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	0,10	0,11	0,12	0,13	0,14	0,15
1	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10	1,11	1,12	1,13	1,14	1,15
2	2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10	2,11	2,12	2,13	2,14	2,15
3	3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9	3,10	3,11	3,12	3,13	3,14	3,15
4	4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	4,10	4,11	4,12	4,13	4,14	4,15
5	5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9	5,10	5,11	5,12	5,13	5,14	5,15
6	6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9	6,10	6,11	6,12	6,13	6,14	6,15
7	7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9	7,10	7,11	7,12	7,13	7,14	7,15
8	8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9	8,10	8,11	8,12	8,13	8,14	8,15
9	9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9	9,10	9,11	9,12	9,13	9,14	9,15
10	10,0	10,1	10,2	10,3	10,4	10,5	10,6	10,7	10,8	10,9	10,10	10,11	10,12	10,13	10,14	10,15
11	11,0	11,1	11,2	11,3	11,4	11,5	11,6	11,7	11,8	11,9	11,10	11,11	11,12	11,13	11,14	11,15
12	12,0	12,1	12,2	12,3	12,4	12,5	12,6	12,7	12,8	12,9	12,10	12,11	12,12	12,13	12,14	12,15
13	13,0	13,1	13,2	13,3	13,4	13,5	13,6	13,7	13,8	13,9	13,10	13,11	13,12	13,13	13,14	13,15
14	14,0	14,1	14,2	14,3	14,4	14,5	14,6	14,7	14,8	14,9	14,10	14,11	14,12	14,13	14,14	14,15
15	15,0	15,1	15,2	15,3	15,4	15,5	15,6	15,7	15,8	15,9	15,10	15,11	15,12	15,13	15,14	15,15

2. Membentuk Co-occurrence Matrix

Untuk proses seslanjutnya adalah menghitung nilai dari matriksdengan mengisikan jumlah hubungan spasial. Matriks kookurensi yang akan dibuat adalah hubungan spasial untuk d=1 dan sudut 0^0. Matriks dibuat dengan mengisikan

jumlah hubungan sasial yang ada matriks grayscale. Maka didapatkanlah matriks kookurensi seperti pada gambar berikut:

Tabel 0.2 Co-occurrence Matrik

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	11	5	3	1	0
11	0	0	0	0	0	0	0	0	1	0	9	96	77	16	0	0
12	0	0	0	0	0	0	0	0	1	2	12	80	278	75	7	0
13	0	0	0	0	0	0	0	0	0	0	2	12	83	192	8	0
14	0	0	0	0	0	0	0	0	0	0	0	1	7	8	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3. Mentranspose Co-occurrence Matrix

Matrix co-occurrence yang sudah didapat selanjutnya di transpose, transpose merupakan proses mengubah matriks baris menjadi kolom, tujuan dari proses ini adalah untuk mendapatkan sudut simetrisnya yaitu sudut 180^0. Berikut merupakan hasil pembentukan matriks transpose :

Tabel 0.3 Tranpose Matriks

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	2	0	1	1	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	9	12	2	0	0
11	0	0	0	0	0	0	0	0	0	0	11	96	80	12	1	0
12	0	0	0	0	0	0	0	0	0	2	5	77	278	83	7	0
13	0	0	0	0	0	0	0	0	0	0	3	16	75	192	8	0
14	0	0	0	0	0	0	0	0	0	0	1	0	7	8	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4. Membuat Matriks Simetris

Pada proses ini adalah menjumlahkan co-occurrence matriks dengan transpose matriks agar membentuk simetris antara sudut 0^0 dan sudut 180^0. Seperti matriks berikut ini :

Tabel 0.4 Simetris Matriks

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	4	0	1	3	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	4	0	0	0
10	0	0	0	0	0	0	0	0	0	0	2	20	17	5	1	0
11	0	0	0	0	0	0	0	0	1	0	20	192	157	28	1	0
12	0	0	0	0	0	0	0	0	0	4	17	157	556	158	14	0
13	0	0	0	0	0	0	0	0	0	0	5	28	158	384	16	0
14	0	0	0	0	0	0	0	0	0	0	1	1	14	16	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5. Membuat Matriks Normalisasi

Untuk mendapatkan nilai dari normalisasi yaitu dengan cara menjumlahkan semua matriks simetris, kemudian dijadikan pembagi untuk semua piksel yang ada pada matriks simetris.untuk mencari probabilitas maka digunakan perhitungan P(i,j) = (i,j)/total jumlah pasangan

Dimana total jumlah pasangan = 1986

Contoh perhitungan P(12,9) = 4/1985 = 0,002014

Berikut tabel matriks hasil perhitungan normalisasi:

Tabel 0.5 Normalisasi matriks

M		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0,002014	0	0,000504	0,001511	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0,000504	0	0	0,002014	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0,001007	0,01007	0,00856	0,002518	0,000504	0
11	0	0	0	0	0	0	0	0	0,000504	0	0,01007	0,096677	0,079053	0,014099	0,000504	0
12	0	0	0	0	0	0	0	0	0	0,002014	0,00856	0,079053	0,27996	0,079557	0,007049	0
13	0	0	0	0	0	0	0	0	0	0	0,002518	0,014099	0,079557	0,193353	0,008056	0
14	0	0	0	0	0	0	0	0	0	0	0,002014	0,002014	0,07049	0,008056	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3.5.1.2 Ekstraksi Ciri Fitur

Setelah didapatkan nilai co-occurrence matriks maka proses selanjutnya adalah menghitung nilai ciri statistiknya. Ciri statistik yang akan di hitung adalah *contras*, *homogenitas*, *energy*, *entropy* dan *dissimilarity*. Berikut adalah perhitungan fitur tekstur :

1. Kontras

Rumus $Kontras = \sum_{i,j=0}^{n-1} (Pi,\ j).(i-j)^2$

Dimana :

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan

p (i,j) adalah Probabilitas kolom(i,j)

Syarat : Ketika nilai i dan j sama, sel berada pada diagonal dan (i-j) = 0. Nilai-nilai ini merepresentasikan pixel yang keseluruhannya mirip dengan tetangga mereka, sehingga mereka diberi bobot 0

Contoh perhitungan :

$P(12,9).(12-9)^2$ = 0,002014 . 9 = 0,018126

Berikut tabel hasil perhitungan kontras :

Tabel 0.6 Matriks Kontras

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0,013609	0,008065	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0,018145	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0,010081	0,034274	0,022681	0,008065	0
11	0	0	0	0	0	0	0	0	0,013609	0	0,010081	0	0,079133	0,056452	0,004536	0
12	0	0	0	0	0	0	0	0	0,008065	0,018145	0,034274	0,079133	0	0,079637	0,028226	0
13	0	0	0	0	0	0	0	0	0	0	0,022681	0,056452	0,079637	0	0,008065	0
14	0	0	0	0	0	0	0	0	0	0	0,008065	0,004536	0,028226	0,008065	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2. Homogenitas

Rumus homogenitas :

$$\sum_{i,j=0}^{n-1} P(i, j) / [1 + (i - j)^2]$$

Dimana:

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan

p (i,j) adalah *Probabilitas kolom(i,j)*

contoh perhitungan :

$P(12,9) / [1 + (12 - 9)^2] = 0,002014/10 = 0,000201$

Tabel 0.7 Matriks Homogen

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0,000151	2,96E-05	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0,000202	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0,001008	0,00504	0,001714	0,000252	2,96E-05	0
11	0	0	0	0	0	0	0	0	0,000151	0	0,00504	0,096774	0,039567	0,002823	5,04E-05	0
12	0	0	0	0	0	0	0	0	2,96E-05	0,000202	0,001714	0,039567	0,280242	0,039819	0,001411	0
13	0	0	0	0	0	0	0	0	0	0	0,000252	0,002823	0,039819	0,193548	0,004032	0
14	0	0	0	0	0	0	0	0	0	0	2,96E-05	5,04E-05	0,001411	0,004032	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3. Entropy

Rumus $Entropi = -\sum_{i,j=0}^{n-1} P(i, j) \ \log(P(i, j))$

Dimana:

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan

p (i,j) adalah *Cooccurence Matrix* Simetris Ternormalisasi

contoh perhitungan $-(P(i, j) \ \log(P(i, j)))$

= -(0,002014.log(0,002014))

= 0.00542962

Tabel 0.8 Matriks Entropi

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0,004265	0,001662	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0,005434	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0,003021	0,020126	0,017712	0,006549	0,001662	0
11	0	0	0	0	0	0	0	0	0,004265	0	0,020126	0,098152	0,087176	0,026114	0,001662	0
12	0	0	0	0	0	0	0	0	0,001662	0,005434	0,017712	0,087176	0,154824	0,087512	0,015181	0
13	0	0	0	0	0	0	0	0	0	0	0,006549	0,026114	0,087512	0,138041	0,016882	0
14	0	0	0	0	0	0	0	0	0	0	0,001662	0,001662	0,015181	0,016882	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4. Energy

Rumus $Energi = \sum_{i,j=0}^{n-1} P(i,j)^2$

Dimana:

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan

p (i,j) adalah *Cooccurence Matrix* Simetris Ternormalisasi

contoh perhitungan $P(i,j)^2 = (0{,}002014)^2 = 4.0562\text{E-}06$

Tabel 0.9 Matriks Energy

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	2,29E-06	2,54E-07	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	4,06E-06	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1,02E-06	0,000102	7,34E-05	6,35E-06	2,54E-07	0
11	0	0	0	0	0	0	0	0	2,29E-06	0	0,000102	0,009365	0,006262	0,000199	2,54E-07	0
12	0	0	0	0	0	0	0	0	2,54E-07	4,06E-06	7,34E-05	0,006262	0,078536	0,006342	4,98E-05	0
13	0	0	0	0	0	0	0	0	0	0	6,35E-06	0,000199	0,006342	0,037461	6,50E-05	0
14	0	0	0	0	0	0	0	0	0	0	2,54E-07	2,54E-07	4,98E-05	6,50E-05	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabel 0.10 Matriks Dissimilarity

M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0,004536	0,002016	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0,006048	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0,010081	0,017137	0,00756	0,002016	0
11	0	0	0	0	0	0	0	0	0,004536	0	0,010081	0	0,079133	0,028226	0,001512	0
12	0	0	0	0	0	0	0	0	0,002016	0,006048	0,017137	0,079133	0	0,079637	0,014113	0
13	0	0	0	0	0	0	0	0	0	0	0,00756	0,028226	0,079637	0	0,008065	0
14	0	0	0	0	0	0	0	0	0	0	0,002016	0,001512	0,014113	0,008065	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Setelah semua proses perhitungan fitur dilakukan, langkah selanjutnya adalah menjumlahkan tiap matriks dari fitur tersebut sehingga didapatkan hasil sebagai berikut :

Tabel 0.11 Nilai Fitur Sudut Simetris 0^0, 〖45〗^0, 〖90〗^0, 〖135〗^0

Fitur	Sudut 0	Sudut 45	Sudut 90	Sudut 135
Kontras	0,74193548	0,92924037	0,76209677	0,92924037
Homogenitas	0,76181214	0,71393816	0,75229529	0,71393816
Entropy	0,97791544	1,01666636	0,98840619	1,01666636
Energy	0,15157602	0,13205439	0,14529492	0,13205439
Dissimilarity	0,52016129	0,62955255	0,53830645	0,62955255

Langkah selanjutnya adalah merata-ratakan nilai nilai-nilai dari semua sudut agar didapatkan satu nilai tunggal dari setiap fitur yang telah didapatkan sehingga memudahkan dalam penamaan klasifikasi.berikut nama klasifikasi dengan nilai fitur yang telah dirata-ratakan:

Tabel 0.12 Nama klasifikasi dan rata-rata fitur

Kelas	Kontras	Homogenitas	Entropy	Energy	Dissimilarity
Kw1	0,840628252	0,735495937	0,999913585	0,140244928	0,57939321

3.6 Analisis K-Nearest Neighbor

K-Nearest Neighbor merupakan sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. KNN termasuk dalam golongan *supervised* , dimana hasil query *instance* yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam KNN. Nantinya kelas yang baru dari suatu data akan dipilih berdasarkan grup kelas yang dekat jarak vektornya.[8]

Berikut ini merupakan data fitur dari ekstraksi ciri yang telah dilakukan :

Tabel 0.13 Data Fitur Ekstraksi Ciri

NO	Kontras	Homogen	Entropy	Energy	Dissimilarity	Kualitas
1	0,84063	0,73549594	0,99991	0,1402449	0,5793932	KW1
2	0,86502	0,71461713	0,9921	0,157529	0,6193256	KW2
3	0,84176	0,73543265	1,02263	0,1508495	0,5808159	KW2
4	0,71875	0,75252378	0,97816	0,1719445	0,531754	KW3
5	0,99741	0,72354421	1,06102	0,1468904	0,625626	KW3
6	0,88041	0,72738139	0,99813	0,1684139	0,6005626	KW3
7	0,78626	0,74458208	1,01596	0,1491679	0,5560126	KW3
8	0,78768	0,73004611	1,05025	0,1250305	0,5805395	KW3
9	0,7671	0,74833839	0,97848	0,1772233	0,545745	KW3
10	0,82856	0,73098391	1,0254	0,1498693	0,5858969	KW3
11	0,86661	0,7347063	1,0016	0,1583219	0,5858643	KW3
12	1,09934	0,70566359	1,06476	0,1336976	0,6688102	KW3
13	0,7745	0,73378537	1,00315	0,1511677	0,5725969	KW3
14	0,98534	0,72181986	1,06477	0,1383272	0,6252032	KW1
15	0,89584	0,72037709	1,03308	0,1460091	0,6145535	KW1
16	0,88325	0,72378444	1,0434	0,1371562	0,6072044	KW1
17	0,94905	0,71179787	1,10545	0,1143583	0,6375114	KW1
18	0,96721	0,70730541	1,08783	0,1299694	0,6485432	KW1
19	0,93986	0,7051477	1,04588	0,1281893	0,6462344	KW1
20	1,14044	0,70003985	1,04454	0,1461614	0,6871829	KW2
21	0,82856	0,73098391	1,0254	0,1498693	0,5858969	KW2
22	0,7671	0,74833839	0,97848	0,1772233	0,545745	KW2
23	0,78768	0,73004611	1,05025	0,1250305	0,5805395	KW2
24	0,78626	0,74458208	1,01596	0,1491679	0,5560126	KW2
25	0,88041	0,72738139	0,99813	0,1684139	0,6005626	KW2

Data uji : terdapat nilai fitur citra baru yaitu sebagai berikut :

Tabel 0.14 Data Fitur Ekstraksi baru

Kontras	Homogen	Entropy	Energy	Dissimilarity	Kualitas
0,71875	0,752523781	0,97816	0,17194454	0,53175403	?

Langkah pertama yang dilakukan adalah menentukan K tetangga terdekat. Nilai K yang terbaik untuk metode ini tergantung pada data. Secara umum nilai K yang tinggi akan mengurangi efek noise pada klasifikasi tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur[11]. Untuk itu kita ambil nilai K adalah ¼ dari data uji. Dalam contoh kasus ini adalah 6,5 ≈ 6 sehingga, selanjutnya hitung kuadrat jarak euclid pada masing-masing objek dengan rumus euclidean distance yaitu :

$$d(x-y) = \sqrt{\sum_{j=1}^{i}(x_j - y_j)^2} \qquad \text{(Persamaan 7)}$$

Keterangan :

d = jarak data uji ke data pembelajaran.

x_j = data uji ke-j, dengan j = 1, 2, . . . n.

y_j = data pembelajaran ke-j dengan j = 1, 2, . . . n.

Urutkan berdasarkan nilai fitur terkecil:

Tabel 0.15 Hasil Pengurutan Fitur

NO	Kontras	Homogen	Entropy	Energy	Dissimilarity	Kualitas	Euclidian Distance
1	0,84063	0,73549594	0,99991359	0,14024493	0,57939321	KW1	2,60280244
4	0,71875	0,75252378	0,97815542	0,17194454	0,53175403	KW3	2,352008903
9	0,7671	0,74833839	0,97848427	0,17722334	0,54574499	KW3	2,43512448
22	0,7671	0,74833839	0,97848427	0,17722334	0,54574499	KW2	2,43512448
13	0,7745	0,73378537	1,00315364	0,15116771	0,5725969	KW3	2,49533117
7	0,78626	0,74458208	1,01595654	0,14916792	0,55601262	KW3	2,536172605
24	0,78626	0,74458208	1,01595654	0,14916792	0,55601262	KW2	2,536172605
8	0,78768	0,73004611	1,05024585	0,12503055	0,58053948	KW3	2,609082919
23	0,78768	0,73004611	1,05024585	0,12503055	0,58053948	KW2	2,609082919
10	0,82856	0,73098391	1,02539607	0,14986927	0,58589685	KW3	2,6380288
21	0,82856	0,73098391	1,02539607	0,14986927	0,58589685	KW2	2,6380288
2	0,86502	0,71461713	0,99210331	0,15752899	0,61932557	KW2	2,651580422
3	0,84176	0,73543265	1,02263229	0,15084949	0,58081588	KW2	2,655297614
11	0,86661	0,7347063	1,00160353	0,15832186	0,58586433	KW3	2,662319219
6	0,88041	0,72738139	0,99812988	0,1684139	0,60056257	KW3	2,689514886
25	0,88041	0,72738139	0,99812988	0,1684139	0,60056257	KW2	2,689514886
16	0,88325	0,72378444	1,04339988	0,13715622	0,60720441	KW1	2,780189458
15	0,89584	0,72037709	1,03308042	0,14600906	0,61455352	KW1	2,787715194
19	0,93986	0,7051477	1,04587929	0,12818929	0,64623439	KW1	2,908480404
14	0,98534	0,72181986	1,0647745	0,13832716	0,62520324	KW1	3,035681829
17	0,94905	0,71179787	1,1054542	0,11435825	0,63751138	KW1	3,048883632
18	0,96721	0,70730541	1,08782619	0,12996939	0,64854318	KW1	3,056633328
5	0,99741	0,72354421	1,06102021	0,1468904	0,62562598	KW3	3,057084841
12	1,09934	0,70566359	1,06476113	0,13369764	0,66881016	KW3	3,305396907
20	1,14044	0,70003985	1,04453961	0,14616138	0,68718295	KW2	3,375301213

Dengan nilai K yang telah ditentukan sebelumnya, maka kelas-kelas terdekat adalah seperti pada tabel berikut :

Tabel 0.16 Kelas Fitur terdekat

NO	Kontras	Homogen	Entropy	Energy	Dissimilarity	Kualitas	Euclidian Distance
1	0,84063	0,73549594	0,99991359	0,14024493	0,57939321	KW1	2,60280244
4	0,71875	0,75252378	0,97815542	0,17194454	0,53175403	KW3	2,352008903
9	0,7671	0,74833839	0,97848427	0,17722334	0,54574499	KW3	2,43512448
22	0,7671	0,74833839	0,97848427	0,17722334	0,54574499	KW2	2,43512448
13	0,7745	0,73378537	1,00315364	0,15116771	0,5725969	KW3	2,49533117
7	0,78626	0,74458208	1,01595654	0,14916792	0,55601262	KW3	2,536172605

Karena mayoritas kelasnya adalah KW3, maka data baru yang menjadi perbandingan termasuk kelas KW3.

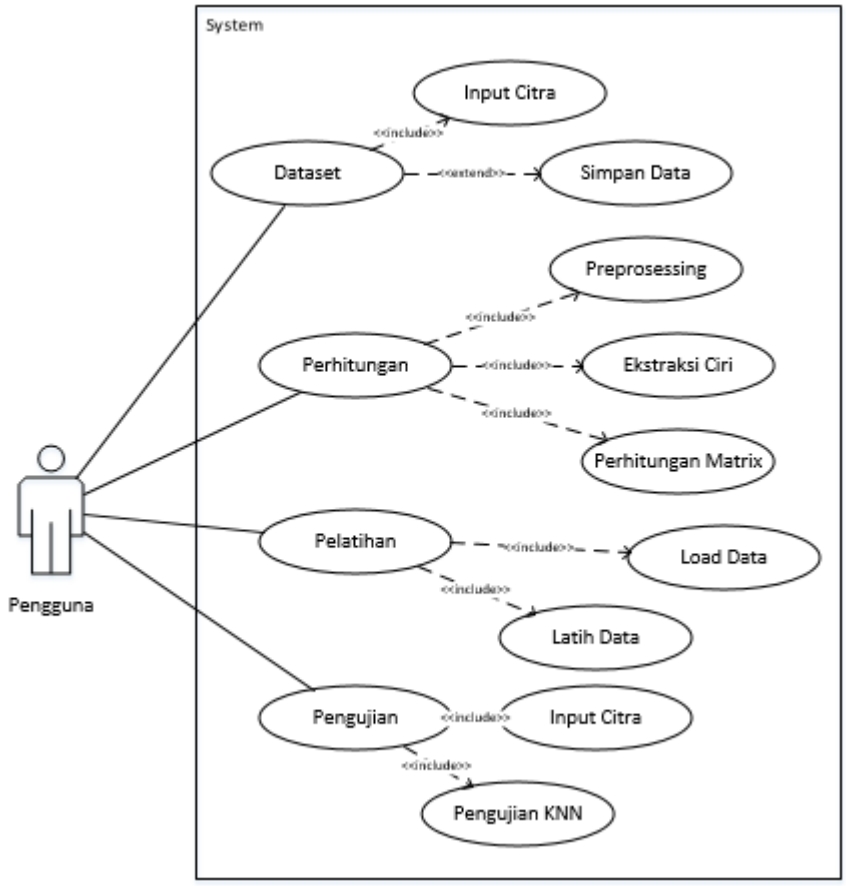
3.7 Analisis Kebutuhan Sistem

3.7.1 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan gambaran spesifikasi tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan. Analisis kebutuhan ini dipelukan untuk menentukan hasil keluaran pada sebuah sistem, masukan yang diperlukan sistem lingkup proses yang digunakan untuk mengolah menjadi keluaran.

1. Use case diagram

Use case diagram adalah pemodelan yang menunjukan interaksi antara sistem dan lingkungannya. Use case diagram memberikan gambaran yang cukup sederhana dari interaksi-interaksi yang terlibat. Berikut adalah perancangan proses-proses pada sistem yang akan dibangun digambarkan dengan use case diagram dibawah ini:



Gambar 0.14 Use case diagram

a. Definisi Actor

Actor merupakan pihak yang mengakses use case dengan berperan sebagai pengguna yang akan menggunakan sistem. Definisi actor pada penelitian ini sebagai berikut :

Tabel 0.17 Definisi Aktor

No	Actor	Desktipsi
1	Pengguna	Pengguna Aplikasi

b. Definisi Use Case

Use case berfungsi untuk mewakili apa yang sistem bisa lakukan. Definisi use case bisa dilihat pada tabel dibawah ini:

Tabel 0.18 Definisi Use case

No	*Use Case*	Deskripsi
1	Dataset	menu tambah dataset
2	Pengujian	menu pengujian
3	Pelatihan	Menu Pelatihan
4	Perhitungan	Menu perhitungan
5	Pengujian Knn	Proses Pengujian dengan metode KNN
6	Input citra	proses untuk mengginputkan gambar dalam menu tambah dataset
7	Praprosesing	proses untuk melakukan proses preprocessing
8	Ekstraksi ciri	proses untuk melakukan ekstraksi ciri setelah melakukan proses preprocessing sebelumnya
9	Simpan dataset	proses untuk menyimpan hasil ekstraksi kedalam database
10	Load data	proses untuk memanggil data sampel
11	Latih data	Proses untuk melatih data sampel
12	Input citra	proses untuk menginputkan citra uji
13	Uji	proses untuk melakukan pengujian

c. Skenario Use case

Skenario use case merupakan hasil intansiasi dari setiap *use case*. Berikut skenario use case :

Tabel 0.19 Skenario usecase Dataset

Nama *Use Case*	Dataset	
Tujuan	Menampilkan menu dataset	
Kondisi Awal	Tampilan masih di menu utama	
Kondisi Akhir	Menampilkan menu input citra, ekstraksi ciri, dan perhitungan matriks dan simpan dataset	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1.	Pengguna Menekan menu Dataset	
2.		sistem menampilkan tampilan menu dataset

Tabel 0.20 Skenario Usecase Tambah Data

Tujuan	Menambahkan dataset untuk proses pelatihan	
Kondisi Awal	Dataset untuk proses pengujian belum cukup / belum tersedia	
Kondisi Akhir	Dataset berhasil ditambahkan	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1	Pengguna Memasukan Citra Inputa	
2		Sistem menampilkan citra inputan
3	Pengguna menekan tombol proses	
4 Include : Proses Preprossesing		Sistem Melakukan Proses Preprocessing
5 Include : Proses Ekstraksi Ciri		Sistem Melakukan Proses Ekstraksi Ciri
6	Pengguna memilih nama kelas	

7	Pengguna menekan tombol simpan	
8. Extend : Simpan Data		Sistem menyimpan dataset pada database

Tabel 0.21 Input Citra

Nama *Use Case*	Input Citra	
Tujuan	Menampilkan Citra yang akan disimpan	
Kondisi Awal	Belum ada citra yang di inputkan	
Kondisi Akhir	Citra berhasil di inputkan	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1.	pengguna menekan tombol input citra	
2.		Sistem menampilkan dialogbox browse file untuk input gambar
3.	Pengguna memilih citra	
4.	Pengguna menekan tombol OK	
5. Include : Preporsessing		Sistem menampilkan citra pada imagebox

Tabel 0.22 Skenario Usecase Prerosessing

Nama *Use Case*	Prerposessing	
Tujuan	Menampilkan Citra yang telah melalui tahap resize, greyscale dan kuantisasi	
Kondisi Awal	Citra yang diinputkan masih dalam bentuk resize	
Kondisi Akhir	Citra yang di inputkan sudah melalui tahap greyscale dan kuantisasi	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1.	pengguna menekan tombol uji citra	

2.		Sistem melakukan proses preprocessing
3.		sistem menampilkan citra yang sudah melalui proses preprocessing

Tabel 0.23 Skenario Usecase Ekstraksi

Nama *Use Case*	Ekstraksi Ciri	
Tujuan	Menampilkan Nilai citra berdasarkan derajat jarak	
Kondisi Awal	Citra belum mempunyai nilai dari matriks ekstraksi ciri	
Kondisi Akhir	System menampilkan tabel matriks hasil ekstraksi ciri	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1.	pengguna menekan tombol Ekstraksi Ciri	
2.		System melakukan Proses Ekstraksi Ciri
3.		System menampilkan hasil ekstraksi ciri

Tabel 0.24 Skenario usecase Simpan data

Nama *Use Case*	Simpan Data	
Tujuan	Menyimpan data hasil ekstraksi ciri berdasarakan kelas	
Kondisi Awal	Citra belum tersimpan di database	
Kondisi Akhir	Citra tersimpan di database	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1.	Pengguna menekan tombol select kelas, Kualitas1, Kualitas2, atau Kualitas3	
2.		sistem menampilkan select box dengan pilihan kelas
3.	Pengguna memilih kelas	
4.	Pengguna menekan tombol simpan	

5.		Sistem menyimpan dataset citra berdasarkan kelas yang dipilih

Tabel 0.25 Skenario Usecase Pengujian

Nama *Use Case*	pengujian	
Tujuan	menghitung dan menampilkan hasil pengujian data	
Kondisi Awal	tampilan pengujian awal dan masih belum muncul satu angka pun di dalam halaman pengujian	
Kondisi Akhir	hasil pengujian tampil	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1.	Pengguna Menekan menu Pengujian	
2.		sistem menampilkan tampilan menu pengujian

Tabel 0.26 Skenario Usecase Load Data

Nama *Use Case*	Load Data	
Tujuan	Menampilkan dataset yang telah disimpan	
Kondisi Awal	Tampilan load data awal belum data	
Kondisi Akhir	System menampilkan data set	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1.	Pengguna menekan tombol Load data	
2.		sistem menampilkan nilai-nilai fitur dataset dari database

Nama *Use Case*	Perhitungan
Tujuan	Menampilkan menu dari perhitungan
Kondisi Awal	Tampilan masih di menu utama

Kondisi Akhir	System menampilkan menu perhitungan	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1.	Pengguna menekan tombol Perhitungan	
2.		sistem menampilkan menu perhitungan matriks

Tabel 0.27 Skenario Usecase Perhitungan Matriks

Nama *Use Case*	Perhitungan Matriks	
Tujuan	Menampilkan menu dari perhitungan matriks dalam bentuk tabel	
Kondisi Awal	Tampilan perhitungan masih kosong	
Kondisi Akhir	System menampilkan menu perhitungan dalam bentuk matriks	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1.	Pengguna menekan tombol Perhitungan matriks	
2.		sistem menampilkan menu perhitungan matriks dalam bentuk tabel

Tabel 0.28 Skenario usecase Pelatihan

Nama *Use Case*	Pelatihan	
Tujuan	Menampilkan menu pelatihan	
Kondisi Awal	Tampilan masih dimenu utama	
Kondisi Akhir	System menampilkan menu pelatihan	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1	Pengguna menekan tombol Load data matriks	

2.		Sisten Menampilkan data sampel pada database
3.	Pengguna memasukan nilai K ke textbox	
4.		Sistem menerima inputan parameter K
5.	Pengguna menekan tombol proses	
6.		Sistem memproses dengan metode KNN
7.		Sistem menampilkan hasil pengujian berupa klasifikasi citra berdasarkan kualitas

Tabel 0.29 Skenario usecase Load Data

Nama *Use Case*	Load Data	
Tujuan	Menampilkan data sampel	
Kondisi Awal	Tampilan Load data masih kosong	
Kondisi Akhir	System menampilkan data sampel pada database	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1	Pengguna menekan tombol load data	
2.		Sistem menampilkan data sampel pada database

Tabel 0.30 Skenario Usecase Latih Data

Nama *Use Case*	Latih Data	
Tujuan	Melatih data sampel	
Kondisi Awal	Tampilan Latih data masih kosong	
Kondisi Akhir	System menampilkan data sampel yang sudah di latih	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1	Pengguna menekan tombol Latih Data	

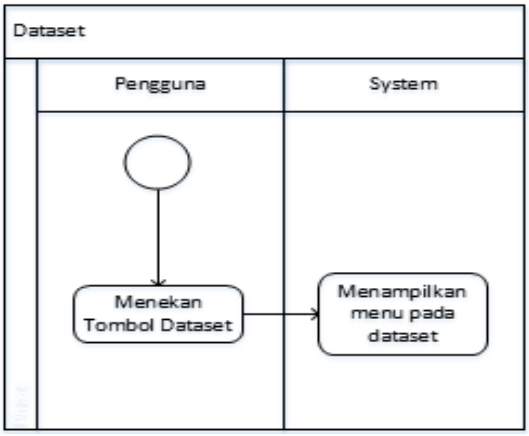
2.		Sistem menampilkan data sampel yang sudah di latih

Tabel 0.31 Skenario Pengujian KNN

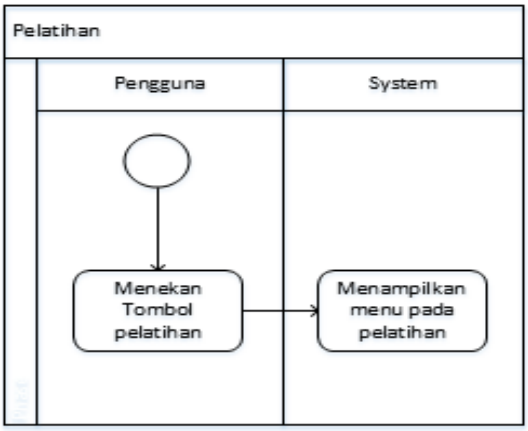
Nama *Use Case*	Pengujian KNN	
Tujuan	Menguji Citra dengan klasifikasi KNN	
Kondisi Awal	Tampilan hasil pengujian masih kosong	
Kondisi Akhir	System menampilkan hasil pengujian citra beras dengan metode KNN	
Aktor	Pengguna	
Skenario Utama		
Langkah	**Aksi Aktor**	**Reaksi System**
1 Include : Input Citra	Pengguna menekan tombol input citra matriks	
2.		Sistem menampilkan citra yang aka di uji
3.	Pengguna memasukan nilai K ke textbox	
4.		Sistem menerima inputan parameter K
5.	Pengguna menekan tombol proses	
6.		Sistem memproses dengan metode KNN
7.		Sistem menampilkan hasil pengujian berupa klasifikasi citra berdasarkan kualitas

2. Activity diagram

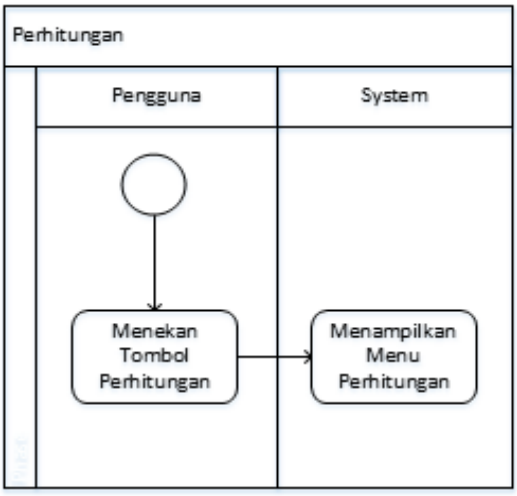
Diagram aktivitas (*Activity Diagram*) memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Adapun rincian dari activity diagram sistem dapat dilihat pada gambar beikut:



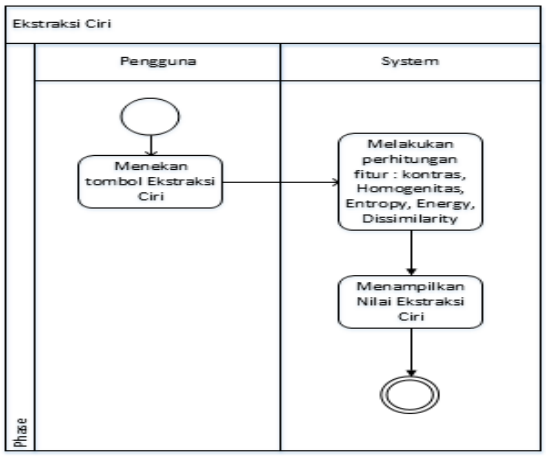
Gambar 0.15 Activity Dataset



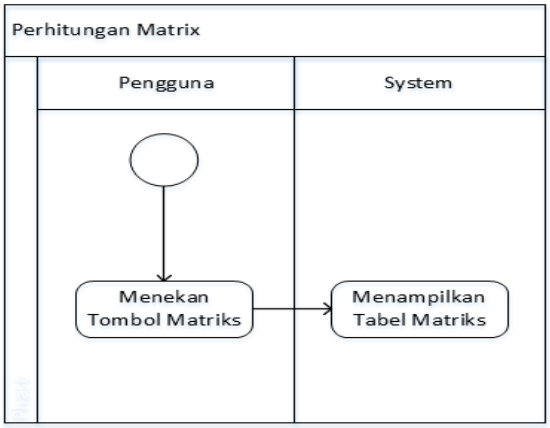
Gambar 0.16 Activity Pelatihan



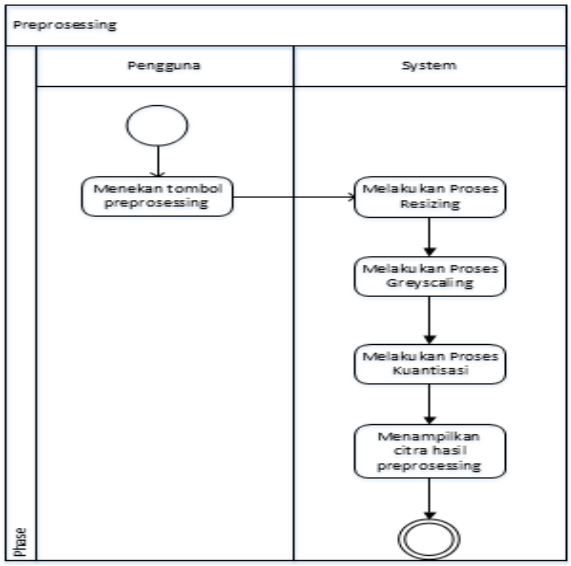
Gambar 0.17 Activity perhitungan



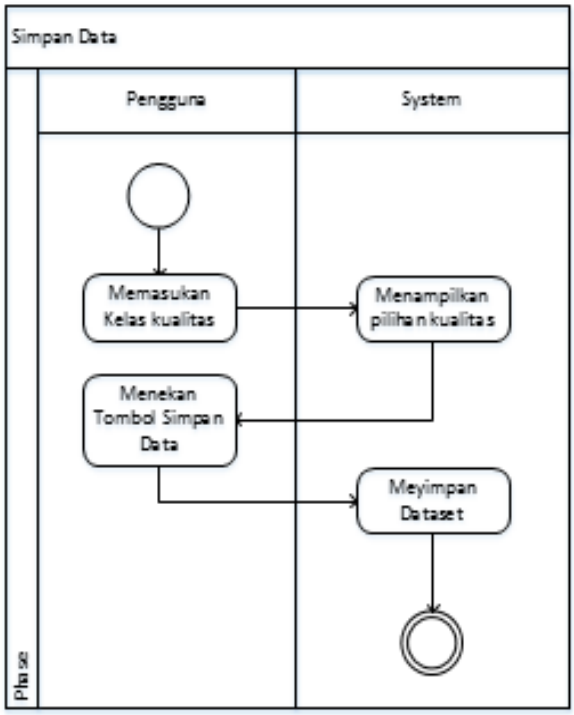
Gambar 0.18 Activity Ekstraksi Ciri



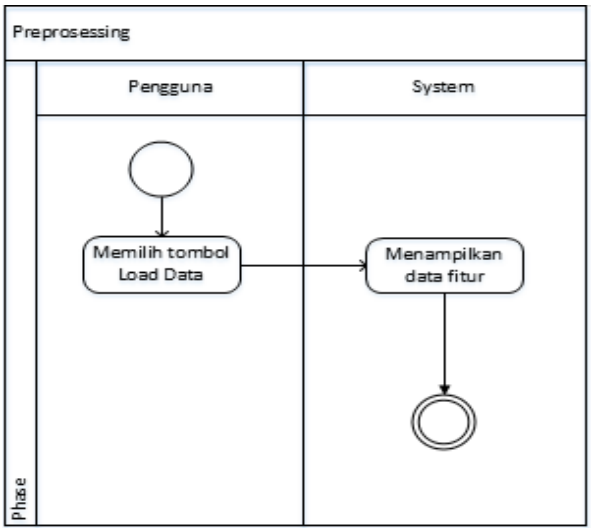
Gambar 0.19 Activity Perhitungan Matriks



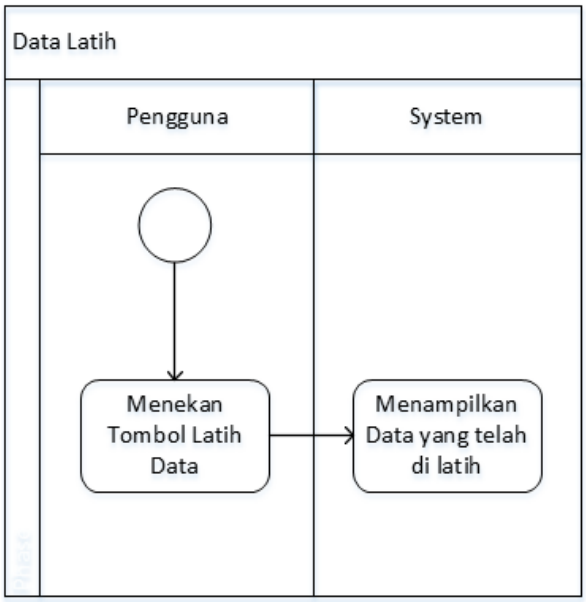
Gambar 0.20 Activity Preprosessing



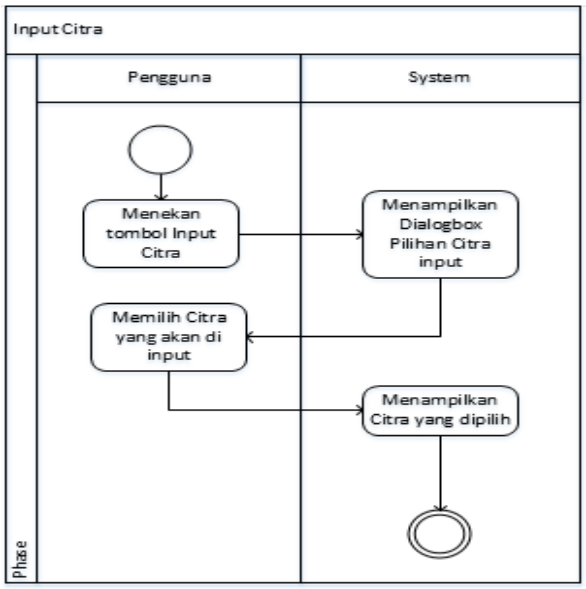
Gambar 0.21 Activity Simpan Data



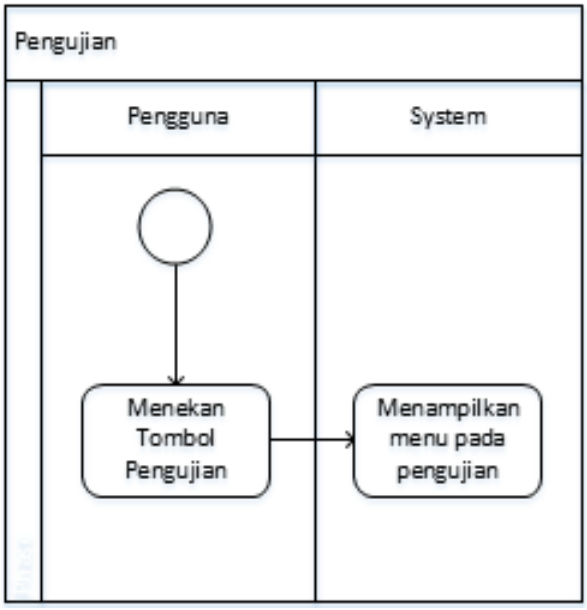
Gambar 0.22 Activity Prerosessing



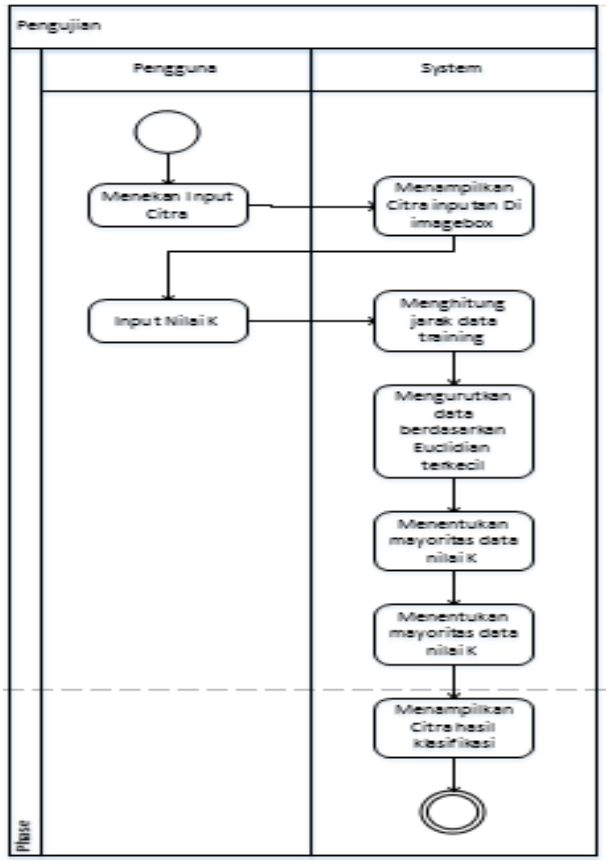
Gambar 0.23 Activity Data Latih



Gambar 0.24 Activity Input Citra



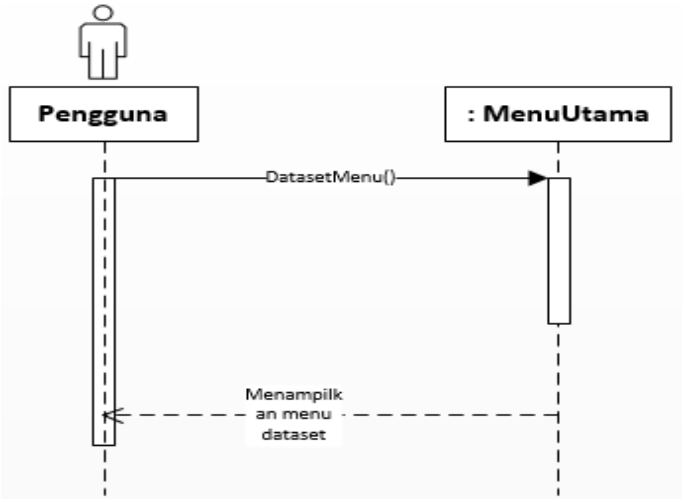
Gambar 0.25 Activity Pengujian



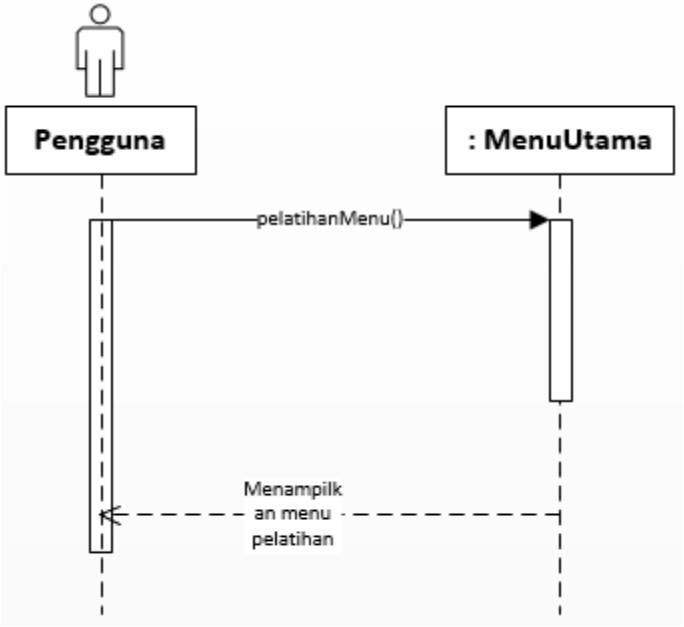
Gambar 0.26 Activity Pengujian

3. Sequence diagram

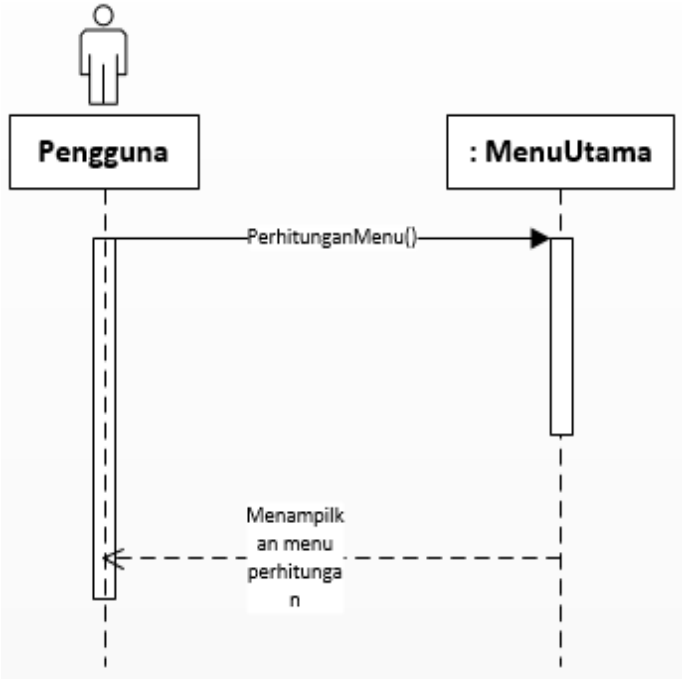
Sequence diagram menggambarkan interkasi antara sejumlah objek dalam urutan waktu. Adapun rincian dari sequence diagram sistem dapat dilihat pada gambar dibawah ini:



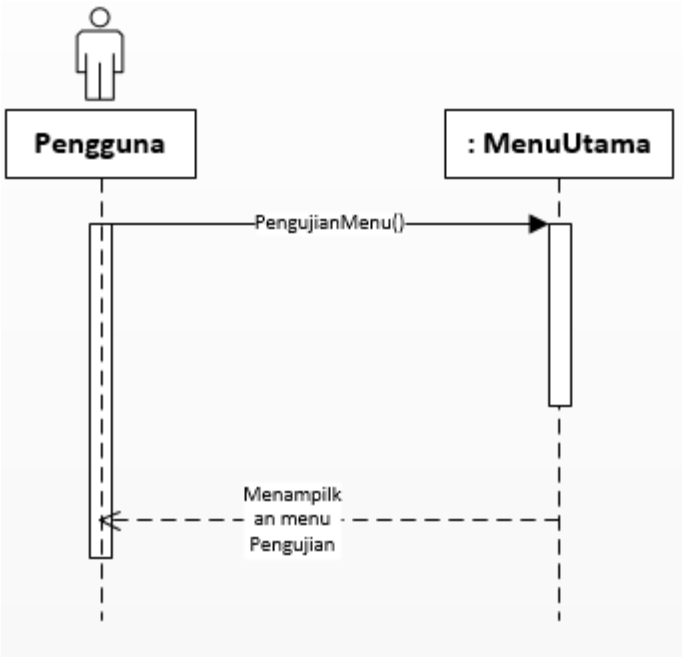
Gambar 0.27 Sequence Menu Dataset



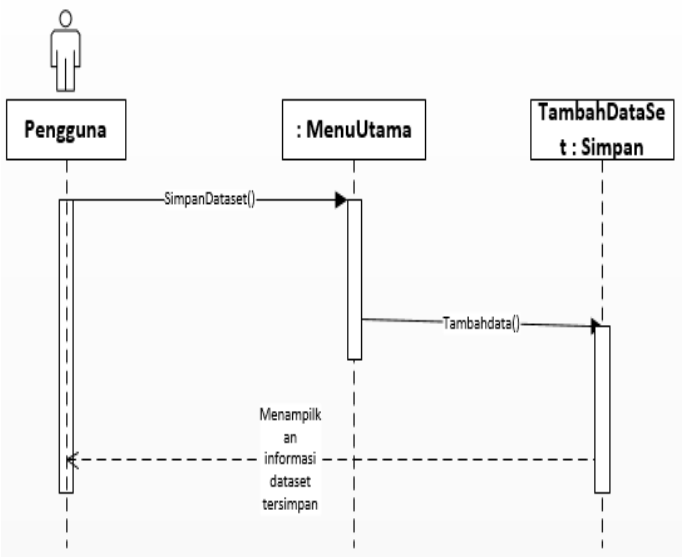
Gambar 0.28 Sequence menu Pelatihan



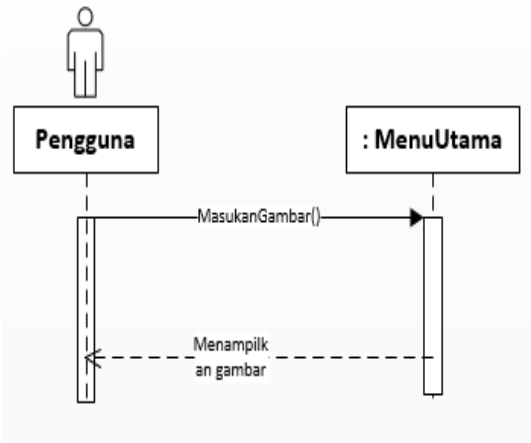
Gambar 0.29 Sequence Perhitungan



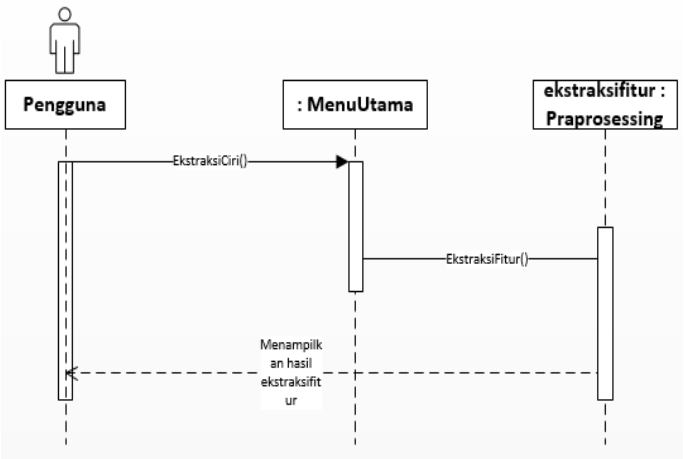
Gambar 0.30 Sequence Pengujian



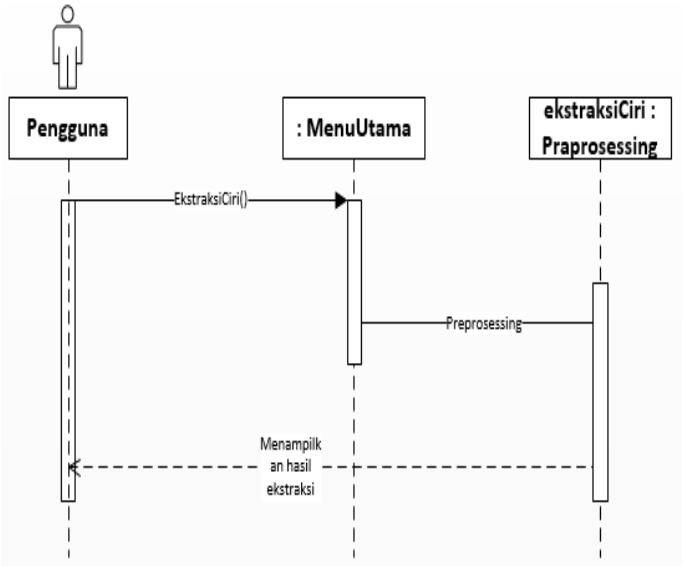
Gambar 0.31 Sequence simpan Dataset



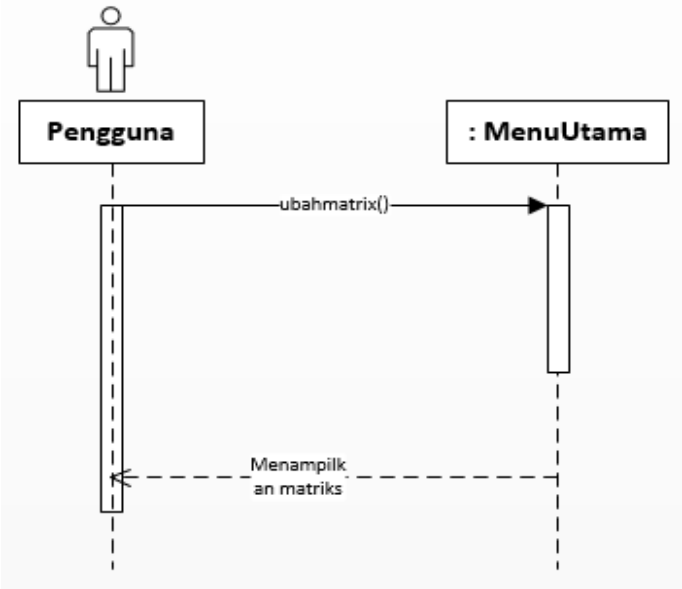
Gambar 0.32 Sequence Menampilkan gambar



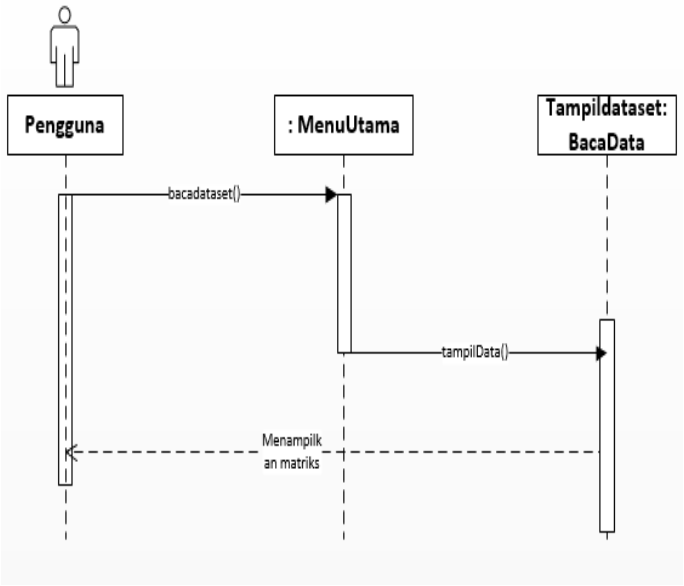
Gambar 0.33 Sequence Ekstraksi Fitur



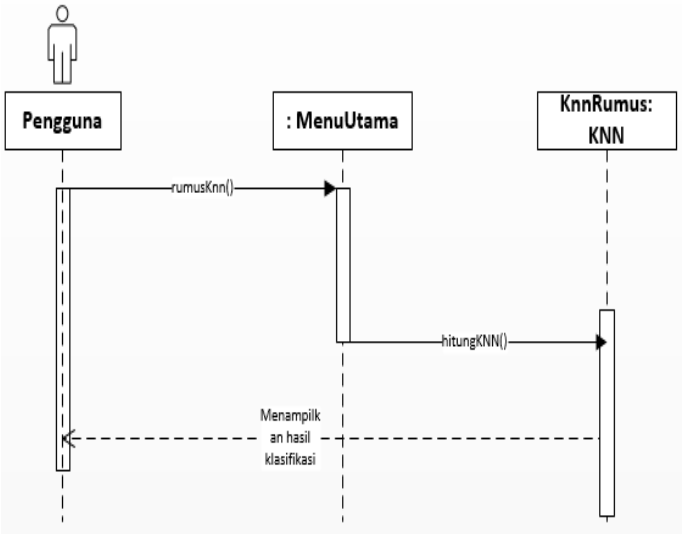
Gambar 0.34 Sequence Ekstraksi Ciri



Gambar 0.35 Sequence Ubah matriks



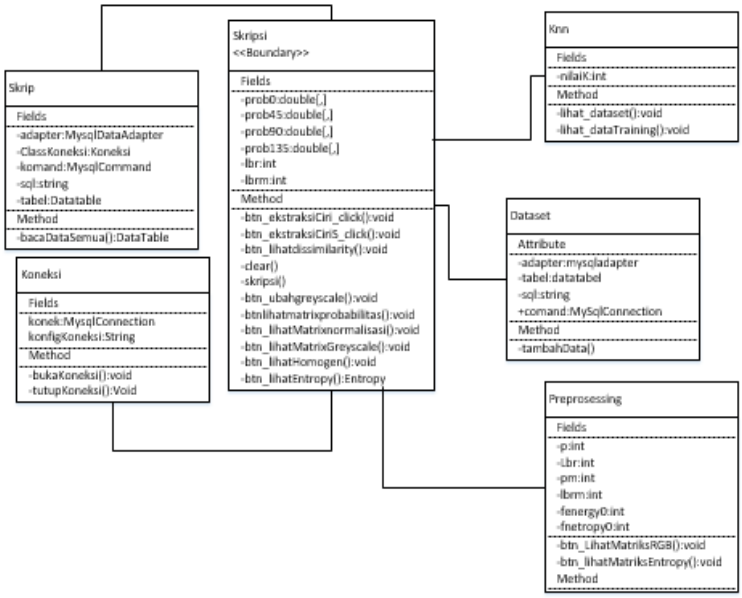
Gambar 0.36 Sequence Load data



Gambar 0.37 Sequence Pengujian KNN

4. Class diagram

Class diagram membantu menggambarkan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling sering ditemui dalam pemodelan sistem berbasis *object-oriented*. Adapun rincian dari class diagram sistem dapat dilihat pada gambar berikut :



Gambar 0.38 Class Diagram

3.7.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah langkah dimana seorang pembangun perangkat lunak (*software development*) menganlisis sumber daya yang akan digunakan dan menggunakan perangkat lunak yang dibangun. Perangkat keras dan perangkat lunak yang dimiliki harus sesuai dengan kebutuhan atau permintaan, sehingga dapat ditentukan kompatibilitasnyaterhadap sumber daya yang ada.

Pada analisis kebutuhan non fungsionalyang dilakukan dibagi dalam tiga tahap yaitu:

1. Analisis Kebutuhan Perangkat Keras

Spesifikasi perangkat keras minimum yang dibutuhkan untuk menjalankan sistem ini, dapat dilihat pada tabel dibawah ini:

Gambar 0.39 Spesifikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Prosesor	2.00 GHz
2	Monitor	Monitor 14.1''
3	Memori	4 GB

2. Analisis Kebutuhan Perangkat Lunak

Komponen perangkat lunak pendukung yang dibutuhkan untuk menjalankan sistem ini, seperti tabel dibawah ini:

Gambar 0.40 Spesifikasi Perangkat Lunak

No	Perangkat Lunak
1	Sistem operasi Windows 8.1
2	Visual Studio 2012

3. Analisis Pengguna

Analisis kebutuhan pengguna merupakan uraian mengenai siapa saja yang akan menggunkan sistem dan terlibat dalam pengolahan data berserta karakteristiknya sehingga dapat diketahui tingkat pengalaman dan pemahaman pengguna terhadap sistem. Dapat dilihat sebagai berikut:

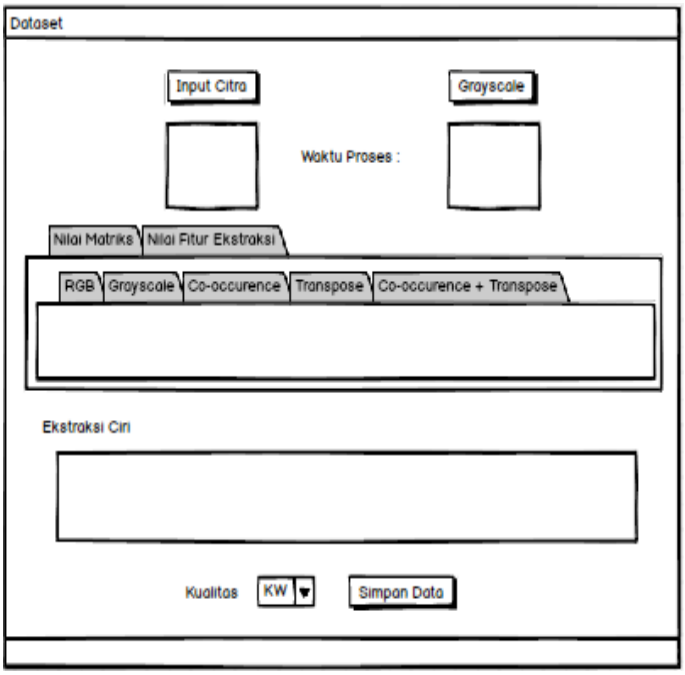
a. Pengguna dapat membaca tulisan
b. Pengguna minimal mampu menggunakan *keyboard* dan *mouse* sebagai untuk interaksi dengan sistem
c. Pengguna sudah terbiasa menggunakan komputer

4. Perancangan Simulasi

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

a. Perancangan AntarmukaPerancangan antarmuka dibuat untuk memberi konsep pada implementasi pembuatan suatu antarmuka sistem yang akan dibangun. Adapun rincian dari perancangan antarmuka sistem dapat dilihan pada gambar dibawah ini:

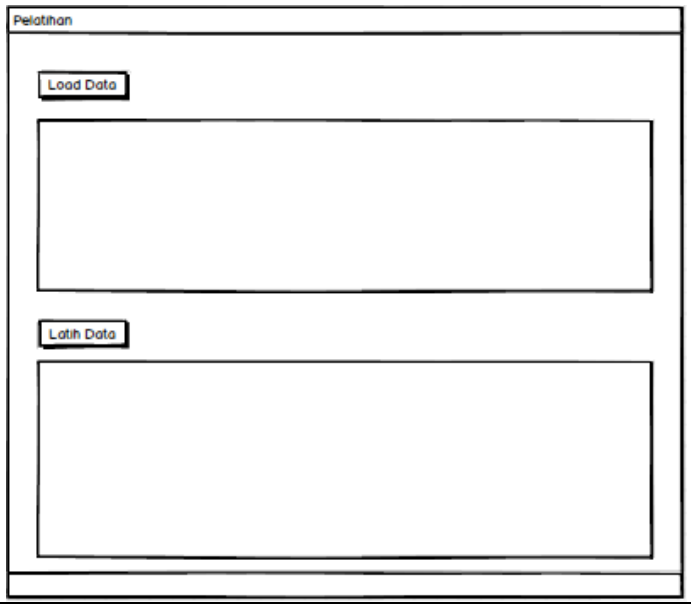
No F01



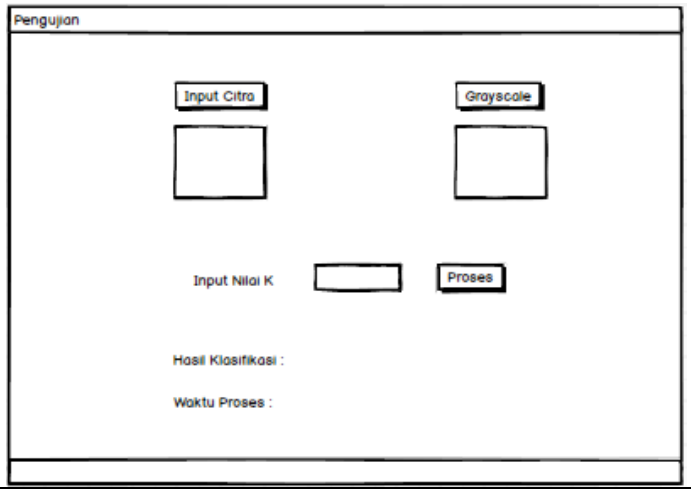
- Klik tombol **INPUT** untuk memasukan gambar
- Klik tombol **Grayscale** untuk memproses citra menjadi matrik coocurence
- Klik tab Nilai Matriks untuk menampilkan tabel matriks dari ekstraksi ciri kookurensi, hasil akan ditampilkan pada datagridview
- Klik select kualitas untuk memilih jenis kualitas beras
- Klik tombol simpan untuk menambahkan dataset

Ukuran 1090 x 591, warna disesuaikan

Gambar 0.41 Antarmuka Dataset

No F02	
	• Klik tombol **Load data** untuk menampilkan data • Klik tombol **Latih data** untuk melatih data
Ukuran 1090 x 591, warna disesuaikan	

Gambar 0.42 Antarmuka Pelatihan

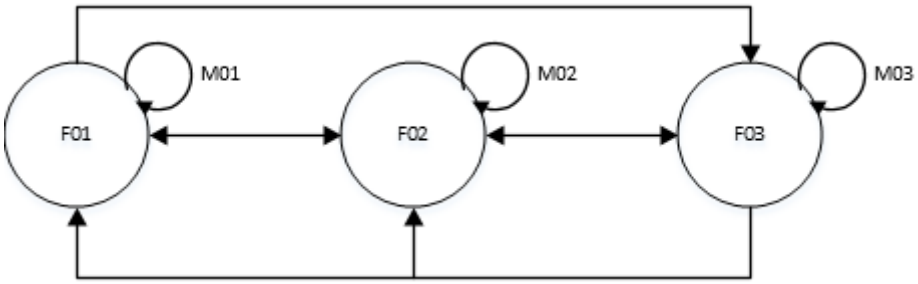
No F03	
	• Klik tombol **INPUT** untuk menampilkan gambar • Klik tombol **Grayscale** untuk mengubah citra ke dalam bentuk gryascalee • Klik tombol **Proses** untuk memproses seluruh data sampai dihasilkan hasil klasifikasi
Ukuran 1090 x 591, warna disesuaikan	

Gambar 0.43 Antarmuka Pengujian

5. Jaringan Semantik

Jaringan semantik merupakan gambaran pengetahuan grafis yang menunjukan hubungan antar berbagai objek. Jaringan semantik terdiri dari lingkaran-lingkaran

yang menunjukan objek dan informasi tentang objek-objek tertentu, untuk lebih jelasnya dapat dillihat pada gambar berikut ini:



Gambar 0.44 Jaringan Semantik