

Tracking Object Using Backpropagation method

Bagus Dwi Andika¹, Setiawardhana², Fernando Ardilla², Rizky Yuniar Hakkun²

Mahasiswa¹, Dosen²

Jurusan Teknik Informatika

Politeknik Elektronika Negeri Surabaya

Institut Teknologi Sepuluh Nopember

Kampus PENS-ITS Keputih Sukolilo Surabaya 60111

Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114

Email: andika.shinbe@gmail.com

Abstrak

Dalam proyek akhir ini dibuat aplikasi penjejakan wajah menggunakan kamera wireless. Kamera wireless *dikoneksikan* dengan komputer. Kamera secara *online mengcapture* obyek. Proses pengenalan wajah menggunakan prinsip – prinsip yang ada di *image prosesing* dan dibandingkan dengan *template* wajah yang ada. Proses selanjutnya adalah mencari posisi wajah dan menentukan titik tengah wajah. kemudian penjejakan wajah menggunakan metode *backpropagation*. Hasil yang didapatkan adalah dapat melakukan penjejakan wajah menggunakan kamera *wireless*.

Kata kunci : *kamera wireless, capture online, image prosesing, pendeteksian, penjejakan, backpropagation.*

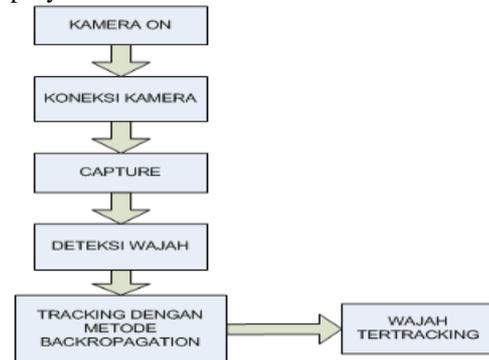
1. Pendahuluan

Sudah banyak peneliti yang menerapkan teori pengolahan citra di berbagai terapan, seperti bidang keamanan, medis, robotika maupun game (permainan). Kamera yang biasanya digunakan adalah kamera yang di kontrol melalu kabel usb. Tapi yang membedakan dari proyek yang lain adalah penulis menggunakan kamera wireless, yaitu Linksys wvc200. Dengan memanfaatkan *wireless* yang ada pada kamera, maka kamera dapat di kontrol secara jarak jauh. Kamera bisa bergerak ke kanan, ke kiri, ke atas dan ke bawah sesuai dengan pergerakan obyek yaitu wajah.

Secara garis besar, proses dalam proyek akhir ini adalah bagaimana caranya mengetahui letak wajah, agar dapat mengerjakan kamera sesuai gerakan wajah. Yang pertama dilakukan adalah mencari 1 obyek wajah, setelah mendapatkan 1 obyek. Kemudian langkah selanjutnya adalah mencari titik tengah obyek. Nilai titik tengah kamera disamakan dengan nilai titik tengah obyek. Nilai titik tengah ini yang akan digunakan untuk memutuskan gerakan kamera untuk selalu mengikuti gerakan wajah.

2. Metodologi

Metodologi yang digunakan dalam menyelesaikan proyek akhir ini



1. Koneksi dengan kamera wireless

Pertama dilakukan koneksi *peer to peer* antara komputer dan kamera *wireless*. IP kamera di *setting* statis dan juga IP komputer juga di *setting* statis. Setelah IP di *setting*, maka langkah selanjutnya adalah mengkoneksikan kamera *wireless* dengan komputer

Script untuk koneksi dengan kamera wireless

```
CameraController cam = new  
CameraController();  
Camera c = new Camera();  
cam.SetCameraIP("192.168.1.171");
```

Script diatas untuk membuat obyek camera. Dan juga setting ipcamera yang akan dikoneksikan.

2. Capture Online Obyek

Untuk meload gambar di program yang dibuat, maka digunakan PlayerPT kontrol, fungsinya untuk play video. Dalam *library* di *axplayerPT.ocx* tidak bisa menyertakan fungsi untuk mengkonversi gambar menjadi bitmap. Maka gambar tidak bisa langsung diproses. Yang perlu dilakukan adalah *mencapture image* secara manual dari form proyek dan merubah ke format bitmap. Dalam form proyek, yang *dicapture* adalah obyek child yaitu *axplayerPT1*. Lalu dirubah menjadi bitmap. Setelah menjadi bitmap maka bisa dilakukan proses terhadap obyek yang terdeteksi.

Secara garis besar proses capture gambar adalah sebagai berikut :

- Digunakan timer agar proses capture bisa online
- Hasil load video di axplayerPT1 di capture menggunakan klas imagecapture
- Karena axplayerPT1 sebagai child, maka dikirim parameter client diisi false.
- Lalu gambar dikembalikan dalam bentuk bitmap
- Setelah bitmap didapatkan dan disimpan di di variable b.
- Maka bitmap sudah bisa diproses.

Listing untuk capture image

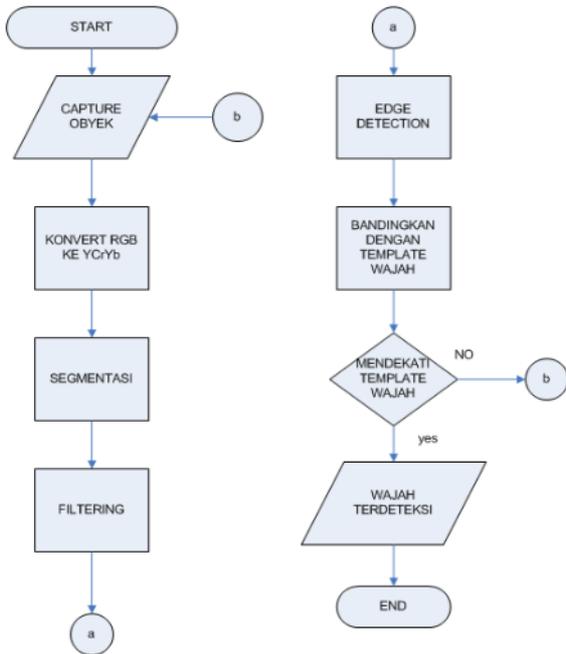
```
imageCapture g = new imageCapture();
Bitmap b =
g.CaptureControl(this.axPlayerPT1,
false, false);
pictureBox1.Image = b;
```

Script ini untuk mencapture image dari axplayerPT1, dan dikonversi menjadi bitmap. Lalu ditampilkan di pictureBox1.

3. Deteksi Wajah

Untuk proses pendeteksian wajah menggunakan library dari AForge.NET. library aforge.imaging digunakan dalam pendeteksian wajah.

Flowchart pendeteksian wajah:



Dalam library Aforge.imaging terdapat fungsi – fungsi untuk segmentasi warna, filtering, dan edge

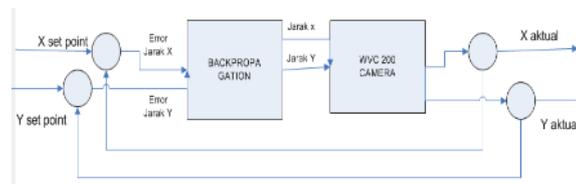
detection. Fungsi segmetasi warna digunakan untuk memisahkan antara warna kulit dengan warna selain warna kulit, untuk deteksi tepi digunakan algoritma Roberts Cross Edge detection. Setelah warna menjadi grayscale, maka dibandingkan dengan template wajah yang telah ada.

4. Tracking wajah

Tahap berikutnya adalah menelusuri atau tracking posisi dari objek. Pertama dilakukan adalah mencari titik tengah obyek (dx, dy). Penentuan titik tengah wajah mencari diagonal wajah, titik tengah diagonal tersebut merupakan titik tengah wajah. Titik pusat wajah yang telah didapatkan dicari errornya yang paling kecil dengan menggunakan metode backpropagation. Agar mendapatkan titik tengah wajah yang presisi.

Untuk training data, layar yang berukuran 320x240 dibagi menjadi beberapa bagian atau disebut proses mapping. Tiap bagian berukuran 20x20. Dicari titik tengahnya, terdapat 768 pasangan data titik tengah, tiap titik tengah tersebut dibandingkan jaraknya dengan titik tengah kamera, maka terdapat 768 pasangan data jarak antara titik tengah kamera dengan titik tengah obyek. Selanjutnya disebut error jarakX dan error jarakY. Pasangan data ini yang akan digunakan sebagai data training dalam system ini. Inputan berupa error jarakX dan error jarakY sedangkan outputnya jarakX dan jarakY.

Pengaplikasian metode backpropagation digunakan untuk mendapatkan jarak yang tepat setelah dikoreksi errornya dengan backpropagation. Berikut gambar pengaplikasian metode backpropagation.



Gambar aplikasi backpropagation

Dari gambar di atas X set point dan Y set point adalah titik pusat kamera. Lalu dihitung jarak antara titik tengah wajah dengan titik tengah kamera. Kemudian jarak ini disebut error jarakX dan error jarakY. Inputan di backpropagation adalah errorX dan errorY. Kemudian dilakukan metode backpropagation, unit keluarannya adalah jarakX dan jarakY. Kemudian jarakX dan jarakY inilah yang dikirimkan ke dalam kamera agar bergerak sejauh jarakX dan jarakY. Setelah kamera bergerak sejauh jarakX dan jarakY, maka titik tengah wajah akan berada pada titik tengah kamera.

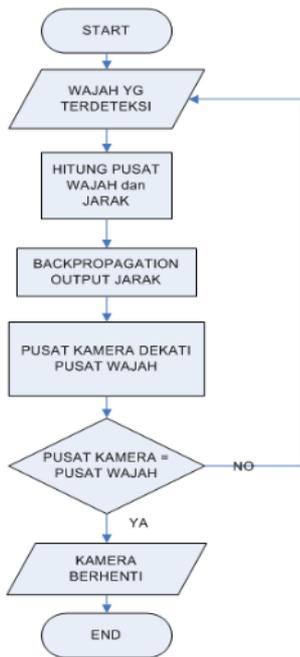
Proses *tracking* wajah dilakukan dengan cara membandingkan titik tengah wajah dengan titik tengah kamera. Setelah diketahui jarak antara titik tengah wajah dengan titik kamera, lalu sistem memberi perintah terhadap kamera agar titik pusat kamera menuju ke titik pusat wajah. Maka kamera akan mengikuti gerakan wajah.

Script untuk mengejar titik tengah obyek

```
cam.MoveTo(tujuanx, tujuany);
```

Script diatas untuk memerintahkan kamera untuk menuju ke titik tengah wajah. Parameter yang dikirim adalah tujuanx, berarti jarak x dari titik pusat kamera dengan titik pusat wajah. Begitu juga parameter tujuany, tujuany merupakan jarak antara titik pusat wajah dengan titik pusat kamera.

Flowchart untuk tracking wajah



Flowchart diatas menggambarkan langkah untuk penjejakan wajah. Setelah wajah terdeteksi. Dicari titik tengah wajah, lalu titik tengah kamera mendekati titik tengah wajah. Kamera berhenti jika titik tengah wajah sudah sama dengan titik tengah kamera.

3. Result

Berikut hasil output *face tracking*



Form *facetracking* merupakan form utama dalam proyek ini. Dalam form ini memiliki fungsi untuk *play video* dari kamera wvc200 dan fungsi untuk *tracking* wajah. Dalam form ini terdapat *picture box* yang menunjukkan wajah yang terdeteksi. Form ini juga menampilkan informasi lokasi titik pusat wajah. Dari titik pusat wajah ini akan didekati oleh titik pusat kamera, maka posisi wajah akan selalu di tengah.

Berikut ini keterangan dari GUI :

- *Button PLAY* : *button* untuk *play video*.
- *Button Capture* : *button* untuk mengaktifkan *facetracking*.
- *Button STOP* : *button* untuk menghentikan video dan mengarahkan kamera ke *center* kamera atau *reset* kamera.
- *axplayerPT* : untuk memload gambar yang di *capture* oleh kamera
- *pictureBox1* : untuk menampilkan hasil proses *facetedetection* dan *facetracking*.
- *PictureBox2* : untuk menampilkan wajah yang terdeteksi.
- *Textbox1* dan *textbox2*: untuk menampilkan posisi pusat kamera
- *Textbox3* dan *textbox4* : untuk menampilkan posisi pusat wajah.

Hasil uji coba face detect dan face tracking

Intensitas cahaya sangat berpengaruh dalam proses deteksi wajah, pada uji coba ini telah dilakukan dengan berbagai macam tingkat intensitas cahaya, dan *system* deteksi wajah member hasil yang cukup baik, hal itu terbukti dari hasil uji coba ini yang sebagian besar bisa mendeteksi wajah meskipun dengan tingkat intensitas cahaya yang rendah.

Tingkat intensitas cahaya juga berpengaruh pada gerakan kamera dalam melakukan penjejakan wajah. Hal ini dikarekan sebelum melakukan gerakan kamera, hal yang dilakukan adalah deteksi wajah. Jadi semakin cepat deteksi wajah, maka proses penjejakan wajah juga semakin cepat, begitu pula sebaliknya.

Dibawah ini adalah tabel hasil pengujian terhadap deteksi wajah dalam bermacam macam intensitas cahaya.

hasil pengujian dengan intensitas cahaya

NO	Kondisi	Rata- rata waktu
1	A	1931.25
2	B	1584.375
3	C	1867.188
4	A+B	1540.625
5	B+C	1862.5
6	A+B+C	1654.688
7	PADAM	Tidak bisa
8	Luar ruangan	1201.563

Dari table pengujian di atas, kecepatan pendeteksian wajah dipengaruhi intensitas cahaya. Namun pada cahaya yang terlalu banyak juga mengganggu proses deteksi wajah. Contohnya pada kondisi A+B+C semua lampu dinyalakan, namun kecepatan deteksi juga berkurang. Hal tersebut terjadi karena intensitas cahaya terlalu tinggi. Bisa dibandingkan dengan kondisi A+B, dengan intensitas cahaya yang pas, di depan dan disamping. Hal tersebut terbukti dari waktu pendeteksian lebih cepat dibanding dengan semua lampu menyala. Saat lampu semua padam, proses pendeteksian tidak bisa dilakukan. Pendeteksian paling cepat dilakukan diluar ruangan pada siang hari. Di karenakan saat di luar ruangan, intensitas cahaya cukup baik sehingga proses pendeteksian berjalan dengan baik.

Uji coba berikutnya adalah uji coba *system* secara keseluruhan. Uji coba dilakukan di dalam ruangan dan luar ruangan. Posisi titik tengah kamera yaitu di $x = 160, y = 120$; Table hasil pengujian face tracking di dalam dan luar ruangan

No	lokasi	hasil	Titik
1	Luar ruangan	Berhasil	$x= 70,y= 110$; $x= 110,y= 110$; $x= 150,y= 110$;
2	Dalam ruangan	berhasil	$x= 270,y= 130$; $x= 230,y= 130$; $x= 170,y= 130$;

Dari percobaan *tracking* wajah, proses tracking wajah bisa dilakukan di dalam ruangan maupun di luar ruangan. Titik pusat kamera jarang bisa mencapai tepat di titik pusat obyek, karena titik pusat obyek selalu diupdate dan selalu berubah meskipun benda tidak berubah.

4. Penutup

Kesimpulan

Dari hasil uji coba sistem ini dapat ditarik beberapa kesimpulan:

- ❖ Kamera yang digunakan dalam system ini adalah kamera wireless sehingga hasil capture gambar, deteksi wajah dan tracking wajah juga lambat.
- ❖ Sistem ini bisa mendeteksi wajah dengan beberapa posisi wajah, seperti wajah dengan posisi tegak, posisi miring, posisi wajah menghadap ke atas.
- ❖ Proses deteksi wajah juga mengalami error saat mendeteksi warna yang mendekati warna kulit. Namun menggunakan library yang ada template wajah, maka error itu bisa dikurangi.
- ❖ Pada saat pencahayaan kurang maka hasilnya juga kurang baik saat mendeteksi wajah.

Saran

- ❖ Sebaiknya menggunakan algoritma yang lebih bagus agar bisa melakukan tracking lebih cepat.
- ❖ Untuk peng-aplikasi-an dari sistem ini, kamera wireless diletakan di tempat dengan cahaya yang cukup, agar bisa mendeteksi obyek dengan baik.
- ❖ Untuk pengembangan lebih lanjut, sistem ini dapat dikombinasikan dengan beberapa metode lain seperti *motion tracking*, serta metode-metode lain.

Daftar Pustaka

- [1] Basuki ,Achmad & Nana Ramadijanti. *Pengantar Grafika Komputer*. Laboratorium Computer Vision – (PENS-ITS).
- [2] Agus Rachmat Aris Efendi. *Trajectory Following Stering Mobile Robot Dengan Sistem Jaringan Saraf Tiruan*. 2007.