

## BAB III

### ANALISA DAN DESAIN SISTEM

#### A. Analisa Sistem

##### 1. Analisa Sistem Lama

K-means digunakan untuk melakukan segmentasi awal dengan mengelompokkan piksel-piksel menjadi beberapa kelompok berdasarkan warna. Setelah itu, metode Otsu digunakan untuk menentukan ambang batas yang optimal dari masing-masing kelompok warna yang dihasilkan oleh K-means, dari pengujian citra diperoleh dari nilai PSNR dan MSE. Oleh karena itu peneliti menggunakan metode K-means dan Otsu untuk mengetahui hasil segmentasi citra objek daun Bawang Merah, rumusan masalah yang dapat di simpulkan adalah.

- a. Bagaimana proses segmentasi citra menggunakan metode *K-means* dan *Otsu* pada citra daun bawang merah?
- b. Bagaimana mendapatkan hasil efisiensi segmentasi citra menggunakan nilai (*Mean Squared Error*) MSE dan (*Peak Signal to Noise Ratio*) PSNR pada metode *K-means* dan *Otsu*?

##### 2. Analisa Sistem Yang Diusulkan

###### a. Analisa Kebutuhan Fungsi

Sistem ini adalah solusi yang dikembangkan untuk segmentasi citra daun Bawang Merah dengan menggunakan metode K-means dan Otsu. Tujuan dari sistem ini adalah untuk memperoleh informasi tentang hasil segmentasi citra tersebut, termasuk nilai hasil perhitungan MSE (Mean Squared Error) dan PSNR (Peak Signal-to-Noise Ratio). Dengan menggunakan metode K-means, sistem ini dapat mengelompokkan piksel-piksel pada citra daun Bawang Merah ke dalam beberapa kelompok berdasarkan kemiripan warna. Sedangkan metode Otsu digunakan untuk menentukan ambang batas optimal dalam proses segmentasi. Dengan demikian, sistem ini memberikan kemampuan untuk menganalisis dan mengukur kualitas segmentasi citra daun Bawang Merah secara objektif melalui perhitungan MSE dan PSNR.

## b. Analisa Kebutuhan Data

### 1) Data *Input*

Untuk mendapat data Gambar dibutuhkan tabel alternatif dan kriteria seperti Format Gambar, Background, keterangan pengambilan gambar. Data yang di ambil menggunakan kamera hp. Adapun tabel kriteria sebagai berikut :

Tabel 3. 1Data Input Citra

NO	Format Gambar	Background	Keterangan
1.	JPG	Background Pasir	Pada penelitian ini, dilakukan segmentasi pada 5 data citra daun bawang merah yang merupakan data primer. Citra-citra tersebut diambil dengan kondisi pencahayaan dan latar belakang yang berbeda-beda. Citra memiliki format RGB, berukuran piksel 500 x 500, dan berformat JPG/JPEG.
2.	JPG	Background Pasir Gelap	Pada penelitian ini, dilakukan segmentasi pada 5 data citra daun bawang merah yang merupakan data primer. Citra-citra tersebut diambil dengan kondisi pencahayaan dan latar belakang yang berbeda-beda. Citra memiliki format RGB, berukuran piksel 500 x 500, dan berformat JPG/JPEG.

3.	JPG	Background Putih	Pada penelitian ini, dilakukan segmentasi pada 5 data citra daun bawang merah yang merupakan data primer. Citra-citra tersebut diambil dengan kondisi pencahayaan dan latar belakang yang berbeda-beda. Citra memiliki format RGB, berukuran piksel 500 x 500, dan berformat JPG/JPEG.
4.	JPG	Background Putih Gelap	Pada penelitian ini, dilakukan segmentasi pada 5 data citra daun bawang merah yang merupakan data primer. Citra-citra tersebut diambil dengan kondisi pencahayaan dan latar belakang yang berbeda-beda. Citra memiliki format RGB, berukuran piksel 500 x 500, dan berformat JPG/JPEG
5.	JPG	Random	Pada penelitian ini, dilakukan segmentasi pada 10 data citra daun bawang merah yang merupakan data primer. Citra-citra tersebut diambil dengan kondisi pencahayaan dan latar belakang yang berbeda-beda. Citra memiliki format RGB, berukuran piksel 500 x 500, dan berformat JPG/JPEG

## 2) Gambaran Proses

Proses segmentasi citra melibatkan beberapa langkah. Pertama, persiapkan data citra yang akan diolah. Selanjutnya, lakukan preprocessing pada data citra, termasuk konversi ke citra grayscale jika masih berwarna, dan aplikasikan filter untuk menghilangkan noise serta meningkatkan kualitas citra. Setelah itu, pilih metode segmentasi yang akan digunakan, antara K-Means atau Otsu. Selanjutnya, ekstraksi fitur citra dilakukan menggunakan metode tersebut, kemudian data citra dibagi menjadi beberapa cluster. Setelah mendapatkan hasil segmentasi, tentukan batas-batas cluster sebagai hasil segmentasi. Terakhir, lakukan penilaian terhadap hasil segmentasi menggunakan metrik yang sesuai, seperti MSE atau PSNR.

## 3) Data Output

Data output dari proses segmentasi citra menggunakan metode K-Means dan Otsu mencakup citra yang telah tersegmentasi menjadi beberapa cluster sesuai dengan metode yang dipilih, nilai penilaian hasil segmentasi menggunakan MSE dan PSNR, serta laporan hasil segmentasi yang berisi visualisasi citra yang telah tersegmentasi dan nilai penilaian hasil segmentasi. Hasil segmentasi citra akan menunjukkan bagaimana citra daun bawang merah terpisah dari latar belakangnya dan bagaimana bentuk bercak atau bintik teridentifikasi. Selain itu, nilai MSE dan PSNR akan memberikan informasi tentang tingkat akurasi dan kualitas segmentasi. Laporan hasil segmentasi akan menjadi gambaran menyeluruh tentang kinerja metode K-Means dan Otsu dalam melakukan segmentasi citra daun bawang merah, membantu dalam evaluasi dan pemahaman lebih lanjut mengenai efektivitas metode yang digunakan.

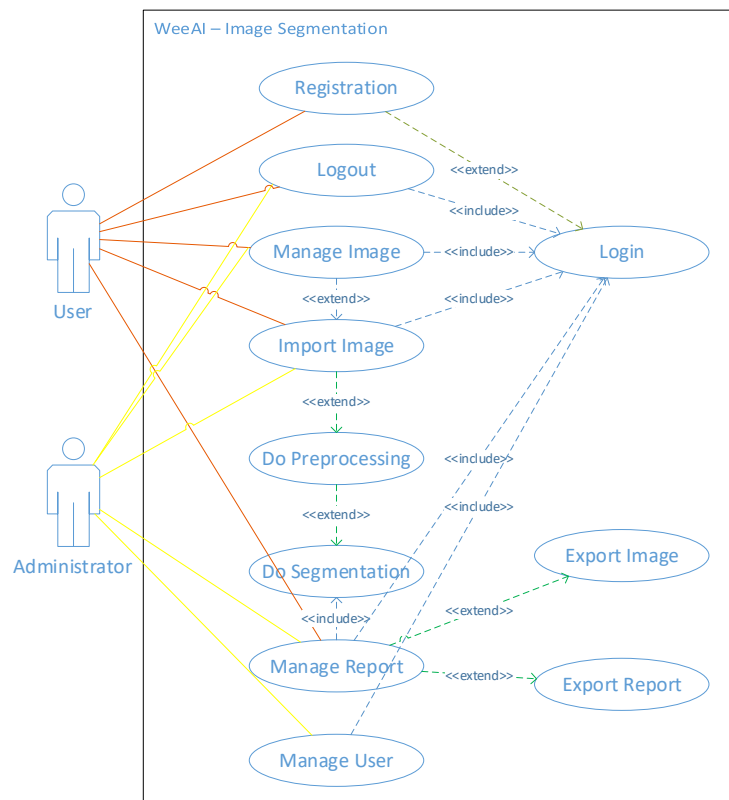
### 3. Analisa Kebutuhan Perangkat

Dalam penelitian ini, penulis memerlukan alat dan perangkat keras yang mendukung untuk pembuatan sistem. Salah satu perangkat keras yang digunakan adalah komputer dengan spesifikasi sebagai berikut :

- 1) Intel(R) Pentium(R) Silver N6000 @ 1.10GHz 1.11 GHz
- 2) Ram 4.00GB
- 3) Menggunakan perangkat lunak Operating System Windows 11 System type 64-bit operating system, x64-based processor, dan memerlukan software untuk mengerjakan sistem segmentasi citra menggunakan software visual studio code versi 1.80.1.

## B. Desain Sistem (Arsitektur)

### 1. Use Case Diagram



Gambar 3. 1 Use Case Diagram

Use Case diagram pada Gambar 3.1 menggambarkan interaksi antara pengguna dan administrator dengan sistem yang akan dibuat.

Pengguna adalah aktor eksternal yang menggunakan sistem melalui antarmuka pengguna untuk melakukan tugas seperti login, mengunggah file, dan melakukan proses analisis. Administrator adalah aktor yang memiliki akses khusus dan bertanggung jawab dalam mengelola sistem, termasuk mengatur pengaturan sistem, mengelola pengguna, dan melakukan tugas administratif lainnya. Tabel 3.2 dapat digunakan untuk menjelaskan secara lebih detail setiap tugas yang dilakukan oleh pengguna dan administrator dalam sistem tersebut:

Tabel 3. 2 Deskripsi Aktor

No.	Aktor	Deskripsi
1.	<i>User</i>	Pengguna biasa yang memiliki hak akses terbatas pada sistem
2.	<i>Administrator</i>	Pengguna yang memiliki hak akses lebih luas dan dapat mengelola sistem

Use Case diagram pada Gambar 3.1 menggambarkan interaksi antara pengguna dan administrator dengan sistem yang akan dibuat. Interaksi ini melibatkan beberapa use case yang dilakukan oleh pengguna dan administrator untuk memahami proses-proses dalam sistem dan interaksi antara aktor-aktor dengan sistem. Rincian lebih lanjut mengenai proses-proses tersebut dan interaksi antara aktor-aktor dengan sistem dapat ditemukan pada Tabel 3.3:

Tabel 3. 3 Deskripsi *Use Case*

No.	Use Case	Deskripsi
1	<i>Registration</i>	Proses pendaftaran akun baru bagi pengguna dan <i>administrator</i>
2	<i>Login</i>	Proses masuk ke sistem dengan menggunakan akun yang telah didaftarkan
3	<i>Logout</i>	Proses keluar dari sistem
4	<i>Manage Image</i>	Proses pengelolaan citra yang telah diupload ke sistem
5	<i>Import Image</i>	Proses mengimpor citra dari sumber luar ke sistem
6	<i>Do Preprocessing</i>	Proses persiapan citra sebelum dilakukan segmentasi
7	<i>Do Segmentation</i>	Proses pembagian citra menjadi beberapa bagian yang terpisah menurut perhitungan segmentasi
8	<i>Manage Report</i>	Proses pengelolaan laporan hasil segmentasi citra yang telah dilakukan
9	<i>ExportImage</i>	Proses meng- <i>export</i> hasil segmentasi citra ke dalam format gambar

10	<i>Export Report</i>	Proses meng- <i>export</i> laporan hasil segmentasi citra ke dalam format dokumen
11	<i>Manage User</i>	Proses pengelolaan informasi dan hak akses pengguna yang terdaftar dalam sistem

Dalam konteks ini, use case terkait dengan Login terhubung menggunakan metode include menunjukkan bahwa use case Login harus dilakukan terlebih dahulu sebelum use case lainnya dapat dilakukan. Sebagai contoh, sebelum pengguna dapat mengunggah file atau melakukan proses analisis, mereka harus terlebih dahulu melakukan Login ke dalam sistem. Use case Login menjadi bagian integral dari langkah-langkah yang harus dilakukan sebelum akses ke fitur-fitur lainnya dapat diberikan.

Di sisi lain, use case terkait dengan cara extend menunjukkan pengembangan dari use case yang lain. Sebagai contoh, use case Mengedit Data dapat menjadi ekstensi dari use case Mengunggah File. Dalam skenario ini, setelah pengguna mengunggah file ke dalam sistem, mereka diberikan opsi untuk memilih untuk mengedit file tersebut. Use case Mengedit Data menjadi opsional dan melengkapi use case Mengunggah File dengan memberikan fungsionalitas tambahan.

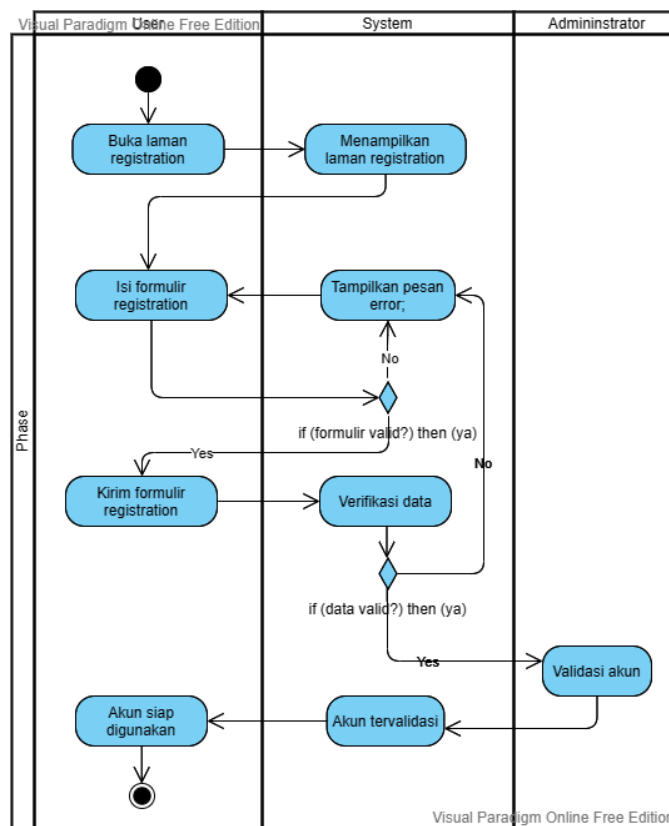
Dengan menggunakan metode include dan extend, use case-use case tersebut dapat saling terhubung dan membentuk alur kerja yang terstruktur dalam sistem, di mana use case Login menjadi langkah utama yang harus dilakukan sebelum menjalankan use case lainnya, dan use case Mengedit Data menjadi ekstensi opsional yang berkaitan dengan use case Mengunggah File.

## 2. Activity Diagram

*Activity Diagram* merupakan diagram yang menggambarkan segala aktivitas yang dilakukan oleh pengguna untuk *activity diagram* dari aplikasi yang di bangun ini adalah sebagai berikut:

### a. Activity Diagram Registration

*Activity Diagram Registration* adalah aktivitas semua pengguna aplikasi ketika akan mengakses aplikasi, berikut akan ditunjukkan oleh Gambar 3.2:



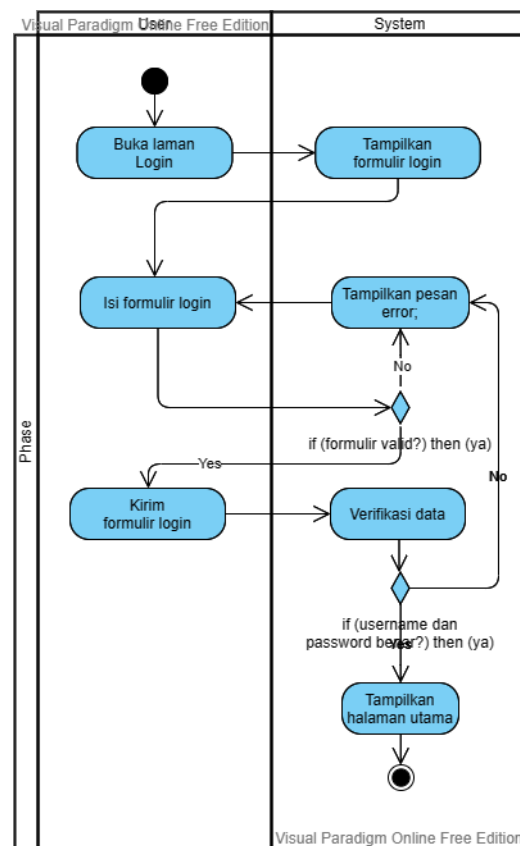
Gambar 3. 2 *Activity Diagram Registration*

Berdasarkan Diagram *Activity* pada Gambar 3.2 menggambarkan proses pendaftaran akun baru oleh pengguna. Sistem menampilkan formulir pendaftaran kepada pengguna. Pengguna mengisi dan mengirimkan formulir ke sistem. Sistem memverifikasi data yang dikirimkan oleh pengguna. Jika data valid, sistem membuat akun baru untuk pengguna dan mengirim konfirmasi validasi ke administrator. Jika data tidak valid, sistem menampilkan pesan error kepada pengguna.



## b. Activity Diagram Login

*ActivityDiagramLogin* adalah aktivitas semua pengguna aplikasi ketika akan mengakses aplikasi, berikut akan ditunjukkan oleh Gambar 3.3:

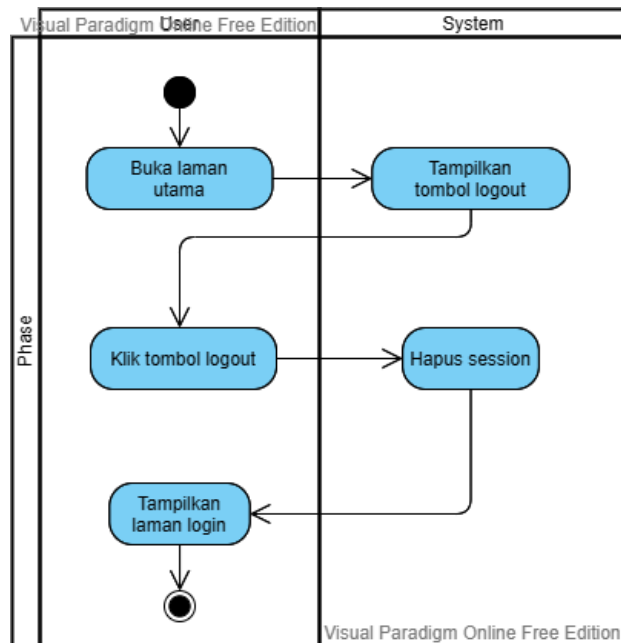


Gambar 3.3 *Activity Diagram Login*

Berdasarkan Diagram *activity* pada Gambar 3.3 ini menggambarkan proses login yang dilakukan oleh aktor. Sistem menampilkan formulir login kepada aktor. Aktor mengisi formulir dengan username dan password, dan mengirimkannya ke sistem. Sistem memverifikasi username dan password yang dikirimkan oleh aktor. Jika username dan password benar, sistem menampilkan halaman utama kepada aktor. Jika username atau password salah, sistem menampilkan pesan error kepada aktor.

### c. Activity Diagram Logout

*Activity Diagram Logout* adalah aktivitas semua pengguna aplikasi ketika akan mengakses aplikasi, berikut akan ditunjukkan oleh Gambar 3.4 :

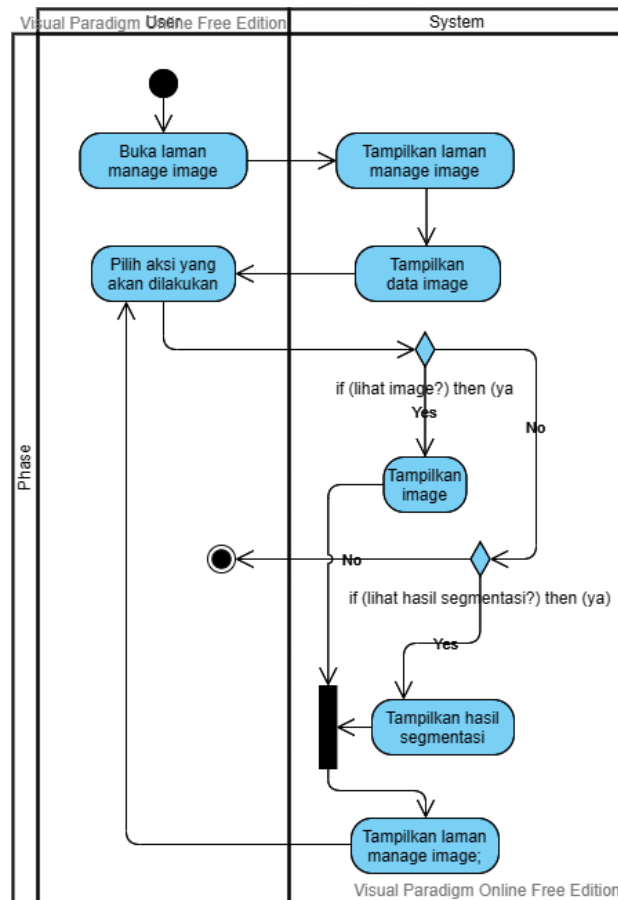


Gambar 3. 4Activity Diagram Logout

Berdasarkan DiagramActivity pada Gambar 3.4 diagram ini menggambarkan proses logout yang dilakukan oleh aktor. Sistem menampilkan tombol logout kepada aktor. Aktor menekan tombol logout. Sistem menghapus sesi yang terkait dengan akun aktor dan menampilkan halaman login. Proses logout selesai.

### d. Activity Diagram Manage Image

*Activity Diagram Manage Image* adalah aktivitas semua pengguna aplikasi ketika akan mengakses aplikasi, berikut akan ditunjukkan oleh Gambar 3.5:

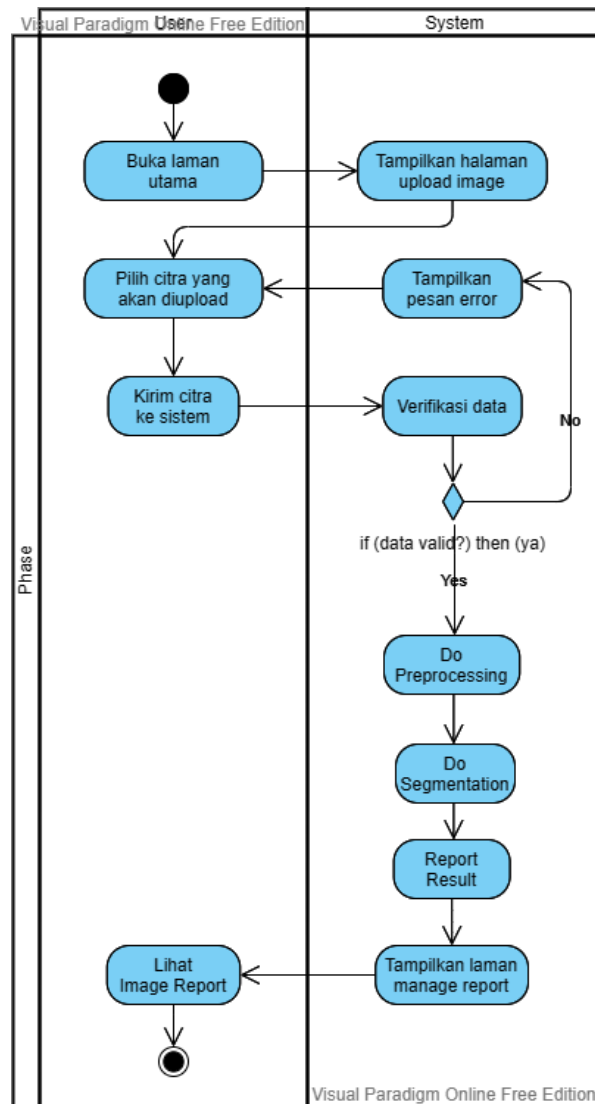


Gambar 3. 5 Activity Diagram Manage Image

Berdasarkan Diagram Activity pada Gambar 3.5 menggambarkan proses pengelolaan gambar oleh aktor. Halaman "Manage Image" ditampilkan. Data gambar ditampilkan. Aktor memilih aksi yang akan dilakukan: "View Image" (lihat gambar) atau "View Segmentation Result" (lihat hasil segmentasi). Jika aktor memilih "View Image", gambar ditampilkan. Jika aktor memilih "View Segmentation Result", hasil segmentasi ditampilkan. Setelah itu, halaman "Manage Image" ditampilkan kembali.

#### e. Activity Diagram Import Image

Activity Diagram Import Image adalah aktivitas semua pengguna aplikasi ketika akan mengakses aplikasi, berikut akan ditunjukkan oleh Gambar 3.6:

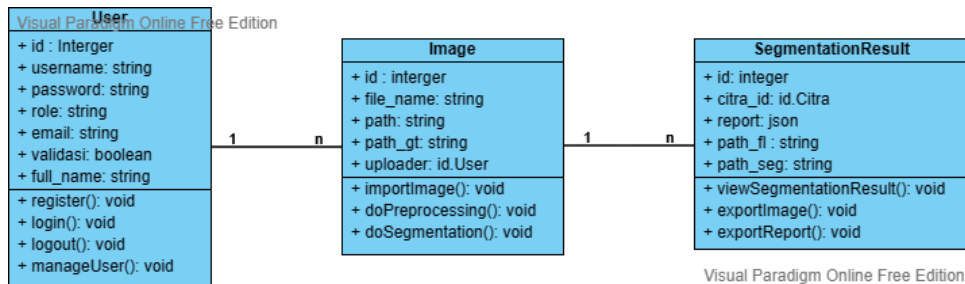


Gambar 3. 6Activity Diagram Import Image

Berdasarkan Diagram *Activity* pada Gambar 3.6 Diagram proses upload citra dimulai dengan sistem menampilkan halaman upload citra kepada aktor. Setelah memilih citra, aktor mengirimkannya ke sistem. Sistem melakukan verifikasi terhadap data citra dan jika valid, melalui proses preprocessing dan segmentasi, citra tersebut dimasukkan ke dalam sistem pengelolaan citra. Sistem kemudian menampilkan halaman manage report image kepada aktor. Jika data citra tidak valid, sistem menampilkan pesan error kepada aktor.

### C. Desain Database

Desain *database* merupakan struktur dan relasi yang terdapat antara tabel-tabel yang akan digunakan dalam sistem. Pada Gambar 3.9, terdapat tiga tabel yaitu tabel pengguna, tabel citra, dan tabel hasil segmentasi:



Gambar 3. 7Class Diagram Database

Class diagram pada Gambar 3.7, Berikut adalah struktur tabel dan relasi antar tabel dalam sistem yang akan dibuat:

1. Tabel User:

- Kolom: id, username, password, role, email, validasi, full\_name

2. Tabel Image:

- Kolom: id, filename, path, path\_gt, uploader

3. Tabel SegmentationResult:

- Kolom: id, citra\_id, hasil, path\_fl, path\_seg

4. Relasi antar tabel:

- Tabel User dan Tabel Image memiliki relasi one-to-many, dengan kolom uploader pada Tabel Image mengacu pada kolom id pada Tabel User.
- Tabel Image dan Tabel SegmentationResult memiliki relasi one-to-many, dengan kolom citra\_id pada Tabel SegmentationResult mengacu pada kolom id pada Tabel Image.

5. Aksi yang dapat dilakukan oleh pengguna (User):

- register(): Melakukan pendaftaran pengguna baru.
- login(): Melakukan proses login.
- logout(): Melakukan proses logout.
- manageUser(): Mengelola data pengguna seperti menambah, mengedit, atau memvalidasi pengguna.

6. Aksi yang dapat dilakukan oleh citra (*Image*):
  - `importImage()`: Mengimpor citra ke dalam sistem.
  - `doPreprocessing()`: Melakukan proses pra-pemrosesan pada citra.
  - `doSegmentation()`: Melakukan proses segmentasi pada citra.
7. Aksi yang dapat dilakukan oleh hasil segmentasi (*SegmentationResult*):
  - `viewSegmentationResult()`: Melihat hasil segmentasi.
  - `exportImage()`: Melakukan ekspor citra hasil segmentasi.
  - `exportReport()`: Melakukan ekspor laporan hasil segmentasi.

Dengan struktur tabel dan relasi yang disebutkan di atas, serta aksi yang dapat dilakukan oleh pengguna, citra, dan hasil segmentasi, sistem dapat mengelola data pengguna, citra, dan hasil segmentasi dengan baik.-  
`manageUser()`: Mengelola data pengguna seperti menambah, mengedit, atau memvalidasi pengguna.

Aksi yang dapat dilakukan oleh citra (*Image*):

- `importImage()`: Mengimpor citra ke dalam sistem.
- `doPreprocessing()`: Melakukan proses pra-pemrosesan pada citra.
- `doSegmentation()`: Melakukan proses segmentasi pada citra.

Aksi yang dapat dilakukan oleh hasil segmentasi (*SegmentationResult*):

- `viewSegmentationResult()`: Melihat hasil segmentasi.
- `exportImage()`: Melakukan ekspor citra hasil segmentasi.
- `exportReport()`: Melakukan ekspor laporan hasil segmentasi.

Dengan struktur tabel dan relasi yang disebutkan di atas, serta aksi yang dapat dilakukan oleh pengguna, citra, dan hasil segmentasi, sistem dapat mengelola data pengguna, citra, dan hasil segmentasi dengan baik.

Relasi antar tabel yang terdapat pada class diagram ini adalah sebagai berikut:

- a. *User* (id) memiliki banyak *Image*(uploader)

- b. *Image(id)* hanya dimiliki oleh satu *User (uploader)*
- c. *Image(id)* memiliki satu *SegmentationResult(citra\_id)*
- d. *SegmentationResult(id)* hanya dimiliki oleh satu *Image(citra\_id)*

#### D. Simulasi Algoritma



Gambar 3. 8 Data Citra Asli

Pada Gambar 3. 8 merupakan contoh sample data citra daun bawang yang selanjutnya akan dirubah menjadi bentuk matrix 3x3.

R:150	R:250	R:180
G:100	G:255	G:130
B:200	B:45	B:255
R:120	R:0	R:110
G:250	G:170	G:125
B:150	B:65	B:30
R:250	R:55	R:45
G:0	G:100	G:200
B:35	B:75	B:15

Gambar 3. 9 Contoh Matrix 3x3

##### 1. *Grayscale*

Data citra *RGB* diubah menjadi *Luminosity* greyscale terlebih dahulu menggunakan rumus pada persamaan.

$$\text{Grey} = (0.21 \times 150) + (0.71 \times 100) + (0.07 \times 200) = 116,5$$

$$\text{Grey} = (0.21 \times 120) + (0.71 \times 250) + (0.07 \times 150) = 213,2$$

$$\text{Grey} = (0.21 \times 250) + (0.71 \times 0) + (0.07 \times 35) = 54,95$$

$$\text{Grey} = (0.21 \times 250) + (0.71 \times 255) + (0.07 \times 45) = 236,7$$

$$\text{Grey} = (0.21 \times 0) + (0.71 \times 170) + (0.07 \times 65) = 125,25$$

$$\text{Grey} = (0.21 \times 55) + (0.71 \times 100) + (0.07 \times 75) = 87,8$$

$$\text{Grey} = (0.21 \times 180) + (0.71 \times 130) + (0.07 \times 255) = 147,95$$

$$\text{Grey} = (0.21 \times 110) + (0.71 \times 125) + (0.07 \times 30) = 113,95$$

$$\text{Grey} = (0.21 \times 45) + (0.71 \times 200) + (0.07 \times 15) = 152,5$$

Dari hasil perhitungan sebelumnya, maka didapat pixel 3x3 citra greyscale dan sudah dibulatkan seperti gambar.

116	236	148
213	125	114
55	89	152

Gambar 3.10 Hasil Grayscale

Pada gambar 3.10 merupakan hasil dari perhitungan data citra asli yang diubah menjadi greyscale.

## 2. Manualisasi *Pre-processing* Citra untuk *K-Means Clustering*

Pada proses manualisasi *pre-processing* citra, terdapat dua tahap yaitu *Rescaling* dan perubahan ruang warna RGB menjadi Lab\*. Perubahan ruang warna RGB melibatkan tahap pertama yaitu mengubah ruang warna RGB ke XYZ, kemudian dari XYZ mengubahnya menjadi ruang warna Lab\*. Tahap pertama dalam proses *pre-processing* adalah citra *input* yang akan diolah setiap bagian variabel ruang warna dipisahkan seperti yang ditunjukkan pada **Error! Reference source not found.**

Tabel 3. 4 Citra Input Proses Perubahan Ruang Warna

Red			Green			Blue		
64	108	129	81	96	186	142	84	104
205	78	151	82	84	208	121	203	140
113	234	213	85	173	194	233	151	98
96	79	186	151	104	238	92	229	94



### Manualisasi RGB to L\*a\*b\*

Proses perubahan ruang warna RGB menjadi Lab\* melibatkan beberapa tahap. Tahap pertama adalah mengubah ruang warna RGB menjadi XYZ dengan menggunakan transformasi matrix 3x3 dan nilai tristimulus sesuai dengan persamaan(1). Citra *input* yang akan diolah ditunjukkan pada. Setelah tahap ini selesai, citra akan siap untuk diubah ruang warnanya dari XYZ menjadi Lab\* melalui tahap selanjutnya.

Dalam konversi ruang warna RGB ke XYZ, transformasi matrix 3x3 yang digunakan adalah sebagai berikut:

$$[X] = (0,412453 \times R) + (0,357580 \times G) + (0,180423 \times B)$$

$$[Y] = (0,212671 \times R) + (0,715160 \times G) + (0,072169 \times B)$$

$$[Z] = (0,019334 \times R) + (0,119193 \times G) + (0,950227 \times B)$$

Dimana:

- 3) [X], [Y], dan [Z] adalah nilai tristimulus pada ruang warna XYZ
- 4) R, G, dan B adalah nilai tristimulus pada ruang warna RGB

Contoh perhitungan piksel (0,0) konversi ruang warna RGB ke XYZ, transformasi matrix 3x3 yang digunakan adalah sebagai berikut:

$$[X] = (0,412453 \times 64) + (0,357580 \times 81) + (0,180423 \times 142) = 133,4$$

$$[Y] = (0,212671 \times 64) + (0,715160 \times 81) + (0,072169 \times 142) = 105,6$$

$$[Z] = (0,019334 \times 64) + (0,119193 \times 81) + (0,950227 \times 142) = 217,3$$

Jadi, hasil konversi nilai tristimulus (R,G,B) = (64, 81, 142) adalah (X,Y,Z) = (133,4 , 105,6 , 217,3).sedangkan perhitungan pada seluruh piksel dapat dilihat melalu Tabel 3. 5.

Tabel 3. 5 Hasil Ruang Warna XYZ

X			Y			Z		
133,4	156,1	258,3	105,6	125,7	221,7	217,3	178	287,4
188,7	153,2	295,9	135	116	250,9	201,1	278,5	344,1
174	297,1	299,9	126,6	234,4	247,1	308,7	321,1	291,4
207,6	178,3	332	178,8	138	285,1	240,4	323,2	331

Setelah mengkonversi ruang warna RGB ke XYZ **Error! Reference source not found.**, selanjutnya dilakukan proses pembagian dengan *White Reference*-nya untuk mendapatkan nilai ruang warna Lab\*. Penyusun ruang warna Lab\* menggunakan persamaan(2), di mana setiap nilai tristimulus XYZ dibagi dengan *White Reference*-nya masing-masing. *White Reference* merupakan nilai maksimal dari setiap tristimulus pada ruang warna XYZ, yang ditandai dengan  $X_0$ ,  $Y_0$ , dan  $Z_0$ . Nilai  $X_0$  adalah 242,36628,  $Y_0$  adalah 255, dan  $Z_0$  adalah 277,63227, sehingga rentang nilai yang dihasilkan dari proses pembagian tersebut berkisar antara 0 sampai 1. Berikut merupakan contoh perhitungan piksel (0,0) pembagian ruang warna dengan *White Reference*-nya:

$$\frac{X}{X_0} = \frac{133,4}{242,3662} = 0,55$$

$$\frac{Y}{Y_0} = \frac{105,6}{255} = 0,41$$

$$\frac{Z}{Z_0} = \frac{217,3}{277,63227} = 0,78$$

Setelah proses pembagian dengan *White Reference* selesai, maka diperoleh nilai 0.55 untuk  $X/X_0$ , 0.41 untuk  $Y/Y_0$ , dan 0.78 untuk  $Z/Z_0$ . Seluruh hasil perhitungan tersebut dapat dilihat pada Tabel 3.6.

Tabel 3.6 Hasil Pembagian XYZ Dengan *White Reference*

$\frac{X}{X_0}$			$\frac{Y}{Y_0}$			$\frac{Z}{Z_0}$		
0,55	0,64	1,07	0,41	0,49	0,87	0,78	0,64	1,04
0,78	0,63	1,22	0,53	0,45	0,98	0,72	1	1,24
0,72	1,23	1,24	0,50	0,92	0,97	1,11	1,16	1,05
0,86	0,74	1,37	0,70	0,54	1,12	0,87	1,16	1,19

Pada tahap selanjutnya, dilakukan perhitungan fungsi untuk menjadi komponen penyusun dari persamaan(3) yaitu dengan simbol  $f(X/X_0)$ ,  $f(Y/Y_0)$ , dan  $f(Z/Z_0)$ . Persamaan tersebut memiliki dua kondisi, yaitu jika hasil pembagian XYZ dengan *White Reference*-nya lebih dari

0.008856 dan jika hasil pembagian XYZ dengan *White Reference*-nya memiliki rentan nilai antara 0 hingga 0.008856. Berikut ini merupakan contoh perhitungan fungsi pada piksel pertama untuk proses perubahan ruang warna XYZ menjadi Lab\*:

$$f(w) = w^{\frac{1}{3}} \quad \text{for } w > 0.008856$$

$$f(w) = 7.787(w) + 0.1379 \quad \text{for } 0.0 \leq w \leq 0.008856$$

Sebagai contoh, jika ingin mengkonversi nilai tristimulus (X,Y,Z) = (0.74, 0.61, 0.65) dari ruang warna XYZ ke Lab\*, maka cara yang dilakukan adalah sebagai berikut:

$$\frac{X}{X_0} = 0,55 \text{ (masuk kondisi pertama)}$$

$$f\left(\frac{X}{X_0}\right) = \left(\frac{X}{X_0}\right)^{\frac{1}{3}} = 0,55^{\frac{1}{3}} = 0,81$$

$$\frac{Y}{Y_0} = 0,41 \text{ (masuk kondisi pertama)}$$

$$f\left(\frac{Y}{Y_0}\right) = \left(\frac{Y}{Y_0}\right)^{\frac{1}{3}} = 0,41^{\frac{1}{3}} = 0,75$$

$$\frac{Z}{Z_0} = 0,78 \text{ (masuk kondisi kedua)}$$

$$f\left(\frac{Z}{Z_0}\right) = \left(\frac{Z}{Z_0}\right)^{\frac{1}{3}} = 0,78^{\frac{1}{3}} = 0,92$$

Hasil perhitungan fungsi menghasilkan nilai 0.979 untuk  $f(X/X_0)$ , 0.941 untuk  $f(Y/Y_0)$ , dan 0.93 untuk  $f(Z/Z_0)$ . Seluruh perhitungan fungsi dapat dilihat pada Tabel 3.7.

Tabel 3.7 Perhitungan Fungsi Terhadap XYZ

$f\left(\frac{X}{X_0}\right)$	$f\left(\frac{Y}{Y_0}\right)$	$f\left(\frac{Z}{Z_0}\right)$
-------------------------------	-------------------------------	-------------------------------

0,82	0,86	1,02	0,75	0,79	0,95	0,92	0,86	1,01
0,92	0,86	1,07	0,81	0,77	0,99	0,9	1	1,07
0,9	1,07	1,07	0,79	0,97	0,99	1,04	1,05	1,02
0,95	0,9	1,11	0,89	0,82	1,04	0,95	1,05	1,06

Pada tahap akhir perhitungan untuk mengubah ruang warna XYZ menjadi Lab\*, terdapat dua kondisi yang harus diperhatikan sesuai dengan persamaan(4).Pertama, ketika nilai  $Y/Y_0$  lebih besar dari 0.008856. Kedua, ketika nilai  $Y/Y_0$  berada pada rentang 0 hingga 0.008856. Berikut contoh perhitungan konversi ruang warna Lab\* untuk piksel (0,0):

$$L^* = 116 \left( \frac{Y}{Y_0} \right)^{\frac{1}{3}} - 16 \quad \text{for } \frac{Y}{Y_0} > 0,008856$$

$$L^* = 903,3 \frac{Y}{Y_0} \quad \text{for } 0,0 \leq \frac{Y}{Y_0} \leq 0,008856$$

$$\frac{Y}{Y_0} = 0,82 \text{ (masuk kondisipertama)}$$

$$L^* = 116 \left( \frac{Y}{Y_0} \right)^{\frac{1}{3}} - 16 =$$

$$116 (0,82)^{\frac{1}{3}} - 16 = 70,46$$

$$a^* = 500 \left( f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right) \right)$$

$$= 500(0,81 - 0,75) = 37,08$$

$$b^* = 200 \left( f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right) \right)$$

$$= 200(0,75 - 0,92) = 35,25$$

Pada tahap akhir dari proses *pre-processing*, citra asli yang telah di-*rescaling* dan diubah ke ruang warna Lab\* akan disimpan dalam bentuk tabel matrix seperti yang terlihat pada Tabel 3.8. Kemudian, citra tersebut akan digunakan sebagai *input* untuk proses seleksi fitur dan proses

pembelajaran dengan menggunakan algoritma *K-Means Clustering* secara manual.

Tabel 3.8 Hasil Ruang Warna L\*a\*b\*

L*			a*			b*		
70,46	75,6	94,71	37,08	36,74	33,56	35,25	14,46	11,45
77,85	73,2	99,38	55,48	44,57	37,06	17,79	46,42	15,9
75,84	96,78	98,79	51,84	48,98	42,01	48,85	15,49	5,33
87,05	78,54	104,39	30,64	43,83	36,4	12,96	47,4	4,51

### 3. Manualisasi Segmentasi Citra dengan *K-Means Clustering*

Untuk melakukan segmentasi citra menggunakan algoritma *K-Means Clustering*, pertama-tama perlu dilakukan pemilihan nilai *centroid* awal. Pemilihan *centroid* awal ini bertujuan untuk menentukan garis pemisah pada proses seleksi data menggunakan garis *ellipse* yang mengelilingi setiap *centroid*. Pada tahap ini, *centroid* awal yang dipilih menggunakan nilai a\* dan b\* terbanyak dari setiap piksel. Kemudian, dilakukan proses penentuan garis *ellipse* dengan menghitung jarak terdekat dari setiap piksel ke *centroid* yang dipilih. Setelah itu, dilakukan proses pemilihan *centroid* baru dengan mengambil rata-rata a\* dan b\* dari setiap piksel yang tergabung dalam garis *ellipse* tersebut. Proses ini terus dilakukan hingga tidak ada lagi perubahan *centroid*. Pada penelitian ini, proses segmentasi citra menggunakan algoritma *K-Means Clustering* dilakukan secara manual dengan menentukan nilai *centroid* awal sesuai dengan data yang terdapat pada Tabel 3.9.

Tabel 3.9 Data Input *K-Means Clustering*

a*			b*		
37,08	36,74	33,56	35,25	14,46	11,45
55,48	44,57	37,06	17,79	46,42	15,9
51,84	48,98	42,01	48,85	15,49	5,33
30,64	43,83	36,4	12,96	47,4	4,51

### Manualisasi Segmentasi Daun Bawang

Segmentasi daun dilakukan dengan menggunakan algoritma *K-Means Clustering* dengan jumlah *cluster* yang telah ditentukan sebanyak

2. Data yang digunakan pada Tabel 3.9 diolah dengan menggabungkan setiap bagiannya dan memberikan *index* pada setiap piksel sehingga menghasilkan Tabel baru seperti yang ditunjukkan pada Tabel 3.10.

Tabel 3.10 Data *Input* Segmentasi Daun

<b>Index</b>	<b>a*</b>	<b>b*</b>
(0,0)	37,08	35,25
(0,1)	36,74	14,46
(0,2)	33,56	11,45
(1,0)	55,48	17,79
(1,1)	44,57	46,42
(1,2)	37,06	15,90
(2,0)	51,84	48,85
(2,1)	48,98	15,49
(2,2)	42,01	5,33
(3,0)	30,64	12,96
(3,1)	43,83	47,40
(3,2)	36,40	4,51

Untuk memulai proses segmentasi daun menggunakan algoritma *K-Means Clustering*, langkah pertama yang dilakukan adalah menentukan nilai *centroid* awal sebanyak jumlah *cluster* yang telah ditentukan sebanyak 2. Nilai *centroid* awal tersebut ditentukan secara acak, seperti yang terlihat pada Tabel 3.11. Algoritma *K-Means Clustering* akan menggunakan nilai *centroid* tersebut sebagai dasar untuk mengelompokkan data pada setiap iterasinya.

Tabel 3.11 *Centroid* Awal Segmentasi Daun

<b>Centroid</b>	<b>a*</b>	<b>b*</b>	<b>Index</b>
c1	37,08	35,25	(0,0)
c2	37,06	15,9	(1,2)

Pada Tabel 3.11, *centroid* awal yang terpilih secara acak adalah piksel (0,0) dan (1,2). Kemudian proses selanjutnya adalah menentukan titik-titik yang tergabung dengan setiap *centroid*. Titik-titik yang tergabung ditentukan dengan menghitung jarak antara titik tersebut dengan setiap *centroid*. Jarak antara titik tersebut dengan *centroid* dapat dihitung

menggunakan rumus *Euclidean Distance* dengan persamaan(5). Berikut contoh perhitungan jarak antara titik (0,0) dengan setiap *Centroid*:

$$d_{c_1} = \sqrt{(c_1a^* - c_{c_1}a^*)^2 + (c_1b^* - c_{c_1}b^*)^2}$$

$$d_{c_2} = \sqrt{(c_1a^* - c_{c_2}a^*)^2 + (c_1b^* - c_{c_2}b^*)^2}$$

$$d_{c_1} = \sqrt{(37,08 - 37,08)^2 + (35,25 - 35,25)^2} = 0$$

$$d_{c_2} = \sqrt{(37,08 - 37,06)^2 + (35,25 - 15,9)^2} = 19,35$$

Pada iterasi ke 0, hasil perhitungan jarak antara setiap titik dengan *centroid* c1 dan c2 ditunjukkan pada Tabel 3.12. Berdasarkan hasil perhitungan jarak antara titik (0,0) dengan kedua *centroid* tersebut, maka titik (0,0) diputuskan untuk masuk kategori c1. Hasil perhitungan jarak untuk titik lainnya juga dilakukan secara serupa untuk menentukan kategori *centroid* yang sesuai bagi setiap titik.

Tabel 3.12 Jarak *Cluster* Segmentasi Daun Iterasi ke 0

Index	a*	b*	distance c1	distance c2	cluster
(0,0)	37,08	35,25	0	19,35	C1
(0,1)	36,74	14,46	20,8	1,48	C2
(0,2)	33,56	11,45	24,05	5,66	C2
(1,0)	55,48	17,79	25,37	18,52	C2
(1,1)	44,57	46,42	13,45	31,43	C1
(1,2)	37,06	15,90	19,35	0	C2
(2,0)	51,84	48,85	20,07	36,11	C1
(2,1)	48,98	15,49	23,07	11,93	C2
(2,2)	42,01	5,33	30,32	11,66	C2
(3,0)	30,64	12,96	23,21	7,07	C2
(3,1)	43,83	47,40	13,9	32,23	C1
(3,2)	36,40	4,51	30,74	11,4	C2

Pada tahap selanjutnya dari proses segmentasi daun menggunakan algoritma *K-Means Clustering*, dilakukan pembaharuan *centroid* dengan menghitung rata-rata dari tiap variabel a\* dan b\* pada setiap *cluster* yang sama. Berdasarkan Tabel 3.12, terdapat jumlah *cluster* 1 sebanyak 4 dan jumlah *cluster* 2 sebanyak 8. Nilai *centroid* baru dapat dihitung dengan

menggunakan rumus rata-rata yaitu menjumlahkan nilai  $a^*$  dan  $b^*$  pada setiap *cluster* yang sama kemudian dibagi dengan jumlah total piksel pada *cluster* tersebut. Berikut contoh perhitungan *centroid* baru pada iterasi ke 0:

Untuk mengetahui nilai  $a_{c1}$  dan  $b_{c1}$ , digunakan rumus persamaan (6):

$$a_{c1} = \left( \frac{(\sum a \text{ pada cluster 1})}{\text{jumlah data pada cluster 1}} \right)$$

$$= \left( \frac{(37,08 + 44,57 + 51,84 + 43,83)}{4} \right)$$

$$a_{c1} = 11,86$$

$$b_{c1} = \left( \frac{(\sum b \text{ pada cluster 1})}{\text{jumlah data pada cluster 1}} \right)$$

$$= \left( \frac{(35,25 + 46,42 + 48,85 + 47,4)}{4} \right)$$

$$b_{c1} = 29,78$$

Untuk mengetahui nilai  $a_{c2}$  dan  $b_{c2}$ , digunakan rumus persamaan:

$$a_{c2} = \left( \frac{(\sum a \text{ pada cluster 2})}{\text{jumlah data pada cluster 2}} \right)$$

$$a_{c2} = \left( \frac{36,78 + 33,56 + 55,48 + 37,1 + 48,98 + 42,01 + 30,64 + 36,4}{8} \right)$$

$$a_{c2} = 24,42$$

$$b_{c2} = \left( \frac{(\sum b \text{ pada cluster 2})}{\text{jumlah data pada cluster 2}} \right)$$

$$b_{c2} = \left( \frac{14,46 + 11,45 + 17,8 + 15,9 + 15,5 + 5,33 + 12,95 + 4,51}{8} \right)$$

Berdasarkan hasil perhitungan tersebut, didapatkan nilai *centroid* baru untuk iterasi ke 0 pada Tabel 3.13:



Tabel 3.13 *Centroid* Baru Iterasi ke 1 Segmentasi Daun

Centroid	a*	b*
c1	11,86	29,78
c2	24,61	8,464

Untuk iterasi ke 1, *centroid* yang digunakan ditunjukkan pada Tabel 3.13. Proses perhitungan jarak antara setiap data dengan *centroid* dilakukan sesuai dengan yang telah dijelaskan pada iterasi sebelumnya. Hasil perhitungan jarak pada iterasi ke 1 dapat dilihat pada Tabel 3.14.

Tabel 3.14 Jarak *Cluster* Segmentasi Daun Iterasi ke 1

Index	a*	b*	distance c1	distance c2	cluster
(0,0)	37,08	35,25	25,81	29,54	C1
(0,1)	36,74	14,46	29,22	13,52	C2
(0,2)	33,56	11,45	28,41	9,44	C2
(1,0)	55,48	17,79	45,24	32,25	C2
(1,1)	44,57	46,42	36,71	42,88	C1
(1,2)	37,06	15,9	28,78	14,5	C2
(2,0)	51,84	48,85	44,3	48,7	C1
(2,1)	48,98	15,49	39,78	25,36	C2
(2,2)	42,01	5,33	38,82	17,68	C2
(3,0)	30,64	12,96	25,21	7,51	C2
(3,1)	43,83	47,4	36,51	43,42	C1
(3,2)	36,4	4,51	35,22	12,43	C2

Pada tahap ini, algoritma *K-Means Clustering* telah mencapai konvergensi. Artinya, tidak ada kelas *cluster* yang berpindah pada iterasi ke 1 yang ditunjukkan pada Tabel 3.14. Hasil akhir dari Algoritma *K-Means Clustering* dapat dilihat pada yang menyatakan jumlah data pada setiap kelas *cluster* Tabel 3.15.

Tabel 3.15 Hasil K-Means Pada Segmentasi Daun

Index	L*	a*	b*	cluster
(0,0)	70,46	37,08	35,25	C1
(0,1)	75,65	36,74	14,46	C2
(0,2)	94,71	33,56	11,45	C2
(1,0)	77,85	55,48	17,79	C2
(1,1)	73,20	44,57	46,42	C1
(1,2)	99,38	37,06	15,90	C2
(2,0)	75,84	51,84	48,85	C1
(2,1)	96,78	48,98	15,49	C2

(2,2)	98,79	42,01	5,33	C2
(3,0)	87,05	30,64	12,96	C2
(3,1)	78,54	43,83	47,40	C1
(3,2)	104,39	36,40	4,51	C2

### 3. Otsu

Setelah konversi ke Grayscale (Hasil Nilai Greyscale pixel 3x3), langkah selanjutnya adalah menghitung histogram citra dengan mengurutkan piksel yang telah dikonversi ke skala abu-abu dari yang terkecil hingga yang terbesar, dan mengubah intensitas piksel menjadi level 0 hingga 8 (dalam jumlah piksel). Setelah itu, akan dilakukan perhitungan bobot nilai dari kedua kelas. Weight Background dapat dihitung menggunakan persamaan berikut ini:

$$W_b = \frac{1 + 1 + 1 + 1 + 1}{9} = 0,555$$

Selanjutnya mencari Mean Background menggunakan persamaan (9) sebagai berikut:

$$\mu_b = \frac{(0 \times 1) + (1 \times 1) + (2 \times 1) + (3 \times 1) + (4 \times 1)}{5} = 2$$

Setelah mendapatkan hasil Mean kemudian mencari nilai Variance menggunakan persamaan (11) sebagai berikut:

$$\begin{aligned} \sigma_b^2 &= \frac{((0-2)^2 \times 1) + ((1-2)^2 \times 1) + ((2-2)^2 \times 1) + ((3-2)^2 \times 1) + ((4-2)^2 \times 1)}{5} \\ &= 1 \end{aligned}$$

Kemudian mencari Weight Foreground menggunakan persamaan (8) sebagai berikut:

$$W_f = \frac{1 + 1 + 1 + 1}{9} = 0,444$$

Selanjutnya mencari Mean Foreground menggunakan persamaan (10) sebagai berikut:

$$\mu_f = \frac{(5 \times 1) + (6 \times 1) + (7 \times 1) + (8 \times 1)}{4} = 6,5$$

Setelah mendapatkan hasil Mean kemudian mencari nilai Variance menggunakan persamaan (12) sebagai berikut:

$$\sigma_f^2 = \frac{((5 - 6,5)^2 \times 1) + ((6 - 6,5)^2 \times 1) + ((7 - 6,5)^2 \times 1) + ((8 - 6,5)^2 \times 1)}{4} = 5$$

Langkah selanjutnya yaitu mencari nilai Betwen *Class* Variance menggunakan persamaan (13) sebagai berikut :

$$\sigma_w^2 = (0,555 \times 2) + (0,444 \times 6,5) = 3,996$$

Untuk mencari nilai *Within Class* terendah harus menghitung sesuai jumlah pixelnya. Berhubungan data yang digunakan hanya sample dan sebagai simulasi maka tidak dihitung secara merata. Namun dapat dicontohkan pada gambar.

4	255	10
200	90	77
45	75	100

Gambar 3.11 Hasil Perhitungan Thresholding Otsu

Gambar menunjukkan contoh hasil perhitungan *Thresholding Otsu*. Karena menggunakan data sampel, tidak perlu menghitung secara menyeluruh.

#### 4. MSE dan PSNR

Untuk menghitung *MSE* dan *PSNR*, langkah-langkah berikut perlu diikuti. Pertama, hitung selisih kuadrat antara setiap piksel dalam dua citra yang akan dibandingkan. Kemudian, hitung rata-rata dari selisih kuadrat tersebut untuk mendapatkan nilai *MSE*. Selanjutnya, hitung *PSNR* dengan menggunakan rumus logaritma dan skala logaritmik terhadap nilai *MSE*.

*K-Means*

$$MSE = \frac{(116 - 120)^2 + (236 - 230)^2 + (148 - 150)^2 + (213 - 210)^2 + (125 - 120)^2 + (114 - 110)^2 + (55 - 60)^2 + (89 - 90)^2 + (152 - 150)^2}{9}$$

$$MSE = \frac{-16 + 36 + -4 + 9 + 25 + 16 + -25 + 1 + 4}{9} = 8.33$$

$$PSNR = 10 \log_{10} \left( \frac{255}{\sqrt{7800}} \right) = 28,79$$

*Thresholding Otsu*

$$MSE = \frac{(116 - 4)^2 + (236 - 255)^2 + (148 - 10)^2 + (213 - 200)^2 + (125 - 90)^2 + (114 - 77)^2 + (55 - 45)^2 + (89 - 75)^2 + (152 - 100)^2}{9}$$

$$MSE = \frac{12.544 + 361 + 19.044 + 169 + 1.225 + 1.369 + 100 + 196 + 2.704}{9} = 4.190$$

$$PSNR = 10 \log_{10} \left( \frac{236}{\sqrt{4.190}} \right) = 15,61$$

Berdasarkan hasil simulasi perhitungan MSE dan PSNR, dapat disimpulkan bahwa metode Otsu memberikan hasil segmentasi yang lebih baik dibandingkan metode K-Means. Nilai MSE pada metode Otsu lebih kecil, menunjukkan bahwa hasil segmentasi dengan metode Otsu mendekati citra asli dengan lebih baik daripada metode K-Means. Selain itu, nilai PSNR pada metode Otsu lebih besar, menandakan bahwa kualitas hasil segmentasi dengan metode Otsu lebih tinggi

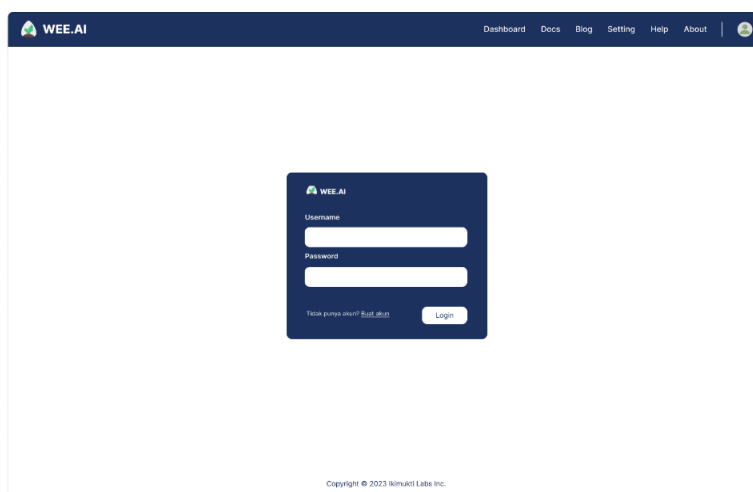
daripada metode K-Means dalam mempertahankan informasi citra. Oleh karena itu, metode Otsu dapat dianggap lebih efektif dan akurat dalam proses segmentasi citra dibandingkan dengan metode K-Means..

## E. Desain Antarmuka

Desain antarmuka menggambarkan tampilan sistem yang dilihat oleh pengguna. Berikut adalah contoh desain menu/aplikasi yang dapat dibuat.

### a. Halaman *Login*

Halaman Login menyediakan form untuk pengguna memasukkan username dan password guna masuk ke sistem. Berikut adalah rancangan halaman *Login* yang ditunjukkan pada Gambar 3.12.



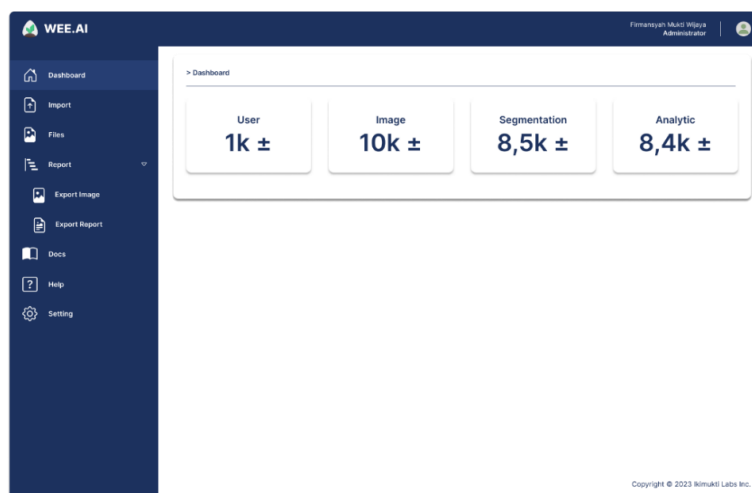
Gambar 3. 12Desain InterfaceLogin

### b. Halaman *Dashboard*

Pada rancangan halaman dashboard yang ditunjukkan pada Gambar 3.13, terdapat sebuah dashboard yang dirancang dengan tujuan memberikan akses cepat dan mudah ke berbagai fitur yang tersedia. Pengguna dapat dengan mudah melakukan beberapa tindakan penting melalui dashboard ini. Salah satu fitur yang dapat diakses dengan cepat adalah "upload image". Fitur ini memungkinkan pengguna untuk mengunggah gambar atau foto ke sistem. Hal ini sangat bermanfaat untuk

memproses gambar tersebut dan melakukan berbagai tugas seperti analisis atau segmentasi.

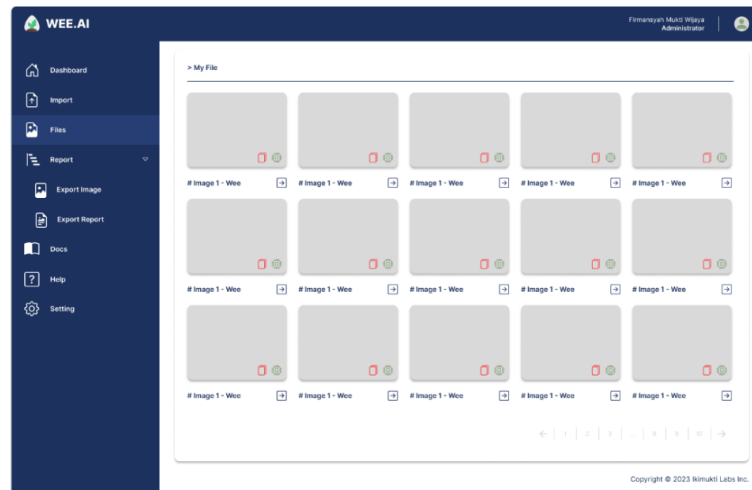
Selanjutnya, pengguna juga dapat menggunakan fitur "lihat hasil segmentasi" yang tersedia di dashboard. Fitur ini memungkinkan pengguna untuk melihat hasil segmentasi dari gambar yang telah diunggah sebelumnya. Segmentasi adalah proses pemisahan atau pengelompokan elemen dalam gambar menjadi bagian-bagian yang berbeda. Dengan fitur ini, pengguna dapat dengan mudah melihat hasil segmentasi yang telah dilakukan. Selain itu, dashboard ini juga menawarkan berbagai fitur lainnya yang mungkin bervariasi tergantung pada konteks dan kebutuhan aplikasi. Fitur-fitur ini dapat meliputi analisis data, pengaturan preferensi, pengelolaan konten, dan banyak lagi. Tujuannya adalah memberikan pengalaman pengguna yang komprehensif dan memenuhi kebutuhan pengguna dalam satu tampilan yang mudah diakses. Gambar 3.11 menunjukkan rancangan halaman dashboard yang menggambarkan tata letak dan penempatan elemen-elemen tersebut. Dengan tampilan yang jelas dan mudah dipahami, pengguna dapat dengan cepat mengakses dan menggunakan berbagai fitur yang tersedia, meningkatkan efisiensi dan produktivitas dalam penggunaan sistem.



Gambar 3. 13Desain InterfaceDashboard

c. Halaman *Files*

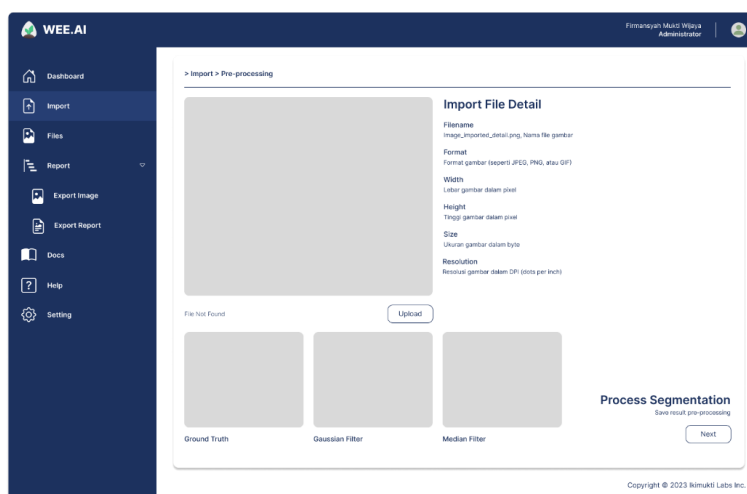
Halaman Manage Files menampilkan daftar gambar yang telah diunggah oleh pengguna dan memungkinkan pengguna untuk mengelola dan melihat hasil gambar yang telah dimasukkan. Berikut adalah rancangan halaman *Files* yang ditunjukkan pada Gambar 3.14.



Gambar 3. 14Desain InterfaceFiles

d. Halaman *upload image*

Halaman Upload Image menampilkan form untuk mengunggah gambar yang akan dilakukan proses segmentasi. Berikut adalah rancangan halaman *upload image* yang ditunjukkan pada Gambar 3.15.

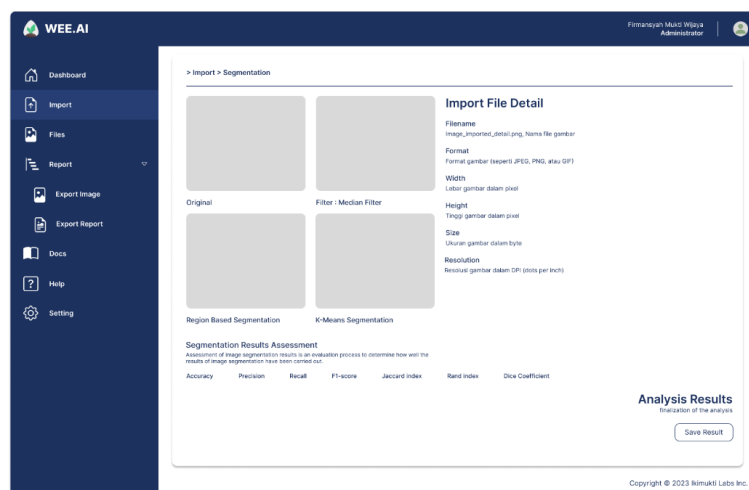


Gambar 3. 15Desain Upload Image

e. Halaman hasil segmentasi

Halaman Hasil Segmentasi adalah halaman yang menampilkan hasil segmentasi gambar yang diunggah oleh pengguna, beserta informasi terkait. Pada halaman ini, pengguna dapat melihat gambar yang telah dipisahkan berdasarkan objek dan latar belakang. Setelah pengguna mengunggah gambar pada Rancangan Halaman Upload Image (Gambar 3.16), sistem akan melakukan proses segmentasi menggunakan metode K-means atau Otsu. Metode yang digunakan bergantung pada implementasi sistem yang dapat dipilih. Hasil segmentasi kemudian akan ditampilkan pada Halaman Hasil Segmentasi. Pengguna dapat melihat citra yang telah dipisahkan, dengan objek dan latar belakang yang terpisah secara visual.

Dengan adanya Halaman Hasil Segmentasi, pengguna dapat dengan mudah melihat hasil dari proses segmentasi gambar yang dilakukan oleh sistem. Informasi tambahan yang ditampilkan juga dapat membantu pengguna dalam mengevaluasi kualitas segmentasi dan memahami lebih lanjut tentang proses yang dilakukan oleh sistem.

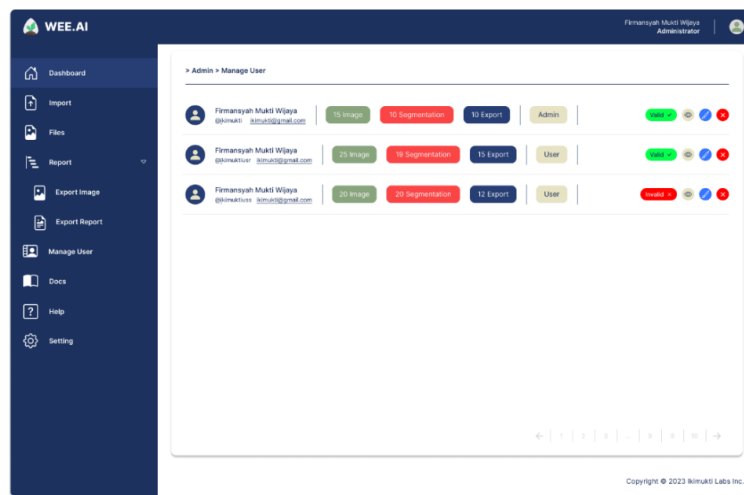


Gambar 3. 16Desain InterfaceResultSegmentation



f. Halaman *Manage User*

Halaman Manage User menampilkan daftar user yang terdaftar dan memberikan administrator kemampuan untuk mengelola user. Halaman Upload Image menampilkan form untuk mengunggah gambar yang akan di-segmentasi yang ditunjukkan pada Gambar 3.17.



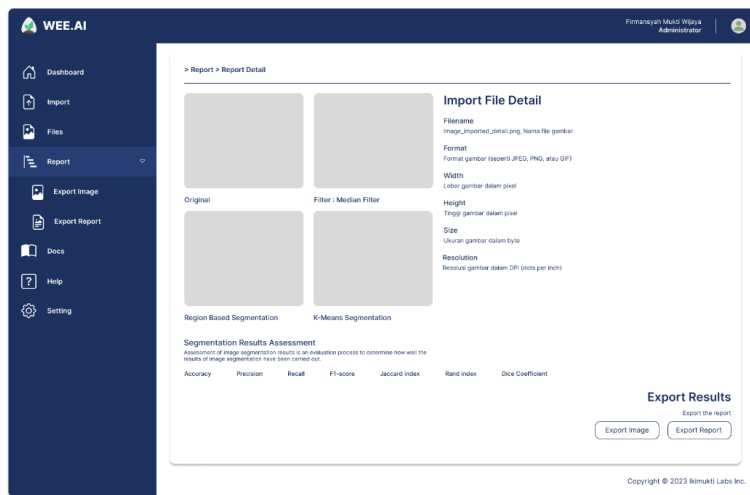
Gambar 3. 17Desain InterfaceManage User

g. Halaman *Export*

Sistem ini memiliki beberapa fitur yang sangat berguna bagi pengguna. Salah satunya adalah halaman Export yang telah disediakan, di mana pengguna dapat memanfaatkannya untuk mengexport laporan dan gambar hasil segmentasi citra per gambar. Dengan adanya opsi ini, pengguna dapat dengan mudah menyimpan dan berbagi hasil segmentasi dengan pihak lain.

Selain halaman Export, sistem ini juga dilengkapi dengan halaman Upload Image yang sangat praktis. Melalui halaman ini, pengguna diberikan kemampuan untuk mengunggah gambar yang nantinya akan menjalani proses segmentasi. Dengan adanya fitur ini, pengguna dapat dengan cepat memasukkan citra yang akan dianalisis dan diolah oleh sistem.

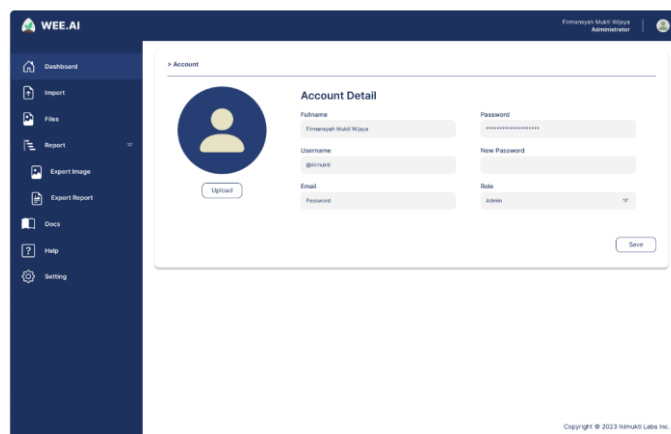
Untuk memvisualisasikan rancangan halaman Upload Image tersebut, Gambar 3.18 telah disediakan. Dalam gambar tersebut, pengguna dapat melihat tampilan antarmuka yang intuitif dan user-friendly yang memudahkan pengguna untuk mengunggah gambar dengan mudah dan lancar.



Gambar 3. 18Desain InterfaceExport

#### h. Halaman pengaturan akun

Halaman Edit Account menampilkan form untuk mengubah informasi akun pengguna, seperti nama, email, dan password. Berikut adalah rancangan halaman *upload image* yang ditunjukkan pada Gambar 3.19.



Gambar 3. 19Desain Interface Pengaturan Akun