

Perancangan Prototype Sistem Kunci Pintu Dengan Pola Jari Tangan Menggunakan Raspberry Pi Berbasis *Video Processing*

I Komang Wahyudi Putra Mahendra^{1,*}, Ratri Dwi Atmaja¹, Unang Sunarya²

1 Universitas Telkom, Fakultas Teknik Elektro, Terusan Buah Batu, Bandung

2 Universitas Telkom, Fakultas Ilmu Terapan, Terusan Buah Batu, Bandung

* E-mail : putra.mahendra@outlook.com

Abstrak. Pintu menjadi sebuah akses orang untuk keluar ataupun masuk rumah. Pada umumnya orang menggunakan kunci konvensional yang rentan hilang, rusak, ataupun disalahgunakan oleh orang lain, hal ini tentu menjadi dirasa kurang praktis dan fleksibel. Maka dari itu dibuatlah sistem yang dapat menggantikan fungsi dari kunci tersebut, dengan perkembangan teknologi memungkinkan adanya interaksi antara manusia dengan computer melalui media visul, dengan kombinasi angka unik yang dibentuk dari pola jari tangan menjadi sistem kunci yang fleksibel dan praktis. Pada pengolahan citra sendiri digunakan dengan perangkat *Raspberry Pi* karena dimensi yang kecil dan performa yang spesifikasi yang cukup tinggi, ditambah dengan adanya kontroler di dalamnya. Metode yang digunakan untuk mengenali pola jari tangan adalah dengan *convex hull* dan *convexity defect*, maka dari citra akan menghasilkan angka dari pola tangan. Hasil dari perancangan ini adalah sistem yang dapat bekerja mengenali pola jari tangan, dengan terdapat lima pola berbeda. Pengujian dilakukan pada beberapa kondisi berbeda, pada pengujian algoritma penyederhaan garis Douglas-Peucker, nilai epsilon terbaik pada $0.01 \times$ panjang kurva, dan akurasi dari pengenalan pola mencapai 100% pada kondisi jarak kurang dari 95 cm tangan terhadap kamera, cahaya ruangan pada lux 22-25, dan posisi tangan tegak lurus terhadap kamera.

Kata Kunci : *Convex Hull*, *Convexity Defect*, Pintu, Kontroler, Pengolahan Citra

1. Pendahuluan

Pintu adalah bagian rumah sebagai akses untuk keluar ataupun masuk. Sekarang umumnya orang menggunakan kunci konvensional untuk mengunci pintu, hal ini tentu akan merepotkan jika kunci menghilang, ataupun disalahgunakan oknum tidak bertanggung jawab, hal tersebut dirasa kurang fleksibel dan praktis ketika harus menggunakan kunci konvensional, dengan perkembangan teknologi dimungkinkan untuk melakukan interaksi manusia dengan komputer melalui media visual [1][2][3] ini untuk mendukung berbagai kebutuhan manusia untuk lebih praktis dan fleksibel.

Penelitian sebelumnya terdapat penguncian pintu dengan menggunakan pengenalan suara ataupun pengenalan wajah, tetapi masih terdapat *delay*, dikarenakan waktu pemrosesan ataupun adanya kesalahan pengenalan [4][5]. Pada penelitian kali ini akan mengembangkan sistem pintu dengan penguncian pintu berupa pola jari tangan, pola ini dapat dilakukan di depan kamera yang sudah terhubung dengan *Raspberry Pi* sebagai perangkat untuk melakukan pengolahan citra pada sisi kontroler akan bertugas dalam mengontrol sensor gerak, *buzzer* dan lampu led sebagai tanda adanya inputan berupa citra pola jari tangan. Digunakannya *Raspberry Pi* dikarenakan perangkat ini mempunyai dimensi yang kecil tapi dapat bekerja layaknya komputer untuk melakukan pemrosesan sinyal dan dapat juga sebagai kontroler. Pada proses rekognisi dilakukan pengenalan pola dengan *convex hull* dan *convexity defect*, sebelum ke proses *convexity defect* dilakukan proses *convex hull* dengan algoritma Skalsky, penelitian sebelumnya menggunakan metode ini memberikan hasil yang bagus ketika diterapkan pada objek polygon berbentuk bintang sehingga sesuai dengan objek penelitian ini yaitu polygon berbentuk tangan [3][6]. Hasil dari pemrosesan ini berupa jari berupa angka yang dapat digunakan sebagai kata sandi yang unik. Dengan penggunaan metode *convex hull* dan *convexity defect* tersebut memberikan akurasi yang tinggi dengan waktu proses yang singkat, agar nanti dapat diterapkan dalam penggunaan sehari hari [1].

2. Dasar Teori dan Perancangan

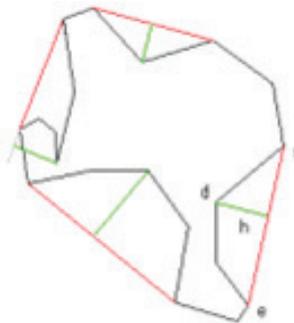
2.1 Convex Hull [6][13]

Convex Hull adalah poligon yang disusun dari subset titik sedemikian sehingga tidak ada titik dari himpunan awal yang berada diluar poligon tersebut (semua titik berada di batas luar atau di dalam area yang dilingkupi oleh poligon tersebut). *Simple polygon* adalah bentuk dua dimensi yang mempunyai banyak sudut dimana tidak terdapat perpotongan antara sudutnya. Setiap *simple polygon* mempunyai daerah dalam dan daerah luar. Sebuah *simple polygon* dikatakan *convex* jika besarnya derajat dalam yang terbentuk untuk setiap sudut lebih kecil dari 180 derajat.

Convex hull dari sebuah polygon P adalah daerah terkecil dari *convex* polygon dimana melingkupi polygon P. Bisa juga dikatakan *rubber band* yang menutupi sekeliling P. *Convex hull* dari sebuah *convex* polygon P adalah P itu sendiri.

2.2 Convexity Defect [6][14]

Convexity Defect adalah sebuah ruang antara garis kontur dengan objek actual. *Defect* tersebut berguna untuk menemukan *feature* pada sebuah poligon, salah satunya yaitu untuk mendeteksi jari tangan manusia.



Gambar 1 Convexity defect sebuah poligon

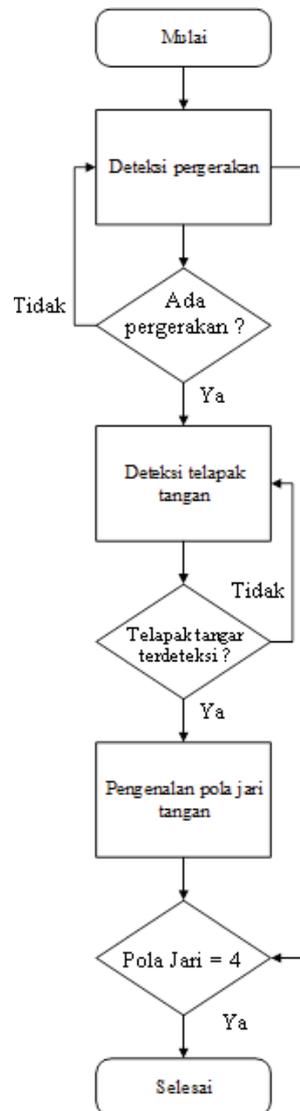
Dari Gbr. 1 di atas, terlihat *convex hull* digambarkan dengan garis merah yang menyelubungi poligon dengan garis konturnya yang berwarna hitam. Simbol "s" dan "e" menunjukkan "start point" dan "end point" dari *convexity defect* tersebut. Sedangkan simbol "d" melambangkan "depth point", yaitu titik kontur yang terletak antara "s" dan "e" yang merupakan titik terjauh antara kontur dengan garis *convex hull* yang dilambangkan dengan "se". Simbol "h" yaitu "depth" atau kedalaman dari *defect* yang merupakan jarak dari "d" hingga garis "se". *Start point*, *end point*, *depth point*, dan *depth*, keempat elemen tersebut yang akan digunakan untuk dapat menemukan *feature* dari sebuah poligon sehingga dapat diolah lebih lanjut.

2.3 Algoritma penyederhanaan garis Douglas-Peucker [15]

Algoritma penyederhanaan garis Douglas-Peucker merupakan salah satu metode yang diakui sebagai salah satu metode yang terbaik dalam memberikan representasi terbaik dari garis aslinya. Algoritma ini banyak digunakan dalam bidang computer grafis dan sistem informasi geografis. Ada dua varian dari algoritma ini, metode O asli (nm) dimana n adalah jumlah n jumlah simpul input dan m adalah jumlah segmen output dan dapat bekerja pada dimensi berapapun, dan O (n log n) yang hanya dapat bekerja pada bidang polyline bidang 2D yang sederhana. Dari kedua varian tersebut garis disederhanakan dengan memberikan pendekatan pada suatu nilai, apabila nilai tersebut dirasa tidak memenuhi maka akan dilakukan penyederhanaan garis.

3 Pemodelan Sistem

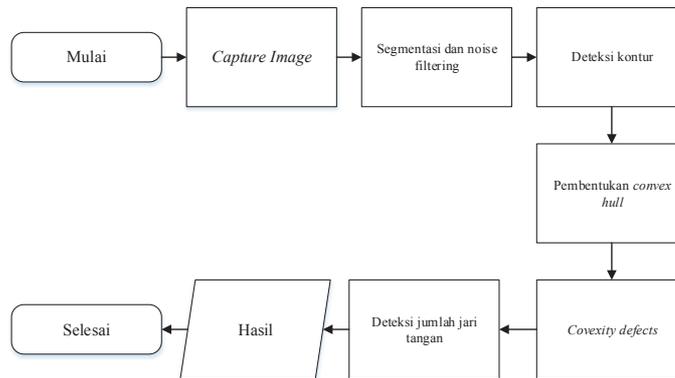
3.1 Diagram Alir pada Sistem



Gambar 2. Diagram Alir Pada Sistem

Pada sistem dengan sensor PIR alat akan melakukan deteksi pergerakan sekitar daerah di depan sensor kurang lebih batasan daerah dari sensor PIR adalah 10 meter , jika terdapat pergerakan yang dideteksi oleh sensor maka buzzer akan berbunyi. Jika sudah berbunyi maka secara otomatis kamera langsung aktif , pada tahap setelah deteksi gerakan , maka terdapat deteksi telapak tangan , deteksi telapak tangan menggunakan metode *Haar Cascade* , kembali buzzer akan berbunyi ketika di deteksi adanya telapak tangan yang tertangkap kamera. Hasil tangkapan dari telapak tangan ini akan membentuk region untuk diolah agar mendapatkan pola jari tangan. Pada deteksi pola jari ada beberapa tahapan yang akan dijelaskan , hasil dari deteksi pola jari tangan adalah jumlah jari tangan yang disesuaikan dengan pola yang diberikan , jika terdapat pola yang dikenai maka buzzer akan berbunyi , pola yang digunakan pada penelitian ini antara lain pola 1,2,3,4,5 yang disesuaikan dengan pola tangan. Apabila sistem sudah mendeteksi ada empat pola yang dideteksi maka sistem selesai mendeteksi.

3.2 Diagram Alir pada pengenalan pola jari tangan



Gambar 3 Diagram Alir Pada Pengenalan Pola

a. *Segmentasi dan Noise Filtering*

Pada bagian ini meliputi beberapa bagian antara lain :Cropping dilakukan dari hasil deteksi telapak tangan kemudian dibentuk region yang melingkupi keseluruhan tangan untuk di deteksi jumlah jarinya.*Resize Image*Proses ini dilakukan untuk mendapatkan citra 320 x 240 agar tingkat akurasi lebih tinggi dan lebih mudah memproses.*Noise Filtering* di sini menggunakan Gaussian Blur , agar citra hasil segmentasi hasilnya lebih bersih dari noise.

b. *Deteksi kontur*

Deteksi kontur dilakukan dengan melihat intensitas warna yang sama , setelah dirubah ke binary tadi , bagian citra yang bernilai 255 akan di label sebagai kontur , kemudian dari hasil pelabelan td , diseleksi bagian yang terbesar konturnya , bagian tersebut dianggap sebagai tangan.

c. *Pembentukan convex hull*

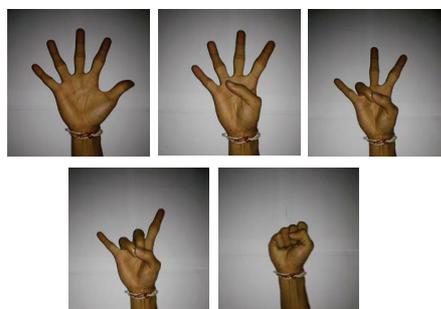
Dari hasil kontur yang telah didapat akan didapatkan titik pada citra untuk membentuk *convex hull* , sebelumnya pada titik ini dilakukan penerapan algoritma Douglas-Peucker untuk mengurangi jumlah titik , namun titik hasil penerapan algoritma tetap merepresentasikan dari kontur aslinya .Pengurangan titik ini dilakukan untuk meningkatkan akurasi dan mengurangi waktu proses.

d. *Convexity defects*

Pada bagian ini , dari lingkup dari *convex hull* akan dilakukan proses *convexity defects* untuk mendapatkan jumlah jari tangan yang sesuai dengan citra.

e. *Deteksi jumlah jari tangan*

Dari hasil *convexity defects* akan dihasilkan jumlah jari tangan.



Gambar 4 Contoh Pola Jari Tangan

4 Analisis dan Hasil Simulasi

4.1 Pengujian terhadap penggunaan algoritma penyederhanaan garis Douglas-Peucker

Penggunaan algoritma garis Douglas Peucker dapat memberikan representasi garis yang sesuai dari garis aslinya , dengan menentukan nilai epsilon yang tepat maka jumlah garis yang dibentuk dapat disesuaikan.

Tabel 1 Penggunaan Algoritma Douglas-Peucker Konstanta 0.01

| No. | Citra uji | Panjang curva | Jumlah point awal | Konstanta | Nilai epsilon | Setelah penyederhanaan | Jumlah jari di deteksi | Jumlah jari pada gambar |
|-----|------------|---------------|-------------------|-----------|---------------|------------------------|------------------------|-------------------------|
| 1 | uji_5.jpeg | 1278.212 | 1103 | 0.01 | 12.78212 | 15 | 5 | 5 |
| 2 | uji_4.jpeg | 1472.697 | 1295 | 0.01 | 14.72697 | 16 | 4 | 4 |
| 3 | uji_3.jpeg | 860.021 | 737 | 0.01 | 8.60021 | 12 | 3 | 3 |
| 4 | uji_2.jpeg | 717.612 | 624 | 0.01 | 7.17612 | 15 | 2 | 2 |
| 5 | uji_1.jpeg | 543.747 | 487 | 0.01 | 5.43747 | 9 | 1 | 1 |

Tingkat akurasi dengan penggunaan konstanta 0.01 sebesar, tingkat akurasi = $\frac{5}{5} \times 100\% = 100\%$

Dari 5 data yang telah diuji, menunjukkan penggunaan konstanta 0.01 mempunyai akurasi 100% .

Tabel 2 Penggunaan Algoritma Douglas-Peucker Konstanta 0.02

| No. | Citra uji | Panjang curva | Jumlah point awal | Konstanta | Nilai epsilon | Setelah penyederhanaan | Jumlah jari di deteksi | Jumlah jari pada gambar |
|-----|------------|---------------|-------------------|-----------|---------------|------------------------|------------------------|-------------------------|
| 1 | uji_5.jpeg | 1278.212 | 1103 | 0.02 | 25.56424 | 13 | 5 | 5 |
| 2 | uji_4.jpeg | 1472.697 | 1295 | 0.02 | 29.45394 | 13 | 4 | 4 |
| 3 | uji_3.jpeg | 860.021 | 737 | 0.02 | 17.20042 | 9 | 3 | 3 |
| 4 | uji_2.jpeg | 717.612 | 624 | 0.02 | 14.35224 | 7 | 1 | 2 |
| 5 | uji_1.jpeg | 543.747 | 487 | 0.02 | 10.87494 | 7 | 1 | 1 |

Tingkat akurasi dengan penggunaan konstanta 0.02 sebesar : tingkat akurasi = $\frac{4}{5} \times 100\% = 80\%$

Dari 5 data yang telah diuji, menunjukkan penggunaan konstanta 0.02 mempunyai akurasi 80% .

Tabel 3 Penggunaan algoritma Douglas-Peucker konstanta 0.03

| No. | Citra uji | Panjang curva | Jumlah point awal | Konstanta | Nilai epsilon | Setelah penyederhanaan | Jumlah jari di deteksi | Jumlah jari pada gambar |
|-----|------------|---------------|-------------------|-----------|---------------|------------------------|------------------------|-------------------------|
| 3 | uji_5.jpeg | 1278.212 | 1103 | 0.03 | 38.34636 | 12 | 4 | 5 |
| 6 | uji_4.jpeg | 1472.697 | 1295 | 0.03 | 44.18091 | 9 | 2 | 4 |
| 15 | uji_3.jpeg | 860.021 | 737 | 0.03 | 25.80063 | 8 | 2 | 3 |
| 9 | uji_2.jpeg | 717.612 | 624 | 0.03 | 21.52836 | 7 | 1 | 2 |
| 12 | uji_1.jpeg | 543.747 | 487 | 0.03 | 16.31241 | 6 | 1 | 1 |

Tingkat akurasi dengan penggunaan konstanta 0.03 sebesar: tingkat akurasi = $\frac{1}{5} \times 100\% = 20\%$

Dari 5 data yang telah diuji, menunjukkan penggunaan konstanta 0.03 mempunyai akurasi 20% .

Dari pengujian yang dilakukan penggunaan algoritma Douglas Peucker dapat mengurangi jumlah titik titik dalam pembentukan garis pendukung , dengan menentukan nilai epsilon yang tepat maka jumlah titik yang didapat akan semakin sesuai , ini akan berpengaruh tingkat akurasi pada proses *convex hull* maupun *convexity defect* .Dari pengujian 3 konstanta, nilai konstanta 0.01 memberikan nilai akurasi mencapai 100%

4.2 Pengujian terhadap kombinasi pola

Pengujian akurasi dilakukan dengan pengambilan sebanyak 10 kode dengan jarak antara kamera dengan tangan kurang dari 100 cm , pada ukuran frame 320x240 ,cahaya pada ruangan pada 22-25 lux, dan posisi tangan tegak lurus terhadap kamera , antara kode dengan pola jari tangan disesuaikan apakah sesuai atau tidak.

Tabel 4 Pengujian dengan kombinasi pola

| No. | Uji | Hasil | Ket |
|-----|------|-------|-----|
| 1 | 5555 | 5555 | v |
| 2 | 5554 | 5554 | v |
| 3 | 5432 | 5432 | v |
| 4 | 4321 | 4321 | v |
| 5 | 5124 | 5134 | v |
| 6 | 5134 | 5134 | v |
| 7 | 1345 | 1345 | v |
| 8 | 1234 | 1234 | v |
| 9 | 1334 | 1334 | v |
| 10 | 1244 | 1244 | v |

$$\text{Tingkat akurasi} = \frac{10}{10} \times 100\% = 100\%$$

Dari 10 data yang telah didapat , menunjukkan bahwa alat bekerja pada akurasi 100% .

5 Kesimpulan

- Sistem ini dapat berjalan dengan baik dan dapat diterapkan dengan beberapa kondisi antara lain : posisi tangan tegak lurus menghadap kamera, dan jarak maksimal kurang dari 95 cm.
- Penggunaan algoritma penyederhaan garis Douglas-Peucker dapat menyederhanakan titik , dengan tetap merepresentasikan titik asli , dari penelitian didapatkan nilai epsilon terbaik pada $0.01 \times$ panjang kurva

6 Daftar Referensi

- [1] F.-S. Chen, "Hand Gesture Recognition Using real-time Tracking Method and Hidden Markov Models," Image and Vision Computing, pp. 745-758, 2003.
- [2] K. Demaagd, A. Oliver , N. Oosterndorp and C. Scoot, Practical Computer Vision, O'REILLY, 2012.
- [3] R. Hartanto and M. N. Aji, "Perancangan Awal Antarmuka Gesture Tangan Berbasis Visual," JNTETI, vol. 1, 2012.
- [4] A. S. N, N. and P. Siwindarto, "Sistem Keamanan Pintu Otomatis Menggunakan Voice Recognition," 2014.
- [5] I. Yugashini, S. Vidhyasari and K. G. Devi, "Design and Implementation of Automated Door Accesing System With Face Recognition," International Journal of Science and Modern Engineering, vol. 1, no. 12, 2013.
- [6] R. Hartanto, H. Susanto and P. I. Santoso, "Real Time Hand Gesture Movements Tracking and Recognizing System," Electrical Power , Electronics ,Comunnications , Control , Informatics Seminar, 2014.
- [7] E. P. W. R. Pruehsner and J. D. Enderle, "Soni-Key Voice Controlled Door Lock," 2000.
- [8] H. Hassan, R. A. Bakar and A. T. Fawwas Mokhtar, "Face Recognition Based on Auto-Switching Magnetic Door Lock System Using Microcontroller," International Conference on System Engineering and Technology , 2012.
- [9] M. Schmidt, Raspberry Pi : A Quick Start Guide, The Pragmatics Programmers, 2014.
- [10] Ladyada, PIR Motion Sensor, adafruit learning system, 2014.
- [11] Muntasa, d. P. Arif and H. Mauridhi , Konsep Pengolahan Citra Digital dan Ekstrasi Fitur, Yogyakarta: Graha Ilmu, 2010.
- [12] D. Putra, Pengolahan Citra Digital, Yogyakarta: Andi, 2010.
- [13] D. M. Mount, Computational Geometry, University of Maryland, 2002.
- [14] M. M. Youssef, "Hull Convexity Defect Features for Human Action Recognition," 2011.
- [15] W. S. Tang and M. R. Gonzalez Marquez, "A non-Self Intersection Douglas-Peucker Algorithm".