

Identifikasi Jenis Bangun Datar dengan Algoritma *Line Hough Transform* dan *Circular Hough Transform*

Eflin Winata¹, Helvi Risna², Renni Angreni³

^{1,2,3}STMIK GI MDP, Jalan Rajawali No.14 Palembang, 0711-376400

^{1,2,3}Jurusan Teknik Informatika, STMIK MDP, Palembang

e-mail: ¹eflin12@yahoo.com, ²helvirisna2412@gmail.com, ³renni@mdp.ac.id

Abstrak

Bangun datar adalah bagian bidang datar yang terdiri dari garis lurus dan garis lengkung, oleh karena itu diperlukan metode khusus untuk identifikasi bangun datar. Metode yang dapat diterapkan adalah *Line Hough Transform* untuk mendeteksi garis lurus dan *Circular Hough Transform* untuk mendeteksi garis lengkung. Penelitian ini menerapkan algoritma *Line Hough Transform* dan *Circular Hough Transform* dalam mengidentifikasi jenis bangun datar dengan membuat aplikasi sebagai media pendukung pengujian. Jenis bangun datar yang digunakan dalam penelitian ini adalah persegi, persegi panjang, trapesium, segitiga, jajar genjang, layang-layang, belah ketupat dan lingkaran. Berdasarkan hasil pengujian menggunakan 20 jenis data testing tiap jenis bangun datar menunjukkan persentase keberhasilan dalam mengidentifikasi bangun datar adalah 88,75%. Banyak faktor yang mempengaruhi tingkat keberhasilan pengenalan bangun datar seperti ukuran, sudut, threshold, serta cara menggambar yang normal.

Kata kunci - Pengolahan Citra, *Line Hough Transform*, *Circular Hough Transform*, Identifikasi Bangun Datar.

Abstract

Two-dimensional figure is consisted of straight and curved lines, which is needed a special method to identify them. The methods that can be used are called *Line Hough Transform* for identifying straight line and *Circular Hough Transform* to identifying curved line. This research is to apply this algorithm, *Line Hough Transform* and *Circular Hough Transform*, in identifying two-dimensional figure by using an application as a supported testing media. The types of two-dimentional figure, that are used in this study, are square, rectangle, trapezium, triangle, parallelogram, kite rhombus, and circle. Based on test results using 20 types of testing data each type two-dimensional figure shows the percentage of success in identifying two-dimensional figure is 87.5%. Many factors affect the rate of successful Identification of two-dimensional figure as size, angle, threshold, and how to draw normal.

Keywords - Image Processing, *Line Hough Transform*, *Circular Hough Transform*, Identification of Two-Dimensional Figure.

1. PENDAHULUAN

Bangun datar ialah bangun yang dibuat (dilukis) pada permukaan datar. Bangun datar yang mempunyai tiga sisi disebut segitiga, empat sisi disebut segi empat, lima sisi disebut pentagon, enam sisi disebut heksagon, dan sebagainya [1].

Jenis bangun datar bermacam-macam, mulai dari segitiga, segi empat, segi lima, dan lain sebagainya. Contoh bangun datar adalah persegi, persegi panjang, segitiga, jajar genjang, trapesium, layang-layang, belah ketupat, dan lingkaran. Garis-garis yang ada pada bangun datar saling terhubung satu sama lain. Sebagai contoh jika ada empat buah garis yang saling terhubung maka dapat diidentifikasi sebagai bangun segi empat, contoh lainnya jika ada tiga buah garis saling terhubung maka bisa diidentifikasikan sebagai bangun segitiga. Apabila ingin mengidentifikasi suatu bangun datar, maka diperlukan proyeksi terhadap keberadaan garis lurus maupun garis lengkung pada objek geometris tersebut.

Salah satu metode yang dapat digunakan untuk membantu mengidentifikasi bangun datar tersebut adalah metode *Hough Transform*. Objek yang biasa dideteksi dengan metode *Hough Transform* berupa garis, lingkaran, dan elips. Metode *Hough Transform* sendiri telah banyak dilakukan pada beberapa penelitian sebelumnya. Seperti halnya penelitian yang dilakukan oleh Sa'diyah, R.Rizal Isnanto, dan Achmad Hidayatno [2] telah membuktikan bahwa suatu garis lurus dapat terdeteksi menggunakan metode *Line Hough Transform* dimana pada proses transformasi dari koordinat kartesian ke dalam koordinat polar dilakukan pada setiap piksel citra batas tepi yang memiliki aras keabuan putih. Lalu pada proses deteksi dan rekonstruksi garis lurus dimulai dengan perhitungan dan penentuan titik asal terhadap titik tengah citra hasil pendeteksian tepi yang dijadikan sebagai titik referensi. Penelitian ini memiliki persentasi keberhasilan sebesar 90%. Selain itu, penelitian lainnya seperti yang dilakukan oleh Riwinoto [3] juga telah membuktikan mampu mendeteksi lingkaran menggunakan metode *Circular Hough Transform* melalui media video dimana dalam implementasinya, Riwinoto melakukan pemetaan terhadap titik-titik pada citra ke dalam *parameter space* berdasarkan suatu fungsi yang mendefinisikan bentuk yang ingin dideteksi. Selain berhasil dalam mendeteksi lingkaran, percobaan yang dilakukan oleh Riwinoto juga dapat menampilkan titik koordinat dengan tepat.

Berdasarkan latar belakang tersebut, penelitian ini menggunakan metode yang sama, akan tetapi dalam bidang yang berbeda yaitu menerapkan algoritma *Line Hough Transform* dan *Circular Hough Transform* dalam mendeteksi keberadaan garis lurus dan garis lengkung untuk nantinya dapat mengidentifikasi jenis bangun datar.

2. METODE PENELITIAN

Berikut ini tahapan-tahapan yang dilakukan dalam menerapkan algoritma *Line Hough Transform* dan *Circular Hough Transform* untuk mengidentifikasi jenis bangun datar.

2.1. Studi Literatur

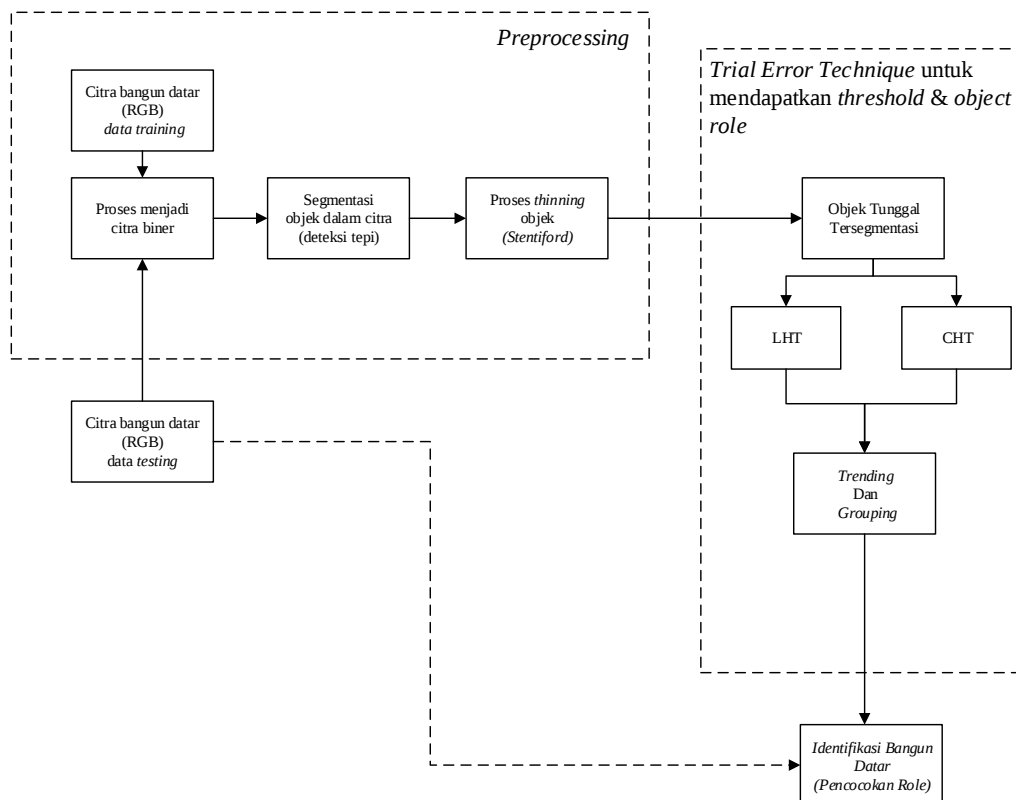
Pada tahap ini dilakukan penelusuran mengenai metode transformasi *Line Hough Transform* dan *Circular Hough Transform*. Selain itu, juga dilakukan penelusuran penelitian-penelitian terdahulu yang pernah dilakukan orang lain mengenai penggunaan algoritma *Line Hough Transform* dan *Circular Hough Transform* pada bidang serupa atau bidang lain. Tahap ini juga dilakukan untuk mengetahui sudah sejauh mana penelitian di bidang serupa telah dilakukan.

2.2. Pengumpulan Data

Pengumpulan data diambil dari gambaran tangan dalam bentuk data gambar RGB. Penelitian ini menggunakan delapan jenis citra yakni persegi, persegi panjang, lingkaran, segitiga, trapesium, belah ketupat, layang-layang, dan jajar genjang. Tiap karakter diambil 20 sampel citra dengan bentuk objek serupa yang bermacam-macam dan digunakan dalam tahap analisis kemungkinan garis dan lingkaran yang terbentuk. Kemungkinan-kemungkinan tersebut akan dipelajari dan dijadikan karakteristik dari tiap objek bangun datar. Untuk memperoleh fitur objek/ karakteristik tersebut, perlu dilakukan pencarian nilai *threshold* dan nilai *gradient* yang cocok. Dari pencarian dan perancangan nilai *threshold* yang cocok kemudian akan dirancang *role* untuk identifikasi jenis bangun datar tersebut.

2.3. Tahap Preprocessing

Tahap *preprocessing* merupakan tahapan awal dalam mengolah data sampel sebelum memasuki tahapan utama yaitu implementasi algoritma *Line Hough Transform* dan *Circular Hough Transform*. Gambar 1 merupakan tahapan-tahapan *preprocessing*.



Gambar 1 Diagram Blok Tahap *Preprocessing*

a. Tahap Deteksi Tepi

Pada tahap ini, data sampel yang berbentuk data gambar RGB akan diubah menjadi data gambar biner terlebih dahulu yaitu hanya piksel hitam dan putih sehingga diperoleh dimensi lebih kecil. *Binary image* disebut juga gambar biner yang berdasarkan suatu nilai tertentu yang menjadi tolak ukurnya yaitu nilai antara nol (putih) dan satu (hitam) [4]. Kemudian data sampel akan dilakukan pendeteksian tepi dengan mengecek setiap piksel hitam untuk mendapatkan tepi objek. Data sampel yang digunakan adalah data yang telah dibinerkan. Tahap ini menggunakan operator Sobel, dimana Operator Sobel ini terbentuk dari matriks berukuran 3x3 (Persamaan 1 dan 2).

$$G_x = \begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (1)$$

$$G_y = \begin{bmatrix} -1 & 0 & 1 \\ 2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2)$$

Sama halnya dengan metode Robert, operator Sobel juga dapat diterapkan secara terpisah untuk mendapatkan gradien horisontal dan gradien vertikalnya. Untuk mendapatkan gradien gabungan dapat dihitung dengan menggunakan penerapan (Persamaan 3 dan 4) sebagai berikut [4].

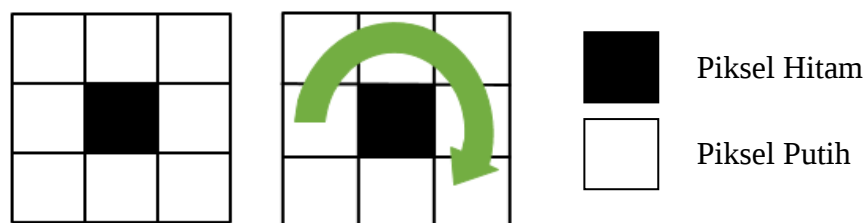
$$|G| = \sqrt{G_x^2 + G_y^2} \quad (3)$$

$$|G| = |G_x| + |G_y| \quad (4)$$

b. Tahap Segmentasi

Data sampel yang telah dideteksi tepi objeknya akan disegmentasi untuk mencari titik-titik terjauh pada objek dan akan diberi *boundries box*. Tahap ini bertujuan untuk mendapatkan objek-objek apa sajakah yang terdapat dalam citra dimana, menggunakan teknik deteksi piksel-piksel hitam pada citra. Untuk kasus ini, citra yang diolah berupa citra biner tanpa *noise*, maka digunakan algoritma untuk mencari batas *region* objek berdasarkan penelusuran piksel [5]. Algoritma yang dilakukan sebagai berikut :

1. Piksel hitam yang dimiliki pada Gambar 2, untuk menemukan piksel hitam berikutnya dilakukan pengecekan piksel-piksel tetangga satu persatu dengan urutan searah putaran jarum jam.



Gambar 2 Arah Putaran Piksel pada Citra untuk Deteksi Objek

2. Ketika piksel hitam ditemukan, maka piksel hitam akan ditandai sebagai piksel tepi dan pengujian dilanjutkan ke piksel tepi berikutnya.
3. Pada pengujian berikutnya, sudut pengujian dimulai dari sudut sebelumnya diputar 135° berlawanan arah jarum jam. Penelusuran berikutnya dilakukan terus menerus seperti ini sampai ke titik awal dan sudut pengujian awal.

c. Tahap *Thinning*

Pada tahap ini, data sampel akan dilakukan *thinning*. Tujuannya adalah mengurangi bagian yang tidak perlu (*redundant*) sehingga dihasilkan informasi yang esensial saja [4]. Metode *thinning* yang digunakan adalah *thinning Stentiford*.

Metode Stentiford ini adalah algoritma yang berlandaskan *template* di dalam gambar, maka titik yang bersangkutan akan dihapus. Selain menggunakan *template*, metode Stentiford juga menggunakan nilai konektivitas. Nilai konektivitas adalah suatu

nilai yang menyatakan jumlah objek yang terhubung dengan piksel tertentu, dimana menurut Prof. (Mrs.) A. A. Chandavale dalam jurnal yang berjudul “*Reduced Process Thinning Algorithm for CAPTCHA Strength Measurement*” nilai konektivitas dapat dihitung menggunakan Persamaan 5.

$$C_n = \sum_{k \in S} N_k - (N_k \cdot N_{k+1} \cdot N_{k+2}) \quad (5)$$

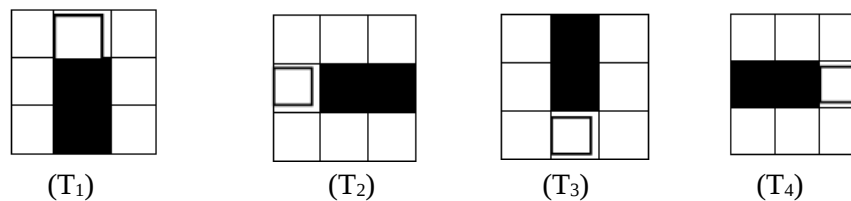
dimana :

$$S = \{1, 3, 5, 7\}$$

N_k adalah nilai warna dari delapan piksel tetangga yang diuji,

N_0 adalah nilai warna dari piksel pusat,

N_1 adalah nilai warna dari piksel di sebelah kanan dari piksel pusat dan sisanya diberi nomor dalam urutan *counter* searah jarum jam di sekitar pusat.



Gambar 3 Template untuk Thinning Metode Stentiford

Algoritma Stentiford dapat dinyatakan sebagai berikut [6] :

1. Temukan lokasi piksel (x,y) dimana piksel tersebut memenuhi *template* T_1 . *Template* ini membuat semua piksel di bagian atas gambar dihapus bergerak dari kiri ke kanan dan dari atas ke bawah.
2. Jika piksel pusat bukan merupakan titik ujung dan memiliki nilai konektivitas = 1, maka tandai piksel ini untuk dihapus kemudian. Suatu piksel dianggap sebagai piksel titik ujung apabila hanya terhubung dengan satu piksel lainnya, yaitu apabila suatu piksel hitam hanya memiliki satu tetangga hitam dari kedelapan tetangganya.
3. Ulangi langkah pertama dan kedua untuk semua piksel yang memenuhi T_1 .
4. Ulangi langkah pertama, kedua dan ketiga untuk semua *template* T_2 , T_3 dan T_4 .
 T_2 akan mencocokkan piksel pada sisi kiri objek, bergerak dari bawah ke atas dan dari kiri ke kanan.
 T_3 akan mencocokkan piksel pada sisi bawah dari citra, bergerak dari kanan ke kiri dan dari bawah ke atas.
 T_4 menempatkan piksel pada sisi kanan objek, bergerak dari atas ke bawah dan dari kanan ke kiri.
5. Ubah warna semua piksel yang sudah ditandai menjadi warna putih.

d. Grouping dan Trending

Algoritma *grouping* yang diterapkan pada penelitian ini menggunakan data berisi garis-garis dari implementasi LHT atau CHT dimana jumlah *votingnya* melebihi *threshold* diberi nama sebagai *Trimsort*. Kemudian urutkan berdasarkan r dan θ , setelah itu tentukan batas selisih dari r dan θ yang akan di-*grouping*, lalu dibuat *list* kosong untuk menampung *group-group* garis. Untuk setiap garis pada *Trimsort* dimisalkan dengan $R1$, untuk setiap *group* garis dimisalkan dengan G dan untuk setiap garis dalam *group* dimisalkan dengan $R2$. Setiap *group* dihitung jarak $R1$ dan $R2$ yang ada ditinjau dari sisi x dan y , dimana sisi x dicari dengan mengkali garis dengan $\cos$ dari sudut garis ke sumbu x dan untuk sisi y dicari dengan mengkali garis dengan $\sin$ dari sudut garis ke sumbu y . Hitung sudut terkecil antara garis lalu ambil sudut terkecil dari

selisih sudut, jika sudut lebih besar dari 90° maka sudut terkecil dikurangi 90° . Jika jarak dan sudut garis berada dalam batas yang ditentukan maka tambahkan ke dalam *group* secara berulang dan jika tidak cocok maka garis buat *group* baru dan tambahkan garis ke dalam *group* tersebut [5].

Dari algoritma tersebut masih dapat muncul *group-group* yang seharusnya dapat digabungkan, dimana antara kedua *group* yang dapat digabungkan tersebut masih terdapat R1 pada *group* satu dan R2 pada *group* yang lainnya, dengan jarak dan sudut antar garis dari R1 dan R2 masih berada dalam batas yang ditentukan. Oleh karena itu, dilakukan algoritma penggabungan *group* dengan menggunakan sistem pengecekan setiap *group* secara *looping* dengan indeks $a = 0$ sampai dengan jumlah *group* $- 1$, lalu pengecekan setiap *group* secara *looping* dengan indeks $b = a + 1$ sampai dengan jumlah *group* $- 1$. Untuk setiap garis pada *group* pertama dimisalkan dengan R1 dan untuk setiap garis pada *group* kedua dimisalkan dengan R2, dihitung jarak antar garis lalu dihitung juga sudut antar kedua garis. Jika jarak antar garis dan sudut dari kedua garis tersebut berada dalam batas yang ditentukan, maka akan digabung kedua *group* tersebut. Semua garis pada *group* kedua akan dimasukkan ke dalam *group* pertama lalu *group* kedua akan dihapus, berikut dengan *group* lainnya.

Dari proses *grouping* tersebut, akan didapatkan *group-group* garis *Hough* dimana garis – garis pada tiap *group* memiliki kemiripan jarak dan sudut sesuai dengan batasan yang ditentukan. Tiap *group* garis ini akan disimpulkan menjadi sebuah garis. Akan tetapi, karena jumlah *voting accumulator array* dari tiap garis dapat berbeda-beda, maka akan lebih akurat apabila pencarian *trend* menggunakan rumus titik berat dengan memperhatikan bobot dari tiap garis (Persamaan 6).

$$X = \frac{\sum M_i X_i}{\sum M_i} \text{ dan } Y = \frac{\sum M_i Y_i}{\sum M_i} \quad (6)$$

Setelah itu, selanjutnya digabungkan dengan metode untuk menghitung rata-rata sudut. Rumus tersebut disesuaikan untuk menghitung r dan θ dari garis-garis *Hough* dengan nilai *voting* N.

Untuk menghitung *trend* nilai R digunakan Persamaan 7 dan 8.

$$Rx = \frac{\sum M_i Rx_i}{\sum N_i} \text{ dan } Ry = \frac{\sum M_i Ry_i}{\sum N_i} \quad (7)$$

$$TrendR = \sqrt{Rx^2 + Ry^2} \quad (8)$$

Untuk menghitung *trend* nilai θ digunakan Persamaan 9 dan 10.

$$SIN = \frac{\sum N_i \sin \theta_i}{\sum N_i} \text{ dan } COS = \frac{\sum N_i \cos \theta_i}{\sum N_i} \quad (9)$$

$$Trend\theta = \tan^{-1} \left(\frac{SIN}{COS} \right) \quad (10)$$

sehingga didapat *Trend Line* : $(TrendR, Trend\theta)$.

Apabila garis-garis *Hough* dianggap sebagai suatu vektor, maka sebenarnya untuk menghitung sudut θ dapat menggunakan Persamaan 11.

$$Trend\theta = \tan^{-1} \left(\frac{Ry}{Rx} \right) \quad (11)$$

akan tetapi persamaan tersebut akan menghilangkan nilai θ apabila terdapat garis $R = 0$ pada *group* garis. Oleh karena itu, digunakanlah perhitungan $Trend\theta$ secara terpisah.

2.4. Penerapan Algoritma

Penerapan algoritma *Line Hough Transform* dan *Circular Hough Transform* untuk deteksi garis dan lingkaran yang ada dengan nilai *threshold* yang cocok untuk diimplementasikan. Melalui nilai *threshold* yang cocok akan dilakukan pengenalan pada citra yang diuji sebagai salah satu bangun datar melalui karakteristik yang sudah dicocokkan.

Algoritma *Hough Transform* menggunakan bentuk parametrik dan pemungutan suara terbanyak (*voting*) untuk menentukan nilai parameter yang tepat. Citra yang memiliki beberapa garis yang saling berpotongan pada suatu titik, maka apabila kemudian ditemukan titik tersebut ditransformasikan ke dalam ruang parameter akan didapati bahwa transformasi dalam ruang parameter adalah sebuah garis lurus dengan persamaan garis dinyatakan sebagai $y_i = mx_i + c$ yang dapat direpresentasikan sebagai titik (c, m) di ruang parameter. Untuk alasan komputasi, Duda dan Hart mengusulkan penggunaan parameter yang berbeda, yakni r dan θ dalam *Hough Transform*. Parameter r merupakan jarak antara garis dan titik asal, dan θ adalah sudut yang dibentuk dari vektor asal ke titik terdekat, yang ditulis dengan persamaan garis $r = x \cos \theta + y \sin \theta$. Penerapan algoritma ini akan dilakukan *looping* pada tiap titik dan tiap nilai *theta* dalam citra [2].

Berbeda dari *Line Hough Transform*, CHT lebih bergantung pada persamaan lingkaran. Persamaan lingkaran tersebut dapat dilihat pada Persamaan 11.

$$r^2 = (x - a)^2 + (y - b)^2 \quad (11)$$

Disini a dan b menunjukkan koordinat titik tengah, dan r adalah jari-jari lingkaran. Representasi parametrik lingkaran ini dapat dilihat pada Persamaan 12 dan 13.

$$x = a + r \cdot \cos(\theta) \quad (12)$$

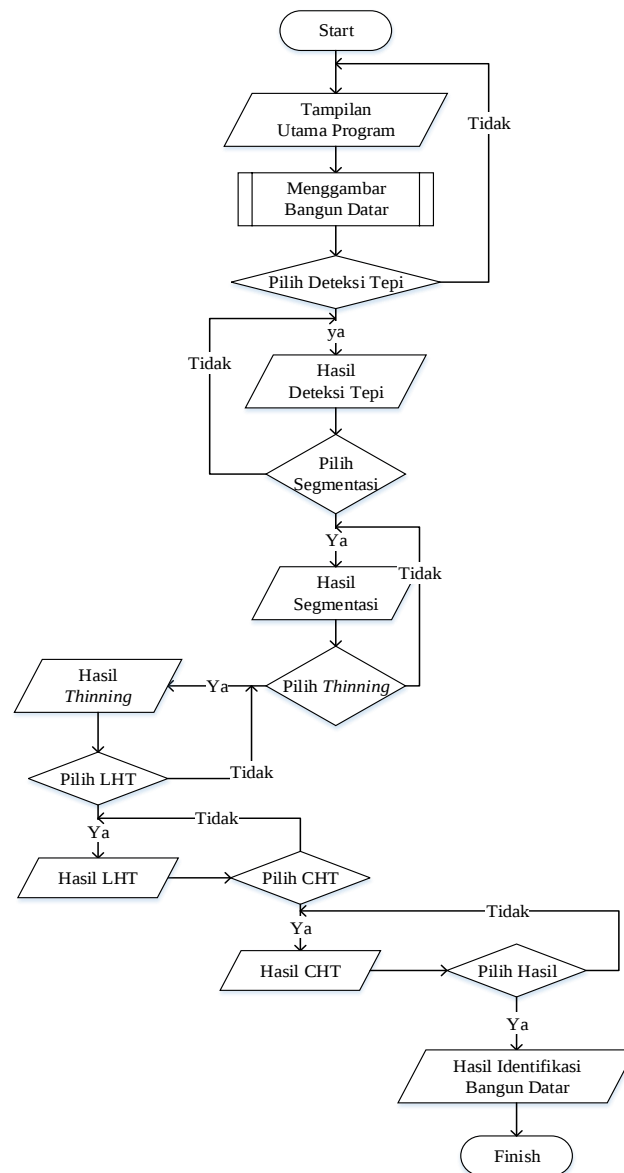
$$y = b + r \cdot \sin(\theta) \quad (13)$$

Untuk setiap titik tepi, lingkaran di gambar dengan titik itu sebagai asal dan jari-jari (r). CHT juga menggunakan *array* (tiga dimensi) dengan dua dimensi awal mewakili koordinat lingkaran dan ketiga terakhir menentukan jari-jari. Nilai-nilai dalam akumulator (*array*) meningkat setiap lingkaran digambar dengan jari-jari yang diinginkan melebihi setiap titik tepi. Akumulator terus melakukan penjumlahan akan berapa banyak lingkaran yang melewati koordinat di tiap titik tepi, hasilnya untuk menentukan jumlah tertinggi. Koordinat pusat lingkaran pada gambar adalah koordinat dengan jumlah tertinggi [3].

2.5. Uji Coba dan Analisis Hasil

Pada tahap ini akan dilakukan uji coba menggunakan data *testing* yang serupa dengan data *training* namun berbeda warna, ukuran, dan bentuk serta dibuat berdasarkan gambaran tangan. Data *testing* terdiri dari dua jenis. Pertama, data *testing* sebagian terdiri dari data *training* dan sebagian lagi terdiri dari data bukan data *training*. Kedua, data *testing* hanya terdiri dari data yang bukan data *training*.

Prosedur uji coba program menjelaskan tentang cara pengoperasian program serta tahap-tahap yang perlu dilakukan pengguna untuk menjalani aplikasi penerapan algoritma *Line Hough Transform* dan *Circular Hough Transform* dalam identifikasi jenis bangun datar. Untuk diagram alir (*flowchart*) dari aplikasi yang telah dibuat dapat dilihat pada Gambar 4.



Gambar 4 Flowchart Utama Program

Tampilan utama program memiliki beberapa *button* antara lain, tulis, hapus, cari, simpan, *fill*, deteksi tepi, segmentasi, *thinning*, LHT, CHT, dan hasil. Untuk proses menjalankan program, dimulai dengan menggambar objek.

Untuk proses mendapatkan hasil identifikasi objek, tahap demi tahap yang telah dijadikan dalam bentuk *button*. Pemberian nama *button* sesuai dengan alur proses utama algoritma. Proses pertama dimulai dengan menekan *button* Deteksi Tepi, dimana hasil yang akan diperoleh adalah seluruh objek berwarna yang ada pada *drawing area* akan berubah menjadi hitam putih dan muncul tepi – tepi dari objek tersebut. Selanjutnya tekan *button* Segmentasi, setelah *button* ditekan maka pada masing - masing objek yang ada pada *drawing area* akan di *boundaries box*, lalu satu per satu objek akan di muncul di kolom pertama *datagridview* proses.

Langkah ketiga adalah menekan *button thinning* lalu pada kolom ketiga di *datagridview* proses akan muncul hasil penipisan objek yang telah di segmentasi. Lalu langkah keempat adalah menekan *button* LHT. Sistem akan mendeteksi objek, apakah ada garis yang terdeteksi

atau tidak, jika ya maka akan muncul garis – garis pada kolom LHT. Langkah kelima adalah menekan *button* CHT, sistem akan mendeteksi garis adanya lingkaran pada objek, jika ya maka akan muncul lingkaran pada kolom CHT. Dan terakhir menekan *button* hasil, hasil pengenalan akan muncul pada kolom hasil di *datagridview* proses dan pada *textbox* hasil.

2.6. Penarikan Kesimpulan

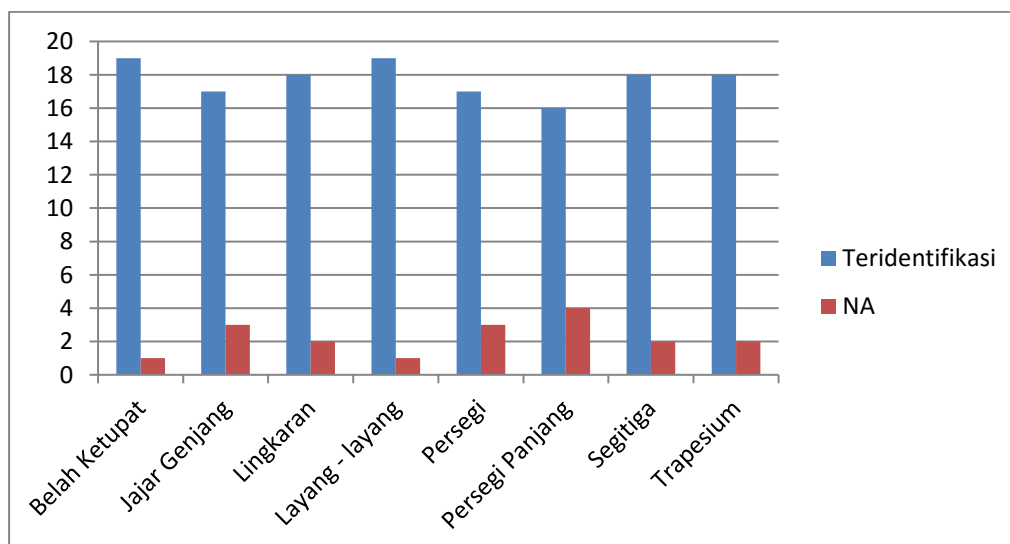
Tahap ini adalah tahap terakhir dari penelitian yaitu menilai seberapa besar keberhasilan pengukuran prediksi atau derajat yang telah dilakukan pada tahap uji coba sebelumnya dalam bentuk presentase.

4. KESIMPULAN

Dalam melakukan proses identifikasi objek, citra akan dicek satu per satu sesuai jumlah citra yang digambar. Identifikasi pertama yang dilakukan adalah mengecek apakah objek yang digambar terdeteksi sebagai *NA* berdasarkan *role* yang telah ditentukan. Selanjutnya dilakukan pengecekan bangun datar yang memiliki sifat yang mirip antara yang satu dengan yang lainnya. Pengecekan dimulai dengan jajar genjang, trapesium, persegi, dan persegi panjang. Setelah itu, dilakukan pengecekan belah ketupat lalu layang – layang, dan yang memiliki sifat yang berbeda dengan yang lainnya yaitu segitiga dan lingkaran.

Selanjutnya dilakukan pengecekan bangun datar yang memiliki sifat yang mirip antara yang satu dengan yang lainnya. Pengecekan dimulai dengan jajar genjang, trapesium, persegi, dan persegi panjang. Setelah itu, dilakukan pengecekan belah ketupat lalu layang – layang, dan yang memiliki sifat yang berbeda dengan yang lainnya yaitu segitiga dan lingkaran.

Berdasarkan hasil pengujian dari data *testing* terdapat 142 objek citra yang berhasil diidentifikasi dan 18 objek citra tidak berhasil diidentifikasi dengan persentase keberhasilan sebesar $\frac{142}{160} \times 100\% = 88,75\%$ dimana data tersebut dapat dilihat lebih jelas pada Gambar 5.



Gambar 5 Tingkat Keberhasilan Penerapan Algoritma

5. SARAN

Berdasarkan dari hasil pengujian dan analisis penelitian yang telah dilakukan, maka saran dari hasil penelitian ini adalah :

1. Penelitian dapat dikembangkan penerapan pada identifikasi bidang geometri tiga dimensi (bangun ruang), karena algoritma ini mampu mengidentifikasi bangun datar yang memiliki sifat yang serupa dengan bangun ruang.
2. Identifikasi dengan data yang serupa dapat dikembangkan dengan menggunakan algoritma yang berbeda.
3. Pengembangan selanjutnya dapat dengan mengidentifikasi bangun datar di dalam beragam objek-objek lainnya.

DAFTAR PUSTAKA

- [1] Wardhani Imalia, et al, 2010, *Makalah Matematika SMP “BANGUN DATAR”*, Universitas Muhammadiyah, Jember.
- [2] Sa’diyah, Halimatus, Isnanto R.Rizal dan, Achmad Hidayatno, 2011, *Aplikasi Transformasi Hough Untuk Deteksi Garis Lurus*, Universitas Diponegoro, Semarang.
- [3] Riwinoto, 2011, *Penggunaan Algoritma Hough Tranforms Untuk Deteksi Bentuk Lingkaran pada Ruang dua dimensi*, Politeknik Batam, Batam.
- [4] T. Sutoyo dan Mulyanto Edy, 2009, *Teori Pengolahan Citra Digital*, Andi, Yogyakarta.
- [5] Angreni Renni, 2013, *Identifikasi Perhitungan Matematika Sederhana Berbasis Tulisan Tangan Menggunakan Algoritma Line Hough Transform, Circular Hough Transform dan Segmentasi Objek*, Tesis Universitas Indonesia, Depok.
- [6] Prasetyo Eko, 2011, *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*, Andi, Yogyakarta.