BAB II. LANDASAN TEORI

Pada bab ini berisi teori-teori yang mendasari dan berkaitan dengan masalah perencanaan dan pembuatan aplikasi yang digunakan untuk memudahkan pemahaman dan pemecahan terhadap masalah yang ada.

2.1 Studi Literatur

Beberapa penelitian yang dijadikan rujukan dalam penelitian ini adalah penelitian yang dilakukan oleh Karina dkk., pada tahun 2018 berjudul "Perbandingan Algoritma *Background Subtraction* dan *Optical Flow* Untuk Deteksi Manusia" pada percobaan menggunakan 1 sampel video dengan durasi 3 menit durasi telah memperoleh hasil dalam mendeteksi keberadaan manusia, menggunakan algoritma pengurangan latar belakang memiliki tingkat kebenaran 80,56%. Hasil penghitungan jumlah manusia terdeteksi, algoritma pengurangan latar belakang memiliki tingkat kebenaran 2,78%. (Kaloh et al., 2018).

Hasil penelitian yang dilakukan oleh Kukuh pada tahun 2017 yang berjudul "MENGHITUNG OBYEK 2D MENGGUNAKAN CONNECTED COMPONENT LABELING" pada algoritma CCL dapat digunakan secara optimal untuk menghitung obyek biji jagung dengan penambahan metode *image preprocessing* lain seperti *binary thresholding* dan median filter sebelum dilakukan CCL agar hasil deteksi dan perhitungan dapat mencapai akurasi di atas 80% (Yudhistiro, 2017).

Pada jurnal hasil penelitian oleh Danang pada tahun 2017 yang berjudul "PENGEMBANGAN SISTEM ESTIMASI KECEPATAN PADA KENDARAAN BERGERAK BERBASIS PENGOLAHAN CITRA DIGITAL" video diekstrak menjadi kumpulan *frame* kemudian dilakukan *preprocessing* untuk minimalisir efek bayangan. Selanjutnya *frame* dicari foreground-nya menggunakan metode Gaussian Mixture Model. Kemudian hasil *foreground* melalui proses filter, *shadow removal*, dan operasi morfologi. Selanjutnya metode *connected component labeling* digunakan untuk deteksi objek dan menghitung centroid dari objek tersebut. Dari hasil uji coba, sistem yang dibangun mampu mengestimasi kecepatan kendaraan dengan tingkat akurasi terendah adalah 87,01% dan tertinggi adalah 99,38% dengan

faktor pemilihan daerah ROI, deteksi objek, dan akurasi input parameter geometris yang baik (Wicaksono, 2017).

Pada penelitian ini, digunakan metode *background subtraction* untuk mendeteksi subtraksi pada *background* dan metode CCL digunakan untuk deteksi objek dan menghitung dari objek. Pada sub bab selanjutnya akan disajikan beberapa teori dasar dalam perhitungan jumlah bibit ikan lele.

2.2 Dasar Teori

2.2.1 Video

Video merupakan kumpulan gambar (*frame*) yang dibaca berurutan dengan jumlah tertentu tiap detik yang menunjukkan tindakan berkesinambungan dalam urutan gambar. Sebuah rekaman video (*video shot*) dikatakan sebagai urutan gambar yang tersusun secara berkelanjutan oleh sebuah kamera. Di dalam sebuah rekaman video dari beberapa adegan yang disebut sebagai *video scene* yang masing-masing menggambarkan suatu peristiwa (Widiarto, 2016).

2.2.2 Citra

Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Tersusun oleh sekumpulan piksel yang memiliki koordinat (x,y) dan amplitudo f(x,y). Koordinat (x,y) menunjukkan letak atau posisi piksel dalam suatu citra, sedangkan amplitudo f(x,y) menunjukkan nilai intensitas warna citra. Citra digital dibagi menjadi tiga jenis yaitu citra warna, citra *grayscale*, dan citra biner (Tarigan et al., 2016).

2.2.3 Pengolahan Citra Digital

Dalam pengolahan citra (*digital image processing)* mempelajari tentang bagaimana suatu citra itu dibentuk, diolah, dan dianalisis sehingga menghasilkan informasi yang dapat dipahami oleh manusia. Teknik olah citra tersebut meliputi modifikasi kecemerlangan, negasi, peningkatan kontras, dan *thresholding*, (Rahmadianto et al., 2019). Dengan kata lain pengolahan citra berarti suatu cara mengusahakan suatu citra menjadi citra lain yang lebih sempurna atau yang diinginkan (Sulistiyani et al., 2016).

2.2.4 *Background Subtraction*

Background subtraction merupakan salah satu teknik pada bidang pengolahan citra dan *computer vision* yang bertujuan untuk mendeteksi atau mengambil *foreground* dari *background* untuk diproses lebih lanjut. *Background subtraction* biasanya digunakan untuk deteksi objek bergerak dengan kamera statis. Jika perbedaan diantara nilai piksel hasil *subtract* lebih besar dari nilai *threshold* maka piksel akan dikategorikan bagian *foreground* (Pambudi et al., 2019). Rumus dari *background subtraction* direpresentasikan pada persamaan 2.1:

$$F(x,y) = |I_t(x,y) - B_t(x,y) > T \quad (2.1)$$

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*.

2.2.5 Morfologi

Morfologi adalah teknik pengolahan citra digital dengan menggunakan bentuk obyek sebagai pedoman dalam pengolahan. Nilai dari setiap piksel dalam citra digital hasil diperoleh melalui proses perbandingan antara piksel yang bersesuaian pada citra digital masukan dengan piksel tetangganya. Operasi morfologi standar yang biasa dilakukan adalah proses erosi dan dilasi (Antika et al., 2012).

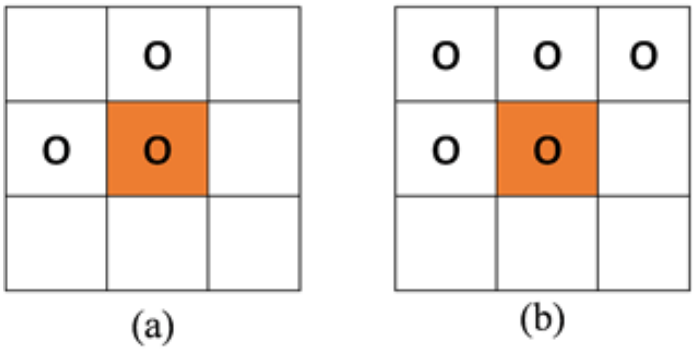
Dilasi akan menyebabkan pertumbuhan ukuran objek dengan menukar tiap nilai piksel dengan nilai maksimum piksel tetangga di sekelilingnya. Hal ini akan meningkatkan jumlah piksel tepi objek di citra. Sedangkan erosi akan mengurangi jumlah piksel dengan menukar nilai tiap piksel dengan nilai minimum piksel tetangga di sekelilingnya. Hal ini berdampak pada hilangnya tepi objek pada citra (Susanto, 2019).

2.2.6 *Connected Component Labeling*

Algoritma *Connected Component Labeling* adalah metode yang dapat digunakan untuk mengklasifikasikan region atau objek dalam citra digital.

Algoritma ini menerapkan teori *connectivity* piksel dari citra. Seluruh piksel pada sebuah region disebut *connected* atau memiliki hubungan bila mematuhi aturan *adjacency* atau "kedekatan" piksel. Aturan kedekatan piksel ini memanfaatkan ketetanggaan antara piksel satu dengan piksel yang lainnya. Oleh karena itu setiap piksel yang bersifat *connected* pada dasarnya memiliki *adjacency* satu sama lain karena mempunyai hubungan ketetangggan atau *neighbourhood*. Citra yang dapat diolah dengan menggunakan algoritma *connected component labeling* ini adalah citra biner atau citra monokrom. Selain itu, ketetanggaan harus memiliki panjang atau jarak 1 unit atau bersifat langsung antara piksel satu dengan yang lain tanpa ada perantaranya.

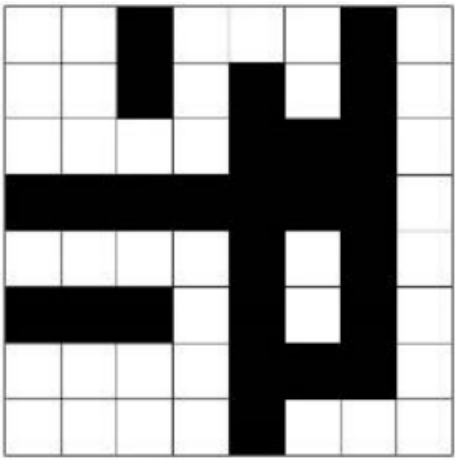
Terdapat dua macam konektivitas yang digunakan pada citra 2 dimensi yaitu: 4-*Connected Neighbors* 8- *Connected Neighbors* (Aini D. et al., n.d.). Dapat dilihat pada Gambar 2.1.



Gambar 2. 1 a.) 4-*connectivity* b.) 8-*connectivity*

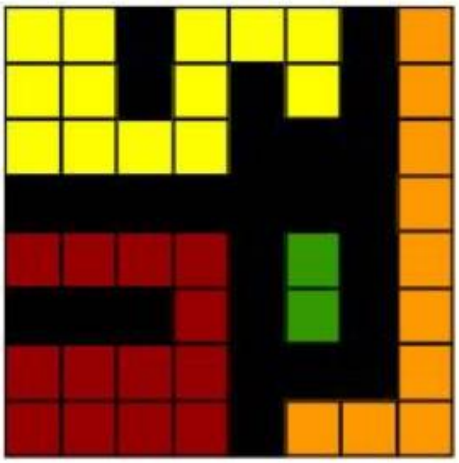
Pada Gambar 2.1 terdapat sebuah *frame*, pada satu *frame* beberapa piksel atau kotak. Lalu pada salah satu piksel terdapat objek yang direpresentasikan dengan warna oranye. Untuk mendapatkan label atau nilai dari objek bersangkutan maka harus dilakukan pencarian atau pembacaan piksel tetangganya pada *frame* bersangkutan. Apabila menganut 4-*connectivity* maka akan melihat piksel tetangganya yang ada pada sebelah kiri piksel bersangkutan lalu pada sebelah atas selanjutnya pembacaan dilanjutkan pada piksel tetangga sebelah kanan dan pembacaan pada piksel pada sebelah bawah dari pksel objek bersangkutan. Apabila menganut 8-*connectivity* maka akan melakukan pencarian pada 8 kotak yang ada pada sekelilingnya.

Berikut penjelasan mengenai algoritma *Connected Componenet Labeling*:



Gambar 2. 2 Awal Berupa Biner (0 dan 1)

Pada Gambar 2.2 "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 2.3.



Gambar 2. 3 Contoh Representasi Visual dari Label

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

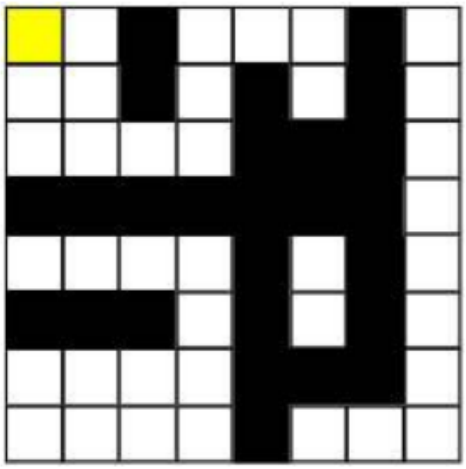
First Pass

Proses ini merupakan proses awal dari pelabelan, pada Gambar 4. merupakan ilustrasi sebagai matrik biner. Yang memiliki nilai 1 sebagai putih dan 0 sebagai hitam. Proses ini dimulai dari sudut kiri atas.

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	1	1	0	0	0	1
1	1	1	1	0	1	1	1

Gambar 2. 4 Ilustrasi Bentuk Biner

Pertama periksa pixel pada pojok kiri atas, apakah pixel pertama bernilai 1/*foreground*. Jika Ya, maka akan diberikan label baru (misalkan label di tandai dengan warna kuning).

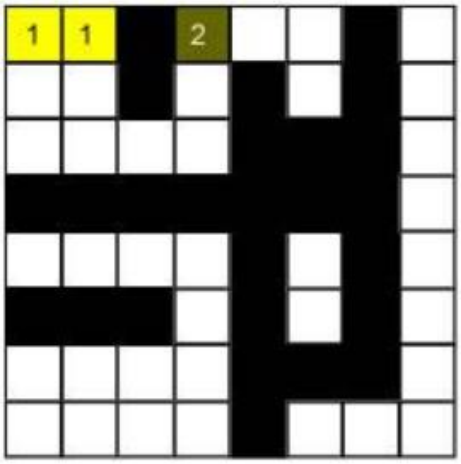


Gambar 2. 5 Label Baru

Selanjutnya periksa pixel di baris 1, kolom 2, atau pixel (1,2), cek apakah pixel (1,2) sama dengan pixel disebelah kiri (1,1) jika “Ya” tandai label dengan warna kuning dan di beri nomor label “1”. Kemudian periksa piksel (1, 3) apakah bernilai 1/foreground atau bernilai 0, apa bila bernilai “0” maka dilewati agar tetap bernilai hitam atau tetap sebagai pixel latar belakang.

Selanjutnya cek pixel (1, 4). Apakah ada piksel di atasnya “Tidak”. Cek pixel ke kiri, apakah pixel disebelah kiri “tidak” kareana pixel sebelah kiri merupakan

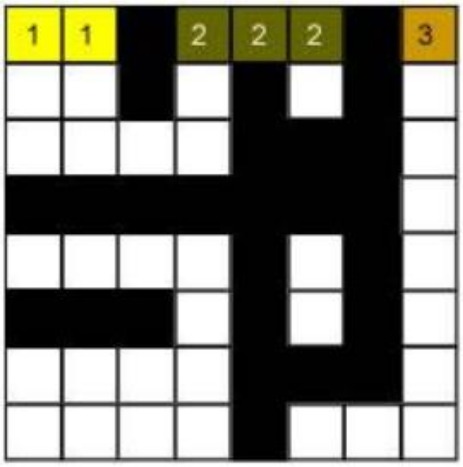
pixel latar belakang. Sehingga membuat label baru pada (1,4) dengan label 2 (ditampilkan sebagai warna kuning gelap).



Gambar 2. 6 Cek Label 2

Kemudian piksel berikutnya adalah (1,5) dan (1,6) akan memiliki piksel di sebelah kirinya. Jadi, pixel (1,5) dan (1,6) merupakan label 2 dapat dilihat pada Gambar 2.6 Pada pixel (1, 7) merupakan pixel latar belakang maka diabaikan. Pada pixel (1,8) membuat label baru dapat dilihat pada Gambar 2.7 (pixel di atas tidak ada, dan pixel ke kiri adalah piksel latar belakang).

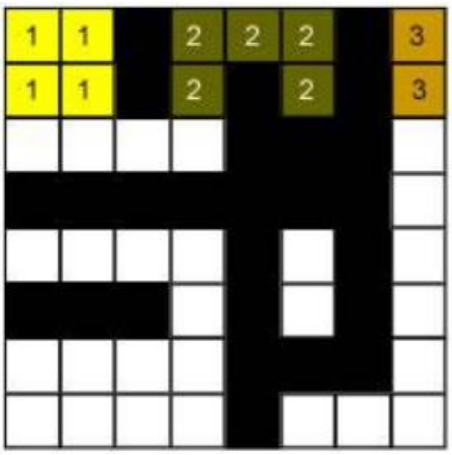
Setelah menyelesaikan baris 1, dapat dilihat pada Gambar 2.7.



Gambar 2. 7 Penyelesaian Baris 1

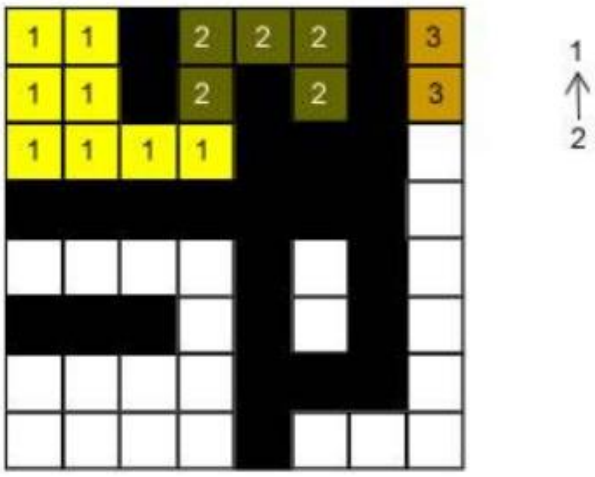
Setelah menyelesaikan baris 1, selanjutnya cek pixel (2,1). Apakah (2,1) memiliki tetangga di sebelah kirinya, apabila tidak cek tetangga atasnya. Label atas bernilai “1”, maka pixel (2,1) akan diberi label dengan label “1” (menyalin). Demikian pula untuk (2, 2). Hal yang sama berlaku untuk seluruh baris. Semua

piksel di baris kedua memiliki piksel tepat di atasnya. Maka hasilnya terlihat seperti Gambar 2.8.



Gambar 2. 8 Penyelesaian Baris 2

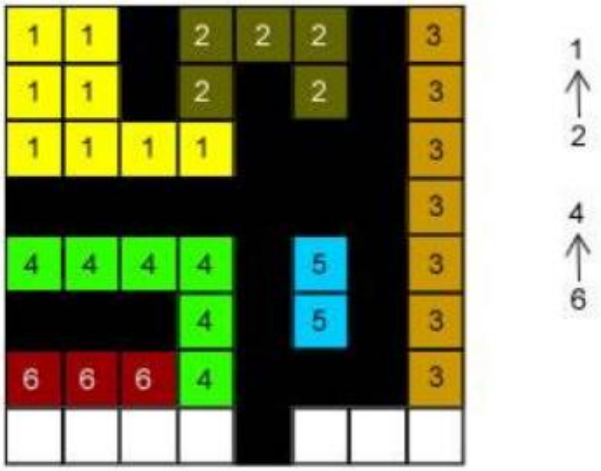
Baris ke 2, selesai di cek. Kemdian pada baris ketiga terdapat piksel (3, 1) (3,2) dan (3, 3) cukup lurus ke depan. Maka akan di labeli dengan label '1'. Akan tetapi pada pixel (3, 4) memiliki sedikit kerumitan. Terdapat piksel di atas dan di sebelah kiri. Sedangkan keduanya memiliki label yang berbeda. Maka, mengambil label yang lebih kecil (paling kecil adalah label '1') dan meletakkannya di (3, 4). Kemudian akan menyimpan juga label 2 (label numerik yang lebih besar) yang merupakan anak dari 1 (menggunakan struktur data union-find). Dapat dilihat pada Gambar 2.9.



Gambar 2. 9 Proses Baris 3

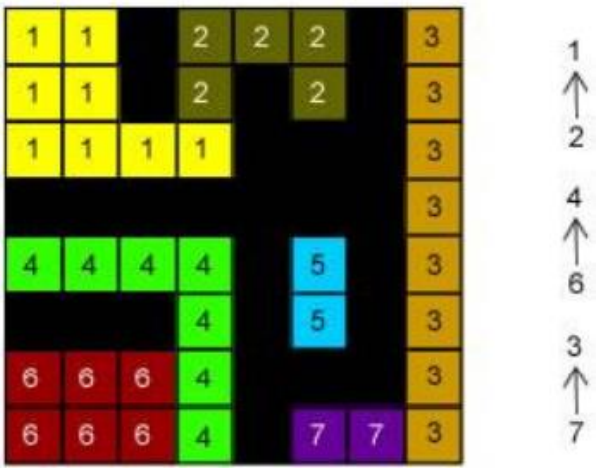
Pixel (3,4) terselesaikan. Proses selanjutnya hampir sama dengan proses diatas, merupakan penyelesaian baris ke 4. Setelah baris 4 selesai, kemudian cek

pada baris ke 5 hingga baris ke 7. Pada baris ke 7, memilik label baru di mulai pada pixel (7,1). Cek hingga proses baris ke 7 terselesaikan. Dapat dilihat pada Gambar 2.9.



Gambar 2. 10 Penyelesaian Baris 7

Setelah menyelesaikan baris ke 7, cek baris ke 8. Baris ke 8 juga memiliki label baru. Yaitu label (8,6). Dapatdilihat pada Gambar 2.11.



Gambar 2. 11 Penyelesaian Baris 8

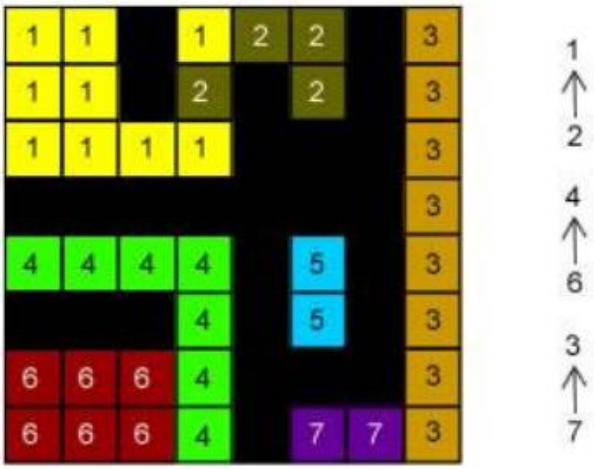
Proses "*First Pass*" selesai, selenjutnya dilanjutkan proses ke dua yaitu proses "*Second Pass*".

Second Pass

Pada Proses "first pass" hasil pelabelan cukup berantakan, sehingga memiliki banyak label yang harus di rapikan atau perbaiki.

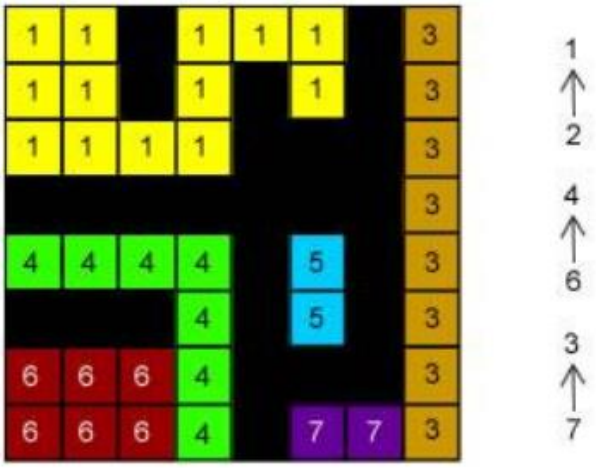
Untuk memulainya, kembali mengecek melalui setiap piksel satu demi satu. Proses ini dimulai dengan pixel (1, 1). Kemudian memeriksa struktur data union-find untuk label '1'. Pada pemeriksaan tersebut, menunjukkan bahwa '1' bukan anak dari label lain. Label 1 merupakan label itu sendiri. Demikian pula untuk pixel (1, 2) dan pixel (1, 3) merupakan piksel latar belakang.

Pada pixel (1,4) memeriksa struktur data untuk label “2”. Kemudian memperhatikan bahwa label “2” adalah anak dari label “1”. Kemudian, memeriksa label “1”. Pemberitahuan '1' adalah root. Maka label (1, 4) akan di gantikan dengan label “1”. Dapat dilihat pada Gambar 2.12.



Gambar 2. 12 Penggantian Label

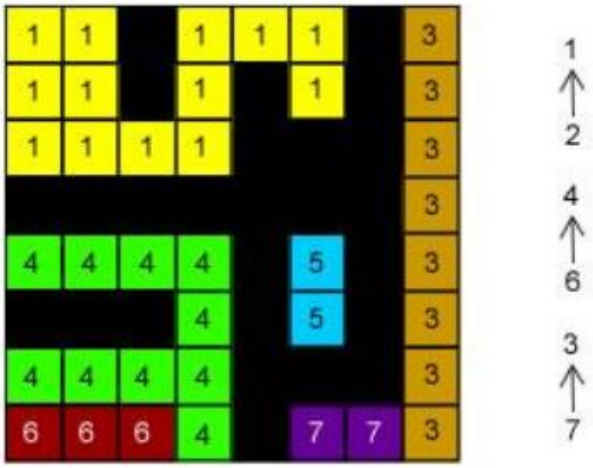
Mengonversi label “2” menjadi label “1” Demikian pula, proses berlangsung untuk seluruh baris. Cek sampai dengan baris ke 4. Hasilnya menyelesaikan baris 4 dapat dilihat pada Gambar 2.13.



Gambar 2. 13 Cek Baris Sampai dengan Baris 4

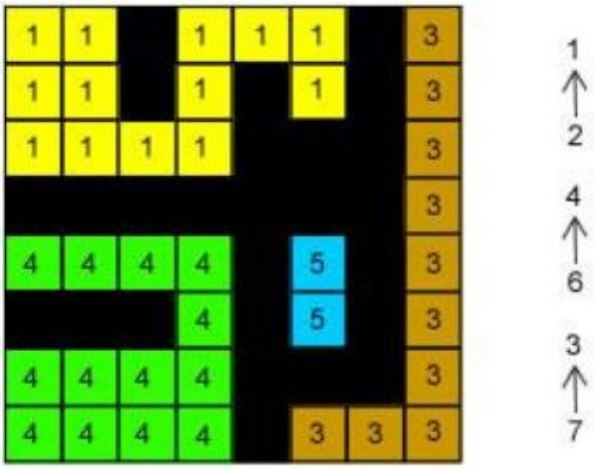
Baris ke 4 terselesaikan. Kemudian cek baris 5, baris ke 5 tidak berubah. Karena '4' adalah label root. Begitu juga baris “5”, baris “3” baris 6 juga tetap tidak berubah.

Selanjutnya, baris 7 mengalami perubahan dapat dilihat pada Gambar 2.14.



Gambar 2. 14 Baris 7 Mengalami Perubahan

Baris 7 terselesaikan. Kemudian cek baris 8, inilah hasil akhirnya dapat dilihat pada Gambar 2.15.



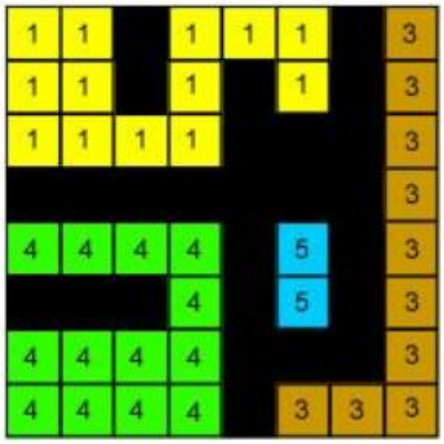
Gambar 2. 15 Hasil Akhir *Second Pass*

Dengan demikian memiliki hasil akhir berupa nilai label dalam setiap wilayah yang terhubung memiliki satu nilai.

Third Pass

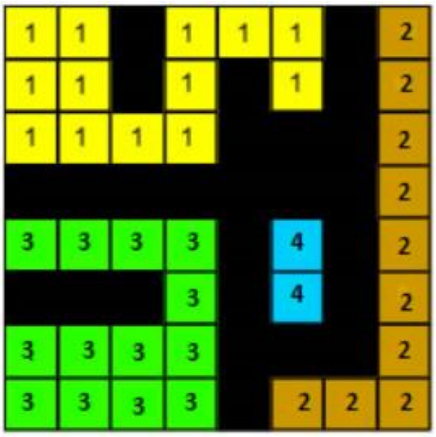
Pada proses ini merupakan proses pengurutan, apabila pada proses “*second pass*” masih memiliki label dengan urutan yang tidak urut maka pada proses “*third*

pass" akan di urutkan sesuai urutan. Pada proses akhir *second pass*, di dapatkan data seperti pada Gambar 2.16.



Gambar 2. 16 Proses Akhir *Second Pass*

Selanjutnya mengurutkan label berdasarkan angka sebenarnya. Jika sebelumnya adalah label "1" dan kemudian label "3" maka akan di cek, apakah ada label "2". Tidak, jika tidak maka label "3" diganti dengan label "2". Begitu seterusnya hingga semua label berurutan. Hasilnyadapat dilihat pada Gambar 2.17.



Gambar 2. 17 Hasil dari Proses *Third Pass*

Pada Gambar 2.17 merupakan hasil akhir dari pelabelan pada objek yang terdeteksi menggunakan CCL.

2.2.7 Ikan Lele

Ikan lele merupakan salah satu jenis ikan air tawar yang banyak dibudidayakan di Indonesia karena pertumbuhan lele cukup cepat dengan daya tahan yang tinggi serta mudah sekali berkembang biak di berbagai lokasi

pemeliharaan. Jenis ikan yang memiliki banyak nama dan julukan yang berbeda di beberapa negara, bahkan di Indonesia, ikan lele memiliki nama yang berbeda pada beberapa daerah, hal ini disebabkan karena ikan lele termasuk jenis ikan yang memiliki banyak spesies, namun demikian, secara ilmiah ikan lele lebih dikenal dengan nama clarias, berasal dari kata chlaros bahasa yunani yang berarti kuat atau lincah, seperti pada kenyataannya di alam bebas, ikan lele memang terkenal lincah dan mampu bertahan hidup meskipun dalam kondisi air dan kadar oksigen yang minimum (Zulkarnain & Susilowati, 2017).

2.2.8 Median Filter

Median filter sangat populer dalam pengolahan citra. Filter ini dapat dipakai untuk menghilangkan derau bintik-bintik. Nilai yang lebih baik digunakan untuk suatu piksel ditentukan oleh nilai median dari setiap piksel dan kedelapan piksel tetangga pada 8- ketetanggaan ((Manurung, 2017).