

BAB IV. ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan langkah-langkah analisis dan perancangan penelitian. Dengan membuat kerangka konsep penelitian dan menggunakan tahapan sesuai dengan metodologi penelitian.

4.1 Analisis

Bagi peternak ikan lele yang ingin menghitung bibit ikan lele sering memiliki kendala dalam menentukan jumlah sebenarnya saat melakukan perhitungan jumlah bibit ikan lele. Dalam perhitungan bibit lele ada aspek yang harus diperhatikan salah satunya adalah gerakan lele. Mengamati gerak bibit lele ini menyulitkan pembudidaya dalam menghitung bibit ikan lele. Sehingga ketepatan dalam mengamatinya bisa meleset dan kurang akurat. Dalam permasalahan tersebut dibuatlah sebuah sistem yang dapat melakukan perhitungan bibit ikan lele dari sebuah video yang memanfaatkan teknologi pengolahan citra digital menggunakan metode *background subtraction* dan *connected component labeling*.

4.1.1 Deskripsi Sistem

Pada penelitian ini dirancang dan dibangun sebuah aplikasi desktop perhitungan bibit ikan lele menggunakan metode *background subtraction* dan *connected component labeling*.

Aplikasi digunakan oleh satu pengguna atau *user*. Pengguna tersebut adalah pembudidaya bibit ikan lele yang memiliki hak akses untuk melihat tahapan proses perhitungan mulai dari memasukkan video sampai dengan *output* yang berupa jumlah bibit ikan lele.

4.1.2 Analisa Pengguna

Analisa pengguna merupakan analisis terhadap siapa saja aktor yang terlibat dalam sistem perhitungan jumlah bibit ikan lele menggunakan metode *background subtraction* dan *connected component labeling*. Analisa pengguna pada penelitian ini berjumlah 2 aktor, yaitu admin dan *user* yang dijelaskan pada Tabel 4.1:

Tabel 4. 1 Analisa Pengguna

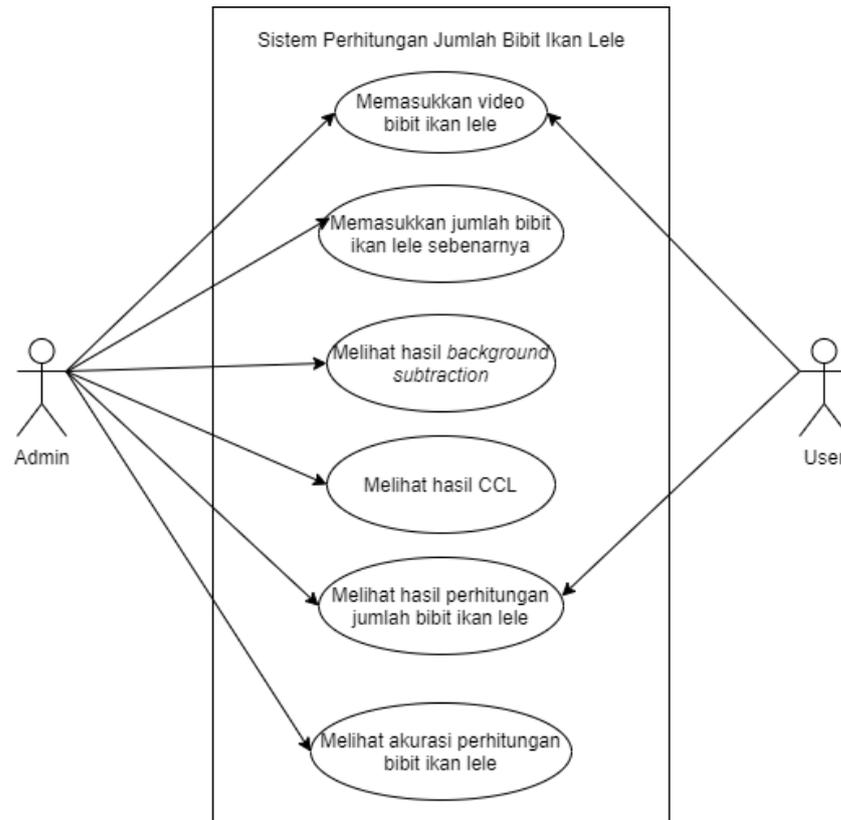
No.	Pegguna	Keterangan
1.	Admin	Admin merupakan aktor yang dapat mengelola sistem secara keseluruhan, seperti: <ul style="list-style-type: none"> • Pemilihan video bibit ikan lele • Memasukkan jumlah bibit ikan lele sebenarnya pada sistem • Melihat hasil <i>background subtraction</i> • Melihat hasil <i>connected component labeling</i> • Melihat hasil perhitungan jumlah bibit ikan lele pada sistem berdasarkan <i>input</i> video yang diberikan • Melihat hasil akurasi prediksi perhitungan jumlah bibit ikan lele
2.	<i>User</i>	<i>User</i> pada sistem ini adalah pembudidaya bibit ikan lele yang dapat melakukan proses pengunggahan <i>input</i> cira video dan melihat hasil perhitungan jumlah bibit ikan lele pada sistem.

4.1.3 Analisa Data

Data yang digunakan pada penelitian ini berupa citra video dari video rekaman gerak bibit ikan lele yang diambil menggunakan kamera *handphone*. Selanjutnya citra video disimpan dengan penamaan yang telah ditetapkan seperti yang telah dijelaskan dalam Bab II pada metode pengambilan.

4.1.4 Kebutuhan Fungsional

Pada analisis ini dijelaskan beberapa langkah aplikasi terdiri dari mendefinisikan data yang diperlukan sesuai dengan kebutuhan sistem yang akan dirancang. Berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Berikut adalah *usecase* kebutuhan fungsional dari aplikasi ini:



Gambar 4. 1 *Usecase* Diagram Sistem

Pada Gambar 4.1 merupakan *usecase* diagram sistem yang menjelaskan kegiatan yang dilakukan admin pada sistem perhitungan jumlah bibit ikan lele menggunakan metode *background subtraction* dan *connected component labeling*. Berikut penjelasan dari Gambar 4.1 :

- a. Admin dapat memasukkan memasukkan video bibit ikan lele yang akan dihitung. Selanjutnya, memasukkan jumlah bibit ikan lele sebenarnya pada sistem, Kemudian admin dapat melihat hasil *background subtraction* dan proses CCL. Selanjutnya admin dapat melihat hasil perhitungan dan akurasi dari hasil perhitungan jumlah bibit ikan lele pada sistem.
- b. *User* pada sistem perhitungan jumlah bibit ikan lele menggunakan metode *background subtraction* dan *connected component labeling*. *User* dapat memasukkan video bibit ikan lele yang akan dihitung. Kemudian pengguna dapat melihat hasil perhitungan jumlah bibit ikan lele pada sistem.

4.1.5 Kebutuhan Non-Fungsional

Merupakan batasan layanan atau fungsi yang ditawarkan sistem seperti batasan waktu . Berikut adalah kebutuhan non-fungsional dari aplikasi ini :

- Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak terdiri atas spesifikasi kebutuhan yang akan digunakan untuk membangun sistem, yaitu:

Tabel 4. 2 Spesifikasi Perangkat Lunak

No.	Nama Perangkat Lunak
1.	Sistem Operasi Windows 10
2.	Visual Studio Code
3.	Python
4.	<i>Library</i> OpenCV
5.	Qt Designer

- Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras yang diperlukan dalam pembuatan sistem, yaitu:

Tabel 4. 3 Spesifikasi Perangkat Keras

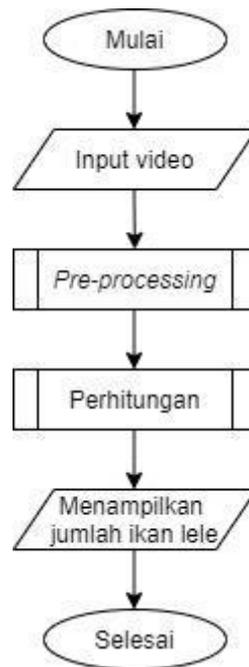
No.	Nama Perangkat Keras
1.	Intel Core i3-6006U Skylake gen 6 th
2.	Memory 4GB DDR4
3.	HardDisk 500 GB
4.	Kamera <i>handphone</i>

4.2 Perancangan

Dalam perancangan aplikasi Perhitungan Bibit Ikan Lele Menggunakan Metode *Background Subtraction* dan *Connected Component Labeling* penulis menggunakan *flowchart* dan *mockup system*. Berikut merupakan penjelasan dari *flowchart* sistem dan *flowchart* perhitungan metode yang digunakan dan *mockup system* :

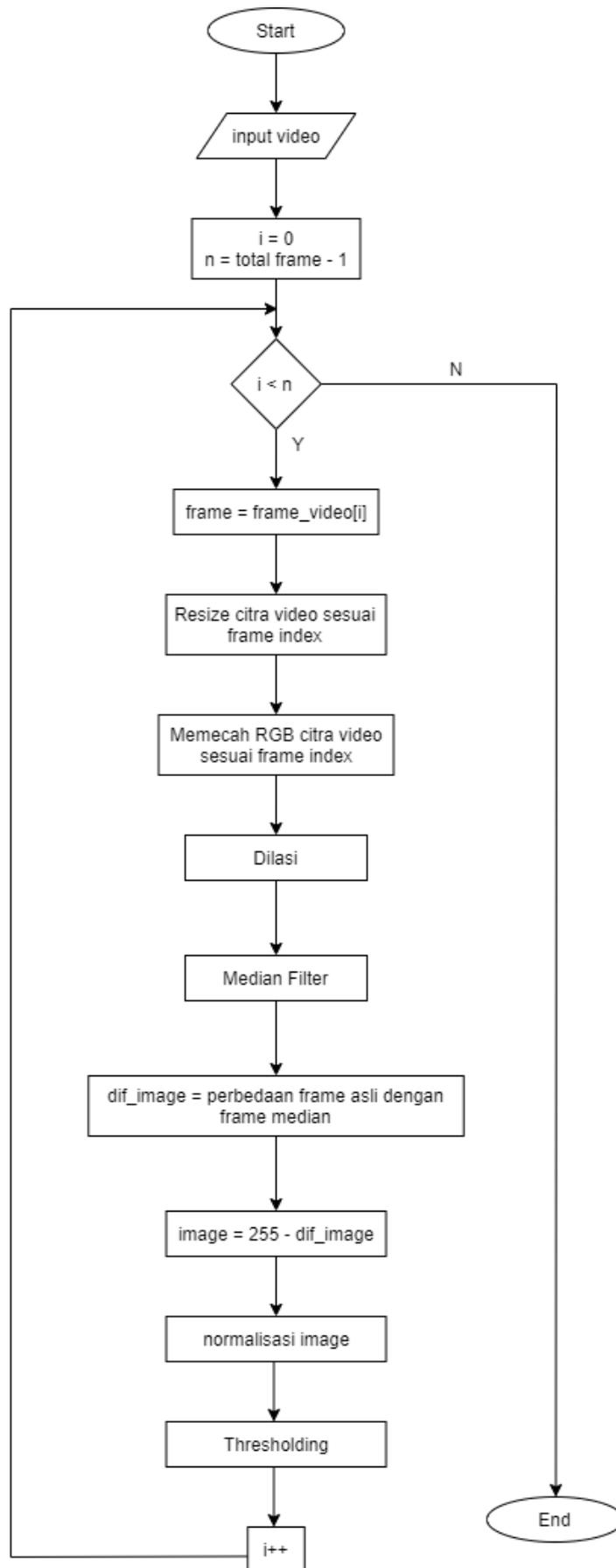
4.2.1 Flowchart Sistem

Desain sistem perhitungan jumlah bibit ikan lele dalam penelitian ini digambarkan menggunakan *flowchart* sebagai berikut:



Gambar 4. 2 *Flowchart* Desain Sistem

Pada gambar 4.2 merupakan *flowchart* desain sistem, proses yang pertama adalah memasukan video gerakan bibit ikan lele ke dalam sistem. Kemudian proses selanjutnya video tersebut dilakukan proses *preprocessing*, setelah itu perhitungan dengan menggunakan metode *Connected Component Labeling* (CCL). Lalu selanjutnya kita mendapatkan *output* jumlah bibit ikan lele.



Gambar 4. 3 *Flowchart Pre-Processing*

Pada gambar 4.3 merupakan *flowchart* dari proses *preprocessing*. Berikut penjelasan dari tahapan-tahapan *flowchart* bersangkutan, yaitu:

1. Pertama melakukan *input* citra video yang akan diproses ke dalam sistem
2. Selanjutnya, membaca total *frame* pada citra video yang dimasukkan dengan menggunakan *library* pada OpenCV, yaitu `cv2.CAP_PROP_FRAME_COUNT`.
3. Kemudian melakukan proses pengolahan citra video ketika nilai dari *looping* ke *i* memiliki nilai kurang dari total *frame* video.
4. Langkah selanjutnya yang dilakukan, yaitu:
 - *Resize frame* video sesuai *frame* index yang bertujuan untuk mengubah resolusi atau ukuran horizontal dan vertikal suatu citra. Pada penelitian ini menggunakan *library* OpenCV yaitu `cv2.resize()`.
 - Kemudian dilakukan proses pemecahan nilai RGB citra video sesuai index *frame* yang bertujuan untuk menghasilkan ekstraksi nilai *red*, *green*, *blue*. Pada penelitian ini menggunakan *library* OpenCV yaitu `cv2.split()`.
 - Selanjutnya dilakukan dilasi yang bertujuan untuk memperbesar segmen objek (citra biner) dengan menambah lapisan disekeliling objek atau dengan penggabungan titik latar (0) menjadi bagian dari objek (1). Pada penelitian ini menggunakan *library* OpenCV yaitu `cv2.dilate()`.
 - Median filter dilakukan bertujuan untuk memperhalus dan mengurangi derau pada citra (Han, 2019). Derau pada penelitian ini berbentuk bintik-bintik putih dan timbulnya derau disebabkan oleh pencahayaan. Pada penelitian ini menggunakan *library* OpenCV yaitu `cv2.medianBlur()`. Cara kerja dari filter ini dirumuskan pada persamaan 4.1 berikut (Guslianto, 2020):

$$g(x, y) = \text{Median}\{f(x - i, y - j), (i, j) \in W\} \quad (4.1)$$

Keterangan:

$g(x, y)$ = Citra yang dihasilkan dari citra keabuan
 $f(x, y)$ = W sebagai *window* yang ditempatkan pada bidang citra
 (i, j) = Elemen dari *window*

Median filter mengambil area tertentu pada citra sesuai dengan ukuran mask yang telah ditentukan (umumnya 3×3), kemudian dilihat setiap nilai piksel pada area tersebut, dan nilai tengah pada area diganti dengan nilai median. Cara memperoleh nilai median adalah: nilai keabuan dari titik-titik pada matriks diurutkan dari nilai terkecil hingga yang terbesar, kemudian ditentukan nilai yang berada di tengah dari deret piksel (Wedianto et al., 2016).

- Kemudian melakukan *background subtraction*, dengan melakukan proses pencarian nilai perbedaan pada frame yang disimpan pada variabel *dif_image*, setelah itu dilakukan pengurangan nilai perbedaan variabel *dif_image* dengan rumus $255 - dif_image$. Pada penelitian ini pencarian perbedaan *frame* dilakukan dengan menggunakan `cv2.absdiff()`. Untuk rumus *background subtraction* direpresentasikan pada persamaan 4.1

$$F(x, y) = |I_t(x, y) - B_t(x, y)| > T \quad (4.2)$$

I_t = *frame* masukan

B_t = model *background*

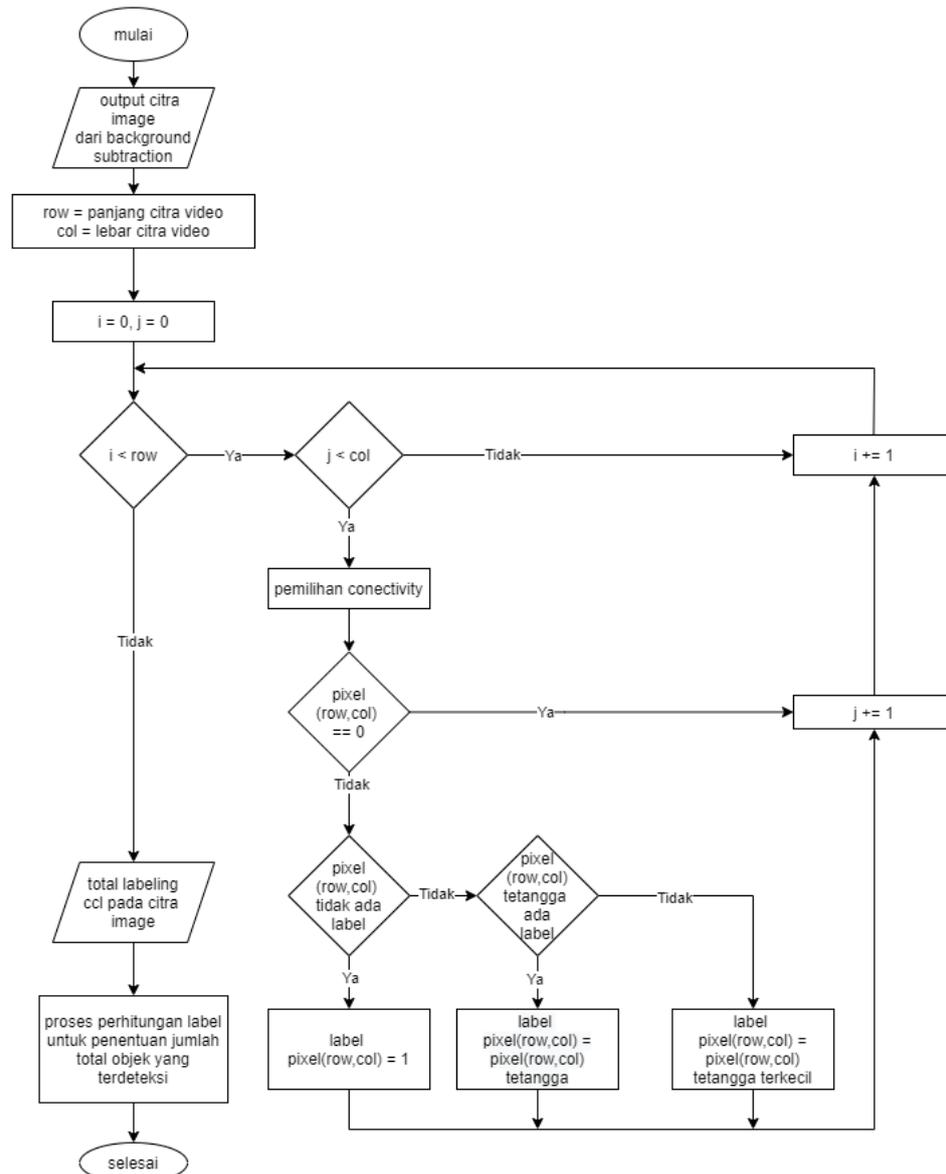
F = *foreground* atau objek bergerak yang terdeteksi

Selisih nilai piksel dari *frame* masukan dan model *background* akan dibandingkan dengan nilai *threshold*. Nilai piksel yang lebih dari nilai *threshold* akan dideteksi sebagai *foreground* dan sebaliknya dideteksi sebagai *background* (Wahyudi et al., 2015).

- Normalisasi *image* bertujuan untuk merubah dimensi region tiap karakter. Pada penelitian ini menggunakan *library* OpenCV yaitu `cv2.normalize()`.
- *Thresholding* bertujuan untuk mempartisi citra dengan mengatur nilai intensitas semua piksel yang lebih besar dari nilai *threshold* T sebagai latar depan dan yang lebih kecil dari nilai *threshold* T sebagai latar belakang atau sebaliknya (Budiarti, 2006). Pada penelitian ini

menggunakan *library* OpenCV yaitu `cv2.THRESH_OTSU` + `cv2.THRESH_BINARY_INV`.

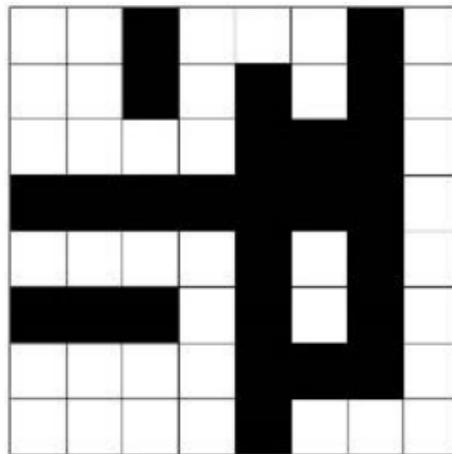
5. Setelah melakukan proses *thresholding*, langkah selanjutnya adalah proses CCL yaitu melakukan pelabelan untuk bibit ikan lele.



Gambar 4. 4 *Flowchart* Perhitungan

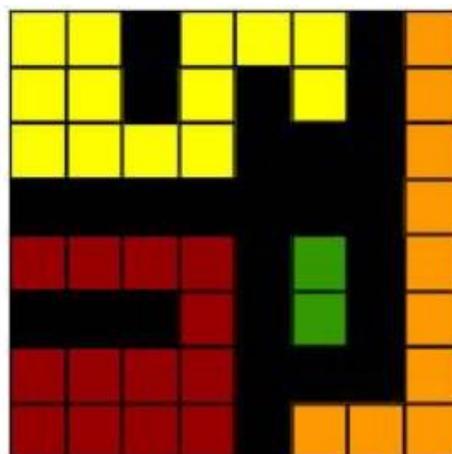
Pada gambar 4.4 diatas merupakan *flowchart* metode *Connected Component Labeling* (CCL). Input berupa citra *image* dari *background subtraction*. Kemudian dilakukan pengecekan nilai piksel pertama pada citra masukan (kolom pertama dan

baris pertama) Selanjutnya jika nilai piksel 0 (latar belakang), lewati dan pindah ke kolom berikutnya dari piksel tetapi masih dalam baris yang sama. Apabila nilai piksel adalah 1 dan tetangga terdekatnya masih belum diberi label, maka diberi label pada piksel ini menggunakan label 1. Akan tetapi jika hanya satu tetangga terdekat yang diberi label, maka berikan label yang sama pada piksel ini dengan menggunakan label tetangga. Namun, jika ada 2 atau lebih label di tetangga, gunakan label terkecil untuk memberi label piksel ini. Setelah memberi label pada setiap piksel dan melewati latar belakang, jangan lupa untuk memperbarui label. Proses ini dilakukan pada seluruh kolom dan baris. Berikut representasi dari alur perhitungan CCL



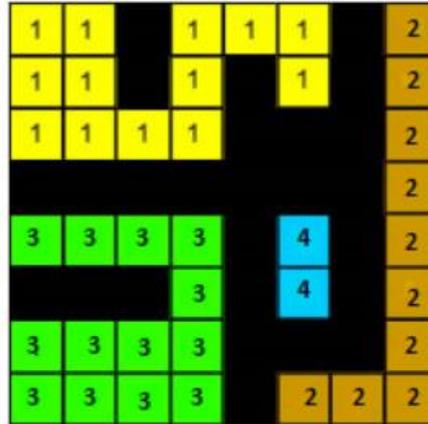
Gambar 4. 5 Awal Berupa Biner (0 dan 1)

Pada Gambar 4.5 "Blok" mewakili piksel. Putih bernilai '1' dan hitam bernilai '0'. Pada gambar diatas piksel hitam atau bernilai 0 tidak ditandai. Sehingga dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Contoh Representasi Visual dari Label

Piksel yang diberikan label bernilai 1 atau putih. Di sini, warna hanya merupakan representasi visual dari label.



Gambar 4. 7 Hasil dari Proses Pelabelan

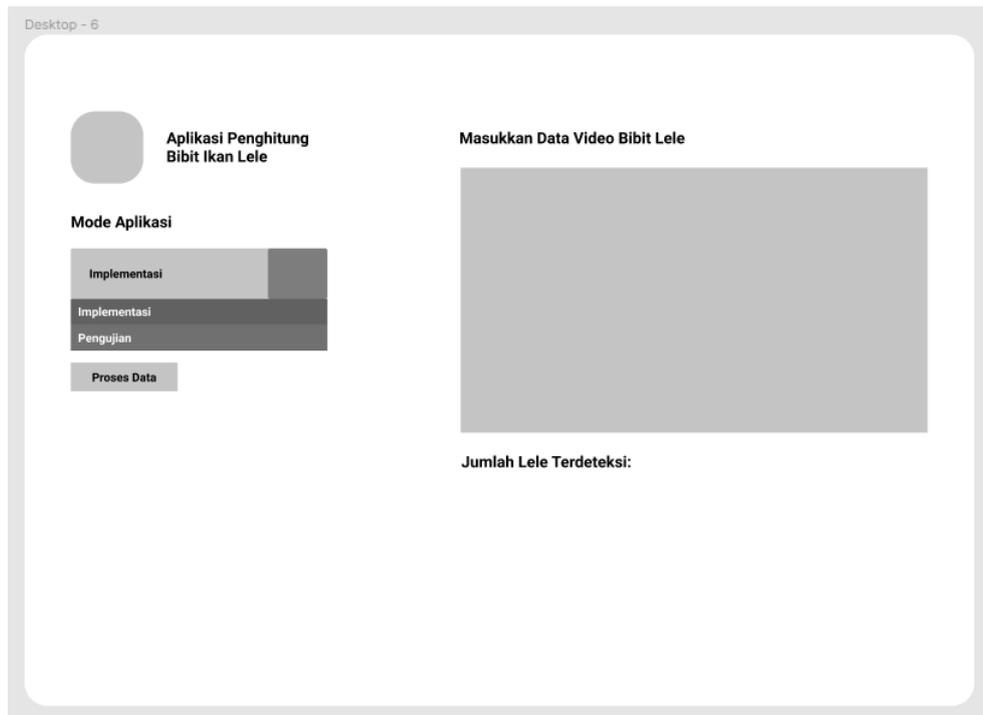
Pada Gambar 4.7 merupakan hasil akhir dari pelabelan pada objek yang terdeteksi menggunakan CCL.

4.2.2 *Mockup* Sistem

Mockup sistem merupakan bagian dari sebuah perancangan desain sistem informasi yang menjembatani interaksi antara pengguna dengan sistem. *Mockup* sistem dapat menerima informasi masukan dari pengguna dan memberikan informasi keluaran untuk pengguna. Pada aplikasi ini perancangan desain sistem dibagi menjadi dua tampilan, yaitu *form* implementasi dan *form* pengujian.

a. *Mockup Form* Utama

Mockup form utama merepresentasikan tampilan utama aplikasi perhitungan jumlah bibit ikan lele



Gambar 4. 8 *Mockup Form Utama*

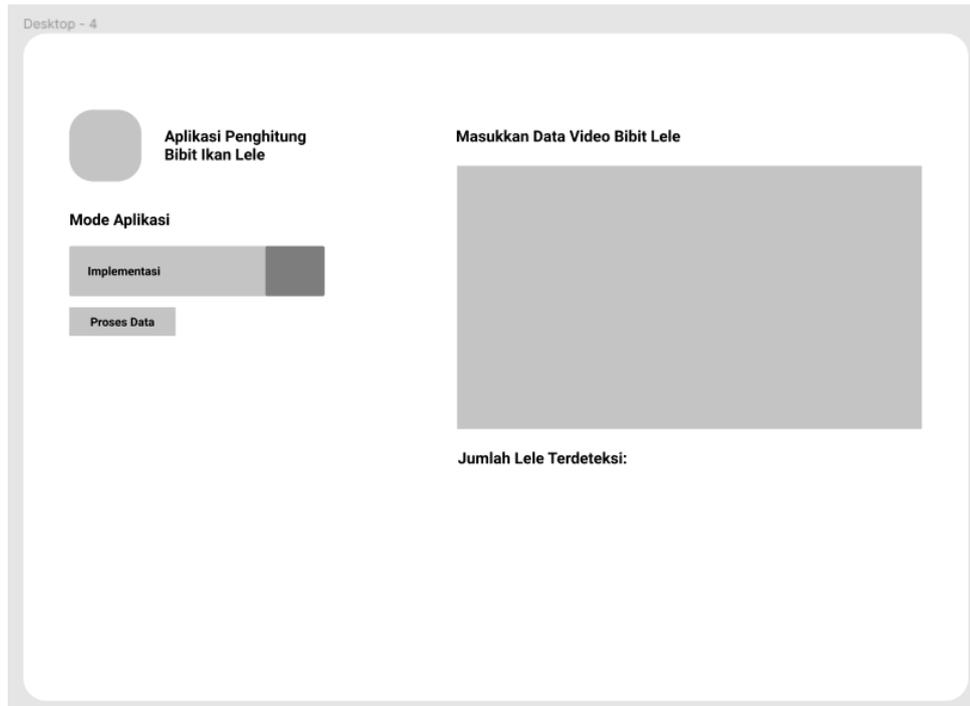
Pada gambar 4.8 terdapat 1 buah *combobox*, 1 buah *button*, dan 1 layar video.

Berikut penjelasan dari *mockup form* utama diatas:

- Sub-menu Mode Aplikasi merepresentasikan mode aplikasi yang ingin digunakan oleh pengguna
- *Combobox* merepresentasikan pilihan menu yang dapat dipilih oleh pengguna yang digunakan sebagai pengujian oleh admin atau implementasi oleh *user*
- Layar Video merupakan layar untuk menampilkan video yang telah diunggah oleh pengguna
- *Button* Proses Data untuk mengambil sebuah video pada *file open dialog* ke dalam sistem dan melakukan deteksi jumlah bibit ikan lele

b. *Mockup Form Implementasi*

Mockup form implementasi merepresentasikan tampilan aplikasi perhitungan jumlah bibit ikan lele yang ditujukan kepada *user*.



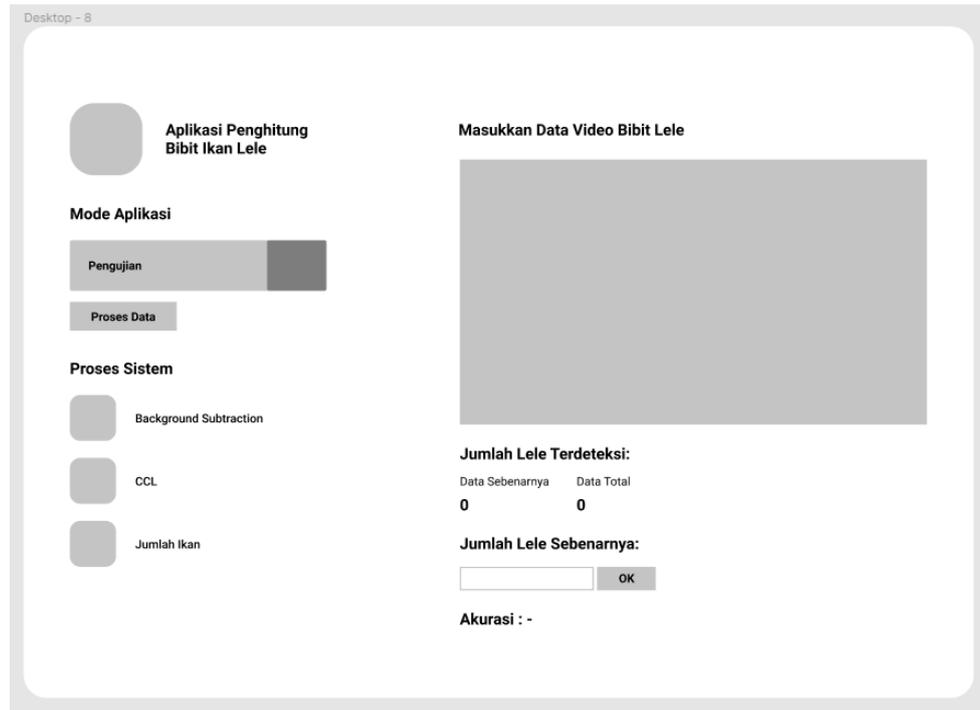
Gambar 4. 9 *Mockup Form Implementasi*

Pada gambar 4.9 terdapat 1 buah *combobox*, 1 buah *button*, dan 1 layar video. Berikut penjelasan dari *mockup form* implementasi diatas:

- Mode aplikasi implementasi merepresentasikan mode aplikasi pengujian sebagai *user*.
- Layar Video merupakan layar untuk menampilkan video yang telah diunggah oleh *user*.
- *Button* Proses Data untuk mengambil sebuah video pada *file open dialog* ke dalam sistem dan melakukan deteksi jumlah bibit ikan lele

c. *Mockup Form Pengujian*

Mockup form pengujian aplikasi perhitungan bibit ikan lele yang ditujukan kepada admin untuk pengujian.



Gambar 4. 10 *Mockup Form Pengujian*

Pada gambar 4.10 terdapat 1 buah *combobox*, 2 buah *button*, dan 1 layar video. Berikut penjelasan dari *mockup form* pengujian diatas:

- Mode Aplikasi pengujian merepresentasikan model aplikasi pengujian sebagai admin.
- Sub-menu Proses Sistem merepresentasikan posisi yang sedang dilakukan oleh aplikasi terhadap citra video inputan dalam melakukan perhitungan jumlah bibit ikan lele.
- Sub-menu Proses Sistem *Background Subtraction* untuk menampilkan hasil proses *background subtraction*.
- Sub-menu Proses Sistem CCL untuk menampilkan hasil proses pelabelan *connected component labeling*.
- Sub-menu Proses Sistem Jumlah Ikan untuk menampilkan hasil akurasi perhitungan jumlah bibit lele.
- Layar Video merupakan layar untuk menampilkan video yang telah diunggah oleh pengguna.
- *Button* Proses Data untuk mengambil sebuah video pada *file open dialog* ke dalam sistem dan melakukan deteksi jumlah bibit ikan lele.