PROYEK AKHIR

SISTEM AUGMENTED REALITY UNTUK ANIMASI GAMES MENGUNAKAN CAMERA PADA PC

Oleh :
MAS ALI BAHTIAR
NRP. 7209.040.513

Dosen Pembimbing:
Akuwan Saleh, SST
NIP. 196711231989021001

Muh. Agus Zainudin, S.T M.T
NIP. 197808182008011015

JURUSAN TEKNIK TELEKOMUNIKASI
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
INSTITUT TEKNOLOGI SEPULUH NOPOMBER
2011

PROYEK AKHIR

SISTEM AUGMENTED REALITY UNTUK ANIMASI GAMES MENGGUNAKAN CAMERA PADA PC

Oleh :
MAS ALI BAHTIAR
NRP. 7209.040.513

Dosen Pembimbing:
Akuwan Saleh, SST
NIP. 196711231989021001

Muh. Agus Zainudin, S.T M.T
NIP. 197808182008011015

JURUSAN TEKNIK TELEKOMUNIKASI
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
INSTITUT TEKNOLOGI SEPULUH NOPOMBER
2011

SISTEM AUGMENTED REALITY UNTUK ANIMASI GAMES MENGGUNAKAN CAMERA PADA PC

Oleh :
MAS ALI BAHTIAR
7209.040.513

Proyek akhir ini sebagai salah satu syarat untuk
Memperoleh gelar Sarjana Sains Terapan (S.ST)
di
Politeknik Elektronika Negeri Surabaya,
Institut Teknologi Sepuluh Nopember Surabaya

Disetujui Oleh :

Tim penguji :

1.
Drs.Miftahul Huda, M.T
NIP. 196310121993031002

2.
Tribudi Santoso S.T. M.T
NIP. 197001051990032001

3.
Arifin S.T M.T.
NIP. 196005031988031004

Dosen pembimbing :

1.
Akuwan Saleh, SST
NIP. 196711231989021001

2.
Muh. Agus Z, S.T M.T
NIP. 197808182008011015

Mengetahui
Ketua Jurusan Telekomunikasi

Arifin , S.T. M.T
NIP. 196005031988031004

SISTEM AUGMENTED REALITY UNTUK ANIMASI GAMES MENGGUNAKAN CAMERA PADA PC

Abstrak

Augmented reality menjadi sangat populer saat ini karena selain menarik, juga dapat di tampilkan secara realtime, Sebuah permainan real-time dengan menggunakan marker untuk menampilkan animasi *games* secara 3D, *Augmented Reality* sendiri adalah sebuah teknologi yang menempatkan suatu gambar virtual dari grafis computer pada dunia nyata, atau dengan kata lain penggabungan antara dunia nyata dengan dunia virtual, serta merupakan salah satu contoh aplikasi bidang seni dan teknologi yang cukup banyak di gemari saat ini. Oleh karena itu, Pada proyek akhir ini di buatlah sistem *Augmented Reality* untuk animasi *games* dengan menggunakan camera sebagai media untuk pembacaan simbol input dari animasi *games* 3D.

Identifikasi marker merupakan suatu cara yang digunakan untuk mengidentifikasi simbol yang akan diterjemahkan maksud dan tujuannya. Dalam proyek akhir ini akan dibuat suatu perangkat lunak yang dapat meng-identifikasi marker melalui citra yang ditangkap oleh kamera yang nantinya ditampilkan dalam bentuk animasi *games* 3D. Proses yang dilakukan meliputi pembacaan simbol marker menggunakan kamera kemudian melakukan tahapan pre *Processing* yaitu proses segmentasi untuk perbandingan simbol marker dengan simbol yang telah menjadi acuan sebelumnya. Bila simbol marker merupakan citra yang memiliki kemiripan dengan data referensi, Maka hasil pengenalan citra itulah yang nantinya akan digunakan untuk menampilkan animasi *games* 3D.

Kata kunci : *Real time, Marker, Animasi games*

AUGMENTED REALITY SYSTEM FOR ANIMATED GAMES USING CAMERA ON PC

Abstract

Augmented reality become very popular now days because in addition to exciting, can also be displayed in realtime, A game of real-time using the marker to display an animation games in 3D, Augmented Reality is a technology that puts a virtual image of computer graphics in the real world , or in other words, the merger between the real world with virtual worlds, and is one example of the application field of art and technology that pretty much enjoy doing today. Therefore, this final project in Augmented Reality systems to create animated games using the camera as a medium for reading the input symbol of the animated 3D games.

The identification marker is used to identify the symbols that will translate the goals and objectives. In this final project will be made a software that can identify markers through the image captured by a camera that will be displayed in the form of animated 3D games. The process was conducted on the reading of marker symbols using the camera and then do the pre processing stage of the process of segmentation for comparison with the marker symbol that has become a symbol of the previous reference. When the marker is a symbol image that has similarities with the reference data, then the results of image recognition that is what will be used to display animated 3D games.

Keywords: Real time, Marker, Animation games.

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah selama pembuatan buku Proyek Akhir ini, sehingga buku Proyek Akhir ini dapat terselesaikan dengan baik. Proyek Akhir ini berjudul:

"SISTEM AUGMENTED REALITY UNTUK ANIMASI GAMES MENGGUNAKAN CAMERA PADA PC"

Pembuatan dan penulisan buku Proyek Akhir ini sebagai salah satu persyaratan untuk menyelesaikan studi di Politeknik Eletronika Negeri Surabaya – ITS dengan jurusan Teknik Telekomuikasi.

Selama penyusunan buku Proyek Akhir ini, banyak hambatan yang ditemui oleh penulis. Dengan rahmat Allah SWT dan bimbingan dari dosen pembimbing serta kemauan yang keras sehingga semua hambatan dan permasalahan dapat teratasi.

Penulis menyadari dalam pembuatan buku Proyek Akhir ini masih banyak kekurangan pada proses pengerjaan. Penulis berharap semoga buku Proyek Akhir ini menjadi sesuatu yang bermanfaat bagi pembaca. Penulis berharap saran dan kritik yang membangun dari pembaca.

UCAPAN TERIMA KASIH

Dalam pelaksanaan dan pembuatan proyek akhir ini penulis banyak menerima bantuan dari berbagai pihak.. Selalu penulis panjatkan rasa syukur dengan sebesar–besarnya kehadirat Allah SWT atas semua karunia yang telah diberikan-Nya sehingga penulis dapat menyelesaikannya dengan baik. Dan tanpa menghilangkan rasa hormat yang mendalam penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu penulis antara lain :

1. Bapak dan Ibu tercinta yang telah memberi banyak dukungan baik moral, material, spiritual serta doa. Tugas akhir ini adalah persembahanku untuk kalian berdua.
2. Bapak Dr. Ir. dadet, M.Eng, selaku Direktur Politeknik Elektronika Negeri Surabaya.
3. Bapak Arifin,S.T, M.T , selaku Ketua Jurusan Teknik Telekomunikasi
4. Bapak Akuwan saleh, S.ST selaku dosen pembimbing.
5. Seluruh Dosen Teknik Telekomunikasi dan seluruh Dosen yang Penguji.
6. Ustadz. Baidun Maknun yang telah memberi banyak dukungan baik secara moral maupun spiritual serta bimbingannya selama ini.
7. Teman-teman LJ telkom yang banyak membantu memberikan semangat dalam penyelesaian tugas akhir ini. Mohon maaf yang sedalam dalamnya bila ada salah kata atau tingkah selama mengerjakan proyek akhir ini.
8. Dan kepada pihak yang telah membantu saya dalam menyelesaikan Proyek Akhir ini yang tidak bisa saya sebutkan satu-persatu.

Segala ucapan terimakasih dari kami tentunya belum cukup, semoga Allah SWT. Membalas kebaikanNya. Akhir kata, semoga proyek akhir ini dapat memberikan manfaat dan tambahan pengetahuan bagi pembaca.

DAFTAR ISI

Halaman ini sengaja di kosongkan

DAFTAR GAMBAR

DAFTAR TABEL

Halaman ini sengaja di kosongkan

BAB 1
PENDAHULUAN

1.1 LATAR BELAKANG

Digital *Image Processing* , Analisis dan Computer telah dikembangkan dan di aplikasikan dengan mengesankan selama beberapa dekade ini. Perkembangan dan aplikasi *image* ini telah memimpin teknologi di beberapa bidang seperti komunikasi digital dan internet, penyiaran (broadcasting), alat kedokteran, sistem multimedia, biologi, ilmu pengetahuan material, robot, dan manufaktur, sistem intelligent sensing, remote sensing, seni grafik dan proses print. Pertumbuhan yang pesat ini direfleksikan dengan diterbitkannya paper di jurnal ilmiah internasional dan dengan diluncurkannya buku tentang Pemrosesan *Image* Digital. Pada buku dan paper tersebut dibahas tentang beberapa algoritma pemrosesan dan analisa *image* dan beberapa latihan aplikasi pengolahan *image*. Dan pada umumnya paper tersebut disebarkan dalam jurnal-jurnal ilmiah yang bersangkutan. Dan beberapa juga dijelaskan secara khusus dalam buku. Khususnya dalam era milenium ke III ini telah banyak diciptakan alat-alat yang berteknologi tinggi yang menggunakan *image* sebagai inputnya atau sensor. Sebagai contoh yaitu Sistem *Augmented Reality* yang di gunakan untuk modul pembelajaran anak dalam pengenalan bentuk bangun ruang. Contoh lain yaitu sistem augmented reality untuk menampilkan animasi *games*. Dan masih banyak yang lainnya yang sangat menarik bila ingin mendalaminya.

Karena pengolahan citra merupakan salah satu proses intelligent dengan fleksibilitas yang sangat tinggi, maka pada proyek akhir ini kami mencoba untuk melakukan pembuatan animasi *games* menggunakan marker yang akan di baca oleh camera pada PC. Hal ini dimaksudkan untuk menampilakan secara nyata bentuk animasi *games* secara 3D yang akan di baca oleh camera pada PC . Secara garis besar prosesnya adalah dengan pembacaan citra pada marker yang secara automatis akan dicapture oleh kamera , camera akan mendeteksi marker tersebut dan akan di bandingkan dengan gambar marker yang telah mejadi acuan. Kemudian bila marker di kenali maka akan di tampilkan animasi *games* secara 3D pada layar monitor.

1.2 PERUMUSAN MASALAH

Permasalahan yang akan ditangani adalah bagaimana menentukan teknik untuk mengidentifikasi marker dengan baik dan mengolah marker dengan menggunakan *image Processing* hingga dapat menampilkan bentuk 3D animasi *games* secara realtime, serta membuat sebuah animasi *games* yang menarik.

1.3 BATASAN MASALAH

Batasan masalah dalam pembuatan proyek akhir kali ini adalah hanya menggunakan 1 marker sebagai inputan, Dimana marker ini nantinya sebagai trigger untuk menampilkan animasi tersebut bila marker tersebut benar dan sesuai dengan data acuan. animasi 3D ini juga dapat di control menggunakan keyboard.

1.4 TUJUAN :

Adapun tujuan dalam pembuatan proyek akhir ini adalah untuk membuat sebuah animasi *games* 3D yang akan di tampilkan pada layar computer secara real time dengan symbol marker yang akan di baca oleh camera pada PC, sehingga seolah-olah terdapat sebuah animasi 3D yang berjalan di dunia nyata, pada hal sebenarnya hanyalah animasi virtual yang semu hasil pengolahan citra yang di render oleh program yang telah di buat .

1.5 METODOLOGI

Perencanaan dalam pembuatan system *Augmented reality* untuk animasi *games* menggunakan camera pada PC melalui tahap sebagai berikut:

1.5.1Studi Literatur

Dalam pembuatan proyek akhir ini harus terlebih dahulu mempelajari tentang *image Processing* untuk pembacaan marker dan pengolahannya serta penbuatan animasi *games* 3D yang menarik..

1.5.2 Perancangan Sistem

Prinsip kerja dari system ini adalah identifikasi marker melalui citra yang ditangkap oleh kamera yang nantinya ditampilkan dalam bentuk animasi *games* secara 3D. Proses yang dilakukan meliputi pembacaan simbol marker menggunakan kamera kemudian melakukan tahapan pre *Processing* yaitu proses segmentasi untuk perbandingan simbol marker dengan simbol yang telah menjadi acuan sebelumnya. Bila simbol marker merupakan citra yang memiliki kemiripan dengan data referensi, Maka hasil pengenalan citra itulah yang nantinya akan digunakan untuk menampilkan animasi *games* 3D.

1.6 SISTEMATIKA PENULISAN

Dalam penulisan laporan Buku tugas akhir ini, penulis menggunakan metode studi literatur/kepustakaan, yang disusun dalam empat bab sebagai berikut:

BAB I : Membahas tentang pendahuluan, yang terdiri dari Latar Belakang,Rumusan masalah, Batasan masalah, Tujuan Proyek Akhir, Metodologi, dan Sistematika Penulisan Laporan.

BAB II : Berisi tentang Teori penunjang yang meliputi bahan dan alat, cara kerja, tempat dan waktu pelaksanaan dan cara pengumpulan data atau analisis data..

BAB III : Berisi pembahasan tentang Perencanaan Sistem yang akan di buat.

BAB IV : Berisi pembahasan tentang Pembuatan sistem / Implementasi dari perencanaan bab sebelumnya serta hasil penelitian/hasil dari analisis data dan pembahasannya.

BAB V: Bab Penutup yang berisi tentang Kesimpulan dan Saran penulis.

Halaman ini sengaja di kosongkan

BAB 2
TEORI PENUNJANG

2.1 *Image Prosesing*

Data atau informasi tidak hanya disajikan dalam bentuk teks, tetapi juga dapat berupa gambar, audio (bunyi, suara, musik), dan video. Keempat macam data atau informasi ini sering disebut multimedia. Era teknologi informasi saat ini tidak dapat dipisahkan dari multimedia. Situs web (website) di Internet dibuat semenarik mungkin dengan menyertakan visualisasi berupa gambar atau video yang dapat diputar. Beberapa waktu lalu istilah SMS begitu populer diantara pengguna telepon genggam (handphone). Tetapi, saat ini orang tidak hanya dapat mengirim pesan dalam bentuk teks tapi juga dalam bentuk gambar maupun video yang dikenal dalam layanan MMS (Multimedia Message Service).

Citra (*image*) adalah istilah lain untuk gambar sebagai salah satu komponen multimedia memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Maksudnya sebuah gambar dapat memberikan informasi lebih banyak daripada informasi tersebut disajikan dalam bentuk teks.

Pengolahan gambar digital atau *Digital Image Processing* adalah bidang yang berkembang sangat pesat sejalan dengan kemajuan teknologi pada industri saat ini. Fungsi utama dari Digital *Image Processing* adalah untuk memperbaiki kualitas dari gambar sehingga gambar dapat dilihat lebih jelas tanpa ada ketegangan pada mata, karena informasi penting diekstrak dari gambar yang dihasilkan harus jelas sehingga didapatkan hasil yang terbaik. Selain itu DIP digunakan untuk memproses data yang diperoleh dalam persepsi mesin, yaitu prosedur – prosedur yang digunakn untuk mengekstraksi informasi dari gambar informasi dalam bentuk yang cocok untuk proses komputer [2].Proses pengolahan citra digital dengan menggunakan komputer digital adalah terlebih dahulu mentransformasikan citra ke dalam bentuk besaran-besaran diskrit dari nilai tingkat keabuan pada titik-titik elemen citra. Bentuk citra ini disebut citra digital. Elemen-elemen citra digital apabila ditampilkan dalam layar monitor akan menempati sebuah ruang yang disebut dengan pixel (picture elemen/pixel).

Teknik dan proses untuk mengurangi atau menghilangkan efek degradasi pada citra digital meliputi perbaikan citra (*image* enhancement), restorasi citra (*image* restoration), dan transformasi spasial (spatial transformation). Subyek lain dari pengolahan citra digital diantaranya adalah pengkodean citra (*image* coding), segmentasi citra (*image* segmentation), representasi dan diskripsi citra (*image* representation and description).

2.1.1 *Thresholding*

Selama proses *thresholding*, individu pixel dalam gambar ditandai sebagai "objek" pixel jika nilai mereka lebih besar dari beberapa nilai *threshold* (asumsi benda menjadi lebih terang daripada latar belakang) dan sebagai latar belakang "pixel" sebaliknya. Konvensi ini dikenal sebagai *threshold above.* Varian termasuk *threshold* di bawah ini, yang merupakan kebalikan dari *threshold* above; *threshold outside*, di mana sebuah pixel diberi label "obyek" jika nilai adalah antara dua ambang, dan *threshold outside*, yang merupakan kebalikan dari *threshold inside* . Biasanya, sebuah pixel objek diberi nilai "1" sementara pixel background diberikan sebuah nilai dari Akhirnya, suatu citra biner yang dibuat oleh masing-masing pixel warna putih atau hitam, tergantung pada label pixelnya yaitu bernilai "0.".

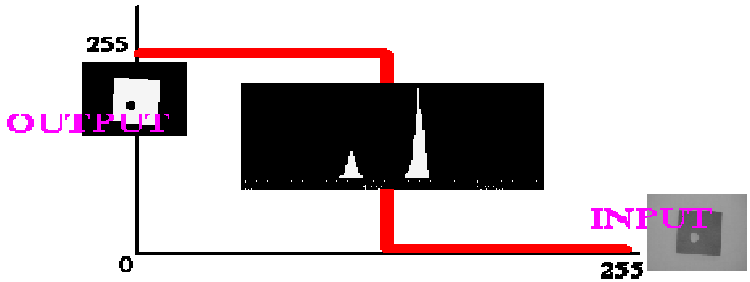
2.1.2 Seleksi *Threshold*

Parameter kunci dalam proses *thresholding* adalah pilihan dari nilai ambang (atau nilai-nilai, seperti yang disebutkan sebelumnya). Beberapa yang berbeda metode untuk memilih ambang ada; pengguna dapat secara manual memilih nilai ambang, atau algoritma *thresholding* dapat menghitung nilai secara otomatis, yang dikenal sebagai *thresholding* otomatis. Sebuah metode sederhana akan memilih mean atau median nilai, dasar pemikiran adalah bahwa jika pixel objek lebih terang dari latar belakang, mereka juga harus lebih terang dari rata-rata. Dalam gambar bersuara dengan latar belakang seragam dan nilai-nilai objek, median berarti atau akan bekerja dengan baik sebagai ambang pintu, bagaimanapun, ini umumnya tidak akan terjadi. Sebuah pendekatan yang lebih canggih mungkin untuk membuat histogram dari intensitas pixel gambar dan menggunakan jalur lembah sebagai ambang batas. Pendekatan histogram mengasumsikan bahwa ada beberapa nilai rata-rata untuk pixel latar belakang dan objek, tetapi bahwa nilai pixel yang sebenarnya memiliki beberapa variasi di sekitar nilai rata-rata. Namun, ini mungkin komputasi mahal, dan histogram gambar mungkin

tidak jelas poin lembah, sering membuat pilihan ambang akurat sulit. Salah satu metode yang relatif sederhana, tidak memerlukan pengetahuan khusus banyak gambar, dan tahan terhadap noise , adalah sebagai berikut metode iteratif :

1. *Thresholding* awal (T) dipilih, hal ini dapat dilakukan secara acak atau sesuai dengan metode lainnya yang diinginkan.
2. Gambar akan tersegmentasi ke dalam pixel objek dan latar belakang seperti diuraikan di atas, menciptakan dua set:
    1. G_1 = {f(m,n):f(m,n)>T} (object pixels) G_1 = {f (m, n): f (m, n)> T} (pixel obyek)
    2. G_2 = {f(m,n):f(m,n) G_2 = {f (m, n): f (m, n) $\leq$T} (background pixels) (note, f(m,n) adalah nilai dari pixel yang terletak di kolom ^{t} m th column, n th row) T} catatan, f (m, n) ^{h} m, n baris $^{h\,t)}$
3. Rata-rata masing-masing set dihitung.
    1. m_1 = average value of G_1
    2. m_2 = average value of G_2
4. *threshold* baru dibuat yaitu rata-rata m_1 dan m_2
    1. T' = (m_1 + m_2)/2 T '= (m_1 + $m_{2)}$ / 2
5. Kembali ke langkah dua, sekarang menggunakan ambang batas baru dihitung pada langkah empat, terus mengulanginya sampai ambang baru cocok dengan satu sebelum itu (yaitu sampai konvergensi telah tercapai).

Algoritma iteratif adalah kasus satu-dimensi khusus dari k-means algoritma, yang telah terbukti untuk berkumpul di sebuah lokal minimum-yang berarti bahwa batas awal yang berbeda dapat memberikan hasil akhir yang berbeda.



Gambar2.1 *Threshold*, Density slicing

Dalam banyak visi aplikasi, hal ini berguna untuk dapat memisahkan daerah dari *image* sesuai dengan benda-benda yang membuat tertarik, dari daerah *image* yang sesuai dengan background. *Thresholding* sering menyediakan cara yang mudah dan nyaman untuk melakukan segmentasi berdasarkan intensitas yang berbeda atau warna di daerah foreground dan background dari suatu gambar.

input untuk *thresholding* biasanya disebut *grayscale* atau color *image*. Dalam pemakaian paling sederhana, output adalah suatu citra biner yang mewakili segmentasi. pixel hitam sesuai dengan background dan pixel putih sesuai dengan foreground (atau sebaliknya*).* Dalam implementasi sederhana, segmentasi tersebut ditentukan oleh parameter tunggal yang dikenal sebagai intensitas *threshold.* Dalam single pass, setiap pixel dalam gambar dibandingkan dengan *threshold* ini. Jika intensitas pixel ini lebih tinggi dari *threshold*, pixel diatur ke putih pada output. Jika kurang dari *threshold*, ini akan diatur untuk hitam.

Dalam implementasi yang lebih canggih, beberapa batasan bisa ditentukan, sehingga nilai-nilai intensitas band dapat diatur untuk putih sementara segala sesuatu yang lain diatur ke hitam. Untuk warna atau gambar multi-spektral, dimungkinkan untuk menetapkan batasan berbeda untuk setiap warna channel, Varian lain yang umum adalah untuk mengatur semua pixel menjadi hitam yang sesuai dengan background, tapi tinggalkan pixel foreground pada warna asli / intensitas mereka (sebagai lawan untuk memaksa mereka untuk putih), sehingga informasinya tidak hilang.

2.1.3 Nilai Pixel

Setiap pixel yang mewakili suatu gambar yang disimpan di dalam komputer memiliki nilai pixel yang menjelaskan tentang kecerahan pixel atau warna apa yang seharusnya. Dalam kasus yang paling sederhana dari gambar biner , nilai pixel adalah 1 bit angka yang menunjukkan tiap-tiap foreground atau background. Untuk *grayscale image*s , nilai pixel adalah angka tunggal yang mewakili kecerahan pixel. Yang paling umum format pixel adalah byte *image*, dimana jumlah ini disimpan sebagai integer 8-bit memberikan rentang nilai yang mungkin dari 0 sampai 255. Biasanya nol diambil harus hitam, dan 255 diambil untuk menjadi putih. nilai di antara membentuk berbagai nuansa abu-abu.

Multi-spektral gambar dapat berisi bahkan lebih dari tiga komponen untuk setiap pixel, dan dengan ekstensi ini disimpan dalam cara yang sama, sebagai nilai pixel vektor, atau sebagai pesawat warna terpisah.

Meski simple 8-bit integer atau vektor dari bilangan bulat 8-bit adalah jenis yang paling umum dari nilai-nilai pixel yang digunakan, beberapa format gambar yang mendukung berbagai jenis nilai, misalnya untuk integer ditandatangani 32-bit atau nilai floating point. Such values are extremely useful in *image Processing* as they allow *Processing* to be carried out on the *image* where the resulting pixel values are not necessarily 8-bit integers. Nilai tersebut sangat berguna dalam pengolahan citra karena mereka memungkinkan untuk pemrosesan dilaksanakan pada gambar di mana nilai-nilai pixel yang dihasilkan tidak selalu integer 8-bit. Jika pendekatan ini digunakan maka biasanya diperlukan untuk membuat sebuah colormap yang berhubungan rentang tertentu nilai pixel untuk ditampilkan warna tertentu.

Look-Up Tabel atau LUT merupakan dasar untuk banyak aspek dari pengolahan citra. LUT hanyalah sebuah tabel dari *cross-reference* yang menghubungkan nomor indeks untuk nilai output. Yang paling umum digunakan adalah untuk menentukan warna dan nilai intensitas dengan gambar tertentu yang akan ditampilkan, dan dalam konteks ini LUT ini sering disebut dengan sebuah *colormap*. , LUT sering digunakan untuk memetakan kembali nilai-nilai pixel dalam gambar. , LUT sering digunakan untuk memetakan kembali nilai-nilai pixel dalam gambar. Serta penggunaannya dalam colormaps Ini adalah dasar dari banyak umum pengolahan gambar operasi titik seperti *thresholding*, koreksi gamma dan kontras peregangan. Proses ini sering disebut sebagai *anamorphosis.*

di balik Ide colormap adalah bahwa alih-alih menyimpan warna yang pasti untuk setiap pixel dalam sebuah gambar, misalnya dalam bit RGB format-24 , setiap nilai pixel itu adalah diperlakukan sebagai pengganti nomor indeks ke colormap tersebut. Bila gambar yang akan ditampilkan atau diproses secara lain, colormap digunakan untuk mencari warna yang sebenarnya yang sesuai dengan setiap nomor indeks. Biasanya, output nilai-nilai yang tersimpan dalam LUT akan menjadi nilai warna RGB .

Komponen pelabelan Terhubung dengan memindai gambar, pixel demi pixel (dari atas ke bawah dan kiri ke kanan) dalam rangka untuk mengidentifikasi daerah pixel, daerah *yaitu* pixel berdekatan akan berbagi nilai set yang sama dengan intensitas *V*. (Untuk gambar biner *V = {1}*, namun dalam gambar graylevel *V* akan mengambil berbagai nilai, misalnya: *V = {51, 52, 53, ..., 77, 78, 79, 80}* .) sedangkan mbar biner adalah pixel gambar yang hanya memiliki dua kemungkinan nilai intensitas . warna yang di pakai biasanya ditampilkan sebagai hitam dan putih. nilai 0 biasanya untuk hitam, dan 1 atau 255 untuk putih.

2.2 Adobe Flex

ActionScript adalah bahasa pemrograman untuk Adobe Flash Player dan Adobe AIR run-time environments. disini memungkinkan interaktivitas, penanganan data, dan banyak lagi di Flash, Flex, AIR pada konten dan aplikasi. ActionScript dilakukan oleh ActionScript Virtual Machine (AVM), yang merupakan bagian dari Flash Player dan AIR. Aksi Kode script biasanya dikompilasi ke format bytecode (semacam bahasa pemrograman yang ditulis dan dipahami oleh komputer) oleh kompilator, seperti yang dibangun ke dalam Adobe Flash CS3 Professional atau Adobe Flex Builder , atau yang tersedia dalam Adobe Flex SDK dan Flex Layanan Data. bytecode ini tertanam dalam file SWF, yang dijalankan oleh Flash Player dan AIR.ActionScript 3.0 menawarkan model pemrograman yang kuat yang akan akrab bagi pengembang dengan pengetahuan dasar tentang pemrograman berorientasi objek.Beberapa fitur kunci ActionScript 3.0 meliputi:

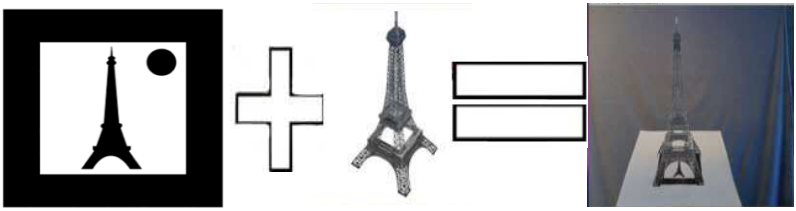
1. Sebuah baru ActionScript Virtual Machine, atau disebut AVM2, menggunakan satu set bytecode instruksi baru dan menyediakan significant perbaikan kinerja
2. Sebuah basis kode kompiler yang lebih moderen yang menganut standar (ECMA 262) ECMAScript dan yang melakukan optimasi lebih baik dari versi kompilator sebelumnya
3. Sebuah antarmuka pemrograman aplikasi diperluas dan ditingkatkan (API), dengan kontrol obyek tingkat rendah dan berorientasi pada obyek model
4. Sebuah inti bahasa berdasarkan ECMAScript (ECMA-262) edisi 4 draft spesifikasi bahasa

5. Suatu API XML berdasarkan ECMAScript untuk XML (E4X) spesifikasi (ECMA-357 Edisi 2). E4X adalah bahasa ekstensi untuk ECMAScript yang menambahkan XML sebagai tipe data asli bahasa.

6. Event Model berdasarkan Document Object Model (DOM) dengan spesifikasi Level 3.

2.3 PV3D

PV3D adalah sebuah library yang ditulis dalam bahasa pemrograman AS3 yang ditujukan untuk mengembangkan aplikasi *augmented reality* atau mixed reality dengan menggabungkan computer vision based tracking libraries (seperti FLARToolKit, FLARManager, Nyartoolkit dan BazAR) dengan 3D scene libary (PV3D) seperti ditunjukkan pada gambar dibawah ini.

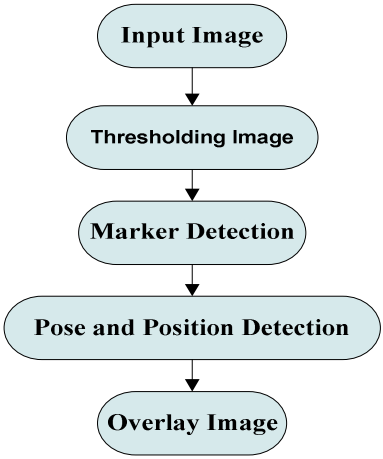


Gambar 2.2. Penambahkan fungsi AR dengan PV3D

Sudah banyak tersedia toolkit untuk membuat dan mengembangkan aplikasi AR, mulai dari low-level programming (e.g. ARToolKit) sampai high-level programming. Keberhasilan ARToolKit untuk membuat aplikasi AR disebabkan karena kesederhanaan tingkat pemrogramannya. Oleh karena itu PV3D dibuat dengan tujuan untuk mempertahankan kesederhanaan yang ada pada FLARToolKit dan membawa FLARToolKit ke generasi yang lebih tinggi. Papaervision3D memiliki library tersendiri untuk urusan rendering model, sehingga kualitas grafis akan menjadi lebih baik ketika dipakai pada sebuah aplikasi[4]. Selain itu, PV3D ini bersifat bebas dan dapat digunakan untuk keperluan akademik. mahasiswa dapat dengan bebas melakukan penelitian tentang AR menggunakan toolkit ini. Toolkit ini tetap mengacu pada prinsip kesederhanaan untuk mengembangkan aplikasi yang akan dibuat (membuat video, menggunakan tracker, dan lain sebagainya). Nantinya, programmer mengembangkan aplikasi mereka dalam bahasa java action script, tetapi programer juga dapat

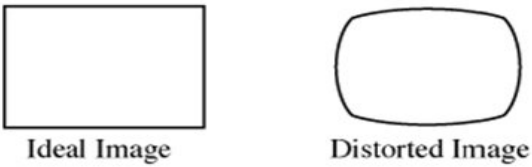
mengembangkan aplikasinya dalam bahasa pemrograman lain seperti Xml, Mxml. PV3D mempunyai berbagai macam fitur yang telah didesain sedemikian rupa untuk dapat lebih memaksimalkan pengembangan aplikasi AR.

2.4 FLARToolkit



Gambar 2.3. Pipeline FLARToolKit

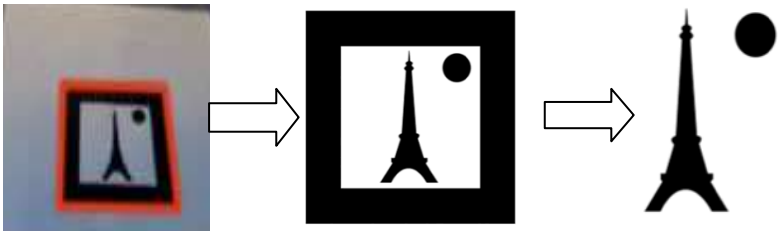
Seperti ditunjukkan pada gambar 3, langkah awal yang harus dilakukan adalah mendapatkan masukan video dari sebuah kamera. Video yang di-streaming secara real-time ini akan diolah oleh sistem untuk dianalisa frame per frame. Sebelum kamera digunakan, kamera harus dikalibrasi terlebih dahulu. Kaliberasi kamera merupakan bagian yang sangat penting dalam proses pengambilan masukan video. Hal ini disebabkan oleh distorsi pada lensa kamera yang tiap-tiap kamera berbeda karakteristiknya (gambar 4). Tujuan dari kalibrasi kamera adalah untuk menghitung tingkat distorsi dari sebuah lensa kamera yang digunakan agar *image* yang dihasilkan mendekati *image* ideal. Parameter ini nantinya digunakan dalam perhitungan pada proses Pose and Position Estimation agar model menara eifel dapat ditampilkan tepat diatas marker



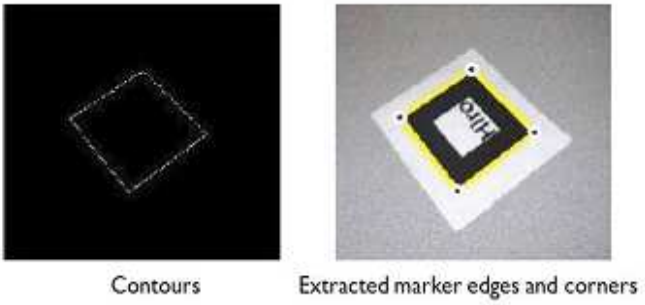
Gambar 2.4. Perbandingan antara *image* yang ideal dengan *image* yang disebabkan oleh faktor distorsi

Video yang diterima selanjutnya akan mengalami proses binarisasi (gray-scale), kemudian nilai *threshold* ditentukan sehingga mengasilkan gambar hitam-putih. Nilai *threshold* berada pada angka 0 – 255 dan secara default, *threshold* bernilai 100. Fungsi dari proses ini adalah untuk membantu sistem agar dapat mengenali bentuk segi empat dan pola di marker pada video yang diterima. Nilai *threshold* dapat dirubah dan disesuaikan dengan kondisi cahaya disekitar marker untuk tetap membuat marker terlihat sebagai segi empat, karena ketika cahaya disekitar marker berkurang ataupun berlebih pada saat proses *thresholding*, sistem tidak dapat mendeteksi marker. Hal ini penting mengingat aplikasi ini bekerja dengan cara mengenali marker.

Setelah video mengalami proses *thresholding*, langkah selanjutnya adalah mendeteksi marker, dimana sistem akan mengenali bentuk dan pola yang ada pada marker. Sistem akan mencari bagian yang memiliki bentuk segi empat dan menandainya. Sistem juga akan menghilangkan area yang tidak berbentuk segi empat sehingga yang akan ditampilkan pada layar hanyalah area yang memiliki bentuk segi empat



Gambar 2.5. Contoh FLARToolKit



Gambar 2.6. Hasil dari contour extraction dan corner detection

Contour extraction dan corner detection digunakan untuk mendapatkan koordinat dari empat sisi dan empat titik sudut pada segi empat yang tersisa setelah proses *image* labeling (gambar 8). Setelah proses ini selesai dilakukan, dua garis paralel pada marker diproyeksikan sehingga persamaan garisnya pada koordinat layar kamera adalah seperti berikut ini :

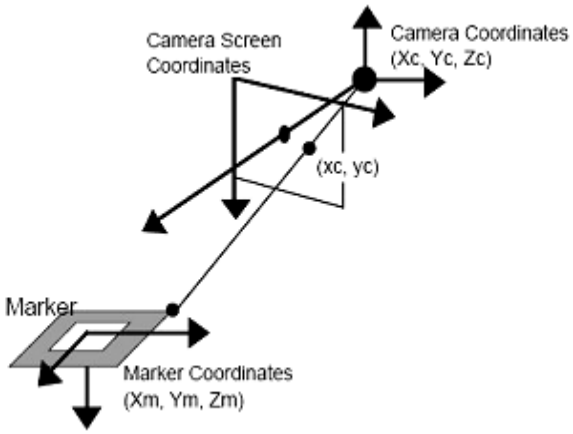
$$a1\ x + b1\ y + c1 = 0 \qquad a2\ x + b2\ y + c2 = 0 \qquad (1)$$

Parameter pada persamaan 1 akan disimpan dan dipakai pada proses selanjutnya.

Karena sudut dari lensa kamera tidak tegak lurus terhadap marker ketika mengambil video, sudut-sudut marker yang dibentuk oleh sisi-sisi segi empat tidak 90° (9). Hal ini membuat pola yang ada didalam marker tidak dapat dikenali dengan baik. Pattern normalization berperan untuk mengubah sudut marker yang tidak 90° menjadi 90° agar pola dapat dikenali dan dicocokkan menggunakan template matching dengan pola (template) yang telah ada pada sistem untuk memperoleh positif ID dari marker tersebut. Sebuah gambar, foto, maupun nama dapat dijadikan pola pada sebuah marker agar sistem dapat mengenali pola itu. Untuk menaruh objek 3D tepat diatas marker, sistem perlu mengetahui koordinat dari marker dan kamera.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & V_{13} & W_x \\ V_{21} & V_{22} & V_{23} & W_y \\ V_{31} & V_{32} & V_{33} & W_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} & \mathbf{V}_{3\times3} & & \mathbf{W}_{3\times1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \mathbf{T_{cm}} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (2)$$



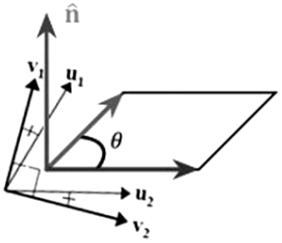
Gambar 2.7. Hubungan antara koordinat marker dengan koordinat kamera

Matrix transformasi (Tcm) dari koordinat marker ke koordinat kamera seperti pada gambar 10 diberikan pada persamaan 2[5] . Untuk marker yang sudah dikenali, nilai dari parameter a1 ,b1 , c1 dan a2 ,b2 , c2 didapatkan ketika proses contour extration. Matrix proyeksi P pada persamaan 3 diperoleh ketika proses kalibrasi kamera. Dengan mengganti xc dan xy pada persamaan 3 untuk x dan y pada persamaan 1 didapat persamaan garis seperti persamaan 4.

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & 0 \\ 0 & P_{22} & P_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = P \begin{bmatrix} Z_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{3}$$

$$\begin{aligned} a_1 P_{11} X_c + (a_1 P_{12} + b_1 P_{22}) Y_c + (a_1 P_{13} + b_1 P_{23} + c_1) Z_c = 0 \\ a_2 P_{11} X_c + (a_2 P_{12} + b_2 P_{22}) Y_c + (a_2 P_{13} + b_2 P_{23} + c_2) Z_c = 0 \end{aligned} \tag{4}$$

Marker segi empat yang digunakan mempunyai empat sisi dimana dua sisi adalah garis yang paralel. Vektor normal dari marker adalah ñ yang dihasilkan dari perkalian cross vektor u1 dan u2, seperti ditunjukkan pada gambar 11. Pada kenyataanya, vektor u1 dan u2 seharusnya tegak lurus, hal ini disebabkan oleh sudut kamera ketika pengambilan gambar yang tidak tegak lurus terhadap marker. Vektor v1 dan v2 dibuat agar memiliki sudut 90° dengan menggunakan nilai dari vektoru1 dan u2untuk memperkecil kesalahan. Setelah v1 dan v2 tegak lurus, v3 dihasilkan dari perkalian cross v1×v2. Nilai v1, v2, dan v3 adalah komponen rotasi pada matrix transformasi Tcm dari koordinat marker ke koordinat kamera seperti yang disampaikan pada persamaan 2. Setelah komponen rotasi V3×3 pada matrix transformasi diketahui, komponen translasi W1, W2, dan W3 dapat diperloleh dengan menggunakan persamaan 2 dan 3. Setelah transformasi matrix didapat, langkah terakhir yang dilakukan adalah menggambar objek virtual 3D pada frame video tepat diatas permukaan marker dan hasilnya dapat dilihat pada keluaran videonya. Dengan demikian model menara eifel virtual seolah-olah ada diatas marker



Gambar 2.8. Dua buah vektor yang tegak lurus : v1 dan v2 didapat dari u1 dan u2 .

2.5 Autodesk 3DMax

Autodesk 3ds Max, 3D Studio MAX sebelumnya, adalah pemodelan, animasi dan rendering paket yang dikembangkan oleh Autodesk Media dan Entertainment. Autodesk memiliki kemampuan pemodelan, arsitektur plugin yang fleksibel dan dapat digunakan pada platform Microsoft Windows. Software Ini sering digunakan oleh pengembang video animation, studio TV komersial dan studio visualisasi arsitektur. Hal ini juga digunakan untuk efek-efek film dan film pra-visualisasi.

Selain pemodelan dan tool animasi, versi terbaru dari 3DS Max juga memiliki fitur shader (seperti ambient occlusion dan subsurface scattering), dynamic simulation, particle systems, radiosity, normal map creation and rendering, global illumination, customize user interface, dan bahasanya scripting untuk 3DMax.

2.5.1 Fitur-Fitur 3DMax

2.5.1.1 MAXScript

MAXScript adalah bahasa scripting, yang dapat digunakan untuk mengotomatisasi gerakan yang berulang-ulang, menggabungkan fungsionalitas yang sudah ada dengan cara baru, mengembangkan tool baru dan user interface dan lebih banyak lagi. Modul Plugin dapat dibuat sepenuhnya dalam MAXScript.

2.5.1.2 Karakter Studio

Karakter Studio adalah sebuah plugin yang sejak versi 4 telah terintegrasi dalam 3D Studio Max, membantu pengguna untuk menghidupkan karakter virtual. Sistem ini bekerja dengan menggunakan rig karakter atau "Biped" yang sebelumnya dibuat dan memungkinkan pengguna untuk menyesuaikan rig agar sesuai dengan karakter mereka serta akan terlihat hidup. Dedicated curve editors dan motion capture data import tools akan membuat Karakter ideal Studio untuk karakter animasi. "Biped" objek memiliki fitur yang berguna otomatis pada produksi siklus berjalan dan jalur gerakan, serta gerakan sekunder.

2.5.1.3 Scene Explorer

Scene Explorer, sebuah tool yang menyediakan hierarchical view of scene data dan analisis, bekerja memfasilitasi adegan-adegan yang lebih kompleks. Scene Explorer memiliki kemampuan untuk menyortir, menyaring, dan pencarian adegan pada setiap jenis objek atau properti

(termasuk metadata). Ditambahkan dalam 3ds Max 2008, itu adalah komponen pertama untuk memfasilitasi. NET managed code di 3ds Max.

2.5.1.4 DWG Import

3DS Max mendukung both import dan menghubungkan file DWG. Peningkatan memori manajemen dalam 3DS Max 2008 memungkinkan scenes yang lebih besar yang harus diimpor dengan beberapa objek.

2.5.1.5 Texture Assignment/Editing

3DS Max menawarkan operasi untuk kreatif tekstur, planar mapping, including tiling, mirroring, decals, angle, rotate, blur, UV stretching, and relaxation; Remove Distortion; Preserve UV; and UV template *image* export. Alur kerja tekstur mencakup kemampuan untuk menggabungkan jumlah tekstur yang tidak terbatas, Material / Map browser dengan dukungan untuk drag-and-drop, dan hirarki dengan thumbnail.

2.5.1.6 General Keyframing

Dua keying mode - mengatur tombol dan tombol otomatis - menawarkan dukungan untuk keyframing workflow. Pengguna dapat membuat animasi dengan mudah. lintasan Animasi dapat dilihat dan diedit langsung di viewport.

2.5.1.7 Constrained animated

Objek dapat di jadikan animasi sepanjang kurva dengan kontrol untuk penyelarasan, kecepatan, kelancaran, dan perulangan, dan sepanjang permukaan dengan kontrol untuk penyelarasan. Objek dapat dibatasi untuk menghidupkan dengan obyek lain dalam banyak cara - termasuk melihat, orientasi dalam ruang koordinat yang berbeda, dan menghubungkan di berbagai titik dalam waktu

2.5.1.8 Skinning

Kulit atau pengubah fisik dapat digunakan untuk mencapai precise control pada deformasi tulang, sehingga karakter deformasi lancar seperti sendi yang bergerak, bahkan di daerah yang paling menantang, seperti bahu. Kulit deformasi dapat dikendalikan dengan menggunakan direct vertex weights, volumes of vertices didefinisikan oleh envelopes, atau keduanya.

2.5.1.9 Integrate cloth Solver

Selain pengubah reactor's cloth modifier, 3DS Max software memiliki cloth simulation engine terpadu yang memungkinkan pengguna untuk mengubah hampir semua objek 3D ke pakian, Lokal simulasi memungkinkan seniman menggantungkan kain secara real time untuk mendirikan negara pakaian awal sebelum pengaturan kunci animasi.

Cloth simmulation dapat digunakan dalam hubungannya dengan kekuatan lain dari 3DS Max dinamis, seperti Space Warps. Beberapa sistem kain independen dapat animasi dengan obyek mereka sendiri dan kekuatan. data deformasi Kain dapat di-cache ke hard drive untuk memungkinkan iterasi tak rusak dan untuk meningkatkan kinerja pemutaran.

2.5.1.10 Integrasi dengan Autodesk Vault

Autodesk Vault plug-in, mengkonsolidasikan 3DS pengguna aset Max dalam satu lokasi, memungkinkan mereka untuk secara otomatis melacak file dan mengelola pekerjaan dalam penyelesaian. Pengguna dapat dengan mudah dan aman berbagi, menemukan, dan menggunakan kembali 3ds Max (dan desain) aset dalam produksi skala besar atau lingkungan visualisasi.

2.6 VRML

VRML adalah singkatan dari Virtual Reality Modeling Language suatu bahasa pemrograman yang digunakan untuk membentuk objek 3D yang dapat dibaca oleh browser internet. VRML dipublikasikan pada Mei 1995 dan kemudian dilakukan standarisasi pada VRML97.

2.6.1 Konsep-Konsep VRML97

Bagian ini membahas tentang konsep-konsep dasar di dalam spesifikasi VRML97. Hal ini penting diketahui agar Anda mempunyai dasar pengetahuan yang kuat tentang VRML97 sebelum mengaplikasikannya ke dalam desain suatu lingkungan virtual 3D. Dari beberapa contoh program yang telah diberikan sebelumnya, tentunya Anda sudah mulai mengenal struktur file VRML97. Hal ini akan dibahas lagi dengan uraian lebih rinci. Pembahasan difokuskan pada topik-topik berikut:

1. Struktur file VRML97
2. Sintaks file UTF-8
3. Struktur scenegraph

Pemahaman tentang struktur file VRML97 akan mempercepat Anda dalam mempelajari pemrograman dan pemodelan lingkungan virtual menggunakan spesifikasi VRML97. Anda akan melihat bahwa pembahasan lebih bersifat teoritis daripada praktis. Karena itu, sebaiknya Anda membaca secara cepat dan menangkap pengertian-pengertian dasar yang diberikan. Sintaks UTF-8 dibahas dengan maksud agar Anda mempunyai gambaran umum tentang aturan-aturan dalam penulisan pemyataan-pemyataan di dalam VRML97. Karena bagian ini tergolong sulit, Anda tidak harus mengerti sekali membaca. Anda dapat membaca lagi bagian ini sewaktu-waktu ketka -memerlukan, misalnya untuk memahami program-program VRML di dalam buku atau ketika ingin mengembangkan program yang tidak ada dalam buku.

BAB 3

PERENCANAAN SISTEM

Dalam bab ini akan dijelaskan tentang uraian bahan yang akan digunakan dan cara kerja system dalam proyek akhir ini bersama teori-teori yang mendukung dalam pembuatan sistem *Augmented Reality* untuk animasi *games* menggunakan camera pada PC. Adapun teori yang di gunakan dalam mendukung proyek akhir ini akan di jelaskan di bawah ini.

3.1 Bahan dan alat

Dalam pembuatan Proyek Akhir ini bahan dan alat yang digunakan adalah sebagai berikut:

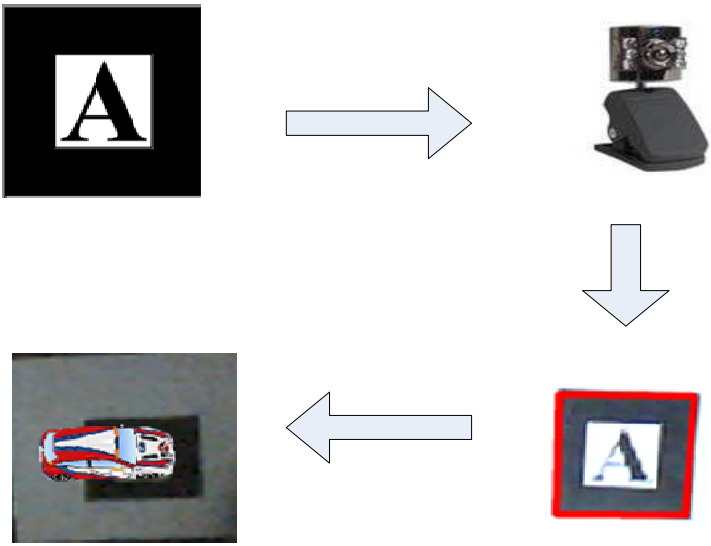
1. Marker
2. Camera
3. Software Adobe Flex
4. Software 3D max studio
5. Papervision 3D
6. FLARToolkit

3.2 Cara kerja

Dalam membuat aplikasi ini, diperlukan camera yang cukup bagus untuk proses pengenalan marker agar aplikasi tersebut dapat berjalan dengan baik. Serta sebuah desain program yang baik agar dapat mengenali marker dengan cepat sehingga animasi dapat mudah di tampilkan dalam PC.

3.3 Perancangan system

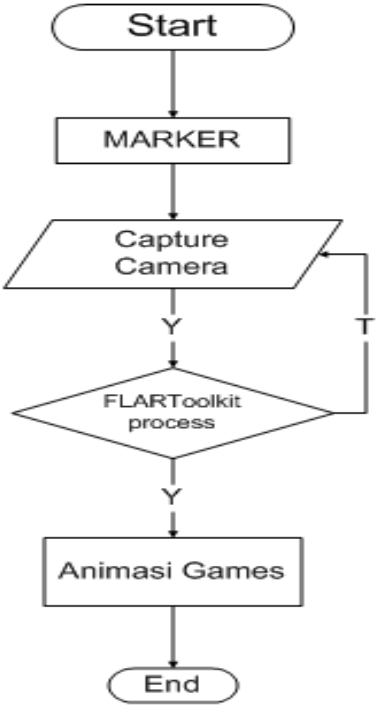
Pada tahap ini di uraikan tentang perancangan sistem yang akan dibuat untuk terwujudnya proyek akhir yang diinginkan, dimana pada dasarnya sistem ini dikerjakan secara software saja. seperti yang ditunjukkan blok diagram dibawah ini :



Gambar 3.1 Blok Diagram Sistem *Augmented Reality*

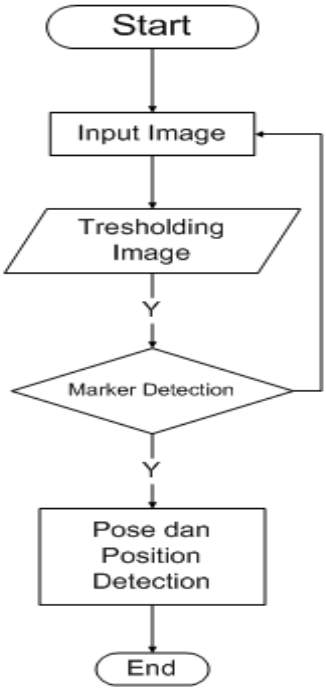
Pada dasarnya sebelum sistem ini berjalan maka sbelumnya harus dibuat markernya terlebih dahulu, untuk membuat markernya dapat dibuatnya dengan paint atau software editor gambar yang banyak di gemari saat ini seperti adobe photoshop, setelah pembuatan marker selesai maka langkah selanjutnya adalah inisialisasi marker dengan menggunakan marker generator, tapi sebelumnya terlebih dahulu harus mencetak marker yang telah dibuat untuk di tampilkan di depan kamera sebagai inisialisasi pattern yang nantinya akan di jadikan acuan dalam langkah pembuatan program.

Untuk lebih jelasnya proses alur dari sistem *augmented reality* ini maka dapat di lihat pada gambar 3.2 berikut ini:



Gambar 3.2. Flowchart Sistem

Marker yang sudah di print di tampilkan di depan kamera, lalu kamera akan membaca marker tersebut dan di olah di flartoolkit process. Bila marker yang di deteksi oleh kamera sesuai dengan marker yang telah menjadi acuan sebelumnya maka akan di tampilkan animasi *games* 3D namun bila marker yang di baca oleh camera tidak sama dengan marker yang menjadi acuan maka flartoolkit process akan kembali melakukan pembacaan input *image* dari camera.Flowchart Flartoolkit prosesnya dapat di lihat pada gambar 3.3 di bawah ini:



Gambar 3.3. Flowchart Flartoolkit Process

Image yang di baca oleh camera akan di lakukan tresholding *image*, ini berfungsi sebagai metode sederhana yang akan memilih nilai mean atau median dengan cara menghitung nilai pixel pada object gambar. Dimana jika nilai pixel pada object gambar lebih terang dibandingkan dengan background, maka nilai pixel pada object gambar juga harus lebih terang dari pada nilai rata-rata.

3.4 Waktu dan Tempat

Pembuatan Proyek akhir ini dimulai pada bulan Agustus hingga bulan januari 2011 dan dikerjakan di laboratorium POLITEKNIK ELEKTRONIKA NEGERI SURABAYA –ITS.

3.5 Metode Pengumpulan data dan cara analisis data

Pengumuplan data dari proyek akhir ini di peroleh dari internet dan cara analisis data dapat dilakukan dengan pengujian program tersebut dalam hal ini sistem dapat berjalan dengan baik dan sesuai dengan yang di harapkan, sehingga dapat di implementasikan.

Untuk dapat melakukan pengujian maka harus mengetahui algoritma dari sistem *Augmented Reality*. Algoritma dari sistem tersebut adalah sebagai berikut:

1 Print marker

2 Tampilkan marker di depan kamera

3 Conrtrol animasi tersebut dengan keyboard

Setelah semua persyaratan di atas di penuhi maka selanjutnya adalah melakukan pengujian dengan menampilkan marker di depan camera, data yang di ambil dalam pengujian ini adalah berupa keberhasilan program untuk dapat melakukan *images prosesing*dari marker yang di baca oleh camera hingga dapat menapilkan Animasi *Games* 3D yang bergerak. Bila marker yang telah di deklarasikan dalam program di tampilkan di depan camera dan dapat menampilkan animasi *Games* 3D maka dapat di katakan bahwa proses pengenalan citra ini telah berhasil, namun bila marker ini tidak dapat melakukan *images prosessing* maka pengolahan citra pada sistem *augmented reality* ini masih belum benar.

Halaman ini sengaja di kosongkan

BAB 4

PEMBUATAN SISTEM, HASIL serta PEMBAHASAN

4.1 Pembuatan Sistem

Dalam membuat sebuah aplikasi *augmented reality* .maka terlebih dahulu harus mendownload semua library dan juga software yang digunakan untuk dapat membuat programnya, karena program tidak akan bisa di jalankan bila librarynya tidak ada, dan akan muncul banyak error dalam setiap baris program.

4.1.1 Pembuatan marker

Sebelum membuat program maka harus dibuat markernya terlebih dahulu. Dalam pembuatan marker ini dapat menggunakan program paint atau software yang sangat digemari saat ini seperti adobe Photoshop. Dalam pembuatan markernya maka dapat anda lihat contoh di bawah ini.

Gambar 4.1 Contoh Marker

Kemudian setelah membuat markernya maka dapat langsung di cetak ke mesin printer.

Setelah selesai mencetak markernya maka selanjutnya dilakukan proses inisialisasi marker tersebut untuk disimpan sebagai pattern. Pattern ini berfungsi sebagai acuan dalam pembacaan marker. Untuk membuat pattern ini maka dapat digunakan tool ARToolkitgenerator untuk generating markernya menjadi pattern. Bila pada computer yang di pakai belum terinstall maka dapat di lakukan penginstallan terlebih dahulu.

Gambar 4.2 di bawah ini adalah contoh gambar hasil generating marker menggunakan ARToolkit marker Generator:

Gambar 4.2 Generate Marker

Dan berikut adalah hasil code dari pattern yang telah di generate menggunakan ARToolkit marker generator

```
199 207 211 204 190 168 160 162 175 204 220 211 203 201 200 125
219 219 216 184 125  98  99 103 105 124 183 214 213 212 207 128
217 217 146  77  73  64  67  70  70  70  81 160 208 210 202 121
215 164  82  78  75  60  60  61  66  75  75  90 170 204 201 117
189  98  85 105 116 149 162 156 134  92  71  87 107 203 198 122
150  89 178 229 227 220 219 221 224 230 222 165 111 169 198 121
128 188 222 209 210 213 213 215 215 213 216 222 176 139 202 124
129 217 211 212 215 213 214 215 215 216 217 208 218 145 197 124
151 214 211 214 214 208 212 214 212 215 217 213 216 159 195 121
130 204 208 211 210 168 209 212 210 213 217 214 213 152 199 120
121 180 193 191 166 137 209 209 210 215 217 215 193 149 206 121
125  63  60  59  53 106 207 208 212 214 213 204 135 169 202 126
147  81  67  64  67 107 206 209 210 210 204 144  92 197 197 123
185  95  81  76  63 112 210 210 206 180 132  75  93 174 195 118
195 211 213 210 194 152 203 208 204 184 187 201 217 210 193 115
192 195 204 214 205 188 202 216 210 210 211 209 213 216 200 115
190 196 201 202 184 155 140 138 158 192 212 200 189 190 192 122
223 222 229 203 128  95  87  85  95 122 191 219 217 218 222 132
223 228 160  83  68  68  67  66  67  65  81 168 219 218 222 132
225 180  79  74  68  58  58  58  62  68  69  85 179 216 223 127
208 108  68  94 110 142 152 147 128  89  61  67  97 221 220 128
171  81 158 224 225 221 220 222 223 228 216 150  85 181 224 127
134 175 219 220 218 218 219 218 219 219 218 224 154 130 233 128
119 216 218 218 218 217 218 218 219 221 220 221 218 119 228 132
130 224 217 218 216 210 214 218 219 221 220 220 227 136 221 133
101 222 216 216 219 167 213 217 219 220 220 219 225 136 225 132
 93 203 209 204 189 129 218 216 217 219 220 222 207 134 233 130
103  63  69  69  66  86 223 215 217 218 217 219 147 153 225 130
138  62  64  64  65  81 222 213 216 219 214 172  88 188 217 127
189  68  68  68  52  80 223 214 214 191 136  88  73 163 216 124
208 191 194 195 180 119 217 214 213 190 178 193 195 195 214 120
208 208 215 222 215 169 220 227 220 218 216 219 215 214 221 122
181 193 199 198 190 164 138 137 163 196 212 200 190 192 190 116
203 210 220 196 130 102  84  81  98 123 183 211 211 214 216 125
211 225 166  83  68  69  64  61  63  61  72 156 211 216 219 128
214 178  86  72  64  57  58  55  55  59  60  74 169 211 218 123
198 106  71  87 103 137 151 145 115  75  53  60  91 209 216 123
163  74 155 215 213 208 211 212 208 214 211 146  82 166 219 124
128 168 212 214 215 215 214 214 213 212 212 216 149 123 223 127
113 209 215 215 213 213 212 215 216 213 215 215 209 124 212 129
119 217 214 214 212 207 211 215 216 213 214 216 215 149 205 125
 88 211 207 211 213 163 211 213 214 214 214 213 213 149 209 124
 83 190 196 196 183 121 213 211 214 214 215 215 197 143 219 123
 98  56  58  59  63  78 212 212 216 213 214 214 142 148 216 123
136  60  59  57  63  69 209 210 214 213 210 167  91 172 210 120
186  67  65  60  48  64 210 208 209 187 135  86  78 151 211 121
```

```
203 192 195 188 173 102 206 206 205 187 178 190 194 189 208 120
204 204 210 214 205 151 212 221 216 216 213 214 209 209 219 122

125 128 121 117 122 121 124 124 121 120 121 126 123 118 115 115
200 207 202 201 198 198 202 197 195 199 206 202 197 195 193 200
201 212 210 204 203 169 139 145 159 152 149 169 197 174 210 216
203 213 208 170 107 111 176 218 216 213 193 135  92  93 217 213
211 214 160  90  87 165 222 208 213 214 215 204 144  75 201 209
220 183  81  75  71 222 216 217 217 217 217 213 204 132 187 211
204 124  70  75  92 230 213 216 215 213 215 214 210 180 184 210
175 105  70  66 134 224 215 215 212 210 210 212 210 206 204 210
162 103  70  61 156 221 215 215 214 212 209 208 209 210 208 216
160  99  67  60 162 219 213 214 212 209 209 207 206 210 203 202
168  98  64  60 149 220 213 213 208 168 137 106 107 112 152 188
190 125  73  75 116 227 210 215 214 210 166  53  67  63 194 205
204 184  77  78 105 229 209 212 214 211 191  59  64  76 210 214
211 216 146  82  85 178 222 211 211 208 193  60  67  81 213 204
207 219 217 164  98  89 188 217 214 204 180  63  81  95 211 195
199 219 217 215 189 150 128 129 151 130 121 125 147 185 195 192
122 132 132 127 128 127 128 132 133 132 130 130 127 124 120 122
192 222 222 223 220 224 233 228 221 225 233 225 217 216 214 221
190 218 218 216 221 181 130 119 136 136 134 153 188 163 195 214
189 217 219 179  97  85 154 218 227 225 207 147  88  73 195 215
200 219 168  85  67 150 224 221 220 219 222 219 172  88 193 219
212 191  81  69  61 216 218 220 220 220 220 217 214 136 178 216
192 122  65  68  89 228 219 221 221 220 219 218 219 191 190 218
158  95  67  62 128 223 219 219 219 219 217 217 216 214 213 220
138  85  66  58 147 222 218 218 218 217 216 215 213 214 214 227
140  87  67  58 152 220 219 218 214 213 218 223 222 223 217 220
155  95  68  58 142 221 218 217 210 167 129  86  81  80 119 169
184 128  68  68 110 225 218 218 216 219 189  66  65  52 180 215
202 203  83  74  94 224 220 218 218 216 204  69  64  68 195 222
201 229 160  79  68 158 219 218 217 216 209  69  64  68 194 215
196 222 228 180 108  81 175 216 224 222 203  63  62  68 191 208
190 223 223 225 208 171 134 119 130 101  93 103 138 189 208 208
116 125 128 123 123 124 127 129 125 124 123 123 120 121 120 122
190 216 219 218 216 219 223 212 205 209 219 216 210 211 208 219
192 214 216 211 209 166 123 124 149 149 143 148 172 151 189 209
190 211 211 169  91  82 149 209 215 213 197 142  91  78 194 209
200 211 156  74  60 146 216 215 216 213 215 214 167  86 190 214
212 183  72  60  53 211 212 215 214 214 215 214 210 135 178 213
196 123  61  59  75 214 212 213 213 214 214 213 213 187 187 216
163  98  63  55 115 208 213 216 216 214 214 216 214 209 205 216
137  81  61  55 145 212 214 215 215 213 211 212 210 208 206 221
138  84  64  58 151 211 214 212 211 211 213 212 209 210 206 212
164 102  69  57 137 208 215 213 207 163 121  78  69  64 102 151
190 130  68  64 103 213 215 213 212 213 183  63  63  48 173 205
198 196  83  72  87 215 214 215 214 211 196  59  57  60 188 214
199 220 166  86  71 155 212 215 214 207 196  58  59  65 195 210
193 210 225 178 106  74 168 209 217 211 190  56  60  67 192 204
181 203 211 214 198 163 128 113 119  88  83  98 136 186 203 204

115 200 216 213 209 211 210 210 216 202 188 205 214 204 195 192
115 193 210 217 201 187 184 204 208 203 152 194 210 213 211 195
118 195 174  93  75 132 180 206 210 210 112  63  76  81  95 185
123 197 197  92 144 204 210 210 209 206 107  67  64  67  81 147
126 202 169 135 204 213 214 212 208 207 106  53  59  60  63 125
121 206 149 193 215 217 215 210 209 209 137 166 191 193 180 121
120 199 152 213 214 217 213 210 212 209 168 210 211 208 204 130
121 195 159 216 213 217 215 212 214 212 208 214 214 211 214 151
124 197 145 218 208 217 216 215 215 214 213 215 212 211 217 129
124 202 139 176 222 216 213 215 215 213 213 210 209 222 188 128
121 198 169 111 165 222 230 224 221 219 220 227 229 178  89 150
122 198 203 107  87  71  92 134 156 162 149 116 105  85  98 189
117 201 204 170  90  75  75  66  61  60  60  75  78  82 164 215
121 202 210 208 160  81  70  70  70  67  64  73  77 146 217 217
128 207 212 213 214 183 124 105 103  99  98 125 184 216 219 219
125 200 201 203 211 220 204 175 162 160 168 190 204 211 207 199
```

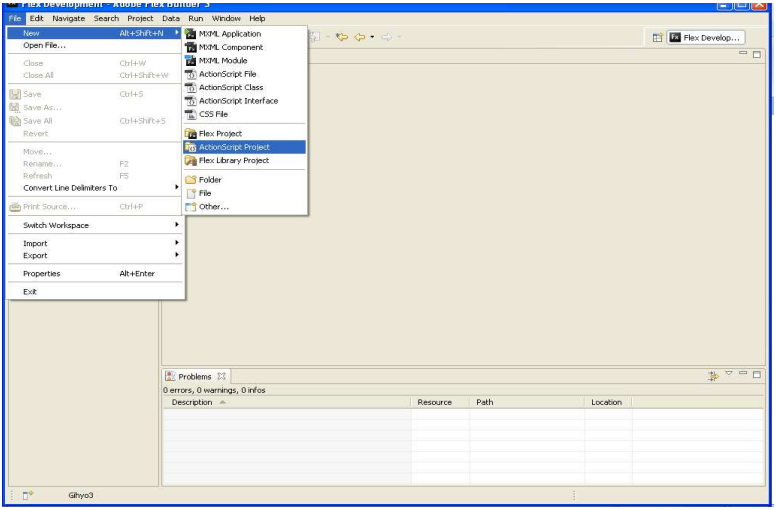
```
122 221 214 215 219 216 218 220 227 220 169 215 222 215 208 208
120 214 195 195 193 178 190 213 214 217 119 180 195 194 191 208
124 216 163  73  88 136 191 214 214 223  80  52  68  68  68 189
127 217 188  88 172 214 219 216 213 222  81  65  64  64  62 138
130 225 153 147 219 217 218 217 215 223  86  66  69  69  63 103
130 233 134 207 222 220 219 217 216 218 129 189 204 209 203  93
132 225 136 225 219 220 220 219 217 213 167 219 216 216 222 101
133 221 136 227 220 220 221 219 218 214 210 216 218 217 224 130
132 228 119 218 221 220 221 219 218 218 217 218 218 218 216 119
128 233 130 154 224 218 219 219 218 219 218 218 220 219 175 134
127 224 181  85 150 216 228 223 222 220 221 225 224 158  81 171
128 220 221  97  67  61  89 128 147 152 142 110  94  68 108 208
127 223 216 179  85  69  68  62  58  58  58  68  74  79 180 225
132 222 218 219 168  81  65  67  66  67  68  68  83 160 228 223
132 222 218 217 219 191 122  95  85  87  95 128 203 229 222 223
122 192 190 189 200 212 192 158 138 140 155 184 202 201 196 190
122 219 209 209 214 213 216 216 221 212 151 205 214 210 204 204
120 208 189 194 190 178 187 205 206 206 102 173 188 195 192 203
121 211 151  78  86 135 187 209 208 210  64  48  60  65  67 186
120 210 172  91 167 210 213 214 210 209  69  63  57  59  60 136
123 216 148 142 214 214 213 216 212 212  78  63  59  58  56  98
123 219 143 197 215 215 214 214 211 213 121 183 196 196 190  83
124 209 149 213 213 214 214 214 213 211 163 213 211 207 211  88
125 205 149 215 216 214 213 216 215 211 207 212 214 214 217 119
129 212 124 209 215 215 213 216 215 212 213 213 215 215 209 113
127 223 123 149 216 212 212 213 214 214 215 215 214 212 168 128
124 219 166  82 146 211 214 208 212 211 208 213 215 155  74 163
123 216 209  91  60  53  75 115 145 151 137 103  87  71 106 198
123 218 211 169  74  60  59  55  55  58  57  64  72  86 178 214
128 219 216 211 156  72  61  63  61  64  69  68  83 166 225 211
125 216 214 211 211 183 123  98  81  84 102 130 196 220 210 203
116 190 192 190 200 212 196 163 137 138 164 190 198 199 193 181

192 195 185 147 125 121 130 151 129 128 150 189 215 217 219 199
195 211  95  81  63 180 204 214 217 188  89  98 164 217 219 207
204 213  81  67  60 193 208 211 211 222 178  85  82 146 216 211
214 210  76  64  59 191 211 214 212 209 229 105  78  77 184 204
205 194  63  67  53 166 210 214 215 210 227 116  75  73 125 190
188 152 112 107 106 137 168 208 213 213 220 149  60  64  98 168
202 203 210 206 207 209 209 212 214 213 219 162  60  67  99 160
216 208 210 209 208 209 212 214 215 215 221 156  61  70 103 162
210 204 206 210 212 210 210 212 215 215 224 134  66  70 105 175
210 184 180 210 214 215 213 215 216 213 230  92  75  70 124 204
211 187 132 204 213 217 217 217 217 216 222  71  75  81 183 220
209 201  75 144 204 215 214 213 208 222 165  87  90 160 214 211
213 217  93  92 135 193 213 216 218 176 111 107 170 208 213 203
216 210 174 197 169 149 152 159 145 139 169 203 204 210 212 201
200 193 195 197 202 206 199 195 197 202 198 198 201 202 207 200
115 115 118 123 126 121 120 121 124 124 121 122 117 121 128 125
208 208 189 138 103  93 101 130 119 134 171 208 225 223 223 190
208 191  68  62  63 203 222 224 216 175  81 108 180 228 222 196
215 194  68  64  69 209 216 217 218 219 158  68  79 160 229 201
222 195  68  64  69 204 216 218 218 220 224  94  74  83 203 202
215 180  52  65  66 189 219 216 218 218 225 110  68  68 128 184
169 119  80  81  86 129 167 210 217 218 221 142  58  68  95 155
220 217 223 222 223 218 213 214 218 219 220 152  58  67  87 140
227 214 214 213 215 216 217 218 218 218 222 147  58  66  85 138
220 213 214 216 217 217 219 219 219 219 223 128  62  67  95 158
218 190 191 219 218 219 220 221 221 219 228  89  68  65 122 192
216 178 136 214 217 220 220 220 220 218 216  61  69  81 191 212
219 193  88 172 219 222 219 220 221 224 150  67  85 168 219 200
215 195  73  88 147 207 225 227 218 154  85  97 179 219 217 189
214 195 163 188 153 134 136 136 119 130 181 221 216 218 218 190
221 214 216 217 225 233 225 221 228 233 224 220 223 222 222 192
122 120 124 127 130 130 132 133 132 128 127 128 127 132 132 122
204 203 186 136  98  83  88 119 113 128 163 198 214 211 203 181
204 192  67  60  56 190 211 217 209 168  74 106 178 225 210 193
210 195  65  59  58 196 207 214 215 212 155  71  86 166 220 199
```

```
 214 188  60  57  59 196 211 214 215 214 215  87  72  83 196 198
 205 173  48  63  63 183 213 212 213 215 213 103  64  68 130 190
 151 102  64  69  78 121 163 207 213 215 208 137  57  69 102 164
 212 206 210 209 212 213 211 211 212 214 211 151  58  64  84 138
 221 206 208 210 212 211 213 215 215 214 212 145  55  61  81 137
 216 205 209 214 216 214 214 216 216 213 208 115  55  63  98 163
 216 187 187 213 213 214 214 213 213 212 214  75  59  61 123 196
 213 178 135 210 214 215 214 214 215 212 211  53  60  72 183 212
 214 190  86 167 214 215 213 216 215 216 146  60  74 156 211 200
 209 194  78  91 142 197 213 215 209 149  82  91 169 211 211 190
 209 189 151 172 148 143 149 149 124 123 166 209 211 216 214 192
 219 208 211 210 216 219 209 205 212 223 219 216 218 219 216 190
122  0 121 120 123 123 124 125 129 127 124 123 123 128 125 116
```

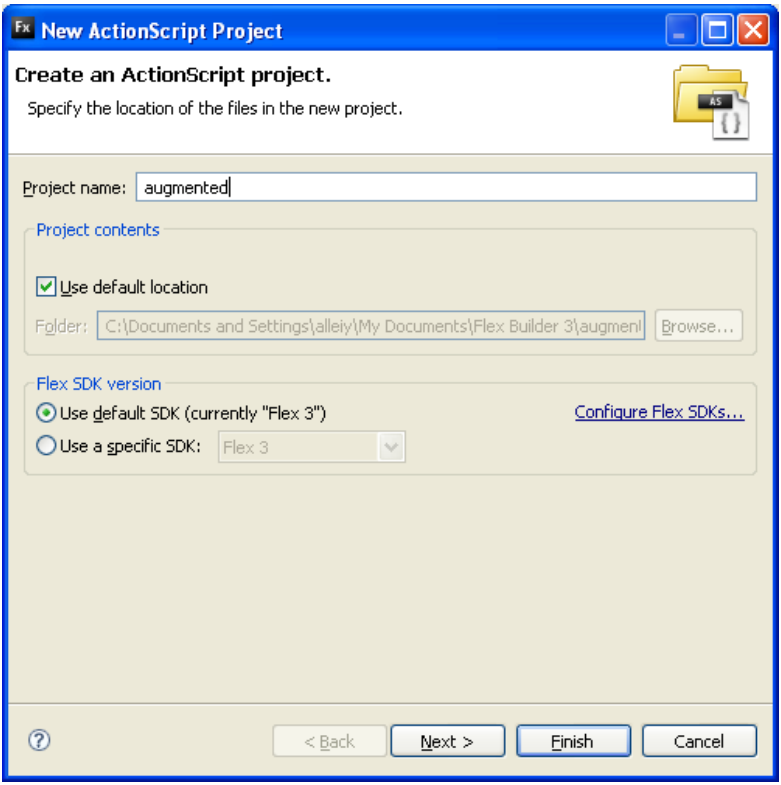
4.2 Menjalankan Adobe Flex Builder

Untuk membuat project maka diperlukan langkah- langkah sebagai berikut:



Gambar 4.3 Tampilan Running Adobe Flex

Jalankan adobe flex builder dan Pada kolom file pilih *new* lalu pilih *actionscript project*, setelah muncul window *create actionscript project* lalu isi project name dengan nama project yang akan dibuat dan dalam hal ini akan di buat project augmented, setelah selesai klik tombol finish. seperti gambar 4.4 pada tampilan di bawah ini



Gambar 4.4 Tampilan Create Action Sript project

Jika sudah selesai maka dapat dilanjutkannya dengan membuat program. Namun sebelum pembuatan program terlebih dahulu harus dimasukkan library-library yang di butuhkan dalam pembuatan program ini karena bila library-library tersebut tidak di masukkan dalam program maka saat program di running akan mengalami banyak error.

4.3 Pembuatan Program

4.3.1 Llibrary Flartoolkit

4.3.1.1 Koneksi dengan Camera

Untuk pembacaan pada camera, dibutuhkan koneksi antara program dengan camera.Berikut ini adalah pembuatan koneksi pada camera.

```
this.webcam = Camera.getCamera();
//membuat project camera
if (!this.webcam) {
      throw new Error('No webcam!!!!');
                  }
this.webcam.setMode(this.Width,this.Height,30);
//pengaturan setting camera dengan file video swf agar sama
this.video = new Video(this.Width,this.Height);
this.video.attachCamera(this.webcam);
//menampilkan kamera ke video
this.capture = new Bitmap(new
BitmapData(this.width, this.Height, false, 0),
PixelSnapping.AUTO, true);
//menampilkan hasil video project
```

4.3.1.2 Pembuatan Flar

Untuk pembuatan flar ini berfungsi untuk deklarasi file marker dan camera, berikut ini adalah script pembuatan function flash AR.

```
ar_params = new FLARParam();
this.param = new FLARParam();
this.loader.load(new
URLRequest(this.CodeFile));
//membuat parameter untuk project
this.param.changeScreenSize(this.Width,
this.Height);
//membuat marker pada project
this.param.loadARParam(this.loader.data);
//memasukkan paramaters pada webcam
this.loader.dataFormat =
URLLoaderDataFormat.TEXT;
//mengisi filemarker dengan data
```

```
this.loader.addEventListener(Event.COMPLETE,
this._onLoadCode);
//loading file
```

4.3.1.3 Pembuatan Bitmap data

Untuk membuat Bitmapdata harus di sesuaikan dengan fungsi sebelumnya. Berikut adalah salah satu potongan *script* untuk pembuatan function bitmap data:

```
ar_bmp = new BitmapData(640, 480);
this.raster = new
FLARRgbRaster_BitmapData(this.capture.bitmapDat
a);
//membuat Bitmap canvas baru untuk di samakan dg ukran project
this.detector = new
FLARSingleMarkerDetector(this.param, this.code,
this.CodeWidth);
// untuk deteksi marker
```

4.3.1.4 Pembuatan Papervision

Papervision adalah untuk menampilkan animasi 3D dalam video yang telah di baca oleh camera yang nantinya akan di tampilkan pada video. Berikut ini adalah salah satu potongan *script* untuk pembuatan function papervision 3D *:*

```
this.base = this.addChild(new Sprite()) as
Sprite;
this.capture.width = 640;
this.capture.height = 480;
//menentukan ukuran video
this.base.addChild(this.capture);

this.viewport = this.base.addChild(new
Viewport3D(320, 240)) as Viewport3D;
this.viewport.scaleX = 640 / 320;
this.viewport.scaleY = 480 / 240;
this.viewport.x = -4; // 4pix ???
//membuat viewport
```

```
this.camera3d = new FLARCamera3D(this.param);
//deklarasi 3D camera
this.scene = new Scene3D();
//membuat scene 3d
this.baseNode = this.scene.addChild(new
FLARBaseNode()) as FLARBaseNode;
//membuat basenode
this.renderer = new
LazyRenderEngine(this.scene, this.camera3d,
this.viewport);
//membuat object rendering

//penambahan file animasi pada Papervision object
var car:DAE = new DAE();
      car.load('mobil.dae');
      car.scaleX = 10;
      car.scaleY = 10;
      car.scaleZ = 10;
      car.rotationX = 90;
      car.rotationZ = -90;
      this.baseNode.addChild(car);
      this.rootNode = car;
```

4.3.1.5 Pembuatan control

Untuk membuat mobil dapat berjalan maka perlu di buat program untuk mengcontrol mobil tersebut. Berikut ini adalah programnya.

```
private function
keyDownHandler( event :KeyboardEvent ):void
            {
                  switch( event.keyCode )
                  {
                        case "W".charCodeAt():
                        case Keyboard.UP:
                              keyForward =
true;
                              keyReverse =
false;
                              break;
```

```
                        case "S".charCodeAt():
                        case Keyboard.DOWN:
                              keyReverse =
true;
                              keyForward =
false;
                              break;

                        case "A".charCodeAt():
                        case Keyboard.LEFT:
                              keyLeft = true;
                              keyRight = false;
                              break;

                        case "D".charCodeAt():
                        case Keyboard.RIGHT:
                              keyRight = true;
                              keyLeft = false;
                              break;
                              }
            }
private function driveCar():void
            {
                  // Speed
                  if( keyForward )
                  {
                        topSpeed = 2;
                  }
                  else if( keyReverse )
                  {
                        topSpeed = -1;
                  }
                  else if( keyUp )
                  {
                        topUp = +1;
                  }
                  else
                  {
                        topSpeed = 0;
                  }
```

```
                    speed -= ( speed - topSpeed )
/ 20;

                    // Steer
                    if( keyRight )
                    {
                          if( topSteer < 45 )
                          {
                                topSteer += 5;
                          }
                    }
                    else if( keyLeft )
                    {
                          if( topSteer > -45 )
                          {
                                topSteer -= 5;
                          }
                    }
                    else
                    {
                          topSteer -= topSteer /
5;
                    }
              steer -= ( steer - topSteer ) / 2;
              }
```

4.3.2 Library ARToolkit

Bila menggunakan tool ARToolkit maka pertama-pertama adalah membuat animasi menggunakan pemrograman VRML, atau dapat menggunakan 3D max lalu di convert ke VRML, setelah animasi telah di buat maka selanjutnya di teruskan dengan pembuatan script object_data_VRML seperti program berikut ini:

```
#the number of patterns to be recognized
3

#pattern 1
VRML  Wrl/toy.dat
Data/patt.a
```

```
80.0
0.0 0.0

#pattern 2
VRML  Wrl/bud_B.dat
Data/patt.b
80.0
0.0 0.0

#pattern 3
VRML  Wrl/kupu2.dat
Data/patt.f
80.0
0.0 0.0
```

Pada tool ARToolkit ini dapat dengan mudah membuat animasi dengan multi marker karena tool ini sudah sangat lengkap sehingga hanya dengan menambahkan program inisialisasi pattern dan animasi yang akan di jalankan saja. Seperti pada program object_data_vrml diatas maka dapat di mengerti bahwa pada setiap pattern di inisialisasikan animasi beserta marker yang akan di gunakan agar saat marker di tampilkan di depan camera maka animasi yang telah di deklarasikan akan di tampilkan di layar monitor.

Setelah membuat object_data_vrml maka selanjutnya dibuat Script pemanggil animasi seperti program berikut ini:

```
kupu2.wrl
0.0 0.0 0.0             # Translation
0.0 00.0 0.0 0.0        # Rotation
10.0 10.0 10.0          # Scale

toy.wrl
0.0 0.0 0.0             # Translation
0.0 0.0 0.0 0.0         # Rotation
10.0 10.0 10.0          # Scale

bud_B.wrl
0.0 0.0 0.0             # Translation
0.0 0.0 0.0 0.0         # Rotation
10.0  10.0 10.0         # Scale
```

4.4 File-file penting

Untuk membuat program ini dapat berjalan , maka dibutuhkan file – file penting yang di sebut juga sebagai library, file-file library tersebut terdapat pada folder-folder berikut :

- ascollada

Merupakan folder yang berisi library untuk pengaturan animasi 3d dari file COLLADA.

- libspark

Dalam folder ini berisi beberapa library untuk flartoolkit seperti deteksi marker, transformation matrix,,camera3D,dll .

- papervision

Dalam folder ini berisi beberapa library untuk animasi seperti 3D object, render engine, bitmap material, view port ,point light 3d ,dll

4.5 Hasil Pengujian Program

Pada bagian pengujian ini di jelaskan tentang pengujian program, analisa serta hasil yang di dapatkan dalam pembuatan tugas akhir ini. Untuk melakukan pengujian maka harus dilakukan running program sebagai output dari data yang nantinya akan di analisis.

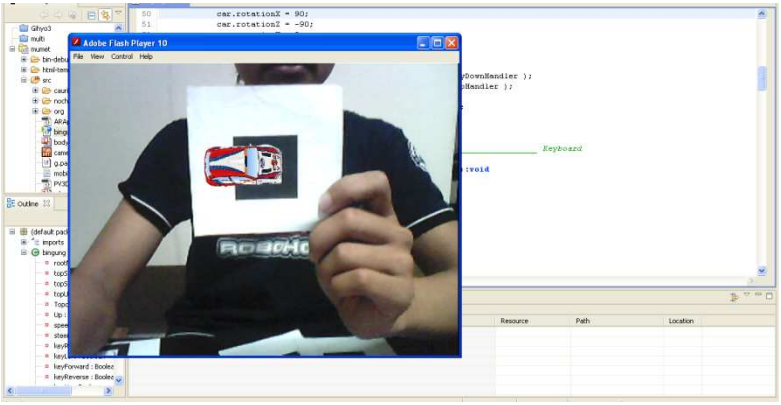
Sebelum melakukan pengujian, maka harus di ketahui terlebih dahulu bahwa setiap marker telah di definisikan animasinya. Berikut adalah tabel animasi pada setiap marker tersebut.

Tabel 4.1 *Tabel animasi setiap marker di Flartoolkit dan ARToolkit*

Marker	Flartoolkit	ARToolkit
Marker A	Mobil	Boneka
Marker B	-	Lebah
Marker F	-	Kupu

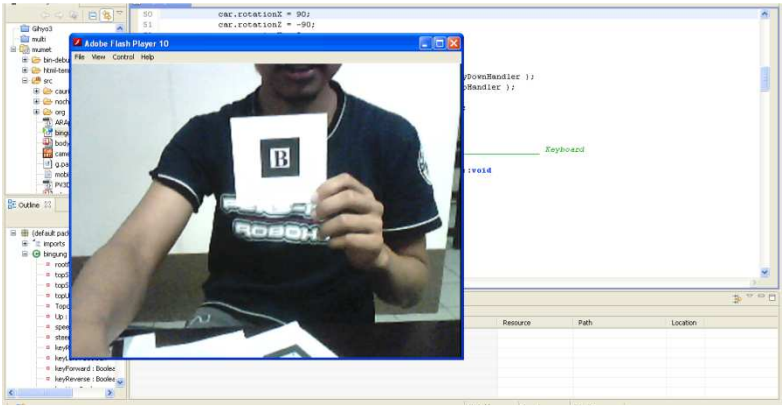
4.5.1 Pengujian FlarToolkit

Berikut ini adalah hasil dari pengujian sistem *augmented reality* untuk animasi *games* 3d ,yang akan dijalankan dengan menggunakan sistem operasi Windows dan flash palyer. Gambar 4.5 berikut adalah tampilan dari hasil tugas akhir ini.



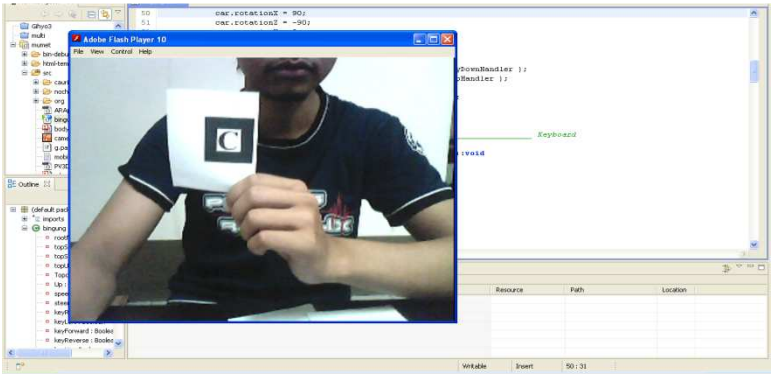
Gambar 4.5 Tampilan running program saat berhasil

Saat marker A di tampilkan di camera maka program akan mengenali marker tersebut dan akan menampilkan animasi mobil seperti gambar 4.5. dan saat marker yang di tampilkan di camera bukan marker A maka animasi tidak akan di tampilkan / di generate. Gambar 4.6 berikut adalah contoh saat menggunakan marker B.



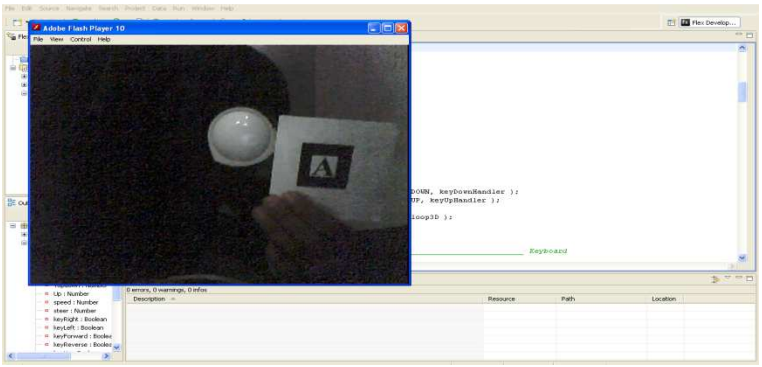
Gambar 4.6 Tampilan running program saat di marker B tidak di kenali

Begitu juga saat merker yang lainnya di tampilkan di camera seperti gambar 4.7 di bawah ini maka animasi juga tidak akan di tampilkan.



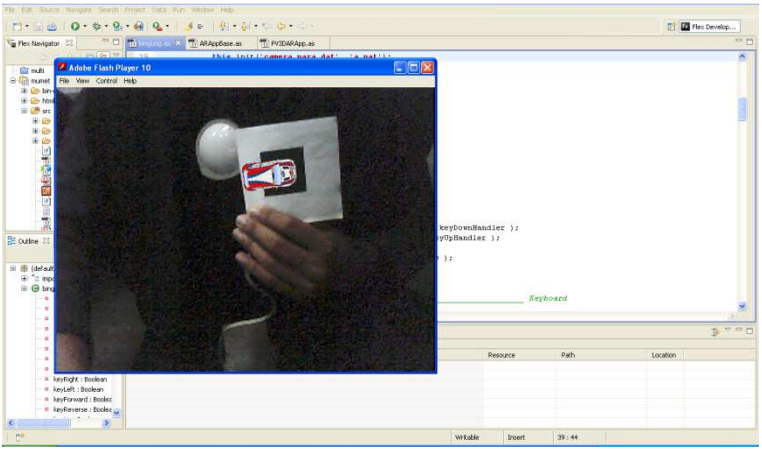
Gambar 4.7 Tampilan running program saat marker C tidak di kenali

Untuk dapat menampilkan animasi 3D ini harus menampilkan marker di depan camera terlebih dahulu, marker yang di tampilkan adalah marker yang telah di deklarasikan pada program ini. Harus juga di ingat bahwa dalam menampilkan marker di depan camera intensitas cahaya harus sesuai dengan kapasitas kamera karena saat terlalu cerah atau terlalu petang maka program tidak dapat menjalankan animasi dengan baik **.** gambar 4.8 di bawah ini adalah contoh tampilan saat intensitas cahaya terlalu gelap.



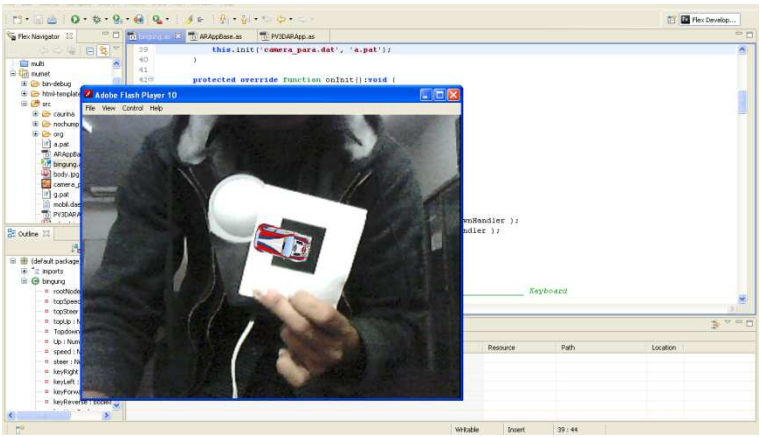
Gambar 4.8 Tampilan saat intensitas cahaya di bawah range camera bernilai 30 Cd.

Pada gambar 4.8 diatas menunjukkan bahwa saat intensitas cahaya terlalu gelap atau saat bernilai 30 Cd maka camera tidak dapat membaca marker tersebut dan animasi tidak dapat di tampilkan..Hal ini di karenakan camera yang di gunakan pada tugas akhir ini hanya memiliki sensivitas cahaya yang tidak begitu bagus sehingga saat intensitas cahaya berada di bawah nilai range dari camera maka camera tidak dapat membaca marker tersebut. Namun saat intensitas cahaya berada di atas 30 Cd maka camera dapat membaca marker tersebut dan dapat melakukan proses flartoolkit hingga dapat menampilkan animasi games 3D. Berikut adalah tampilan saat intensitas berada di atas range 30 Cd:



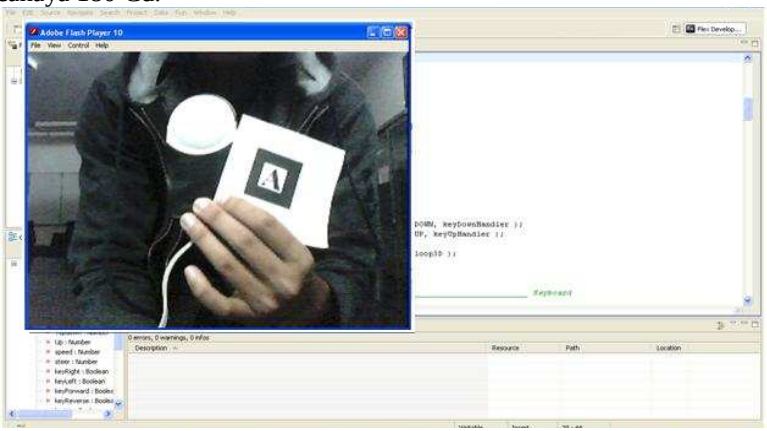
Gambar 4.9 Tampilan running program saat intensitas cahaya bernilai 50 Cd

Gambar 4.9 adalah tampilan saat intensitas cahaya bernilai 50 Cd. terlihat bahwa camera dapat membaca merker dan dapat melakukan proses Flartoolkit hingga dapat menampilkan animasi 3D. Begitupun saat intensitas cahaya berada pada nilai 140 Cd camera dapat melakukan proses Flartoolkit dan menampilkan animasi 3D. Berikut adalah tampilan gambar saat nilai intensitas cahaya bernilai 140 Cd:



Gambar 4.10 Tampilan running program saat intensitas cahaya bernilai 140 Cd

Namun saat intensitas cahaya bernilai di atas 180 Cd maka camera tidak dapat membaca marker tersebut. Gambar 4.11 di bawah ini adalah tampilan saat camera tidak dapat membaca marker dengan intensitas cahaya 180 Cd.

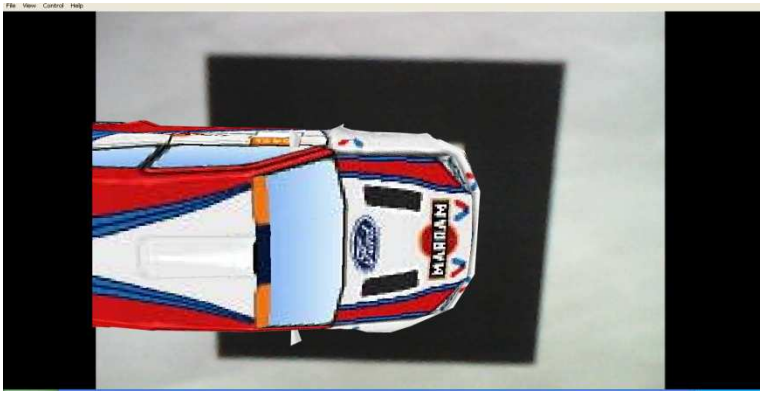


Gambar 4.11 Tampilan running program saat intensitas cahaya bernilai 180 Cd

Dari hasil pengujian dengan beberapa gambar diatas maka dapat di ketahui bahwa dalam tugas akhir ini intensitas cahaya sangat berpengaruh penting dalam berjalannya program augmented reality ini.

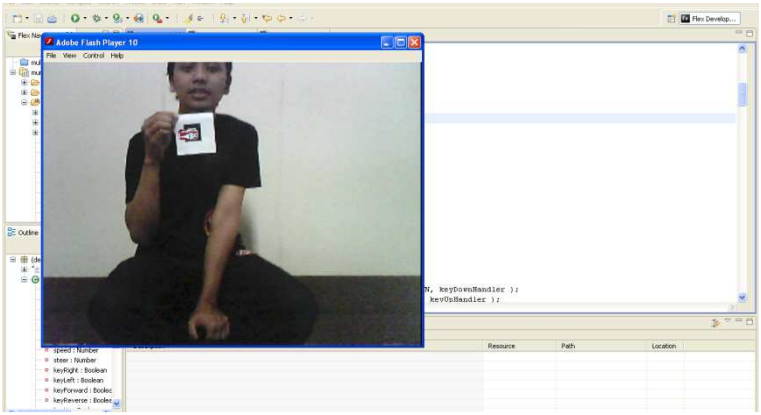
Dimana pada camera yang di gunakan dalam tugas akhir ini memiliki sensitivitas cahaya dengan range camera dimulai dari 50 Cd hingga 140 Cd. Dan untuk di luar range tersebut maka camera tidak dapat membaca merker dengan baik sehingga tidak dapat melakukan *Image Prosesing* dengan baik sehingga tidak dapat program tidak dapat mengenali marker yang di tampilkan di depan camera, sehingga proses flartoolkit tidak berjalan dan animasi 3D tidak dapat di tampilkan.

Selain intensitas cahaya yang dapat berpengaruh dengan proses berjalannya program dan pembacaan marker, jarak juga sangat berpengaruh dalam pembacaan marker. Jauh atau dekatnya marker dari camera akan berpengaruh terhadap berjalannya program ini karena dengan jauhnya marker dari camera maka camera akan sulit mengenali marker tersebut, sehingga proses pembacaan marker tidak dapat berjalan dengan baik. Berikut adalah tampilan hasil pengujian dengan jarak terdekat.



Gambar 4.12 Tampilan running program saat berada pada jarak terdekat.

Pada gambar 4.12 adalah gambar saat di lakukan pengujian dengan jarak terdekat yaitu dengan jarak 7 cm dari camera. Dimana pada jarak ini adalah batas minimum antara marker dengan camera. Untuk lebih dekat dari jarak ini maka animasi tidak akan di tampilkan karena sensitivitas camera tidak dapat membaca marker dengan jarak kurang dari 7 cm. dan jarak terjauh antara camera dengan marker adalah saat berjarak 1 meter dari camera. Berikut adalah tampilan dari running program saat berada pada jarak terjauh.



Gambar 4.13 Tampilan running program saat berada pada jarak terjauh.

Dari hasil pengujian di atas maka dapat di ketahui bahwa jarak juga berpengaruh pada system Augmented Reality ini, dimana jarak mempengaruhi sensitivitas camera dalam membaca marker. Semakin jauh marker dari camera maka proses *image prosesing* akan semakin sulit sehingga akan membuat program sulit mengenali marker meskipun marker telah di deklarasikan terlebih dahulu. Karena gambar yang di baca oleh camera akan hambar sehingga camera tidak dapat membaca marker tersebut dengan baik saat marker berada di tempat lebih dari 1 meter dari camera. Begitu juga saat marker berada pada jarak terdekat yaitu kurang dari 7 cm maka camera tidak akan dapat membaca marker secara penuh sehingga marker tidak dapat dikenali oleh program.

Dalam pembuatan tugas akhir ini juga di lakukan pengujian dengan menjalankan animasi tersebut.Dan untuk menjalankan animasi tersebut dapat dilakukan dengan menekan tombol ‘w’ untuk berjalan maju, tombol ‘s’ untuk berjalan kebelakang, tombol ‘a’ untuk belok ke kiri dan tombol ‘d’ untuk belok ke kanan. Dapat juga menekan tombol enter untuk *holding* gambar animasi.

4.5.2 Pengujian ARToolkit

Pada dasarnya pengujian menggunakan library ARToolkit ini sama dengan menggunakan library Flartoolkit hanya saja dengan library ARToolkit maka akan mempermudah dalam pembuatan program system augmented reality ini. dapat di lihat bahwa dengan

menggunakan program yang sederhana maka dapat di buat sebuah system augmented reality dengan menggunakan multimarker yang menarik sehingga dapat menampilkan beberapa animasi , hanya saja dengan menggunakan library ARToolkit ini kita tidak dapat memberikan inputan data berupa control dari keyboard atau dari marker yang lainnya karena pada library ARToolkit ini hanya digunakan utuk menampilkan animasi saja dan untuk pergerakan animasi di lakukan di dalam pembuatan animasi tersebut.

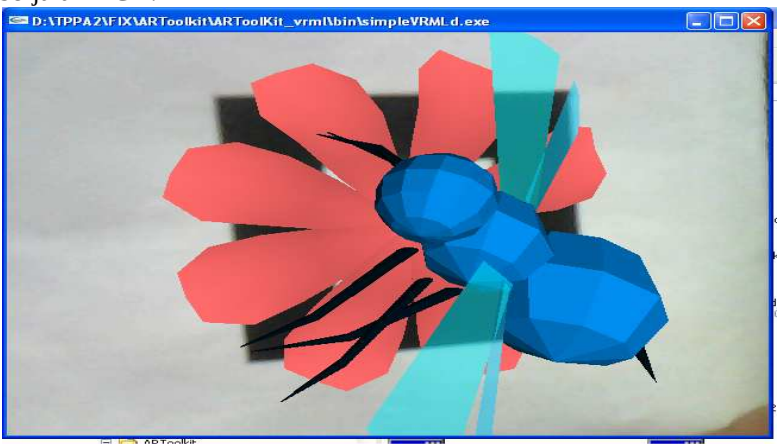
Setalah pembuatan program maka untuk menjalankan program tersebut maka perlu di jalankan program simpleVrmld.exe yang nantinya akan membaca script yang telah di buat sebelumnya.Seperti yang telah di tampilkan pada gambar 4.14 di bawah ini dapat kita lihat bahwa setiap animasi berjalan di atas marker-marker yang telah di deklarasikan terlebih dahulu. Animasi ini berjalan secara bersamaan ketika marker di tampilkan di depan camera. Berikut adalah tampilan pengujian menggunakan library ARToolkit.

Gambar 4.14 Tampilan running program menggunakan multimarker

Sama halnya saat menggunakan Flartoolkit, pada pengujian menggunakan ARToolkit ini juga di pengaruhi intensitas cahaya dan jarak. Hali ini dapat di lihat dari hasil pengujian di bawah ini. Gambar 4.15 di bawah ini adalah tampilan jarak terjauh antara marker dengan camera yaitu berjarak 50 Cm.

Gambar 4.15 Tampilan running Program menggunakan ARToolkit pada jarak 50 Cm

Dan tampilan di bawah ini adalah jarak terdekat antara marker dengan camera.dimana jarak terdekat dari marker dengan camera hanya berjarak 7 Cm.



Gambar 4.16 Tampilan jarak terdekat pada ARToolkit

Saat intensitas cahaya terlalu kecil atau telalu besar maka animasi tidak dapat di tampilkan karena sensitivitas camera hanya mampu membaca marker saat memiliki intensitas cahaya 50 Cd - 140 Cd. Namun untuk jarak yang dapat di toleransi di mulai dari range 7cm hingga 40 cm hal ini dikarenakan image prosesing dalam program

tersebut kurang baik sehingga saat marker berada agak jauh dari camera program sudah tidak dapat mengenali marker tersebut.

Pada pengujian ini juga dilakukan dengan mencoba coba menampilkan marker yang memiliki kemiripan, dan pada pengujian ini di dapatkan bahwa pada beberapa marker yang memiliki kemiripan, program tidak dapat mengenali marker yang asli / marker yang sebenarnya. Dan ada beberapa marker yang dapat di kenali oleh program tersebut dan dapat membedakan antara merker asli dengan marker pembanding. Berikut adalah hasil pengujian dari beberapa marker tersebut.

Tabel 4.2 *Tabel Pengujian dengan beberapa marker pembanding*

Marker		Hasil Pengujian
Marker Asli	Marker Pembanding	
A	A’	Terkadang error dan menampilkan animasi di atas marker pembanding
B	B’	Terkadang error dan menampilkan animasi di atas marker pembanding
C	G	Terkadang error dan menampilkan animasi di atas marker pembanding
E	F	Terkadang error dan menampilkan animasi di atas marker pembanding
S	5	Hanya menampilkan animasi pada marker Asli
I	1	Terkadang error dan menampilkan animasi di atas marker pembanding
あ	お	Hanya menampilkan animasi pada marker Asli
か	が	Terkadang error dan menampilkan animasi di atas marker pembanding
オ	ホ	Hanya menampilkan animasi pada marker Asli

Dari seluruh hasil pengujian dengan menggunakan ARToolkit dan Flartoolkit maka didapatkan bahwa dengan menggunakan Flartoolkit diperoleh jarak terjauh yaitu 1 Meter sedangkan dengan menggunakan ARToolkit hanya mendapatkan jarak terjauh yaitu 40 Cm seperti di tampilkan pada Tabel 4.3 di bawah ini

Tabel 4.3 *Tabel Pengujian menggunakan Flartoolkit dan ARToolkit*

Library	Jarak	
	Terdekat	Terjauh
Flartoolkit	7 Cm	1 M
ARTolkit	7 Cm	40 Cm

Dari tabel 4.3 di atas maka dapat di analisis bahwa dengan menggunakan Flartoolkit maka sensitivitas camera terhadap marker lebih baik, hal ini dapat di lihat dari jarak camera membaca marker yaitu sejauh 1 meter. Sedangkan saat menggunakan ARToolkit jarak camera dalam membaca marker hanya 40 Cm. hal ini di karenakan dengan menggunakan Flartoolkit image prosesingnya lebih baik baik di bandingkan dengan menggunakan ARToolkit.sehingga daya baca camera terhadap marker lebih detail saat menggunankan Flartoolkit di bandingkan dengan menggunakan ARTollkit. Dimana ukuran minimum dari marker yang dapat di baca oleh camera sebesar 50 x 53 pixel sesuai dari pengukuran yang telah dilakukan pada gambar 4.17 pengujian berikut.

Gambar 4.17 Tampilan ukuran pixel minimum

Sehingga saat marker yang dibaca oleh camera lebih kecil dari 50 x 53 pixel maka camera tidak dapat membaca marker tersebut, karena saat berada di bawah nilai pixel tersebut data marker yang di baca akan hambar dan sulit untuk di bandingkan dengan data marker yang telah di inisialisasi. Jadi untuk dapat menampilkan animasi dengan jarak terjauh juga dipengaruhi dengan ukuran marker. Seandainya marker di buat dengan ukuran yang sangat besar maka camera pun akan dapat mengenali marker tersebut meskipun berada pada jarak yang sangat jauh.

Halaman ini sengaja di kosongkan

BAB 5

PENUTUP

KESIMPULAN

Dari hasil pengujian dan analisa pada bab sebelumnya maka dapat diambil kesimpulan

1. Pembacaan marker oleh kamera sangat di pengaruhi oleh pencahayaan, intensitas cahaya untuk camera yang di pakai saat ini, hanya mampu membaca marker yang memiliki intensitas cahayanya berada pada range antara 50 Cd hingga 140 Cd. Untuk intensitas cahaya di luar range itu maka camera tidak akan dapat membaca marker dengan baik.
2. Selain intensitas cahaya, Jarak marker dengan camera juga sangat berpengaruh dalam proses berjalannya program ini dimana Marker yang telah di baca oleh camera ini nantinya akan di bandingkan dengan data marker yang telah menjadi acuannya.bila terlalu dekat atau terlalu jauh maka kamera tidak dapat membaca marker dengan baik sehingga program tidak dapat mengenali marker tersebut.
3. Animasi akan di generate / di tampilkan jika marker tersebut telah di definisikan terlebih dahulu atau telah di kenali oleh program, jika marker yang di gunakan bukan marker yang telah di definisikan pada program, maka animasi tidak akan di tampilkan.
4. Penggunaan Library juga akan berpengaruh dengan terhadap proses berjalannya program. Dimana saat menggunaka Flartoolkit jarak terjauh marker terhadap camera hingga 1 M, dan saat menggunakan ARTookit jarak terjauh hanya 40 Cm.

SARAN

Saran-saran yang bisa disampaikan adalah sebagai berikut:

1. Desain Animasi dibuat lebih menarik lagi.
2. Mungkin dapat di kembangkan lebih baik lagi sehingga *games* ini akan terlihat lebih menarik.

Halaman ini sengaja di kosongkan

DAFTAR PUSTAKA

[1] A*dobe Flex 3.0 programing Action Script 3*. by Adobe System incorporated

[2] http://blog.papervision3d.org/2009/01/07/augmented-reality-with-flartoolkit/

[3] http://saqoo.sh/a/flartoolkit/start-up-guide

[4] http://www.mikkoh.com/blog/2008/12/flartoolkitflash-augmented-realitygetting-started/

[5] Kato, H., Billinghurst, M., dan Poupyrev, I., 2000, ”ARToolKit version 2.33: A software library for Augmented Reality Applications”, Human Interface Technology Laboratory, University of Washington

[6] Rekimoto J., “Matrix : A Real-Time Object Identification and Registration Method for Augmented Reality”, Proceedings of the third Asia Pacific on computer–human interactions, Kangawa Japan, p. 63–98, 1998

[7] Tedy Gorbala, Mochamad Hariadi, Aplikasi Augmented Reality untuk Katalog Penjualan Rumah Bregga, Teknik Komputer & Telematika, Teknik Elektro ITS Surabaya

Halaman ini sengaja di kosongkan

LAMPIRAN

Program Bingung.as

```
package {

   import flash.display.Sprite;
   import flash.events.Event;
   import flash.events.KeyboardEvent;
   import flash.ui.Keyboard;

   import
org.papervision3d.objects.DisplayObject3D;
   import org.papervision3d.objects.parsers.DAE;

   [SWF(width=640, height=480,
backgroundColor=0x0, frameRate=30)]

      public class bingung extends PV3DARApp {

            //
__________________________________________________
_____________________ 3D vars

            private var rootNode
:DisplayObject3D;

            //
__________________________________________________
_____________________ Car vars

            private var topSpeed  :Number = 0;
            private var topSteer  :Number = 0;
            private var topUp   :Number = 0;
            private var Topdown       :Number =
0;
            private var Up            :Number =
0;
            private var speed     :Number = 0;
            private var steer     :Number = 0;

            //
```

```
_____ Keyboard vars

      private var keyRight   :Boolean = false;
      private var keyLeft    :Boolean = false;
      private var keyForward :Boolean = false;
      private var keyReverse :Boolean = false;
      private var keyUp    :Boolean = false;
      private var keyDown    :Boolean = false;

      public function bingung() {
            this.init('camera_para.dat',
'g.pat');
            }

            protected override function
onInit():void {
                  super.onInit();

                  var car:DAE = new DAE();
                  car.load('Focus.dae');
                  car.scaleX = 10;
                  car.scaleY = 10;
                  car.scaleZ = 10;
                  car.rotationX = 90;
                  car.rotationZ = -90;
                  this._baseNode.addChild(car);
                  this.rootNode = car;

                  stage.addEventListener(
KeyboardEvent.KEY_DOWN, keyDownHandler );
                  stage.addEventListener(
KeyboardEvent.KEY_UP, keyUpHandler );

                  this.addEventListener(
Event.ENTER_FRAME, loop3D );
            }


            //
_____
```

```
_____________________ Keyboard

            private function keyDownHandler(
event :KeyboardEvent ):void
            {
            switch( event.keyCode )
                  {
                  case "W".charCodeAt():
                  case Keyboard.UP:
                        keyForward = true;
                        keyReverse = false;
                        break;

                  case "S".charCodeAt():
                  case Keyboard.DOWN:
                        keyReverse = true;
                        keyForward = false;
                        break;

                  case "A".charCodeAt():
                  case Keyboard.LEFT:
                        keyLeft = true;
                        keyRight = false;
                        break;

                  case "D".charCodeAt():
                  case Keyboard.RIGHT:
                        keyRight = true;
                        keyLeft = false;
                        break;

                  case "X".charCodeAt():
                        keyUp = true;
                        keyDown = false;
                        break;


case Keyboard.ENTER:
   this.enableDetection = !this.enableDetection;
   break;
                  }
```

```
//trace("keyDownHandler: " + event.keyCode);
            }

private function keyUpHandler( event
:KeyboardEvent ):void
            {
      switch( event.keyCode )
                  {
            case "W".charCodeAt():
            case Keyboard.UP:
                  keyForward = false;
                  break;

            case "S".charCodeAt():
            case Keyboard.DOWN:
                  keyReverse = false;
                  break;

            case "A".charCodeAt():
            case Keyboard.LEFT:
                  keyLeft = false;
                  break;

            case "D".charCodeAt():
            case Keyboard.RIGHT:
                  keyRight = false;
                  break;

            case "X".charCodeAt():
                  keyUp = false;
                  break;
                  }
//trace("keyUpHandler: " + event.keyCode);
            }


            //
_____________________________________________________
____________________ driveCar

            private function driveCar():void
```

```
		{
			// Speed
			if( keyForward )
			{
				topSpeed = 2;
			}
			else if( keyReverse )
			{
				topSpeed = -1;
			}

			else if( keyUp )
			{
				topUp = +1;
			}

			else
			{
				topSpeed = 0;
			}

		speed -= ( speed - topSpeed ) / 20;

			// Steer
			if( keyRight )
			{
				if( topSteer < 45 )
				{
					topSteer += 5;
				}
			}
			else if( keyLeft )
			{
				if( topSteer > -45 )
				{
					topSteer -= 5;
				}
			}

		else
		{
```

```
                  topSteer -= topSteer / 5;
            }

            steer -= ( steer - topSteer ) / 2;
            }


            //
_____________________________________________________
_____________________________________________
updateCar

private function updateCar( car :DisplayObject3D
):void
            {
      // Steer front wheels
      var steerFR :DisplayObject3D =
      car.getChildByName( "Steer_FR" );
      var steerFL :DisplayObject3D =
      car.getChildByName( "Steer_FL" );

      steerFR.rotationY = steer;
      steerFL.rotationY = steer;

      // Rotate wheels
      var wheelFR :DisplayObject3D =
      steerFR.getChildByName( "Wheel_FR" );
      var wheelFL :DisplayObject3D =
      steerFL.getChildByName( "Wheel_FL" );
      var wheelRR :DisplayObject3D =
      car.getChildByName( "Wheel_RR" );
      var wheelRL :DisplayObject3D =
      car.getChildByName( "Wheel_RL" );

      var roll :Number = speed/2
      wheelFR.roll(  roll );
      wheelRR.roll(  roll );
            wheelFL.roll( -roll );
            wheelRL.roll( -roll );

            // Steer car
```

```
            car.yaw( -speed * steer / 10 );

            // Move car
            car.moveForward( speed );

            }


            //
_____________________________________________________________
_____________________________________________
loop3D

private function loop3D( event :Event ):void
            {
      // Get plane from rootNode
var car :DisplayObject3D =
this.rootNode.getChildByName("Focus", true);

      // Check if car has been loaded
                  if( car )
                  {
      // Calculate current steer and speed
            driveCar();

      // Update car model
            updateCar( car );
                  }

            }
      }
}
```

Halaman ini sengaja di kosongkan

TENTANG PENULIS

Nama : **MAS ALI BAHTIAR**

TTL : **Gresik 15 maret 1989**

Alamat : **Sunan Giri 18 D no 13, Gresik**

Telp : **Home** : **(031) 3984168**

Hp : **(031) 60653005**

Email : **aleiytelkom06@yahoo.com**

Hobby : **Maen bola, catur, nonton, membaca novel**

RIWAYAT PENDIDIKAN

Institusi	Jurusan	Periode
PENS-ITS	Telekomunikasi	2009-20011
PENS-ITS	Telekomunikasi	2006-2009
SMU Muhammadiyah 1 Gresik	IPA	2003-2006
SLTP Muhammadiyah 1 Gresik	-	2000-2003
SD Muhammadiyah 1 Giri Gresik	-	1994-2000

Pada tanggal 25 januari 2011 penulis mengikuti Seminar Proyek Akhir sebagai salah satu persyaratan untuk mendapatkan gelar Sarjana Sains Terapan (SST) di Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh Nopember Surabaya (ITS).