

Pengontrolan Gerak Menggunakan Marker Untuk Aplikasi Augmented Reality

Liliana, Andreas Yohan, Kartika Gunadi

Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Email: lilian@petra.ac.id, kgunadi@petra.ac.id

abstrak

Perkembangan teknologi *Virtual Reality* belakangan ini diimplementasikan dalam berbagai bidang, terutama game dan media komunikasi. Salah satu variasi dari *virtual reality* adalah *augmented reality*. Augmented Reality masih memadukan obyek kasat mata dengan obyek virtual hasil ciptaan dengan menggunakan komputer. Dalam aplikasi berupa game komputer, pemanfaatan teknologi augmented reality adalah menggantikan mouse dengan menggunakan marker sebagai pengontrol gerak. Penggunaan marker dimaksudkan untuk mendapatkan interaksi yang lebih alami dari pengguna kepada sistem. Pada penelitian ini, dikembangkan sebuah sistem untuk mengenali adanya sebuah marker pengganti mouse dan menangkap gerakan yang dilakukan pada marker. Hasil gerakan yang berhasil dikenali akan menjadi masukan untuk mengatur gerak dari karakter yang dimunculkan di layar komputer. Fokus dari penelitian ini adalah mengenali adanya marker dan mendeteksi terjadinya gerakan pada posisi marker dibandingkan dengan posisi awal saat marker dikenali. Sensor yang digunakan dalam sistem ini adalah webcam dan dilakukan secara real time. Saat pertama kali aplikasi dijalankan, maka sistem akan mencari koneksi dari webcam dan langsung mengambil frame pertama yang terbaca dari webcam tersebut untuk dilakukan pendeteksian marker. Dari hasil pendeteksian marker tersebut, akan dilakukan estimasi rotasi yang terjadi pada *marker*. Kemudian akan ditambahkan (*overlay*) suatu objek 3D ke tepat diatas gambar *marker* tadi. Berdasarkan hasil pengujian yang dilakukan, pendeteksian *marker* dan estimasi rotasi relatif akurasi dengan perbedaan presisi +/- 10 derajat jika dibandingkan dengan sudut gerakan aslinya.

Kata Kunci : *Augmented Reality, Marker Detection, virtual reality*, deteksi gerak

Latar Belakang

Teknologi Virtual Reality (VR) dikembangkan sebagai media interaktif yang memungkinkan penggunaan indera manusia sebagai alat masukan bagi sistem (Sherman, 2003). Teknologi VR menciptakan sebuah lingkungan yang tidak berwujud secara fisik yang diciptakan dengan menggunakan komputer, namun bisa diindera, seperti dapat dilihat lewat Head Mounted Display ataupun dapat dipegang dengan menggunakan glove khusus. Untuk menciptakan sebuah sistem VR secara menyeluruh membutuhkan banyak biaya. Oleh karena itu muncul variasi dari VR, yaitu Augmented Reality (AR) yang menggabungkan antara obyek yang nyata secara fisik dengan obyek yang dihasilkan oleh komputer dan ditampilkan dengan menggunakan peralatan display yang biasa, seperti monitor ataupun layar *mobile phone*.

Untuk memudahkan merepresentasikan suatu objek 3D yang sudah dibuat ke dalam dunia nyata, dapat dilakukan pelacakan suatu tanda (*marker*) khusus. Selain itu, kemampuan untuk melacak suatu tanda tertentu dalam suatu kumpulan objek memberikan keuntungan yang cukup besar terhadap interaksi manusia dan komputer. Untuk pelacakan tersebut dapat digunakan media kamera sebagai indera untuk menerima input.

Beberapa aplikasi AR didesain untuk mengkombinasikan representasi virtual dengan persepsi dari dunia nyata. Representasi virtual memberikan informasi tambahan yang mana tidak didapatkan oleh user yang tidak memakai alat bantu. *Augmented reality* dapat digunakan untuk membantu memberikan gambaran atau simulasi kepada pengguna. Misalnya untuk visualisasi pengenalan *concept car* atau visualisasi buku cerita 3D, menampilkan maket bangunan secara virtual, dll.

Studi Pustaka

a. Augmented Reality

Menurut William R & Allan B, 2003; Bowo Dwi, 2012, *Augmented reality* memiliki pengertian sebagai sebuah istilah untuk lingkungan buatan yang menggabungkan dunia *virtual* yang dihasilkan oleh komputer dengan dunia nyata pada media tertentu seperti monitor / layar LCD / *Head Mounted Display*. Sehingga terjadi interaksi antara objek nyata tersebut dengan objek virtual. *Augmented reality* menampilkan objek virtual yang dibuat dengan bantuan komputer sebagai ganti obyek nyata yang tampak dari sensor pada komputer sementara obyek nyata masih

tetap tampak. Hal ini menyebabkan augmented reality lebih berupa gabungan obyek nyata dengan obyek virtual.

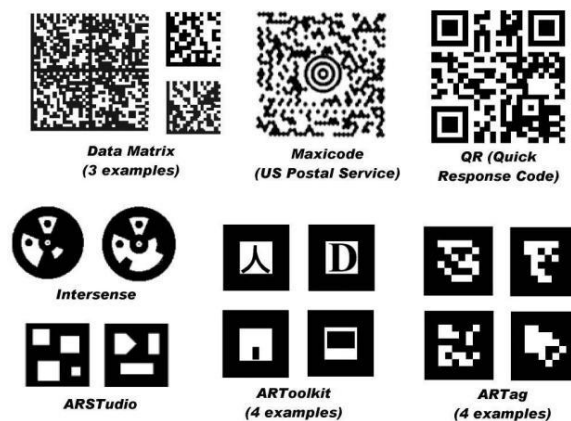
Augmented reality termasuk variasi bentuk dari *virtual reality*. *Virtual reality* adalah sebuah media komunikasi yang interaktif yang melibatkan keberadaan sebuah dunia virtual yang dihasilkan dengan menggunakan komputer (Sherman, 2003). Dunia virtual ini dapat disensor oleh indera namun tidak berwujud secara nyata. Semua obyek yang dihasilkan adalah rekayasa dari komputer. Sedangkan perbedaan *Augmented reality* dengan *virtual reality* adalah adanya keterlibatan obyek pada dunia nyata. Pada teknologi *Augmented reality*, input berupa informasi yang terlihat pada dunia nyata. Sistem akan mengenali obyek dari dunia nyata. Hasil pengenalan tersebut yang akan menghasilkan respon berupa kemunculan obyek virtual secara real time, seperti terlihat pada **Error! Reference source not found.**Gambar 1.



Gambar 1. perbandingan apa yang dilihat (kiri) dengan output *Augmented reality* (kanan)
 Sumber: <http://www.creativeapplications.net/wp-content/uploads/2011/12/Amnon-Owed-ARtut-01.jpg>

b. *Planar Marker System*

Banyak praktisi *computer vision* menggunakan pola 2D untuk menyimpan informasi seperti terlihat pada Gambar (Hirzer, 2008). Area yang menggunakan pola tersebut beraneka ragam, dari sistem industri (dimana tanda didesain untuk digunakan dalam pencarian lokasi, seperti *Maxicode* yang digunakan oleh kantor Pos di Amerika) maupun *DataMatrix* dan *QR code* yang digunakan untuk memberi label suatu barang ataupun yang sudah umum di kalangan masyarakat, seperti pada Gambar 2.

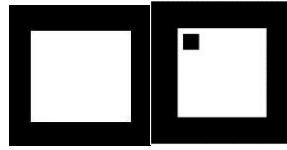


Gambar 2. Macam – macam *planar marker system*

Untuk mengurangi sensitivitas kondisi pencahayaan, *planar marker system* biasanya menggunakan warna bitonal (2 warna) yaitu kombinasi hitam dan putih (Hirzer,2008). Hal ini menyebabkan tidak diperlukannya identifikasi warna. Selain itu beberapa *marker system* menggunakan sistem data bit untuk menyimpan data.

Desain suatu *marker* biasanya ditentukan dari kebutuhan aplikasi. Pada aplikasi *Augmented reality*, marker yang mudah terbaca dan sudut pandang yang luas merupakan hal yang sangat penting. Aplikasi harus tetap dapat mendeteksi meskipun *image* di yang ditangkap kamera terdistorsi. Informasi yang disimpan pada *marker* sebaiknya tidak terlalu padat untuk meningkatkan waktu pendeteksian. Tetapi pada umumnya *marker Augmented reality* hanya menyimpan informasi yang sedikit, yaitu hanya id yang digunakan untuk membedakan *marker* yang satu dengan yang lain.

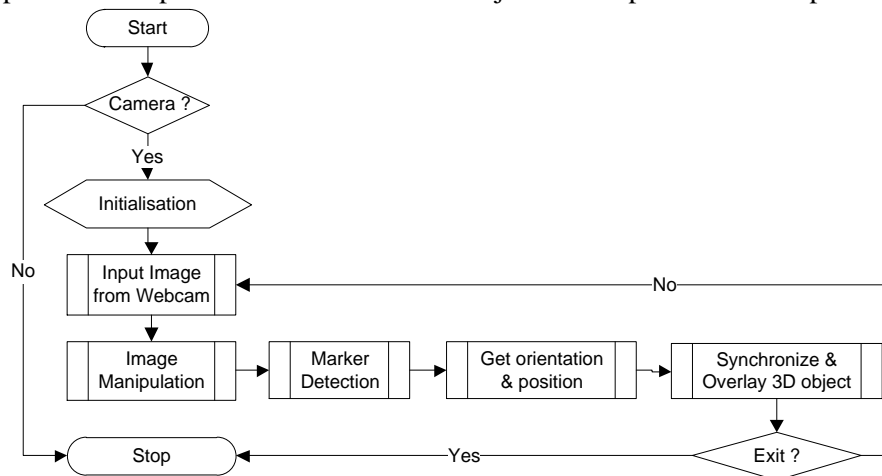
Augmented reality System memerlukan setidaknya minimal 4 *unique point* yang dapat digunakan untuk memprediksi jarak (Hirzer , 2008). Pada umumnya *marker* yang digunakan terdapat garis tepi saling tegak lurus (*quadrilateral outline*). Sehingga 4 garis tepi tersebut dapat digunakan sebagai titik acuan dalam perhitungan perkiraan jarak dalam 3 dimensi. Contoh *marker* seperti pada Gambar 3.



Gambar 3. Spesifikasi minimal *marker* (kiri) dan contoh *marker* dengan fitur khusus (kanan)

Metodologi Penelitian

Dalam pengembangan sistem pendeteksian ini, pertama mendesain alur kerja sistem, lalu melakukan implementasi berdasarkan persamaan literatur yang menunjang dan terakhir melakukan uji coba sistem. Dalam alur sistem yang dibuat terdapat beberapa tahapan yang harus dilakukan. Adapun garis besar rancangan sistem yang akan digunakan dapat dilihat pada Gambar 4. Sedangkan proses untuk pendeteksian *marker* akan dijabarkan seperti flowchart pada Gambar 5.



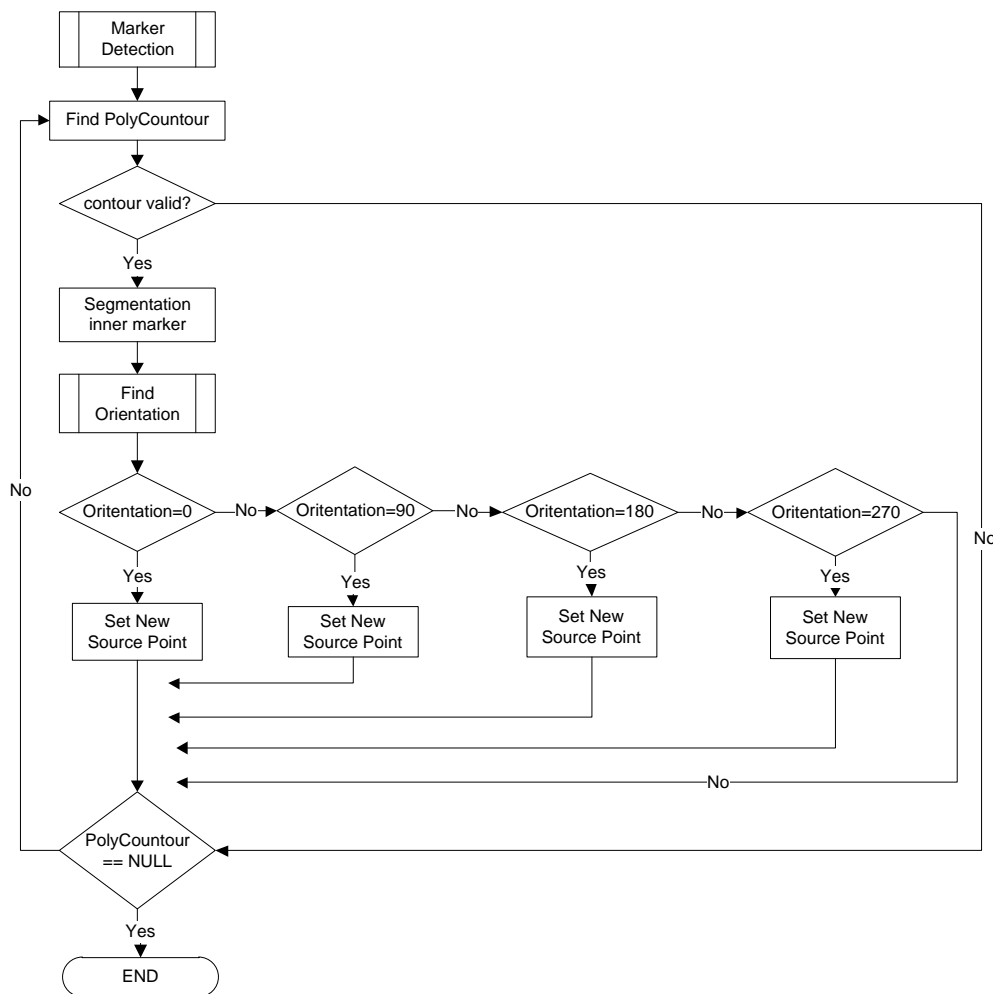
Gambar 4. Flowchart garis besar sistem

Input dari sistem berupa *image* yang ditangkap dengan menggunakan kamera *webcam*. Oleh karena itu, maka tahap pertama yang perlu dilakukan adalah mengecek koneksi dari kamera *webcam* dengan sistem. *Initialization* adalah proses inisialiasi variabel-variabel yang akan digunakan dalam aplikasi. Memberi nilai untuk flag, mengambil nilai intrinsic camera, menyiapkan matrix kalibrasi dan storage contour. Setelah proses penyiapan variable dan nilai awal yang dibutuhkan, langkah berikutnya adalah merekam input image. *Input image* adalah gambar bergerak yang direkam dari *webcam* yang terhubung dengan sambungan USB. Sistem akan mendeteksi kamera yang terhubung, lalu mengambil data *image* untuk dipersiapkan untuk tahap selanjutnya, yaitu *camera calibration*.

Camera calibration merupakan suatu tahap pilihan untuk melakukan koreksi output *image* dari *webcam* yang dikarenakan oleh titik fokus lensa yang kurang tepat, *skew factor*, *scaling factor* maupun *lens distortion*. Pada tahap ini akan dilakukan pengambilan 10 *image checkerboard* dengan sudut dan lokasi yang berbeda. Kumpulan data *image* tersebut akan dilakukan perhitungan untuk mendapatkan parameter kamera sebagai acuan koreksi.

Pada tahap *Image Manipulation* dilakukan perubahan image ke grayscale. Lalu di lakukan proses smoothing untuk menghilangkan noise dan dilanjutkan dengan proses binerisasi (image hanya berupa hitam atau putih). Proses berikutnya adalah dilakukan thresholding. Thresholding ini dilakukan untuk memperkecil *range* kontur yang akan diproses oleh *marker detection*. *Thresholding* yang digunakan adalah *adaptive thresholding* menyesuaikan dengan kondisi pencahayaan dari inputan yang diterima (Greenstead, 2010).

Marker Detection adalah proses dimana perangkat lunak akan mencari lokasi *marker* dari input *image*, dilakukan dengan cara mencari setiap bentuk yang sudut tepinya adalah 4. Apabila sudah terdeteksi maka akan dicatat posisi tiap tepinya. *Get orientation & position of marker*, setelah perangkat lunak berhasil mendeteksi *marker* yang ada, maka dapat diketahui posisi *marker* di dunia nyata. Kemudian untuk mencari orientasi *marker*, yang menjadi pusat perhatian (*point of interest*) kita berikutnya adalah fitur bagian dalam (*inner marker*). Dari *inner marker* tersebut aplikasi akan dicari orientasi suatu fitur yang nantinya akan dipakai sebagai titik pusat / acuan dari *marker* tersebut. Sehingga dari sana dapat dilakukan estimasi rotasi yang terjadi. Cara perhitungan estimasi rotasi adalah dengan menghitung sudut antara garis pusat dengan garis setelahnya terhadap garis *horizontal* dari titik pusat tersebut. *Synchronize & overlay 3D object*, merupakan tahap dimana dilakukan pengiriman data *image* dari *webcam* ke interface.



Gambar 5. Flowchart marker detection

Pada tahap ini akan dilakukan pencarian *marker*. *Marker* yang dideteksi diharuskan memiliki spesifikasi minimal seperti yang sudah dijelaskan di atas, dengan penambahan suatu fitur berbentuk kotak kecil. Fitur tersebut akan digunakan sebagai referensi untuk menentukan posisi titik 0 / titik pusat. Sehingga kita dapat mengetahui posisi acuan yang nantinya akan digunakan untuk proses penentuan orientasi. Langkah utama dalam pengenalan *marker* adalah dengan mengenali contour dari gambar (Gopall,2008).

Pada tahap “Find Orientation” dilakukan pendeteksian sederhana untuk mencari fitur kotak kecil. Dari pendeteksian fitur tersebut, kita dapat mengetahui bahwa titik itu adalah posisi titik 0 /

titik pusat nya (Peter,2005). Apabila kita merotasi *marker*, dapat kita ketahui dimana titik pusat yang baru berada. Sehingga dari sana dapat kita cari orientasi *marker* tersebut. Cara pendeteksian dilakukan dengan menghitung jumlah pixel berwarna putih yang ada pada beberapa sudut perputaran. Yaitu apabila sudut putar 0° ,maka koordinat (10,10) atas kiri dihitung. Untuk perputaran 90° akan dicari pada koordinat atas kanan (60,10). Untuk 180° bawah kanan (60,60). Untuk 270° pada koordinat (10,60) bawah kiri. Apabila jumlah tersebut melebihi nilai yang sudah ditentukan, maka disimpulkan bahwa orientasi *marker* berada pada sudut tersebut.

Pengujian Sistem

Pengujian dilakukan terhadap dua aspek, yaitu jarak antara webcam dengan marker serta kemiringan bidang marker yang terlihat dari webcam.

a. Pengujian jarak optimal pendeteksian *marker*

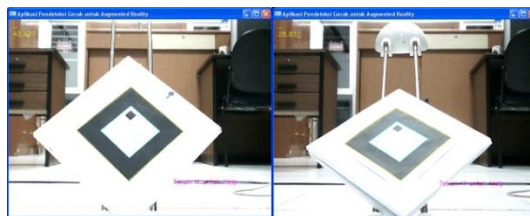
Pengujian aspek ini dilakukan dengan variabel jarak dan variabel rotasi. Pengujian dengan variabel jarak yaitu, *marker* akan di uji coba pada jarak yang berbeda-beda dimulai dari 20 cm dari kamera, 30 cm, kemudian kelipatan 30 cm hingga 3.6 meter. Sedangkan variabel rotasi adalah *marker* diuji dengan sudut rotasi $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ$ Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. hasil pengujian jarak dan rotasi marker

		rotasi marker				
		0°	45°	90°	135°	180°
Jarak (cm)	20	359.26	43.64	91.45	133.42	179.68
	30	359.5	42.06	91.53	133.95	180
	60	357.09	42.14	88.06	132.93	177.09
	90	355.71	42.88	86.99	135	175.6
	120	356.05	42.14	87.95	130.82	176.19
	150	3.01	38.66	92.86	131.42	174.56
	180	356.63	39.81	96.71	133.03	183.37
	210	6.34	39.81	96.71	132.51	173.29
	240	3.81	39.81	7.59	42.27	172.41
	270	4.09	45	12.09	39.29	4.09
	300	4.76	45	5.19	34.99	4.76
	330	5.71	36.87	5.71	37.87	5.71
	360	-	-	-	-	-

b. Pengujian tingkat kemiringan *marker* pada jarak dan rotasi tertentu

Pada pengujian kali ini faktor rotasi marker juga akan diperhitungkan. Marker akan di miringkan pada sudut $30^\circ, 60^\circ, 120^\circ, 150^\circ$ pada jarak 30 cm hingga 210 cm dan akan dirotasi sebanyak 45° Hasil dari pengujian kemiringan dapat dilihat pada Tabel 2. Contoh posisi kemiringan dapat dilihat pada Gambar 6.



Gambar 6. Jarak marker 30 cm dari kamera dengan kemiringan 120° dan 150°

Tabel 2. Perbedaan sudut estimasi rotasi

		Tingkat Kemiringan			
		30°	60°	120°	150°
Jarak (cm)	30	16.07	2.6	5	9.4
	60	16.93	1.47	3.47	8.37
	90	7.43	5.39	7.85	13.5
	120	15.75	3.99	3.99	9.46
	150	9.78	6.71	9.46	4.09
	180	14.53	3.81	5.19	-
	210	5.19	8.75	9.78	-

Kesimpulan

Berdasarkan sistem yang telah dikembangkan dan hasil pengujian yang telah ditentukan, dapat disimpulkan beberapa hal sebagai berikut:

- Estimasi rotasi dengan cara menghitung 2 garis terkadang memberikan hasil yang relative akurat dengan perbedaan kurang lebih ± 10 derajat. Hal ini dikarenakan *parallax* atau perbedaan posisi sudut pengamatan yang menyebabkan terjadi perbedaan posisi apa yang dilihat kamera dengan yang sebenarnya.
- Sistem pendeteksian *marker* tergolong memiliki performa yang cukup bagus. *Marker* dapat dideteksi hingga pada jarak 3.3 meter(330 cm). Namun jarak yang efektif untuk pengenalan rotasi adalah 20 cm hingga 2.1 meter(210 cm) dari kamera.
- Derajat kemiringan *marker* memiliki pengaruh dalam kinerja pendeteksian *marker*. Karakteristik jarak efektif maksimal *marker* dari kamera adalah :
 - Pada derajat kemiringan *marker* 30° adalah 210 cm.
 - Pada derajat kemiringan *marker* 60° adalah 240 cm.
 - Pada derajat kemiringan *marker* 90° adalah 270 cm.
 - Pada derajat kemiringan *marker* 120° adalah 180 cm
 - Pada derajat kemiringan *marker* 150° adalah 150 cm
- Tingkat kemiringan *marker* memiliki pengaruh terhadap estimasi rotasi
- Sistem pendeteksian *marker* dapat berjalan pada webcam dengan spesifikasi resolusi rendah, namun tingkat akurasi estimasi rotasi semakin menurun.

Daftar Pustaka

- Ariyanto, Bowo Dwi ; M. Hariadi. (2010). simulasi perilaku pergerakan objek 3D media augmented reality berbasis logika fuzzy, from <http://digilib.its.ac.id/ITS-Master-3100012045760/18002>
- Gopall, D.J, Jayanthi,S (2008). A simple Scheme for Contour Detection. Retrieved 24 February 2012, from: <http://cvit.iiit.ac.in/papers/>
- Greensted, Andrew (2010). *Otsu Thresholding*. Retrieved Februari 23, 2012, from: <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>
- Martin, Hirzer. (2008). *Marker Detection for Augmented Reality Application*. Retrieved October 11, 2011, from: http://studierstube.icg.tu-graz.ac.at/thesis/marker_detection.pdf
- Peter, Michael, Stephen, Erik. (2005). *Fundamental of Computer Graphics(2nd ed.)*, India : A.K Peters,LTD
- Sherman, William R; Craig, Alan B. (2003). *“Understanding Virtual Realty, Interface, Application and Design”*. New York : Elsevier