

## **BAB II**

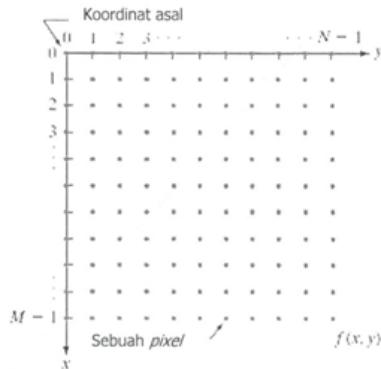
### **LANDASAN TEORI**

#### **2.1 Pengolahan Citra Digital**

Citra atau Gambar (Image) adalah sebuah informasi yang memiliki bentuk visual. Suatu citra yang diperoleh dari hasil kamera dan dapat diolah menggunakan komputer atau laptop merupakan citra digital, dapat juga diartikan bahwa citra adalah kumpulan dari piksel-piksel yang disusun dalam dua dimensi. Piksel merupakan elemen terkecil yang menyusun citra pada suatu titik tertentu mengandung nilai yang mewakili kecerahan dari sebuah warna, Umumnya Citra digital berbentuk persegi panjang yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya piksel sehingga ukuran citra selalu bernilai bulat.

Dalam komputer, setiap piksel diwakili oleh dua buah bilangan bulat (integer) untuk menunjukkan lokasinya dalam bidang citra dan sebuah nilai dalam bilangan bulat untuk menunjukkan cahaya atau terang-gelapnya piksel tersebut. Untuk menunjukkan lokasi pada suatu piksel, koordinat (0,0) digunakan sebagai posisi kiri atas dalam bidang citra, dan koordinat (m-1,n-1) digunakan sebagai posisi kanan bawah dalam citra berukuran (m x n) piksel. dapat dilihat pada gambar

2.1.



## Gambar 2.1 Koordinat Citra Digital

Sedangkan untuk menunjukkan tingkat pencahayaan suatu piksel, sering kali menggunakan bilangan bulat yang besarnya 8-bit, dengan lebar selang nilai 0 hingga 255, di mana nilai 0 merupakan warna hitam, nilai 255 merupakan warna putih dan tingkat abu-abu berada diantara nilai-nilai 0 dan 255

Pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya digunakan untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Secara umum, istilah pengolahan citra digital merupakan pemrosesan gambar berdua dimensi melalui komputer digital (Efford,2000). Suatu citra dapat juga didefinisikan sebagai fungsi  $f(x, y)$  berukuran M baris dan N kolom, dengan x dan y yang merupakan koordinat spasial, dan amplitudo  $f$  pada titik koordinat  $(x, y)$  dinamakan sebagai intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai dari x, y, dan nilai dari amplitudo  $f$  bernilai diskrit, maka dapat dikatakan bahwa citra tersebut merupakan citra digital,

Matrik pada citra digital dapat dituliskan sebagai berikut :

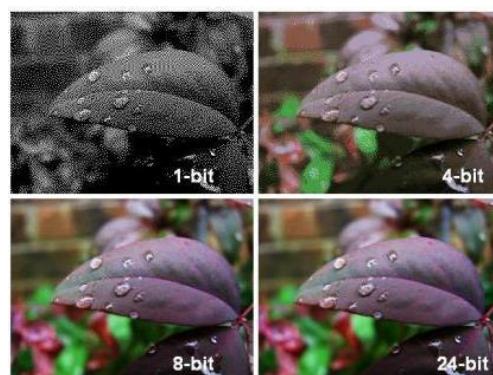
$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix} \quad \dots \quad (1)$$

Nilai pada suatu irisan antara baris dengan kolom (pada posisi x, y) disebut dengan *picture elements*, *image elements*, atau *pixels*. Istilah terakhir (pixel) merupakan yang paling sering digunakan pada citra digital. Nilai pada suatu pixel memiliki nilai rentang tertentu, dari nilai minimumnya hingga nilai maksimumnya. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. namun secara umum jangkauannya adalah 0 hingga 255. Citra dengan penggambaran seperti ini tergolong ke dalam citra integer. Berikut adalah jenis-jenis citra berdasarkan nilai pixelnya :

### 2.1.1 Citra Warna ( 24 bit )

Setiap *pixel* dari citra warna 24 bit diwakili dengan 24 bit sehingga memiliki total 16.777.216 variasi warna. Variasi warna ini untuk memvisualisasikan seluruh warna yang dapat dilihat oleh penglihatan manusia sudah lebih dari cukup. Karena penglihatan manusia hanya dapat membedakan hingga 10 juta warna.

Setiap poin informasi yang dimiliki *pixel* (RGB) disimpan ke dalam 1 byte data. Pada 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai Hijau pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah.



**Gambar 2.2 Citra Warna**

### 2.1.2 Citra Grayscale

Menurut Basuki (2005), Proses yang banyak dilakukan dalam image processing pada proses awal adalah merubah citra berwarna menjadi citra grayscale. Hal ini bertujuan untuk menyederhanakan model citra. Terdapat 3 layer matrix pada citra berwarna, yaitu Red-layer, Green-layer dan Blue-layer maka untuk melakukan proses-proses berikutnya tetap diperhatikan pada tiga layer diatas. Bila setiap proses perhitungan dilakukan menggunakan tiga layer, maka dilakukan tiga perhitungan yang sama. Sehingga konsep itu dirubah dengan mengubah tiga layer diatas menjadi 1 layer matrix grayscale dan akan menghasilkan citra grayscale. Dalam citra grayscale tidak ada lagi warna, yang ada adalah derajat keabu-abuan. Untuk mengubah citra berwarna yang mempunyai nilai matrix masing-masing red, green, dan blue menjadi citra grayscale dengan nilai  $S$ , maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai red, green, dan blue sehingga dapat dituliskan menjadi : 
$$S = \frac{Red+Green+Blue}{3}$$

Contoh citra grayscale dapat dilihat pada gambar 2.3.



**Gambar 2.3** Citra Grayscale

### 2.1.3 Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai pixel yaitu hitam dan putih, seperti pada gambar 2.4 Citra Biner. Citra B&W (*black and white*) atau citra monokrom disebut juga sebagai Citra biner. Hanya membutuhkan 1 bit untuk dapat mewakili nilai setiap pixel dari citra biner. Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti pengambangan (thresholding).

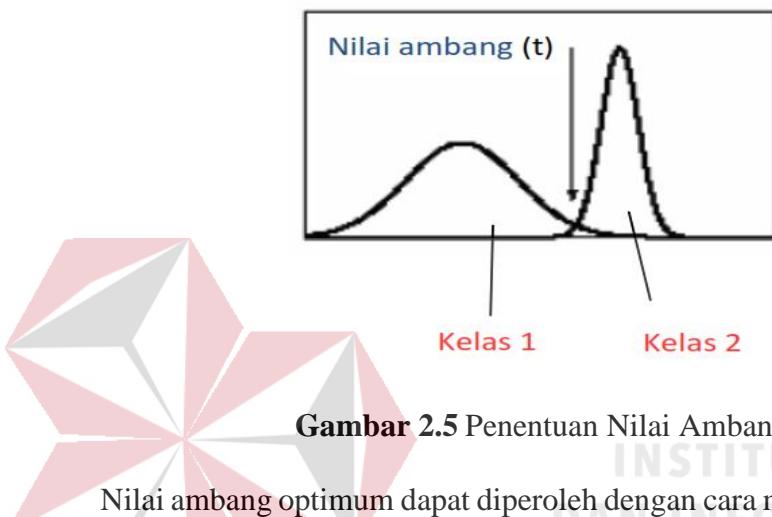


Gambar 2.4 Citra Biner

Operasi pengambangan (thresholding) mengelompokkan nilai derajat keabuannya pada setiap *pixel* ke dalam 2 kelas, hitam dan putih. Dua pendekatan yang digunakan dalam melakukan operasi pengambangan adalah pengambangan secara global dan pengambangan secara lokal. Dua pendekatan ini memiliki kelebihan dan kekurangan salah satunya adalah pada pengambangan secara global proses yang dibutuhkan sangat cepat sedangkan proses pengambangan secara local prosesnya lebih lambat bila dibandingkan dengan pengambangan secara global. Ada banyak algoritma yang digunakan untuk citra biner, salah satu di antaranya adalah menggunakan metode threshold otsu.

### 2.1.3.1 Metode Otsu

Metode Otsu pertama kali dipublikasikan oleh Nobuyuki Otsu pada tahun 1979. Metode ini menentukan nilai ambangnya dengan cara membedakan menjadi dua kelompok, yaitu objek dan latar belakang, yang memiliki bagian yang saling bertumpukan, berdasarkan histogram seperti gambar 2.5.



Nilai ambang optimum dapat diperoleh dengan cara memaksimumkan BCV. Dalam hal ini BCV disebut sebagai between-class variance. BCV dinyatakan dengan persamaan sebagai berikut :

$$\sigma_B^2 = w_1 \cdot [m_1(t) - m_t]^2 + w_2 [m_2(t) - m_t]^2 \quad (1)$$

$$m_t = \sum_{i=1}^N i \cdot p(i) \quad (2)$$

Prinsip dari metode *Otsu* adalah Pertama-tama, probabilitas nilai intensitas  $i$  dalam histogram dihitung melalui

$$p(i) = \frac{n_i}{N}, p(i) \geq 0, \sum_{i=1}^{256} p(i) = 1 \quad (3)$$

dengan  $n_i$  menyatakan jumlah piksel berintensitas  $i$  dan  $N$  menyatakan jumlah semua piksel dalam citra. Jika histogram dibagi menjadi dua kelas yaitu

objek dan latar belakang, maka pembobotan pada kedua kelas dinyatakan sebagai berikut:

$$W_1(t) = \sum_{i=1}^t p(i) \quad (4)$$

$$W_2(t) = \sum_{i=t+1}^L p(i) = 1 - W_1(t) \quad (5)$$

Dalam hal ini, L menyatakan jumlah aras keabuan. Rerata kedua kelas dihitung melalui:

$$m_1(t) = \sum_{i=1}^t i \cdot p(i) \cdot \frac{1}{w_1} \quad (6)$$

$$m_2(t) = \sum_{i=1}^t i \cdot p(i) \cdot \frac{1}{w_2} \quad (7)$$

Berdasarkan persamaan persamaan tersebut, Perhitungan dengan cara BCV memiliki keunggulan dalam menghemat proses komputasi.

## 2.2 Tanda Nomor Mobil

Tanda Nomor mobil atau sering disebut plat nomor atau nomor polisi (disingkat dengan nopol) adalah plat berbahan aluminium yang merupakan tanda dari mobil di Indonesia dan sudah terdaftar pada Kantor Bersama Samsat. Samsat (Sistem Administrasi Manunggal Satu Atap) dibentuk untuk dapat memperlancar dan mempercepat pelayanan kepentingan masyarakat yang kegiatannya dilakukan di dalam satu gedung. Contoh dari samsat adalah dalam pengurusan dokumen mobil.

### 2.2.1 Spesifikasi Teknis

Tanda Nomor Mobil berbentuk plat aluminium dengan cetakan tulisan dua baris.

- Pada baris pertama menunjukkan: kode wilayah (huruf), nomor polisi (angka), dan kode/seri akhir wilayah (huruf).
- Pada baris kedua menunjukkan bulan dan tahun masa berlaku, masing-masing dua digit (misalnya **10.19** berarti berlaku hingga **Oktober 2019**)

Bahan baku TNKB adalah aluminium dengan ketebalan 1 mm. Ukuran TNKB untuk mobil adalah  $395 \times 135$  mm. Terdapat cetakan garis lurus pembatas lebar 5 mm di antara ruang nomor polisi dengan ruang angka masa berlaku (yang lama), pada teknis baru terdapat garis putih di sekitar plat nomor dan tidak terdapat batas pemisah antara nomor polisi dan masa berlaku (dari tahun 2011).

### 2.2.2 Spesifikasi Teknis Baru

Korps Lantas Mabes Polri terhitung mulai April 2011 sudah mengganti desain pada plat nomor kendaraan. Ukurannya lebih panjang 5 centimeter daripada plat nomor sebelumnya. Perubahan ukuran plat ini dilakukan karena terdapat penambahan menjadi tiga huruf di belakang nomor (Contoh B 1932 **FJA**), sementara sebelumnya hanya terdapat dua huruf (Contoh B 1531 **AF**). Perubahan ini membuat angka dan huruf pada plat nomor berdesakan, sehingga akan sulit dibaca. Dengan diperpanjangnya ukuran plat nomor tersebut, jarak antara nomor dan huruf pada plat lebih luas sehingga mudah terbaca.

Ukuran TNKB untuk mobil adalah panjangnya 430 mm dengan lebar 135 mm. pada spesifikasi teknis baru ini plat nomor menggunakan rupa huruf (*font*) yang sama, dapat dilihat pada gambar 2.6.



**Gambar 2.6** Gambar Plat Nomor Teknis Baru.

### 2.3 Deteksi Tepi

Deteksi tepi pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek citra. Tepi-tepi ini akan menandai bagian detail citra. Tepi-tepi pada gambar tersebut terletak pada titik-titik yang memiliki perbedaan tinggi, seperti pada gambar 2.7. Dengan perbedaan tinggi tersebut tercipta suatu pola atau guratan yang membentuk suatu objek dapat diperoleh menggunakan High Pass Filter (HPF). Keberadaan tepi unsur ditandai dengan tingginya perubahan nilai piksel atau kontras.



**Gambar 2.7** Deteksi Tepi

### 2.3.1 Deteksi Tepi Canny

Canny edge detector dikembangkan oleh John F. Canny pada tahun 1986 dan menggunakan algoritma multi-tahap untuk mendekripsi berbagai tepi dalam gambar. Walaupun metode tersebut telah berumur cukup lama, namun metode ini telah menjadi metode deteksi tepi standar dan masih dipakai dalam penelitian.

#### Algoritma

Algoritma canny edge detection secara umum (detilnya tidak baku atau bisa divariasikan) beroperasi sebagai berikut :

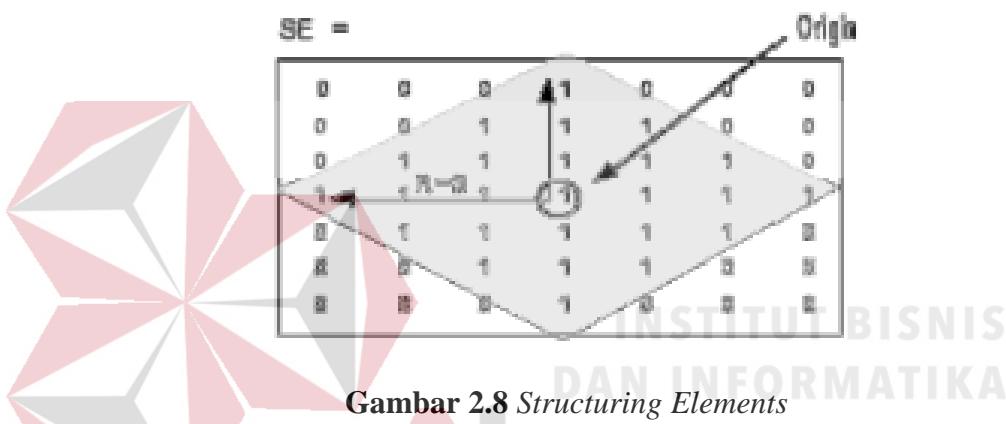
- Penghalusan untuk mengurangi dampak noise terhadap pendekripsi edge
- Menghitung potensi gradien citra
- non-maximal suppression dari gradien citra untuk melokalisasi edge secara presisi
- hysteresis thresholding untuk melakukan klasifikasi akhir

### 2.4 Morfologi

Morfologi merupakan sebuah teknik pengolahan citra digital dengan menggunakan bentuk (shape) sebagai pedoman dalam melakukan pengolahan. Pada operasi morfologi, nilai dari tiap piksel pada suatu citra keluaran didasarkan pada perbandingan piksel pada citra masukkan dengan piksel di tetangganya. Hanya menentukan ukuran dan bentuk dari neighborhood, telah dapat membangun suatu operasi morfologi yang sensitif terhadap suatu bentuk spesifik di citra masukkan. Jumlah piksel yang ditambahkan atau dihilangkan tergantung ukuran dan bentuk dari *Structuring Element* yang digunakan untuk dapat memproses citra.

### 2.4.1 Structuring Elements

*Structuring elements* (strel) dapat diibaratkan dengan mask pada proses citra biasa, *Structuring elements* dua dimensi memiliki ukuran yang biasanya jauh lebih kecil dibandingkan dengan citra yang akan diolah. Origin merupakan bagian tengah dari structuring element, *origin* mengidentifikasi piksel yang menjadi inti perhatian, dapat dilihat pada gambar 2.8.



Gambar 2.8 Structuring Elements

Pemilihan *strel* merupakan kunci penting pada operasi morfologi. *Strel* memiliki dua komponen penting yaitu bentuk dan ukuran, dua komponen ini sangat mempengaruhi hasil dari operasi morfologi. Pemilihan *structuring element* (*strel*) sangat mempengaruhi hasil pemrosesan pada citra. Penggunaan dua buah *structuring element* yang berbeda akan dapat menghasilkan hasil yang berbeda meskipun obyek/citra yang akan dianalisa adalah sama.

*Structuring element* memiliki berberapa bentuk yang biasa digunakan, yaitu *disk*, *linear*, *rectangle*, *square*, dan *diamond*. Setiap bentuk *structuring element* tersebut memiliki kelebihan dan kekurangan masing-masing.

#### 2.4.2 Dilasi

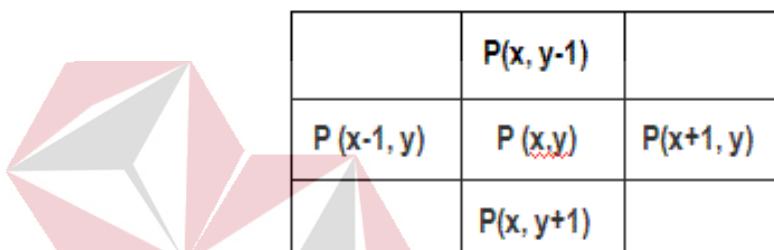
Dilasi merupakan suatu proses menambahkan piksel pada batasan dari objek dalam citra sehingga dilakukan operasi ini maka citra hasilnya lebih besar ukurannya dibandingkan dengan citra aslinya. Operasi dilasi dilakukan untuk memperbesar ukuran segmen pada objek dengan menambahkan lapisan di sekeliling objek.

Dilasi ini sangat berguna ketika diterapkan pada obyek-obyek yang terputus dikarenakan hasil pada pengambilan citra yang terganggu oleh *noise*, kerusakan obyek pada citra digital, atau disebabkan resolusi yang jelek pada citra, misalnya teks pada kertas yang agak rusak sehingga bentuk hurufnya terputus, dan sebagainya. Dengan melakukan proses dilasi maka obyek atau tepi pada citra dapat disambung kembali.

#### 2.5 Connected Component Labeling

Connected component labeling adalah teknik yang digunakan untuk mengklasifikasikan region atau objek pada citra digital. Teknik ini memanfaatkan teori connectivity piksel pada citra. Piksel-piksel dalam region disebut connected (ada koneksi antaranya atau connectivity) bila mematuhi aturan adjacency atau aturan “kedekatan” piksel. Aturan kedekatan piksel ini memanfaatkan sifat ketetanggaan piksel. Dengan demikian piksel-piksel yang dikatakan connected pada dasarnya memiliki sifat adjacency satu sama lain karena mereka masih memiliki hubungan neighbourhood atau ketetanggaan. Pada connected component labeling citra yang bisa diolah adalah citra monokrom atau citra biner. Ketetanggaan harus memiliki panjang atau jarak 1 unit (langsung antara piksel dengan piksel tanpa ada perantara

nya). Menurut Gonzales dan Woods (1992, p40), ada dua jenis konektivitas yang digunakan pada citra dua dimensi yaitu 4-Konektivitas (4-Connected Neighbors) dan 8-Konektivitas (8-Connected Neighbors), tetapi yang digunakan pada penelitian ini adalah 4-Konektivitas karena pada 4-Konektivitas jika terdapat 2 pixel yang bersinggungan secara diagonal maka akan dianggap sebagai 2 objek sedangkan pada 8-Konektivitas jika terdapat 2 pixel yang bersinggungan maka akan hanya dianggap 1 objek. 4-Konektivitas (4-Connected Neighbors)



Gambar 2.9 4-Konektivitas

Piksel-piksel yang berdekatan dikatakan memiliki hubungan 4-konektivitas jika piksel-piksel tersebut terletak berdampingan secara horizontal dan vertical N4(P). Kumpulan dari piksel-piksel ini disebut dengan 4 neighbors of P. “Pada konsep 4-Connected Neighbors bila terdapat 2 pixel yang bersinggungan secara diagonal maka akan dianggap 2 objek” .

Pada dasarnya dalam pemberian label untuk piksel yang terkoneksi, kita melakukan scanning terhadap semua piksel citra dari piksel paling atas yaitu mulai dari kiri ke kanan dan dari atas ke bawah. Tujuan dari aktifitas ini adalah untuk menemukan cluster terhadap region-region di dalam citra biner.

Dalam proses pelabelan piksel, menggunakan pendekataan 4-connected neighbors untuk menghubungkan antara piksel yang memiliki nilai sama dan menggunakan teknik flood fill untuk mengisi warna acak pada piksel terpilih ke sebuah matrix penampung yang biasa disebut matrix mapping. Sebelum melakukan proses scanning dan labeling, terlebih dahulu harus mendefenisikan bahwa objek gambar/piksel yang akan di ambil adalah tipe foreground dan yang ditinggalkan merupakan background. Adapun langkah – langkah dalam proses scanning, labeling, dan merging adalah sebagai berikut : Pertama membuat dua buah matrix, matrix pertama digunakan untuk merepresentasikan objek gambar/citra biner yang akan diolah sedangkan matrix kedua digunakan untuk tempat meletakkan piksel-piksel terpilih yang disebut dengan matrix mapping.



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

Matrix Original

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Matrix Mapping

**Gambar 2.10** Matrix Original dan Matrix Mapping.

Setelah membuat dua buah matrix maka langkah selanjutnya yaitu melakukan scanning piksel-piksel terhadap citra foreground yang dimulai dari sisi atas matrix yaitu dari kiri ke kanan dan dari atas ke bawah.

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

 To its right is an 8x8 matrix of all 0s."/>

**Gambar 2.11** Scanning Piksel.

Jika ditemukan piksel foreground, maka lakukan pelabelan terhadap piksel tersebut, sebagai contoh label 1 (satu) dan pindahkan piksel pertama ke matrix mapping. Selanjutnya ubah nilai piksel pertama pada matrix original yang telah ditemukan tadi dengan nilai 0.

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

 To its right is an 8x8 matrix of all 0s. Below this is another 7x8 matrix where the red '1' is now at position (3,1). The matrix is as follows:
 

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

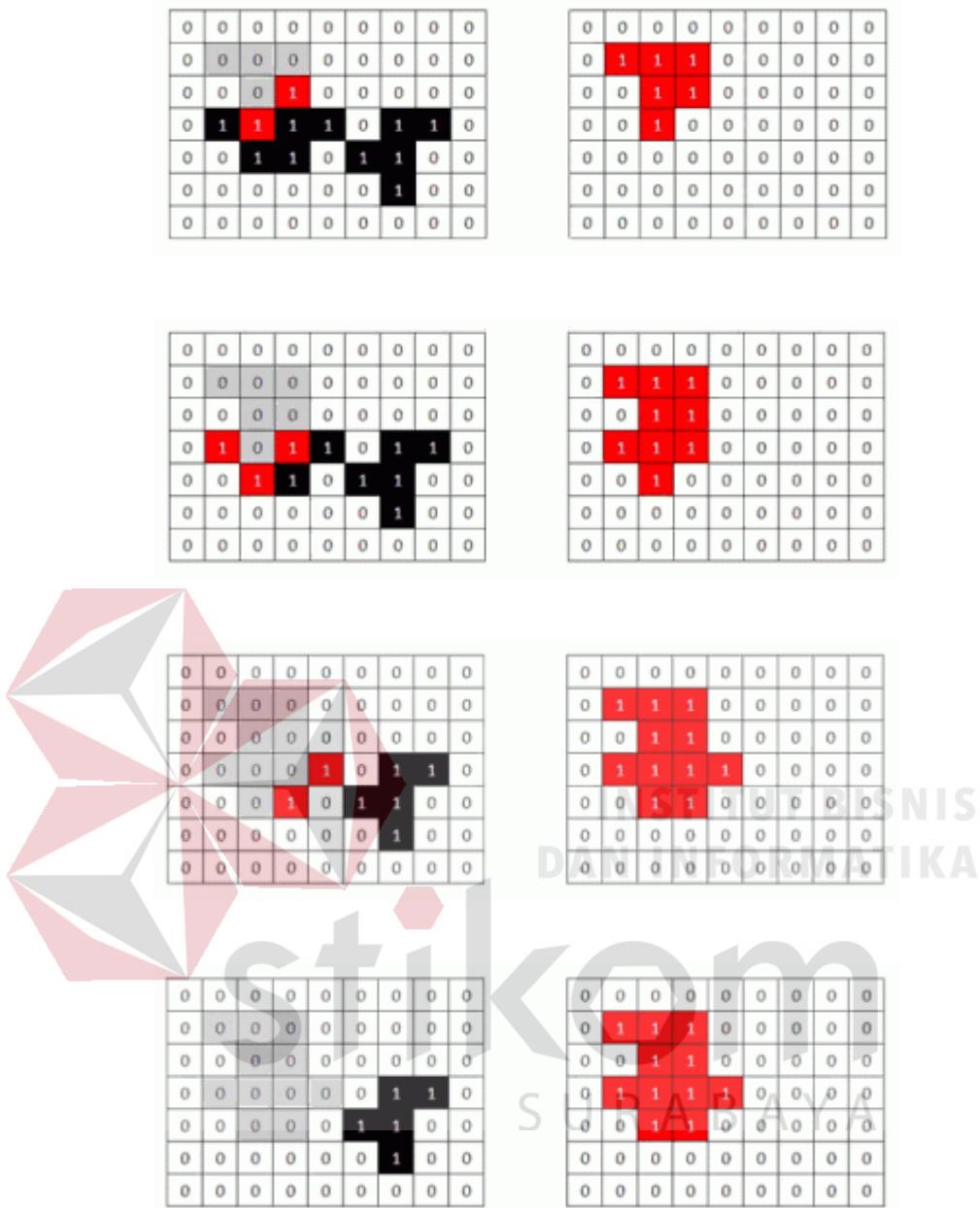
 To its right is an 8x8 matrix of all 0s."/>


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

 To its right is an 8x8 matrix of all 0s."/>


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

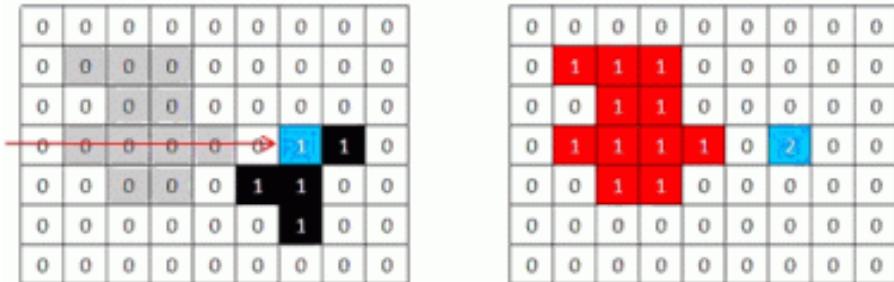
 To its right is an 8x8 matrix of all 0s."/>



**Gambar 2.12** Melabeli dengan Label (1).

Jika sudah tidak ditemukan lagi piksel-piksel tetangga yang terdapat kedekatan secara 4-connected neighbors, maka lakukan proses merging pada matrix mapping.

Langkah selanjutnya yaitu melakukan lagi scanning terhadap citra foreground untuk mendapatkan karakter objek yang lain.

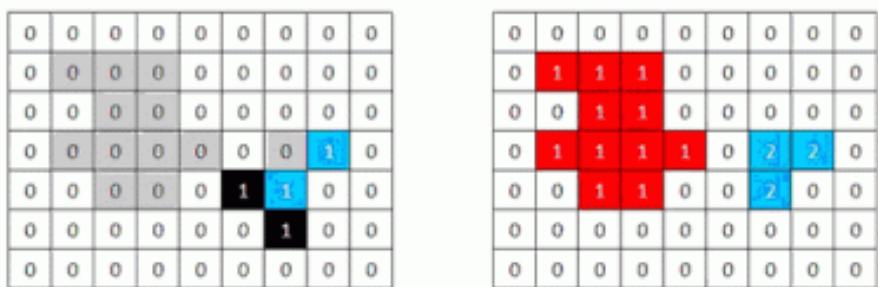


The diagram illustrates a 5x5 input matrix (left) and its corresponding 8x8 output matrix (right) after a scanning operation. The input matrix contains a 3x3 block of value 1 (foreground) in the center. The output matrix shows the result of the scan, with the value 2 appearing in the 4th column of the 4th row, indicating the result of the scan operation.

**Gambar 2.13** Scanning Untuk Objek Lain

Jika ditemukan piksel foreground, maka lakukan pelabelan terhadap piksel tersebut, sebagai contoh label 2 (dua) dan pindahkan piksel pertama ke matrix mapping. Selanjutnya ubah nilai piksel pertama pada matrix original yang telah ditemukan tadi dengan nilai 0.

Langkah selanjutnya yaitu melakukan pendekatan 4-connected neighbors secara berulang-ulang terhadap piksel-piksel yang memiliki kedekatan dan kesamaan nilai intensitas sampai tidak ada lagi kedekatan secara 4-connected neighbors antara piksel-piksel yang telah dilabeli dengan label (2) dua.



The diagram illustrates a 5x5 input matrix (left) and its corresponding 8x8 output matrix (right) after an iterative 4-connected neighbor processing operation. The input matrix contains a 3x3 block of value 1 (foreground) in the center. The output matrix shows the result of the iterative process, with the value 2 appearing in the 4th column of the 4th row, indicating the result of the iterative scan operation.

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

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

**Gambar 2.14** Melabeli dengan Label (2).

Jika sudah tidak ditemukan lagi piksel-piksel tetangga yang terdapat kedekatan secara 4-connected neighbors, maka lakukan proses merging pada matrix mapping.

Periksa kembali piksel pada matrix original dengan melakukan pelacakan atau scanning piksel – piksel citra foreground, jika tidak ditemukan, maka selesailah proses scanning, labeling dan merging.

gambar hasil akhir bisa dilihat pada gambar 2.15.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	6	0	0	0
0	0	1	1	0	0	0	0	0	0
0	1	1	1	1	0	2	2	0	0
0	0	1	1	0	2	2	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	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

**Gambar 2.15** Hasil Akhir Connected Component Labeling

## 2.6 Cropping

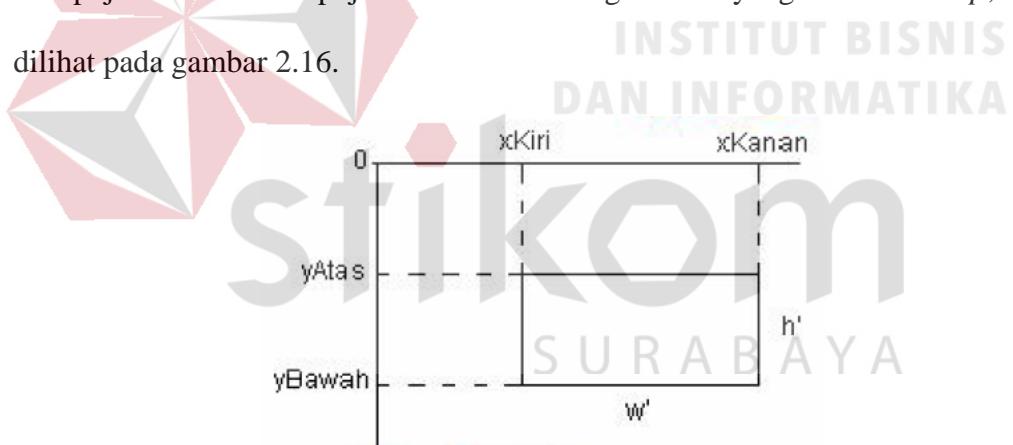
*Cropping* adalah memotong satu bagian pada citra sehingga memperoleh hasil citra yang berukuran lebih kecil. Terdapat dua koordinat untuk memotong bagian dari citra , koordinat awal merupakan titik awal citra hasil pemotongan dan koordinat akhir adalah titik akhir citra hasil potongan,

Rumus yang digunakan untuk menggunakan operasi ini adalah sebagai berikut (Ahmad, 2004:172 (dalam tugas akhir Oksatana, Miga. 2009) ):

$x' = x - xKiri$  untuk  $x = xKiri$  sampai  $xKanan$

$y' = y - yAtas$  untuk  $y = xAtas$  sampai  $yBawah$

(xKiri, yAtas) dan (xKanan, yBawah) masing-masing merupakan koordinat titik pojok kiri atas dan pojok kanan bawah bagian citra yang hendak dicrop, dapat dilihat pada gambar 2.16.



**Gambar 2.16** Koordinat Titik Pojok Bagian Citra Yang Di-Crop.

Maka ukuran citra berubah menjadi ( Achmad, 2004: 173) :

Dan transformasi baliknya adalah :

$x = x' + x_{\text{Kiri}}$  untuk  $x' = 0$  sampai  $w' - 1$

$y = y' + y$  Atas untuk  $y' = 0$  sampai  $h' = 1$

## 2.7 Microsoft Visual Studio

Visual Studio 2013 dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu.

Visual Studio 2013 (yang sering juga disebut dengan VB .Net 2013) selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasis windows. Beberapa kemampuan atau manfaat dari Visual Studio 2013 diantaranya seperti :

- Untuk membuat program aplikasi berbasis windows.
- Untuk membuat objek - objek pembantu program misalnya seperti : kontrol ActiveX, file Help, aplikasi Internet dan sebagainya.
- Menguji program (debugging) dan menghasilkan program berakhiran EXE yang bersifat executable atau dapat langsung dijalankan.

Visual Studio 2013 adalah bahasa yang cukup mudah untuk dipelajari. Bagi programer pemula yang baru ingin belajar program, lingkungan Visual Studio dapat membantu membuat program dalam sekejap mata. Sedang bagi programer tingkat lanjut, kemampuan yang besar dapat digunakan untuk membuat program-program yang kompleks, misalnya lingkungan net-working atau client server.

Bahasa Visual Studio cukup sederhana dan menggunakan kata-kata bahasa Inggris yang umum digunakan. Kita tidak perlu lagi menghafalkan sintaks-sintaks maupun format-format bahasa yang bermacam-macam, di dalam Visual Basic semuanya sudah disediakan dalam pilihan-pilihan yang tinggal diambil sesuai dengan kebutuhan. Selain itu, sarana pengembangannya yang bersifat visual

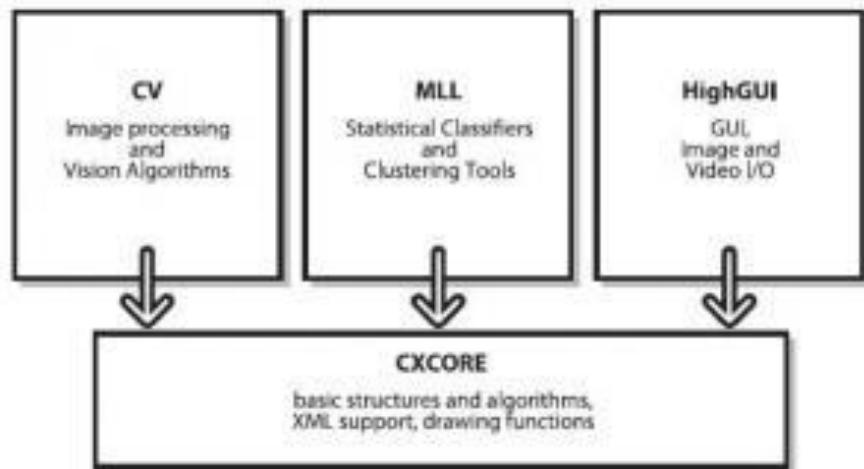
memudahkan kita untuk mengembangkan aplikasi berbasiskan Windows, bersifat mouse-driven (digerakkan dengan mouse) dan berdaya guna tinggi.

## 2.8 OpenCV

OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) Library yang sudah sangat familiar pada Pengolahan Citra *Computer Vision*. *Computer Vision* merupakan salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan computer untuk dapat melihat seperti manusia. Beberapa pengimplementasian dari *Computer Vision* adalah *Face/Object Tracking*, *Road Tracking*, *Face Recognition*, *Face Detection*, *dll.*

OpenCV adalah library *Open Source* untuk Computer Vision untuk C/C++, seperti yang terlihat pada gambar 2.17. OpenCV didesain untuk aplikasi real-time, seperti image/video. OpenCV terdiri dari 5 library, sebagai berikut :

1. CV : untuk algoritma Image processing dan Vision.
2. ML : untuk machine learning library
3. Highgui : untuk GUI, Image dan Video I/O.
4. CXCORE : untuk struktur data, support XML dan fungsi-fungsi grafis.
5. CvAux



Gambar 2.17 Struktur dan Konten OpenCV

