

DETEKSI POLA TANGAN PADA AREA REGION OF INTEREST (ROI) DENGAN FITUR CONVEXITY DEFECTS

Sugiyatno¹, Untung Subagyo²

^{1,2}Teknik Informatika /STMIK EL RAHMA YOGYAKARTA

e-mail: sugiyatno@stmikelrahma.ac.id, uunboy@gmail.com

Abstract

The need for human interaction with computers today does not only use a keyboard or mouse, but has used other devices such as touch screens. However, the interaction still uses touch between the device and the user. Interaction without using touch or direct interaction is still not widely used. Like the use of a camera as an interaction tool, it is still very rarely used.

The solution offered is to make a system modeling that is able to detect patterns of hand movements with the camera. The detection results can then be used as controls or as a trigger instrument for a process.

Keywords: *Interaction, Camera, Detection, Pattern.*

PENDAHULUAN

Proses citra (Image Processing) merupakan sebuah hal yang penting dalam pemodelan multimedia. Proses pengolahan citra untuk disajikan sebagai sebuah informasi menjadi salah satu hal menarik untuk dikaji. Jika zaman dulu proses citra adalah barang mewah dan sulit dijumpai karena menggunakan alat yang mahal, namun saat ini telah dimiliki oleh nyaris seluruh orang yang hidup di zaman modern ini.

Kebutuhan interaksi manusia dengan komputer saat ini tidak hanya menggunakan keyboard atau mouse, namun sudah menggunakan device lain seperti layar sentuh (touchscreen). Namun demikian, interaksi tersebut masih menggunakan sentuhan antara piranti dengan pengguna. Interaksi tanpa menggunakan sentuhan atau interaksi langsung masih belum banyak dimanfaatkan. Seperti penggunaan kamera sebagai alat interaksi, masih sangat jarang digunakan.

Dalam penelitian ini, mencoba membuat solusi pemanfaatan pengenalan obyek sebagai data pola yang nantinya mampu digunakan sebagai instrumen trigger untuk proses selanjutnya. Solusi yang ditawarkan adalah membuat sebuah pemodelan sistem yang mampu mendeteksi pola gerakan tangan yang selanjutnya disimpan dalam database pola.

METODE PENELITIAN

A. Tahapan Penelitian

Tahapan dalam penelitian ini adalah berikut:

1. Tahap Analisis

Pada tahap ini kegiatan yang dilakukan adalah:

- a) Analisis tentang masalah penelitian dan menentukan pemecahan masalah yang tepat untuk menyelesaikannya. Termasuk didalamnya mendefinisikan batasan sistem dan strategi pengembangan yang digunakan.
- b) Mengumpulkan seluruh informasi yang terkait dan dianalisis. Informasi-informasi tersebut merupakan dasar untuk menetapkan persyaratan bisnis dari sistem yang akan dikembangkan.
- c) Sistem dimodelkan secara logis berdasarkan persyaratan-persyaratan bisnis yang telah ditentukan.

2. Tahap Desain
 - a) Membuat desain berdasarkan model logis yang telah dibuat.
 - b) Membuat analisis pola gerakan tangan yang akan digunakan untuk melakukan query pada sistem.
 - c) Membuat rancangan antar muka sistem
3. Konstruksi Program
Pada tahap ini dibangun aplikasi deteksi gerakan dan database pola.
4. Pengujian

Pada tahap ini dilakukan pengujian pada aplikasi yang sudah dibangun, pengujian dilakukan dengan validitas deteksi gerakan tangan dan dicocokkan dengan pola. Keakuratan deteksi gerakan didapatkan dengan membandingkan pola template yang telah dibuat sebelumnya.

B. Landasan Teori

1. Fitur

Ciri atau fitur adalah segala jenis aspek pembeda atau ciri-ciri yang membedakan. Ciri inilah yang digunakan dalam melakukan identifikasi terhadap objek yang dikenali.[1] Oleh karena itu pemilihan fitur sangat menentukan keberhasilan dalam pengenalan pola. Ukuran fitur diperoleh dari hasil ekstraksi fitur pada objek. Ukuran fitur bisa berwujud simbolik (misalnya warna) atau *numeric* (misalnya tinggi). Fitur yang bagus adalah fitur yang memiliki daya pembeda yang tinggi, sehingga pengenalan/pengelompokan pola berdasarkan fitur dapat dilakukan dengan keakuratan yang tinggi. Contoh fitur objek terlihat pada table 1.

Tabel 1. Contoh fitur objek

Objek	Ciri/Fitur
Huruf	Tinggi, tebal, titik sudut, lengkungan garis, dll.
Suara	Amplitudo, frekuensi, nada, intonasi, dll.
Tanda tangan	Panjang, kerumitan, tekanan, dll.
Sidik jari	Lengkungan, jumlah garis, dll.
Buah	Bentuk, warna, dll.

2. Ekstraksi Fitur

Ekstraksi fitur merupakan proses yang menonjolkan ciri atau karakteristik pada suatu citra yang mengandung informasi penting untuk proses pengklasifikasian ataupun analisa data citra. Tujuan dari proses ekstraksi fitur adalah untuk meningkatkan efektivitas dan efisiensi dari proses analisa ataupun pengklasifikasian. Salah satu teknik computer vision untuk mengekstraksi fitur adalah dengan teknik pendeteksian contour, convex-hull dan convexity defects.

3. Region Of Interest (ROI)

Region of Interest (ROI) adalah daerah yang dipilih sebagai daerah yang paling signifikan di dalam sebuah data yang akan diidentifikasi untuk tujuan tertentu. Tujuan dari penggunaan seleksi ROI adalah memperkecil jumlah fitur citra yang akan diproses, dengan demikian akan meningkatkan kecepatan proses[2]

4. Convexity Defects

Convexity defects merupakan titik atau vector yang diperoleh dari titik contour dan garis convex-hull. Kontur (contour) dapat didefinisikan sebagai suatu keadaan yang terjadi karena adanya perubahan intensitas pada piksel-piksel yang bertetangga. Dengan adanya perubahan intensitas ini, maka titik-titik tepi (edge) pada citra dapat terdeteksi. Kontur juga dapat didefinisikan sebagai urutan titik yang dapat menguraikan bentuk atau region.[3]. Convex-hull merupakan garis luar yang melingkupi seluruh titik-titik kontur (contour).

Convexity defects direpresentasikan dengan 4 (empat) elemen titik atau vector, yaitu:

1. Start_index, yaitu titik kontur yang mendefinisikan permulaan dari defects
2. End_index, yaitu titik kontur yang mendefinisikan akhir dari defects
3. Farthest_index, yaitu titik terjauh dari garis convex-hull ke titik kontur
4. Fix_depth, yaitu perkiraan fixed-point dari jarak antara titik terjauh dari garis convex-hull ke titik kontur.[3]

Gambaran convexity defects dapat dilihat pada Gambar 1



Gambar 1 Convexity Defects

HASIL DAN PEMBAHASAN

Alur penelitian adalah sebagai berikut:

1. Membuat Prototype Device
2. Rancangan Sistem dan kontruksi program
4. Analisis data pengamatan

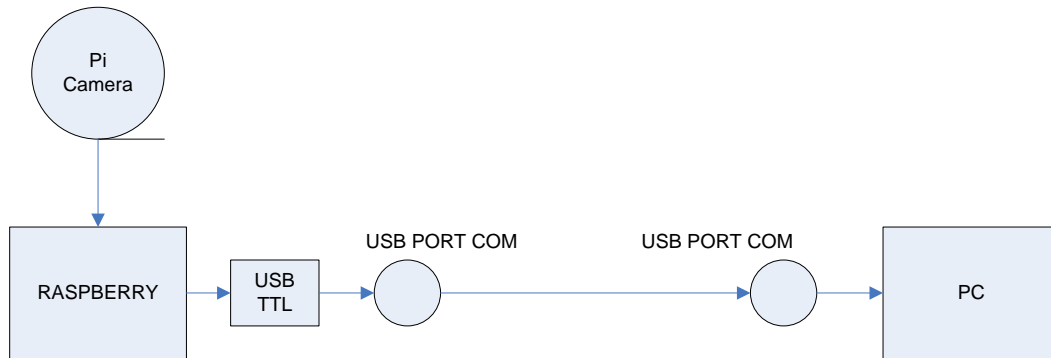
A. Prototype Device

Alat dan bahan

1. Raspberry Pi3

2. SD Memory 16GB kelas 10
3. LCD 3,5" Raspberry
4. Adaptor Micro USB
5. Pi Camera
6. USB to TTL interface
7. Kabel USB male to female 3 meter

Bagan Alat:



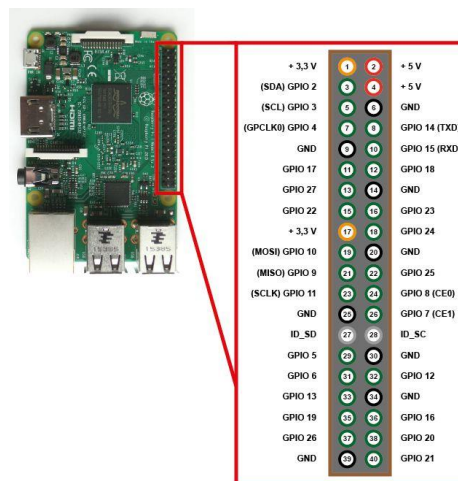
Gambar 2. Bagan Alat

Pi camera dihubungkan ke perangkat raspberry pi3 yang telah memiliki OS Raspbian. Selanjutnya output dari GPIO dihubungkan perangkat USB TTL. USB TTL berfungsi merubah data dari GPIO (Pin dalam raspberry) menjadi data serial yang akan dikomunikasikan ke PC menggunakan interface USB.

GPIO

General-purpose input/output (GPIO) adalah pin generik pada sirkuit terpadu (chip) yang perilakunya (termasuk apakah pin itu input atau output) dapat dikontrol (diprogram) oleh pengguna saat berjalan.[4]

GPIO dalam raspberry dapat dilihat dalam gambar berikut:

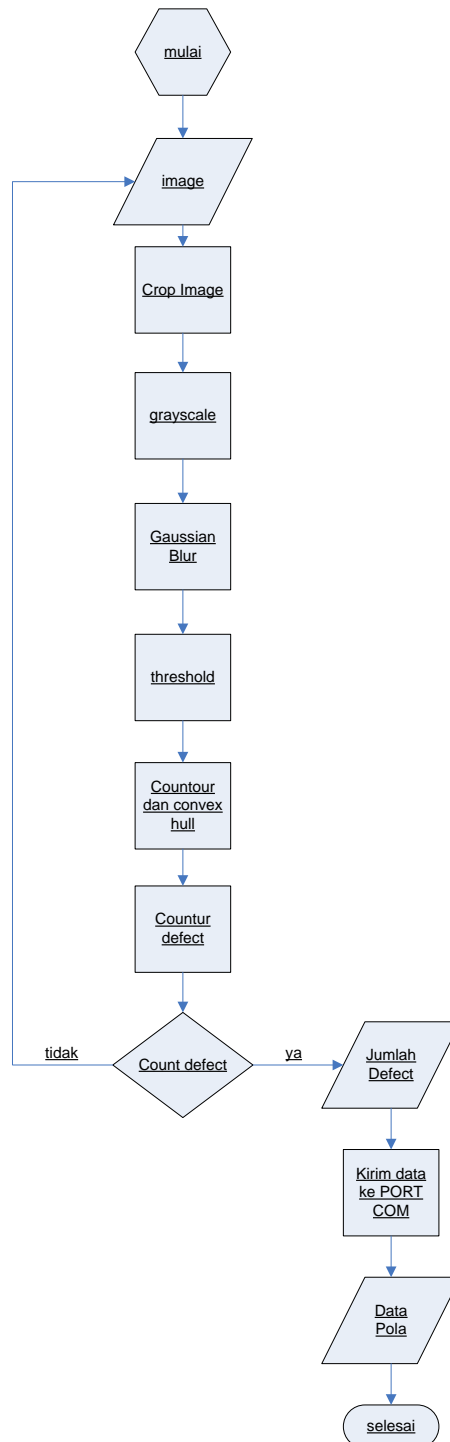


Gambar 3. GPIO Raspberry

B. Rancangan Sistem

Sistem yang dikembangkan dibuat menggunakan bahasa python versi 2.7. Python adalah bahasa pemrograman yang bersifat open source dan dapat digunakan dengan bebas. [5] Python di installkan dalam Raspberry Pi dengan OS Raspbian. Sistem ini juga menggunakan library dari OpenCV versi 3.0.0. Library ini diinstallkan secara mandiri setelah program python diinstallkan.

Adapun Flowchart program yang digunakan untuk membuat deteksi pola adalah seperti gambar 4.



Gambar 4 Flowchart Program

Berdasarkan flowchart diatas dapat dijelaskan masing-masing proses sebagai berikut ini. Proses dimulai dengan menangkap (*capture*) gambar dari camera. Gambar yang ditangkap tidak disimpan dalam sebuah file melainkan di jadikan stream data yang disimpan dalam memory. Berikut cuplikan program untuk mengcapture gambar.

```
cap = cv2.VideoCapture(0)
```

Gambar 5. Video Capture

Gambar yang telah dicapture adalah gambar yang memiliki jenis gambar color atau RGB. Selanjutnya gambar dipotong (*crop*) pada area tertentu saja, hal ini dimaksudkan karena pencarian pola tidak dilakukan diseluruh gambar melainkan pada area yang telah ditentukan. Gambar hasil crop inilah yang akan dijadikan data gambar untuk dicari polanya.

Untuk meningkatkan tingkat akurasi dalam pendeteksian, citra RGB yang telah disimpan dalam memory selanjutnya diberikan filter gaussian blur. Filter ini akan menghilangkan noise-noise pada gambar.

Gambar RGB yang telah dihilangkan noisenya selanjutnya diubah menjadi grayscale. Proses perubahan ini dimaksudkan untuk meningkatkan kemudahan dalam hal pendeteksian, karena gambar akan dirubah menjadi skala abu-abu yang hanya memiliki nilai antara 0 – 255. Cuplikan program untuk merubah ke bentuk grayscale adalah:

```
# convert to grayscale
grey = cv2.cvtColor(crop_img, cv2.COLOR_BGR2GRAY)
```

Gambar 6. Konversi Grayscale

Gambar yang telah diubah menjadi Grayscale selanjutnya dirubah menjadi citra biner. Citra biner adalah citra yang hanya memiliki nilai 0 atau 1. Nilai 0 mengandung warna hitam sedangkan 1 mengandung nilai putih. Proses merubah citra biner ini menggunakan algoritma Threshold. Perintah untuk mengubah citra biner ditunjukkan dalam gambar 9.

```
_, thresh1 = cv2.threshold(blurred, 127, 255,
                          cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
```

Gambar 7. Konversi Grayscale

Gambar yang telah berubah menjadi citra biner, selanjutnya dicari counturnya. Perintah untuk mencari contour adalah seperti pada gambar 10

```
cnt = max(contours, key = lambda x: cv2.contourArea(x))
```

Gambar 8. Pencarian contour

Gambar yang telah ditemukan contournya selanjutnya dicari sisi luarnya dengan proses convex hull. Proses ini ibarat seperti karet gelang yang akan mengelilingi dari sebuah counter obyek. Perintah untuk mencari convex hull adalah seperti gambar 11.

```
hull = cv2.convexHull(cnt, returnPoints=False)
```

Gambar 9. Convex Hull

Gambar yang telah diproses convex hull selanjutnya dicari titik convexity defect, yaitu titik jarak antara counter dengan convex hull. Titik inilah nanti yang akan dicounting sebagai acuan untuk membuat pola gerakan. Perintah untuk melakukan convexity defect adalah seperti gambar 12

```
defects = cv2.convexityDefects(cnt, hull)
count_defects = 0
cv2.drawContours(thresh1, contours, -1, (0, 255, 0), 3)
```

Gambar 10. Convexity Defect

Hasil perhitungan titik defect selanjutnya dimasukkan dalam decision / percabangan pola berdasarkan jumlah titiknya. Jika tidak ditemukan titik defect proses akan dimulai lagi dari capture image, jika ditemukan titik yang jumlahnya sesuai dengan kriteria, hasilnya dikirimkan ke PC melalui USB TTL. Hasil pendeteksian pola seperti terlihat dalam gambar 13.



Gambar 11. Hasil deteksi pola

A. Rancangan Database Aturan Pola

Dalam sistem ini dibentuk beberapa aturan pola berdasarkan jumlah titik yang ditemukan dalam proses pengolahan citra. Aturan tersebut seperti terlihat dalam tabel 2.

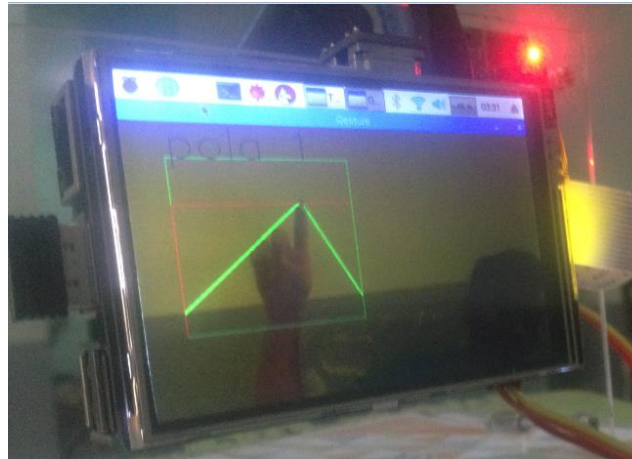
Tabel 2. Aturan Pola

Jumlah Defect	Nama Pola
1	Pola 1
2	Pola 2
3	Pola 3
4	Pola 4

C. Implementasi dan Hasil Pengamatan

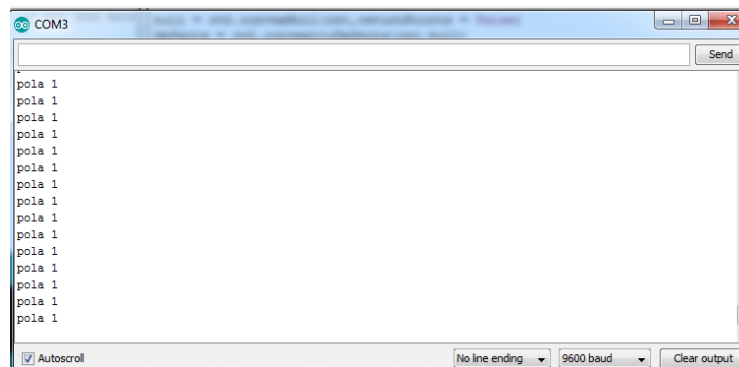
Tahapan ini dilakukan setelah seluruh komponen di kontruksi, baik device maupun programnya. Selanjutnya sistem diuji coba untuk melakukan proses pendeteksian pola gerakan tangan. Proses uji coba dilakukan dengan melakukan perubahan jarak obyek dengan camera dengan intensitas cahaya ruang yang cukup.

Pengujian salah satu posisi camera dalam ukuran jarak tertentu dapat dilihat dalam gambar beriku ini.



Gambar 14. Hasil deteksi Pola

Contoh hasil pengiriman data hasil pedeteksian pola di pc dapat dilihat dalam gambar sebagai berikut ini.



Gambar 15. Hasil pengiriman data Pola 1

Hasil rekap pengujian dapat dilihat dalam tabel 3.

Tabel 3. Rekap Hasil Pengamatan Percobaan

jarak (cm)	Uji	POLA (1=terdeteksi, 0=tidak terdeteksi)				rata-rata (%)
		pola 1	pola 2	pola 3	pola 4	
25	1	1	1	1	1	
	2	1	1	1	1	
	3	1	1	1	1	
	4	1	1	1	1	
	5	1	1	1	1	
	Pros. (%)	100	100	100	100	100
50	1	1	1	1	1	
	2	1	1	1	1	
	3	1	1	1	1	
	4	1	1	1	1	
	5	1	1	1	1	
	Pros. (%)	100	100	100	100	100
75	1	1	1	1	1	
	2	1	1	1	1	
	3	1	1	1	1	
	4	1	1	1	1	
	5	1	1	1	1	
	Pros. (%)	100	100	100	100	100
100	1	1	1	1	1	
	2	1	1	1	1	
	3	1	1	1	0	
	4	1	1	1	1	
	5	1	1	1	0	
	Pros. (%)	100	100	100	60	90
125	1	1	1	1	0	
	2	1	1	1	0	
	3	1	1	1	0	
	4	1	1	1	0	
	5	1	1	0	0	
	Pros. (%)	100	100	80	0	70
Rata-Rata						92

Berdasarkan tabel 3. dapat dilihat bahwa hasil pendeteksian pola dengan jarak 25cm memiliki tingkat akurasi 100%. Semua pola terdeteksi dengan baik. Dalam jarak 50 cm proses deteksi pola juga tidak mengalami penurunan. Semua pola terdeteksi dengan baik. Jarak 75cm proses deteksi tidak mengalami penurunan, tingkat akurasi masih 100%. Jarak 100cm proses deteksi dengan titik defect yang banyak terdapat penurunan tingkat akurasi yaitu 90%. Jarak 125 hampir semua pola yang memiliki defect banyak tidak terdeteksi. Tingkat akurasi menurun 70%. Dari tabel tersebut dapat dihitung rata-rata

tingkat akurasi yaitu 92%.

KESIMPULAN

Dihasilkan sebuah prototipe model sistem untuk melakukan deteksi pola hand gesture yaitu dengan perangkat raspberry dengan camera yang dihubungkan dengan kabel serial. Dengan model ini sistem yang dihasilkan mampu mendeteksi obyek dengan jarak antara 25 – 125cm dengan tingkat akurasi 92%.

Aplikasi yang dihasilkan mampu melakukan proses pengiriman data ke sebuah piranti atau device yaitu PC dengan interface USB yang selanjutnya hasil proses tersebut dapat digunakan sebagai trigger proses komputasi sesuai dengan keinginan pengguna.

SARAN

Penulis menyadari bahwa penelitian ini masih jauh dari sempurna. Untuk itu perlu peneliti lain agar penelitian ini dapat dikembangkan sehingga memiliki manfaat yang lebih luas. Untuk bisa melengkapi penelitian ini, pada peneliti selanjutnya, penulis memberikan saran sebagai berikut:

1. Menggunakan metode segmentasi citra lain yang mampu mendeteksi obyek dengan background yang variasi.
2. Komunikasi pengiriman data dengan piranti dapat dilakukan tanpa kabel

DAFTAR PUSTAKA

- [1] Wakhidah, Nur. 2012. *Deteksi Plat Nomor Kendaraan Bermotor Berdasarkan Area pada Image Segmentation (Licence Vehicles Detection With Area Based on Image Segmentation)*, Jurnal Transformatika, Volume 9, No.2, Januari 2012 : 55 – 63.
- [2] Sajati, Haruno, Dkk, 2017, *Pencocokan Gambar Sidik Jari Dengan Kamera Handphone Menggunakan Metode Ransac Dan Transformasi Affine Berbasis Android*, STTA, Volume 6, Nomor 1, Mei 2017
- [3] Dhawan, Amiraj, Honrao, Vipul, *Implementation of Hand Detection based Techniques for Human Computer Interaction*, International Journal of Computer Applications (0975 – 8887) Volume 72– No.17, June 2013
- [4] Gudanglinux, <http://gudanglinux.com/glossary/gpio-general-purpose-inputoutput/>, 16 Agustus 2013, akses tanggal 10 Oktober 2018
- [5] Tutorials point, 2014, *Python Programming Language Tutorials point simply easy learning*, www.Tutorialspoint.com.