

Perbandingan Metoda Klasifikasi *K-Nearest Neighbor* dan *Support Vector Machine* pada Pengenalan Benda Terhalang berbasis Kode Rantai

RAHMADI KURNIA¹, MELIA ASMITA², ROZAKY IHSAN³, IKHWANA ELFITRI⁴, DANANG KUMARA HADI⁵

^{1,2,3,4}Departemen Teknik Elektro, Universitas Andalas, Indonesia

⁵Graduate School of Science and Engineering, Ibaraki University, Jepang

Email : kurnia@eng.unand.ac.id

Received 15 Juni 2024 | Revised 20 Juli 2024 | Accepted 25 Juli 2024

ABSTRAK

Benda yang terhalang oleh benda lain memiliki bentuk yang tidak sempurna karena sebagian sisinya tidak terlihat. Untuk mengatasi permasalahan tersebut, digunakan metoda yang dapat mengenali bentuk pada benda pada sisi yang masih nampak. Penelitian ini membandingkan metoda klasifikasi K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM) berbasis kode rantai untuk mendeteksi bentuk benda terhalang. Terdapat 15 sampel untuk lima bentuk bangun datar pada 2 jenis citra benda. Hasil untuk dua jenis citra, metoda KNN memiliki rata-rata ketepatan sebesar 89,6% sedangkan metoda SVM sebesar 88,4%. Waktu komputasi citra animasi menggunakan metoda SVM lebih cepat 0,044 detik dari pada metoda KNN dan lebih cepat 0,034 detik untuk citra riil. Rata-rata memori yang digunakan dengan metoda SVM pada citra animasi lebih sedikit 0,32 Mb dari pada metoda K-NN Pada citra riil rata-rata memori yang digunakan dengan metoda SVM lebih sedikit 0,44 Mb dari metoda K-NN.

Kata kunci: transformasi Hough, kode rantai, bentuk benda, KNN, SVM

ABSTRACT

Object that are blocked by other objects have an imperfect shape because some of their side are not visible. To overcome this problem, we propose a comparison the K Nearest Neighbor classification (K-NN) and the Support Vector Machine (SVM) method based on chain code algorithm. We used 15 samples for each shape of the object for two kind of images. The result of KNN method classification has an average accuracy of 89,6%. The SVM method has an average accuracy of 88,4%. The average computing time for the SVM method is 0,044 seconds faster than KNN method for drawing image and 0,0034 seconds faster for real images, The average memory for drawing image using the SVM method is 0,32Mb less than K-NN. In the real images the average memory used with the SVM method is 0,44 Mb less than the K-NN.

Keywords: hough transform, chain code, shape object, KNN, SVM

1. PENDAHULUAN

Diantara persoalan–persoalan dalam pengenalan sebuah benda, sebuah komputer mempunyai pengetahuan yang sangat terbatas mengenai objek yang akan dikenali **(Kurnia, dkk, 2017)**, memilih metoda yang paling tepat untuk mengenali benda pada berbagai kondisi dan resolusi dari citra yang harus sesuai agar dapat dikelompokkan dengan tepat dan dengan waktu yang cepat **(Kabir, dkk, 2020)**. Penelitian sebelumnya menggunakan metoda regional Convolutional Neural Network untuk mengenali benda dengan cara menerapkan jaringan saraf konvolusional untuk melokalisasi dan mensegmentasi benda **(Girshick, dkk, 2015)**. Metoda lainnya adalah FAST Regional Convolutional Neural Network (Fast RCNN). Metoda pengenalan ini untuk mengklasifikasi benda dengan menggunakan jaringan konvolusional secara mendalam **(Girshick, 2015)**. Metoda YOLO (*You Only Look Once*) merupakan metoda lain untuk mendeteksi benda. Metoda ini menskalakan benda yang akan dikenali ke dalam ukuran yang dikenali dan membaginya menjadi beberapa kisi lalu memprediksi katagori target sesuai dengan kisi tempat pusat benda berada **(Cong, dkk, 2023)**. Pengenalan benda dengan metoda klasifikasi KNN dan SVM juga telah dilakukan sebelumnya. Tingkat akurasi yang didapatkan adalah 87% untuk metoda SVM dan 82 % untuk KNN **(Muralidharan, dkk, 2011)** **(Muralidharan & Chandrasekar, 2011)**. Namun semua metoda tersebut hanya fokus pada pengenalan benda pada kondisi tunggal di mana bentuk benda terlihat seutuhnya.

Penelitian ini fokus kepada pendeteksian benda pada kondisi terhalang oleh benda lainnya. Pada kondisi benda terhalang, maka sebagian sisi benda tidak terlihat karena terhalang oleh benda lain. Hal ini akan membuat pengenalan bentuk benda menjadi sulit dilakukan. Metoda yang digunakan untuk mendeteksi sisi dari benda yang terhalang tersebut dilakukan dengan menggunakan metoda transformasi hough dan kode rantai. Transformasi hough merupakan transformasi untuk mendeteksi garis, kurva dan lingkaran pada benda. Transformasi ini menemukan garis-garis pada objek dari suatu kelas dengan teknik *votting* **(Dalitz, dkk, 2017)** **(Duan, dkk, 2010)**. Sedangkan metoda kode rantai mendeteksi tepi dari suatu benda dengan berbasis kode-kode tertentu yang kemudian dikenali sebagai bentuk dari objek seperti yang disampaikan oleh Kahsma **(Fating, dkk, 2014)** **(Hasan, dkk, 2011)** **(Rachmawati, dkk, 2017)**.

Pada penelitian sebelumnya, kendala ketika mendeteksi benda terhalang adalah kondisi ambigu pada sistem. Kondisi ambigu terjadi ketika sistem mengenali objek dengan kesamaan ciri, ukuran, atau bentuk. Hal ini mengakibatkan sistem menjadi keliru dan bingung ketika mendeteksi benda tersebut. Terdapat dua jenis kondisi ambigu pada deteksi benda yaitu ambigu untuk dua bentuk benda dan ambigu untuk tiga bentuk benda. Ambigu untuk dua bentuk benda terjadi ketika hasil pendeteksian kode rantai dapat menggambarkan dua bentuk benda (segitiga, trapesium) sekaligus, sedangkan ambigu tiga bentuk benda terjadi karena hasil kode rantai dapat menggambarkan tiga bentuk benda sekaligus. Gambar 1 di bawah ini adalah contoh benda yang ambigu ketika terhalang.



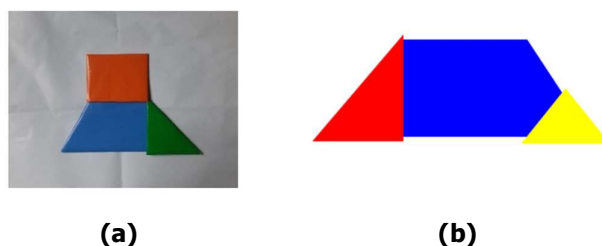
Gambar 1. Citra benda terhalang yang ambigu

Pada gambar di atas, sebuah benda berwarna kuning terhalang oleh benda lainnya sehingga sebagian sisinya tidak terlihat. Bagian sisi yang terlihat tersebut memberikan beberapa kemungkinan dari bentuk benda yang utuh. Bentuk benda utuh tersebut bisa berbentuk

segitiga sama sisi, jajaran genjang atau segitiga siku-siku. Penelitian ini mengatasi kondisi ambigu tersebut dengan menambahkan metoda klasifikasi pada hasil kode rantai. Terdapat 2 metoda klasifikasi yang dibandingkan dalam penelitian ini yaitu metoda klasifikasi *K-Nearest Neighbor* seperti yang diteliti oleh Shichao, dkk (**Zhang, dkk, 2018**) (**Zhang, 2021**) dan *Support Vector Machine* dalam riset yang dilakukan oleh C.Bo (**Bo, dkk, 2016**). Perbandingan yang dilakukan untuk melihat akurasi, kecepatan proses dan besar kapasitas media penyimpanan yang terpakai. Penelitian ini bertujuan untuk menganalisa akurasi pendeteksian benda dalam keadaan terhalang meskipun hanya sebagian sisi atau sudutnya saja yang terlihat dengan metoda klasifikasi terbaik, waktu proses tercepat dan penggunaan media penyimpanan yang paling efisien.

2. METODA PENELITIAN

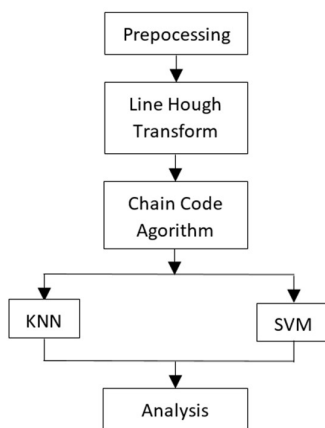
Pada penelitian ini digunakan citra animasi dan citra riil. Citra animasi adalah citra yang digambar dengan komputer dengan warna yang seragam pada keseluruhan benda. Sedangkan citra riil adalah citra yang dihasilkan dari pengambilan foto benda langsung menggunakan kamera digital. Warna benda dari citra riil ini tidak seragam karena pengaruh cahaya saat pengambilan gambar. Terdapat 5 bentuk benda yang digunakan yaitu persegi panjang, segitiga siku siku, segitiga sama sisi, trapesium dan jajaran genjang. Masing-masing bentuk terdiri dari 15 citra. Gambar 2 di bawah adalah sampel untuk benda trapesium yang terhalang.



Gambar 2. Citra input (a) Citra riil (b) Citra animasi

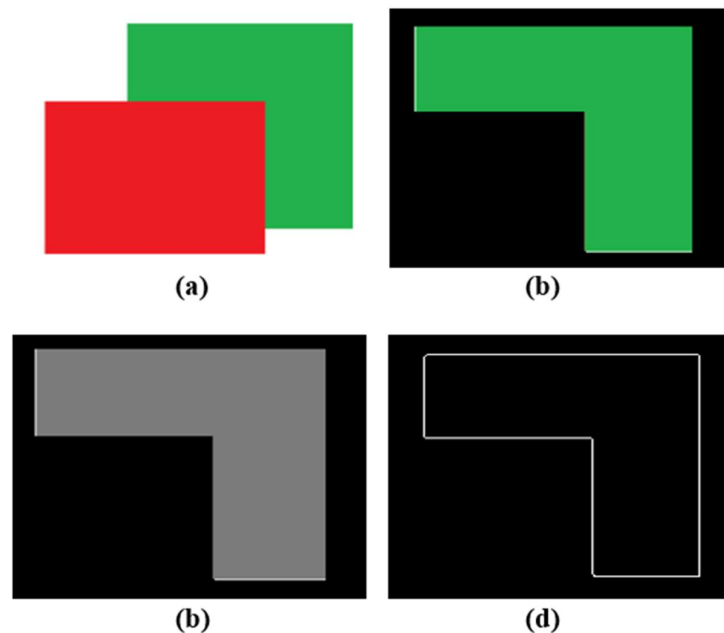
2.1 Blok Diagram Sistem

Langkah-langkah penelitian yang dilakukan dapat dilihat pada Gambar 3 berikut ini :



Gambar 3. Langkah-langkah penelitian

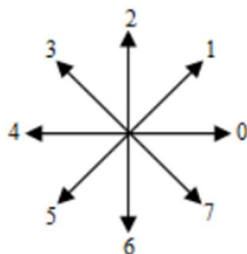
Citra warna yang merupakan input pada penelitian ini, diubah bentuknya menjadi citra grayscale. Citra grayscale ini kemudian difilter dengan detektor canny untuk mendapatkan bagian tepi objeknya. Detektor tepi *canny* merupakan salah dari filter yang paling cocok dan bagus untuk pendeteksian objek dibanding filter lainnya (Kurnia, dkk, 2017). Citra hasil filterisasi dengan menggunakan deteksi canny ini kemudian dimorphologi dengan menggunakan proses dilasi dan erosi untuk menghasilkan garis tepi yang jelas (Kumari, dkk, 2013). Hasil citra preprocessing diperlihatkan pada Gambar 4 di bawah ini :



Gambar 4. Hasil Citra Preprocessing. (a) Citra asli; (b) Hasil segmentasi dan thresholding (c) Citra gray scale (d) Citra hasil deteksi tepi

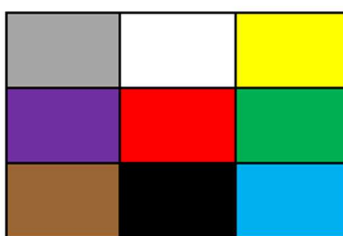
Selanjutnya setelah bagian tepi dari benda didapatkan, koordinat dari setiap pixel tepi diubah menjadi ruang parameter garis (m, c) menggunakan algoritma hough transform. Setelah semua piksel dipetakan ke dalam ruang parameter garis maka dilakukan *votting scheme* untuk memperoleh local maxima. Jika nilai *local maxima* besar dari threshold maka akan terbentuk garis pada citra. Menurut Christoph, setiap piksel pada citra menggambarkan setiap garis dalam ruang parameter dipresentasikan oleh dua parameter yaitu jarak dan sudut (Dalitz, dkk, 2017).

Dalam penelusuran batas tepi suatu benda, kode rantai menggunakan prinsip delapan arah mata angin yang ditunjukkan oleh Gambar 5 (Rachmawati, dkk, 2017). Pendeteksian kode rantai dilakukan melalui titik-titik piksel yang berdekatan. Setiap titik pada kontur objek diberi label atau kode berdasarkan arah relatif dari titik tersebut terhadap titik sebelumnya (Azmi, dkk, 2016).



Gambar 5. Skema arah mata angin dari kode rantai (Azmia, dkk, 2016)

Gambar 6 mengilustrasikan bagaimana kode rantai ini bekerja.



Gambar 6. Pixel warna sebagai ilustrasi kode rantai

Misalnya warna merah adalah hasil scan titik pertama kali. Selanjutnya apabila dari warna merah dilanjutkan ke warna biru maka proses perabaan bergerak ke arah tenggara. Arah tenggara kodenya adalah 7 pada kode rantai. Kemudian bila dari warna biru diteruskan ke warna hitam berarti proses perabaannya berpindah ke arah timur. Arah timur kodenya adalah 4 pada kode rantai. Bila dari warna hitam proses perabaannya berpindah ke arah warna ungu maka proses perabaannya bergerak ke arah barat laut. Arah barat laut pada kode rantai kodenya adalah 3. Sehingga pergerakan dari ketiga warna tersebut kodenya adalah 7 4 3. Demikian seterusnya mengikuti aturan mata rantai ini, sesuai dengan tetangga pixel yang berdekatan.

Hasil pengkodean pada berbagai jenis bangun datar pada kondisi tunggal dan kondisi terhalang ditunjukkan pada Tabel 1 berikut ini :

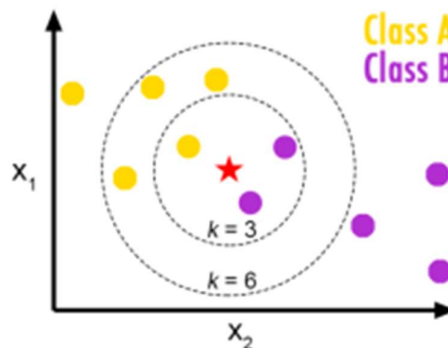
Tabel 1. Kode rantai dari beberapa bentuk benda

No	Bentuk Benda	Kondisi	
		Tunggal	Terhalang
1	Persegi Panjang	6-0-2-4	6-0-x-2-4, x-6-0-2-4, 6-0-2-x-4, x-6-0-x-2-4, x-0-2-4, 6-0-x-4, 6-x-0-x-2-x-4, 6-x-0-x-2-4, x-0-2-x-4, 6-x-0-2-4
2	Segitiga siku-siku	6-0-3	6-x-0-3, 6-0-x-3, 6-x-0-x-3.
3	Segitiga sama sisi	5-0-3	5-x-0-3, 5-0-x-3, x-0-3, 5-0-3-x, 5-x-0-x-3, 5-0-x-3-x
4	Trapesium	5-0-3-4	5-x-0-3-4, 5-0-x-3-4, 5-0-x-4, 5-0-x, 5-x-0-3-4-x, x-5-0-3-4
5	Jajaran genjang	5-0-1-4	x-0-1-4, 5-x-0-x-1-4, 5-0-x-4, x-5-0-x-1-x-4, x-0-1-x-4

* X adalah code 0 sampai 7 yang menunjukkan bagian yang benda yang terhalang sehingga arah kode rantainya berubah.

2.2. Perbandingan Metoda Klasifikasi

Setelah didapatkan serangkaian kode pada tepi benda maka tahap berikutnya adalah membuat algoritma untuk klasifikasi. Pada penelitian ini dibandingkan dua buah metoda klasifikasi yaitu K-NN dan SVM. Algoritma K-NN mengklasifikasi objek baru dari data sampel dan data latih. Data sampel yang digunakan sebanyak 250 citra animasi dan 250 citra riil. Sedangkan untuk data uji coba atau data latih digunakan citra untuk masing-masing bangun datar sebanyak sebanyak 15 sampel. Data latih dan data uji merupakan data yang terdiri dari 10 fitur diantaranya: banyaknya kode, jumlah penelusuran arah 0, jumlah penelusuran arah 1, jumlah penelusuran arah 2, jumlah penelusuran arah 3, jumlah penelusuran arah 4, jumlah penelusuran arah 5, jumlah penelusuran arah 6, dan jumlah penelusuran arah 7. Pelatihan dilakukan terhadap beberapa variasi nilai k dan menentukan nilai k terbaik dalam pengenalan bangun datar terhalang. Selanjutnya proses klasifikasi mengelompokkan jarak pixel tetangga yang merupakan data ketetanggaan untuk memprediksi nilai *instance* yang baru (Hasan, dkk, 2011). Algoritma ini akan memproyeksikan *training* sampel ke ruang yang memiliki dimensi banyak. Masing-masing dimensi merepresentasikan sebuah fitur data. Data baru dapat dikatakan masuk klasifikasi suatu kelas jika kelas itu merupakan kelas yang paling banyak ditemukan pada nilai k tetangga dari data yang tersebut (Zhang, dkk, 2018). Gambar 7 menunjukkan penentuan kelas k dari algoritma *KNN*.



Gambar 7. Penentuan kelas dari k-nearest neighbor (H. Hasan et al., 2011)

Dalam perhitungan jarak ketetanggaan, untuk melakukan klasifikasi objek baru algoritma *k-nearest neighbor* menggunakan jarak *Euclidean*. Rumus *Euclidean* dapat dilihat pada Persamaan (1) di bawah (Bo, dkk, 2016):

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

dimana:

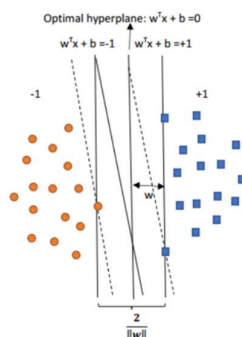
- $d(x,y)$ = jarak data
- n = dimensi data
- i = variabel data
- x_i = sampel data
- y_i = data uji atau *testing*

Setelah tetangga terdekat dipilih berdasarkan jumlah k , selanjutnya dilakukan *voting* dengan tujuan mendapatkan kelas data *testing* dari kelas terbanyak dalam k tetangga.

2.3. Support Vector Machine

Untuk memisahkan dua buah kelas dari input-input yang berbeda, *Support Vector Machine* memformulasikan *hyperplane* terbaik sebagai batas antar kelas. Penentuan formulasi

hyperplane ini dilakukan berdasar prinsip *Structural Risk Minimization* (SRM) (Hasan, dkk, 2014). Trik kernel pada ruang berdimensi tinggi merupakan algoritma yang terdapat pada metoda *Support Vector Machine* untuk melakukan klasifikasi data baik linier maupun non linier (Ritonga, dkk, 2018). Pada Gambar 8, diilustrasikan bagaimana konsep dasar dari SVM ini bekerja. Penyebaran data yang telah diinputkan sebelumnya ditunjukkan oleh warna yang berbeda. Warna-warna ini menunjukkan pada kelas mana data tersebut menjadi anggota. Misalnya data yang berwarna biru adalah anggota dari anggota kelas +1 sedangkan data yang berwarna merah adalah anggota dari kelas -1. Selanjutnya dihitung margin antara kedua kelas untuk mendapatkan *hyperplane* sebagai pemisah antar kelas. Margin yang terbesar akan membentuk *hyperplane* yang terbaik. Data yang terdekat dengan *hyperplane* disebut sebagai *support vector*.



Gambar 8. Hyperplane dari Support Vector Machine (Ritonga, dkk, 2018)

Dari ilustrasi fungsi *hyperplane* seperti pada Gambar 8, didapatkan persamaan garis regresi sebagai berikut:

$$f(x) = w^T \phi(x) + b \quad (2)$$

Pada persamaan di atas, $\phi(x)$ adalah suatu titik di dalam *feature space* F yang merupakan hasil dari pemetaan x di dalam *input space*. Selanjutnya, koefisien w dan b diestimasi. Proses estimasi dilakukan dengan cara meminimalkan fungsi resiko (*risk function*). Fungsi resiko tersebut ditunjukkan dalam persamaan-persamaan berikut:

$$\min \frac{1}{2} \|w\|^2 + C \frac{1}{\lambda} \sum_{i=1}^{\lambda} L_{\epsilon}(y_i, f(x_i)) \quad (3)$$

dengan ketentuan :

$$\begin{aligned} y_i - w\phi(x_i) - b &\leq \epsilon \\ w\phi(x_i) - y_i + b &\leq \epsilon \\ i &= 1, 2, \dots, \lambda \end{aligned} \quad (4)$$

$$L_{\epsilon}(y_i, f(x_i)) = |y_i - f(x_i)| - \epsilon \quad |y_i - f(x_i)| \geq 0 \quad (5)$$

Fungsi harus dibuat setipis mungkin agar kapasitasnya dapat dikontrol. Hal ini dapat dilakukan dengan cara meminimalkan $\|w\|^2$ dan kesalahan kesalahan empirik (*empirical error*) yang diukur dengan ϵ -insensitive lossfunction norm dari w harus diminimalkan. Cara ini juga untuk mendapatkan generalisasi yang baik pada fungsi regresi f . Pada penelitian ini, klaster yang dilakukan adalah klaster ukuran. Setelah kedua metoda klasifikasi digunakan secara terpisah

maka selanjutnya hasil dari masing-masing metoda klasifikasi tersebut dibandingkan dengan parameter : akurasi, waktu proses dan memori.

2.4. Pengujian Akurasi

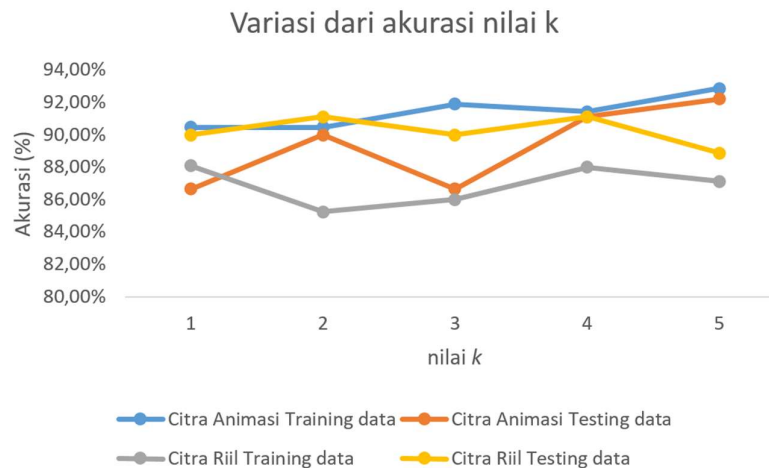
Pengujian sistem pada penelitian ini dilakukan dengan cara memvariasikan jumlah sudut yang dihalangi pada masing-masing bangun datar. Hal ini bertujuan untuk melihat kemampuan sistem dalam pengenalan bangun datar terhadap variasi keterhalangan, waktu komputasi, dan penggunaan memori. Bentuk benda yang memiliki empat sudut maka variasi keterhalangannya terdiri dari satu sudut terhalang, dua sudut terhalang, dan tiga sudut terhalang. Bentuk benda tersebut adalah persegi panjang, jajaran genjang, dan trapesium. Bentuk benda yang memiliki jumlah sudut tiga maka variasi keterhalangannya terdiri dari satu sudut terhalang dan dua sudut terhalang. Bangun datar yang memiliki tiga sudut adalah segitiga sama sisi dan segitiga sama sisi. *Confusion matrix* adalah metode untuk mengevaluasi model klasifikasi dengan melakukan perkiraan terhadap benar atau salahnya benda yang dikelompokkan. Kelas asli dari data yang diinputkan dibandingkan dengan sebuah matrik prediksi (Bo, dkk, 2016). Rumus perhitungan tingkat akurasi pada *confusion matrix* dapat dilihat pada Persamaan (6) di bawah (Junaidi, dkk, 2020):

$$Akurasi = \frac{Juml \ data \ benar}{Total \ Data} \times 100\% \quad (6)$$

3. HASIL DAN PEMBAHASAN

3.1 Hasil Nilai Terbaik k untuk KNN

Pengujian model KNN bertujuan untuk mengukur dan mengevaluasi kinerja serta keandalan model terhadap data yang tidak pernah dilihat sebelumnya selama proses pelatihan. Selain itu, pengujian juga menentukan pemilihan parameter nilai k yang optimal untuk diterapkan dalam model KNN. Pada proses pengujian ini menggunakan citra animasi dan citra riil yang dilakukan pengujian secara terpisah. Pengujian dilakukan pada setiap model KNN yang telah dilatih sebelumnya dengan variasi k yang sama pada proses pelatihan. Adapun hasil pengujian tersebut dapat dijabarkan oleh Gambar 9:

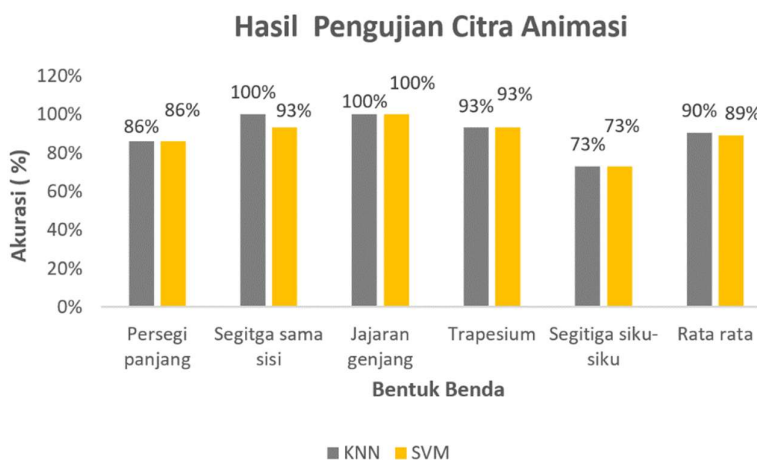


Gambar 9. Pengujian nilai k pada metoda KNN

Untuk citra animasi, akurasi terbaik mencapai 92,86 % untuk training data dan 92,22 % untuk testing data saat $k = 5$. Sedangkan untuk citra riil, akurasi terbaik mencapai 88% untuk training data saat $k = 4$ dan 91,11% untuk testing data saat $k = 4$.

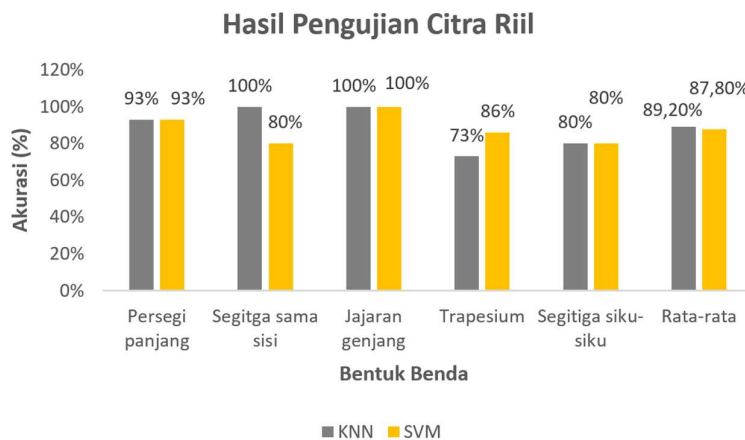
3.2 Hasil Pengujian Sistem

Perbandingan akurasi metoda klasifikasi KNN dengan metoda SVM untuk citra animasi ditunjukkan pada Gambar 10:



Gambar 10. Hasil pengujian akurasi citra animasi

Akurasi tertinggi baik dengan menggunakan metoda KNN maupun SVM terdapat pada bangun jajaran genjang. Sedangkan akurasi yang terendah pada segitiga sama sisi yaitu sebesar 73%. Hal ini disebabkan bentuk segitiga sama sisi pada kondisi terhalang memiliki kondisi yang hampir menyerupai bentuk bangun datar yang lain. Metoda KNN memiliki akurasi rata-rata untuk seluruh bangun datar sebesar 90 %. Sedangkan akurasi rata-rata metoda klasifikasi SVM yaitu sebesar 89%. Hasil yang diperoleh ini menunjukkan kesamaan dengan hasil pada metoda yang dilakukan oleh Kurnia (**Kurnia, dkk, 2017**). Pada bangun datar segitiga sama sisi dan jajaran genjang, menghasilkan pengenalan yang tepat untuk keseluruhan sampel. Kedua jenis bangun datar tersebut memiliki pola kode rantai yang jauh berbeda dengan bangun datar lainnya sehingga ketika dieksttrak, jumlah fitur setiap perubahan arahnya maka juga sangat berbeda dengan bangun yang lain. Hasil pengujian akurasi untuk citra riil dapat dilihat pada Gambar 11.



Gambar 11. Hasil pengujian akurasi citra riil

Dari hasil pengujian akurasi, didapatkan nilai yang hampir sama dari kedua metoda klasifikasi ini. Pada bentuk segitiga sama sisi, hasil klasifikasi dengan metoda KNN memiliki akurasi 100% . Sedangkan metoda SVM memiliki akurasi 80%. Pada trapesium, hasil metoda klasifikasi SVM memiliki akurasi 86%. Hasil ini lebih baik daripada akurasi dengan metoda KNN yaitu 73%. Hasil akurasi citra riil lebih rendah daripada citra animasi pada bentuk trapesium dikarenakan segmentasi yang tidak sempurna yang terjadi pada tahapan preprocessing untuk keempat sudut trapesium. Hasil segmentasi ini berpengaruh pada deteksi tepi dan pengkodean oleh transformasi hough sehingga akurasinya menjadi lebih rendah pada citra yang memiliki segmentasi yang lebih baik pada citra animasi trapesium.

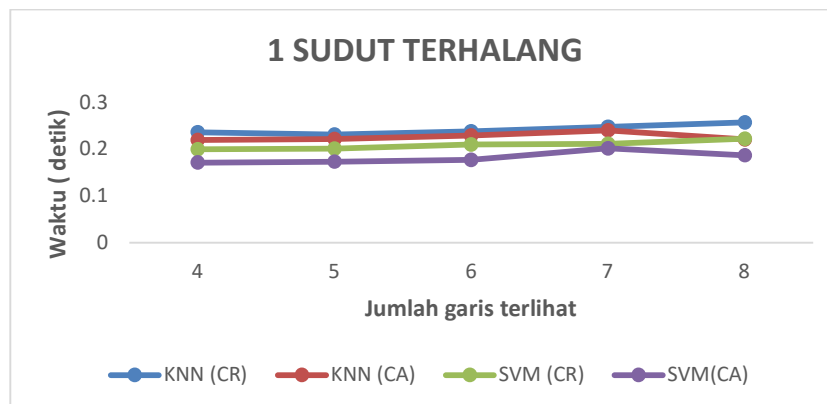
Rata rata akurasi dari metoda KNN adalah 89,2 % sedangkan rata-rata akurasi dari metoda SVM sebesar 87,8 %. Hasil ini jauh lebih baik dari metoda sebelumnya yaitu sebesar 83,57% dan berhasil mengatasi persoalan ambigu pada pengenalan terhalang. Ketika proses segmentasi citra, citra riil memiliki kontur objek yang kasar dibandingkan dengan citra animasi. Hal ini diakibatkan adanya faktor intensitas cahaya ketika mengakuisisi citra tersebut. Hasil segmentasi yang kurang sempurna tersebut mengakibatkan hasil deteksi tepi citra riil tidak sehalus citra animasi.

Pada penelitian ini penggunaan algoritma *line hough transform* dapat menyempurnakan hasil deteksi tepi *canny* yang kurang sempurna. Hal ini sesuai dengan kelebihan algoritma ini yang mampu mendeteksi garis walaupun citra tersebut tidak memiliki garis lurus yang sempurna atau ber-*noise*. Dengan cara ini, transformasi *hough* membantu menemukan garis lurus dalam citra meminimalisir kesalahan penelusuran kontur oleh algoritma *kode rantai*. Untuk kedua jenis citra, maka rata rata akurasi metoda klasifikasi KNN adalah sebesar 89,6% dan SVM sebesar 88,4 %. Hasil efisiensi waktu proses deteksi dari berbagai sudut keterhalangan benda ditunjukkan pada Gambar 12.

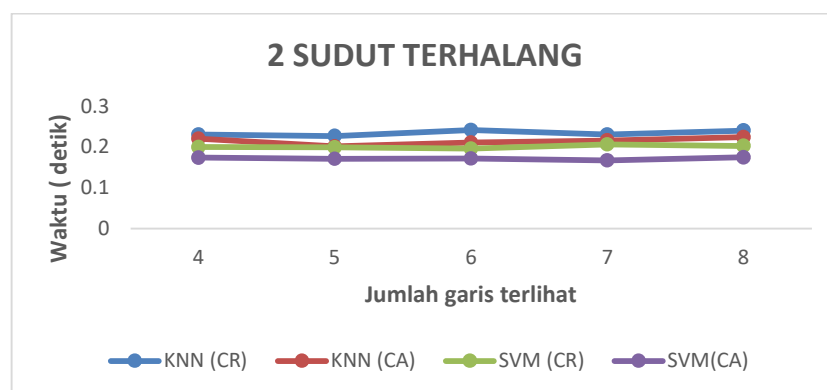
Pada Gambar 12, KNN (RI) adalah hasil dari KNN untuk citra riil, KNN (CA) adalah hasil dari KNN untuk citra animasi, SVM (CA) hasil dari SVM untuk citra animasi, SVM (RI) adalah hasil dari SVM untuk citra riil. Secara umum waktu rata-rata komputasi dari citra animasi lebih cepat dari pada waktu komputasi citra riil. Hal ini dikarenakan proses segmentasi pada citra riil lebih kompleks dari pada citra animasi. Citra animasi memiliki warna-warna yang lebih homogen daripada citra riil sehingga proses segmentasinya lebih cepat. Pada penelitian ini diujicobakan 3 variasi keterhalangan sudut dari benda dengan jumlah garis yang terlihat sebanyak 4, 5, 7 and 8 garis.

Pada Gambar 12 (a), untuk kondisi 1 sudut terhalang, waktu komputasi pada citra animasi dan citra riil didapatkan waktu terendah dengan menggunakan metoda klasifikasi SVM. Rata-rata selisih waktu komputasi antara metoda klasifikasi SVM dan KKN untuk citra animasi adalah 0,044 detik. Sedangkan untuk citra riil metoda klasifikasi SVM rata-rata lebih cepat 0,033 detik daripada metoda klasifikasi KNN. Untuk kondisi 2 sudut terhalang seperti terlihat pada Gambar 12(b) , Rata-rata selisih waktu komputasi antara metoda klasifikasi SVM dan KKN untuk citra animasi adalah 0,043 detik. Sedangkan untuk citra riil metoda klasifikasi SVM rata-rata lebih cepat 0,033 detik daripada metoda klasifikasi KNN. Untuk kondisi 3 sudut terhalang seperti terlihat pada Gambar 12 (c), rata-rata selisih waktu komputasi antara metoda klasifikasi SVM dan KKN untuk citra animasi adalah 0,037 detik. Sedangkan untuk citra riil metoda klasifikasi SVM rata-rata lebih cepat 0,035 detik daripada metoda klasifikasi KNN. Hal ini disebabkan metoda KNN menggunakan lebih banyak tetangga untuk proses klasifikasinya sehingga membutuhkan waktu proses yang lebih lama dari metoda klasifikasi SVM yang hanya membentuk *hyperplane* untuk klasifikasi hasil dari kode rantai.

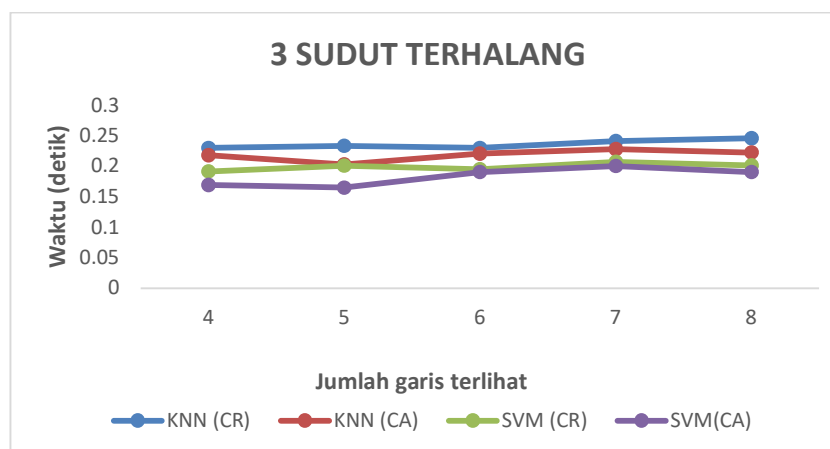
Perbandingan Metoda Klasifikasi *K-Nearest Neighbor* dan Support Vector Machine pada Pengenalan Benda Terhalang berbasis Kode Rantai



(a)



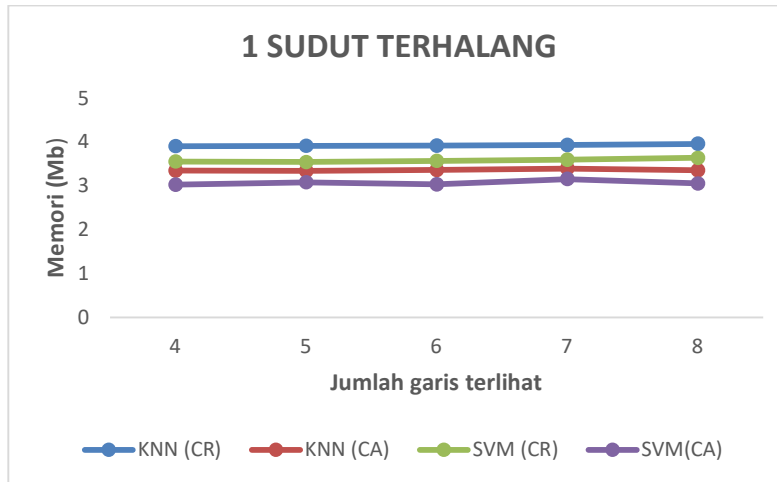
(b)



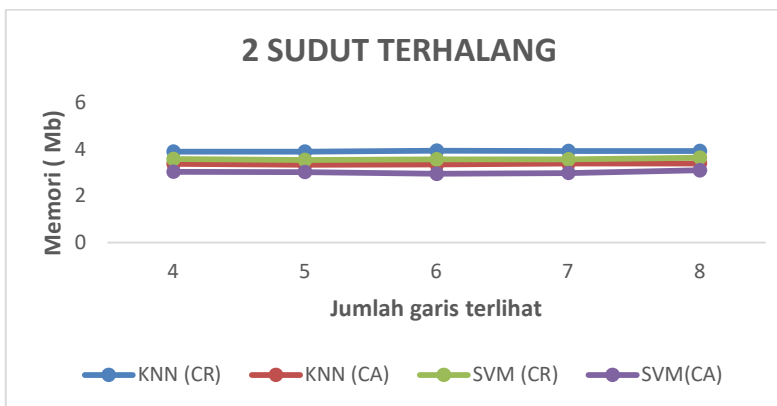
(c)

Gambar 12. Hasil pengujian waktu proses

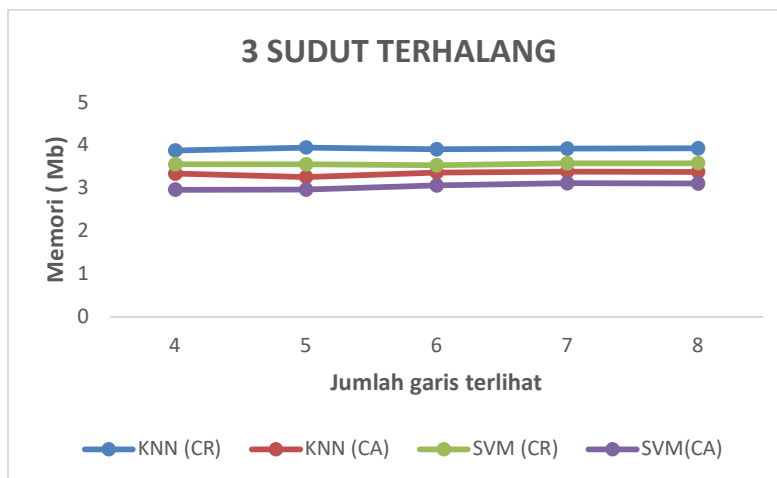
Secara umum memori yang terpakai untuk memproses citra animasi lebih sedikit dari pada memory yang terpakai untuk memproses citra riil. Hal ini dikarenakan citra riil memiliki kombinasi warna yang lebih banyak dari pada citra animasi. Warna-warna ini akan memerlukan memori yang banyak pada preprocessing dan proses pengkodeannya. Pada percobaan 3 variasi keterhalangan sudut dari benda dengan jumlah garis yang terlihat sebanyak 4, 5, 7 dan 8 garis. Hasilnya diperlihatkan oleh Gambar 13:



(a)



(b)



(c)

Gambar 13. Hasil pengujian memori yang digunakan

Pada Gambar 13 (a) untuk kondisi 1 sudut terhalang, besar memori yang paling sedikit pada citra animasi dan citra riil didapatkan pada proses klasifikasi dengan menggunakan metoda

SVM. Rata-rata selisih memori yang digunakan antara metoda klasifikasi SVM dan KNN untuk citra animasi adalah 0,3 Mb. Sedangkan untuk citra riil metoda klasifikasi SVM rata-rata memori yang digunakan lebih sedikit 0,35 Mb daripada metoda klasifikasi KNN. Untuk kondisi 2 sudut terhalang seperti terlihat pada Gambar 13 (b), Rata-rata selisih memori yang terpakai antara metoda klasifikasi SVM dan KNN untuk citra animasi adalah 0,346 Mb. Sedangkan untuk citra riil metoda klasifikasi SVM rata-rata memori yang terpakai lebih dikit 0,336 Mb daripada metoda klasifikasi KNN. Untuk kondisi 3 sudut terhalang seperti terlihat pada Gambar 13 (c), Rata-rata memori yang terpakai antara metoda klasifikasi SVM dan KNN untuk citra animasi adalah 0,3 Mb. Sedangkan untuk citra riil metoda klasifikasi SVM menggunakan memori rata-rata lebih cepat 0,356 Mb daripada metoda klasifikasi KNN. Metoda KNN menggunakan lebih banyak tetangga pixel untuk proses klasifikasinya dan terdapat proses votting untuk penentuan klasifikasinya. Hal inilah yang menyebabkan penggunaan memori pada metoda klasifikasi KNN lebih besar dari pada metoda klasifikasi SVM.

4. KESIMPULAN

Pengenalan bentuk benda terhalang dengan menggunakan metoda klasifikasi KNN dan SVM dengan berbasis kode rantai berhasil meningkatkan akurasi dan efisiensi sistem. Akurasi rata-rata dari metoda klasifikasi KNN sebesar 89,6% dan SVM sebesar 88,4 %. Hasil ini lebih baik dari metoda yang dilakukan oleh Kurnia,dkk yang hanya memiliki akurasi rata rata sebesar 83,57 %. Hasil akurasi yang diperoleh ini juga lebih baik dari hasil penelitian yang dilakukan oleh Muralidharan, dkk yang menghasilkan akurasi sebesar 87 % untuk klasifikasi dengan metoda SVM dan 72 % untuk klasifikasi dengan menggunakan metoda KNN pada kondisi benda tunggal. Hasil akurasi ini juga berhasil menyelesaikan persoalan benda ambigu yang terdapat pada penelitian sebelumnya oleh Kurnia, dkk. Akurasi metoda klasifikasi KNN lebih baik dari metoda SVM baik untuk citra riil maupun citra animasi. Waktu proses dan penggunaan memori, dengan metoda klasifikasi SVM lebih baik dan lebih cepat dari pada metoda klasifikasi KNN.

DAFTAR RUJUKAN

- Azmia, A. N., Nasien, D., & Samah, A. A. (2016). Freeman chain code as representation in offline signature verification system. *Jurnal Teknologi*, 78(8-2), 89-94. <https://doi.org/10.11113/jt.v78.9546>
- Bo, C., Lu, H., & Wang, D. (2016). Hyperspectral Image Classification via JCR and SVM Models with Decision Fusion. *IEEE Geoscience and Remote Sensing Letters*, 13(2), 177-181. <https://doi.org/10.1109/LGRS.2015.2504449>
- Cong, X., Li, S., Chen, F., Liu, C., & Meng, Y. (2023). A Review of YOLO Object Detection Algorithms based on Deep Learning. *Frontiers in Computing and Intelligent Systems*, 4(2), 17-20.
- Dalitz, C., Schramke, T., & Jeltsch, M. (2017). Iterative hough transform for line detection in 3D point clouds. *Image Processing On Line*, 7, 184-196. <https://doi.org/10.5201/ipol.2017.208>
- Duan, D., Xie, M., Mo, Q., Han, Z., & Wan, Y. (2010). An improved Hough transform for line

- detection. *ICCASM 2010 - 2010 International Conference on Computer Application and System Modeling, Proceedings*, 2, 354–357. <https://doi.org/10.1109/ICCASM.2010.5620827>
- Fating, K., & Ghotkar, A. (2014). Performance Analysis of Chain Code Descriptor for Hand Shape Classification. *International Journal of Computer Graphics & Animation*, 4(2), 9–19. <https://doi.org/10.5121/ijcga.2014.4202>
- G, A., T, H. N., Kumari, J., & M, S. (2013). Analysis of Digital Images Using Morphological Operations. *International Journal of Computer Science and Information Technology*, 5(1), 145–159. <https://doi.org/10.5121/ijcsit.2013.5112>
- Girshick, R. (2015). Fast R-CNN. *Proceeding of IEEE International Conference of Computer Vision*, 1440–1448.
- Hasan, H., Haron, H., & Hashim, S. Z. M. (2011). Heuristic algorithm to generate modified freeman chain code from thinned binary image. *Australian Journal of Basic and Applied Sciences*, 5(11), 752–762.
- Hasan, M. A. M., Nasser, M., Pal, B., & Ahmad, S. (2014). Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS). *Journal of Intelligent Learning Systems and Applications*, 06(01), 45–52. <https://doi.org/10.4236/jilsa.2014.61005>
- Junaidi, M., & Khuzaini, K. (2020). Big Data Analysis Model Profitability Ratio in Determining Prediction of Company Performance Era 4.0. *1st International Conference of Business ...*, 793–807.
- Kabir, M. M., Ohi, A. Q., Rahman, M. S., & Mridha, M. F. (2020). An Evolution of CNN object classifiers on low-resolution images. *HONET 2020 - IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI*, 209–213. <https://doi.org/10.1109/HONET50430.2020.9322661>
- Kurnia, R., Asmita, M., & Elfitri, I. (2017). Object detection on hindered condition by using chain code-based angle detection. *ACM International Conference Proceeding Series, 2017-October*, 50–56. <https://doi.org/10.1145/3145777.3145780>
- Muralidharan, R., & Chandrasekar, C. (2011). Object Recognition using SVM-KNN based on Geometric Moment Invariant. *International Journal of Computer Trends and Technology- July to Aug Issue*, 215–220. <http://www.internationaljournals.org>
- Rachmawati, E., Khodra, M. L., & Supriana, I. (2017). Shape based recognition using freeman chain code and modified Needleman-Wunsch. *Proceedings of 2016 8th International Conference on Information Technology and Electrical Engineering: Empowering Technology for Better Future, ICITEE 2016*, (pp. 1–6).

<https://doi.org/10.1109/ICITEED.2016.7863307>

Ritonga, A. S., & Purwaningsih, E. S. (2018). Penerapan Metode Support Vector Machine (SVM) Dalam Klasifikasi Kualitas Pengelasan Smaw (Shield Metal Arc Welding). *Ilmiah Edutic*, 5(1), 17–25.

Zhang, S. (2021). Challenges in KNN Classification. *IEEE Transactions on Knowledge and Data Engineering*, 34(10). <https://doi.org/10.1109/TKDE.2021.3049250>

Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2018). Efficient kNN classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5), 1774–1785. <https://doi.org/10.1109/TNNLS.2017.2673241>