

Rancang Bangun Robot PANTILT: Pendeteksian Posisi Kepala Manusia Menggunakan Algoritma Genetika

Ferman Hakiki¹, Indra Adji Sulistijono²

¹Jurusan Teknik Elektornika, Politeknik Elektronika Negeri Surabaya

Kampus PENS-ITS SUKOLILO, SURABAYA

www.hak@student.eepis-its.edu

www.indra@eepis-its.edu

Abstrak--- Persoalan penentuan posisi kepala manusia merupakan persoalan optimasi untuk mencari koordinat dari kepala manusia. Metode penentuan posisi kepala manusia ini banyak sekali macamnya. Paper ini menjelaskan bagaimana menggunakan metode algoritma genetika untuk mengetahui posisi koordinat dari kepala manusia. Algoritma genetika adalah algoritma pencarian dan optimasi yang terinspirasi oleh prinsip genetika dan seleksi alam (teori evolusi Darwin). Sebelum proses ini dilakukan, terlebih dahulu dilakukan proses deteksi warna kulit, sehingga antara kulit dan bukan kulit akan terlihat. Agar dapat mendeteksi bagian kepala manusia, maka digunakan template kepala manusia pada umumnya. Proses penentuan posisi kepala manusia ini adalah salah satu bagian dari sistem PanTilt robot dimana PanTilt robot adalah robot yang dapat mendeteksi kepala manusia dan mengikuti kemana kepala bergerak.

I. PENDAHULUAN

Saat ini banyak sekali metode penentuan posisi kepala manusia. Metode-metode tersebut mempunyai kelemahan dan kelebihan. Pada paper ini akan dijelaskan bagaimana menentukan posisi kepala manusia menggunakan algoritma genetika.

Algoritma genetika adalah algoritma pencarian dan optimasi yang terinspirasi dari teori evolusi Darwin. Pada teori tersebut, hanya individu-individu terbaik yang dapat bertahan.

Penentuan posisi kepala manusia ini adalah salah satu bagian dari sistem robot PanTilt. Robot PANTILT adalah sebuah robot yang dapat mendeteksi posisi kepala manusia dan mengikutinya. Robot ini sebenarnya meniru gaya komunikasi manusia yang selalu melihat wajah ketika berkomunikasi dengan manusia lainnya. Disebut PANTILT karena robot ini terdiri dari 2 gerakan yaitu menggeleng dan mengangguk.

Sebelum robot ini dapat mengikuti pergerakan wajah seseorang, terdapat beberapa langkah yang harus diperlukan yaitu robot harus mendeteksi adanya kepala manusia (untuk deteksi ini dilakukan dengan metode algoritma genetika).

Setelah diketahui adanya kepala manusia, maka komputer akan mengirim data koordinat x dan y kepada servo motor sehingga kamera dapat mengikuti pergerakan kepala manusia yang ada didepannya.

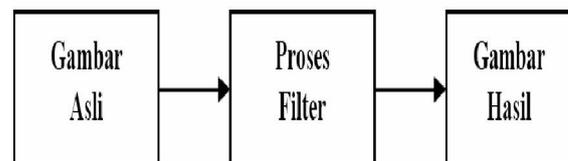
Robot ini adalah salah satu bentuk aplikasi dari *computer vision*, dimana robot ini dapat mendeteksi wajah manusia dan mengikuti pergerakan kepala manusia tersebut.

Paper ini difokuskan kepada penentuan posisi kepala manusia menggunakan algoritma genetika. Bagian yang kedua adalah *image processing* yang digunakan untuk mendeteksi warna kulit.

II. IMAGE PROCESSING

Image processing atau sering disebut dengan pengolahan citra digital merupakan suatu metode yang digunakan untuk mengolah atau memproses dari gambar asli sehingga menghasilkan gambar lain yang sesuai dengan kebutuhan. Pengambilan gambar bisa dilakukan oleh kamera video atau alat-alat yang lain yang dapat digunakan untuk mentransfer gambar. Dalam pengolahan citra, dilakukan operasi terhadap citra asli menjadi citra baru berdasarkan citra asli.

Misalnya ada suatu gambar yang terlalu gelap maka dengan *image processing* gambar tersebut bisa diproses sehingga mendapatkan gambar yang jelas. Secara garis besar bisa gambarkan seperti blok diagram pada gambar 2.1.1 dibawah ini:



Gambar 2.1 Blok Diagram Pengolahan Citra

Warna yang diterima dari kamera adalah warna RGB. Untuk memodelkan warna kulit sistem warna RGB ini ditransformasikan ke sistem warna YCrCb untuk memisahkan intensitas Y dengan chromaticity yang dinyatakan dalam dua

variabel Cr dan Cb. Harga Cr membedakan warna antara jarak intensitas terhadap unsur warna merah sedangkan Cb menyatakan jarak intensitas terhadap unsur warna merah. Dalam memodelkan warna kulit hanya informasi Cr dan Cb yang dipakai, sehingga pengaruh perubahan intensitas dapat dihilangkan. Pada daerah saturasi dari cahaya yang tertangkap kamera, harga Cr dan Cb sangat stabil, sehingga nilai Cr dan Cb merupakan informasi handal untuk proses klasifikasi warna.

Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing r, g dan b menjadi citra warna YCrCb dapat dilakukan dengan menghitung seperti pada persamaan (2.1), (2.2) dan (2.3).

$$Y = 0.59G + 0.31R + 0.11B \quad (2.1)$$

$$Cr = 0.713 * (R-Y) \quad (2.2)$$

$$Cb = 0.564 * (B-Y) \quad (3.3)$$

Untuk mencoba proses konversi citra berwarna RGB menjadi citra warna YCrCb dapat dibuat program seperti gambar 2.2.



Gambar 2.2 Konversi citra ke YcrCb

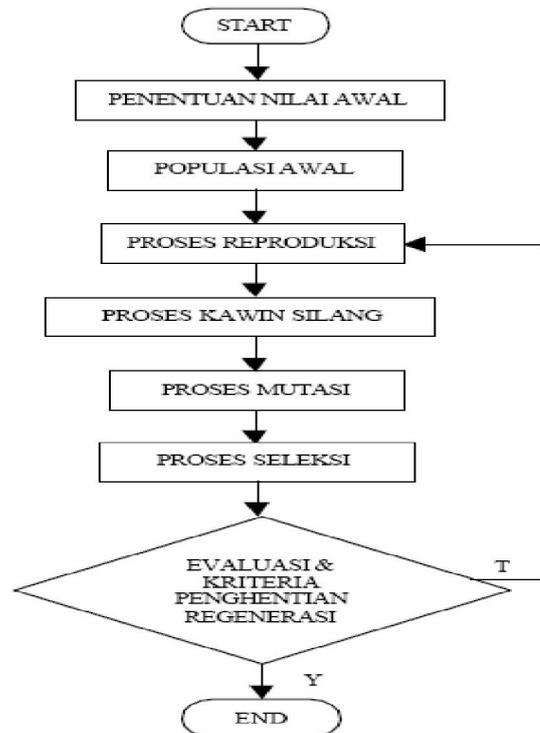
III. ALGORITMA GENETIKA

Algoritma genetika pertama kali dikemukakan oleh John Holland awal tahun 1975. Algoritma ini banyak dipakai dalam aplikasi bisnis, teknik maupun aplikasi lainnya. Algoritma genetika ini dimulai dengan kumpulan-kumpulan solusi yang dibangkitkan secara acak. Yang kemudian akan diseleksi dengan nilai fitness. Sebenarnya prinsip algoritma genetika ini terinspirasi dari teori evolusi Charles Darwin yaitu setiap makhluk hidup akan menurunkan satu atau beberapa karakter ke anak atau keturunannya (Bambrik,1997). Dalam proses tersebut bisa terdapat variasi karena didalam algoritma genetika terdapat proses mutasi sehingga keturunannya memiliki kelebihan atau kekurangan dari induknya.

Setiap makhluk hidup akan mengalami seleksi alam, sehingga makhluk hidup yang dapat beradaptasi dengan baik yang mampu bertahan sampai generasi selanjutnya. Semakin bagus atau sesuai fitness dari sebuah solusi, maka semakin besar kemungkinan individu untuk dipilih. Proses ini akan terjadi secara berulang-ulang sampai kondisi tertentu terpenuhi.

Deskripsi algoritma genetika adalah:

1. [start] *Generate* populasi pertama secara random sebanyak n individu.
2. [Fitness] Evaluasi nilai fitness dari masing masing individu.
3. [New Populasi] Bentuk populasi baru dengan melakukan pengulangan langkah-langkah seperti berikut:
 - a. Seleksi
Pilih 2 individu yang akan dijadikan sebagai induk dari sebuah populasi sesuai dengan fitness mereka (semakin besar nilai fitness suatu individu, maka semakin besar kemungkinan mereka untuk dipilih).
 - b. Crossover
Lakukan persilangan diantara kedua induk sesuai probabilitas crossover untuk membentuk keturunan baru.
 - c. Mutation
Mutasi keturunan yang baru sesuai dengan probabilitas mutasi dari setiap gen.
 - d. Accepting
Tempatkan keturunan yang baru sebagai populasi yang baru.
4. [Replace] Gunakan populasi yang baru dibentuk untuk menjalankan algoritma genetika.
5. [Test] jika kondisi akhir terpenuhi, maka tampilkan solusi dari populasi.
6. [Loop] Kembali ke no 2.

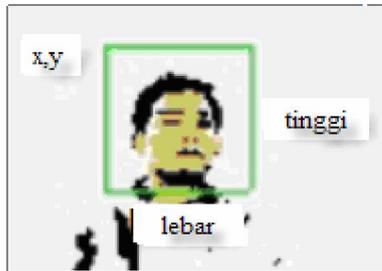


Gambar 3.1 Flowchart Algoritma Genetika

Berikut adalah urutan-urutan algoritma genetika dalam penentuan posisi kepala manusia menggunakan algoritma genetika.

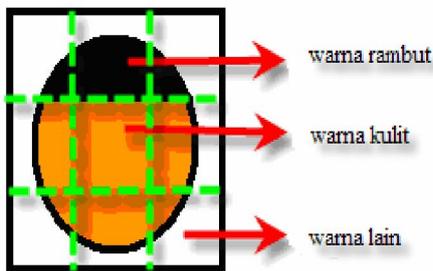
Ø Inisialisasi

Pada proses ini dilakukan pembangkitan solusi secara acak. Individu-individu yang dibangkitkan terdiri dari posisi (x,y), lebar dan tinggi.



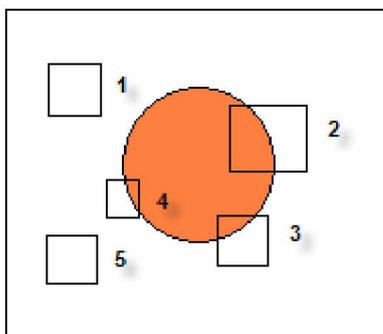
Gambar 3.2 Gen gen yang dibangkitkan pada tiap tiap individu

Untuk memodelkan kepala manusia, maka perlu dibuat *template* dari kepala manusia seperti berikut:



Gambar 3.3 Template kepala manusia

Ilustrasi dari proses pembangkitan populasi secara acak adalah sebagai berikut:



Gambar 3.4 Proses pembangkitan populasi

Gambar di atas menunjukkan 5 buah individu yang dibangkitkan secara acak. Kotak diatas menunjukkan individu

Ø Seleksi

Proses ini bertujuan untuk mencari individu yang nantinya akan menjadi induk untuk menghasilkan keturunan-keturunan yang baru.

Proses seleksi digunakan metode *roulette wheel*. Seleksi dilakukan dengan menggunakan prosentasi fitness setiap individu, dimana setiap individu mendapatkan luas bagian sesuai dengan prosentase nilai fitnessnya. Nilai fitness dapat dicari dengan menggunakan rumus berikut:

Untuk menghitung nilai fitness, digunakan persamaan sebagai berikut:

$$f_i^{He} = C_{kulit}^{Kp} + C_{rambut}^{Kp} + \eta_1^{He} \cdot C_{kulit}^{Kp} \cdot C_{rambut}^{Kp} - \eta_2^{Kp} \cdot C_{lainnya}^{Kp} \quad (2.4)$$

Dimana:

f_i^{Kp} = nilai fitness

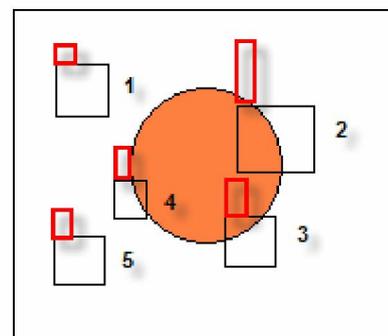
C_{kulit}^{Kp} = jumlah piksel warna kulit

C_{rambut}^{Kp} = jumlah piksel warna rambut

$C_{lainnya}^{Kp}$ = jumlah piksel warna lainnya

η_1^{Kp} dan η_2^{Kp} = koefisien warna rambut, kulit dan warna lain

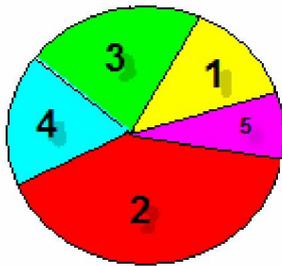
Selain dengan menggunakan seleksi *roulette wheel*, dapat juga langsung dicari individu dengan nilai maksimum yang nantinya akan digunakan pada proses selanjutnya.



Gambar 3.5 Proses seleksi

Individu no. 2 mempunyai nilai fitness lebih banyak. Dapat dilihat dari indikator kotak merah. Individu tersebut mempunyai kesempatan yang lebih besar untuk menjadi induk dalam proses crossover. Setelah diketahui nilai fitness dari masing masing individu, maka proses selanjutnya adalah dicari

probabilitas dari masing masing individu tersebut untuk menjadi induk. Berikut adalah ilustrasi probabilitas individu individu untuk menjadi induk.



Gambar 3.6 Seleksi dengan roulette wheel

Ø Crossover

Setelah didapat nilai fitness yang mendekati referensi, maka dilakukan proses *crossover* antar induk untuk mendapatkan keturunan baru. Jenis crossover untuk *steady state* adalah *elitist crossover* yang menggabungkan gen-gen individu terbaik dengan individu terburuk. Sedangkan untuk *simple genetic algorithm* adalah *simple crossover* yang induk-induknya didapatkan secara acak.

0 0 1 1 1 0 0 1 β induk 1
 1 0 0 1 1 0 1 0 β induk 2
 0 0 1 1 1 0 1 1 β induk 1
 1 0 0 1 1 0 0 0 β induk 2

Ø Mutasi

Setelah proses crossover selesai, maka proses selanjutnya adalah mutasi. Disini kami menggunakan adaptive mutation untuk *steady state genetic algorithm*. Rumusnya adalah sebagai berikut:

$$g_{i,j} \leftarrow g_{i,j} + \left(\alpha_j \cdot \frac{fit_{max} - fit_i}{fit_{max} - fit_{min}} + \beta_j \right) N(0,1) \quad (3.5)$$

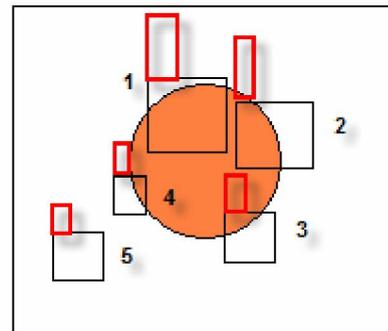
Dimana fit_i adalah nilai fitness dari individu ke i , fit_{max} dan fit_{min} adalah nilai fitness maksimal dan minimal dalam suatu populasi; $N(0,1)$ adalah nilai acak dari nol sampai 1. α_j dan β_j adalah koefisien dan offset.

Untuk simple genetic algorithm digunakan *simple mutation* yang menukar salah satu gen-gen didalam individu tersebut.

0 0 1 1 1 0 1 1 β induk (sebelum mutasi)
 0 0 1 1 1 1 0 1 β induk (setelah mutasi)

Setelah didapat generasi baru dari hasil kawin silang dan mutasi, maka proses akan kembali lagi ke

seleksi. Dimana akan dipilih individu dengan nilai fitness yang lebih baik. Hal ini akan terus menerus berlangsung sampai iterasi ke n atau nilai fitness yang dihasilkan tidak berubah.



Gambar 3.7 Terbentuk populasi baru

Dari hasil *crossover* dan mutasi diatas, maka individu dengan nilai terburuk akan diganti dengan individu baru yang mempunyai nilai fitness yang lebih baik. Individu no 1 merupakan individu dengan nilai terburuk. Setelah mengalami *crossover* dan mutasi, maka individu no 1 mempunyai nilai fitness yang lebih baik.

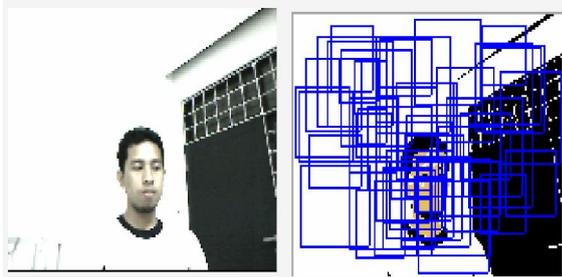
IV. HASIL PERCOBAAN dan ANALISIS

Dari hasil percobaan yang kami lakukan, untuk proses pendeteksian posisi kepala manusia dengan algoritma genetika ini, kami membangkitkan 50 individu secara acak. Proses pertama kali yaitu kamera mengambil gambar. Setelah itu, dari informasi warna RGB yang diperoleh, akan ditransformasi menjadi YCrCb. Harga Cr membedakan warna antara jarak intensitas terhadap unsur warna merah sedangkan Cb menyatakan jarak intensitas terhadap unsur warna merah. Berikut adalah hasil transformasi warna dari RGB ke YCrCb:



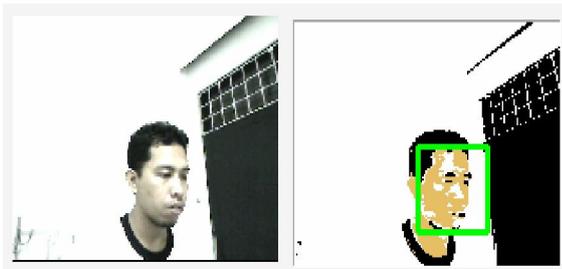
Gambar 4.1 Proses Deteksi Kulit

Banyaknya cahaya yang ditangkap oleh kamera akan sangat berpengaruh dalam proses transformasi warna ini. Semakin banyak cahaya yang ditangkap, maka hasil yang diperoleh akan lebih jelek. Jadi harus diatur terlebih dulu pencahayaannya. Setelah proses pendeteksian warna kulit selesai, maka proses penentuan posisi kepala manusia dimulai. Mula-mula dibangkitkan n individu secara acak.



Gambar 4.2 Pembangkitan individu secara acak

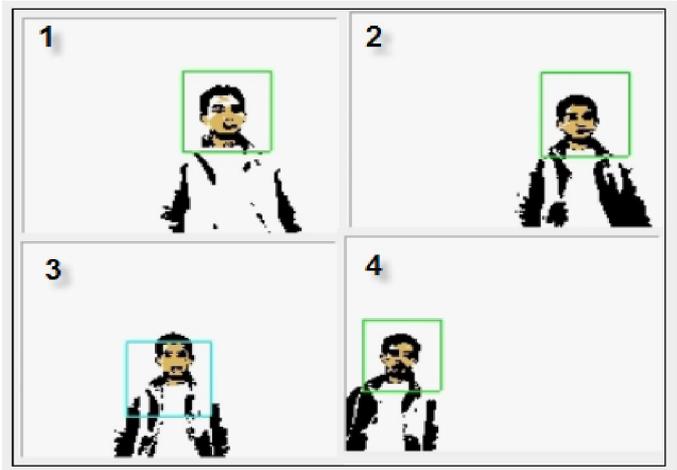
Pada gambar diatas, tampak banyak kotak-kotak yang posisinya acak. Setelah dibangkitkan inidividu secara acak, maka individu akan diseleksi untuk dicari individu terbaik. Proses penyeleksian ini dicari dengan menghitung nilai fitness dari masing masing individu kemudian dicari nilai fitness terbesar.



Gambar 4.3 Individu dengan nilai fitness maksimum

Kotak dengan warna hijau diatas adalah individu yang mempunyai nilai fitness terbanyak. Setelah itu, dilakukan proses crossover dan mutasi sampai maksimal iterasi tercapai.

Berikut adalah hasil pendeteksian posisi kepala manusia menggunakan steady state genetic algorithm (SSGA):



Gambar 4.4 Hasil pengambilan data

Berikut adalah kecepatan dari individu untuk mencapai Posisi kepala manusia:

n o	Ind ke	X	Y	Lebar	Tinggi	Nilai Fitness	V (ms)
1	67	81	29	45	45	97.77	55
2	30	97	33	45	47	134.48	21
3	30	53	55	43	42	104.20	55
4	65	8	46	40	40	60.66	53

Masalah muncul ketika kepala manusia tersebut bergerak. Program akan menghitung lagi nilai fitness dari individu awal sampai akhir sehingga proses akan lebih lambat dan individu tersebut kemungkinan tidak bisa mencapai atau mendekati kepala manusia. Jadi pergerakan kepala manusia harus diatur dulu kecepatannya. Sehingga individu tersebut dapat mencapai koordinat dari kepala dengan optimal.

V. KESIMPULAN

Persoalan penentuan posisi kepala manusia dapat diselesaikan dengan algoritma genetika. Kelebihan algoritma genetika adalah pada setiap generasi kita bisa memperoleh solusi. Algoritma genetika sangat cocok untuk pemecahan masalah optimasi.

Penentuan posisi kepala manusia menggunakan algoritma genetika ini digunakan untuk robot PanTilt agar robot tersebut dapat mendeteksi kepala manusia dan mengikuti pergerakannya.

SSGA ini mengganti beberapa individu dalam beberapa generasi. Individu terburuk akan diganti dengan individu baru hasil dari *crossover* dan mutasi. Untuk proses crossover ini digunakan elitist *crossover* dimana secara acak mendapatkan satu induk dan nilai gennya dikombinasikan dengan individu terbaik. Untuk menghemat waktu komputasi, maka induk yang dihasilkan secara acak diganti dengan individu terburuk.

Dalam menentukan jumlah individu, hendaknya jangan terlalu banyak. Hal ini akan mengakibatkan proses jalannya program menjadi lambat sehingga hasilnya kurang optimal.

DAFTAR PUSTAKA

- [1] **Sams, A.T.**, “*Teach Yourself Visual C++ in 21 Days*”.
- [2] **Parker J.R.**, “*Algorithm For Image Processing And Computer Vision*”, Wiley Computer Publishing 1997.
- [3] **Gen Mitsuo, Runwei Cheng.**, “*Genetic Algorithms and Engineering optimization*”, Wiley Computer Publishing 2000.
- [4] **Sigit Riyanto.**, “*Praktikum Pengolahan Citra Digital (Image Processing)*”, PENS ITS 2002.
- [5] **Adji Indra S.**, “*Evolutionary Robot Vision for Multiple Human Heads Tracking of A Partner Robot*”, Korea, 26-29 Agustus 2007.
- [6] **Aulia Fitriah, Achmad Zaky, Fitrasani.**, “*penerapan Algoritma Genetika Pada Persoalan Pedagang Keliling*”, ITB.