

**RANCANG BANGUN APLIKASI PENGONTROL ROBOT  
BERBASIS *AUGMENTED REALITY***

**SKRIPSI**



**Disusun Oleh :  
I KOMANG TRIPAYANA  
09.12.518**

**PROGRAM STUDI TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2013**

---

# LEMBAR PERSETUJUAN

## RANCANG BANGUN APLIKASI PENGONTROL ROBOT BERBASIS AUGMENTED REALITY

### SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Teknik Strata Satu (S-1)*

Disusun oleh :  
**I KOMANG TRIPAYANA**  
NIM. 09.12.518

Mengetahui,  
Ketua Program Studi Teknik Elektro S-1

**M. Ibrahim Ashari, ST, MT**  
NIP.P.1030100358

Diperiksa dan Disetujui

Mengetahui  
Pembimbing I



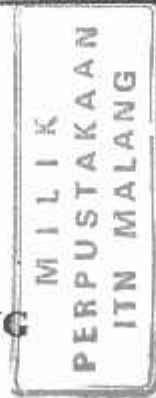
**M. Ibrahim Ashari, ST, MT**  
NIP.P.1030100358

Mengetahui  
Pembimbing II



**Bima Aulia Firmandani, ST**

**PRODI TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2013**



# **RANCANG BANGUN APLIKASI PENGONTROL ROBOT BERBASIS AUGMENTED REALITY**

**I Komang Tripayana (09.12.518)**

**Dosen Pembimbing : M. Ibrahim Ashari, ST, MT, Bima Aulia Firmandani, ST**

Jurusan Teknik Elektro S-1, Konsentrasi Teknik Komputer  
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang  
Jln. Raya Karanglo Km 2 Malang  
Email: [tripayana44@yahoo.com](mailto:tripayana44@yahoo.com)

## **ABSTRAK**

Sistem Pengontrol memberikan alternatif bagi manusia dalam menggerakkan atau mengendalikan suatu sistem dari jarak jauh. Sensor accelerometer pada smartphone kini telah banyak dimanfaatkan menjadi sebuah pengontrol dengan memanfaatkan sumbu X,Y dan Z. Bluetooth merupakan salah satu teknologi yang memungkinkan dua perangkat saling berkomunikasi secara nirkabel. Robot beroda merupakan salah satu contoh robot yang dapat dikendalikan.

Aplikasi ini memanfaatkan webcam dan ARToolkit library untuk menirukan sensor accelerometer pada smartphone. Dalam melakukan pengontrolan terhadap robot beroda ini, aplikasi yang dibangun dapat melakukan pengendalian robot dengan mengirimkan perintah untuk bergerak maju, mundur, ke kiri dan ke kanan. Dengan adanya aplikasi ini diharapkan membuktikan bahwa augmented reality dapat melakukan kendali terhadap sebuah robot beroda, dan juga aplikasi ini dapat digunakan sebagai alat bantu riset tim Robot ITN Malang.

Dari hasil pengujian konsep augmented reality dapat diterapkan sebagai pengontrol sebuah robot beroda. Pada kondisi kekurangan cahaya ( malam hari / lampu ruangan ) dan dengan rentang jarak yang cukup jauh antara marker dengan kamera aplikasi sulit melakukan deteksi marker dengan baik. Data yang dikirim berupa karakter, sehingga pergerakan kecepatan robot masih sulit untuk di control. Marker dicetak pada kertas yang tebal agar lebih mudah saat mendeteksi. Marker yang berukuran 40mmx40mm, jarak optimal agar aplikasi pengontrol robot dapat mendeteksi marker dengan baik yakni antara 15- 60 cm dari posisi kamera / webcam.

**Kata kunci:** Augmented Reality, Robot, Accelerometer.

---

## **SURAT PERNYATAAN ORISINALITAS**

Yang bertanda tangan di bawah ini:

Nama : I Komang Tripayana

NIM : 0912518

Program Studi : Teknik Elektro S-1

Konsentrasi : Teknik Komputer

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, September 2013

  ; Membuat Pernyataan,  


**I Komang Tripayana**  
NIM : 0912518

## KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas segala limpahan rahmat-Nya sehingga penulis dapat menyelesaikan laporan skripsi ini dengan judul “Rancang Bangun Aplikasi Pengontrol Robot Berbasis Augmented Reality”.

Adapun maksud dan tujuan dari penulisan laporan ini merupakan salah satu syarat untuk dapat menyelesaikan studi dan mendapatkan gelar Sarjana Jurusan Teknik Elektro S-1, Konsentrasi Teknik Komputer ITN Malang.

Penulis menyadari tanpa adanya kemauan dan usaha serta bantuan dari berbagai pihak, maka laporan ini tidak dapat diselesaikan tepat waktu. Oleh sebab itu penulis mengucapkan terimakasih kepada yang terhormat :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. H. Anang Subardi, MT. selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ibrahim Ashari, ST, MT selaku Ketua Jurusan Elektro sekaligus Dosen pembimbing 1.
4. Bapak Bima Aulia Firmandani, ST selaku Dosen pembimbing 2.
5. Kedua orang tua dan kakak saya yang selalu mendukung dalam keadaan apapun.
6. Teman-teman seperjuangan dalam mengerjakan skripsi, Huget adi, Kembo Ambon, Sakir dan spesial thx to orang berek sedunia yang selalu menemani.
7. Teman-teman yang selalu menghibur, Ohank alias ibo dan hendra, thx guys.
8. Dan semua Pihak yang telah membantu dalam penulisan dan penyusunan penelitian ini.

Penulis menyadari bahwa penelitian ini masih jauh dari sempurna, untuk itu kritik dan saran dari pembaca sangat penulis harapkan untuk perbaikan penelitian ini.

Malang, Agustus 2013

Penulis

## DAFTAR ISI

KATA PENGANTAR .....	i
DAFTAR ISI .....	ii
DAFTAR GAMBAR .....	iv
DAFTAR TABEL .....	vi
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan .....	2
1.4. Manfaat .....	2
1.5. Batasan Masalah .....	2
1.6. Metodologi Penelitian .....	2
1.7. Sistematika Penulisan .....	3
BAB II DASAR TEORI .....	5
2.1. Pendahuluan .....	5
2.2. Sensor Accelerometer .....	5
2.3. Augmented Reality .....	6
2.3.1. Jenis-jenis Augmented Reality .....	8
2.3.2. Cara Kerja <i>Augmented Reality</i> .....	8
2.4. Robot .....	13
2.4.1. Perkembangan Robot .....	14
2.4.2. Jenis-jenis Robot .....	15
2.5. Visual Basic .....	16
2.5.1. Sejarah Visual Basic .....	18
2.6. Visual Studio .....	22
2.7. Bluetooth .....	23
2.7.1. Pengertian Bluetooth .....	23
2.7.2. Perkembangan Sejarah Bluetooth .....	24
2.7.3. Cara Kerja Bluetooth .....	24
BAB III PERANCANGAN DAN DESAIN APLIKASI .....	27
3.1. Analisa Sistem .....	27
3.2. Deskripsi Umum Sistem .....	27
3.3. Analisa Kebutuhan Sistem .....	27
3.3.1. Kebutuhan Perangkat Lunak .....	28
3.3.2. Kebutuhan Perangkat Keras .....	28

3.4. Perancangan Aplikasi .....	28
3.4.1. Diagram Blok .....	29
3.4.2. Diagram Alir .....	29
3.5. Perancangan Tata Letak Antarmuka .....	31
3.5.1. Tampilan Antarmuka Video Capture .....	31
3.5.2. Tampilan Antarmuka Menu Utama .....	31
3.6. Perancangan Proses Deteksi Marker .....	32
3.6.1. Sistem Koordinat AR .....	32
3.6.2. Kalibrasi Kamera .....	33
3.6.3. Flowchart Sistem Deteksi Marker .....	35
BAB IV IMPLEMENTASI DAN PENGUJIAN .....	37
4.1. Implementasi .....	37
4.2. Pengujian .....	37
4.2.1. Pengujian spesifikasi .....	37
4.2.1.1. Spesifikasi Perangkat Keras .....	38
4.2.1.2. Spesifikasi Perangkat Lunak .....	38
4.2.2. Pengujian Bluetooth .....	38
4.2.3. Pengujian Aplikasi .....	40
4.2.3.1. Pengujian Arah Pergerakan Robot .....	40
4.2.4. Pengujian Deteksi Marker .....	46
4.2.4.1. Deteksi Marker Pada Intensitas Cahaya Tertentu & Pengaruh Arah Pergerakan Robot .....	47
4.2.4.2. Deteksi Marker Pada Jarak Tertentu & Pengaruh Arah Pergerakan Robot .....	49
BAB V PENUTUP .....	53
5.1. Kesimpulan .....	53
5.2. Saran .....	53
DAFTAR PUSTAKA .....	54



## DAFTAR GAMBAR

Gambar 2.1 Sistem Koordinat Dari Telepon Genggam.....	6
Gambar 2.2 Accelometer pada <i>Smartphone</i> .....	6
Gambar 2.3 Contoh AR .....	7
Gambar 2.4 Proses deteksi <i>Marker</i> .....	8
Gambar 2.5 Prinsip kerja <i>Augmented Reality</i> .....	9
Gambar 2.6 Tampilan webcam awal.....	10
Gambar 2.7 Proses Greyscale <i>Marker</i> .....	10
Gambar 2.8 Process Convert menjadi hitam putih .....	11
Gambar 2.9 Proses Labelling.....	11
Gambar 2.10 Match up the patterns .....	12
Gambar 2.11 Match up the patterns .....	12
Gambar 2.12 Hasil Render objek .....	13
Gambar 2.13 Logo Visual Basic .....	16
Gambar 2.14 Logo Visual Studio .....	23
Gambar 2.15 Logo Bluetooth.....	24
Gambar 2.16. Blok fungsional sistem bluetooth.....	25
Gambar 2.17. Layer-layer pada sistem bluetooth. ....	25
Gambar 3.1 Diagram Blok .....	29
Gambar 3.2 Diagram Alir ( Flowchart ) .....	30
Gambar 3.3 tata letak halaman video capture .....	31
Gambar 3.4 tata letak halaman utama .....	32
Gambar 3.5 Sistem koordinat kamera terhadap marker.....	32
Gambar 3.6 Perbandingan antara gambar ideal dengan gambar yang disebabkan factor distorsi.....	34
Gambar 3.7 Diagram Alir Proses Deteksi Marker.....	35
Gambar 4.1 jendela accessport.....	39
Gambar 4.2 jendela konfigurasi accessport.....	39
Gambar 4.3 pengujian pengiriman data melalui Bluetooth .....	40
Gambar 4.4 posisi marker saat bergerak maju .....	40
Gambar 4.5 koordinat sumbu X,Y dan Z saat bergerak maju.....	41
Gambar 4.6 Robot bergerak maju .....	41
Gambar 4.7 Posisi marker saat bergerak mundur .....	42
Gambar 4.8 koordinat sumbu X, Y dan Z saat bergerak mundur .....	42
Gambar 4.9 Robot bergerak mundur.....	43
Gambar 4.10 posisi marker saat bergerak ke kanan.....	43
Gambar 4.11 koordinat sumbu X, Y dan Z saat bergerak ke kanan .....	44
Gambar 4.12 Robot bergerak ke kanan.....	44
Gambar 4.13 posisi marker saat bergerak ke kiri.....	45
Gambar 4.14 koordinat sumbu X, Y dan Z saat bergerak ke kiri .....	45
Gambar 4.15 Robot bergerak ke kiri.....	46
Gambar 4.16 pola marker yang digunakan .....	47



Gambar 4.17 pengujian intensitas cahaya sedang (pagi) .....	47
Gambar 4.18 pengujian intensitas cahaya baik ( siang hari ).....	48
Gambar 4.19 pengujian intensitas cahaya kurang ( malam hari ).....	48
Gambar 4.20 posisi marker dengan jarak dekat.....	49
Gambar 4.21 hasil koordinat sumbu X, Y, Z saat jarak dekat .....	50
Gambar 4.22 posisi marker saat jarak sedang .....	50
Gambar 4.23 hasil koordinat sumbu X, Y, Z saat jarak sedang.....	51
Gambar 4.24 posisi marker saat jarak jauh .....	51
Gambar 4.25 hasil koordinat sumbu X, Y, Z saat jarak jauh .....	52

## DAFTAR TABEL

Tabel 3.1 Spesifikasi Perangkat Lunak.....	28
Tabel 3.2 Spesifikasi Perangkat Keras.....	28
Tabel 4.1 spesifikasi perangkat keras untuk pengujian.....	38
Tabel 4.2 spesifikasi perangkat lunak untuk pengujian .....	38
Tabel 4.3 hasil pengujian arah gerak robot .....	46
Tabel 4.4 hasil pengujian deteksi marker pada intensitas cahaya tertentu.....	49
Tabel 4.5 hasil pengujian deteksi marker dengan jarak tertentu.....	52

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

*Augmented Reality* adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata (*real time*) . Tidak seperti realitas maya yang sepenuhnya menggantikan kenyataan, namun *Augmented Reality* hanya menambahkan atau melengkapi kenyataan.

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu ( kecerdasan buatan ).

Sensor accelerometer adalah sebuah sensor yang dapat mengukur percepatan, mendeteksi dan mengukur getaran ( vibrasi ) dan mengukur kecepatan akibat gravitasi. Pada awalnya sensor accelerometer pada smartphone dimanfaatkan untuk merubah tampilan dari landscape ke portrait pada layar ataupun sebaliknya.

Dengan adanya sensor accelerometer pada smartphone, kini telah banyak dikembangkan aplikasi yang memanfaatkan sensor ini, seperti game yang terdapat pada smartphone yang menggunakan sensor accelerometer sebagai controller.

Pemanfaatan sensor accelerometer pada smartphone telah banyak dikembangkan dan dimanfaatkan sebagai controller robot.

Dari pembahasan diatas, peneliti tertarik untuk memanfaatkan dan menggabungkan teknologi augmented reality dengan sensor accelerometer dan mengemukakan sebuah ide “RANCANG BANGUN APLIKASI PENGONTROL ROBOT BERBASIS AUGMENTED REALITY” dengan harapan dapat menjadi pengganti sensor accelerometer pada smartphone yang digunakan sebagai pengontrol robot dan juga dapat menjadi aplikasi yang membantu tim robotika ITN Malang dalam melakukan riset-risetnya.

## 1.2. Rumusan Masalah

Apakah konsep *Augmented Reality* dapat digunakan sebagai pengontrol robot?

## 1.3. Tujuan

Adapun tujuan dari pembuatan program aplikasi *Augmented Reality* sebagai pengontrol robot adalah sebagai berikut :

1. Pemanfaatan *Augmented Reality* sebagai pengontrol robot.
2. Menunjukkan bahwa konsep *Augmented Reality* dapat diterapkan sebagai pengontrol robot.

## 1.4. Manfaat

Aplikasi ini nantinya dapat dimanfaatkan sebagai alat bantu riset tim robotika Elektro ITN Malang.

## 1.5. Batasan Masalah

Agar pembahasan dalam penulisan ini lebih terarah dan mencegah adanya perluasan masalah dan pembahasan yang terlalu kompleks, maka penulis membuat batasan masalah yang akan dijadikan pedoman dalam pelaksanaan tugas akhir yaitu :

1. Tidak membahas proses pembuatan *accelkit*
2. Tidak membahas perancangan robot.
3. Tidak membahas *module* komunikasi robot dengan *controller*.
4. Jarak maximum robot dengan *controller* adalah 10m.
5. *User* harus sejajar dengan robot.

## 1.6. Metodologi Penelitian

Metodologi penelitian yang digunakan oleh penulis dalam pengumpulan data adalah:

1. Studi literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

---

## 2. Analisa Kebutuhan Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem dimana nantinya akan digunakan sebagai acuan perancangan sistem.

## 3. Perancangan dan Implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat dan diimplementasikan ke dalam sistem.

## 4. Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

### 1.7. Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

#### Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, Metode Penelitian dan Sistematika Penulisan.

#### Bab II : Tinjauan Pustaka

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

#### Bab III : Perancangan dan Analisa Sistem

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik *software* maupun *hardware* yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

#### Bab IV : Pembuatan dan Pengujian Sistem

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

---

## Bab V : Penutup

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

---

## BAB II

### DASAR TEORI

#### 2.1 Pendahuluan

Pada bab ini akan diterangkan mengenai media dan teori penunjang yang digunakan untuk pembuatan aplikasi. Pembahasan pada bab ini meliputi:

1. *Sensor Accelerometer*
2. *Augmented Reality*
3. Robot
4. Visual Basic
5. Visual Studio
6. Bluetooth

#### 2.2 Sensor Accelerometer

*Accelerometer* adalah sensor yang dapat di gunakan untuk mengukur percepatan, termasuk percepatan gravitasi bumi :  $9,8 \text{ m/s}^2$  . Seperti kita ketahui bahwa percepatan adalah hasil turunan pertama dari besaran kecepatan dan atau turunan kedua dari posisi. Yang artinya, hanya dengan menggunakan sensor accelerometer kita dapat membuat alat ukur untuk besaran percepatan, kecepatan dan posisi, baik angular maupun linear seperti gyroscope, linear acceleration , speedometer , sensor kemiringan, sensor getaran dsb.

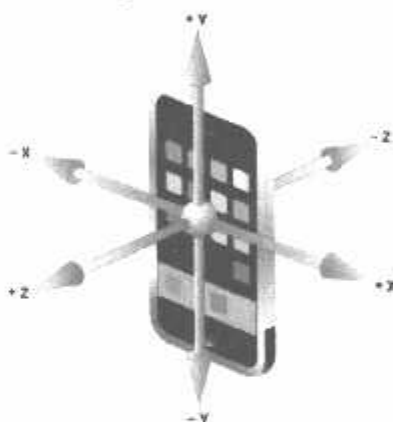
Munculnya *Android Smartphone* yang telah dilengkapi dengan berbagai macam sensor memungkinkan kita untuk membuat sendiri sebuah *Handled Instrument* yang multi fungsi, ditambah lagi dengan kemampuannya dalam akses internet memungkinkan kita untuk menciptakan sendiri sebuah sistem alat ukur dan kontrol yang dapat di monitor ataupun di kendalikan dari jarak jauh melalui jaringan internet ataupun *bluetooth*.





Gambar 2.1 Sistem Koordinat Dari Telepon Genggam

Android *Smartphone* di lengkapi dengan sebuah 3 axis Accelerometer pada sumbu X, Y dan Z dengan ilustrasi sebagai berikut:



Gambar 2.2 Accelometer pada *Smartphone*

### 2.3 Augmented Reality

*Augmented Reality* adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata (*real time*). Tidak seperti realitas maya yang sepenuhnya menggantikan kenyataan, namun *Augmented Reality* hanya menambahkan atau melengkapi kenyataan.

Menurut Ronald Azuma pada tahun 1997, *Augmented Reality* adalah menggabungkan dunia nyata dan virtual, bersifat interaktif secara *real time*, dan bentuknya merupakan animasi 3D. Yang dimaksud interaktif disini adalah, adanya interaksi dari *user* ke AR tersebut. Sehingga ada pengaruh di *Augmented Reality* tersebut, seperti misalnya, *user* menggunakan *handphone* yang terdapat tombol-tombol

untuk menjalankan atau member efek pada *Augmented Reality*. *Augmented Reality* seperti ini biasanya ada digunakan *Smartphone* untuk membuat *Game Augmented Reality* yang bersifat Interaktif.

Benda-benda maya menampilkan informasi yang tidak dapat diterima oleh pengguna dengan inderanya sendiri. Hal ini membuat *Augmented Reality* sesuai sebagai alat untuk membantu persepsi dan interaksi penggunaanya dengan dunia nyata. Informasi yang ditampilkan oleh benda maya membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata.

Ada tiga karakteristik yang menyatakan suatu teknologi menerapkan konsep *Augmented Reality*, yaitu :

1. Mampu mengkombinasikan dunia nyata dan dunia maya..
2. Mampu memberikan informasi secara interaktif dan real-time.
3. Mampu menampilkan dalam bentuk 3D ( tiga dimensi ).



Gambar 2.3 Contoh AR

Selain menambahkan benda maya dalam lingkungan nyata, *Augmented Reality* juga berpotensi menghilangkan benda-benda yang sudah ada. Menambah sebuah lapisan gambar maya dimungkinkan untuk menghilangkan atau menyembunyikan lingkungan nyata dari pandangan pengguna. Misalnya, untuk menyembunyikan sebuah meja dalam lingkungan nyata, perlu digambarkan lapisan representasi tembok dan lantai kosong yang diletakkan di atas gambar meja nyata, sehingga menutupi meja nyata dari pandangan pengguna.

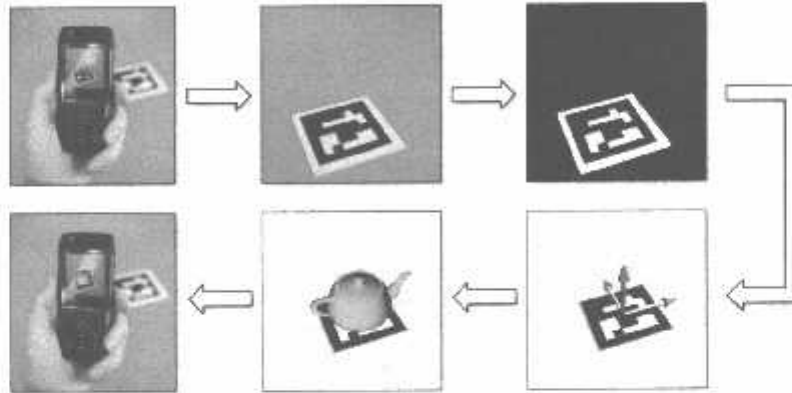
Sebagai contoh adalah saat pembawa acara televisi membawakan berita, ada animasi atau objek *virtual* yang ikut bersamanya, jadi seolah-olah dia berada didalam dunia *virtual* tersebut, padahal sebenarnya itu adalah tehnik penggabungan antara dunia *virtual* dengan dunia nyata yang dinamakan dengan *Augmented Reality*.

### 2.3.1. Jenis-jenis Augmented Reality

*Augmented Reality* ( AR ) terbagi menjadi dua macam berdasarkan metode penggunaannya yaitu *Marker Augmented Reality* dan *Markerless Augmented Reality*.

#### A. *Marker Augmented Reality ( Marker Based Tracking )*

Merupakan salah satu metode yang memanfaatkan *Marker* berupa ilustrasi hitam dan putih berbentuk persegi atau lainnya dengan batas hitam tebal dan latar belakang putih. Melalui posisi yang dihadapkan pada sebuah kamera computer, maka computer akan melakukan proses menciptakan dunia virtual 2D atau 3D.



Gambar 2.4 Proses deteksi *Marker*

#### B. *Markerless Augmented Reality*

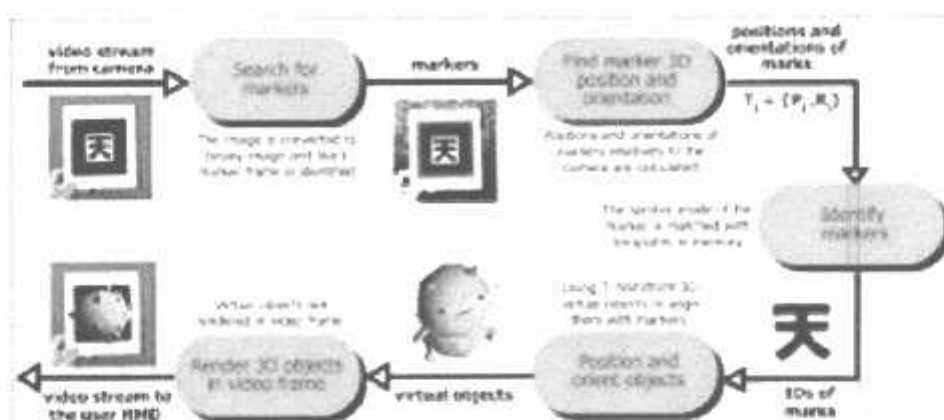
Salah satu metode *Augmented Reality* yang saat ini sedang berkembang adalah metode "*Markerless Augmented Reality*", dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *Marker* untuk menampilkan elemen-elemen digital.

Seperti yang saat ini dikembangkan oleh perusahaan *Augmented Reality* terbesar di dunia Total Immersion, mereka telah membuat berbagai macam teknik *Markerless Tracking* sebagai teknologi andalan mereka, seperti Face Tracking, 3D Object Tracking, dan Motion Tracking.

### 2.3.2 Cara Kerja Augmented Reality

Sistem *Augmented Reality* bekerja berdasarkan deteksi citra dan citra yang digunakan adalah *Marker*. Prinsip kerjanya sebenarnya cukup sederhana. Webcam yang telah dikalibrasi akan mendeteksi *Marker* yang diberikan, kemudian setelah mengenali dan menandai pola *Marker*, webcam akan melakukan perhitungan apakah *Marker*

sesuai dengan database yang dimiliki. Bila tidak, maka informasi *Marker* tidak akan diolah, tetapi bila sesuai maka informasi *Marker* akan digunakan untuk me-render dan menampilkan objek 3D atau animasi yang telah dibuat sebelumnya.

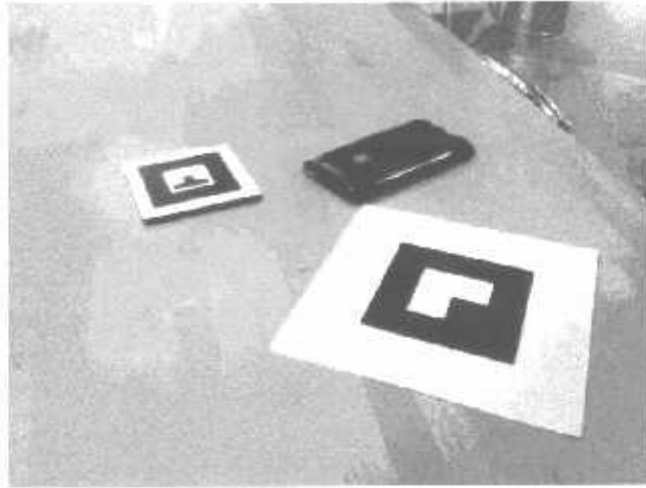


Gambar 2.5 Prinsip kerja *Augmented Reality*

Untuk lebih lengkapnya, berikut tahapan utama sistem *Augmented Reality* tersebut :

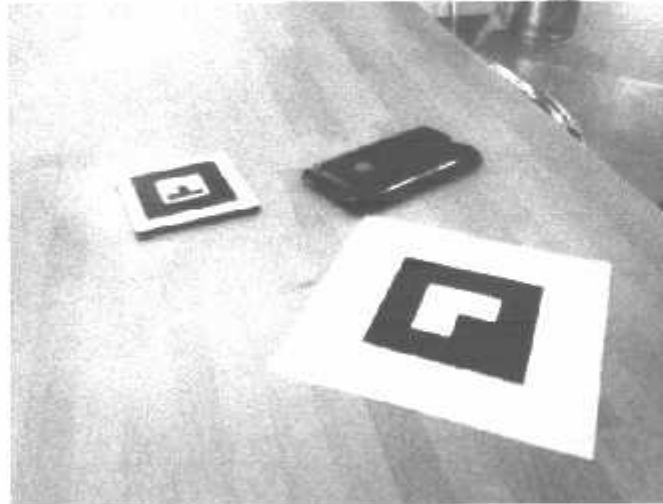
1. Pertama – tama dibuat terlebih dahulu objek yang akan ditampilkan. Secara umum objek yang dibuat adalah benda 3D , foto, video, ataupun animasi yang dibuat dengan software perancangan objek seperti Google sketchup, 3DMax , atau dengan Blender.
2. Setelah objek jadi, maka objek tersebut akan disimpan ke dalam library. Kemudian setelah itu yang perlu dibuat adalah *Marker*. *Marker* adalah sebuah penanda yang memiliki pola khusus. *Marker* yang digunakan untuk teknologi *Augmented Reality* ini adalah pola kotak dengan standard tertentu. *Marker* inilah yang nantinya akan dideteksi oleh webcam untuk menampilkan objek. Setelah itu pola *Marker* yang dibuat harus disimpan ke dalam library juga agar nantinya aplikasi yang dibuat dapat membedakannya dengan *Marker* yang lain. Biasanya penyimpanan *Marker* ini membutuhkan bantuan aplikasi lain seperti *Marker generator*.
3. Kedua komponen utama yaitu *Marker* dan objek telah jadi. Setelah itu kita membuat aplikasi yang dapat membangkitkan objek dari *Marker* yang dibuat dengan bantuan builder, untuk tugas akhir ini adalah dengan Adobe Flex SDK yang bersifat open source dan berbasis action script.
4. Sekarang saatnya mencoba dengan menghadapkan *Marker* ke depan webcam. Akan ada beberapa proses pendeteksian dan pengkalkulasian gambar yaitu :

- **Get Webcam Image.** Webcam mendeteksi seluruh citra yang berada di dalam lingkungannya. Saat ini webcam berfungsi seperti webcam biasa yaitu menampilkan gambar seperti kebalikan dari cermin.

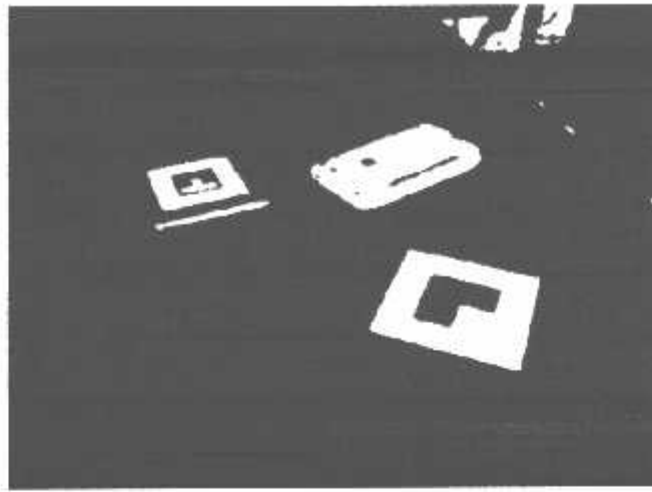


Gambar 2.6 Tampilan webcam awal

- **Process webcam image.** Untuk bagian ini webcam mengubah citra yang terekam dari komposisi warna RGB menjadi grey scale kemudian warna diconvert menjadi hitam putih

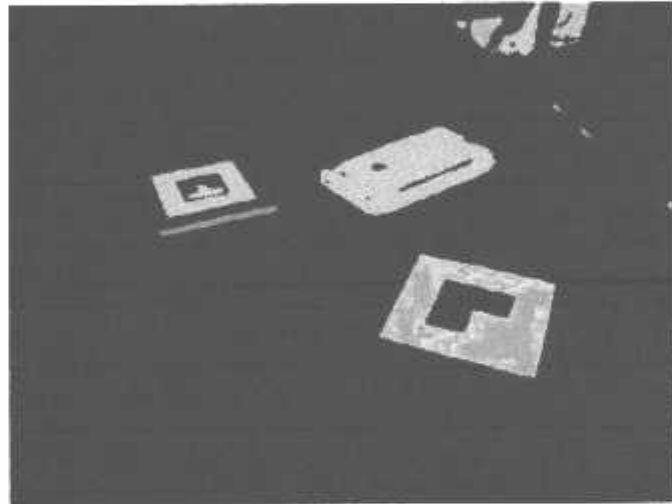


Gambar 2.7 Proses Greyscale Marker



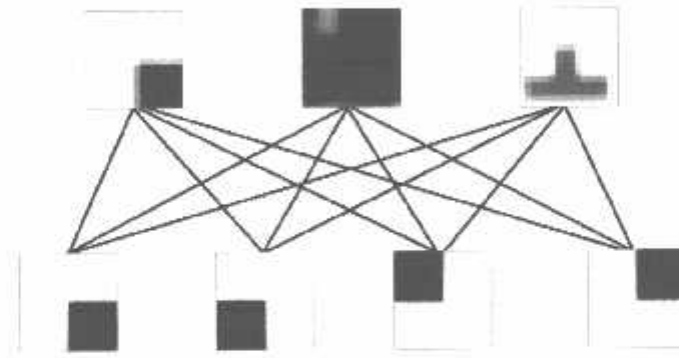
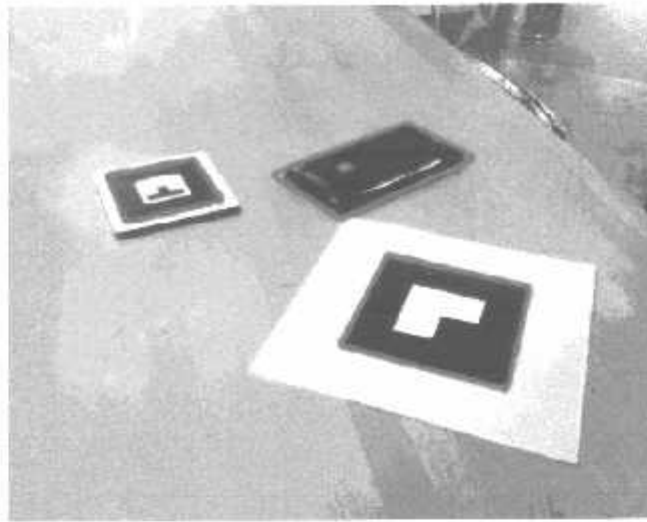
Gambar 2.8 Process Convert menjadi hitam putih

- **Labeling.** Setelah itu akan ditemukan pola kotak-kotak yang merupakan pola dasar *Marker AR*. Dari pola dasar inilah akan dilakukan labeling :



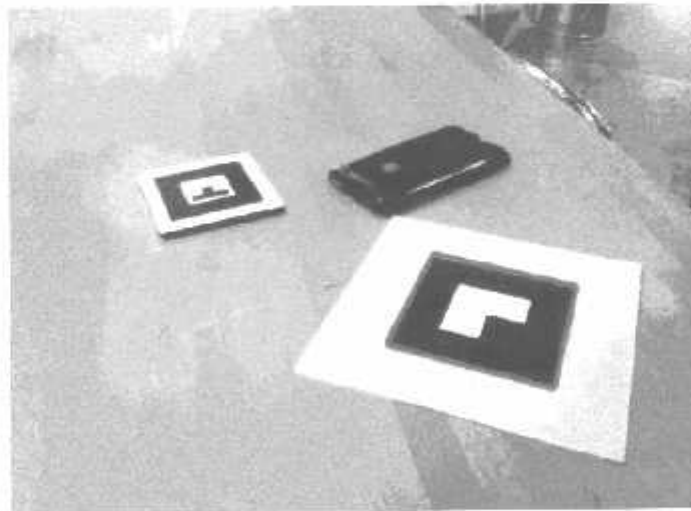
Gambar 2.9 Proses Labelling

- **Match up the patterns .** Labelling menentukan pola-pola yang kira-kira dideteksi sebagai *Marker*. Sekarang akan dilakukan perhitungan dan pencocokan *Marker* yang dideteksi dengan *Marker* di dalam database.
-



Gambar 2.10 Match up the patterns

- Jika *Marker* telah sesuai dengan *Marker* yang ada didalam database, baru setelah itu image dirender dan ditampilkan ke dalam dunia maya.



Gambar 2.11 Match up the patterns





Gambar 2.12 Hasil Render objek

#### 2.4. Robot

Robot berasal dari bahasa Czech, *robota*, yang berarti pekerja. Pada dasarnya robot dibuat untuk mendukung dan membantu pekerjaan manusia, seperti yang banyak terlihat di bidang industri dimana robot dapat meningkatkan hasil produksi industri. Kata robot diperkenalkan dalam bahasa Inggris pada tahun 1921 oleh Wright Karel Capek dalam salah satu drama satiristiknya, *R.U.R (Rossum's Universal Robots)*.

Sedangkan pengertian robot adalah sistem atau alat yang dapat berperilaku atau meniru perilaku manusia dengan tujuan untuk menggantikan dan mempermudah kerja/aktivitas manusia. Awal muncul robot dapat diketahui dari bangsa Yunani kuno yang membuat patung yang dapat dipindah-pindahkan. Sekitar 270 BC, seorang insinyur Yunani membuat dengan komponen yang dapat dipindahkan. Tahun 1941, barulah istilah *robotics* digunakan dalam teknologi robot oleh penulis fiksi ilmiah Isaac Asimov. Dia juga memprediksi akan munculnya robot-robot industri canggih di masa industri. Jika kita lihat hari ini, maka apa yang dibayangkan olehnya terbukti dimana begitu pesatnya perkembangan robot-robot industri saat ini. Istilah revolusi robot, era robot sudah menjadi hal biasa untuk menjelaskan perkembangan itu. *Robotics* diterima sebagai istilah atau kata untuk mendeskripsikan semua kemajuan teknologi yang berhubungan dengan robot.

#### 2.4.1. Perkembangan Robot

Ketika Georde Devil dan Joseph Engelberger membentuk perusahaan robot pertama kali tahun 1956. Devil memprediksi robot akan menjadi bagian penting di industri sebagai operator pabrik dan membantu pekerja dalam menjalankan mesin-mesin pabrik. Beberapa tahun kemudian atau tepatnya 1961, General Motor pertama kali menggunakan robot untuk pabrik otomotifnya. Robot industri kemudian berkembang dan mulai banyak digunakan tahun 1980 oleh perusahaan selain otomotif dimana perkembangan elektronik dan computer membuat robot modern lahir.

Karakuri ningyo, merupakan istilah Jepang yang berarti boneka mekanik atau automata, ditemukan pada abad ke-18 dan 19 Masehi. Di Jepang Karakuri dapat dibagi menjadi tiga tipe utama yakni:

1. Butai karakuri (stage karakuri), digunakan untuk keperluan dunia teater
2. Zashiki karakuri (tatami room karakuri), merupakan tipe karakuri berukuran kecil dan digunakan sebagai elemen dekorasi ruangan
3. Dashi karakuri (festival car karakuri), digunakan dalam acara atau festival keagamaan, dengan menampilkan mitos-mitos tradisional atau legenda-legenda Bangsa Jepang.

Ketiga jenis karakuri tersebut dinilai telah memberikan pengaruh besar bagi perkembangan dunia teater Jepang seperti Noh, Kabuki, Bunraku.

Dewasa ini, karakuri pun berkembang menjadi Matsuri karakuri, digunakan untuk keperluan festival, Kogyo karakuri, digunakan untuk keperluan hiburan seperti pertunjukan boneka. Zashiki karakuri, digunakan untuk keperluan dekorasi (elemen dekoratif) dalam ruangan. Saat ini konfigurasi kinematik dikenal sebagai standart lengan robot. Terakhir pada tahun 2000 Honda memamerkan robot yang dibangun bertahun-tahun lamanya bernama ASIMO, serta disusul oleh Sony yaitu robot anjing AIBO.

Robot banyak dibuat oleh industry Riset, universitas, Departemen Pertahanan, serta institusi besar lainnya seperti NASA dan SONY. Saat ini hampir semua industry manufaktur menggunakan robot, karena biaya per jam untuk mengoperasikan robot jauh lebih murah dibandingkan menggunakan manusia. Robot pada awalnya digunakan untuk melakukan fungsi spesifik, misalnya pengecoran, penyoderan, dan lain-lain. Namun saat ini sudah banyak robot yang melakukan banyak fungsi.

---

Bentuk robot seperti manusia tidak lagi diperhatikan meski perkembangan robot android atau humanoid tetap berlangsung dan mengalami penyempurnaan. Kini robot adalah pekerja industri atau berupa tangan dan lengan yang dikontrol oleh computer dan dapat dirubah fungsinya dengan mengedit program robot. Bentuk robot industri ini lebih dikenal sekarang dibanding robot menyerupai manusia.

Kemampuan robot untuk melakukan gerakan manusia sangat membantu dunia industri seperti industri mobil, proses pengelasan, perakitan, pemindahan dan banyak lagi. Gerakan berulang yang presisi adalah salah satu keunggulan robot daripada manusia sehingga didapat hasil produksi yang konstan dan standard.

Robot industri harus diprogram untuk melakukan semua step gerakan atau kerja sebelum ia digunakan. Tahap awal ini bisa disebut merangkai atau membangun pola berfikirnya robot. Benda kerja harus ditempatnya ditempat yang pasti dan tidak berubah-ubah selama proses (meski sekarang kemajuan object recognition sudah maju namun dalam prakteknya benda kerja masih harus diposisikan ditempat yang tetap). Jika benda kerja melcsot dari posisinya maka proses akan salah dan robot tidak bisa mengkoreksinya. Robot tidak bisa melihat dan mendengar. Dia tidak bisa merasakan objek dan meprediksi adanya kesalahan dan robot tidak memiliki kemampuan mengadopsi situasi baru yang terjadi disekitarnya.

Robot memberikan keuntungan tersendiri bagi pekerja industri dan suatu negara dimana ia bisa memperbaiki kualitas hidup manusia karena bebas dari pekerjaan yang menjenuhkan, kotor dan penuh resiko atau dalam istilahnya 3D= Dull, Dirty and Dangerous. Benar bahwa robot akan menimbulkan pengangguran tapi jangan lupa robot juga menciptakan lapangan pekerjaan; Insinyur robot, Teknisi, Sales, Programmer dan Pengawas/supervisor. Robot memberikan keuntungan bagi industri karena adanya peningkatan output dan perbaikan kualitas. Industri robot tidak mengenal lelah dan keluhan, ia bisa bekerja tanpa lelah siang malam dengan performance yang sama. Akibatnya, biaya produk per unit akan turun, menaikkan keuntungan dan memberi dampak positif terhadap pasar serta ekonomi dunia secara keseluruhan.

#### **2.4.2. Jenis-jenis Robot**

Namun secara garis besar robot dapat diklasifikasikan menjadi beberapa jenis antara lain:

- 1) Robot industri
- 2) Robot antariksa
- 3) Robot transportasi
- 4) Robot perang
- 5) Robot kendali jarak jauh
- 6) Robot kedokteran
- 7) Robot riset
- 8) Robot bermain, dll

## 2.5. Visual Basic

Microsoft Visual Basic (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang bersifat event driven dan menawarkan Integrated Development Environment (IDE) visual untuk membuat program aplikasi berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman Common Object Model (COM). Visual Basic merupakan turunan bahasa BASIC dan menawarkan pengembangan aplikasi komputer berbasis grafik dengan cepat, akses ke basis data menggunakan Data Access Objects (DAO), Remote Data Objects (RDO), atau ActiveX Data Object (ADO), serta menawarkan pembuatan kontrol ActiveX dan objek ActiveX. Beberapa bahasa skrip seperti Visual Basic for Applications (VBA) dan Visual Basic Scripting Edition (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda.



Gambar 2.13 Logo Visual Basic

Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh Microsoft Visual Basic. Program-program yang ditulis dengan Visual Basic juga dapat menggunakan Windows API, tapi membutuhkan deklarasi fungsi eksternal tambahan.

---

Dalam pemrograman untuk bisnis, Visual Basic memiliki pangsa pasar yang sangat luas. Dalam sebuah survey yang dilakukan pada tahun 2005, 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk Visual Basic, yang diikuti oleh C++, JavaScript, C#, dan Java.

Visual Basic merupakan bahasa yang mendukung OOP, namun tidak sepenuhnya. Beberapa karakteristik obyek tidak dapat dilakukan pada Visual Basic, seperti Inheritance tidak dapat dilakukan pada class module. Polymorphism secara terbatas bisa dilakukan dengan mendeklarasikan class module yang memiliki Interface tertentu. Visual Basic (VB) tidak bersifat case sensitif.

Pada dasarnya Visual Basic diawali dari perkembangan bahasa BASIC di dardmouth College, Amerika Serikat, pada awal tahun 1960-an. Sejak semula BASIC memang dirancang untuk mudah dipelajari. Begitu sederhananya sehingga nyaris semua pakar pemrograman komputer menggunakan BASIC sebagai bahasa pemrograman pertamanya. Pada tahun 1982 IBM/PC diperkenalkan pada masyarakat, Microsoft pun membuat sistem operasi MS-DOS untuk komputer ini. Didalamnya disertakan pula bahasa BASIC yang dikenal sebagai QuickBasic (QBASIC). Pada tahun 1990-an era DOS berlalu digantikan era Windows. Tampilan grafis windows yang sangat bagus dan lebih interaktif mengubah pemrograman dari pekerjaan yang memusingkan kepala menjadi sebuah pekerjaan yang sangat menyenangkan. Microsoft pun akhirnya membuat BASIC versi Windows yang dikenal dengan Microsoft Visual Basic. Visual Basic memungkinkan pembuatan aplikasi Grapichal User Interface (GUI) atau pemrograman yang menggunakan tampilan grafis sebagai alat komunikasi dengan pemakainya. Pembuatan tampilan user interface dapat dilakukan dengan mcletakkan objectobject grafis ke lembar (form) yang sudah disediakan oleh Visual Basic.

Seperti yang telah diketahui, Visual Basic merupakan bahasa pemrograman Visual yang dapat mempermudah dalam mendesain tampilan program atau lebih dikenal dengan istilah *user interface*. Sehingga hal ini sangat bermanfaat untuk membuat program yang bekerja dalam lingkungan windows yang tampilannya lebih rumit. Dengan Bahasa Pemrograman biasa / Non Visual, waktu seorang programmer lebih banyak dihabiskan untuk mendesain tampilan program dibandingkan dengan penulisan program utamanya. Visual basic adalah suatu bahasa pemrogramman visual yang merupakan pengembangan terakhir dari Basic.

---

### 2.5.1. Sejarah Visual Basic

- 20 Mei 1991

Microsoft merilis Visual Basic versi 1.0 untuk Windows di Windows World 1991, Atlanta, Amerika Serikat. Programmer dapat membuat antar muka pengguna dengan mudah. Pembuatan aplikasi bisa lebih cepat dibandingkan sebelumnya. Kode ditulis oleh programmer untuk memberikan aksi terhadap Event dari pengguna (bagaimana pengguna merespon kepada object yang terdapat di antar muka pengguna).

- 1 September 1992

Microsoft mengumumkan Microsoft Visual Basic for MS-DOS® dalam edisi Standard dan Professional. Seperti Visual Basic untuk Windows, versi ini mengkombinasikan kemudahan perancangan secara grafis dengan kekuatan dan keunggulan dalam banyak hal di pemrograman tradisional.

- 2 November 1992

Microsoft mengumumkan kemampuan dari Visual Basic versi 2.0 untuk Windows pada edisi Professional dan Standard. Versi ini menyertakan lebih dari 300 fitur baru dan peningkatan untuk pengembangan aplikasi yang lebih cepat, pengaksesan ke fitur-fitur tingkat lanjut di Windows, dan produktivitas pengembang yang lebih besar lagi. Di antaranya: MDI Forms, ODBC, dan variabel object.

- 14 Mei 1993

Microsoft mengumumkan Visual Basic versi 3.0 untuk edisi Standard dan Professional. Versi 3.0 menyediakan kemudahan akses ke berbagai sumber data yang banyak dengan mengintegrasikan mesin database Microsoft Access Database for Windows 1.1 dan kemampuan di bidang aplikasi melalui Object Linking and Embedding (OLE) 2.0. Juga ditambahkan tools baru berupa controls baru, penggunaan yang lebih mudah, dan sebuah peningkatan standarisasi control bagi pengguna.

- 29 Juni 1993

Microsoft mengumumkan bahwa Microsoft Visual Basic for Applications (VBA) akan diintegrasikan ke dalam Microsoft Excel 5.0 dan Microsoft Project 4.0. Visual Basic akan melayani bahasa Macro yang umum di versi yang akan datang dari aplikasi-aplikasi untuk Windows dan Macintosh. Fitur Macro yang umum untuk aplikasi dan OLE 2.0, keduanya merupakan elemen penting di visi dan strategi

---



Microsoft untuk aplikasi yang dapat diprogram —yang akan memberikan kemampuan bagi pengguna untuk mengintegrasikan aplikasi-aplikasi, tugas-tugas otomatis, dan pembuatan solusi yang bersifat kostumais. VBA memiliki mesin pengembangan tangguh dan fleksibel yang sama yang membuat Visual Basic menjadi terkenal, teknologi pemrograman yang memperoleh penghargaan.

- 14 November 1994

Visual Basic versi 4.0 untuk Windows didemonstrasikan di Fall/COMDEX '94 di Las Vegas. Demonstrasi ini berfokus pada perannya sebagai aplikasi berbasis Windows 32-bit pertama yang menggunakan kontrol kostumais OLE (OCXs) —komponen perangkat lunak yang dapat digunakan kembali yang ditetapkan dengan spesifikasi OLE. Pada versi ini diperkenalkan: Class, OXC's, dan programmer dapat membuat add-ins sendiri dengan mudah.

- 12 September 1995

Visual Basic versi 4.0 untuk MS-DOS, Microsoft Windows NT®, dan Windows 95 diumumkan. Untuk meningkatkan edisi Standard dan Professional, maka Edisi Enterprise yang baru diperkenalkan untuk memenuhi kebutuhan perusahaan dan team pengembang. Semua versi menyertakan dukungan teknologi OLE yang lebih dikembangkan, perbaikan akses data, sebuah lingkungan pengembangan yang bersifat visual, dan migrasi ke Windows 95 dan Windows NT untuk aplikasi-aplikasi yang sudah ada.

- 7 Desember 1995

Microsoft mengumumkan Microsoft Visual Basic Scripting Edition (VBScript), sebuah bahasa skrip Internet berbasis pada Visual Basic. VBScript merupakan sebuah skrip dengan performansi yang tinggi yang didesain untuk membuat isi yang bersifat aktif pada suatu situs. VBScript membolehkan pengembang untuk mengaitkan dan mengotomatiskan berbagai jenis object di halaman situs, termasuk object OLE. VBScript menjadi sebuah spesifikasi yang diusulkan berlisensi gratis yang tersedia bagi komunitas Internet.

- 3 Februari 1997

Microsoft membuka sistem pemrograman Visual Basic versi 5.0, Edisi Professional — tool versi terakhir yang paling terkenal di dunia untuk membangun aplikasi yang tangguh (Rapid Application Development RAD) tool. Fitur-fitur Visual Basic versi 5.0 terdiri dari sejumlah performansi yang signifikan dan peningkatan produktivitas

---



yang akan membuat para pengembang lebih efisien dan fleksibel serta cepat dalam memberikan solusi bagi pengguna akhir. Fitur-fitur tingkat lanjut — seperti kompilasi berjenis native code, akses database berkecepatan tinggi, dan sebuah lingkungan pengembangan yang ditingkatkan — membuat Visual Basic versi 5.0 menjadi versi yang paling hebat. Sebagai tambahan, kemampuan teknologi pembuatan komponen Microsoft ActiveX® di Visual Basic versi 5.0 akan membolehkan lebih dari 3 juta pengembang menggunakan Visual Basic dalam membuat aplikasi yang berbasis pada komponen yang bertujuan untuk digunakan di Internet, intranets, dan lingkungan tradisional client/server.

- 10 Maret 1997

Versi terakhir yang dirilis dari Visual Basic 5.0, Edisi Pembuatan Kontrol tersedia. Edisi Pembuatan Kontrol adalah cara mudah untuk membuat ActiveX Controls bagi Internet, intranets, dan aplikasi client/server.

- 15 Juni 1998

Microsoft mengumumkan Visual Basic versi 6.0, dan dimasukkan ke dalam Microsoft Visual Studio® versi 6.0. Fitur-fitur Visual Basic versi 6.0 menyediakan pengaksesan data secara terintegrasi dan bersifat grafis ke sumber data (data source) ODBC atau OLE DB manapun, dan perangkat tambahan database yang didisain untuk database Oracle dan Microsoft SQL Server™. Fitur unggulan di versi ini adalah: ActiveX Data Objects (ADO) untuk memanipulasi dan membuat database. Fitur Pengembangan Situs membawa kemudahan dalam penggunaan, model pemrograman berbasis komponen dari Visual Basic untuk membuat HTML – dan Dynamic HTML (DHTML) – berbasis aplikasi. Fitur-fitur baru ini — dikombinasikan dengan optimisasi performansi, pengembangan aplikasi yang disederhanakan dan debugging, dan dukungan untuk Microsoft teknologi server — membuat Visual Basic versi 6.0 sebuah pilihan yang ideal untuk membangun aplikasi berskala perusahaan.

- 2 September 1998

Visual Studio 6.0, solusi tool pengembangan lengkap berskala perusahaan tersedia. Visual Studio 6.0 menyertakan Visual Basic versi 6.0, Microsoft Visual C++®, Microsoft Visual FoxPro®, Microsoft Visual InterDev® Web, dan Microsoft Visual J++®. Bersamaan dengan diluncurkannya Developer Days '98, lebih dari 115 perusahaan papan atas mengumumkan dukungan terhadap Visual Studio 6.0.

---

- 28 April 1999

Lingkungan pengembangan VBA versi 6.0 dan Software Development Kit (SDK) VBA (SDK) versi 6.0 tersedia bagi vendor perangkat lunak pihak ketiga melalui program lisensi VBA. VBA 6.0 merupakan sebuah teknologi pengembangan yang tangguh untuk aplikasi yang dikemas secara komersial dan merupakan sebuah komponen utama di Microsoft Office 2000.

- 15 Februari 2000

Steve Ballmer, presiden yang baru saja ditunjuk dan CEO dari Microsoft Corporation, berbicara kepada lebih dari 2.000 pengembang dalam pidatonya pada Visual Basic Insiders Technical Summit (VBITS), menyatakan visi Microsoft terhadap situs yang dapat diprogram dan memaksa kembali komitmen perusahaan yang dalam terhadap para pengembang Visual Basic.

- 11 Juli 2000

Microsoft menyampaikan Tool Platform .NET untuk XML Web Services. Dalam pidatonya yang ditujukan pada acara Microsoft Professional Developers Conference (PDC) 2000 yang kedelapan, Paul Maritz, wakil presiden kelompok dari Platforms Group di Microsoft, memperlihatkan Microsoft .NET Framework dan Microsoft Visual Studio .NET. Ini merupakan versi terakhir dari perangkat pengembangan yang digunakan di seluruh dunia, Visual Studio .NET menyediakan dukungan untuk pengembangan drag-and-drop terhadap XML Web Services. Secara bersama, kedua produk tersebut menyediakan produktivitas yang tinggi kepada para pengembang, suatu lingkungan yang terdiri dari banyak bahasa untuk pembangunan, pengantaran, dan pengintegrasian XML Web Services pada platform Microsoft .NET.

- 13 November 2000

Microsoft mengumumkan kemampuan Visual Studio .NET versi Beta 1. Dalam pidato acara COMDEX/Fall 2000, Bill Gates mengumumkan kemampuan versi beta pertama Visual Studio.NET dan .NET Framework, dua teknologi kunci untuk memungkinkan para pengembang membangun XML Web services pada platform .NET. XML Web services merupakan aplikasi dan komponen yang dibuat tersedia melewati situs dengan menggunakan XML dan Simple Object Access Protocol (SOAP), dan kunci yang dapat diprogram untuk membangun generasi masa depan

---

Internet. Microsoft membuat Visual Studio .NET Beta 1 dan .NET Framework tersedia bagi jutaan pelanggan dan rekanan industri.

- 13 Februari 2002

Microsoft mengumumkan kemampuan dari Visual Studio .NET versi akhir. Maka, dapat disimpulkan ringkasan versinya sebagai berikut :

1. Visual Basic 1.0 : 1991
2. Visual Basic 2.0 : 1992
3. Visual Basic 3.0 : 1993
4. Visual Basic 4.0 : 1996
5. Visual Basic 5.0 : 1997
6. Visual Basic 6.0 : 1998
7. Visual Basic 7.0 : 2003
8. Visual Basic 8.0 : 2005
9. Visual Basic 9.0 : 2008
10. Visual Basic 10.0 : 2010
11. Hingga Sekarang versi terbaru visual basic terus dikembangkan

## 2.6. Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup compiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

---



Gambar 2.14 Logo Visual Studio

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

## 2.7. Bluetooth

### 2.7.1. Pengertian Bluetooth

Bluetooth adalah sebuah teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frequency hopping tranceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas.

Bluetooth sendiri dapat berupa card yang bentuk dan fungsinya hampir sama dengan card yang digunakan untuk wireless local area network (WLAN) dimana menggunakan frekuensi radio standar IEEE 802.11, hanya saja pada bluetooth mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah.

---



Gambar 2.15 Logo Bluetooth

Pada dasarnya bluetooth diciptakan bukan hanya menggantikan atau menghilangkan penggunaan kabel didalam melakukan pertukaran informasi, tetapi juga mampu menawarkan fitur yang baik untuk teknologi mobile wireless dengan biaya yang relatif rendah, konsumsi daya yang rendah, interoperability yang menjanjikan, mudah dalam pengoperasian dan mampu menyediakan layanan yang bermacam-macam.

#### 2.7.2. Perkembangan Sejarah Bluetooth

Nama bluetooth berawal dari proyek prestisius yang dipromotori oleh perusahaan-perusahaan raksasa internasional yang bergerak di bidang telekomunikasi dan komputer, di antaranya Ericsson, IBM, Intel, Nokia, dan Toshiba.

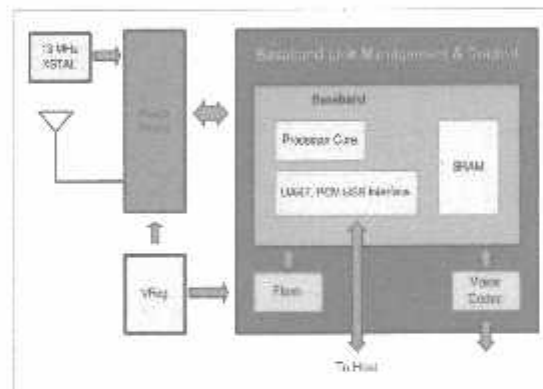
Proyek ini di awal tahun 1998 dengan kode nama bluetooth, karena terinspirasi oleh seorang raja Viking (Denmark) yang bernama Harald Blatand. Raja Harald Blatand ini berkuasa pada abad ke-10 dengan menguasai sebagian besar daerah Denmark dan daerah Skandinavia pada masa itu. Dikarenakan daerah kekuasaannya yang luas, raja Harald Blatand ini membiayai para ilmuwan dan insinyur untuk membangun sebuah proyek berteknologi metamorfosis yang bertujuan untuk mengontrol pasukan dari suku-suku di daerah Skandinavia tersebut dari jarak jauh. Maka untuk menghormati ide raja Viking tersebut, yaitu Blatand yang berarti bluetooth (dalam bahasa Inggris) proyek ini diberi nama.

#### 2.7.3. Cara Kerja Bluetooth

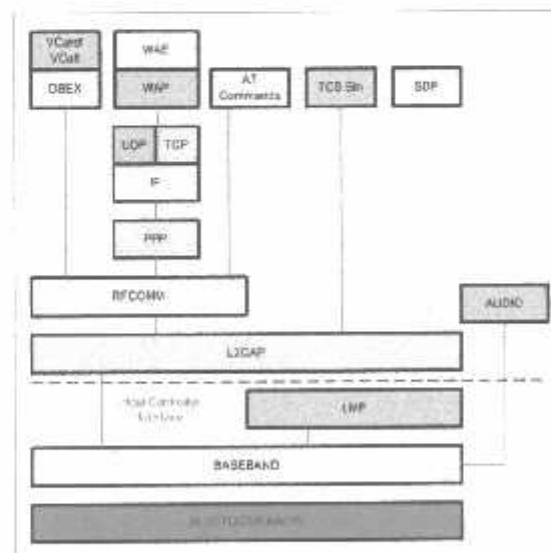
Protokol bluetooth menggunakan sebuah kombinasi antara circuit switching dan packet switching. Bluetooth dapat mendukung sebuah kanal data asinkron, tiga kanal suara sinkron simultan atau sebuah kanal dimana secara bersamaan mendukung layanan data asinkron dan suara sinkron. Setiap kanal suara mendukung sebuah kanal suara sinkron 64 kb/s. Kanal asinkron dapat mendukung kecepatan maksimal 723,2 kb/s

asimetris, dimana untuk arah sebaliknya dapat mendukung sampai dengan kecepatan 57,6 kb/s. Sedangkan untuk mode simetris dapat mendukung sampai dengan kecepatan 433,9 kb/s.

Sebuah perangkat yang memiliki teknologi wireless bluetooth akan mempunyai kemampuan untuk melakukan pertukaran informasi dengan jarak jangkauan sampai dengan 10 meter (~30 feet), bahkan untuk daya kelas 1 bisa sampai pada jarak 100 meter. Sistem bluetooth terdiri dari sebuah radio transceiver, baseband link Management dan Control, Baseband (processor core, SRAM, UART, PCM USB Interface), flash dan voice code. sebuah link manager. Baseband link controller menghubungkan perangkat keras radio ke baseband processing dan layer protokol fisik. Link manager melakukan aktivitas-aktivitas protokol tingkat tinggi seperti melakukan link setup, autentikasi dan konfigurasi. Secara umum blok fungsional pada sistem bluetooth secara umum dapat dilihat pada Gambar dibawah ini.



Gambar 2.16. Blok fungsional sistem bluetooth.



Gambar 2.17. Layer-layer pada sistem bluetooth.

Tiga buah lapisan fisik yang sangat penting dalam protokol arsitektur Bluetooth ini adalah :

1. Bluetooth radio, adalah lapis terendah dari spesifikasi Bluetooth. Lapis ini mendefinisikan persyaratan yang harus dipenuhi oleh perangkat tranceiver yang beroperasi pada frekuensi 2,4 GHz ISM.
  2. Baseband, lapis yang memungkinkan hubungan RF terjadi antara beberapa unit Bluetooth membentuk piconet. Sistem RF dari bluetooth ini menggunakan frekuensi-hopping-spread spectrum yang mengirimkan data dalam bentuk paket pada time slot dan frekuensi yang telah ditentukan, lapis ini melakukan prosedur pemeriksaan dan paging untuk sinkronisasi transmisi frekuensi hopping dan clock dari perangkat bluetooth yang berbeda.
  3. LMP, Link Manager Protocol, bertanggung jawab terhadap link set-up antar perangkat Bluetooth. Hal ini termasuk aspek securiti seperti autentifikasi dan enkripsi dengan pembangkitan, penukaran dan pemeriksaan ukuran paket dari lapis baseband.
-



## **BAB III**

### **PERANCANGAN DAN DESAIN APLIKASI**

#### **3.1. Analisa Sistem**

Aplikasi yang di buat diharapkan memiliki kemampuan untuk mengontrol sebuah robot beroda dengan memanfaatkan teknologi *augmented reality* dan juga menggunakan komunikasi Bluetooth yang diharapkan dapat dikontrol dari jarak jauh. Dan juga diharapkan dapat menggantikan sensor *accelerometer* pada smartphone sebagai controller robot.

#### **3.2. Deskripsi Umum Sistem**

Aplikasi pemanfaatan *augmented reality* sebagai pengontrol robot adalah aplikasi yang dibangun untuk mempermudah melakukan pengontrolan sebuah robot beroda dengan memanfaatkan teknologi *augmented reality*, dan juga dapat menggantikan sensor *accelerometer* pada smartphone yang digunakan sebagai controller robot.

Aplikasi berbasis desktop ini ketika dijalankan secara otomatis akan mendeteksi *Capture device* dan menyalakannya. *Capture device* ( *webcam* ) berfungsi sebagai pembaca marker. Saat marker telah terdeteksi oleh capture device, program akan menyesuaikan dengan markeryang telah tersimpan pada library dengan bantuan marker generator. Jika pola marker sesuai secara otomatis akan menampilkan objek 3D.

Selanjutnya ketika marker digerakan kearah tertentu akan diperoleh koordinat 3D, hasil pergerakan marker yang berupa koordinat 3D ini akan diolah oleh program dan diterjemahkan dalam bentuk character. Character inilah yang nantinya akan dikirim ke mikrokontroler yang terdapat pada robot melalui Bluetooth. Mikrokontroller akan memberikan perintah gerak kepada motor servo sesuai dengan arah gerak yang diterima.

#### **3.3. Analisa Kebutuhan Sistem**

Analisa kebutuhan menjadi dasar penentuan komponen, fasilitas, dan solusi yang akan diimplementasikan di dalam aplikasi ini. Adapun kebutuhan perangkat-perangkat pendukung terbagi menjadi dua kategori, yakni kebutuhan akan perangkat lunak dan perangkat keras.

3.3.1. Kebutuhan Perangkat Lunak

Tabel 3.1 menunjukan perangkat lunak yang digunakan dalam proses pembuatan aplikasi ini.

Tabel 3.1 Spesifikasi Perangkat Lunak

No	Nama Kebutuhan	Spesifikasi	Keterangan
1	Sistem Operasi	Windows Xp	
2	Accelkit		Source untuk memperoleh koordinat 3D dari pergerakan marker
3	Visual Studio 9	2008	Untuk pembuatan aplikasi

3.3.2. Kebutuhan Perangkat Keras

Adapun kebutuhan perangkat keras dalam proses pembuatan aplikasi, seperti yang ditunjukkan pada tabel 3.2 berikut ini :

Tabel 3.2 Spesifikasi Perangkat Keras

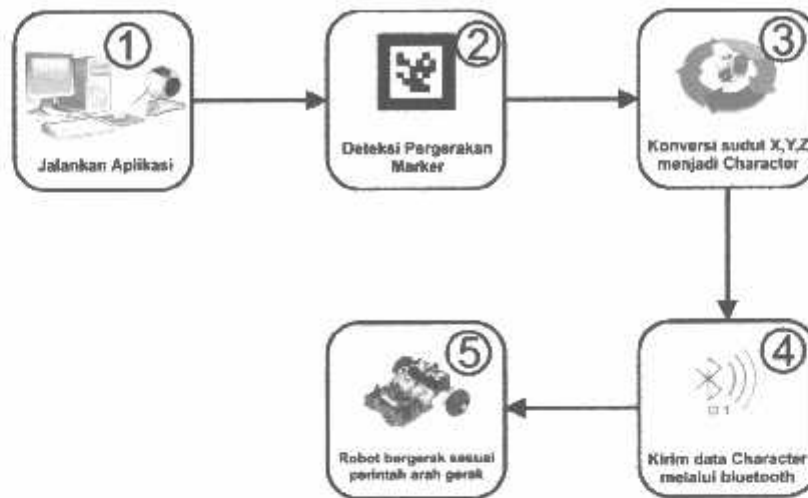
No	Nama Kebutuhan	Spesifikasi	Keterangan
1	Komputer/Laptop	Dengan prosesor setara Intel Dual Core	Untuk menjalankan aplikasi
2	Kamera (webcam)	0.3 MP (setara kualitas VGA)	Sebagai sensor input yang menangkap marker
3	Printer	Deskjet/Inkjet	Untuk mencetak pola marker
4	Marker	Berukuran 40mmx40mm	Sebagai patokan bagi aplikasi
5	Bluetooth	2.4 Ghz	Sebagai jalur komunikasi

3.4. Perancangan Aplikasi

Perancangan aplikasi adalah tahap penyusunan struktur, alur kerja, dan bagaimana interaksi antar komponen-komponen di dalam aplikasi pengontrol robot. Tahapan ini terbagi menjadi dua, yakni perancangan diagram blok dan diagram alir (flowchart).

### 3.4.1. Diagram Blok

Diagram blok dibuat untuk mempermudah pemahaman terhadap cara kerja aplikasi dan interaksi antar komponen-komponen yang terlibat.



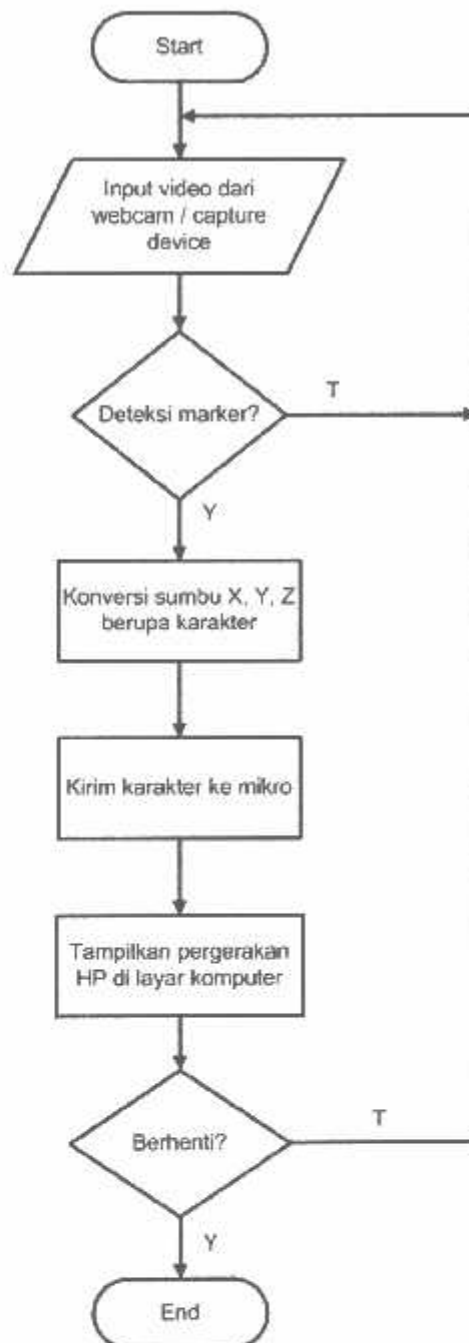
Gambar 3.1 Diagram Blok

Seperti yang terlihat pada gambar 3.1, user mengakses aplikasi melalui komputer atau laptop yang sudah terdapat kamera (webcam). Kamera (webcam) berfungsi menangkap pola marker. Aplikasi akan melakukan render objek 3D.

Saat user menggerakkan marker akan diperoleh sudut X,Y,Z. Nilai sudut ini akan dikonversikan atau diproses oleh program menjadi sebuah karakter, dan katakter inilah yang akan dikirim oleh aplikasi melalui bluetooth ke mikrokontroler. Saat data diterima oleh mikrokontroler, data tersebut akan dicocokkan dengan data karakter yang telah diprogram pada mikrokontroler sebagai arah gerak. Jika data cocok, maka secara otomatis robot akan bergerak sesuai arah gerak yang dikirim.

### 3.4.2. Diagram Alir

Untuk memperjelas aplikasi, maka dibuat diagram alir seperti gambar 3.2 dibawah ini :



Gambar 3.2 Diagram Alir ( Flowchart )

Seperti yang terlihat pada gambar 3.2 saat aplikasi dijalankan, secara otomatis mendeteksi *Capture device* (webcam) dan menyalakannya, dan menerima input berupa pola marker. Saat marker terdeteksi secara otomatis me-render objek 2D atau 3D. Pada saat marker digerakkan akan memperoleh koordinat X,Y,Z, lalu dikonversikan ke dalam bentuk karakter huruf. Data yang telah diterjemahkan akan dikirim ke mikrokontroller melalui bluetooth.

### 3.5. Perancangan Tata Letak Antarmuka

#### 3.5.1. Tampilan Antarmuka Video Capture

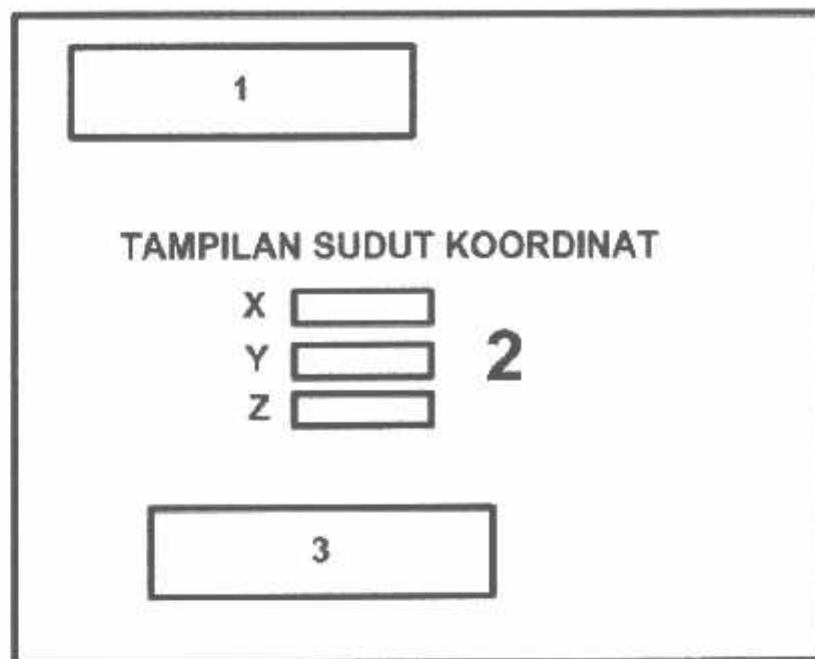
Pada halaman ini adalah halaman yang berfungsi sebagai sensor input aplikasi. halaman ini menjadi sangat penting karena dihalaman ini akan mendeteksi marker yang muncul di depan *Capture device* (webcam). Rancangan tata letak video capture ditunjukkan pada gambar 3.3 di bawah ini.



Gambar 3.3 tata letak halaman video capture

#### 3.5.2. Tampilan Antarmuka Menu Utama

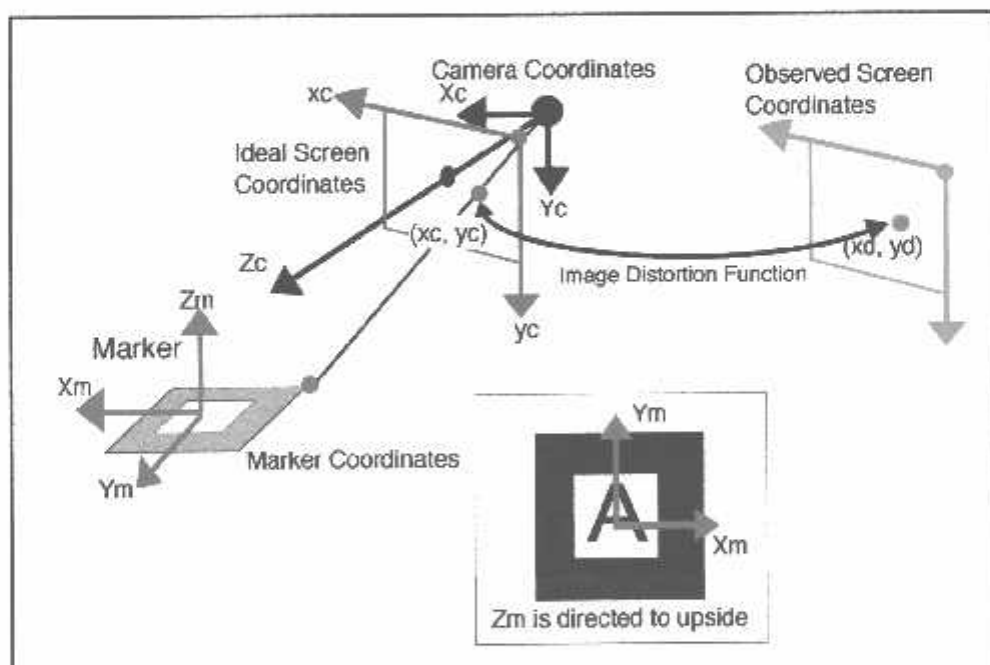
Pada halaman yang ditunjukkan gambar 3.4 merupakan tampilan menu utama. Pada halaman ini berisi status koneksi dengan bluetooth, menampilkan sudut X,Y dan Z yang dihasilkan dari pergerakan marker, dan juga menampilkan karakter yang akan dikirim beserta perintah arah gerak.



Gambar 3.4 tata letak halaman utama

### 3.6. Perancangan Proses Deteksi Marker

#### 3.6.1. Sistem Koordinat AR



Gambar 3.5 Sistem koordinat kamera terhadap marker

Pada gambar 3.5 merupakan sistem koordinat kamera terhadap posisi marker atau penanda saat berada di depan kamera (web cam).

Dalam menampilkan objek 3D yang sesuai dengan posisi dan orientasi marker, perlu diperhitungkan hasil proyeksi yang diterima *viewplane* (bidang proyeksi di layar) untuk kemudian ditampilkan. Menurut [4] selain proyeksi pada bidang 2D, dalam pergeseran *marker* maupun kamera perlu diperhatikan perubahan posisi dan rotasi dalam sistem koordinat 3D. Posisi dan orientasi dari *marker* didapat dari hasil *tracking marker* yang ditransformasi dengan operasi translasi dan rotasi, sedangkan posisi dan orientasi yang ada pada proyeksi di layar didapat dari perhitungan transformasi proyeksi perspektif.

a. Transformasi Translasi

$$\begin{bmatrix} Xc' \\ Yc' \\ Zc' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Tx \\ Ty \\ Tz \\ 1 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix} \quad (1)$$

b. Transformasi Rotasi

$$\begin{bmatrix} Xc' \\ Yc' \\ Zc' \\ 1 \end{bmatrix} = \begin{bmatrix} R11 & R12 & R13 & 0 \\ R21 & R22 & R23 & 0 \\ R31 & R32 & R33 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix} \quad (2)$$

c. Transformasi Proyeksi Perspektif

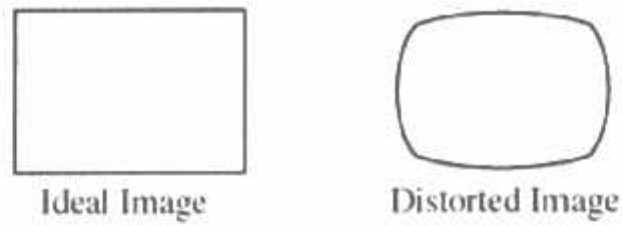
$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} fc & 0 & u & 0 \\ 0 & fc & v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix} \quad (3)$$

d. Transformasi Objek pada Sistem AR

$$\begin{bmatrix} Ox \\ Oy \\ Oz \\ 1 \end{bmatrix} = \begin{bmatrix} fc & 0 & u & 0 \\ 0 & fc & v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R11 & R12 & R13 & Tx \\ R21 & R22 & R23 & Ty \\ R31 & R32 & R33 & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Cx \\ Cy \\ Cz \\ 1 \end{bmatrix} \quad (4)$$

### 3.6.2. Kalibrasi Kamera

Tujuan dari kalibrasi kamera adalah untuk menghitung tingkat distorsi dari sebuah lensa kamera yang digunakan agar gambar yang dihasilkan mendekati gambar yang ideal. Inilah yang menjadi acuan nantinya dalam perhitungan pada proses sistem koordinat.



Gambar 3.6 Perbandingan antara gambar ideal dengan gambar yang disebabkan faktor distorsi

Berikut adalah persamaan hubungan antara Ideal and Observed Screen Coordinates.

$$d^2 = (x_I - x_0)^2 + (y_I - y_0)^2$$

$$p = \{1 - fd^2\}$$

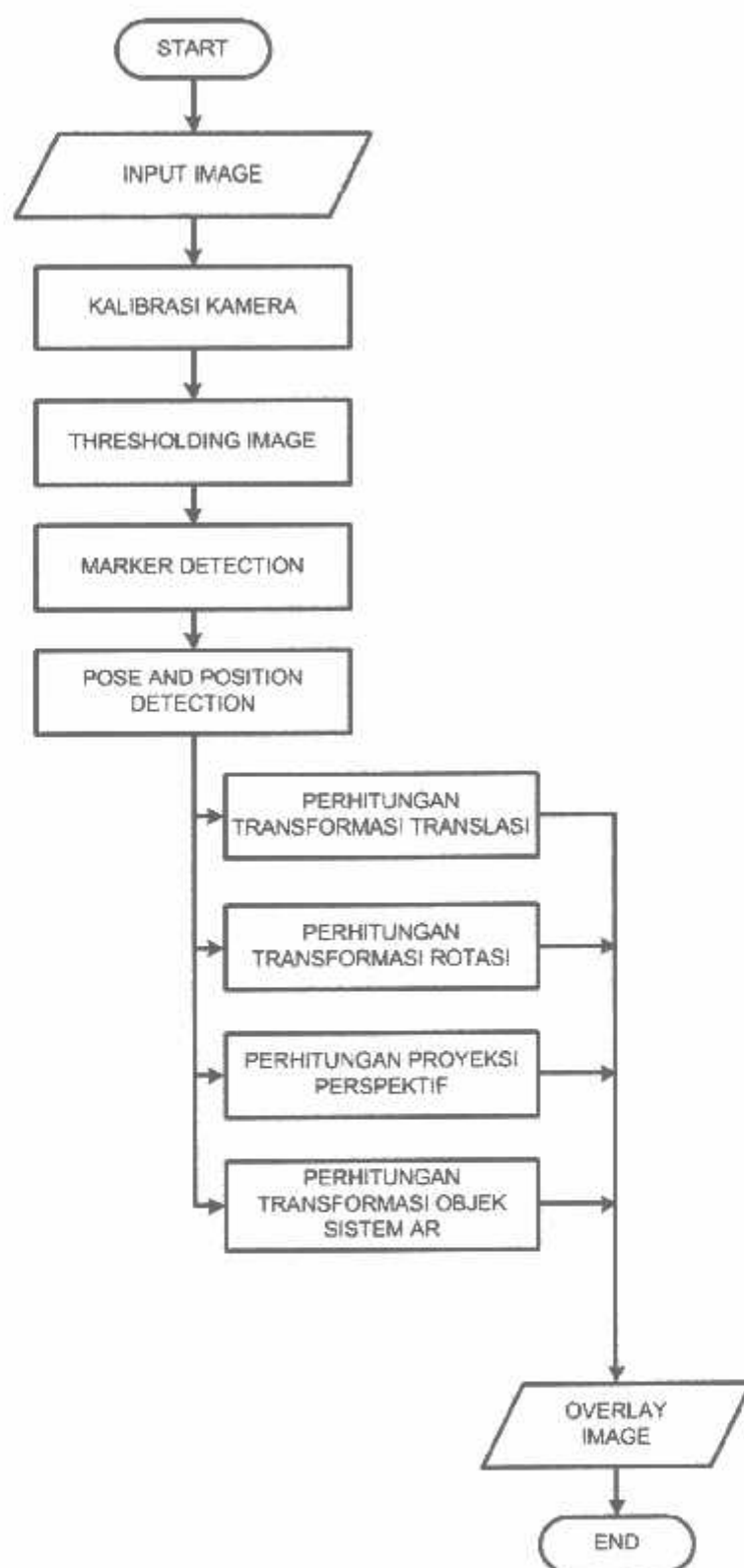
$$x_O = p(x_I - x_0) + x_0, \quad y_O = p(y_I - y_0) + y_0$$

$(x_0, y_0)$  : Center Coordinates of Distortion

$f$ : Distortion Factor



### 3.6.3. Flowchart Sistem Deteksi Marker



Gambar 3.7 Diagram Alir Proses Deteksi Marker

Pada gambar 3.7 merupakan proses deteksi marker untuk memperoleh nilai sudut koordinat 3D yaitu sumbu x, y dan z. Langkah pertama inputkan image atau dalam hal ini kita menggunakan marker, selanjutnya sistem akan melakukan kalibrasi kamera untuk menghitung tingkat distorsi dari sebuah kamera. Setelah 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, dan menghilangkan area yang tidak berbentuk segi empat, sehingga yang akan ditampilkan pada layar hanyalah area yang memiliki segi empat. Langkah selanjutnya adalah pose and position detection, pada proses ini terjadi perhitungan transformasi translasi, rotasi, proyeksi perspektif dan transformasi objek sistem AR. Langkah terakhir adalah menampilkan objek 2D atau 3D di atas marker.

---

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1. Implementasi**

Tahap implementasi merupakan proses perubahan rancangan dan spesifikasi yang telah disusun sebelumnya menjadi suatu aplikasi yang siap untuk dijalankan. Implementasi aplikasi pengontrol robot berbasis augmented reality dilakukan sesuai tujuan awal yakni pemanfaatan augmented reality sebagai pengontrol robot dan juga menjadi alat bantu riset bagi tim robot elektro ITN Malang. Proses implementasi terbagi menjadi dua bagian, yakni implementasi pembaca pola atau marker dan program pengirim perintah gerak

Pembuatan aplikasi pembaca marker menggunakan source code yang sudah ada, yakni menggunakan accelkit yang dikembangkan dari library-library ARToolkit dengan menggunakan bahasa C/C++.

Aplikasi utama yakni aplikasi yang bertugas sebagai pengirim data arah gerak kepada mikrokontroler melalui Bluetooth dibuat dengan menggunakan Visual Basic (VB) karena program ini berbasis desktop.

#### **4.2. Pengujian**

Pengujian adalah tahap uji coba beberapa komponen vital dalam system agar dapat diambil kesimpulan, apakah aplikasi ini sudah mencapai tujuan yang diharapkan atau tidak. Pengujian dilakukan terhadap beberapa komponen utama system, yang terkait langsung dengan jalannya aplikasi inti pengontrol robot berbasis augmented reality.

##### **4.2.1. Pengujian spesifikasi**

Pada pengujian spesifikasi terbagi menjadi dua bagian, yaitu spesifikasi perangkat keras dan perangkat lunak :

#### 4.2.1.1. Spesifikasi Perangkat Keras

Pada tabel 4.1 menunjukkan spesifikasi perangkat keras yang digunakan untuk melakukan pengujian aplikasi pengontrol robot berbasis augmented reality.

Tabel 4.1 spesifikasi perangkat keras untuk pengujian

No	Nama Kebutuhan	Spesifikasi	Keterangan
1	Komputer/Laptop	Dengan prosesor setara Intel Dual Core	Untuk menjalankan aplikasi
2	Kamera (webcam)	0.3 MP (setara kualitas VGA)	Sebagai sensor input yang menangkap marker
4	Marker	Berukuran 40mmx40mm	Sebagai patokan bagi aplikasi
5	Bluetooth	2.4 Ghz	Sebagai jalur komunikasi

#### 4.2.1.2. Spesifikasi Perangkat Lunak

Pada tabel 4.2 menunjukkan spesifikasi perangkat lunak yang digunakan dalam melakukan pengujian aplikasi pengontrol robot berbasis augmented reality.

Tabel 4.2 spesifikasi perangkat lunak untuk pengujian

No	Nama Kebutuhan	Spesifikasi	Keterangan
1	Sistem Operasi	Windows Xp	
2	Accelkit		Source untuk memperoleh koordinat 3D dari pergerakan marker
3	Visual Basic	2008	Untuk pembuatan aplikasi

#### 4.2.2. Pengujian Bluetooth

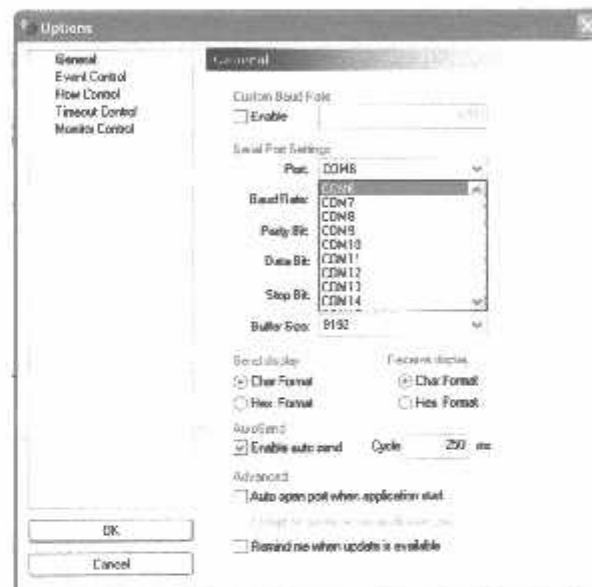
Pengujian Bluetooth dilakukan dengan mencoba komunikasi Bluetooth slave ke laptop atau computer. Pengujian dilakukan dengan menggunakan AccessPort.

- Sebelumnya koneksikan terlebih dahulu Bluetooth slave dengan laptop.
- Jalankan AccessPort
- Selanjutnya akan muncul tampilan jendela dari accessport seperti gambar 4.1 dibawah ini.



Gambar 4.1 jendela accessport

- Selanjutnya melakukan konfigurasi dengan klik icon pada pojok kiri atas
- Maka akan muncul jendela seperti gambar 4.2 dibawah ini.



Gambar 4.2 jendela konfigurasi accessport

- Pada gambar 4.2 diatas, pilih General, lalu pada bagian Port pilih COM6, COM6 didapat saat kita melakukan koneksi Bluetooth slave dengan laptop/komputer.
- Untuk baudratanya setting pada nilai 115200 sesuai dengan spesifikasi Bluetooth slavenya.
- Langkah selanjutnya mengirim karakter "aa;" lalu tekan send maka status TX akan menampilkan data yang dikirim. Seperti yang ditunjukkan pada gambar 4.3 dibawah ini.



Gambar 4.3 pengujian pengiriman data melalui Bluetooth

Pada gambar 4.3 di atas nilai RX tidak mengalami perubahan dikarenakan Bluetooth slave hanya menerima data dan tidak ada data balikan yang dikirim.

#### 4.2.3. Pengujian Aplikasi

Pengujian aplikasi bertujuan untuk mengetahui kinerja system secara keseluruhan apakah sesuai dengan rancangan dan spesifikasi yang telah direncanakan.

##### 4.2.3.1. Pengujian Arah Pergerakan Robot

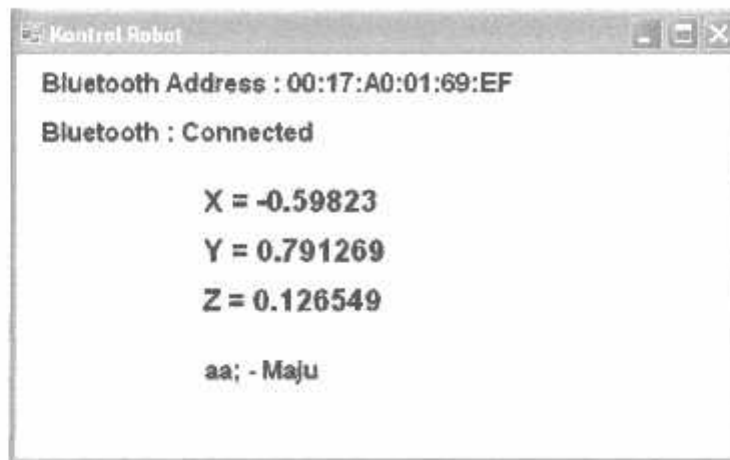
Pengujian arah pergerakan robot dilakukan untuk mengetahui arah pergerakan robot apakah sesuai dengan perubahan marker.

Pengujian pertama dilakukan dengan menggerakkan robot ke arah maju. Seperti yang ditunjukkan pada gambar 4.4 dan 4.5.



Gambar 4.4 posisi marker saat bergerak maju

Gambar 4.4 memperlihatkan posisi marker saat melakukan pengontrolan robot ke arah depan atau berjalan maju. Sedangkan pada gambar 4.5 memperlihatkan koordinat sumbu X, Y dan Z yang dihasilkan dari posisi marker pada gambar 4.4. Data yang dikirim berupa karakter aa;



Gambar 4.5 koordinat sumbu X,Y dan Z saat bergerak maju

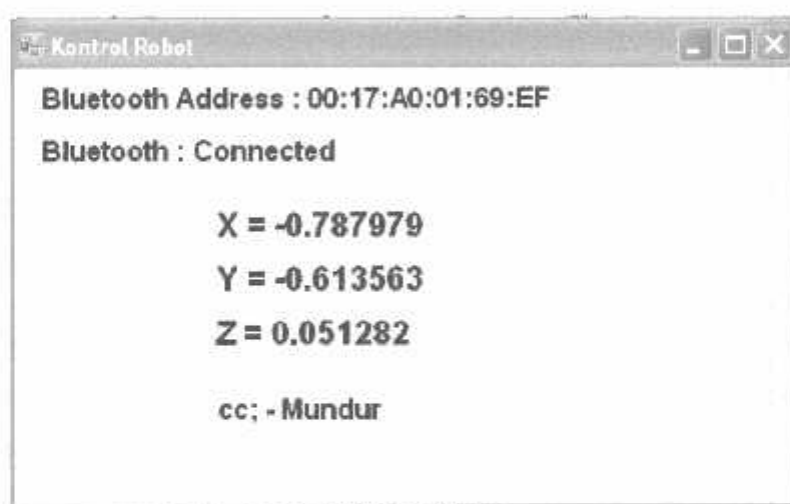


Gambar 4.6 Robot bergerak maju

Pengujian kedua dilanjutkan dengan menggerakkan robot ke arah belakang atau mundur. Seperti yang ditunjukkan gambar 4.7 memperlihatkan posisi marker saat ingin melakukan pengontrolan robot. Dan dari posisi marker tersebut di dapat nilai koordinat sumbu X, Y dan Z seperti yang ditunjukkan pada gambar 4.8, dengan data karakter yang dikirim adalah cc;



Gambar 4.7 Posisi marker saat bergerak mundur



Gambar 4.8 koordinat sumbu X, Y dan Z saat bergerak mundur



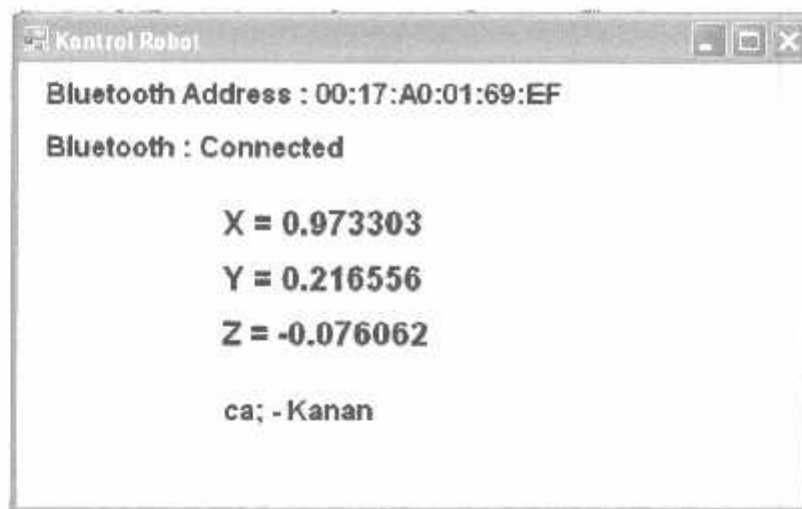


Gambar 4.9 Robot bergerak mundur

Pengujian ketiga adalah pengujian arah gerak kanan. Pada gambar 4.10 menunjukkan posisi marker saat ingin menggerakkan robot ke kanan. Koordinat sumbu X, Y dan Z yang dihasilkan dari posisi marker tersebut dapat dilihat pada gambar 4.11 karakter yang dikirim adalah ca;.



Gambar 4.10 posisi marker saat bergerak ke kanan



Gambar 4.11 koordinat sumbu X, Y dan Z saat bergerak ke kanan

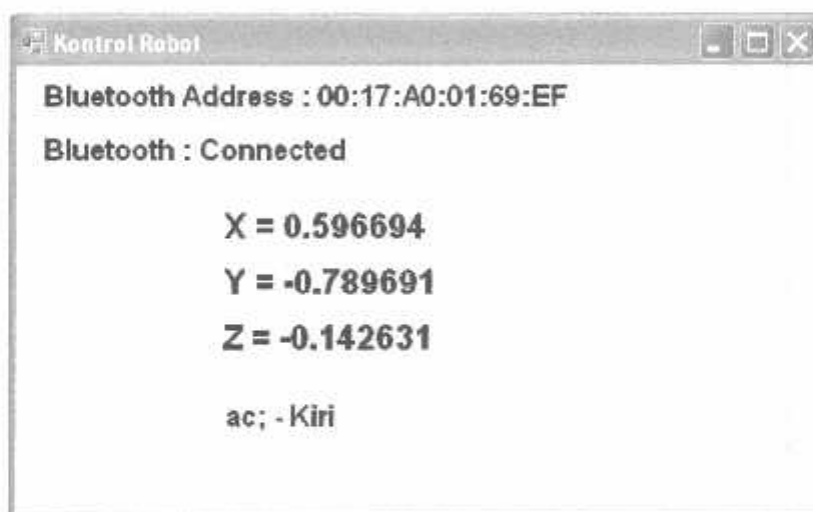


Gambar 4.12 Robot bergerak ke kanan

Pengujian terakhir yang dilakukan adalah menggerakkan robot ke kiri ( belok kiri ). Seperti pada gambar 4.13 yang menunjukkan posisi marker. Sedangkan pada pada gambar 4.14 memperlihatkan koordinat sumbu X, Y dan Z dengan data karakter yang dikirim adalah ac;.



Gambar 4.13 posisi marker saat berkerak ke kiri



Gambar 4.14 koordinat sumbu X, Y dan Z saat bergerak ke kiri



Gambar 4.15 Robot bergerak ke kiri

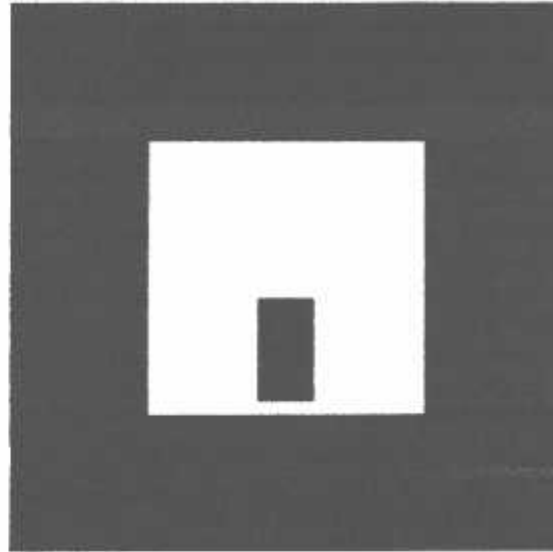
Dari hasil pengujian arah gerak robot ditunjukkan pada tabel 4.3 berikut ini.

Tabel 4.3 hasil pengujian arah gerak robot

No	Arah Gerak	Nilai Koordinat X, Y	Parameter
1	Maju	$X < 0 ; Y > 0$	aa;
2	Mundur	$X < 0 ; Y < 0$	cc;
3	Kanan	$X > 0 ; Y > 0$	ca;
4	Kiri	$X > 0 ; Y < 0$	ac;
5	Berhenti	Saat marker tidak terdeteksi oleh kamera	

4.2.4. Pengujian Deteksi Marker

Pengujian deteksi marker merupakan proses yang sangat vitak di dalam sebuah aplikasi berbasis augmented reality. Proses ini menentukan apakah aplikasi akan dapat koordinat sumbu X, Y, Z dari pergerakan marker. Berdasarkan hal tersebut, maka perlu dilakukan pengujian untuk melihat sejauh mana aplikasi dapat mendeteksi marker dengan baik. Pola marker yang digunakan dalam pengujian seperti terlihat dalam gambar 4.16.



Gambar 4.16 pola marker yang digunakan

#### **4.2.4.1. Deteksi Marker Pada Intensitas Cahaya Tertentu & Pengaruh Arah Pergerakan Robot**

Pengujian pertama yakni pada intensitas cahaya yang beragam. Hal ini dilakukan karena kemungkinan saat melakukan pengontrolan robot pada tempat yang berbeda-beda dan dalam waktu yang berbeda pula. Pengujian dilakukan pada pagi, siang dan malam hari untuk melihat perbedaan intensitas cahaya di sekitar kamera.

Pengujian pagi dengan intensitas cahaya yang tidak terlalu terang seperti terlihat pada gambar 4.17 dibawah ini menunjukkan bahwa marker masih dapat dideteksi namun tidak stabil karena aplikasi kadang tidak dapat membaca marker.



Gambar 4.17 pengujian intensitas cahaya sedang (pagi)

Pengujian berikutnya dilakukan pada siang hari, dimana intensitas cahaya sangat cukup terang. Hasil yang didapat sangat memuaskan karena dalam pengujian aplikasi selalu dapat mendeteksi keberadaan marker dengan tepat seperti ditunjukkan pada gambar 4.18 dibawah ini.



Gambar 4.18 pengujian intensitas cahaya baik ( siang hari )

Pengujian terakhir dilakukan pada malam hari, dimana intensitas cahaya rendah dengan sumber cahaya utama dari lampu kamar dan cahaya dari layar pada laptop. Dari hasil pengujian diperoleh bahwa aplikasi masih dapat membaca marker, namun terkadang aplikasi tidak dapat membaca marker pada posisi kemiringan tertentu dan pada jarak tertentu. Seperti yang ditunjukkan pada gambar 4.19 dibawah ini.



Gambar 4.19 pengujian intensitas cahaya kurang ( malam hari )

Hasil pengujian terhadap intensitas cahaya yang berbeda ditunjukkan dalam tabel 4.4 berikut.

Tabel 4.4 hasil pengujian deteksi marker pada intensitas cahaya tertentu

No	Waktu Pengujian/Intensitas	Hasil Pendeteksian	Hasil pergerakan robot
1	Pagi / intensitas cahaya cukup	Cukup, kurang stabil	Cukup baik
2	Siang / intensitas cahaya baik	Sangat baik	Baik
3	Malam hari Intensitas cahaya kurang	Tidak stabil	Kurang baik

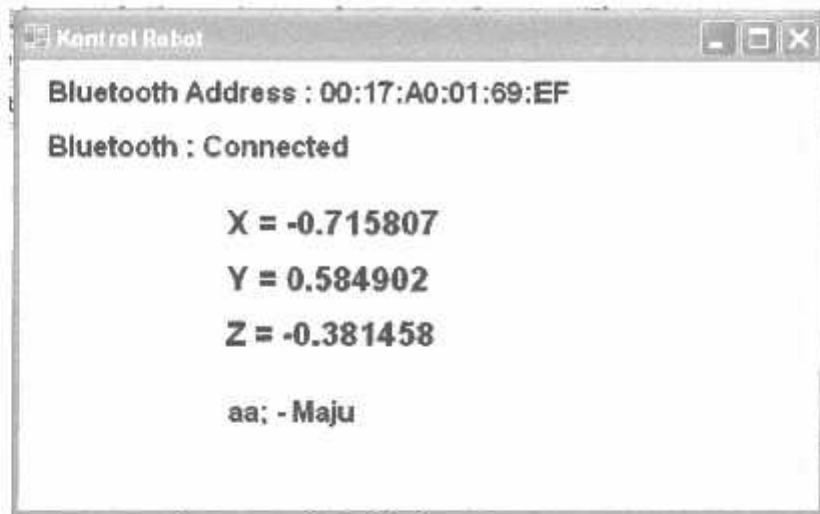
**4.2.4.2. Deteksi Marker Pada Jarak Tertentu & Pengaruh Arah Pergerakan Robot.**

Jarak merupakan factor lain yang berpengaruh langsung terhadap proses deteksi marker. Semakin jauh jarak marker terhadap kamera, semakin kecil ukuran dan kualitas pola marker yang diekstrak oleh aplikasi. Pengujian jarak ini bertujuan untuk mengetahui rentang jarak yang optimal agar marker dapat terdeteksi dengan baik.

Pengujian dilakukan terhadap 3 rentang jarak, yakni dekat (0-30 cm), sedang (31-60 cm), jauh (61-90 cm). Dalam gambar 4.20 ditunjukkan pengujian rentang jarak dekat yakni  $\leq 30$  cm dan dilihat bahwa marker dideteksi dengan baik. Dan pada gambar 4.21 terlihat hasil koordinat sumbu X, Y dan Z yang dihasilkan.



Gambar 4.20 posisi marker dengan jarak dekat



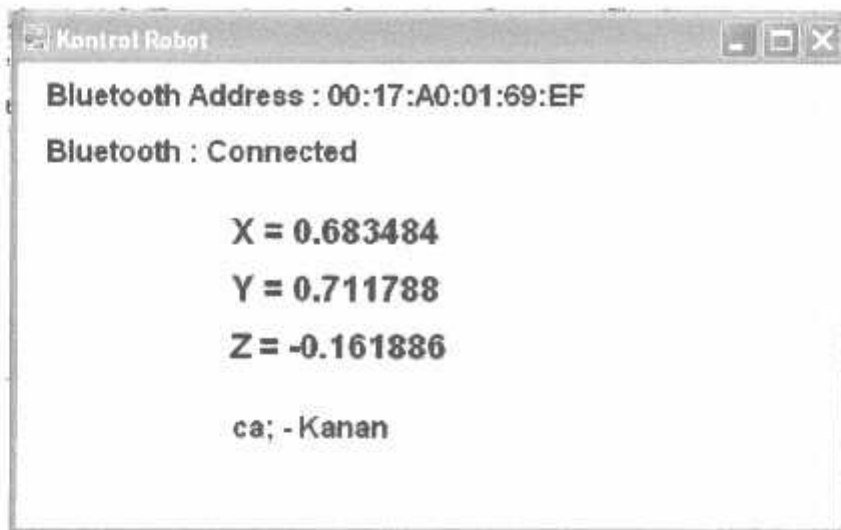
Gambar 4.21 hasil koordinat sumbu X, Y, Z saat jarak dekat

Pengujian berikutnya yakni pada rentang jarak sedang, yaitu antara 31-60 cm. pada pengujian ini terlihat pada gambar 4.22, marker dapat dideteksi dengan cukup baik seperti halnya saat jarak dekat. Dan pada gambar 4.23 menunjukkan hasil koordinat sumbu X, Y dan Z yang dihasilkan.



Gambar 4.22 posisi marker saat jarak sedang





Gambar 4.23 hasil koordinat sumbu X, Y, Z saat jarak sedang

Pengujian jarak terakhir yaitu rentang jarak jauh antara 61-90 cm. pada jarak 90 cm, marker tidak dapat terdeteksi dengan baik. Terkadang kamera tidak bias menangkap pola marker saat marker digerakkan terlalu cepat. Akan tetapi masih bisa melakukan pengontrolan jika marker digerakan dengan cukup pelan. Seperti yang ditunjukkan pada gambar 4.24 dan gambar 4.24 dibawah ini.



Gambar 4.24 posisi marker saat jarak jauh

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

1. Konsep *augmented reality* dapat diterapkan sebagai pengontrol sebuah robot beroda.
2. Pada kondisi kekurangan cahaya ( malam hari / lampu ruangan ) dan dengan rentang jarak yang cukup jauh antara marker dengan kamera aplikasi sulit melakukan deteksi marker dengan baik.
3. Data yang dikirim berupa karakter, sehingga pergerakan kecepatan robot masih sulit untuk di kontrol.
4. Marker dicetak pada kertas yang tebal agar lebih mudah saat mendeteksi.
5. Marker yang berukuran 40mmx40mm, jarak optimal agar aplikasi pengontrol robot dapat mendeteksi marker dengan baik yakni antara 15- 60 cm dari posisi kamera / webcam.

#### **5.2. Saran**

Berdasarkan kesimpulan di atas, sangat mungkin untuk dilakukan perkembangan ke depan terhadap aplikasi pengontrol robot berbasis android. Berikut ini saran-saran untuk pengembangan aplikasi, antara lain :

1. Menggunakan kamera dengan resolusi yang lebih tinggi sehingga dapat menangkap pola marker lebih baik.
2. Pengiriman data tidak hanya dalam bentuk karakter, melainkan hasil dari koordinat sumbu X, Y dan Z agar dapat mengendalikan kecepatan pergerakan robot.

## DAFTAR PUSTAKA

- [1]. Afissunani, Akhmad, Akuwan Saleh, M. Hasbi Assidiqi. *Multi Marker Augmented Reality Untuk Aplikasi Magic Book*. Surabaya
  - [2]. Andriatma, Ryza. 2013. *Rancang Bangun Sistem Pengontrolan Robot Dengan Bluetooth dan Sensor Accelerometer Pada Smarthphone Android*. Malang.
  - [3]. Anharku. (2009). Bluetooth. *Bluetooth*.
  - [4]. Budi Raharja, I Wayan. 2012. *Rancang Bangun Aplikasi "Virtual Fitting Room" Pada Toko Baju Online Dengan Konsep Augmented Reality Menggunakan Flartoolkit*. Malang
  - [5]. Danto, Walesa, Agung Toto Wibowo, Bedy Purnama. *Analisa Metode Occlusion Based pada Augmented Reality Studi Kasus : Interaksi dengan Objek Virtual Secara Real Time Menggunakan Gerakan Marker*. Bandung
  - [6]. Kato. H., Billinghurst, M. Poupyrev, I. Tetsutani, N. dan Tachibana, K. 2001. *Tangible Augmented Reality*. Nagoya, Japan : *Proceedings of Nicograph 2001*.
  - [7]. Nkast, Situs Blog, <http://blog.nkast.gr/post/2010/09/19/AccelKit-an-Augmented-Reality-Accelerometer-Kit-for-Windows-Phone-7-Emulator.aspx>, 8 April 2013
  - [8]. <http://blog.unsri.ac.id/userfiles/Sejarah%20dan%20Perkembangan%20Pemrograman%20Visual.pdf> ( diakses tanggal 14 agustus 2013 )
-

# LAMPIRAN

---



PT. BNI /PERSERO/ MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Keranglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : I KOMANG TRIPAYANA  
NIM : 09.12.518  
JURUSAN : Teknik Elektro S-1  
KONSENTRASI : Teknik Komputer  
MASA BIMBINGAN: SEMESTER GENAP 2012/2013  
JUDUL : RANCANG BANGUN APLIKASI PENGONTROL ROBOT  
BERBASIS AUGMENTED REALITY

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Jumat  
Tanggal : 16 Agustus 2013  
Dengan Nilai : 80,7 (A) *re*

**PANITIA UJIAN SKRIPSI**

**Ketua Majelis Penguji**

**M. Ibrahim Ashari, ST, MT**  
NIP.Y.1030100358

**Sekretaris Majelis Penguji**

**Dr. Eng. Aryanto S, ST, MT**  
NIP.P.1030800417

**ANGGOTA PENGUJI**

**Dosen Penguji I**

**Irmalia Suryani Faradisa, ST, MT**  
NIP.R.1030000365

**Dosen Penguji II**

**Bambang Prio Hartono, ST, MT**  
NIP.Y.1028400082



PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417836 Fax. (0341) 417834 Malang

## FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : I KOMANG TRIPAYANA  
NIM : 09.12.518  
JURUSAN : Teknik Elektro S-1  
KONSENTRASI : Teknik Komputer  
MASA BIMBINGAN : SEMESTER GENAP 2012/2013  
JUDUL : RANCANG BANGUN APLIKASI PENGONTROL ROBOT  
BERBASIS AUGMENTED REALITY

Tanggal	Uraian	Paraf
Penguji I 16-08-2013	1. Tujuan, Rumusan masalah, Manfaat di perbaiki 2. Perancangan deteksi marker, penentuan sumbu x, y dan z	

Disetujui,

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT  
NIP.P.1030000365

Dosen Penguji II

Bambang Prio Hartono, ST, MT  
NIP.Y.1028400082

Mengetahui,

Dosen Pembimbing I

M. Ibrahim Ashari, ST, MT  
NIP.P.1030100358

Dosen Pembimbing II

Bima Aulia Firmandani, ST



INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO

### FORMULIR BIMBINGAN SKRIPSI

Nama : I KOMANG TRIPAYANA  
Nim : 09.12.518  
Masa Bimbingan : Maret 2013 s/d agustus2013  
Judul Skripsi : "RANCANG BANGUN APLIKASI  
PENGONTROL ROBOT BERBASIS  
AUGMENTED REALITY"

No	Tanggal	Uraian	Paraf Pembimbing
1	18-07-2013	Acc BAB I	
2	18-07-2013	Revisi BAB II	
3	18-07-2013	Acc BAB III	
4	27-07-2013	Acc BAB IV	
5	27-07-2013	Acc BAB V	
6	27-07-2013	Acc Makalah	
7			
8			
9			
10			

Malang  
Dosen Pembimbing,

**M. Ibrahim Ashari, ST, MT**  
NIP.P.1030100358




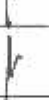


Form.S-4b



INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO

### FORMULIR BIMBINGAN SKRIPSI

Nama : I KOMANG TRIPAYANA  
Nim : 09.12.518  
Masa Bimbingan : Maret 2013 s/d agustus2013  
Judul Skripsi : "RANCANG BANGUN APLIKASI  
PENGONTROL ROBOT BERBASIS  
AUGMENTED REALITY"

No	Tanggal	Uraian	Paraf Pembimbing
1	07-06-2013	Referensi AR – Contoh-contoh implementasu	
2	18-07-2013	Perubahan judul menjadi Rancang bangun aplikasi pengontrol robot berbasis augmented reality	
3	18-07-2013	Acc BAB I, II, III	
4	27-07-2013	Acc BAB IV, V	
5	27-07-2013	Acc Makalah	
6	27-07-2013	Revisi redaksiona judul	
7			
8			
9			
10			

Malang  
Dosen Pembimbing,

  
**Bima Aulia Firmandani, ST**

Form.S-4b





PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

SEROJ MALANG  
VGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Huniling), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Surat : ITN-137/EL-FTI/2013

1 Mei 2013

in : -

: BIMBINGAN SKRIPSI

: Yth. Bapak/Ibu M. Ibrahim Ashari, ST, MT

Dosen Teknik Elektro S-1

ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : IKOMANG TRIPAYANA

Nim : 0912518

Fakultas : Teknologi Industri

Program Studi : Teknik Elektro S-1

Konsentrasi : Teknik Komputer

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Genap Tahun Akademik 2012 - 2013 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Ketua Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

SERO) MALANG  
GA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Surat : ITN-137/EL-FTI/2013

1 Mei 2013

n : -

: BIMBINGAN SKRIPSI

: Yth. Bapak/Ibu **Bima Aulia Firmandani, ST**  
Dosen Teknik Elektro S-1  
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **I KOMANG TRIPAYANA**  
Nim : **0912518**  
Fakultas : **Teknologi Industri**  
Program Studi : **Teknik Elektro S-1**  
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Genap Tahun Akademik 2012 - 2013 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui  
Ketua Program Studi Teknik Elektro S-1

**M. Ibrahim Ashari, ST, MT**

NIP.P. 1030100358

# LISTING PROGRAM

---

```
Imports System.Net
Imports System.Net.Sockets
Imports System.IO
```

```
Public Class frm_utama
    Dim request As WebRequest
    Dim response As WebResponse
    Dim dataStream As Stream
    Dim reader As StreamReader
    Dim responseFromServer As String
    Dim responseStatus As String
    Dim koordinat As String() = {"", "", ""}
    Dim x, y, z As Double

    Dim i As Integer = 0
    Dim action As String = ""
    Dim arah As String = ""

    Dim maju As String = "aa;"
    Dim kiri As String = "ac;"
    Dim kanan As String = "ca;"
    Dim astop As String = "bb;"
    Dim mundur As String = "ba;"
```

```
Private Sub frm_utama_Load(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MyBase.Load
```

```
    Try
        SerialPort1.Open()
        timer_accel.Enabled = True
    Catch ex As Exception
        MsgBox("Error: " & ex.ToString)
    End Try
```

```
End Sub
```

```
Private Sub timer_accel_Tick(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles timer_accel.Tick
```

```
    If SerialPort1.IsOpen Then
        lbl_status_bt.Text = "Bluetooth : Connected"
    Else
        lbl_status_bt.Text = "Bluetooth : Disconnected"
    End If
```

```
    Try
        request = WebRequest.Create("http://127.0.0.1:88")
        request.Credentials = CredentialCache.DefaultCredentials

        response = request.GetResponse()
        responseStatus = CType(response, HttpWebResponse).StatusDescription

        If responseStatus = "OK" Then
            dataStream = response.GetResponseStream()

            reader = New StreamReader(dataStream)

            responseFromServer = reader.ReadToEnd()

            updateArah()
        Else
            action = "bb;"
        End If
    Catch ex As Exception
        MsgBox("Error: " & ex.ToString)
    End Try
```

```
End Sub
```

```
Sub updateArah()
    koordinat = responseFromServer.Split(",")
```

```

x = Double.Parse(koordinat(0))
y = Double.Parse(koordinat(1))
z = Double.Parse(koordinat(2))

lbl_X.Text = "X = " & x.ToString
lbl_Y.Text = "Y = " & y.ToString
lbl_Z.Text = "Z = " & z.ToString

If x < 0.0 And y > 0.0 Then
    action = "aj;"
    arah = "Maju"
Elseif x > 0.0 And y > 0.0 Then
    action = "aj;"
    arah = "Kanan"
Elseif x > 0.0 And y < 0.0 Then
    action = "ak;"
    arah = "Kiri"
Elseif x < 0.0 And y < 0.0 Then
    action = "ak;"
    arah = "Mundur"
End If

lbl_action.Text = action & " - " & arah

sendDataBT()
End Sub

Sub sendDataBT()
    If SerialPort1.IsOpen Then
        SerialPort1.Write(action)
    End If
End Sub

Private Sub lbl_X_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles lbl_X.Click
End Sub
End Class

```

```

/*
'accelKit'
Copyright (C) 2010 Nikos Kastellanos

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; version 2 of the License.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

e-mail: nkastellanos@gmail.com
*/

// =====
// Includes
// =====

#include <stdio.h>
#include <stdlib.h>
#include <GL/glut.h>
#include <AR/config.h>
#include <AR/video.h>
#include <AR/param.h>
#include <AR/ar.h>
#include <AR/gsub_lite.h>

#include <wingdi.h>
#include "resource.h"

#include "WindowGrab.h"
#include "WServer.h"

// =====
// Constants
// =====

#define VIEW_SCALEFACTOR      0.025      // 1.0 ARToolkit unit becomes 0.025 of my ✓
      OpenGL units.
#define VIEW_DISTANCE_MIN     0.1        // Objects closer to the camera than this ✓
      will not be displayed.
#define VIEW_DISTANCE_MAX     100.0      // Objects further away from the camera ✓
      than this will not be displayed.

// =====
// Global variables
// =====

// Preferences.
static int prefWindowed = TRUE;
static int prefWidth = 640;           // Fullscreen mode width.
static int prefHeight = 480;         // Fullscreen mode height.
static int prefDepth = 32;           // Fullscreen mode bit depth.
static int prefRefresh = 0;          // Fullscreen mode refresh rate. Set to 0 ✓
      to use default rate.

// Image acquisition.
static ARuint8 *gARTImage = NULL;

// Marker Detection.
static int qARTThreshold = 100;
static long gCallCountMarkerDetect = 0;

```

```

// Transformation matrix retrieval.
static double gPatt_width = 80.0; // Per-marker, but we are using only 1 ✓
marker.
static double gPatt_centre[2] = {0.0, 0.0}; // Per-marker, but we are using only ✓
1 marker.
static double gPatt_trans[3][4]; // Per-marker, but we are using only 1 ✓
marker.
static int gPatt_found = FALSE; // Per-marker, but we are using only 1 ✓
marker.
static int gPatt_id; // Per-marker, but we are using only 1 ✓
marker.

// Drawing.
static ARParam gARTCparam;
static ARGV_CONTEXT_SETTINGS_REF gArgvSettings = NULL;

//textures
HBITMAP bmpA = 0;
HBITMAP bmpB = 0;
GLuint texture[2]; //define 2 textures

// =====
// Functions
// =====

void LoadGLTexture(HBITMAP bitmap) // Load Bitmaps And Convert To Textures
{
    // store bitmap data in a vector
    unsigned char data[256*479*3];
    unsigned char buff;
    int i;
    HDC hDC;
    HWND hWnd = 0;
    int res;

    // setup 24 bits bitmap structure
    BITMAPINFO info;
    BITMAPINFOHEADER header;
    header.biSize = sizeof(BITMAPINFOHEADER);
    header.biWidth = 256;
    header.biHeight = 479;
    header.biPlanes = 1;
    header.biBitCount = 24;
    header.biCompression = BI_RGB;
    header.biSizeImage = 0;
    header.biClrUsed = 0;
    header.biClrImportant = 0;
    info.bmiHeader = header;
    info.bmiColors->rgbRed = NULL;
    info.bmiColors->rgbGreen = NULL;
    info.bmiColors->rgbBlue = NULL;
    info.bmiColors->rgbReserved = NULL;

    // store bitmap data in a vector
    hDC = GetDC(hWnd);
    res = GetDIBits(hDC, bitmap, 0, 479, &data, &info, DIB_RGB_COLORS);
    res = ReleaseDC(hWnd, hDC);

    // convert from BGR to RGB
    for(i=0; i<256*479; i++)
    {
        buff = data[i*3];
        if(i>=3)
        {
            data[i*3] = data[i*3+2];
            data[i*3+2] = buff;
        }
    }
}

```

```

    }
}

glGenTextures(1, &texture[0]); // Create The Texture
// Build Texture
{
// Typical Texture Generation Using Data From the Bitmap
glBindTexture(GL_TEXTURE_2D, texture[0]);
// Generate The Texture
glTexImage2D(GL_TEXTURE_2D, 0, 3, 256, 479, 0, GL_RGB, GL_UNSIGNED_BYTE, &data);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
}

}

// Something to look at, draw a rotating colour cube.
static void DrawWP7(void)
{
    BITMAPINFOHEADER bmih_out;
    HDC inDC, outDC;

    BITMAP bmp;
    HBITMAP old_bitmap;
    HBITMAP image;
    void *pixels=NULL;
    float w=256,h=256;
    int texh=0, texw=0;
    int ret;

    //paint hLocalBitmap on texture
    HBITMAP wp7bmp = wgCapture(bmpA);
    if(wp7bmp!=NULL)
    {
        /*
        // get the dimensions of the passed bitmap
        SetObject(hLocalBitmap, sizeof(BITMAP), &bmp);
        h = (float)bmp.bmHeight;
        w = (float)bmp.bmWidth;

        // figure out what the texture dimensions have to be in order to encompass the
entire image
        texh = next_po2((int)h);
        texw = next_po2((int)w);

        // create a DIB big enough to hold the original image and all of it's padding
        bmih_out.biSize = sizeof(bmih_out);
        bmih_out.biWidth = texw;
        bmih_out.biHeight = texh;
        bmih_out.biPlanes = 1;
        bmih_out.biBitCount = 32;
        bmih_out.biCompression = BI_RGB;
        bmih_out.biSizeImage = 0;
        bmih_out.biXPelsPerMeter = 0;
        bmih_out.biYPelsPerMeter = 0;
        bmih_out.biClrUsed = 0;
        bmih_out.biClrImportant = 0;

        image = CreateDIBSection(NULL, (BITMAPINFO*)&bmih_out, DIB_RGB_COLORS, &
(pixels), NULL, 0);

        // select the bitmap resource into input DC
        inDC = CreateCompatibleDC(NULL);
        SelectObject(inDC, hLocalBitmap);

```



```

    // select the new DIB section into the output DC
    outDC = CreateCompatibleDC(NULL);
    old_bitmap = (HBITMAP)SelectObject(outDC, image);

    // copy the bitmap into the new DIB section and then unselect the DIB section
    from the outDC
    ret = BitBlt(outDC, 0, texh-(int)h, (int)w, (int)h, inDC, 0, 0, SRCCOPY);
    SelectObject(outDC, old_bitmap);

    DeleteDC(inDC);
    DeleteDC(outDC);

    // create the texture
    glGenTextures(1, &texture[0]);
    glBindTexture(GL_TEXTURE_2D, texture[0]);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, w, h, 0, GL_BGRA_EXT, GL_UNSIGNED_BYTE,
    . pixels);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);

    if (image!=NULL) DeleteObject(image);
*/
}

LoadGLTexture(bmpA);

glPushMatrix(); // Save world coordinate system.
glTranslatef(0.0, 0.0, 0.0); // Place base of cube on marker surface.

// Enable texturing and select first texture
glColor3f(1.0f,1.0f,1.0f);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D,texture[0]);

// Draw the top face
glBegin(GL_QUADS);
glTexCoord2f(0.0f,1.0f); glVertex3f(-1.6f, 2.43f, 0.0f);
glTexCoord2f(1.0f,1.0f); glVertex3f( 1.6f, 2.43f, 0.0f);
glTexCoord2f(1.0f,0.0f); glVertex3f( 1.6f,-2.90f, 0.0f);
glTexCoord2f(0.0f,0.0f); glVertex3f(-1.6f,-2.90f, 0.0f);

glEnd();

glPopMatrix(); // Restore world coordinate system.

}

BOOL is_po2(int a)
{
    if (a <= 0) return FALSE;
    return ((a & -a) == a);
}

int next_po2(int a)
{
    int i;
    if (is_po2(a)) return a;
    for (i=1; i<=sizeof(int)*8; i++)
    {
        a >>= 1;
        if (a == 0) break;
    }
}

```

```

    }
    return (1 << i);
}

static int setupCamera(const char *cparam_name, char *vconf, ARParam *cparam)
{
    ARParam      wparam;
    int          xsize, ysize;

    // Open the video path.
    if (arVideoOpen(vconf) < 0) {
        fprintf(stderr, "setupCamera(): Unable to open connection to camera.\n");
        return (FALSE);
    }

    // Find the size of the window.
    if (arVideoInqSize(&xsize, &ysize) < 0) return (FALSE);
    fprintf(stdout, "Camera image size [x,y] = (%d,%d)\n", xsize, ysize);

    // Load the camera parameters, resize for the window and init.
    if (arParamLoad(cparam_name, 1, &wparam) < 0) {
        fprintf(stderr, "setupCamera(): Error loading parameter file %s for camera.\n",
            cparam_name);
        return (FALSE);
    }
    arParamChangeSize(&wparam, xsize, ysize, cparam);
    fprintf(stdout, "*** Camera Parameter ***\n");
    arParamDisp(cparam);

    arInitCparam(cparam);

    if (arVideoCapStart() != 0) {
        fprintf(stderr, "setupCamera(): Unable to begin camera data capture.\n");
        return (FALSE);
    }

    return (TRUE);
}

static int setupMarker(const char *patt_name, int *patt_id)
{
    if ((*patt_id = arLoadPatt(patt_name)) < 0) {
        fprintf(stderr, "setupMarker(): pattern load error !!\n");
        return (FALSE);
    }

    return (TRUE);
}

// Report state of ARToolKit global variables arFittingMode,
// arImageProcMode, arGLDrawMode, arTemplateMatchingMode, arMatchingPCAMode.
static void debugReportMode(void)
{
    if (arFittingMode == AR_FITTING_TO_INPUT) {
        fprintf(stderr, "FittingMode (%d): INPUT IMAGE\n");
    } else {
        fprintf(stderr, "FittingMode (%d): COMPENSATED IMAGE\n");
    }

    if (arImageProcMode == AR_IMAGE_PROC_IN_FULL) {
        fprintf(stderr, "ProcMode (%d) : FULL IMAGE\n");
    } else {
        fprintf(stderr, "ProcMode (%d) : HALF IMAGE\n");
    }

    if (arGLDrawModeGet(gArglSettings) == AR_DRAW_BY_GL_DRAW_PIXELS) {
        fprintf(stderr, "DrawMode (C) : GL_DRAW_PIXELS\n");
    } else if (arGLTexmapModeGet(gArglSettings) == AR_DRAW_TEXTURE_FULL_IMAGE) {

```

```

    fprintf(stderr, "DrawMode (C) : TEXTURE MAPPING (FULL RESOLUTION)\n");
} else {
    fprintf(stderr, "DrawMode (C) : TEXTURE MAPPING (HALF RESOLUTION)\n");
}

if( arTemplateMatchingMode == AR_TEMPLATE_MATCHING_COLOR ) {
    fprintf(stderr, "TemplateMatchingMode (M) : Color Template\n");
} else {
    fprintf(stderr, "TemplateMatchingMode (M) : BW Template\n");
}

if( arMatchingPCAMode == AR_MATCHING_WITHOUT_PCA ) {
    fprintf(stderr, "MatchingPCAMode (P) : Without PCA\n");
} else {
    fprintf(stderr, "MatchingPCAMode (P) : With PCA\n");
}
}

static void Quit(void)
{
    arglCleanup(gArglSettings);
    arVideoCapStop();
    arVideoClose();
    exit(0);
}

static void Keyboard(unsigned char key, int x, int y)
{
    int mode;
    switch (key)
    {
        case 0x1B: // Quit.
        case 'Q':
        case 'q':
            Quit();
            break;
        case 'C':
        case 'c':
            mode = arglDrawModeGet(gArglSettings);
            if (mode == AR_DRAW_BY_GL_DRAW_PIXELS)
            {
                arglDrawModeSet(gArglSettings, AR_DRAW_BY_TEXTURE_MAPPING);
                arglTexmapModeSet(gArglSettings, AR_DRAW_TEXTURE_FULL_IMAGE);
            }
            else
            {
                mode = arglTexmapModeGet(gArglSettings);
                if (mode == AR_DRAW_TEXTURE_FULL_IMAGE) arglTexmapModeSet
(gArglSettings, AR_DRAW_TEXTURE_HALF_IMAGE);
                else arglDrawModeSet(gArglSettings, AR_DRAW_BY_GL_DRAW_PIXELS);
            }
            fprintf(stderr, "*** Camera - %f (frame/sec)\n", (double)
gCallCountMarkerDetect/arUtilTimer());
            gCallCountMarkerDetect = 0;
            arUtilTimerReset();
            debugReportMode();
            break;
        case 'D':
        case 'd':
            arDebug = !arDebug;
            break;
        case '?':
        case '/':
            printf("Keys:\n");
            printf(" q or /esc) Quit Here.\n");
            printf(" c Change arglDrawMode and arglTexmapMode.\n");
            printf(" d Activate / Deactivate debug mode.\n");
            printf(" ? or / Show this help.\n");
            printf("\nAdditionally, the ARVideo library supplied the following help
text:\n");
    }
}

```

```

        arVideoDispOption();
        break;
    default:
        break;
    }
}

static void Idle(void)
{
    static int ms_prev;
    int ms;
    float s_elapsed;
    ARUint8 *image;

    ARMarkerInfo *marker_info; // Pointer to array holding the
    details of detected markers.
    int marker_num; // Count of number of markers
    detected.
    int j, k;

    // Find out how long since Idle() last ran.
    ms = glutGet(GLUT_ELAPSED_TIME);
    s_elapsed = (float)(ms - ms_prev) * 0.001;
    if (s_elapsed < 0.01f) return; // Don't update more often than 100 Hz.
    ms_prev = ms;

    // Grab a video frame.
    if ((image = arVideoGetImage()) != NULL)
    {
        gARTImage = image; // Save the fetched image.
        gPatt_found = FALSE; // Invalidate any previous detected markers.

        gCallCountMarkerDetect++; // Increment ARToolKit FPS counter.

        // Detect the markers in the video frame.
        if (arDetectMarker(gARTImage, gARTThreshold, &marker_info, &marker_num) < 0)
        {
            exit(-1);
        }

        // Check through the marker_info array for highest confidence
        // visible marker matching our preferred pattern.
        k = -1;
        for (j = 0; j < marker_num; j++) {
            if (marker_info[j].id == gPatt_id) {
                if (k == -1) k = j; // First marker detected.
                else if (marker_info[j].cf > marker_info[k].cf) k = j; // Higher
confidence marker detected.
            }
        }

        if (k != -1) {
            // Get the transformation between the marker and the real camera into
gPatt_trans.
            arGetTransMat(&marker_info[k], gPatt_centre, gPatt_width, gPatt_trans);
            gPatt_found = TRUE;
        }

        // Tell GLUT the display has changed.
        glutPostRedisplay();
    }
}

//
// This function is called on events when the visibility of the
// GLUT window changes (including when it first becomes visible).
//
static void Visibility(int visible)
{
    if (visible == GLUT_VISIBLE)

```

```

    {
        glutIdleFunc(idle);
    }
    else
    {
        glutIdleFunc(NULL);
    }
}

//
// This function is called when the
// GLUT window is resized.
//
static void Reshape(int w, int h)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glViewport(0, 0, (GLsizei) w, (GLsizei) h);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    // Call through to anyone else who needs to know about window sizing here.
}

//
// This function is called when the window needs redrawing.
//
static void Display(void)
{
    GLdouble p[16];
    GLdouble m[16];

    // Select correct buffer for this context.
    glDrawBuffer(GL_BACK);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // Clear the buffers for new frame. ✓

    arglDisplImage(gARTImage, &gARTCparam, 1.0, gArglSettings); // zoom = 1.0.
    arVideoCapNext();
    gARTImage = NULL; // Image data is no longer valid after calling arVideoCapNext().

    if (gPatt_found)
    {
        // Projection transformation.
        arglCameraFrustumRH(&gARTCparam, VIEW_DISTANCE_MIN, VIEW_DISTANCE_MAX, p);
        glMatrixMode(GL_PROJECTION);
        glLoadMatrixd(p);
        glMatrixMode(GL_MODELVIEW);

        // Viewing transformation.
        glLoadIdentity();
        // Lighting and geometry that moves with the camera should go here.
        // (i.e. must be specified before viewing transformations.)
        //none

        wsoutx = gPatt_trans[1][0];
        wsouty = gPatt_trans[1][1];
        wsoutz = gPatt_trans[1][2];

        // ARToolKit supplied distance in millimetres, but I want OpenGL to work in my units. ✓
        arglCameraViewRH(gPatt_trans, m, VIEW_SCALEFACTOR);
        glLoadMatrixd(m);

        // All other lighting and geometry goes here.
        DrawWP7();
    } // gPatt_found
}

```

```

    // Any 2D overlays go here.
    //none

    glutSwapBuffers();
}

HBITMAP LoadBmp()
{
    // load bitmap from resource file
    HBITMAP bitmap = LoadBitmap(GetModuleHandle(NULL), MAKEINTRESOURCE(IDB_BITMAP1));
    return bitmap;
}

int main(int argc, char** argv)
{
    char glutGamemode[32];
    const char *cparam_name = "Data/camera_para.dat";
    //
    // Camera configuration.
    //
#ifdef _WIN32
    char *vconf = "Data\\WDM_camera_acc.xml";
#else
    char *vconf = "";
#endif
    const char *patt_name = "Data/patt.accel";

    // -----
    // Library inits.
    //
    glutInit(&argc, argv);

    // -----
    // start our web server
    //
    WServerStart(88);

    // -----
    // Hardware setup.
    //
    if (!setupCamera(cparam_name, vconf, &gARTCparam)) {
        fprintf(stderr, "main(): Unable to set up AR camera.\n");
        exit(-1);
    }
    if (!setupMarker(patt_name, &gPatt_id)) {
        fprintf(stderr, "Main(): Unable to set up AR marker.\n");
        exit(-1);
    }

    // -----
    // Library setup.
    //

    // Set up GL context(s) for OpenGL to draw into.
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
    if (!prefWindowed) {
        if (prefRefresh) sprintf(glutGamemode, "%ix%i:%i@%i", prefWidth, prefHeight, ✓
prefDepth, prefRefresh);
        else sprintf(glutGamemode, "%ix%i:%i", prefWidth, prefHeight, prefDepth);
        glutGameModeString(glutGamemode);
        glutEnterGameMode();
    }

```

```
} else {
    glutInitWindowSize(prefWidth, prefHeight);
    glutCreateWindow(argv[0]);
}

// Setup argl library for current context.
if ((gArglSettings = arglSetupForCurrentContext()) == NULL) {
    fprintf(stderr, "main(): arglSetupForCurrentContext() returned error.\n");
    exit(-1);
}

debugReportMode();
glEnable(GL_DEPTH_TEST);
arUtilTimerReset();

// Register GLUT event-handling callbacks.
// NB: Idle() is registered by Visibility.
glutDisplayFunc(Display);
glutReshapeFunc(Reshape);
glutVisibilityFunc(Visibility);
glutKeyboardFunc(Keyboard);

bmpA = LoadBmp();
LoadGLTexture(bmpA);

//wgStartL("XDE_SkinWindow");
wgStartL("XDE_LCDWindow");

glutMainLoop();

DeleteObject(bmpA);

return 0;
}
```

```
#region Copyright notice
//-----
// AccelerometerEmuReadingEventArgs.cs
// Copyright (C) Kastellanos Nikos. All rights reserved.
//-----
#endregion

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.Devices.Sensors;

namespace NKast.Sensors
{
    class AccelerometerEmuReadingEventArgs : EventArgs
    {
        private double _X, _Y, _Z;

        public AccelerometerEmuReadingEventArgs(double x, double y, double z)
        {
            this._X = x;
            this._Y = y;
            this._Z = z;
        }

        public DateTimeOffset Timestamp { get { return new DateTimeOffset(); } }
        public double X { get { return _X; } }
        public double Y { get { return _Y; } }
        public double Z { get { return _Z; } }
    }
}
```



```

#region Copyright notice
//-----
// AccelerometerEmu.cs
// Copyright (C) Kastellanos Nikos. All rights reserved.
//-----
#endregion

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.Devices.Sensors;
using System.Security;
using System.Threading;
using System.Net;
using System.Windows.Media.Animation;

namespace NKast.Sensors
{
    class AccelerometerEmu : IDisposable
    {
        private static Microsoft.Devices.Sensors.Accelerometer accelerometer;
        private Storyboard timer;

        public event EventHandler<AccelerometerEmuReadingEventArgs> ReadingEmuChanged;

        [SecuritySafeCritical]
        public AccelerometerEmu()
        {
            accelerometer = new Microsoft.Devices.Sensors.Accelerometer();

            if (Microsoft.Devices.Environment.DeviceType == Microsoft.Devices.
                DeviceType.Device)
            {
                ReadingChanged += new EventHandler<AccelerometerReadingEventArgs>
                (AccelerometerEmu_ReadingChanged);
            }

            return;
        }

        public SensorState State
        {
            get { return accelerometer.State; }
        }

        // Summary:
        // Occurs when new data arrives from the accelerometer.
        public event EventHandler<AccelerometerReadingEventArgs> ReadingChanged;

        [SecuritySafeCritical]
        public void Dispose()
        {
            accelerometer.Dispose();
        }

        [SecuritySafeCritical]
        public void Start()
        {
            accelerometer.Start();

            if (Microsoft.Devices.Environment.DeviceType == Microsoft.Devices.
                DeviceType.Emulator)
            {
                timer = new Storyboard();
                timer.Duration = TimeSpan.FromMilliseconds(1000/10);
                timer.Completed += new EventHandler(TimerCallBack);
                timer.Begin();
            }
        }
    }
}

```

```

    }
    //
    // Summary:
    // Stops data acquisition from the accelerometer.
    [SecuritySafeCritical]
    public void Stop()
    {
        accelerometer.Stop();
        if (Microsoft.Devices.Environment.DeviceType == Microsoft.Devices.
DeviceType.Emulator)
        {
            timer.Stop();
            timer = null;
        }
    }

    void AccelerometerEmu_ReadingChanged(object sender,
AccelerometerReadingEventArgs e)
    {
        ReadingChanged(this, e);
    }

    public void TimerCallback(object sender, EventArgs e)
    {
        WebClient wc;
        wc = new WebClient();
        wc.AllowReadStreamBuffering = false;
        wc.DownloadStringCompleted += new DownloadStringCompletedEventHandler
(wc_DownloadStringCompleted);
        wc.DownloadStringAsync(new Uri("http://127.0.0.1:88/"));

        return;
    }

    void wc_DownloadStringCompleted(object sender,
DownloadStringCompletedEventArgs e)
    {
        if (e.Error != null) { timer.Begin(); return; }
        if (e.Result == null) { timer.Begin(); return; }
        string[] vc = e.Result.Split(new Char[] { ',', ' ' });
        double x = Convert.ToDouble(vc[0]);
        double y = Convert.ToDouble(vc[1]);
        double z = Convert.ToDouble(vc[2]);

        AccelerometerEmuReadingEventArgs are = new
AccelerometerEmuReadingEventArgs(x, y, z);
        ReadingEmuChanged(this, are);

        timer.Begin();
        return;
    }
}
}

```

```

/*
'accelKit'
Copyright (C) 2010 Nikos Kastellanos

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; version 2 of the License.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

e-mail: nkastellanos@gmail.com
*/

#include "WServer.h"
#include <Windows.h>
#include <string.h>

//A SOCKET is simply a typedef for an unsigned int.
SOCKET server;
//WSADATA is a struct that is filled up by the call to WSASStartup
WSADATA wsaData;
//The sockaddr_in specifies the address of the socket
SOCKADDR_IN local;
char lszThreadParam[3];

double wsoutx,wsouty,wsoutz;

void WServerStart(int port)
{
    HANDLE hThread;

    DWORD dwGenericThread;

    wsoutx=0,wsouty=-1,wsoutz=0;

    lszThreadParam[0] = port;
    hThread = CreateThread(NULL,0,WServerThread,&lszThreadParam,0,&dwGenericThread);

    if(hThread == NULL)
    {
        DWORD dwError = GetLastError();
        //cout<<"SCM:Error in Creating thread"<<dwError<<endl ;
        return;
    }
    return;
}

DWORD WINAPI WServerInit(int port)
{
    //WSASStartup initializes the program for calling WinSock.
    //The first parameter specifies the highest version of the
    //WinSock specification, the program is allowed to use.
    int wsaret=WSASStartup(0x101,&wsaData);

    //WSASStartup returns zero on success.
    //If it fails we exit.
    if(wsaret!=0)
    {
        return 0;
    }

    //Now we populate the sockaddr_in structure

```

```

local.sin_family=AF_INET; //Address family
local.sin_addr.s_addr=INADDR_ANY; //Wild card IP address
local.sin_port=htons((u_short)port); //port to use

//the socket function creates our SOCKET
server=socket(AF_INET,SOCK_STREAM,0);

//If the socket() function fails we exit
if(server==INVALID_SOCKET)
{
    return 0;
}

//bind links the socket we just created with the sockaddr_in
//structure. Basically it connects the socket with
//the local address and a specified port.
//If it returns non-zero quit, as this indicates error
if(bind(server, (SOCKADDR*)&local, sizeof(local))!=0)
{
    return 0;
}

//listen instructs the socket to listen for incoming
//connections from clients. The second arg is the backlog
if(listen(server,10)!=0)
{
    return 0;
}

return -1;
}

DWORD WINAPI WServerListen()
{
    //we will need variables to hold the client socket.
    //thus we declare them here.
    SOCKET client;
    SOCKADDR_IN from;
    int fromlen=sizeof(from);

    char temp[1024*4];
    int rcvcount=0;
    char *res;

    //accept() will accept an incoming
    //client connection
    client=accept(server, (struct sockaddr*)&from,&fromlen);

    // consume the request header
    while(TRUE)
    {
        rcvcount=recv(client,temp,1024*4,0);
        //if(rcvcount==0) continue;
        //res=strstr(temp,"\r\n\r\n");
        break;
    }

    //we simply send this string to the client
    //sprintf(temp,"Your IP is %s\r\n",inet_ntoa(from.sin_addr));
    //send(client,temp,strlen(temp),0);

    //send dummy http header
    sprintf(temp,"HTTP/1.0 200 OK\r\n");
    send(client,temp,strlen(temp),0);
    sprintf(temp,"Server: FTTFaccel/1.0\r\n");
    send(client,temp,strlen(temp),0);
    sprintf(temp,"Content-Type: application/octet-stream\r\n");
    send(client,temp,strlen(temp),0);
    sprintf(temp,"Expires: Thu, 01 Dec 1994 10:00:00 GMT\r\n");
    send(client,temp,strlen(temp),0);
    sprintf(temp,"Pragma: no-cache\r\n");

```

---

```
send(client,temp,strlen(temp),0);
sprintf(temp,"\r\n"); //end
send(client,temp,strlen(temp),0);
//Date: Sat, 27 Nov 2024 10:19:07 GMT
//Content-Length: 10476

sprintf(temp,"%f,%f,%f",wsoutx,wsouty,wsoutz); //end
send(client,temp,strlen(temp),0);

//cout << "Connection from " << inet_ntoa(from.sin_addr) << "\r\n";

//close the client socket
closesocket(client);

return TRUE;
}

DWORD WINAPI WServerCleanup()
{
    //closesocket() closes the socket and releases the socket descriptor
    closesocket(server);

    //originally this function probably had some use
    //currently this is just for backward compatibility
    //but it is safer to call it as I still believe some
    //implementations use this to terminate use of WS2_32.DLL
    WSACleanup();
}

DWORD WINAPI WServerThread(LPVOID iValue)
{
    int port = ((char*)iValue)[0];

    if(!WServerInit(port)) return FALSE;

    while(TRUE) //we are looping endlessly
    {
        WServerListen();
    }

    WServerCleanup();

    return 0;
}
```