

*SKRIPSI*

**DETEKSI KARBON NANO-DOT (C-DOTS) DALAM CITRA SEM  
MENGUNAKAN *SUPPORT VECTOR MACHINE* (SVM)**

**INDRA SETIAWAN**

**H021 18 1329**



**DEPARTEMEN FISIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR  
2023**

**DETEKSI KARBON NANO-DOT (C-DOTS) DALAM CITRA SEM  
MENGUNAKAN *SUPPORT VECTOR MACHINE* (SVM)**

**SKRIPSI**

*Diajukan Sebagai Salah Satu  
Memperoleh Gelar Sarjana Sains  
Pada Program Studi Fisika Departemen Fisika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Hasanuddin*

**INDRA SETIAWAN**

**H021181329**

**DEPARTEMEN FISIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR  
2023**

HALAMAN PENGESAHAN

DETEKSI KARBON NANO-DOT (C-DOTS) DALAM CITRA SEM  
MENGUNAKAN *SUPPORT VECTOR MACHINE* (SVM)

Disusun dan diajukan oleh:

**INDRA SETIAWAN**

**H021181329**

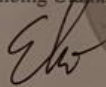
Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka  
Penyelesaian Studi Program Sarjana Program Studi Fisika  
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

Pada 07 Juni 2023

Dan dinyatakan telah memenuhi syarat kelulusan.

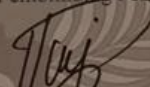
Menyetujui,

Pembimbing Utama



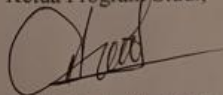
Eko Juarlin, S.Si, M.Si  
NIP. 19811106 200812 1 002

Pembimbing Pertama



Hervanto, S.Si, M.Si  
NIP. 19911129 202005 3 001

Ketua Program Studi,



Prof. Dr. Arifin, M.T  
NIP. 19670520 199403 1 002

## PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : INDRA SETIAWAN

NIM : H021181329

Program Studi : FISIKA

Jenjang : S1

Menyatakan dengan sebenarnya bahwa karya tulisan saya yang berjudul:

### **DETEKSI KARBON NANO-DOT (C-DOTS) DALAM CITRA SEM MENGUNAKAN *SUPPORT VECTOR MACHINE* (SVM)**

Adalah karya tulis berdasarkan hasil pemikiran dan penelitian saya, bukan merupakan hasil pengambil alihan tulisan maupun pemikiran orang lain. Jika terdapat karya orang lain dalam skripsi ini, maka akan dicantumkan sumber yang benar dan jelas. Demikian surat pernyataan ini saya buat dengan sebenarnya, jika dikemudian hari terdapat ketidakbenaran dan penyimpangan dalam pernyataan ini, maka saya berhak menerima sanksi atas perbuatan tersebut.

Makassar, 07 Juni 2023

Menyatakan



**Indra Setiawan**

**H021181329**

## KATA PENGANTAR

Alhamdulillah Rabiil'alamiin puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat dan hidayah nya, berupa nikmat kesehatan dan kesempatan sehingga penulis berhasil menyelesaikan penulisan skripsi dengan judul “**Deteksi karbon Nano-dot (C-Dots) Dalam Citra SEM Menggunakan Support Vector Machine (SVM)**”. Berbagai upaya telah dilakukan penulis untuk menyelesaikan penulisan skripsi ini sebagai salah satu syarat untuk menyelesaikan studi dan memperoleh gelar sarjana di program studi Fisika, Fakultas matematika dan ilmu pengetahuan alam Universitas Hasanuddin.

Penulis menyadari bahwa dalam penulisan skripsi ini banyak kesulitan dan hambatan yang dihadapi dan tidak terlepas dari dukungan berbagai pihak sehingga penulisan skripsi ini masih jauh dari kata sempurna. Namun atas kehendak nya hambatan tersebut berhasil dilalui oleh penulis sehingga penyusunan skripsi ini dapat diselesaikan. Oleh karena itu, dengan segala kerendahan hati, penulis mengucapkan banyak terimakasih kepada:

1. Bapak **Eko Juarlin, S.Si,M.Si** selaku pembimbing utama dan Bapak **Heryanto, S.Si,M.Si** selaku pembimbing pertama yang telah meluangkan banyak waktu dan tenaga nya dalam memberikan bimbingan, memberikan arahan, memberikan dukungan serta motivasi dan kepercayaan kepada penulis untuk berpikir kritis dan logis dalam melaksanakan penelitian dan penyusunan skripsi.
2. **Prof. Dr. rer-net Wira Bahari Nurdin** dan Bapak **Bannu, S.Si, M.Si.** selaku dosen penguji yang telah banyak meluangkan waktu dan tenaga nya untuk memberikan masukan dan saran yang membangun terkait penelitian dan kesempurnaan skripsi ini.
3. **Prof. Dr. Ir. Jamaluddin jompa, M,Sc.** selaku rektor Universitas Hasanuddin.
4. Bapak **Dr. Eng. Amiruddin** selaku Dekan Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Hasanuddin.
5. **Prof. Dr. Erifin, M,T** selaku kepala Program Studi Fisika Fakultas

Matematika Dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

6. **Seluruh dosen** Fakultas Matematika dan Ilmu Pengetahuan Alam yang telah membagikan ilmunya dan membimbing penulis selama masa studi untuk memperoleh pengetahuan yang luas dan bermanfaat kepada penulis.
7. **Seluruh staff** departemen dan fakultas atas segala fasilitas dan pelayanan yang diberikan kepada penulis selama menempuh studi hingga penelitian dan penyusunan skripsi ini.
8. Kepada kedua orang tua tercinta ayahanda **Torisman** dan ibunda tersayang **Fatmawati** yang telah membimbing penulis dari kecil, tidak pernah lelah merawat dan memberikan dukungan kepada penulis baik secara mental maupun material. Semoga Ayah dan Ibu diberikan nikmat dan kesehatan yang panjang, Amiin.

## DAFTAR ISI

	Halaman
<b>HALAMAN SAMPUL</b>	
<b>HALAMAN JUDUL .....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>iii</b>
<b>PERNYATAAN KEASLIAN .....</b>	<b>iv</b>
<b>KATA PENGANTAR.....</b>	<b>v</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xii</b>
<b>ABSTRAK.....</b>	<b>xiii</b>
<b>ABSTRACT .....</b>	<b>xiv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
I.1 Latar Belakang .....	1
I.2 Rumusan Masalah .....	3
I.3 Tujuan Penelitian .....	3
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>4</b>
II.1 Carbon nano-dot .....	4
II.2 Pengolahan Citra Digital .....	5
II.3 Segmentasi Citra Otsu .....	6
II.4 Deteksi Tepi Canny .....	8
II.5 Transformasi Hough .....	10
II.5.1 Line Hough Transform .....	11
II.5.2 Circle Hough Transform .....	13
II.6 Support Vector Machine .....	14
II.7 Confusion Matrix .....	20

<b>BAB III METODE PENELITIAN .....</b>	<b>22</b>
III.1 Python.....	22
III.2 Alat dan Bahan.....	22
III.3 Konversi Citra.....	22
III.4 Deteksi Tepi Canny.....	23
III.5 Transformasi Hough Garis .....	23
III.6 Transformasi Hough Lingkaran.....	23
III.7 Metode SVM .....	23
III.8 Uji Validitas.....	24
III.9 Bagan Alir Penelitian .....	25
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>26</b>
IV.1 Gambar C-Dots dan Non C-Dots.....	26
IV.2 Modul Python .....	26
IV.3 Hasil Thersholding.....	28
IV.4 SVM 2 Parameter .....	29
IV.5 SVM 7 Parameter .....	31
IV.6 Hasil SVM .....	32
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>33</b>
V.1 Kesimpulan.....	33
V.2 Saran.....	33
<b>DAFTAR PUSTAKA.....</b>	<b>34</b>
<b>LAMPIRAN A .....</b>	<b>35</b>
<b>LAMPIRAN B.....</b>	<b>40</b>



## DAFTAR GAMBAR

<b>Gambar 2.1</b> Gambar Carbon Nanodot.....	<b>4</b>
<b>Gambar 2.2</b> Sistem Koordinat Citra berukuran MxN .....	<b>5</b>
<b>Gambar 2.3</b> Ilustrasi Transformasi Hough Garis .....	<b>11</b>
<b>Gambar 2.4</b> Kontur Kejadian Garis.....	<b>12</b>
<b>Gambar 2.5</b> Transformasi Hough Lingkaran.. .....	<b>13</b>
<b>Gambar 2.6</b> Daftar Kejadian Terbesar.....	<b>14</b>
<b>Gambar 2.7</b> SVM dan OSH .....	<b>15</b>
<b>Gambar 2.8</b> Ilustrasi Tabel 2.5 Dalam Koordinat Kartesian .....	<b>17</b>
<b>Gambar 4.1</b> Karbon Nano-Dot.....	<b>26</b>
<b>Gambar 4.2</b> Karbon Non C-Dot.....	<b>26</b>
<b>Gambar 4.3</b> Citra Tresholding Karbon Nano-Dots .....	<b>28</b>
<b>Gambar 4.4</b> Citra Tresholding Karbon Non C-Dots .....	<b>28</b>
<b>Gambar 4.5</b> Sebaran Data Jumlah garis dan White.....	<b>29</b>
<b>Gambar 4.6</b> Sebaran Data Rata-Rata Panjang Garis dan White .....	<b>30</b>
<b>Gambar 4.7</b> Sebaran Data Untuk Parameter Black dan Jumlah Garis .....	<b>30</b>
<b>Gambar 4.8</b> Rata-rata Jari- Jari Lingkaran dan Black .....	<b>31</b>
<b>Gambar 4.9</b> Hasil SVM.. .....	<b>32</b>

**DAFTAR TABEL**

<b>Tabel 2.1</b> Contoh Matriks Deteksi Tepi Canny .....	<b>9</b>
<b>Tabel 2.2</b> Pengujian Mengubah Nilai 128 Menjadi 255 .....	<b>10</b>
<b>Tabel 2.3</b> Daftar Nilai a dan b Setiap Koordinat .....	<b>12</b>
<b>Tabel 2.4</b> Daftar Nilai a dan b Untuk Setiap Scan Sudut .....	<b>14</b>
<b>Tabel 2.5</b> Tabel Contoh Parameter .....	<b>17</b>
<b>Tabel 2.6</b> Confussion Matriks .....	<b>20</b>
<b>Tabel 4.1</b> Besar Nilai W .....	<b>31</b>

**DAFTAR LAMPIRAN**

<b>Lampiran 1</b> Bahasa Pemrograman Python.....	<b>39</b>
<b>Lampiran 2</b> Gambar Karbon Nano-Dot dan Non Karbon Nano-Dot.....	<b>49</b>
<b>Lampiran 3</b> Hasil Seluruh SVM 7 Parameter .....	<b>72</b>

## ABSTRAK

Penelitian ini membahas penggunaan Support Vector Machine (SVM) untuk mendeteksi Karbon Nano-Dot (C-Dots) dalam citra SEM (*Scanning Electron Microscopy*). C-Dots adalah nanopartikel karbon dengan ukuran sangat kecil dan memiliki potensi aplikasi yang luas dalam bidang nanoteknologi. Metode yang diusulkan dalam penelitian ini mengintegrasikan SVM dengan teknik pemrosesan citra untuk mengidentifikasi dan mendeteksi C-Dots dalam citra SEM. Proses deteksi C-Dots melibatkan beberapa tahap, termasuk pra-pemrosesan citra, ekstraksi fitur, dan klasifikasi dengan SVM. SVM adalah metode pembelajaran mesin yang mampu memisahkan kelas data yang kompleks dengan membangun sebuah hyperplane yang optimal di antara kelas-kelas tersebut. Dalam penelitian ini, SVM dilatih menggunakan contoh citra yang telah diberi label sebagai C-Dots atau bukan C-Dots. Setelah pelatihan, SVM digunakan untuk mengklasifikasikan citra-citra baru dan mengidentifikasi apakah terdapat C-Dots di dalamnya. Hasil eksperimen menunjukkan bahwa metode yang diusulkan dapat mendeteksi C-Dots dengan akurasi yang baik. Dalam banyak kasus, metode ini berhasil mengenali C-Dots dengan tingkat akurasi di atas 90%. Hasil ini menunjukkan bahwa SVM berpotensi menjadi alat yang efektif dalam deteksi C-Dots dalam citra SEM.

***Kata Kunci:*** C-Dots, Bukan C-Dots , SEM, SVM.

## **ABSTRACT**

This study discusses the use of Support Vector Machine (SVM) to detect Carbon Nano-Dots (C-Dots) in SEM (Scanning Electron Microscopy) images. C-Dots are carbon nanoparticles with a very small size and have broad potential applications in the field of nanotechnology. The method proposed in this study integrates SVM with image processing techniques to identify and detect C-Dots in SEM images. The C-Dots detection process involves several stages, including image pre-processing, feature extraction, and classification with SVM. SVM is a machine learning method that is able to separate complex data classes by building an optimal hyperplane between these classes. In this study, SVM was trained using sample images that have been labeled as C-Dots or not C-Dots. After training, SVM is used to classify new images and identify whether there are C-Dots in them. The experimental results show that the proposed method can detect C-Dots with good accuracy. In many cases, this method is successful in recognizing C-Dots with an accuracy rate above 90%. These results indicate that SVM has the potential to be an effective tool in detecting C-Dots in SEM images.

**Keywords:** C-Dots, Not C-Dots , SEM, SVM.

# **BAB I**

## **PENDAHULUAN**

### **I.1 Latar Belakang**

Dalam beberapa kasus karbon nano-dot (C-Dots) sangat sulit untuk dapat dibedakan antara jenis carbon atau selulosa yang lainnya. Karbon nano-dot dapat didefinisikan sebagai material nano yang akan dikompositkan dengan berbagai sintetik ataupun polimer alam dengan model yang teratur. Carbon nano-dot adalah salah satu bahan karbon yang memiliki ukuran tidak lebih dari 10 nm. C-Dots diberbagai kalangan ilmuwan dan universitas telah menjadi nanopartikel yang cukup banyak dikembangkan secara intensif, bahkan sampai sekarang terus dikembangkan dengan berbagai jenis penelitian baik itu sintesis C-Dots dengan sintesis yang berupa polimer alam ataupun berbagai sintesis buat untuk mengembangkan dan mengetahui apa sifat dan struktur lain dari C-Dots. Sampai sekarang penelitian tentang karbon yang menarik tentu saja masih tentang ikatan sumber karbon, namun itu tidak membatasi kita untuk memeliti nanopartikel yang satu ini lebih dalam dan lebih kreatif lagi. [1,2].

Salah satu penelitian yang menarik untuk dilakukan adalah mendeteksi karbon nanodot ini agar dapat dibedakan pada beberapa gambar, karena beberapa gambar karbon nanodot tidak dapat dibedakan antara C-dots dan selulosa tempat mereka melekat pada beberapa kasus. Untuk proses pendeteksian hal ini akan digunakan pemanfaatan citra digital. Dalam pengolahan citra digital sebenarnya sudah banyak proses atau cara yang telah dikembangkan, tujuan pengembangan metode tersebut adalah untuk bagaimana computer dapat mendeteksi dan mengenali objek pada suatu citra dengan tepat dan mengolahnya dalam waktu yang akan relative singkat. Citra sendiri dapat di artikan secara matematis sebagai fungsi kontinu dengan intensitas cahaya pada dua dimensi [3,4].

Pada gambar yang akan diolah dalam bentuk citra digital biasanya akan memiliki macam-macam warna yang berbeda artinya komposisi RGB cukup beragam. RGB secara sederhana dapat diartikan sebagai suatu model warna yang terdiri atas 3 buah warna, Red (R), Green (G), dan Blue (B). Komposisi RGB ini

yang akan mempengaruhi warna yang akan muncul pada computer untuk setiap gambar, perbedaan RGB akan dikombinasikan sehingga membentuk warna-warna lain sehingga didapatkan warna yang dimaksud pada gambar yang akan ditayangkan pada computer atau perangkat lainnya. Untuk kombinasi RGB yang sederhana dapat dijelaskan contoh untuk warna hitam nilai  $R=0$ ,  $G=0$ , dan  $B=0$  sedangkan untuk warna putih nilai RGB-nya bernilai  $R=255$ ,  $G=255$ , dan  $B=255$ . Hitam dan putih adalah salah satu kombinasi warna yang mungkin dibuat oleh kombinasi RGB untuk membentuk warna yang diinginkan [5].

Pada proses citra digital akan digunakan warna citra gambar dalam bentuk grayscale. Grayscale sendiri dapat diartikan sebagai nuansa warna monokromatik dari hitam menjadi putih, sehingga grayscale diartikan sebagai warna abu-abu. Penggunaan Grayscale ini berguna untuk proses citra digital. [6].

Pada proses membedakan Carbon nano-dot dengan material lain yang mungkin tempat melekat carbon nanodot seperti selulosa dapat digunakan ccitra digital untuk membedakan bentuk-bentuk citra dari gambar yang diambil. Pada metode citra digital yang dapat dideteksi atau dibedakan hanya dalam bentuk yang sederhana seperti bagun datar pada umumnya, yaitu lingkaran, persegi Panjang, dan garis. Untuk membedakan antara lingkaran dan garis akan digunakan beberapa metode dalam citra digital dan yang cukup populer adalah Hough Transform baik untuk lingkaran maupun garis.

Pada penelitian ini akan digunakan mesin learning yaitu Support Vector mechine (SVM) metode ini akan memisahkan 2 jenis class pada sebuah data dengan mencari hyperplane terbaik pemisah 2 buah class tersebut. Sebelum menggunakan mesin learning tersebut akan digunakan metode citra digital terlebih dahulu yaitu deteksi tepi dan menggunakan hough transform untuk garis dan lingkara, karena carbon nanodot berbentuk lingkaran dan data yang digunakan dalam penelitian juga berupa selulosa dengan bentuk garis. Hasil dari semua citra digital akan dimasukan kedalam SVM untuk menjadi data sehingga dibuat hyperplane dari data tersebut. Sehingga mesin learning SVM sangat efektif dan efisien untuk membedakan antara lingkaran dan garis pada kasus membedakan carbon nanodot dengan material lainnya.

## **I.2 Rumusan Masalah**

1. Bagaimana cara membedakan antara C-Dots dengan citra Non C-Dots dengan metode SVM?
2. Berapa besar akurasi metode SVM untuk mengklasifikasikan C-Dots dengan Citra Non C-Dots?

## **I.3 Tujuan Penelitian**

1. Membuat pembelajaran mesin metode SVM untuk membedakan C-Dots dengan Citra Non C-Dots.
2. Menghitung akurasi metode SVM untuk mengkalasifikasi C-Dots dengan citra Non C-Dots.



## BAB II TINJAUAN PUSTAKA

### II.1 Carbon Nanodot

Carbon nanodots (C-dots) adalah kelas material nano berbasis karbon yang relatif baru dengan ukuran di bawah 10 nm dan bentuk hampir bulat[1]. Selama bertahun-tahun, nanopartikel ini semakin populer karena sifat fotofisika dan optoelektroniknya yang menarik[2]. Selain itu, C-Dots memiliki sifat menarik lainnya seperti toksisitas rendah, biokompatibilitas, dan kelarutan yang baik hingga sangat baik dalam air dan pelarut organik polar biasa[3]. Seperti kebanyakan nanomaterial karbon, pilihan prekursor dan kondisi sintesis yang tepat sangat penting untuk menyetel sifat mereka untuk aplikasi spesifik. [4, 5]. Terutama ketika sebuah molekul digunakan sebagai prekursor, struktur kimia C-Dots yang dihasilkan dapat mencerminkan molekul precursor. Misalnya, dengan menggunakan asam amino dan molekul pembawa amina, dimungkinkan untuk memperoleh partikel nano yang kaya akan amina permukaan dan fungsi nitrogen lainnya. [8] Yang penting, gugus amino permukaan ini dapat meniru perilaku kimia katalis amina molekuler klasik, yang memungkinkan penerapan CD yang mengandung amina sebagai platform nanokatalitik dalam sintesis organik. dan akhirnya kembali ke keadaan awalnya dalam fase padat [2,9].

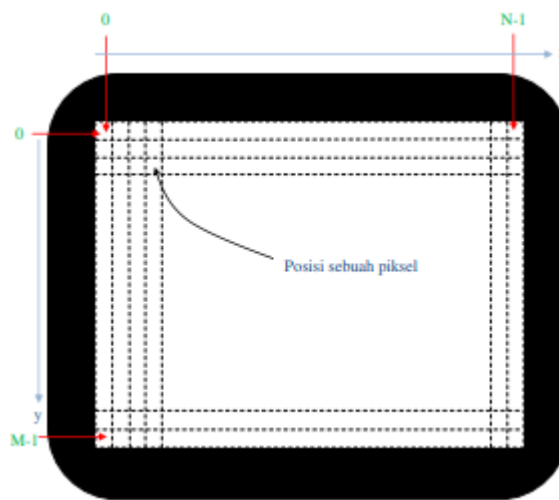


Gambar 2.1 Gambar Carbon Nanodot

## II.2. Pengolahan Citra Digital

Pengolahan citra digital adalah proses manipulasi citra dengan menggunakan bantuan komputer dengan tujuan mendapatkan informasi pada gambar untuk tujuan tertentu dan proses selanjutnya.

Citra digital dibentuk dari sekumpulan piksel. Setiap piksel digambarkan sebagai kotak kecil. Citra digital didefinisikan sebagai fungsi  $f(x,y)$  ukuran N kolom dan M baris, x y merupakan koordinat spasial dan f adalah titik koordinat yang menunjukkan nilai keabuan di titik tersebut[5][9][10].



Gambar 2.2 Sistem koordinat citra berukuran M x N

Berdasarkan nilai piksel, citra digital dikelompokkan kedalam tiga jenis citra yaitu:

1. Citra Warna

Citra Warna (RGB) memiliki 3 komponen warna yakni Red, Green, Blue, di setiap piksel. Setiap komponen warna menggunakan 8 bit (nilai kisaran 0 sampai 255). Dengan demikian, kemungkinan warna yang bisa disajikan mencapai  $256*256*256$  atau 16.777.216 kombinasi warna.

2. Citra Aras Keabuan

Citra aras keabuan memiliki satu nilai terusan disetiap piksel dengan kata lain nilai terusan red=green=blue. Nilai tersebut menunjukkan tingkat intensitas. Tingkat intensitas dimulai dari warna hitam, keabuan sampai

putih. Citra aras keabuan memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan) diperoleh menggunakan persamaan berikut :

$$\text{aras keabuan} = \frac{((red) + (green) + (blue))}{3}$$

3. Citra biner, memiliki dua kemungkinan nilai piksel yakni hitam dan putih atau 0 dan 1. Citra biner sering muncul sebagai hasil proses thresholding, segmentasi ataupun morfologi.

Komponen warna citra digital setiap piksel dikonversi menjadi matriks yang berisi nilai-nilai riil. Matriks didefinisikan sebagai fungsi dua dimensi  $f(x,y)$  dengan ukuran matriks M kali N, dimana M dapat dinyatakan sebagai baris dan N adalah kolom serta x dan y adalah pasangan koordinat spasial. Nilai f pada titik koordinat (x,y) disebut sebagai skala keabuan atau intensitas dari citra digital di koordinat tersebut. Nilai x,y dan f secara keseluruhan berhingga atau bernilai diskrit sehingga citra digital. Citra digital direpresentasikan dalam bentuk matriks persegi yang mewakili ukuran citra tersebut. Misalkan terdapat sebuah citra digital dengan ukuran MxN, maka citra dapat direpresentasikan dalam sebuah matriks berukuran MxN sebagai berikut:

$$f(x, y) = \begin{pmatrix} f(1,1) & f(1,2) & f(1,N) \\ \dots & \dots & \dots \\ f(M,1) & f(M,2) & f(M,N) \end{pmatrix} \quad (2.1)$$

### II.3 Segmentasi Citra Otsu

Segmentasi citra bertujuan mendapatkan objek-objek citra dengan cara membagi citra ke dalam beberapa daerah yang memiliki kemiripan atribut. Terdapat dua teknik segmentasi citra yang dapat digunakan yaitu mendeteksi diskontinuitas dan similaritas. Pendekatan diskontinuitas membagi citra berdasarkan perubahan intensitas yang tiba-tiba. Sedangkan pendekatan similaritas memecah citra ke dalam daerah yang sama menurut beberapa kriteria yang sudah ditentukan, seperti proses *tersholding*.

*Thresholding* mengubah citra keabuan menjadi citra biner bergantung pada nilai threshold atau disebut nilai (T) sehingga dapat diketahui daerah mana yang

termasuk objek dan latar belakang. Jika piksel lebih besar dari threshold diatur menjadi 1 sebaliknya jika kurang dari threshold diatur menjadi 0. Secara umum proses thresholding ditunjukkan pada persamaan 1.1[12]

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (2.2)$$

Persamaan (2.2) menjelaskan bahwa  $g(x,y)$  adalah citra biner dari citra aras keabuan  $f(x,y)$  dan  $T$  menyatakan nilai ambang. Konsep *thresholding* Otsu pertama kali diperkenalkan oleh Nobuyuki Otsu (1979) untuk pengelompokan citra biner berdasarkan bentuk histogram secara otomatis, mengasumsikan bahwa citra berisi dua kelas dasar dengan bentuk histogram bimodal.[12] Tujuan metode Otsu adalah membagi histogram citra keabuan ke dalam dua daerah yang berbeda secara otomatis.

Berikut penurunan matematis Otsu menentukan *threshold* ( $k$ ). Nilai  $k$  berkisar antara 0 sampai 255.[14]

1. Probabilitas setiap piksel pada gray level  $i$

$$p_i = \frac{n_i}{N} \quad (2.3)$$

$n_i = \text{jumlah piksel pada level ke } i$

2. Jumlah kumulatif

$$\omega(k) = \sum_{i=0}^k p_i \quad (2.4)$$

3. Rerata kumulatif

$$\mu(k) = \sum_{i=0}^k i p_i \quad (2.5)$$

4. Rerata intensitas global

$$\mu T = \sum_{i=0}^N ip_i \quad (2.6)$$

5. Nilai ambang  $k$  ditentukan dengan memaksimumkan persamaan 2.7:

$$\sigma_B^2(k) = \frac{[\mu T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \quad (2.7)$$

## II.4 Deteksi Tepi Canny

Deteksi tepi dapat digunakan untuk memperoleh tepi-tepi yang diinginkan dari suatu objek citra yang digunakan. Pada batas dua daerah terdapat perubahan nilai intensitas yang cukup dratis yang disebut dengan tepi. Tepi dapat didefinisikan sebagai jarak singkat dari sebuah perubahan nilai intensitas derajat keabuan suatu citra yang mendadak (besar). Bentuk maupun ukuran objek merupakan informasi penting yang dapat diketahui dari deteksi tepi.[17] Pada deteksi tepi ada banyak teknik yang dapat digunakan, Adapun beberapa teknik dalam deteksi tepi yang dapat digunakan yaitu orde pertama menggunakan turunan pertama seperti operator *Roberts*, *Prewitt*, *Sobel* lalu orde kedua menggunakan turunan kedua seperti *Laplacian of Gaussian (LoG)* dan operator Kompas yakni medeteksi tepi kesegala arah mata angin seperti *Krish*, *Robinson*. [18]

Deteksi tepi Canny pertama kali dikembangkan oleh John Canny pada tahun 1986, dengan menggunakan alogaritma multi-tahap dalam mendeteksi tepi citra. Algoritma deteksi tepi Canny memiliki 3 kriteria yakni memberikan tingkat kesalahan yang paling minimum, serta mengalokasikan titik-titik tepi (jarak piksel-piksel tepi yang ditemukan deteksi dan tepi yang sesungguhnya sangat pendek), dan hanya memberikan satu respon untuk satu buah tepi. Terdapat enam tahap dalam mendeteksi tepi *Canny* sebagai berikut:[22]

1. Mengimplementasikan tapis *Gaussian* untuk mereduksi noise dan meningkatkan kualitas detail citra. Proses ini menghasilkan citra tampak sedikit lebih buram yang tujuan proses ini mendapatkan tepian citra yang sebenarnya sehingga garis-garis halus yang berada di area gradasi pada

citra masih bisa dideteksi. Filter *Gaussian* 2 dimensi dinyatakan dalam persamaan berikut:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.8)$$

Dimana  $G(x,y)$  elemen matriks gaussian pada posisi  $(x,y)$ ,  $\pi = 22/7$ ,  $e = 2.71828182846$ ,  $\sigma$  ialah nilai standar deviasi. Berdasarkan rumus (2.8), salah satu filter *Gaussian* dengan  $\sigma = 1.4$  dan ukuran kernel  $3 \times 3$  adalah:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Contoh: Misalkan matriks citra berukuran  $5 * 5$  yang berisi:

Tabel 2.1 Contoh Matriks Deteksi Tepi Canny

<b>0</b>	<b>16</b>	<b>32</b>	<b>48</b>	<b>64</b>
<b>16</b>	<b>0</b>	<b>16</b>	<b>32</b>	<b>48</b>
<b>32</b>	<b>16</b>	<b>0</b>	<b>16</b>	<b>32</b>
<b>48</b>	<b>32</b>	<b>16</b>	<b>0</b>	<b>16</b>
<b>64</b>	<b>48</b>	<b>32</b>	<b>16</b>	<b>0</b>

Matriks dikalikan antar komponen dengan kernel Gaussian. Proses perkalian komponen utama di 2,2 dijelaskan sebagai berikut:

$$H(2,2) = (0*1 + 16*2 + 32*1 + 16*2 + 0*4 + 16*2 + 32*1 + 16*2 + 0*1) / 16 = 12$$

Proses perkalian komponen utama di 4,2 dijelaskan sebagai berikut:

$$H(4,2) = (32*1 + 48*2 + 64*1 + 16*2 + 32*4 + 48*2 + 0*1 + 16*2 + 32*1) / 16 = 34$$

Proses perkalian komponen utama di 3,4 dijelaskan sebagai berikut:

$$H(3,4) = (32*1 + 16*2 + 0*1 + 48*2 + 32*4 + 16*2 + 64*1 + 48*2 + 32*1) / 16 = 34$$

Perkalian dikerjakan untuk semua komponen matriks citra. Hasil perhitungan adalah:

$$\begin{bmatrix} 12 & 19 & 34 \\ 19 & 12 & 19 \\ 34 & 19 & 12 \end{bmatrix}$$

- Melakukan perhitungan besar gradient dan arah tepi. Perhitungan gradient magnitude menggunakan salah satu kernel *Roberts*, *Prewitt*, atau *Sobel*.
- Memperkecil garis tepi yang muncul dengan menerapkan *nonmaximum suppression* sehingga menghasilkan garis tepian yang terlihat lebih ramping. *Nonmaximum Suppression* sendiri dilaksanakan dengan

mempertimbangkan dan juga memperhatikan dua titik tetangga yang terletak pada arah tepi. Jika nilai piksel titik perhatian lebih besar daripada gradien kedua tetangga, nilainya dipertahankan. Sebaliknya, jika piksel titik perhatian lebih kecil daripada nilai salah satu atau kedua gradien tetangga, nilainya dirubah menjadi 0.

4. Menerapkan dua buah *threshold* (*double thresholding*). Sederhananya *double thresholding* bertujuan klarifikasi dua buah *High-threshold* ( $T_2$ ) dan *low-threshold* ( $T_1$ ), dengan  $T_2 \approx 2T_1$ . Jika nilai piksel lebih besar atau sama dengan  $T_2$  maka diatur nilai 255, jika nilai piksel kurang dari atau sama dengan  $T_1$  maka diatur menjadi 0. Piksel diantara  $T_1$  dan  $T_2$  disebut kandidat piksel tepi maka sementara diberi nilai 128.
5. *Edge tracking by hysteresis* bertujuan memperoleh tepian final dengan menekan semua sisi yang tidak terhubung pada tepian yang sangat kuat. Nilai 128 selanjutnya dilakukan pengecekan pada piksel dari 8 arah tetangganya, sehingga piksel hanya bernilai 0 atau 255. Perubahan nilai 128 menjadi nilai 255 apabila semua kondisi terpenuhi yakni jika salah satu atau semua piksel pada 8 arah tetangganya bernilai 255. Proses pengujian dilakukan samapai tidak ada lagi perubahan dari nilai 128 menjadi 255. Selanjutnya, semua piksel yang bernilai 128 yang tersisa diubah menjadi 0.[21][22]

Tabel 2.2 Pengujian mengubah nilai 128 menjadi 255

	j-1	j	j+1				
i-1	255	255	255	➔	255	255	255
i	255	128	255		255	255	255
i+1	255	255	255		255	255	255

## II.5 Transformasi Hough

Transformasi Hough adalah sebuah transform yang pertama kali diperkenalkan oleh Paul Hough pada tahun 1962, Transformasi Hough (TH) merupakan teknik pengalokasian bentuk-bentuk dalam gambar. Secara khusus, transformasi ini digunakan untuk ekstraksi garis, lingkaran dan elips. TH kemudian diimplementasikan untuk mendapatkan bangun ruang tertentu dalam

gambar dan kemudian meluas karena transformasi ini memiliki banyak kelebihan dan banyak potensi untuk pengembangan lebih lanjut. Kelebihan utamanya yaitu dapat memberikan hasil lebih cepat dan sama dengan pencocokan pola[18][4].

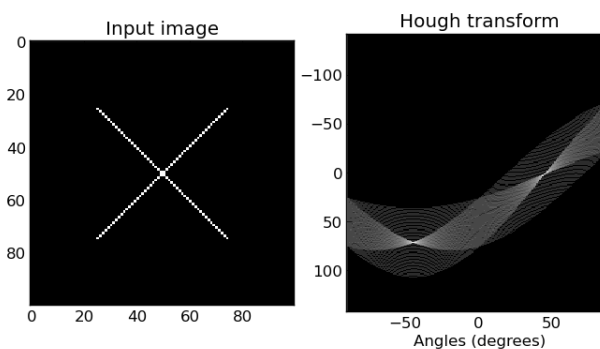
Ide awal TH melakukan pemetaan dari titik gambar menuju sebuah ruang akumulator (ruang Hough). Pemetaan tersebut diperoleh dalam bentuk yang efisien secara matematis, berdasarkan fungsi yang menjelaskan kondisi dari target. Pemetaan ini membutuhkan jauh lebih sedikit sumber perhitungan matematis daripada dengan pencocokan pola. Karena TH ekuivalen dengan pencocokan pola, TH menjadi salah satu dari teknik-teknik ekstraksi bentuk yang sering digunakan.

### II.5.1 Transformasi Hough Garis

Transformasi Hough garis digunakan untuk melakukan deteksi garis lurus. Secara detail Transformasi Hough garis dapat diartikan sebagai teknik transformasi piksel dengan nilai tertentu yang dapat digunakan untuk memperoleh garis dalam sebuah citra. Ini dilakukan untuk mendapatkan suatu fitur yang lebih spesifik sesuai dengan tujuan utama dari transformasi garis ini[1][4]. Dua buah titik bisa membentuk satu garis lurus dengan persamaan:

$$y = ax + b \quad (2.9)$$

dengan nilai a dan b tertentu. Jika ada banyak titik dan dipilih hanya dua titik tertentu, terdapat banyak kombinasi pasangan dua titik yang memiliki nilai a dan b tertentu. Maka, untuk setiap pasangan a dan b tertentu, terdapat sejumlah kejadian. Garis dibentuk dari setiap pasangan a dan b yang memenuhi syarat pengembangan. Ilustrasi tersebut terdapat dalam gambar 2.3.



Gambar 2.3 Ilustrasi Transformasi Hough garis



Contoh :

Misalkan beberapa titik koordinat yang mungkin membentuk garis.

-2,2	-1,2	0,2	1,2	2,2
-2,1	-1,1	0,1	1,1	2,1
-2,0	-1,0	0,0	1,0	2,0
-2,-1	-1,-1	0,-1	1,-1	2,-1
-2,-2	-1,-2	0,-2	1,-2	2,-2

Kotak yang berwarna kuning adalah garis, dan angka di dalam kotak adalah koordinat kartesius. Dalam kotak dibuat ada delapan kotak, didaftarkan semua koordinat. Lalu antar semua titiknya dihitung nilai a dan b yang mengacu pada persamaan (2.9).

Tabel 2.3 Daftar nilai a dan b setiap koordinat

No	Koordinat	Relasi	a	b
1	-2,0	P1 dan P2	-1	-2
2	-1,1	P1 dan P3	-1	-2
3	0,2	P1 dan P4	-3	-2
4	1,1	P1 dan P5	inf	inf
5	2,0	...		
6	1,-1	P2 dan P3	-1	-2
7	0,-2	P2 dan P4	inf	inf
8	-1,-1	P2 dan P5	3	2
..	.....	...	....	....
28	-1,-1	P7 dan P8	-1	2

Lalu membuat koordinat a dan b semua kejadian pasangan nilai a dan b tertentu dituliskan dalam koordinat tersebut. Garis akan dibentuk pada kejadian yang paling besar.

		b									
		-4	-3	-2	-1	0	1	2	3	4	
a	-4	0	0	0	0	0	0	0	0	0	
	-3	0	0	0	0	0	0	0	0	0	
	-2	0	1	0	0	0	0	0	1	0	
	-1	0	0	2	0	0	0	2	0	0	
	0	0	0	0	1	1	1	0	0	0	
	1	0	0	2	0	0	0	2	0	0	
	2	0	1	0	0	0	0	0	1	0	
	3	0	0	0	0	0	0	0	0	0	
	4	0	0	0	0	0	0	0	0	0	
Jumlah kejadian terbanyak bernilai 2 ada 4											
a	b										
	-1	-2									
	1	-2									
	-1	2									
	1	2									

Gambar 2.4 Kontur Kejadian Garis

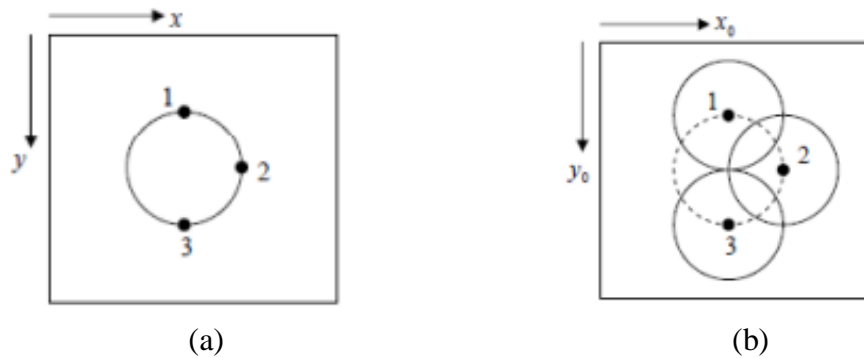
Sehingga didapatkan kejadian terbesar ada pada koordinat (-1,-2), (1,-2), (-1,2), dan (1,2) yang masing-masing sebanyak 2 kejadian.

### II.5.2 Transformasi Hough Lingkaran

Transformasi Hough lingkaran berawal dari persamaan umum lingkaran, berikut:

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \quad (2.10)$$

Di persamaan (2.10) ada tiga tetapan yaitu  $x_0$ ,  $y_0$  dan  $r$ , ada dua variable yaitu  $x$  dan  $y$  yang menunjukkan koordinat piksel lingkaran yang terdapat. Satu pasang koordinat  $x$  dan  $y$  dari gambar dan nilai  $r$  yang dipilih dipetakan ke titik pusat lingkaran pada sudut tertentu. Hasil pemetaan ini adalah titik pusat lingkaran dan jari-jari yang ditetapkan. Untuk banyak titik piksel dan jari-jari yang telah dipilih terdapat banyak hasil pemetaan. Titik pusat lingkaran diambil dari kejadian yang memenuhi ambang batas tertentu.



Gambar 2.5 (a) Citra yang berisi lingkaran

(b) Ruang Akumulator

Contoh:

<b>R=1, (a,b)=(0,0)</b>		
<b>Pilihan r = 1</b>		
		(a,b)
4 titik yang dilewati		
=		(-1,0)
		(0,1)
		(-1,0)
		(0,-1)

Tabel 2.4 Daftar nilai a dan b untuk setiap scan sudut

Scan tiap 90	dipilih	r = 1					
0, 90, 180, 270, 360							
			$\theta$				
		0	90	180	270	360	
a,b	-1,0	0,0	1,1	-2,0	-1,-1	0,0	
	0,1	1,1	0,2	-1,1	0,0	1,1	
	1,0	2,0	1,1	0,0	1,-1	2,0	
	0,-1	1,-1	0,0	-1,-1	0,-2	-1,-1	

Misalkan lingkaran dengan jari-jari 1 dan titik pusat (0,0). Ada 4 titik yang mungkin dilewati lingkaran. Lalu kita pilih jari-jari = 1 scan sudut 0, 90, 180, 270, 360 dan titik pusatnya (-1,0), (0,1), (1,0), (0,-1). Misalkan titik pusat (-1,0) pada sudut 0 derajat memiliki titik lingkaran (0,0) dan seterusnya sesuai dengan scan. Akan didapatkan gambar untuk setiap kejadian sebagai berikut:

		b				
		-2	-1	0	1	2
a	2	0	0	0	0	0
	1	0	2	1	2	0
	0	1	0	4	0	1
	-1	0	2	1	2	0
	-2	0	0	0	0	0

Gambar 2.6 Daftar kejadian terbesar

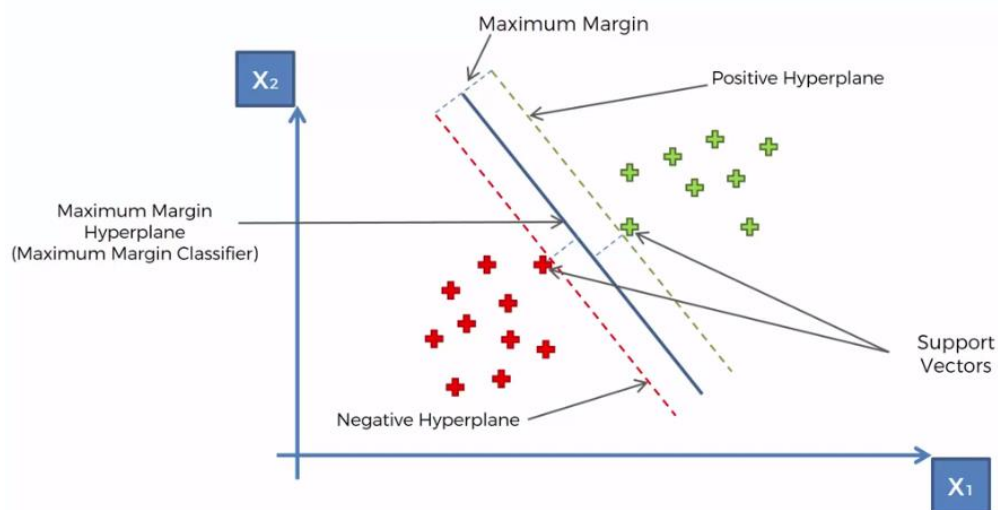
Gambar 2.5 menyatakan koordinat dengan kejadian terbanyak yaitu (0,0) sebanyak 4 kejadian, artinya koordinat (0,0) menjadi pusat lingkaran.

## II.6 Support Vector Machine (SVM)

Support vector machine (SVM) adalah model mesin pembelajaran dengan algoritma yang mempelajari menganalisis data untuk klasifikasi dan analisis regresi. SVM dikembangkan AT&T Bell Laboratories oleh Vladimir Vapnik dengan temannya. SVM adalah salah satu metode prediksi paling kuat, didasarkan pada kerangka pembelajaran statistik atau teori VC yang diusulkan oleh Vapnik (1982) dan Chervonenkis (1974). Diberikan satu set contoh pelatihan, masing-masing ditandai sebagai milik salah satu dari dua kategori, algoritma pelatihan SVM membangun model yang memberikan contoh baru ke satu kategori atau yang lain, menjadikannya sebagai pengklasifikasi linier biner

non-probabilistik (walaupun metode seperti Platt penskalaan ada untuk menggunakan SVM dalam pengaturan klasifikasi probabilistik). SVM memetakan contoh-contoh pelatihan ke titik-titik dalam ruang untuk memaksimalkan lebar celah antara kedua kategori tersebut. Contoh-contoh baru kemudian dipetakan ke dalam ruang yang sama dan diprediksi termasuk dalam kategori berdasarkan di sisi celah mana mereka jatuh.

SVM berusaha untuk mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada *input* yang diinginkan. *Hyperplane* adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Pemisahan yang dibuat oleh OSH (*Optimum Separating Hyperplane*), menjadi pemisah yang sangat optimal yang dapat digunakan untuk klasifikasi. Misalkan  $\{x_1, \dots, x_n\}$  adalah kumpulan data dan  $y_i \in \{+1, -1\}$  adalah sebuah kelas dari data [11][21].



Gambar 2.7 SVM dan OSH

Pada gambar 2.6 dapat dilihat bidang pemisah yang memisahkan semua objek sesuai dengan kelasnya. Data yang di perbatasan area disebut Support Vector. Dua kelas dapat dipisahkan oleh sepasang bidang pembatas yang sejajar. Pertama terlihat *hyperplane* membatasi kelas pertama sedangkan *hyperplane* kedua membatasi kelas kedua, sehingga dapat dibuat persamaan: [11][21]

$$x_i \cdot w + b \geq +1, y_i = +1 \quad (2.11)$$

$$x_i \cdot w + b \leq -1, y_i = -1 \quad (2.12)$$

w adalah Normal plane atau yang disebut sebagai *support vector* dan b adalah posisi relative bidang terhadap pusat koordinat atau yang disebut juga dengan bias. Karena data berupa nonlinear, maka data akan diubah menjadi fitur dimensi menggunakan fungsi pemetaan (transformasi)  $x_k \rightarrow \phi(x_k)$  dengan a sebagai fungsi kernel K, kernel yang digunakan pada persamaan ini adalah kernel RBF yang dapat dilihat pada persamaan berikut:[21]

$$k(x_i, x_j) = \exp\left(-\frac{1}{\delta^2} \|x_i - x_j\|^2\right) \quad (2.13)$$

Dimana  $x_i$  dan  $x_j$  adalah *feature vectors* dan  $\delta$  adalah produk dari parameter C dan gamma SVM dapat diatur oleh user. Persamaan Weight (W) vector dapat dituliskan dalam definisikan berikut:[11]

$$W = \sum_{i=1}^N \alpha_i y_i k(x_i) \quad (2.14)$$

*Weight vector* biasanya akan memiliki nilai yang sangat besar, namun hal ini bergantung pada nilai  $\alpha$ . Lagrange multiplier digunakan untuk menentukan nilai dari  $\alpha$ , sehingga kita dapat menentukan nilai w. Nilai b dapat dditentukan dari persamaan berikut:[21]

$$b = \frac{1}{SV} \sum_{x_i \in SV} \left( \frac{1}{y_i} - \sum_{x_j \in SV} \alpha_j y_j k(x_j, x_i) \right) \quad (2.15)$$

Dimana SV adalah jumlah Support vector. Separator terbaik pada persamaan hyperplane adalah program persamaan kuadrat dengan nilai maximum sehingga kita bisa mencari nilai  $\alpha$ . Setelah persamaan kuadrat yang diinginkan ditemukan maka kelas dari berbagai nilai x dapat ditentukan persamaan determinan fungsi sebagai berikut: [15][21]

$$D(z) = \text{sign} \left( \sum_{i=1}^N \alpha_i y_i k(x_i, z) + b \right) \quad (2.16)$$

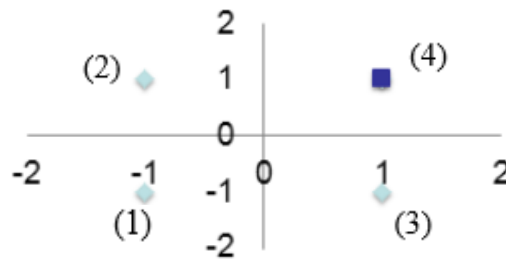
Dimana D(z) adalah persamaan *decision* untuk semua kelas dan N adalah angka untuk support vector.[15]

Misalkan ada data seperti pada tabel 2.5 untuk setiap koordinat yang dicari *hyperplane*:

Tabel 2.5 Tabel Contoh Parameter

x	Y	N
-1	-1	-1
-1	1	-1
1	-1	-1
1	1	1

Keempat data di dalam tabel 2.5 dapat diilustrasikan dalam gambar (2.7)



Gambar 2.8 Ilustrasi tabel 2.5 dalam koordinat kartesian

Keterangan gambar:

$$w_1 \cdot x + w_2 \cdot y + b \geq 1, \text{ termasuk kelompok persegi}$$

$$w_1 \cdot x + w_2 \cdot y + b \leq -1, \text{ termasuk kelompok belah ketupat}$$

Dapat digunakan persamaan:

$$y_i \cdot (x_i \cdot w + b) \geq 1 \tag{2.17}$$

$$N_i(x_i w_x + y_i w_y + b) = 1 \tag{2.18}$$

Nilai i di persamaan 2.18 berarti baris matriks, dan nilai x, y, N dimasukkan ke persamaan 2.18. Dengan memasukkan nilai dalam tabel ke persamaan (2.17), didapatkan empat persamaan:

$$-w_x - w_y + b = -1$$

$$-w_x + w_y + b = -1$$

$$w_x - w_y + b = -1$$

$$w_x + w_y + b = 1$$

Solusi persamaan 2.18 dapat dilakukan dengan metode substitusi. Untuk nilai  $x=-1$ ,  $y=-1$  dan  $N=-1$ , didapatkan:

$$\begin{aligned} (-1)((-1)w_x + (-1)w_y + b) &\geq 1 \\ w_x + w_y - b &\geq 1 \end{aligned}$$

Untuk nilai  $x=-1$ ,  $y=1$  dan  $N=-1$ , didapatkan:

$$\begin{aligned} (-1)((-1)w_x + (1)w_y + b) &\geq 1 \\ w_x - w_y - b &\geq 1 \end{aligned}$$

Untuk nilai  $x=1$ ,  $y=-1$  dan  $N=-1$ , didapatkan:

$$\begin{aligned} (-1)((1)w_x + (-1)w_y + b) &\geq 1 \\ -w_x + w_y - b &\geq 1 \end{aligned}$$

Untuk nilai  $x=1$ ,  $y=1$  dan  $N=1$ , didapatkan:

$$\begin{aligned} (1)((1)w_x + (1)w_y + b) &\geq 1 \\ w_x + w_y + b &\geq 1 \end{aligned}$$

Kemudian dilakukan eliminasi:

$$\begin{aligned} w_x + w_y - b &= 1 \\ w_x - w_y - b &= 1 \\ \hline &(-) \\ 2w_y &= 2 \\ \mathbf{w_y} &= \mathbf{1} \\ w_x + w_y + b &= 1 \\ w_x - w_y - b &= 1 \\ \hline &(+ \\ 2w_x &= 2 \\ \mathbf{w_x} &= \mathbf{1} \\ -w_x - w_y - b &= 1 \\ w_x - w_y - b &= 1 \\ \hline &(+ \\ -2b &= 2 \\ \mathbf{b} &= \mathbf{-1} \end{aligned}$$

Maka didapatkan persamaan *hyperplane* sebagai berikut:

$$w_x \cdot x + w_y \cdot y + b = 0$$

$$1x + 1y + (-1) = 0$$

$$x + y = 1$$

Selain itu, solusi menggunakan aturan Cramer dihitung menggunakan 2 kombinasi pengambilan titik.

Kombinasi pertama diambil titik (2), (3), (4)

Dihitung determinan yang menghasilkan  $\Delta = 4, \Delta W_x = 4, \Delta W_y = 4, \Delta b = -4$ .

Maka nilai  $w_1 = 1 ; w_2 = 1 ; b = -1$

Persamaan hyperplanenya adalah

$$x + y = 1$$

Untuk titik 1 (-1, -1)

$$1(-1) + 1(-1) - 1 = -3$$

Karena  $N$  di titik 1  $\leq -1$ , titik satu merupakan kelompok belah ketupat.

Untuk titik 2 (-1,1)

$$1(-1) + 1(1) - 1 = -1$$

Karena  $N$  di titik 2  $\leq -1$ , titik dua merupakan kelompok belah ketupat.

Untuk titik 3 (1, -1)

$$1(1) + 1(-1) - 1 = -1$$

Karena  $N$  di titik 3  $\leq -1$ , titik tiga merupakan kelompok belah ketupat.

Untuk titik 4 (1,1)

$$1(1) + 1(1) - 1 = 1$$

Karena  $N$  di titik 4  $\geq 1$ , titik empat merupakan kelompok persegi.

Kombinasi kedua diambil titik (1), (2), (4)

Dihitung determinan yang menghasilkan  $\Delta = -4, \Delta W_x = -4, \Delta W_y = 0, \Delta b = 0$ .

Maka nilai  $w_1 = 1 ; w_2 = 0 ; b = 0$

Persamaan hyperplanenya adalah

$$x = 0$$

Untuk titik 1 (-1, -1)



$$1(-1) + 1(0) - 0 = -1$$

Karena  $N$  di titik 1  $\leq -1$ , titik satu merupakan kelompok belah ketupat.

Untuk titik 2 (-1,1)

$$1(-1) + 1(0) - 0 = -1$$

Karena  $N$  di titik 2  $\leq -1$ , titik dua merupakan kelompok belah ketupat.

Untuk titik 3 (1, -1)

$$1(1) + 1(0) - 0 = 1$$

Karena  $N$  di titik 3  $\geq 1$ , titik tiga merupakan kelompok persegi.

Untuk titik 4 (1,1)

$$1(1) + 1(0) - 0 = 1$$

Karena  $N$  di titik 4  $\geq 1$ , titik empat merupakan kelompok persegi.

## II.7 Confusion Matriks

*Confusion matriks* adalah salah satu metode yang sering digunakan dalam melakukan perhitungan akurasi pada konsep *data mining*. *Confusion matrix* digambarkan dengan menyatakan jumlah data uji yang benar diklasifikasikan dalam jumlah data uji yang salah diklasifikasikan. Evaluasi dengan confusion matrix menghasilkan nilai *accuracy*, *precision*, dan *recall*. [23].

Tabel 2.6 *Confusion Matrix*

<i>Correct Classification</i>	<i>Classified as</i>	
	<i>Predicted "+"</i>	<i>Predicted "-"</i>
<i>Actual "+"</i>	<i>True Positive</i>	<i>False Negative</i>
<i>Actual "-"</i>	<i>False Positive</i>	<i>True Negative</i>

Berdasarkan tabel 2.6 *Confusion matrix* diatas, *true positive* (TP) adalah jumlah record data positif yang diklasifikasikan sebagai positif, *False Positive* (FP) adalah jumlah record data negative yang diklasifikasikan sebagai nilai positif, *False Negative* (FN) adalah jumlah record data positif yang diklasifikasikan, sebagai nilai positif, dan *true negative* (TN) adalah jumlah record data negative yang diklasifikasikan sebagai nilai negatif. [23]

Membuat tabel perbandingan actual dan prediksi.

x	Y	N(Actual)	N(Cramer 1)	N(Cramer 2)
-1	-1	-1	-3	-1
-1	1	-1	-1	-1
1	-1	-1	-1	1
1	1	1	1	1

Dari kombinasi pertama metode Cramer dapat dibuat *Confusion matrix*

		Prediksi	
		◆	■
Aktual	◆	3	-
	■	-	1

Dari kombinasi kedua metode Cramer dapat dibuat *Confusion matrix*

		Prediksi	
		◆	■
Aktual	◆	2	1
	■	-	1

Dengan aturan Cramer, kombinasi titik yang diambil mempengaruhi *Confusion matriks*.