# *GAME* MEMECAHKAN BALON BERBASIS AUGMENTED REALITY






**NAMA** : **RICA YUNITA**

**NIM** : **10107875**

**PEMBIMBING** : **GALIH HERMAWAN, S.KOM,M.T**

# LATAR BELAKANG MASALAH

1. Melatih syaraf motorik halus anak harus dilakukan dalam keadaan yang menyenangkan. Permainan adalah alternatif dalam melatih syaraf motorik anak.
2. Pemanfaatan teknologi pada permainan yang dapat melatih syaraf motorik anak. salah satu permainan yang dapat melatih adalah mengelindingkan bola.
3. Pemanfaatan teknologi AR dalam permainan.

# RUMUSAN MASALAH

1. Bagaimana membangun aplikasi permainan mengelindingkan bola dengan mengimplementasikan teknologi *Augmented Reality.*
2. Bagaimana membangun aplikasi permainan mengelindingkan bola sehingga pemain tidak membutuhkan *joystick* maupun tombol control untuk memainkan permainan ini.



# BATASAN MASALAH

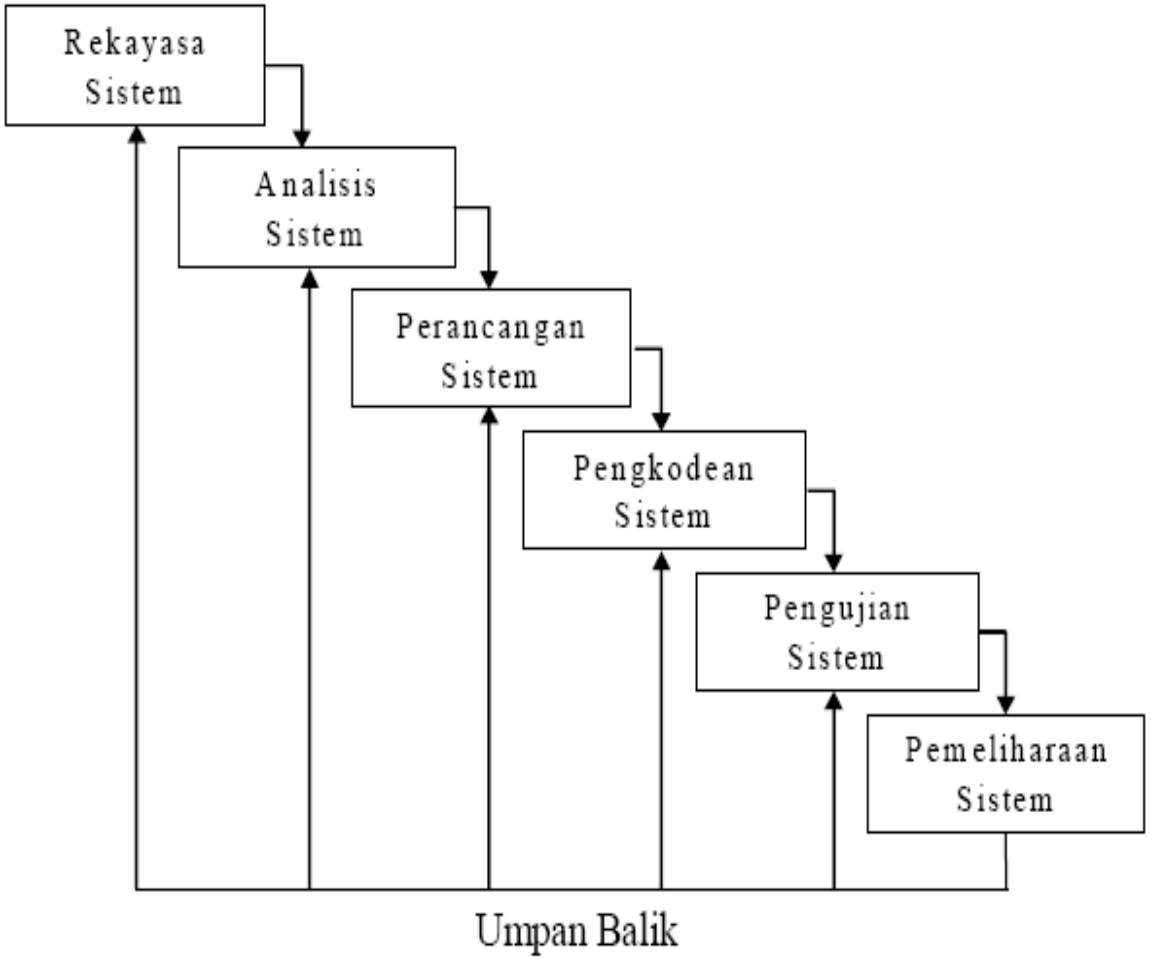
1. Dalam perancangan *game* menggunakan Papervision3D untuk membentuk objek-objek 3D.
2. Untuk membuat permainan lebih intuitif dan lebih nyata maka digunakan library JiglibFlash untuk melakukan deteteksi tabrakan (*collision detection*).
3. Dalam melakukan *tracking marker* permainan ini menggunkan library IN2AR
4. Karena menggunakan library dalam proses *tracking marker,* maka desain marker/penanda harus dikonversi terlebih dahulu ke file berbentuk *. proses konversi masih perlu melibatkan IN2AR *support team*
5. Untuk menjalankan game berbasis *Augmented reality* ini dibutuhkan kamera dan marker/penanda yang telah didesain.

# METODE PENELITIAN

a. Analisis kebutuhan
b. Tahap perancangan
c. Tahap implementasi
d. Pengujian dan evaluasi
e. Dokumentasi



# MODEL PERANCANGAN



# AUGMENTED REALITY

- Adalah teknologi yang menggabungkan benda maya dua dimensi dan atau tiga dimensi kedalam sebuah lingkuangan nyata tiga dimensi lalu memproyeksikan benda-benda meya tersebut dalam waktu nyata.
- Namun AR tidak memisahkan yang nyata dengan virtual, yang terjadi adalah penggabungan antara keduanya pada ruang yang sama.



# TRACKING MARKER

- *Tracking marker* yaitu proses deteksi keberadaan penanda atau marker yang berbentuk matriks.
- Library IN2AR adalah proses *tracking marker* dapat menggunakan gambar yang lebih rumit, sehingga gambar marker yang digunakan tidak monoton dengan gambar persegi dan warna hitam dan putih.





Marker yang biasa digunakan




Marker menggunakan gambar

# LANJUTAN TRACKING MARKER

- Tracking marker pada library IN2AR menggunakan deteksi tepi (*edge detection*), metode deteksi tepi yang digunakann adalah metode canny.
- Metode canny menggunakan Gaussian Derivate Kernel untuk menyaring kegaduhan dari citra. Proses penyaringan yang dilakukan Gaussian Derivate Kernel adalah sebagai berikut:

# LANJUTAN TRACKING MARKER

1. Konversi gambar dari citra berwarna ke citra gray level.

   contoh gambar :




Citra warna




Citra gray level

# LANJUTAN TRACKING MARKER

2. Merupakan proses untuk konversi dari nilai citra gray level menjadi citra yang memiliki dua warna yaitu hitam dan putih.

   dalam proses ini terdapat beberapa proses:

- Proses Dilasi

  yaitu memperbesar ukuran segmen obyek dengna menambah lapisan disekeliling obyek.




Citra Gray Level




Proses Dilasi

# LANJUTAN TRACKING MARKER

- Proses Erosi

  yaitu proses erosi adalah kebalikan dari proses dilasi. Pada proses ini, ukuran obyek diperkecil dengan mengikis sekeliling obyek.




Citra hasil Dilasi




Proses Erosi

# LANJUTAN TRACKING MARKER

- Proses Dilasi-Erosi

  yaitu kombinasi antara proses dilasi dan erosi yang dilakukan secara berurutan.




Dilasi Kedua




Proses Erosi Kedua

# LANJUTAN TRACKING MARKER

- Proses Pengulangan

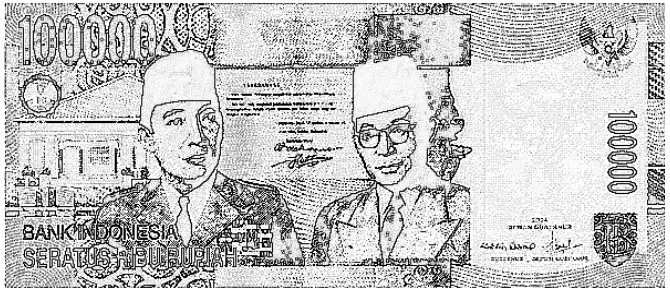
  Pengulangan adalah suatu proses pengikisan sebuah obyek sebanyak mungkin dengan tetap mempertahankan bentuk umum dari polanya.




Proses Pengulangan

# LANJUTAN TRACKING MARKER

- Selanjutnya adalah proses deteksi tepi yang menghasilkan tepi-tepi dari obyek citra yang tujuannya adalah menandai bagian yang menjadi detail citra. Dari proses ini akan diperoleh variabel yang digunakan sebagai penanda/marker

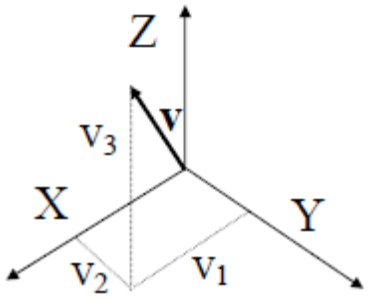


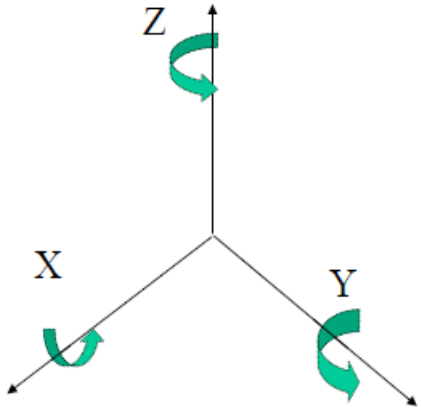

Proses Deteksi Tepi Gambar *Marker*

# KONTROL GAME DENGAN MARKER

- Untuk menggerakan bola sesuai dengan gerakan *marker*, perlu diketahui nilai transformasi *marker* terlebih dahulu. Setelah didapat nilai transformasi 3dimensi berbentuk matriks 4x4 kemudian diubah kematriks *euler*. Metode ini sudah ada dalam library papervision3D.
- Matrix rotasi untuk grafis 3D dengan representasi *euler* terdiri dari rotasi terhadap sumbu x, y, dan z.





# COLLITION DETECTION

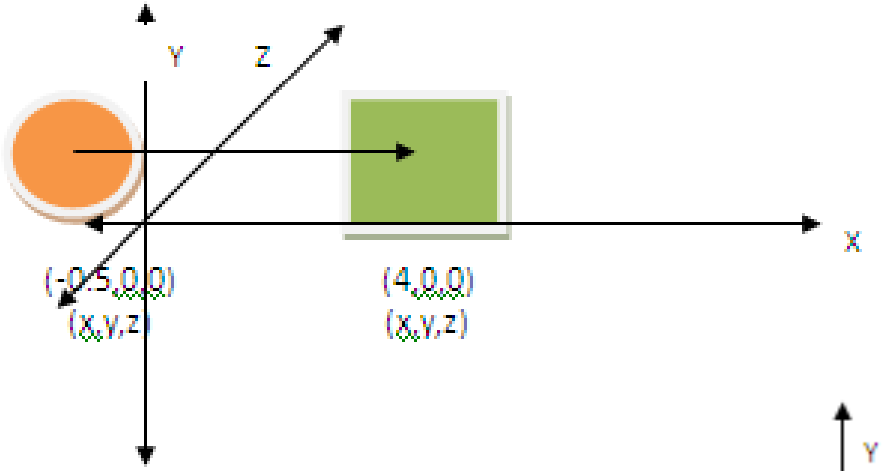
- *Collision Detection* adalah fungsi yang dapat mendeteksi tabrakan antara 2 obyek atau lebih.*Collision detection* juga berguna untuk menentukan posisi dari satu objek dengan objek yang lain sehingga tidak ada objek yang saling menembus. Sehingga game yang akan dibuat memiliki kesamaan dengan realita yang ada. Deteksi tabrakan adalah salah satu fungsi yang telah dimiliki oleh JiglibFlash.

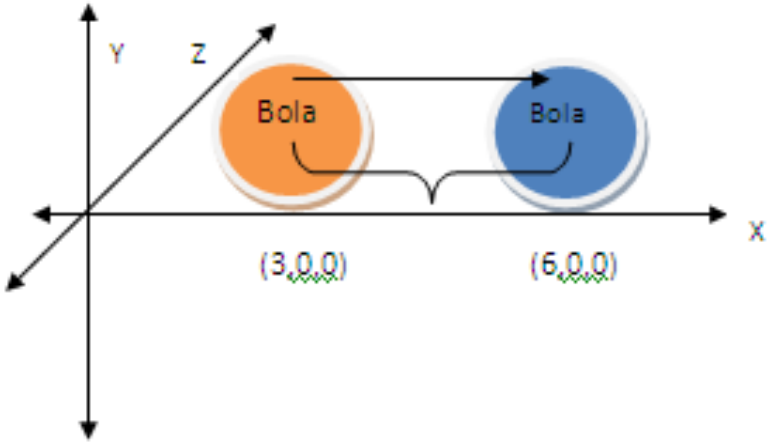
  *Collition detection* yang dilakukan adalah sebagai berikut:

- *Sphere-Plane Collition Detection*

  Pada deteksi *Sphere-Plane Collition Detection* ini menghitung nilai radius atau titik tengah dari obyek simetri *sphere* dan *cube*. Nilai ini nanti yang digunakan dalam menghitung perpotongan obyek. Jika jarak dari pusat *sphere* ke sebuah obyek kurang dari atau sama dengan jari-jari bola, maka tabrakan telah terjadi.

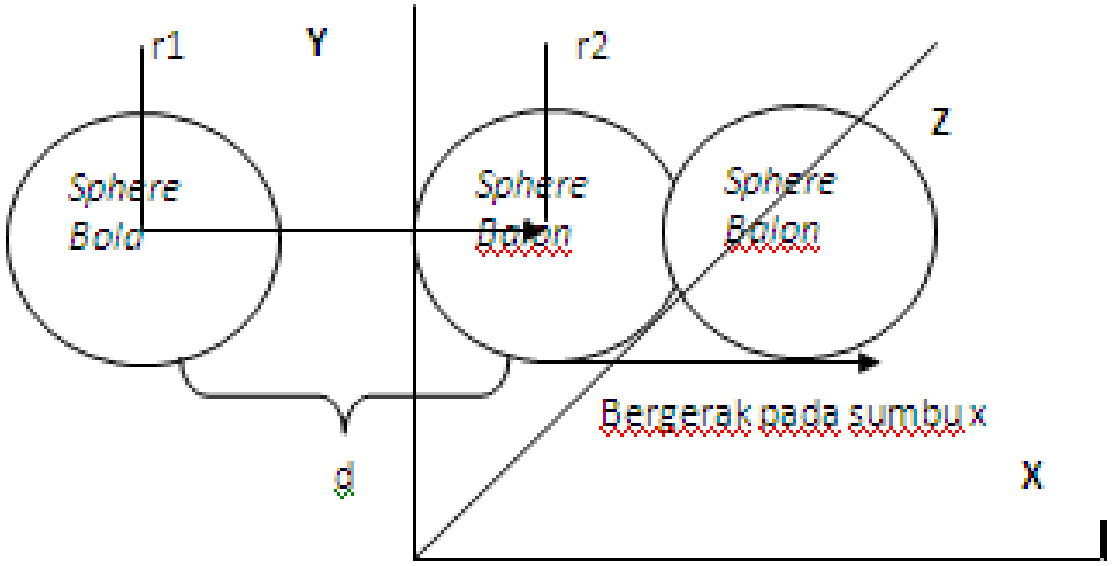
# COLLITION DETECTION



# COLLISION DETECTION

- Deteksi kedua yang dilakukan dalam permainan ini adalah deteksi gerak *Sphere* (balon) jika bersentuhan dengan obyek *sphere* (bola).



# KEBUTUHAN NON FUNGSIONAL

Analisi perangkat keras:

Berdasarkan studi literatur terhadap proyek-proyek pengembangan teknologi AR, maka diperolah spesifikasi minimum perangkat keras untuk teknologi AR seperti berikut ini:

- Processor pentum 4 > 1 GHz atau setara.
- RAM minimal 2 GB
- Ruang sisa hardisk minimal 4 GB
- Motherboard dengan chipset yang kompatibel dengan VGA card yang dipakai.
- Memiliki opsi konektifitas USB 2.0 dan nirkabel.
- Optimum menggunakan VGA card dengan kemampuan me-render grafis 3D, seperti GeForce 6xxx atau ATI 1xxx series.



# KEBUTUHAN NON FUNGSIONAL

Analisi perangkat keras:

- *Webcam*

  *Webcam* dipakai untuk menangkap citra yang kemudian diproses oleh *tracker library.* Hasil dari proses ini akan menghasilkan matriks transformasi *marker* relative terhadap *webcam*. Saat ini di pasaran sudah cukup banyak *webcam* dengan harga yang terjangkau.

- *Marker*

  *Marker* dibutuhkan oleh *tracking library* untuk menempatkan model virtual di dunia nyata dengan cara menentukan koordinat *marker* relative terhadap kamera. Dengan mngetahui koordinat relative tersebut kita dapat berinteraksi dengan AR. *Marker* yang dibutuhkan yaitu, 1 buah *marker* yang telah didesain untuk permainan.

- *Proyektor*

  Proyektor dibutuhkan untuk jumlah partisipan yang banyak, sehingga partisipan yang tidak perlu melihat ke layar monitor yang kecil. Proyektor sangat berguna dalam kerja kolaborasi permainan ini.



# KEBUTUHAN NON FUNGSIONAL

## Analisis Perangkat Lunak

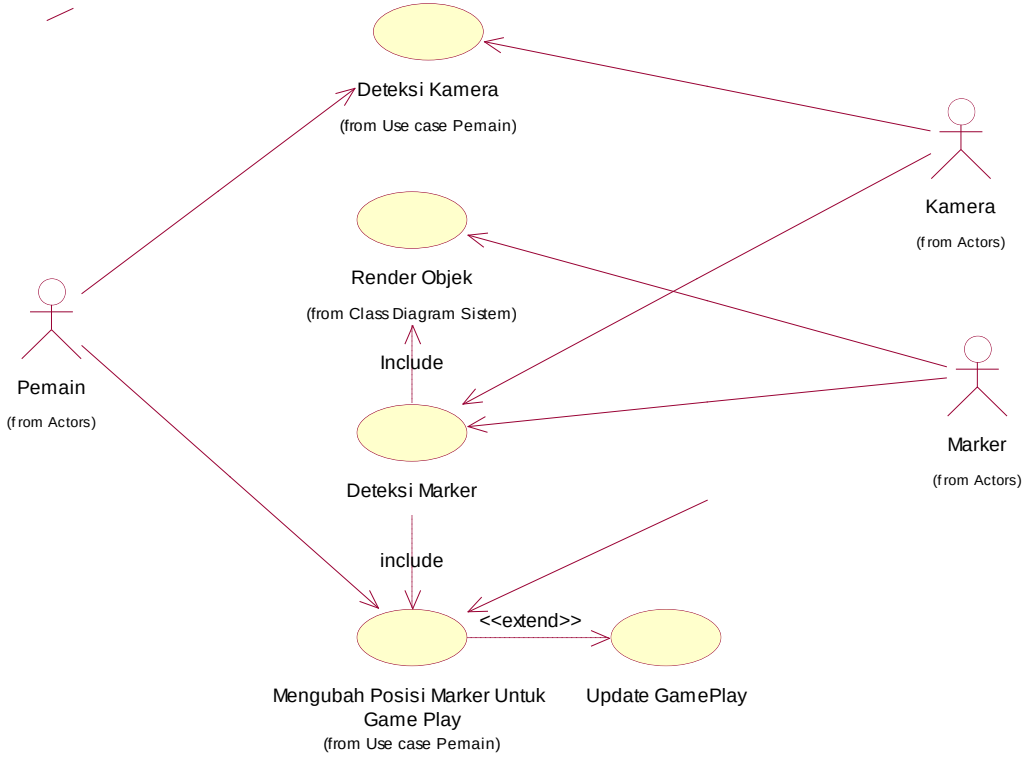
- Perangkat lunak digunakan dalam sebuah sistem merupakan kumpulan perintah-perintah yang diberikan kepada perangkat keras agar saling berinteraksi untuk melakukan suatu tugas. Perangkat lunak yang dibutuhkan untuk membangun aplikasi adalah sebagai berikut :

- Sistem operasi Windows Xp pro 2
- FlashDevelop Version 4.0.0 RC3
- Adobe Flash CS5
- Adobe Dreamweaver CS5



# PEMODELAN

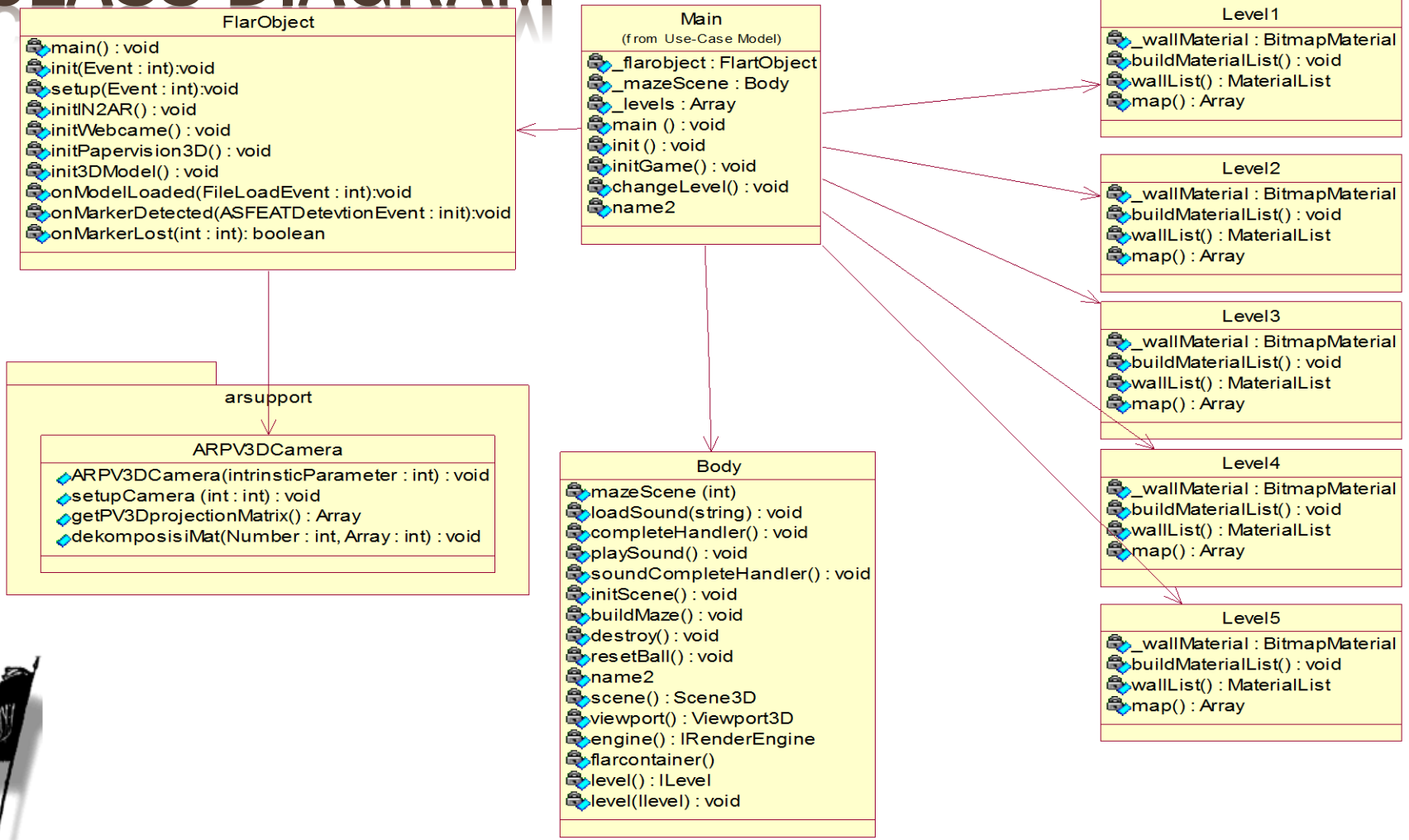
- Use case



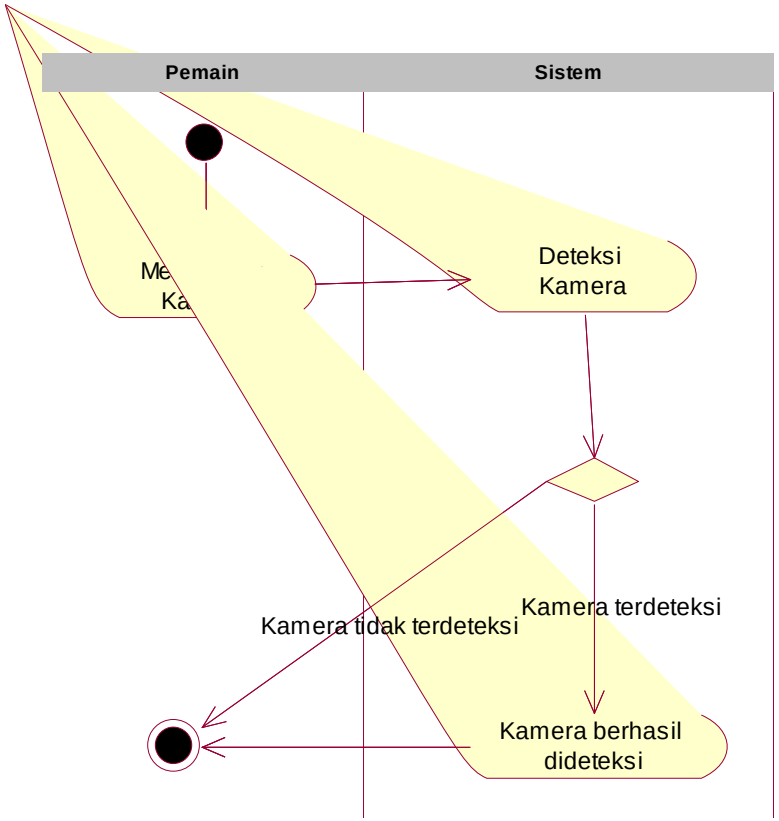



# CLASS DIAGRAM



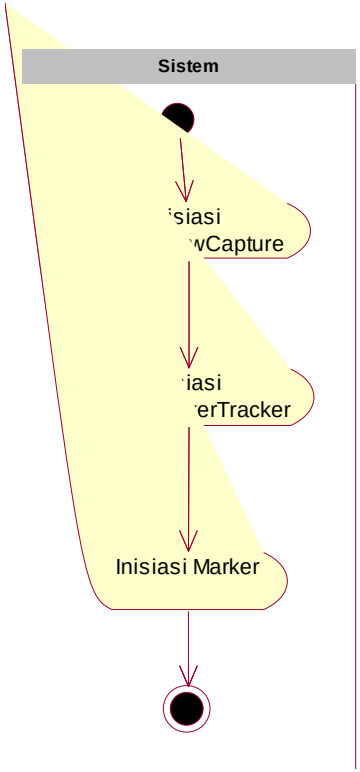


# ACTIVITY DIAGRAM MENDETEKSI KAMERA



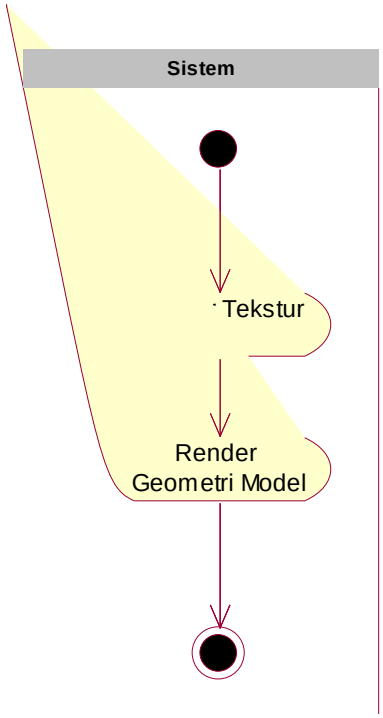



# ACTIVITY DIAGRAM MENDETEKSI MARKER



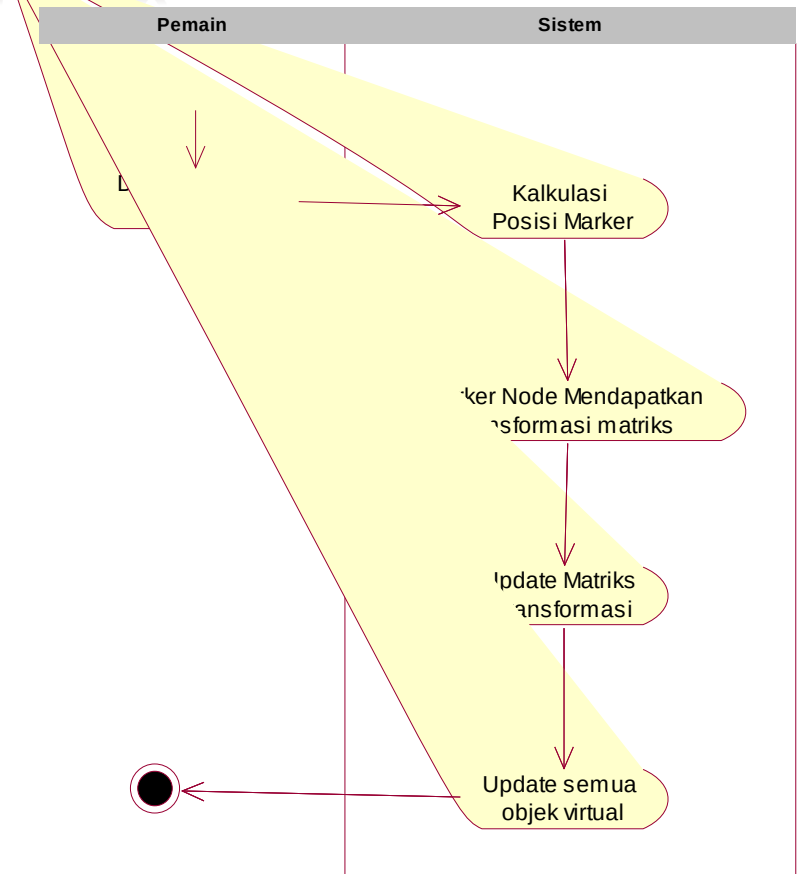



# ACTIVITY DIAGRAM RENDER OBYEK

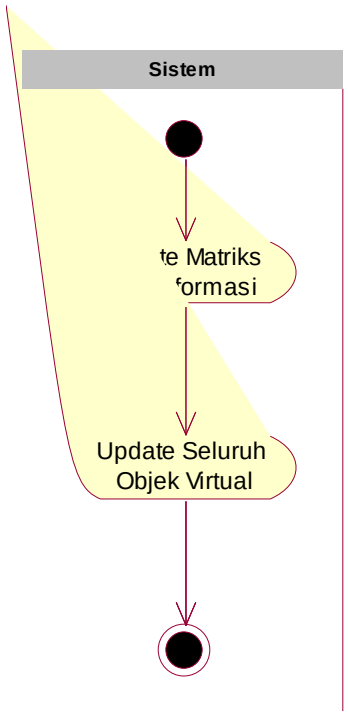





# ACTIVITY DIAGRAM MENGUBAH POSISI MARKER UNTUK GAMEPLAY

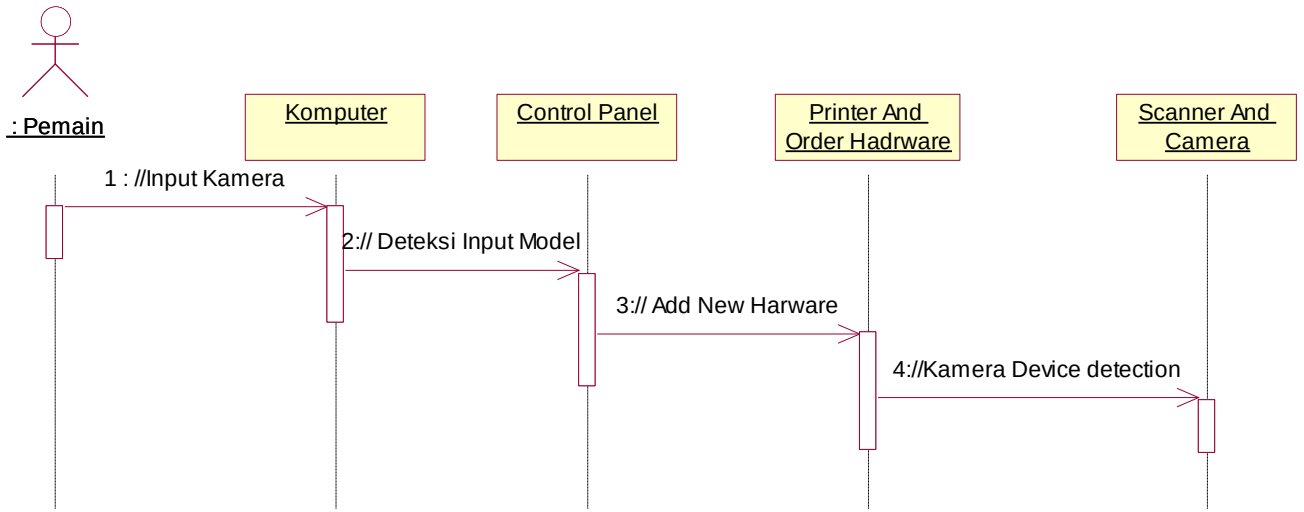


# ACTIVITY DIAGRAM UPDATE GAMEPLAY



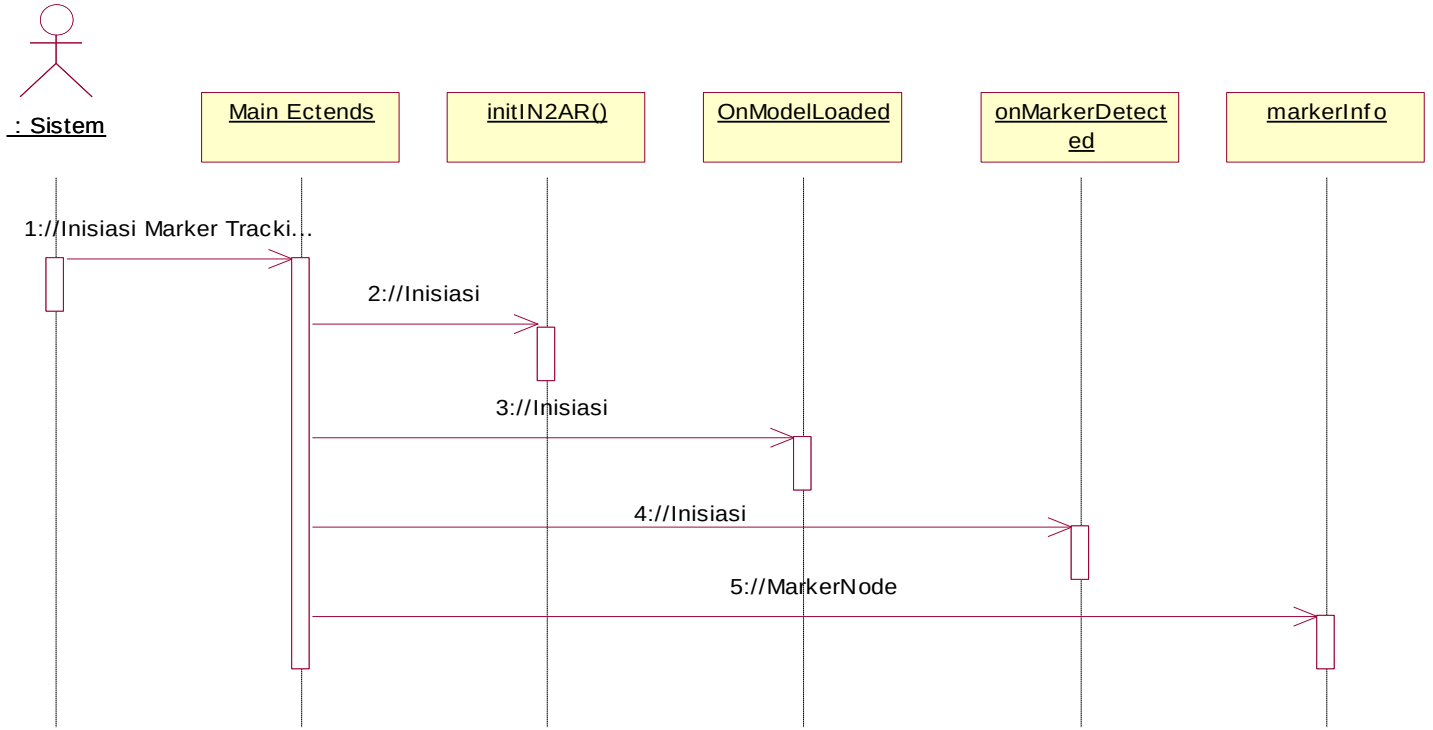



# SEQUENCE DIAGRAM KAMERA



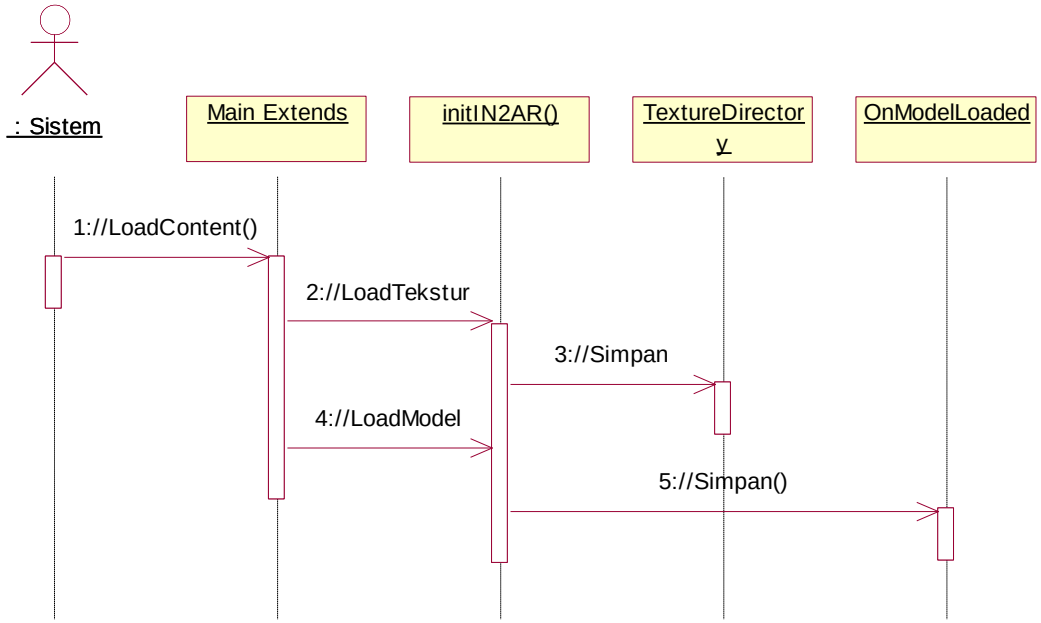



# SEQUENCE DIAGRAM DETEKSI MARKER



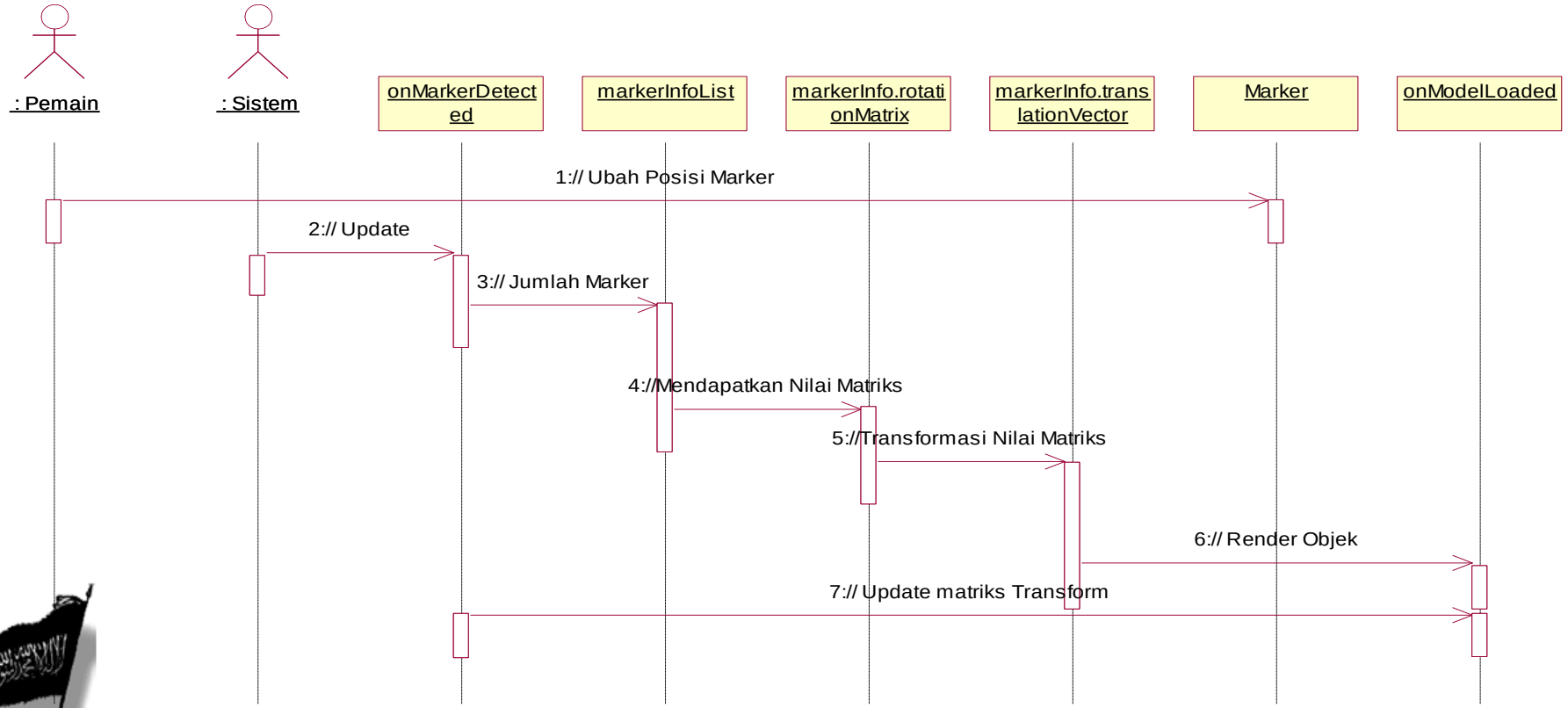



# SEQUENCE DIAGRAM RENDER OBYEK



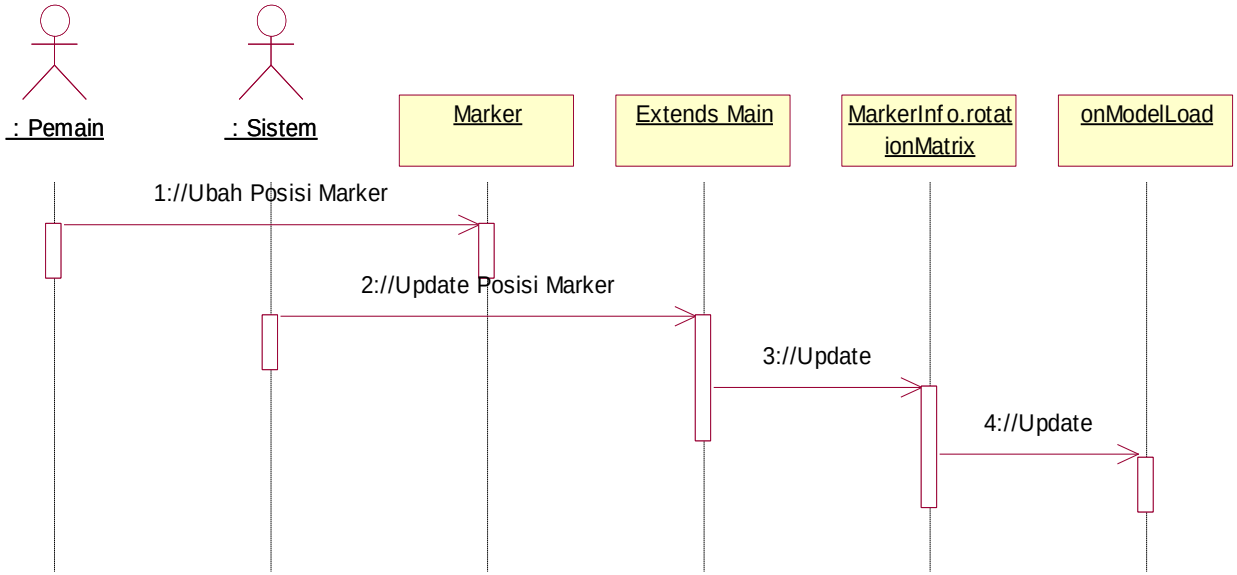



# SEQUENCE DIAGRAM MENGUBAH POSISI MARKER UNTUK GAMEPLAY

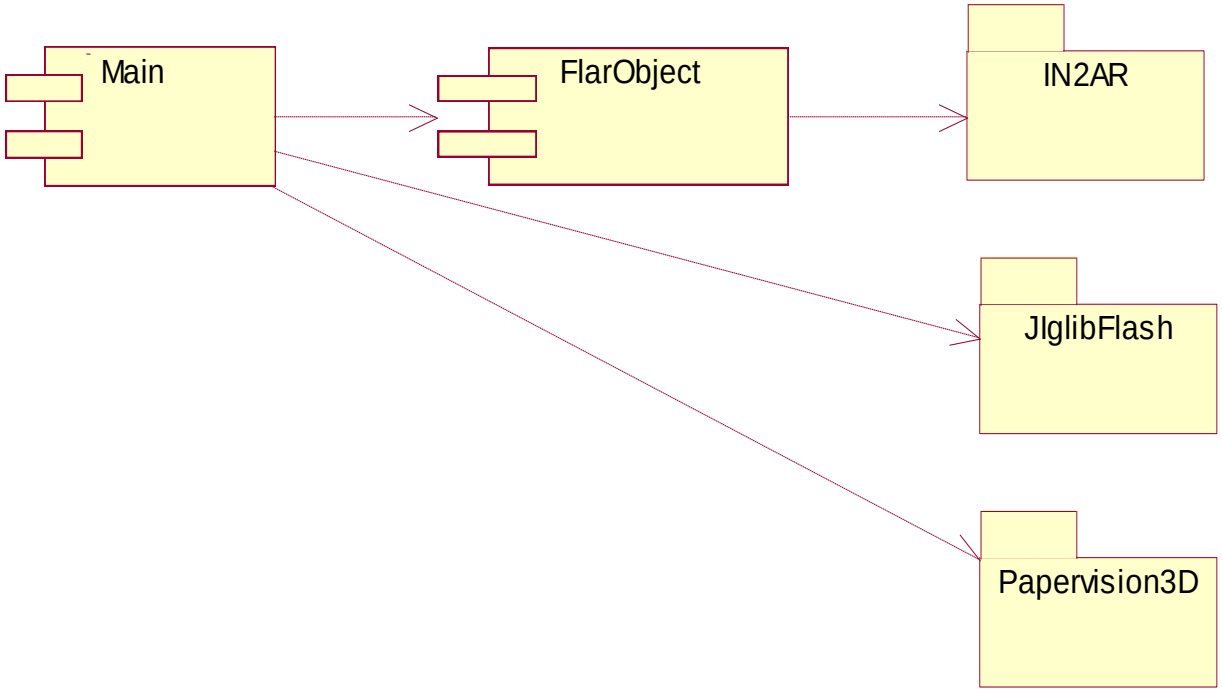




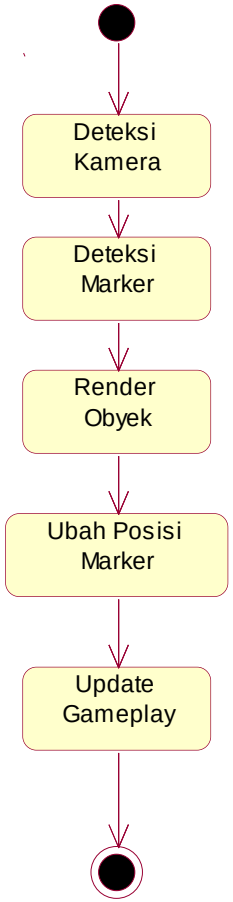

# SEQUENCE DIAGRAM UPDATE GAMEPLAY



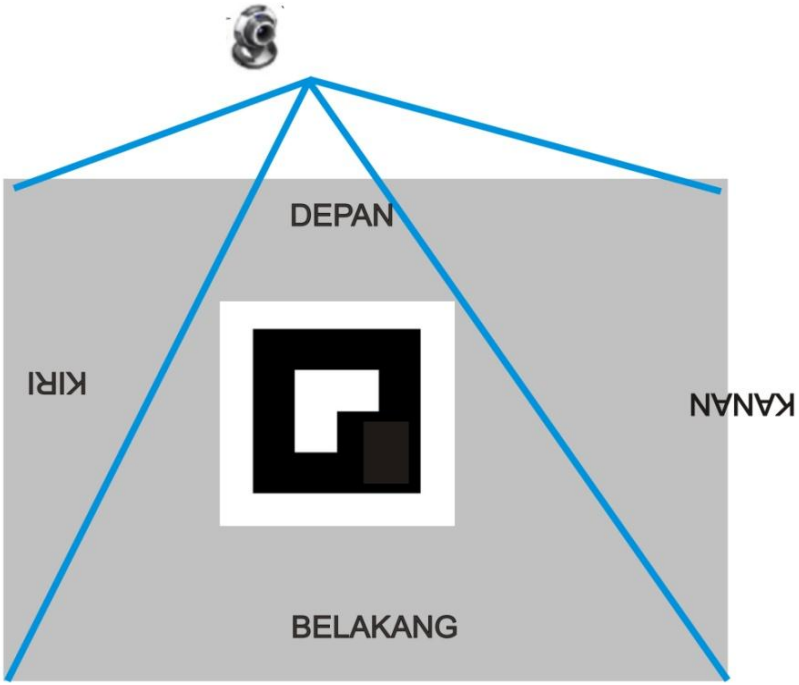
# COMPONENT DIAGRAM



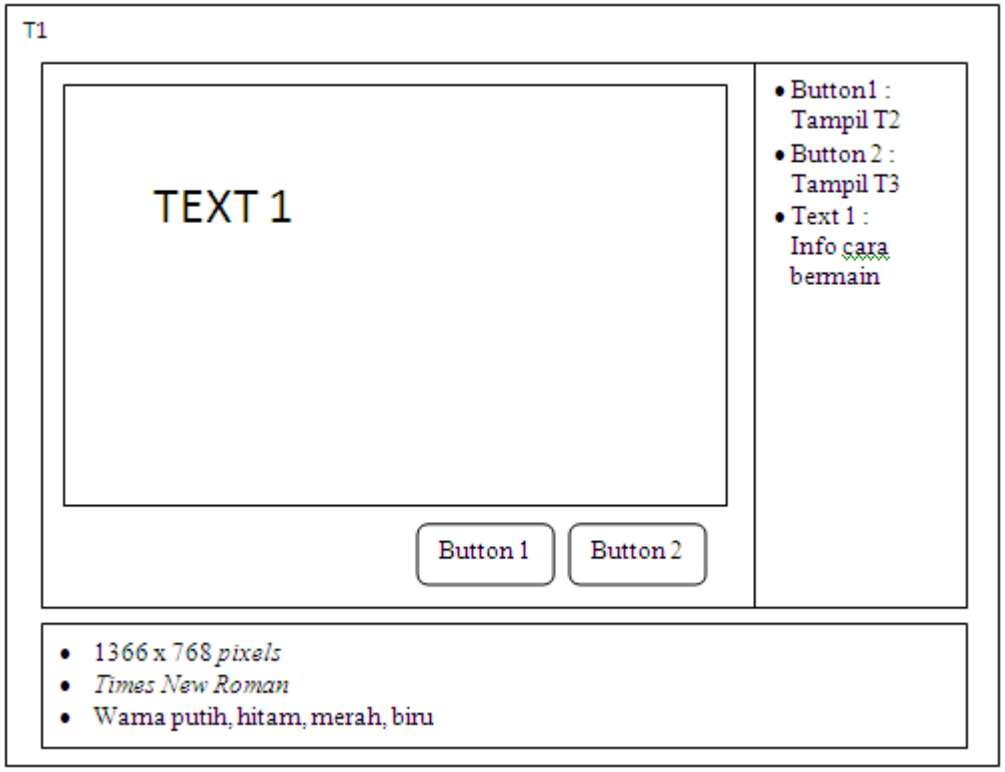
# STATE MACHINE DIAGRAM GAME



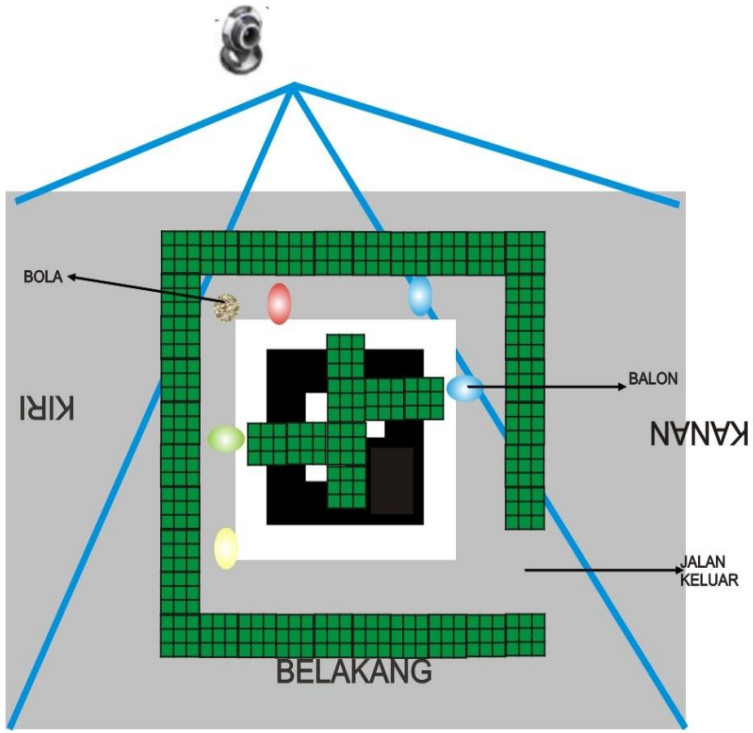
# PERANCANGAN MARKER



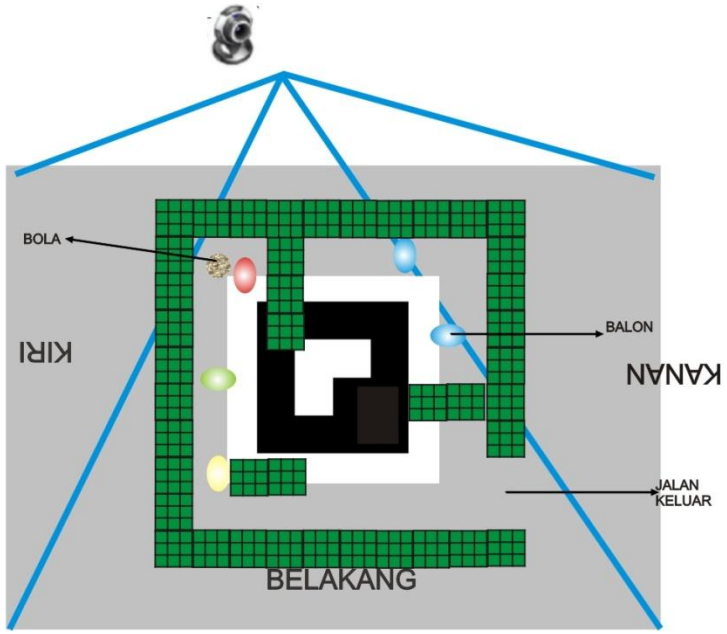
# PERANCANGAN ANTAR MUKA



# PERANCANGAN CARA BERMAIN

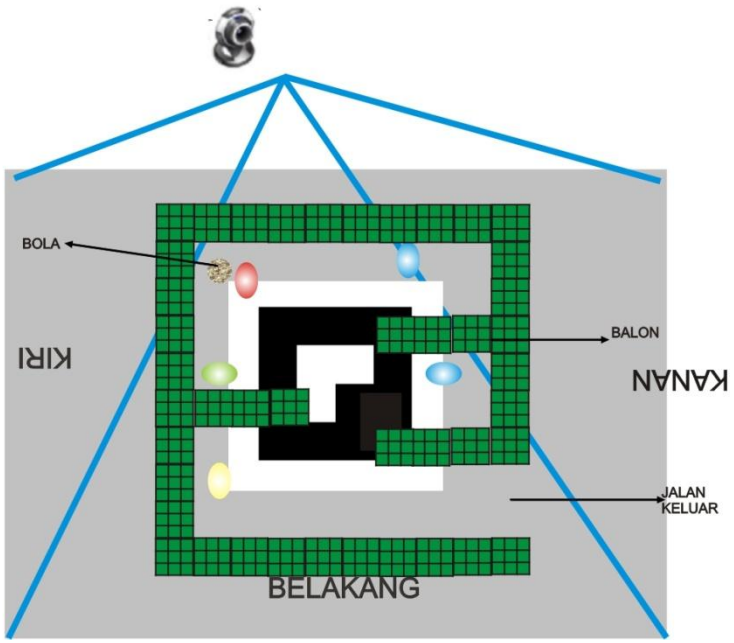



Level 1

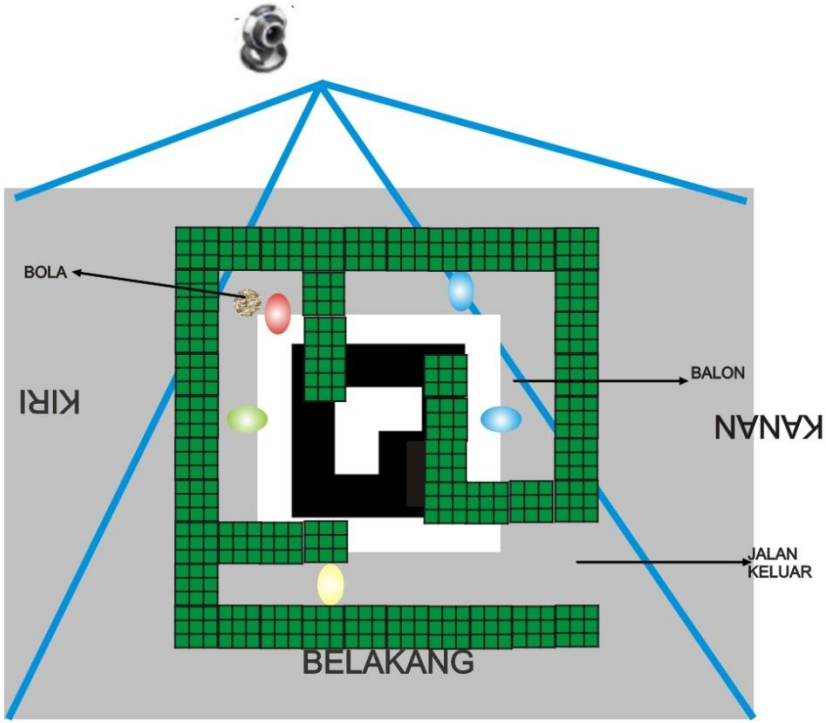



Level 2

# PERANCANGAN CARA BERMAIN

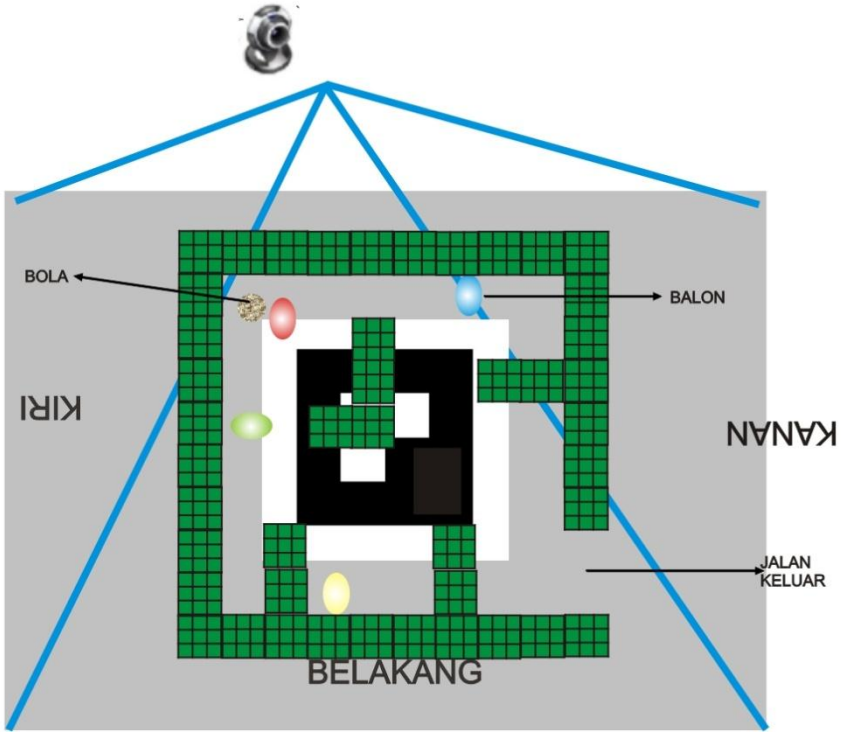



Level 3




Level 4

# PERANCANGAN CARA BERMAIN




Level 5

# IMPLEMENTASI PERANGKAT KERAS

| Komputer | Spesifikasi perangkat keras |
|---|---|
| Prosesor | Intel core 2Duo 1.66 Ghz |
| Layar | Resolusi 1280 x 800 |
| Monitor | LCD 14 inc |
| Memori | Memori 2 GB |
| Harddisk | Harddisk 60 GB |
| Keyboard dan mouse | - |

# IMPLEMENTASI PERANGKAT LUNAK

| Komputer | Spesifikasi perangkat lunak |
|---|---|
| Sistem Operasi | Microsoft Window Xp |
| Bahasa Pemograman | ActionScript 3.0 (FlashDevelop) |

# IMPLEMENTASI *AUGMENTED REALITY*

- Untuk fokus okulasi akan dilakukan pengujian *marker tracking* ketika sebagian marker tertutup halangan.




Marker terdeteksi dengan baik




Marker tidak terdeteksi

# PENGUJIAN JARAK

| Jarak (Meter) | Hasil Marker Tracking |
|---|---|
| 3 | Gagal |
| 2 | Gagal |
| 1 | Terdeteksi, namun tejadi kegagalan *colition detection*. |
| 0.6 | Terdekteksi, Namun kurang collition detectin tidak konsisten |
| 0,5 | Terdekteksi, Namun kurang collition detectin tidak konsisten |
| 0.4 | Terdekteksi, Namun kurang collition detectin tidak konsisten |
| 0.3 | Terdeteksi dengan baik |
| 0.2 | Terdeteksi dengan baik |

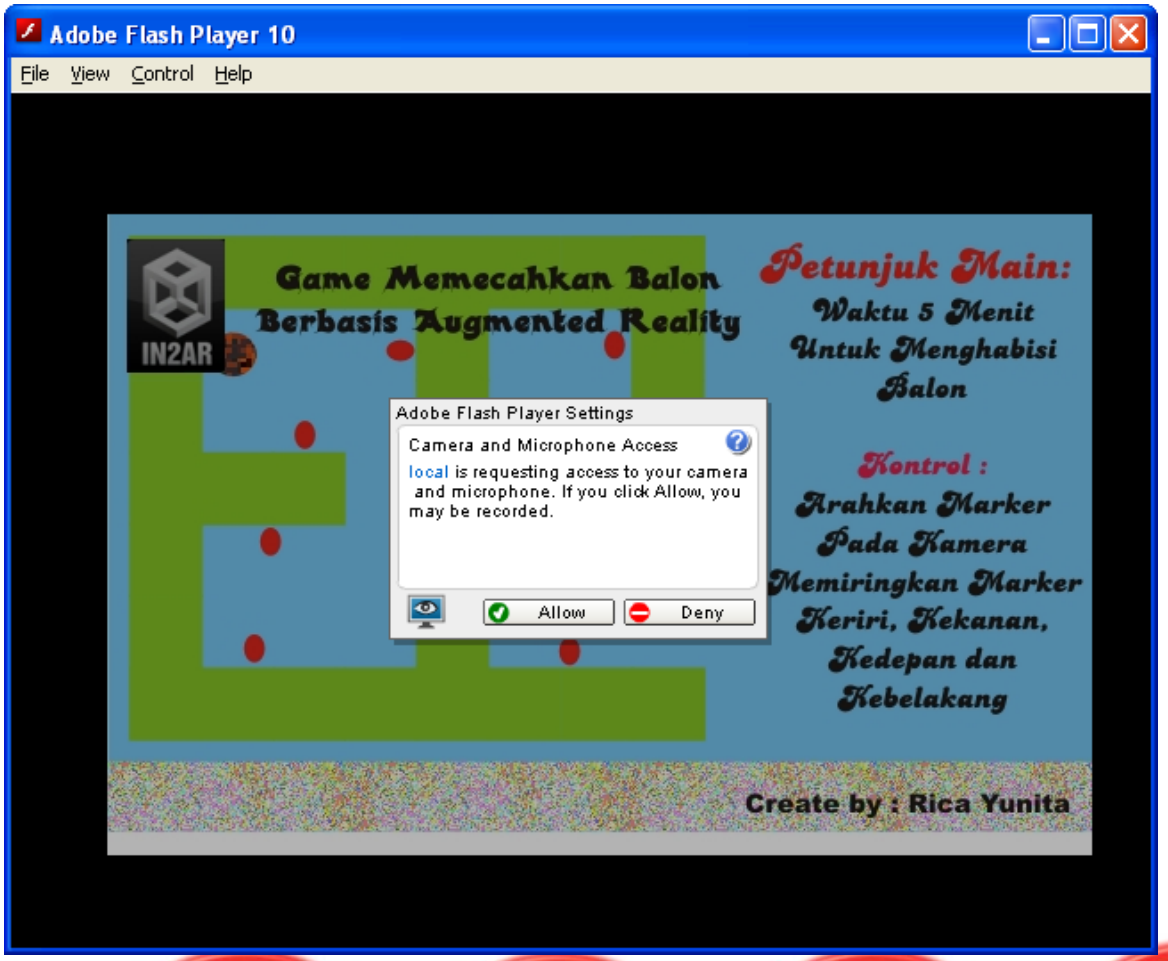
# PENGUJIAN JARAK

- Pada pengujian jarak penggunaan marker, semakin dekat jarak marker dengan kamera akan mengakibatkan ukuran marker yag terdeteksi semakin besar, sehingga bisa tertangkap dengan baik. Namun ketika jark kamera dengan marker semaikin jauh maka ukuran marker yang tertangkap kamera semakin kecil, sehingga pola marker manjadi tidak jelas dan mengakibatkan marker tidak terdeteksi.

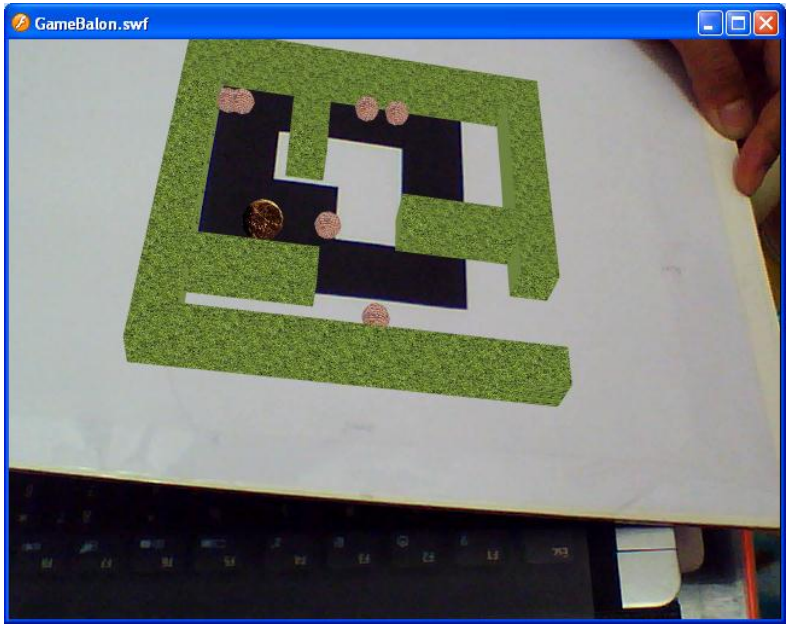
# PENGUJIAN RESOLUSI

| Resolusi | Rata-rata *Frame Rate* |
|---|---|
| 320x240 | 46,8 |
| 640x480 | 36,1 |
| 800x450 | 35,8 |
| 800x600 | 33,3 |

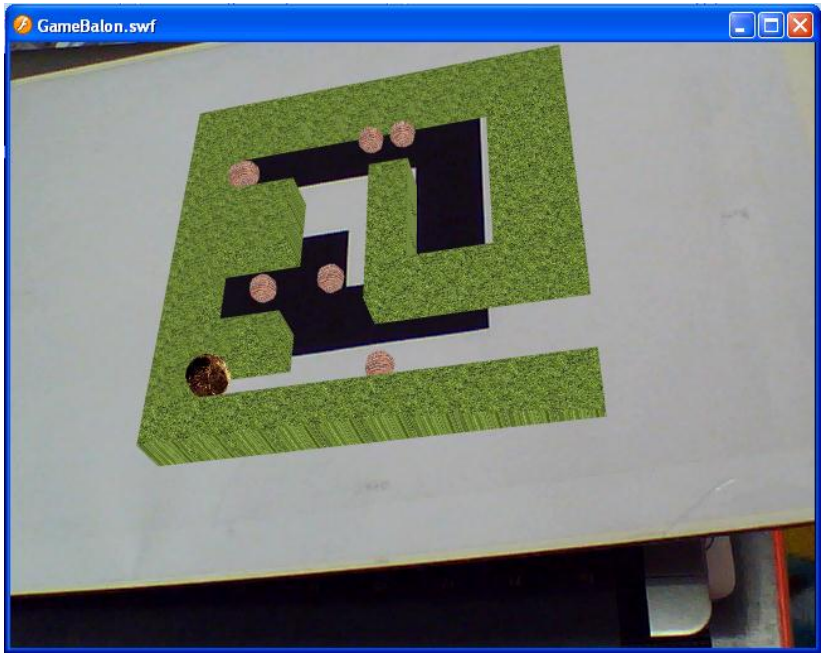
# PENGUJIAN DETEKSI KAMERA



# PENGUJIAN GERAK BOLA

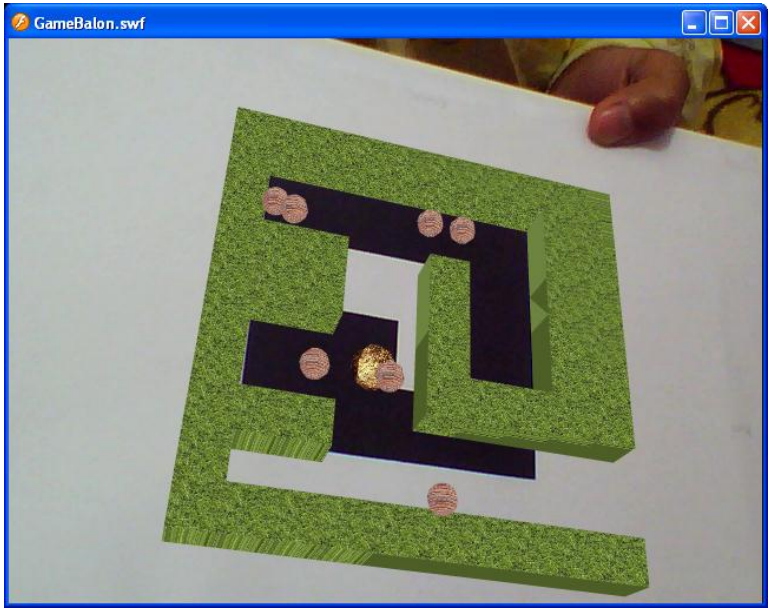



Gerak kekanan




Gerak Kekiri

# PENGUJIAN GERAK BOLA




Gerak Kebawah




Gerak Keatas

# KESIMPULAN

1. Permainan dapat ditampilkan secara virtual (*augmented reality*)
2. *Tracking marker* dan gerak bola udah berhasil dijalankan.
3. Seluruh level permainan sudah dapat dijalankan.
4. Posisi kamera dan pencahayaan mempengaruhi kinerja aplikasi.
5. *Game* Memecahkan Balon Berbasis *Augmented Reality* merupakan permainan yang dimainkan oleh satu orang dengan tujuan untuk melatih otot motorik halus anak.
6. *Collition Detection* yang dihasilkan oleh permainan dipengaruhi oleh cahaya, resolusi kamera dan posisi marker.
7. *Matrix Euler* yang diterapkan sebagai kontrol permainan dapat berjalan dengan baik.

# SARAN

1. Aplikasi yang dibuat lebih optimal jika kamera yang digunakan menggunakan resolusi 5 migapixel.
2. RAM computer yang digunakan sebaiknya berkapasistas minimal 3 GB.
3. Aplikasi dijalankan ditempat dengan factor pencahayaan normal.
4. Aplikasi minimal dijalankan dengan Flash Player 10.
5. *Game* Memecahkan Balon Berbasis *Augmented Reality* dapat dikembangan menjadi game berbasis android.
6. *Game* Memecahkan Balon Berbasis *Augmented Reality* juga dapat dikembangkan dengan menanamnya kedalam *Handphone*, dengan layar *Handphone* sebagai monitor dan kamera pada *Handphone* sebagai kamera untuk menangkap gambar marker.

# Terima Kasih



