

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tanda Nomor Kendaraan Bermotor**

Tanda Nomor Kendaraan Bermotor (disingkat TNKB) atau sering disebut plat nomor atau nomor polisi (disingkat nopol) adalah plat aluminium tanda kendaraan bermotor di Indonesia yang telah didaftarkan pada Kantor Bersama Samsat. Samsat (Sistem Administrasi Manunggal Satu Atap) adalah suatu sistem administrasi yang dibentuk untuk memperlancar dan mempercepat pelayanan kepentingan masyarakat yang kegiatannya diselenggarakan dalam satu gedung. Contoh dari samsat adalah dalam pengurusan dokumen kendaraan bermotor.

##### **2.1.1 Spesifikasi Teknis**

Tanda Nomor Kendaraan Bermotor berbentuk plat aluminium dengan cetakan tulisan dua baris.

- Baris pertama menunjukkan: kode wilayah (huruf), nomor polisi (angka), dan kode/seri akhir wilayah (huruf).
- Baris kedua menunjukkan bulan dan tahun masa berlaku, masing-masing dua digit (misalnya **01.20** berarti berlaku hingga **Januari 2020**)

Bahan baku TNKB adalah aluminium dengan ketebalan 1 mm. Ukuran TNKB untuk kendaraan bermotor roda 2 dan roda 3 adalah 250×105 mm, sedangkan untuk kendaraan bermotor roda 4 atau lebih adalah 395×135 mm. Terdapat cetakan garis lurus pembatas lebar 5 mm di antara ruang nomor polisi dengan ruang angka masa berlaku (yang lama), sedangkan yang baru

terdapat garis putih di sekitar TNKB dan tidak ada batas pemisah antara nomor polisi dan masa berlaku (dari tahun 2011).

### 2.1.2 Spesifikasi Teknis Baru

Korps Lantas Mabes Polri terhitung mulai April 2011 mengganti desain plat nomor kendaraan. Ukurannya lebih panjang 5 centimeter daripada plat nomor sebelumnya. Perubahan ukuran plat dilakukan karena ada penambahan menjadi tiga huruf di belakang nomor (Contoh B 2684 SFJ), sementara sebelumnya hanya dua huruf (Contoh B 1090 CA). Perubahan ini membuat angka dan huruf pada plat nomor berdesakan, sehingga sulit dibaca. Dengan diperpanjangnya plat tersebut, jarak antara nomor dan huruf pada plat lebih luas sehingga mudah terbaca.

Ukuran TNKB untuk kendaraan roda 2 dan 3 sekarang menjadi 275 mm dengan lebar 110 mm, sedangkan untuk kendaraan roda 4 atau lebih adalah panjang 430 mm dengan lebar 135 mm. Sementara ini, plat resmi yang lama masih berlaku (apalagi terkadang sejumlah Samsat di berbagai daerah sering memanfaatkan plat jenis lama untuk kendaraan yang plat nomornya diperpanjang setelah tahun 2011). Selain itu, pada spesifikasi teknis baru ini plat nomor menggunakan rupa huruf (*font*) yang sama.



**Gambar 2.1** Gambar Citra Plat nomor.

## 2.2. Citra

Satuan atau bagian terkecil dari suatu citra disebut piksel (*pixel* atau *picture element*) yang berarti elemen citra. Piksel merupakan sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Umumnya Citra dibentuk dari kotak-kotak persegi empat yang teratur sehingga jarak horizontal dan vertikal antar piksel adalah sama pada seluruh bagian citra (Putra Darma : 2010).

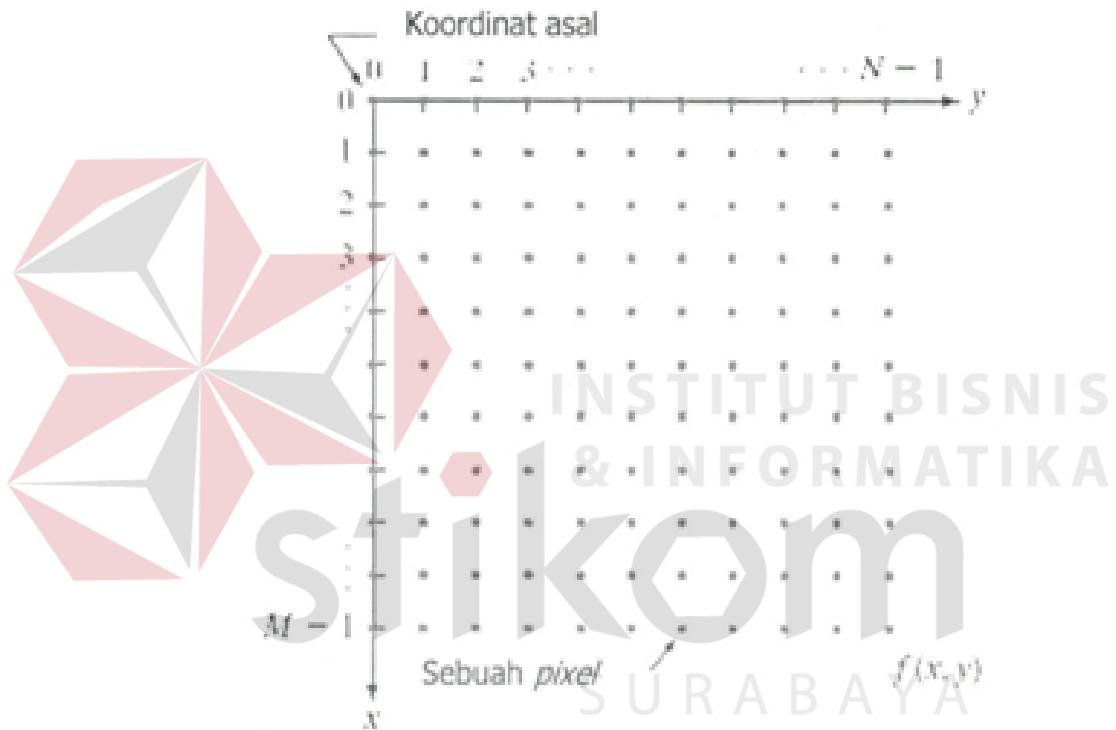
Dalam komputer, setiap piksel diwakili oleh dua buah bilangan bulat (*integer*) untuk menunjukkan lokasinya dalam bidang citra dan sebuah nilai dalam bilangan bulat untuk menunjukkan cahaya atau terang-gelap piksel tersebut. Untuk menunjukkan lokasi suatu piksel, koordinat (0,0) digunakan untuk posisi kiri atas dalam bidang citra, dan koordinat (m-1,n-1) digunakan untuk posisi kanan bawah dalam citra berukuran mxn piksel. Untuk menunjukkan tingkat pencahayaan suatu piksel, seringkali digunakan bilangan bulat yang besarnya 8-bit, dengan lebar selang nilai 0 – 255, di mana 0 untuk warna hitam, 255 untuk warna putih dan tingkat abu-abu berada diantara nilai-nilai 0 dan 255.

Dapat juga dinyatakan bahwa Citra adalah kumpulan piksel-piksel yang disusun dalam larik dua-dimensi.

## 2.3. Pengolahan Citra Digital

Secara umum, istilah pengolahan citra digital menyatakan “pemrosesan gambar berdimensi-dua melalui komputer digital” Menurut Efford (2000), pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara.

Suatu citra dapat didefinisikan sebagai fungsi  $f(x, y)$  berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo  $f$  di titik koordinat  $(x, y)$  dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x, y, dan nilai amplitudo  $f$  secara keseluruhan berhingga (*finite*) dan bernilai diskrit, maka dapat dikatakan bahwa citra tersebut adalah citra digital.



**Gambar 2.2** Koordinat Citra Digital

Citra digital dapat ditulis dalam bentuk matrik 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 dan kolom (pada posisi x, y) disebut dengan *picture elements*, *image elements*, *pels*, atau *pixels*. Istilah terakhir (pixel) paling sering digunakan pada citra digital.

#### 2.4. Prinsip Dasar Dalam Pengolahan Citra

Berikut prinsip dasar dalam pengolahan citra (Putra, Darma, 2010):

- Peningkatan kecerahan dan kontras

Seringkali dijumpai citra yang tidak jelas akibat kekurangan sinar ketika objek dibidik melalui kamera digital. Dengan menggunakan pengolahan citra, citra seperti itu dapat diperbaiki melalui peningkatan kecerahan dan kontras.

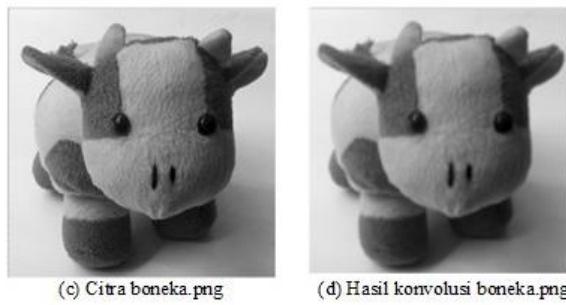


Gambar 2.3 Perubahan kontras citra

- Penghilangan Derau

Citra yang diproses seringkali dalam keadaan terdistorsi atau mengandung derau, untuk kepentingan tertentu derau tersebut perlu dibersihkan terlebih dahulu. Dalam pengolahan citra metode yang dapat dipakai untuk keperluan tersebut salah satu cara yang digunakan adalah Filter Gaussian. Filter Gaussian adalah salah satu filter linear dengan nilai pembobotan untuk setiap anggotanya dipilih berdasarkan bentuk fungsi gaussian. Filter ini sangat baik untuk menghilangkan *noise* yang bersifat sebaran normal, yang banyak dijumpai pada citra hasil proses digitalisasi menggunakan kamera karena

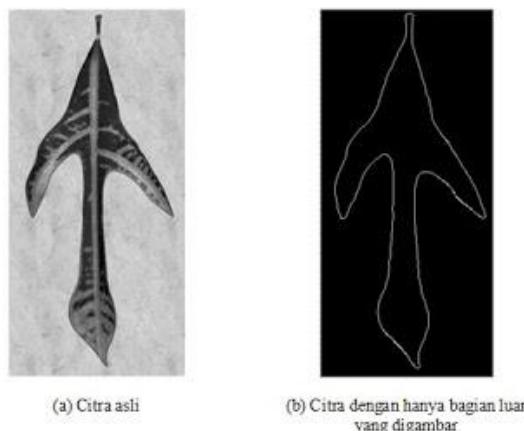
merupakan fenomena alamiah akibat sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri. Gaussian filter ini sangat membantu dalam meminimali noise yang ada pada gambar.



**Gambar 2.4** Menghilangkan derau pada citra

- Pencarian Bentuk Objek

Untuk kepentingan mengenali suatu objek di dalam citra, objek perlu dipisahkan terlebih dahulu dari latar belakangnya. Salah satu pendekatan yang umum dipakai untuk keperluan ini adalah penemuan batas objek. Dalam hal ini batas objek berupa bagian tepi objek. Setelah tepi objek diketahui pencarian ciri terhadap objek dapat dilaksanakan, misalnya berdasarkan perbandingan panjang dan lebar daun.



**Gambar 2.5** Objek daun dan batas daun yang didapatkan

## 2.5. Jenis Citra

Nilai suatu pixel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. namun secara umum jangkauannya adalah 0 – 255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer. Berikut adalah jenis-jenis citra berdasarkan nilai pixelnya :

### 2.5.1 Citra Biner

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



**Gambar 2.6** Citra Biner

Operasi pengambangan mengelompokkan nilai derajat keabuan setiap *pixel* ke dalam 2 kelas, hitam dan putih. Dua pendekatan yang digunakan dalam operasi pengambangan adalah pengambangan secara global dan pengambangan secara lokal.

### 2.5.2 Citra *Grayscale*

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixel-nya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan di sini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Citra *grayscale* berikut memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan).

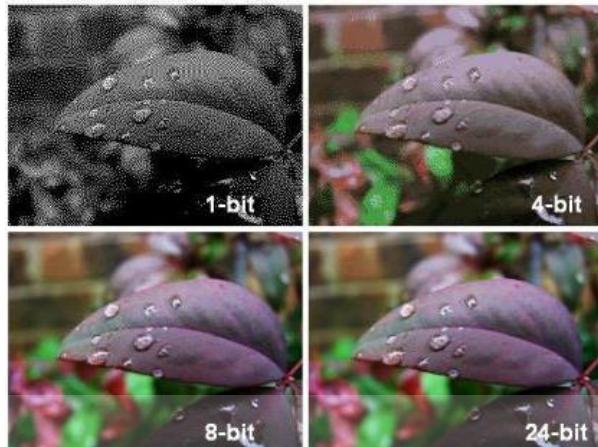


**Gambar 2.7** Citra *Grayscale*

### 2.5.3 Citra Warna ( 24 bit )

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

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



**Gambar 2.8** Citra Warna

## 2.6 Morphological Operation

Ada dua operasi dasar morphologi yaitu dilasi dan erosi. Kedua operasi dasar tersebut menjadi basis untuk membuat berbagai operasi morphologi yang sangat berguna untuk pengolahan citra digital seperti opening, closing, hit and miss transform, thinning dan thickening. (<http://hobbycoding.web.id>, 2012)

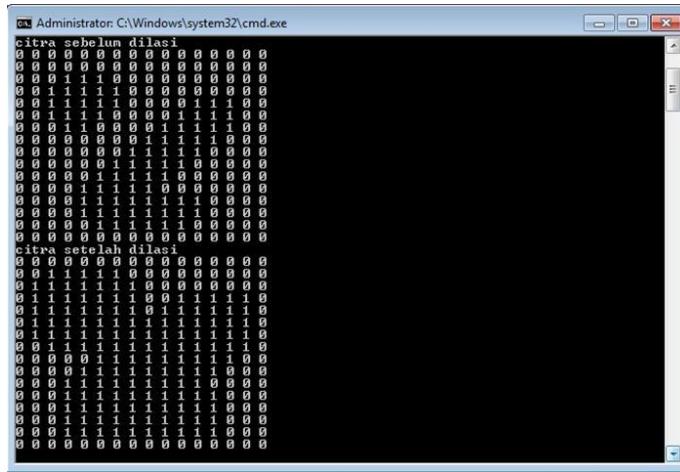
## 2.6.1 Dilasi

Bila suatu objek (citra input) dinyatakan dengan A dan *Structuring Element* (SE) dinyatakan dengan B, seta  $B_x$  menyatakan translasi B sedemikian sehingga pusat B terletak pada x. Operasi dilasi A dengan SE B dapat dinyatakan sebagai berikut :

$$D(A, B) = A \oplus B = \{x : B_x \cap A \neq \emptyset\} \dots \dots \dots \quad (2)$$

Dengan  $\emptyset$  menyatakan himpunan kosong.

Berikut gambar 2.9 merupakan contoh operasi dilasi pada citra biner.



```
Administrator: C:\Windows\system32\cmd.exe
citra sebelum dilasi
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 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 0 0 0 0 0 1 1 0 0
0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 0 0
0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0
0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 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
citra setelah dilasi
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0
0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

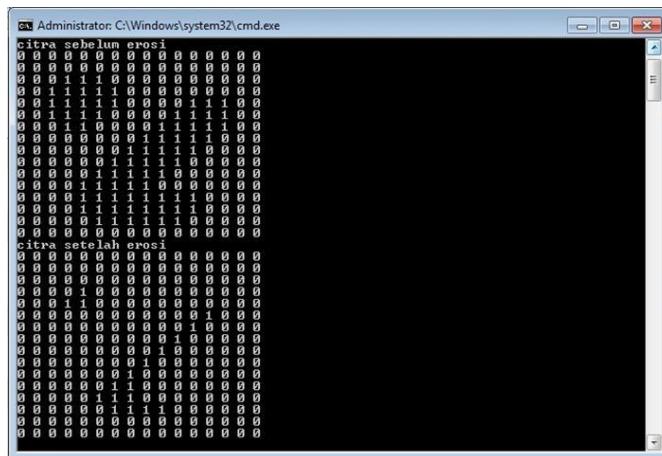
Gambar 2.9 contoh proses dilasi

## 2.6.2 Erosi

Erosi adalah proses kebalikan dari dilasi. Operasi dilasi A dengan SE B dapat dinyatakan sebagai berikut :

$$E(A, B) = A \ominus B = \{x: B_x \subset X\} \dots\dots\dots (3)$$

Gambar 2.10 adalah contoh proses erosi pada citra biner.



```
Administrator: C:\Windows\system32\cmd.exe
citra sebelum erosi
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 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0
0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
citra setelah erosi
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
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 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
```

Gambar 2.10 contoh proses erosi

## 2.7 Cropping

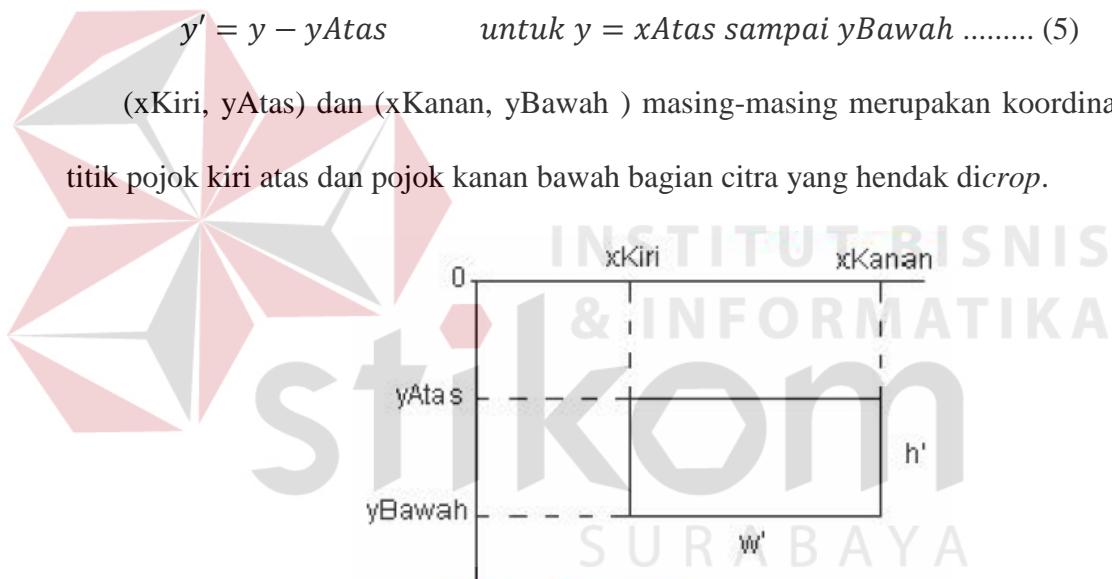
*Cropping* adalah memotong satu bagian dari citra sehingga diperoleh citra yang berukuran lebih kecil. Proses ini bertujuan untuk memisahkan objek yang satu dengan objek yang lain pada karakter angka ataupun huruf pada plat nomor atau biasa disebut segmentasi citra.

Rumus yang digunakan untuk menggunakan operasi ini adalah sebagai berikut (Ahmad, 2004:172) :

$$x' = x - xKiri \quad \text{untuk } x = xKiri \text{ sampai } xKanan \dots\dots\dots (4)$$

$$y' = y - yAtas \quad \text{untuk } y = yAtas \text{ sampai } yBawah \dots\dots\dots (5)$$

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



**Gambar 2.11** Koordinat titik pojok bagian citra yang di-crop.

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

$$h' = yBawah - yAtas \dots\dots\dots (2)$$

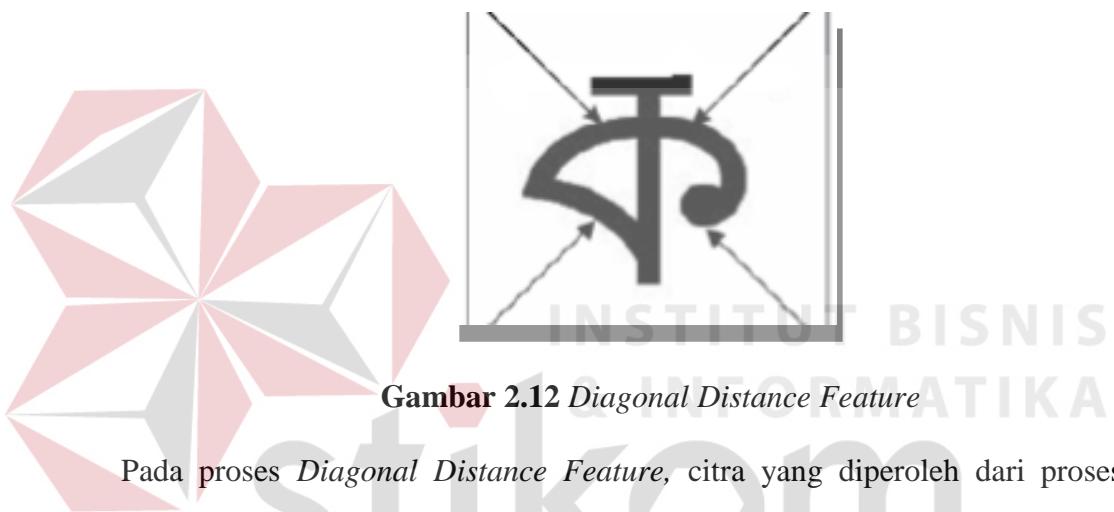
Dan transformasi baliknya adalah :

$$x = x' + xKiri \quad \text{untuk } x' = 0 \text{ sampai } w' - 1 \dots\dots\dots (6)$$

$$y = y' + yAtas \quad \text{untuk } y' = 0 \text{ sampai } h' - 1 \dots\dots\dots (7)$$

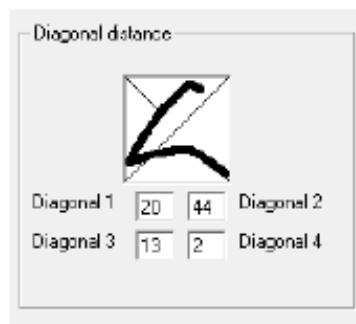
## 2.8 Diagonal Distance Feature

*Diagonal distance feature* adalah metode ekstraksi fitur yang digunakan untuk mengenali suatu pola tulisan angka, atau symbol pada citra dengan cara menghitung jarak sudut diagonal hingga bertemu piksel hitam pada citra (Nibaran das dkk, 2006). Citra yang dapat digunakan dalam metode ini hanya citra biner, oleh karena itu perlu dilakukan konversi citra menjadi biner apabila tidak sesuai. Seperti diilustrasikan pada gambar 2.12.



Pada proses *Diagonal Distance Feature*, citra yang diperoleh dari proses *cropping* akan dilakukan penghitungan jarak di mulai dari sudut siku gambar dan berjalan terus secara diagonal hingga bertemu piksel hitam. Pada ekstraksi ini jaraknya adalah piksel, Citra akan dilakukan perhitungan ke 4 diagonalnya, Diagonal 1 dicari dengan menaikkan satu koordinat (x,y) secara terus menerus hingga mencapai koordinat yang memiliki nilai warna piksel hitam. Setiap nilai koordinat (x,y) dinaikkan satu maka saat itu dilakukan juga perhitungan (*counter*) jarak diagonal 1. Diagonal 2 dicari dengan menaikkan satu koordinat (y) dan menurunkan 1 nilai (x) secara terus menerus sehingga mencapai koordinat yang memiliki warna piksel hitam. Diagonal 3 dicari dengan menaikkan satu koordinat (x) dan menurunkan 1 nilai (y) secara secara terus menerus sehingga mencapai

koordinat yang memiliki warna piksel hitam. Diagonal 4 dicari dengan menurunkan satu koordinat (x,y) secara secara terus menerus sehingga mencapai koordinat yang memiliki warna piksel hitam.



**Gambar 2.13** Pencarian nilai tiap diagonal

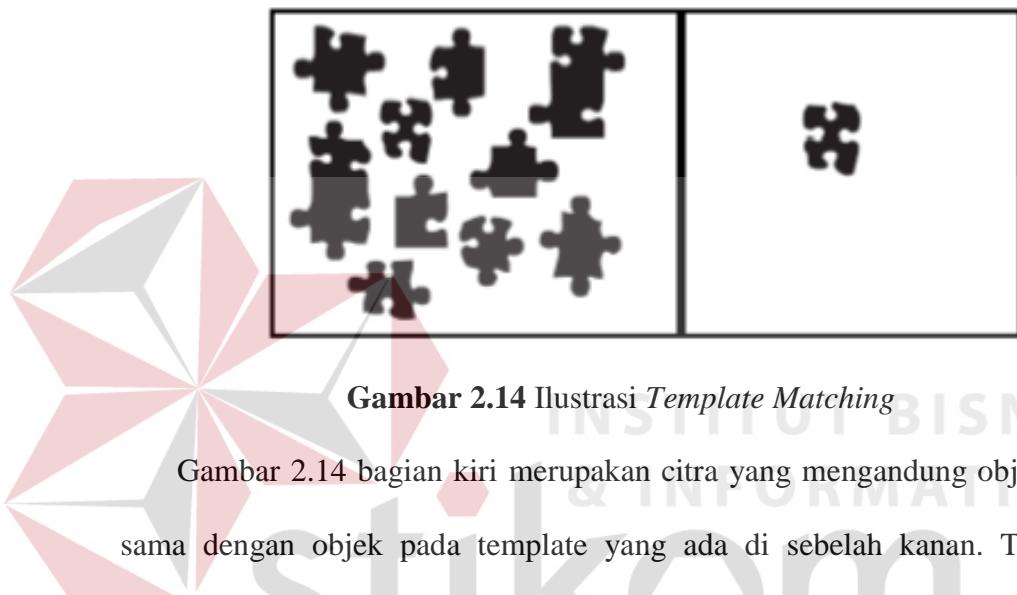
### 2.9 OCR (*Optical Character Recognition*)

*Optical character recognition* (OCR) adalah sebuah sistem komputer yang dapat membaca huruf, baik yang berasal dari sebuah pencetak (printer atau mesin ketik) maupun yang berasal dari tulisan tangan. OCR adalah aplikasi yang menerjemahkan gambar karakter (*image character*) menjadi bentuk teks dengan cara menyesuaikan pola karakter per baris dengan pola yang telah tersimpan dalam database aplikasi. Hasil dari proses OCR adalah berupa teks sesuai dengan gambar *output scanner* dimana tingkat keakuratan penerjemahan karakter tergantung dari tingkat kejelasan gambar dan metode yang digunakan (Darma Putra : 2010).

### 2.10 Template Matching

Pada dasarnya template matching adalah proses yang sederhana. Suatu citra masukan yang mengandung template tertentu dibandingkan dengan template pada basis data. Template ditempatkan pada pusat bagian citra yang

akan dibandingkan dan dihitung seberapa banyak titik yang paling sesuai dengan template. Langkah ini diulangi terhadap keseluruhan citra masukan yang akan dibandingkan. Nilai kesesuaian titik yang paling besar antara citra masukan dan citra template menandakan bahwa template tersebut merupakan citra template yang paling sesuai dengan citra masukan (Rendra P, dkk : 2013).



**Gambar 2.14** Ilustrasi *Template Matching*

Gambar 2.14 bagian kiri merupakan citra yang mengandung objek yang sama dengan objek pada template yang ada di sebelah kanan. Template diposisikan pada citra yang akan dibandingkan dan dihitung derajat kesesuaian pola pada citra masukan dengan pola pada citra template. Tingkat kesesuaian antara citra masukan dan citra template bisa dihitung berdasarkan nilai error terkecil dengan menggunakan persamaan.

I adalah pola pixel citra masukan yang akan dibandingkan. T adalah pola pixel citra template. Template dengan nilai eror paling kecil adalah template yang paling sesuai dengan citra masukan yang akan dibandingkan.

Ukuran objek yang beragam bisa diatasi dengan menggunakan template berbagai ukuran. Namun hal ini membutuhkan tambahan ruang penyimpanan.

Penambahan template dengan berbagai ukuran akan membutuhkan komputasi yang besar. Jika suatu template berukuran persegi dengan ukuran  $m \times m$  dan sesuai dengan citra yang berukuran  $N \times N$ , dan dimisalkan pixel  $m^2$  sesuai dengan semua titik citra, maka komputasi yang harus dilakukan adalah  $O(N^2m^2)$ .

Komputasi tersebut harus dilakukan dengan template yang tidak beragam. Jika parameter template bertambah, seperti ukuran template yang beragam, maka komputasi yang dilakukan juga akan bertambah. Hal ini yang menyebabkan metode template matching menjadi lamban.

## 2.11 Microsoft Visual Studio

Visual Studio 2010 pada 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 2010 (yang sering juga disebut dengan VB .Net 2010) selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasiskan windows. Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti :

- Untuk membuat program aplikasi berbasiskan windows.
- Untuk membuat objek-objek pembantu program seperti, misalnya : 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 2010 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 networking 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.

## 2.12 OpenCV

Dikutip dari situs resminya, OpenCV (*Open Source Computer Vision Library*) adalah sebuah *library* perangkat lunak *open source computer vision* dan *machine learning*. OpenCV dirilis dibawah lisensi BSD oleh karena itu disediakan gratis baik untuk penggunaan akademis dan komersial. OpenCV merupakan *library* yang sering digunakan untuk keperluan akademis baik tesis maupun disertasi.

OpenCV memiliki *interface* C++, C, Python dan Java dan dapat digunakan pada operating system Windows, Linux, Mac OS, IOS dan Android. OpenCV dirancang untuk hal efisiensi komputasi dan berfokus pada aplikasi *realtime*. OpenCV memiliki lebih dari 47.000 pengguna yang tergabung dalam komunitas dan perkiraan jumlah download melebihi 9.000.000.

OpenCV dibangun untuk menyediakan infrastruktur umum untuk aplikasi *computer vision* dan untuk mempercepat penggunaan *machine perception* dalam produk-produk komersial. Didalam OpenCV terdapat lebih dari 2.500 algoritma yang mencakup sekumpulan algoritma *computer vision* dan *machine learning* baik klasik maupun lanjutan. Algoritma-algoritma itu dapat digunakan untuk mendeteksi dan mengenali wajah, identifikasi objek, ekstrasi objek model 3D, dan lain sebagainya ([opencv.org](http://opencv.org)).



**Gambar 2.15** Logo OpenCV