BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis merupakan tahapan yang paling penting, karena kesalahan dalam tahap ini akan menyebabkan kesalahan di tahap selanjutnya.

Dari hasil analisis tersebut dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efisien. Pada tahap analisis sistem ini dibagi menjadi beberapa bagian, yaitu sebagai berikut :

1. Analisis Masalah.
2. Analisis Sistem yang Sedang Berjalan
3. Analisis Arsitektur Sistem
4. Analisis Metode
5. Analisis Materi
6. Analisis Kebutuhan Non Fungsional
7. Analisis Kebutuhan Fungsional

3.1.1 Analisis Masalah

Analisis masalah adalah langkah awal dari analisis sistem. Langkah ini diperlukan untuk mengetahui pemasalahan apa saja yang terjadi didalam sistem yang telah berjalan. Setelah melakukan wawancara dengan guru mata pelajaran IPA kelas 4, dalam proses belajar mengajar mata pelajaran IPA, khususnya pada materi Rangka Manusia serta Struktur dan Fungsi Bagian Tumbuhan masih terdapat beberapa kendala. Guru mengalami kesulitan dalam mengimplementasikan materinya ke dalam sebuah media bantu, karena alat peraga ataupun fasilitas yang

mendukung pelajaran IPA yang terdapat disekolah masih kurang. Hal ini dapat mengakibatkan materi yang disampaikan oleh guru tidak dapat diserap atau dipahami secara maksimal oleh siswa.

Dari kuesioner yang diajukan kepada 22 siswa kelas 4, didapat hasil sebagai berikut. Sebanyak 20 siswa waktu belajar di rumah kurang dari 1 jam. Hal ini menandakan bahwa minat baca dan belajar siswa-siswi masih kurang. Ketika sedang menggunakan komputer, sebanyak 13 siswa menggunakan komputer bukan untuk belajar, melainkan untuk bermain game. Sebagian besar siswa memilih media interaktif seperti animasi dan video untuk mendukung kegiatan belajar mereka dalam mata pelajaran IPA. Sebanyak 22 siswa menyatakan jika soal-soal latihan dapat membantu mereka dalam mempelajari pelajaran IPA.

3.1.2 Analisis Sistem yang Sedang Berjalan

Mata pelajaran Ilmu Pengetahuan Alam (IPA) merupakan salah satu mata pelajaran pokok yang diberikan kepada semua siswa-siswi di Sekoah Dasar (SD). Yaitu mulai dari siswa kelas satu sampai kelas enam. Materi yang diberikan sesuai dengan kurikulum saat ini, yaitu Kurikulum Tingat Satuan Pendidikan (KTSP). KTSP dikembangkan oleh Badan Standar Nasional Pendidikan (BSNP) yang mengacu pada standar nasional pendidikan untuk menjamin pencapaian tujuan pendidikan nasional, dimana pada setiap tingkatan kelas dibagi menjadi dua semester. Pelajaran diberikan mulai dari tingkat dasar pengenalan Ilmu Pengetahuan Alam (IPA) bagi siswa kelas satu SD sampai tingkat yang lebih detail bagi siswa kelas enam SD.

Sistem pengajaran mata pelajaran IPA dilakukan dengan tatap muka secara langsung dikelas. Teknik pengajaran yang biasa dilakukan banyak dalam bentuk teori dengan media buku pelajaran, papan tulis, beberapa alat peraga seperti patung rangka dan miniatur tata surya, serta diselingi dengan sedikit praktek yaitu pada pertemuan terakhir setiap bab pembahasan selesai. karena kurangnya sarana dan prasarana untuk melakukan praktek. Sistem pengajaran yang masih secara konvensional membuat sebagian siswa cepat merasa jenuh dalam belajar IPA.

Untuk mengevaluasi hasil belajar siswa, guru mengadakan tes (ulangan) secara tertulis dengan mengerjakan soal-soal yang diberikan oleh guru stidaknya tiga kali ulangan dalam satu semester, satu kali ulangan umum di pertengahan semester (UTS), dan satu kali ulangan umum diakhir semester (UAS). Selain itu siswa juga diberikan pekerjaan rumah baik secara individu maupun secara kelompok agar siswa tidak cepat bosan dalamn belajar IPA.

Struktur materi mata pelajaran IPA kelas 4 SD yang akan dibahas adalah sebagai berikut :

Bab I Rangka Manusia

A. Rangka Badan
    a. Tulang Belakang
    b. Tulang Rusuk dan Tulang Dada
    c. Tulang Gelang Bahu
    d. Tulang Gelang Panggul
B. Rangka Anggota Gerak
    a. Tulang Tangan
    b. Tulang Kaki

Bab II Struktur dan Fungsi Bagian Tumbuhan

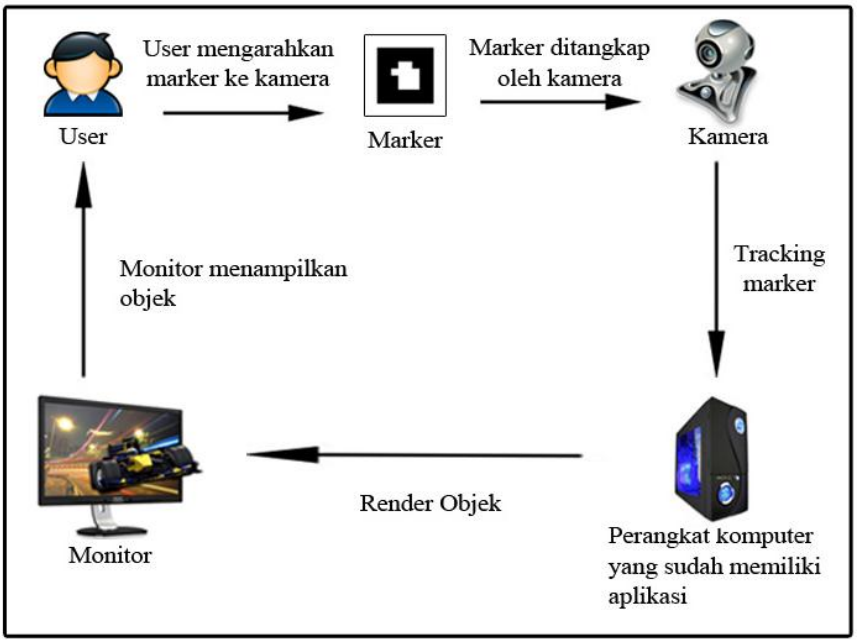
A. Struktur Akar dan Fungsinya
B. Struktur batang dan Fungsinya
C. Struktur Daun dan Fungsinya
D. Struktur bunga dan Fungsinya

3.1.3 Analisis Arsitektur Sistem

Sistem yang akan dibangun adalah suatu aplikasi Pengenalan rangka dan bagian tumbuhan dengan menggunakan teknologi *augmented reality*. Proses aplikasi pengenalan rangka dan bagian tumbuhan dengan menggunakan teknologi augmented reality yang akan dibangun menggunakan sistem marker yang disupport oleh Lib FLARToolkit.

Pada arsitektur aplikasi yang akan dibangun terdiri dari beberapa komponen, yaitu : *user* adalah pengguna yang menggunakan aplikasi pengenalan

rangka dan bagian tumbuhan menggunakan teknologi *augmented reality*, *user* mengarahkan *marker* sehingga *marker* dapat tertangkap olah kamera. Kemudian dari gambar yang didapat dari kamera sistem komputer melakukan tracking marker untuk mengidentifikasi *marker* yang digunakan oleh pengguna. Komputer melakukan render objek 3D yang digunakan dalam aplikasi. *User* dapat melihat hasil manipulasi system melalui layar komputer/ monitor. Gambar 3.1 menunjukan deskripsi proses sistem yang akan dibangun.



Gambar 3.1 Arsitektur Sistem

3.1.4 Analisis Alur Sistem

Analisis alur sistem merupakan analisis yang mendeskripsikan bagaimana proses augmented reality dari awal inisialisasi, tracking marker, sampai dengan proses rendering objek 3D dengan metode marker yang diterapkan oleh FLARToolkit. Dalam perancangan aplikasi dengan teknologi AR, seolah-olah menggabungkan objek virtual dengan objek nyata, dalam hal ini objek virtual berupa objek 2D atau 3D dan objek nyatanya berupa gambar dengan pola tertentu (marker).

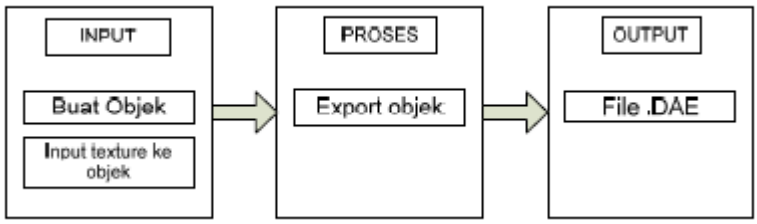
Secara garis besarnya, dalam perancangan ada tiga bagian utama yaitu sebagai berikut :



Gambar 3.2 Alur sistem

3.1.4.1 Inisialisasi Model 3D

Pada tahap ini ditentukan marker yang akan digunakan, sumber input video nya, dan objek 3D yang akan digunakan. Pada bagian inisialisasi ini, objek 3D diinisialisasi terlebih dahulu karena loading objek 3D memerlukan waktu yang cukup lama. Model 3D yang akan ditampilkan di-load terlebih dahulu. Agar aplikasi dapat menampilkan objek 3D tertentu tanpa merubah atau membangun ulang aplikasi, diperlukan sebuah file konfigurasi untuk menentukan objek 3D yang akan di-load sesuai dengan pola marker yang dideteksi.



Gambar 3.3 Proses pembentukan data objek 3D



Gambar 3.4 Proses export objek

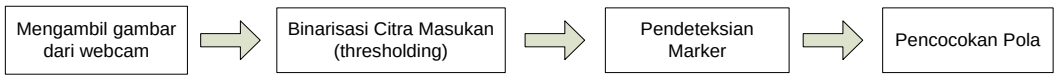
Dalam proses pemodelan objek terdiri dari 3 langkah.

1. Menyesuaikan objek 3D dengan animasi atau bentuk yang akan dibuat.
2. Memasukan teksture sesuai dengan objek 3D.
3. Mengexport objek yang sudah dirancang dan dibuat kedalam format collada (*.DAE).

Pada tahapan ketiga yaitu mengexport objek kedalam format collada (*.DAE), tidak bisa dilakukan secara manual dengan menggunakan export bawaan dari aplikasi 3d pembuat objek. Export objek harus terlebih dahulu meng-install applikasi OpenCOLLADA agar objek dapat ditampilkan dan sesuai dengan yang dibuat.

3.1.4.2 Tracking Marker

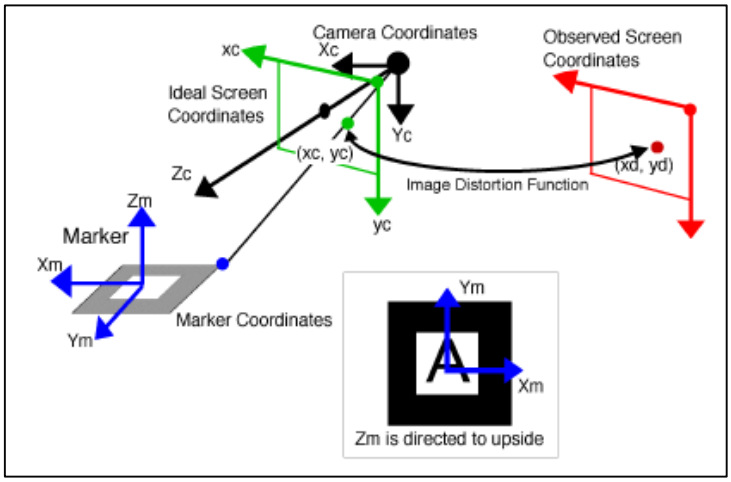
FLARToolkit memiliki kemampuan untuk mendeteksi gambar dan menghitung posisi gambar tersebut menggunakan webcam standar. Informasi posisi yang didapatkan akan dipergunakan untuk menempatkan objek atau model tiga dimensi atau video ke dalam posisi marker. Ada empat langkah, dalam proses kerja Tracking marker FLARToolkit.



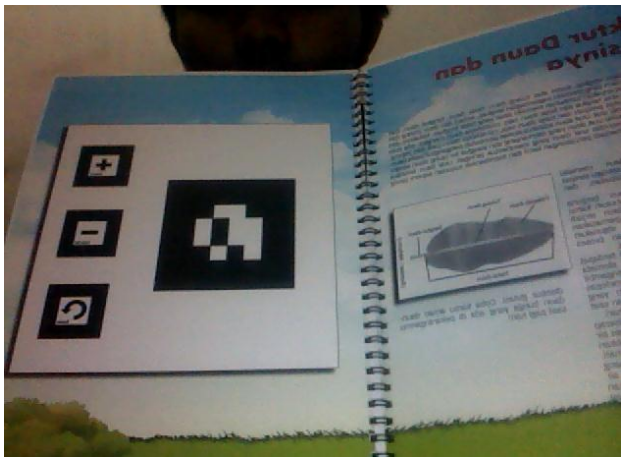
Gambar 3.5 Proses kerja Tracking marker FLARToolkit

1. Mengambil gambar dari webcam

Mendapatkan masukan gambar dari sebuah webcam adalah langkah awal yang harus dilakukan, seperti yang ditunjukan gambar dibawah ini. Sistem mengolah dan menganalisis frame per frame video yang di-streaming secara real-time dan hasilnya berupa citra digital yang akan digunakan untuk tahap berikutnya.



Gambar 3.6 Sistem Koordinat Lingkungan AR



Gambar 3.7 Mengambil gambar dari webcam

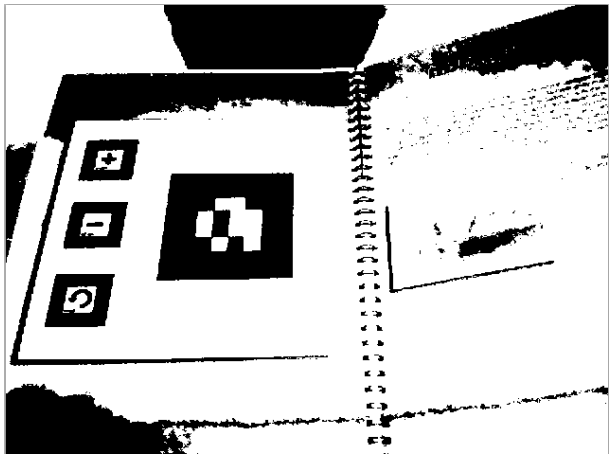
2. Binarisasi citra masukan

Setelah mengambil gambar dengan webcam, kemudian gambar tersebut diubah menjadi gambar grayscale. Setelah diubah menjadi gambar grayscale, kemudian diubah menjadi gambar binary. Gambar binary yaitu gambar yang hanya mempunyai dua nilai derajat keabuan yaitu hitam dan putih. Thresholding mengkonversi citra ke citra binari sehingga memudahkan untuk komputasi. Sebuah citra binari dibuat dengan mengubah pixel yang lebih cerah daripada nilai threshold ke suatu warna, dan pixel yang lebih gelap daripada nilai threshold ke suatu warna lainnya (didefinisikan sebagai gray-scale atau hitam-putih). Nilai threshold berada

pada angka 0 - 255 dan secara default, threshold bernilai 100. Fungsi dari proses ini adalah untuk membantu sistem agar dapat mengenali bentuk segi empat dan pola di marker pada citra yang diterima. Setiap *pixel* di dalam citra dipetakan ke dua nilai, 1 atau 0 dengan fungsi pengambangan:

$$f_B(i,j) = \begin{cases} 1, & f_g(i,j) \le T \\ 0, & \text{lainnya} \end{cases}$$

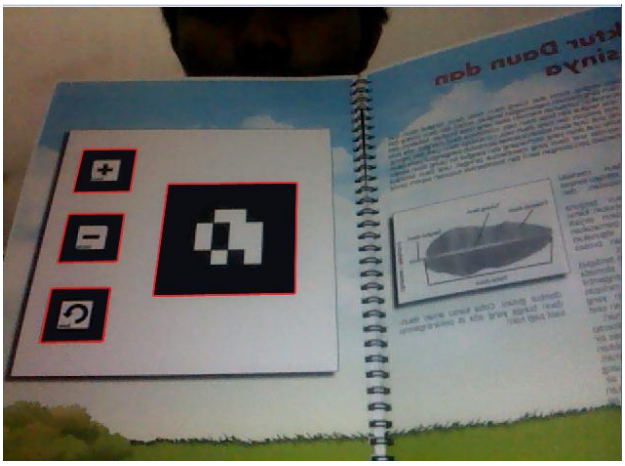
yang dalam hal ini, $fg(i, j)$ adalah citra hitam-putih, $fB(i, j)$ adalah citra biner, dan T adalah nilai ambang yang dispesifikasikan. Dengan operasi pengambangan tersebut, objek dibuat berwarna gelap (1 atau hitam) sedangkan latar belakang berwarna terang (0 atau putih).



Gambar 3.8 Binarisasi citra masukan

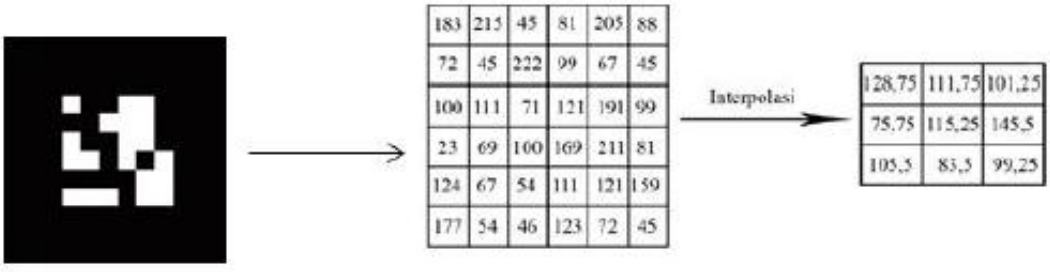
3. Pendeteksian pelacakan marker

Langkah berikutnya dari FLARToolkit adalah menemukan area yang berdampingan dalam citra yang di-threshold. Area yang berdampingan diberi tanda sebagai persegi (marker outline).



Gambar 3.9 Pendeteksian pelacakan marker (*marker detection tracking*)

Setelah proses pelacakan marker selanjutnya masuk ketahap scaling. *Scaling* merupakan proses mengubah ukuran gambar digital, perubahan ukuran gambar sangat penting karena pada sistem pengenalan marker membutuhkan data gambar yang seragam ukurannya. Tahap scaling dapat dilakukan dengan menggunakan properti *width* dan *height* atau properti skala seperti scalex dan scaley. Sebagai contoh perhitungan *matriks image* dengan angka-angka piksel *sampel* dapat dilihat di dalam gambar tabel matrik di bawah ini :



Nilai piksel pada koordinat pada citra hasil interpolasi diperoleh dengan menghitung nilai rata – rata dari 4 nilai piksel pada citra asli, yaitu :

Tabel 3.1 Perhitungan Nilai Piksel Hasil Interpolasi

Nilai Piksel Citra Hasil Asli	Nilai Piksel Citra Interpolasi
(183+215+72+45) / 4	128,75
(100+111+23+69) / 4	75,75
(124+67+177+54) / 4	105,5
(45+81+222+99) / 4	111,75
(71+121+100+169) / 4	115,25
(54+111+46+123) / 4	83,5
(205+88+67+45) / 4	101,25
(191+99+211+81) / 4	145,5
(121+159+72+45) / 4	99,24

4. Pencocokan Pola

Setelah semua area persegi dan pola-pola gambar ditandai, FLARToolkit menganalisa citra yang berada di dalam persegi dan membandingkan polanya dengan sekumpulan pola yang telah ditentukan (pencocokan pola). FLARToolkit mengekstrak pola didalam persegi menggunakan transformasi homography. FLARToolkit memberikan nilai confidance kepada pola yang cocok, jika kecocokannya di atas nilai yang telah ditentukan maka polanya dinyatakan cocok.

Algoritma transformasi *homography* FLARToolkit

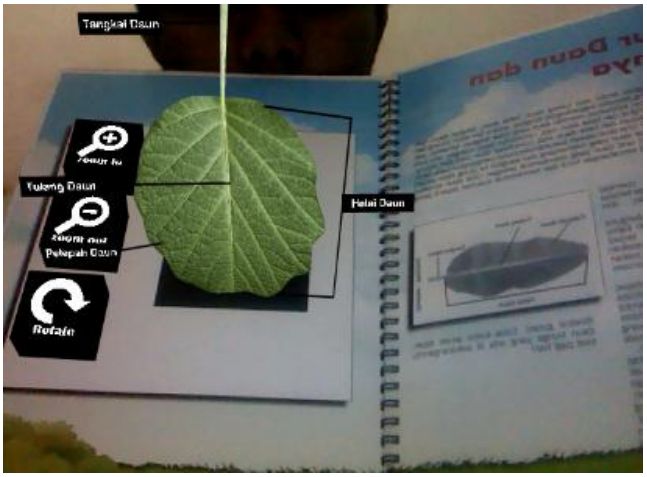
```
Procedure transformasi homography
{IS : nilai koordinat citra ,serta elemen matriks sudah
terdefinisi }
{FS : mencocokan pola dari gambar yang sudah ditandai }
Kamus :
  m11,m12,m13,m21,m22,m23,m31,m32,m33 : number
  x, y : integer
  sigma : real
Algoritma :
m11 <- 1 {matriks}
m12 <- 0
m13 <- 0
m21 <- 0
m22 <- 1
m23 <- 0
m31 <- 0
m32 <- 0
m33 <- 1
function identity() {proses identivikasi}
m11 <- m22 <- m33 <- 1
m12 <- m13 <- m21 <- m23 <- m31 <- m32 <- 0
function projectPoint(point:Point)
```

```
Number x <- point x
Number y <- point y
Number z <- 1.0 / m31 x x+m32 x y+m33
point x <- (m11 x (x) + m12 x y + m13) x z
point x <- (m11 x (x) + m12 x y + m13) x z
return point

function scale(value:Number)
m11 *= value
m12 *= value
m13 *= value
m21 *= value
m22 *= value
m23 *= value
//m31 *= value
//m32 *= value
//m33 *= value
function transpose()
m12 = _m21
m13 = _m31
m21 = _m12
m23 = _m32
m31 = _m13
m32 = _m23
function multiply(mat:HomographyMatrix)
return new HomographyMatrix(Vector.<Number>([ m11 *
mat.m11 + m12*mat.m21 + m13*mat.m31,
m11*mat.m12 + m12*mat.m22 + m13*mat.m32,
m11*mat.m13 + m12*mat.m23 + m13*mat.m33,
m21 * mat.m11 + m22*mat.m21 + m23*mat.m31,
m21*mat.m12 + m22*mat.m22 + m23*mat.m32,
m21*mat.m13 + m22*mat.m23 + m23*mat.m33,
m31 * mat.m11 + m32*mat.m21 + m33*mat.m31,
m31*mat.m12 + m32*mat.m22 + m33*mat.m32,
m31*mat.m13 + m32*mat.m23 + m33*mat.m33]))
{membuat matriks homography baru}
function invert(invertedMatrix:HomographyMatrix)
var det:Number = m11 * (m22*m33 - m23*m32) - m12 *
(m21*m33 - m23*m31) + m13 * (m21*m32 - m22*m31);
if(det == 0) return false;
det = 1 / det
invertedMatrix.m11 = det * (m22*m33 - m23*m32)
invertedMatrix.m12 = det * (m13*m32 - m12*m33)
invertedMatrix.m13 = det * (m12*m23 - m13*m22)
invertedMatrix.m21 = det * (m23*m31 - m21*m33)
invertedMatrix.m22 = det * (m11*m33 - m13*m31)
invertedMatrix.m23 = det * (m13*m21 - m11*m23)
invertedMatrix.m31 = det * (m21*m32 - m22*m31)
invertedMatrix.m32 = det * (m12*m31 - m11*m32)
invertedMatrix.m33 = det * (m11*m22 - m12*m21)
return true
function clone()
return new HomographyMatrix( Vector.<Number>([ m11,
m12, m13, m21, m22, m23, m31, m32, m33 ]) )
```

3.1.4.3 Rendering *objek 3D*

Transformasi matriks yang dikalkulasikan di tahap sebelumnya yang digunakan FLARToolkit dan menampilkan objek yang sesuai dengan sebuah library 3D, seperti yang ditunjukkan gambar 3.9. FLARToolkit menyertakan kelas pendukung yang mengkonversikan transformasi matriks FLARToolkit ke setiap kelas matriks internal library 3D tersebut.



Gambar 3.10 Render Objek 3D

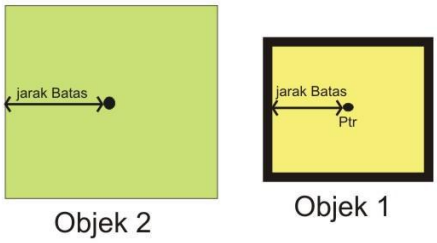
3.1.5 Analisis Metode *Occlusion*

Pada analisis metode ini menjelaskan bahwa ada tidaknya *occlusion* dalam penampilan objek 3D, secara sederhana *occlusion* mendefinisikan dimana suatu objek marker tidak dapat terdeteksi dikarenakan terhalang oleh objek lain. Akan tetapi ada beberapa persyaratan bahwa objek tersebut mengalami *occlusion* yaitu dimana persamaan 1 dan 2 terpenuhi.

$$02x - batas \leq 01x \leq 02x + batas \quad (1)$$

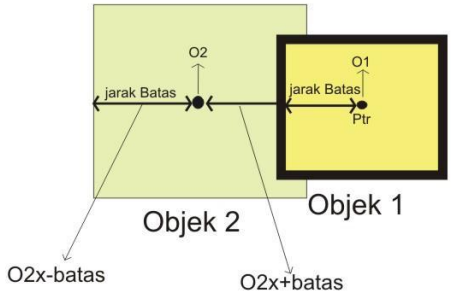
$$02y - batas \leq 01y \leq 02y + batas \quad (2)$$

Hasil deteksi ini berupa nilai kebenaran yang merupakan dasar pendefinisian *event* dari interaksi *occlusion based* jika pertidaksamaan 1 dan 2 terpenuhi. Untuk menjelaskan persamaan (1) agar tidak terjadi event akan diuraikan seperti pada Gambar 3.11. Misalkan titik tengah objek1 diwakili oleh matrik O1(x,y) dan objek 2 diwakilili oleh matrik O2(x,y).



Gambar 3.11 Objek matrik O1(x,y) dan O2(x,y)

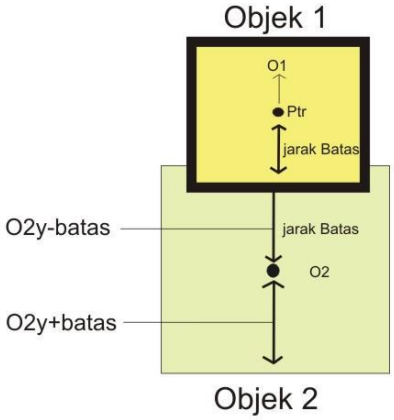
Untuk persamaan (1) tidak terjadi *event* jika :



Gambar 3.12 Objek matik O1 Tidak terjadi event terhadap persamaan (1)

O2x-batas adalah merupakan setengah bagian dari objek matrik O2 yang tidak menutupi koordinat Ptr pada Objek O1, kemudian persinggungan antara matrik O1 lebih kecil dari pada batas objek O2x+batas, maka dari hal tersebut titik objek matrik O1 tidak terjadi *event.*

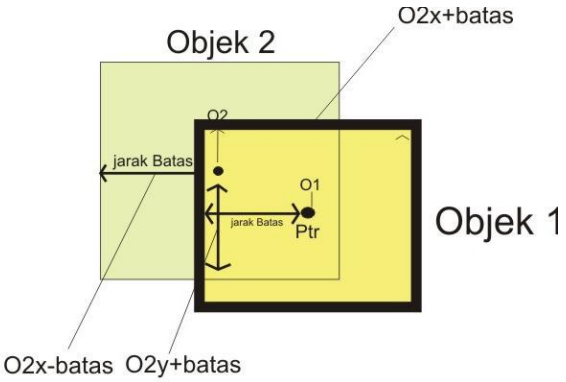
Untuk persamaan (2) tidak terjadi *event* jika :



Gambar 3.13 Objek O1 tidak terjadi event terhadap persamaan (2)

O2y-batas adalah merupakan setengah bagian dari objek matrik O2(x,y) yang tidak menutupi koordinat Ptr pada Objek O1(x,y), kemudian persinggungan antara matrik O1 lebih kecil dari pada batas objek O2y+batas, maka dari hal tersebut titik objek matrik O1 tidak terjadi *event.*

Untuk persamaan (1) dan (2) dapat tepenuhi dan terjadi *event* jika :



Gambar 3.14 Objek matrik O1(x,y) yang memenuhi persamaan (1) dan (2)

Dari Gambar 3.14 Dapat terlihat bahwa objek matrik O1(x,y) melebihi batas dari objek O2(x,y) O2x+batas, O2y+batas dan O2x-batas, O2y-batas, oleh karena itu objek matrik O1 terjadi *event* karena telah memenuhi persamaan (1) dan (2) yaitu Objek matrik O1(x,y) berada di dalam wilayah objek matrik O2(x,y) [18].

3.1.6 Analisis Materi

Aplikasi yang dibangun membahas pelajaran IPA kelas 4 SD semester 1. Materi yang akan dibahas hanya mengenai materi rangka manusia serta struktur dan fungsi bagian tumbuhan. Materi dikutip dari buku Ilmu Pengetahuan Alam untuk SD dan MI kelas IV, penulis Budi Wahyono dan Setyo Nurachmandani [19]. Struktur materi mata pelajaran IPA kelas 4 SD yang akan dibahas pada aplikasi yang akan dibuat adalah sebagai berikut :

1. Bab I Rangka Manusia

Tulang membantu melindungi bagianbagian tertentu pada tubuh kita. Misalnya, bagian yang lunak dan organ-organ dalam penting seperti hati, jantung, paru-paru, dan ginjal. Tulang juga membantu kita bergerak atau melakukan kegiatan. Selain itu, tulang merupakan kerangka tubuh yang menahan dan menjaga bentuk tubuh. Coba bayangkan bila tubuh kita tidak memiliki rangka! Apa yang akan terjadi? Tentu tubuh kita akan jatuh terkulai dan tidak bisa berdiri tegak. Apakah yang menghubungkan antara tulang yang satu dengan tulang yang lain? Antartulang dihubungkan oleh sendi. Sendi menjadikan tulang terikat kuat pada tempatnya.

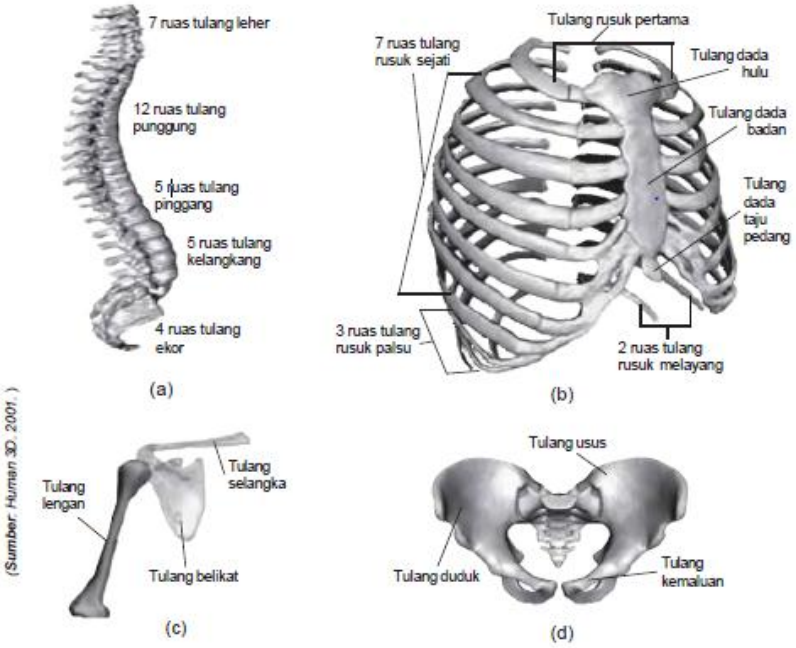
A. Rangka Badan



Gambar 3.15 Rangka Manusia

Rangka badan meliputi tulang belakang, tulang rusuk, tulang dada, tulang gelang bahu, dan tulang gelang panggul. Tulang belakang terdiri atas 7 ruas tulang leher, 12 ruas tulang punggung, 5 ruas tulang pinggang, 5 ruas tulang kelangkang, dan 4 ruas tulang ekor. Tulang rusuk terdiri atas 7 pasang tulang rusuk sejati, 3 pasang tulang rusuk palsu, dan 2 pasang tulang rusuk melayang. Tulang dada terdiri atas tiga bagian, yaitu tangkai atau hulu, badan, dan taju pedang. Tulang dada merupakan tempat melekatnya tulang rusuk bagian depan. Tulang rusuk dan tulang dada membentuk rongga dada. Di atas rongga dada terdapat rangka bahu. Rangka bahu dibentuk oleh tulang gelang bahu. Tulang gelang bahu tersusun dari sepasang tulang belikat dan sepasang tulang selangka. Pada badan bagian bawah terdapat

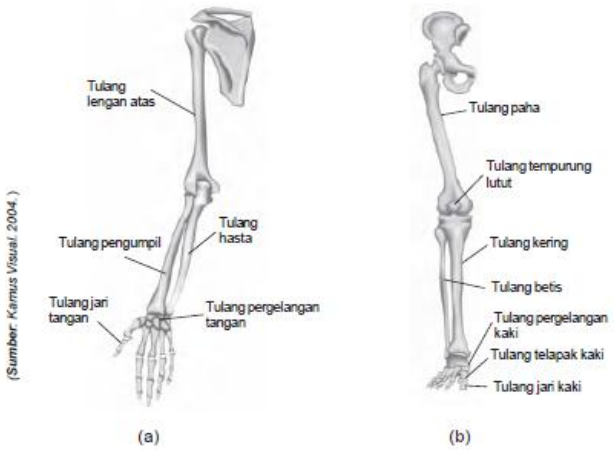
rangka panggul. Rangka panggul dibentuk oleh tulang gelang panggul. Tulang gelang panggul dibentuk oleh 2 tulang usus, 2 tulang duduk, dan 2 tulang kemaluan.



Gambar 3.16 Rangka badan meliputi (a) tulang belakang, (b) tulang rusuk dan tulang dada, (c) tulang gelang bahu, serta (d) tulang gelang panggul.

B. Rangka Anggota Gerak

Rangka anggota gerak terdiri atas tulang-tulang anggota gerak atas (tangan) dan tulang-tulang anggota gerak bawah (tungkai). Tulangtulang anggota gerak atas (tangan), yaitu: tulang lengan atas, tulang hasta, tulang pengumpil, tulang pergelangan tangan, dan tulang telapak tangan, tulang ruas-ruas jari. Tulang-tulang anggota gerak bawah (tungkai), yaitu: tulang paha, tulang kering, tulang betis, tulang tempurung lutut, tulang telapak kaki, tulang pergelangan kaki, dan tulang ruas-ruas jari. Perhatikan gambar di bawah ini.



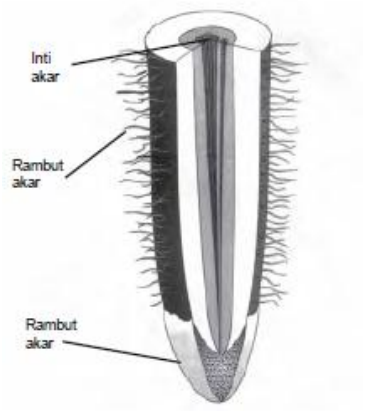
Gambar 3.17 Rangka anggota gerak terdiri atas (a) tulang-tulang anggota gerak atas (tangan) dan (b) tulang-tulang anggota gerak bawah (tungkai).

2. Bab II Struktur dan Fungsi Bagian Tumbuhan

A. Struktur Akar dan Fungsinya

Secara umum, akar memiliki beberapa bagian utama. Bagian-bagian tersebut adalah inti akar, rambut akar, dan tudung akar.

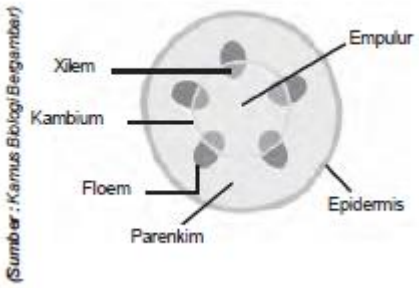
1. **Inti Akar.** Inti akar terdiri atas pembuluh kayu dan pembuluh tapis. Pembuluh kayu berfungsi mengangkut air dari akar ke daun. Pembuluh tapis berfungsi mengangkut hasil fotosintesis dari daun ke seluruh bagian tumbuhan.
2. **Rambut Akar.** Rambut akar atau bulu-bulu akar berbentuk serabut halus. Rambut akar terletak di dinding luar akar. Fungsi rambut akar adalah mencari jalan di antara butiran tanah Hal inilah yang menyebabkan akar dapat menembus masuk ke dalam tanah. Selain itu, rambut akar juga berfungsi menyerap air dari dalam tanah.
3. **Tudung Akar.** Tudung akar terletak di ujung akar. Bagian ini melindungi akar saat menembus tanah.



Gambar 3.18 Bagian-bagian akar.

B. Struktur Batang dan Fungsinya

Batang dapat diumpamakan sebagai sumbu tubuh tumbuhan. Bagian ini umumnya tumbuh di atas tanah. Arah tumbuh batang tumbuhan menuju sinar matahari. Umumnya batang bercabang, tetapi pada tumbuhan tertentu batangnya tidak memiliki cabang seperti pada tumbuhan pisang, kelapa, dan pepaya. Struktur batang terdiri atas epidermis, korteks, endodermis, dan silinder pusat (*stele*). Silinder pusat pada batang ini terdiri atas beberapa jaringan yaitu empulur, perikardium, dan berkas pengangkut yaitu xilem dan floem.



Gambar 3.19 Struktur Batang

C. Struktur Daun dan Fungsinya

Bagian-bagian daun lengkap terdiri atas tulang daun, helai daun, tangkai daun, dan pelepah daun. Contoh daun yang memiliki bagianbagian lengkap, antara lain daun pisang dan daun bambu. Di alam, kebanyakan tumbuhan memiliki daun yang tidak lengkap. Misalnya, ada daun yang hanya terdiri atas tangkai dan helai daun saja, contohnya daun mangga; ada pula daun yang hanya terdiri atas pelepah dan helai daun saja, contohnya daun padi dan jagung.



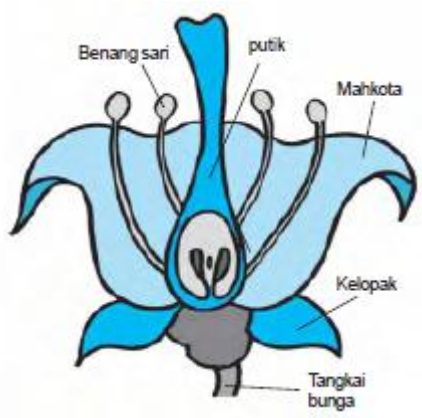
Gambar 3.20 Struktur Daun

D. Struktur Bunga dan fungsinya

Bunga lengkap memiliki bagian-bagian sebagai berikut.

1. Kelopak, umumnya berwarna hijau dan berfungsi menutup bunga di saat masih kuncup.
2. Mahkota, merupakan bagian bunga yang indah dan berwarnawarni.
3. Benang sari dengan serbuk sari sebagai alat kelamin jantan.
4. Putik sebagai alat kelamin betina.
5. Dasar dan tangkai bunga sebagai tempat kedudukan bunga.

Bunga yang memiliki tangkai, kelopak, mahkota, benang sari, dasar bunga, dan putik disebut bunga sempurna. Jika memiliki semua bagian kecuali putik, maka disebut bunga jantan. Jika memiliki semua bagian kecuali benang sari, maka disebut bunga betina. Bunga yang memiliki benang sari dan putik disebut bunga hermafrodit.



Gambar 3.21 Bunga dan bagian-bagiannya.

3.1.7 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan aplikasi yang dibangun. Adapun kebutuhan non-fungsional pada aplikasi pengenalan rangka dan bagian tumbuhan untuk siswa kelas 4 ini meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan memakai aplikasi.

3.1.7.1 Analisis Kebutuhan Perangkat Keras

Analisis Perangkat keras atau *hardware* merupakan salah satu hal yang penting karena tanpa *hardware* yang memenuhi syarat, program yang akan dibuat tidak akan dapat berjalan. Spesifikasi perangkar keras yang dapat dipergunakan untuk membangun aplikasi pengenalan rangka dan bagian tumbuhan menggunakan teknoogi *augmented reality* ini adalah sebagai berikut :

Tabel 3.2 Kebutuhan Perangkat Keras Pengembang

No	Perangkat Keras	Spesifikasi
1	Processor	Intel (R) Core(TM) i5-2430M CPU @ 2.40GHz (4 CPUs)
2	Monitor	15 inch
3	VGA	2 GB
4	RAM	6 GB
5	Hard disk drive	500 GB
6	Webcam	3 MP
7	Keyboard dan Mouse	Standar
8	Speaker	Standar

Rekomendasi minimum spesifikasi kebutuhan perangkat keras yang digunakan untuk menjalankan aplikasi pengenalan rangka dan bagian tumbuhan menggunakan teknologi *augmented reality* ini adalah sebagai berikut :

Tabel 3.3 Kebutuhan Perangkat Keras Pengguna

No	Perangkat Keras	Spesifikasi
1	Processor	Processor dengan kecepatan 1.8 Ghz
2	Monitor	14 inch
3	VGA	512 GB
4	RAM	1 GB
5	Hard disk drive	Free Space 500 MB
6	Webcam	0.3 MP
7	Keyboard dan Mouse	Standar
8	Speaker	Standar

3.1.7.2 Analisis Kebutuhan Perangkat Lunak

Analisis perangkat lunak atau *software* merupakah hal yang terpenting dalam mendukung kinerja sebuah sistem. Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar dapat saling berinteraksi diantara keduannya. Perangkat lunak yang dapat digunakan untuk membangun aplikasi pengenalan rangka dan bagian tumbuhan menggunakan teknologi *augmented reality* ini adalah sebagai berikut :

Tabel 3.4 Kebutuhan Perangkat Lunak Pengembang

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows XP, Windows 7
2	Tool Pembangun	Adobe Flash CS5, Adobe Flash Builder 4
3	Tool Desain	Adobe Photoshop CS5, Autodesk 3ds Max 2010
4	Tool Compiler	Adobe Flash Player 10

Rekomendasi minimum spesifikasi kebutuhan perangkat lunak yang digunakan untuk menjalankan aplikasi pengenalan rangka dan bagian tumbuhan menggunakan teknologi *augmented reality* ini adalah sebagai berikut :

Tabel 3.5 Kebutuhan Perangkat Lunak Pengguna

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows XP, Windows 7
2	Tool Compiler	Adobe Flash Player 10

3.1.7.3 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna merupakan analisis terhadap *user* yang akan menggunakan sistem yang akan dibangun. Adapun kebutuhan pengguna yang terlibat dalam penggunaaan aplikasi pengenalan rangka dan bagian tumbuhan menggunakan teknoogi Augmented Reality untuk kelas 4 ini dapat dilihat pada tabel 3.6 berikut:

Tabel 3.6 Analisis Kebutuhan Pengguna

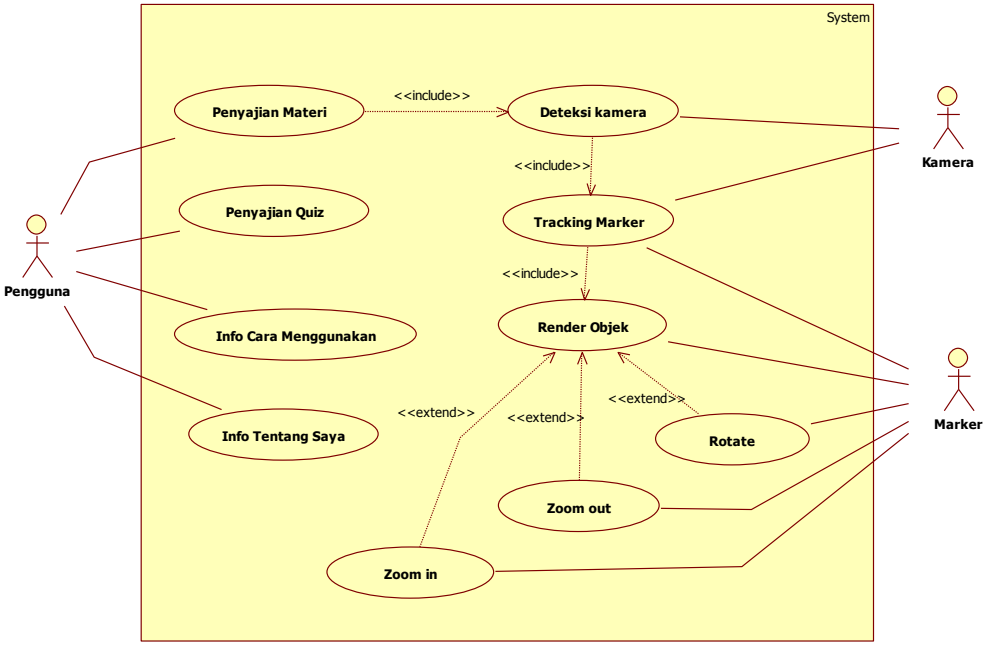
Pengguna	Anak-anak
Hak Akses	Menggunakan Aplikasi
Tingkat Usia	9 sampai 10 tahun (SD kelas 4)
Tingkat Sekolah	Sekolah Dasar
User Experience	Dapat menggunakan komputer, harus bisa membaca, mempunyai pengelihatan yang baik

3.1.8 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan kebutuhan sistem yang akan dibangun pada aplikasi pengenalan rangka dan bagian tumbuhan menggunakan teknologi *augmented reality* ini. Adapun kebutuhan fungsional pada aplikasi yang akan dibangun ini dengan pemodelan berorientasi objek. Perangkat lunak ini dimodelkan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain *use case diagram, skenario, sequence diagram, activity diagram, class diagram.*

3.1.8.1 *Use Case* Diagram

Use Case merupakan gambaran skenario dari interaksi antara *user* dengan sistem. Sebuah diagram *Use Case* menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi.



Gambar 3.22 Use Case Diagram

3.1.8.1.1 Definisi *Actor*

Definisi *Actor* berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use Case Diagram*. Definisi *Actor* diterangkan pada Tabel 3.6.

Tabel 3.7 Definisi Actor

No.	*Actor*	Deskripsi
1	Pengguna	Orang yang berinteraksi dengan system, yaitu pengguna yang mengakses semua fungsi yang disediakan system.
2	Kamera	Kamera adalah alat bantu yang digunakan untuk menggambil

		gambar dan sebagai pendeteksi keberadaan marker.
3	*Marker*	Pola yang dibuat dalam bentuk gambar yang telah dicetak dengan printer yang akan dikenalai oleh kamera. Dan juga digunakan sebagai penanda dimana obyek virtual 3D akan ditampilkan.

3.1.8.1.2 Definisi *Use Case*

Definisi *Use Case* berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use Case Diagram*. Definisi *Use Case* diterangkan pada Tabel 3.8.

Tabel 3.8 Definisi *Use Case*

NO	*Use Case*	Deskripsi
1	Penyajian Materi	Fungsionalitas untuk penyajian materi
2	Penyajian Quiz	Fungsionalitas untuk penyajian quiz
3	Info Cara Menggunakan	Fungsionalitas untuk menampilkan info cara menggunakan
4	Info Tentang Saya	Fungsionalitas untuk menampilkan info pembuat aplikasi
5	Deteksi Kamera	Fungsionalitas mendeteksi ketersediaan kamera
6	Tracking Marker	Fungsionalitas untuk mendeteksi keberadaan atau mengenali marker
7	Render Objek	Fungsionalitas untuk menampilkan objek ke layar
8	Zoom in	Fungsionalitas untuk memperbesar ukuran objek

9	Zoom out	Fungsionalitas untuk memperkecil ukuran objek
10	Rotate	Fungsionalitas untuk memutar objek

3.1.8.2 Skenario *Use Case*

Skenario *Use Case* merupakan penggambaran dari alur penggunaan sistem dimana setiap skenario digambarkan dari sudut pandang *actor*, seseorang atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara.

1. Skenario *Use Case* Penyajian Materi

Tabel 3.9 di bawah ini adalah *Skenario Use Case* belajar pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.9 *Skenario Use Case* Penyajian Materi

Identifikasi	
Nomor	UCS-01
Nama	Penyajian Materi
Tujuan	Menjalankan menu belajar
Deskripsi	Proses untuk penyajian materi
Aktor	Pengguna, Kamera, Marker
Skenario Utama	
Kondisi awal	Pengguna berada pada menu utama
Aksi Aktor	**Reaksi Sistem**
1. Pengguna memilih menu belajar	
	2. Sistem menampilkan menu pilihan materi
3. Pengguna memilih materi yang ingin dipelajari	
	4. Sistem mendeteksi kamera
5. Pengguna memilih allow	
	6. Kamera aktif
7. Pengguna Mengarahkan marker ke kamera	
	8. Sistem merender objek dan menampilkan diatas marker
Kondisi Akhir	Aplikasi menampilkan materi

2. Skenario *Use Case Penyajian* Quiz

Tabel 3.10 di bawah ini adalah *Skenario Use Case* Quiz pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.10 Skenario Use Case Penyajian Quiz

Identifikasi	
Nomor	UCS-02
Nama	Penyajian Quiz
Tujuan	Menjalankan menu Quiz
Deskripsi	Proses untuk penyajian Quiz
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Pengguna berada pada menu utama
Aksi Aktor	**Reaksi Sistem**
1. Pengguna memilih menu quiz	
	2. Menampilkan menu pilihan materi quiz
3. Pengguna memilih materi yang diinginkan	
	4. Menampilkan petunjuk quiz
5. Pengguna menekan tombol mulai	
	6. Menampilkan pertanyaan
7. Pengguna menjawab pertanyaan dengan menekan tombol A,B,C atau D	
	8. Jika jawaban benar maka skor ditambah 20 point dan jika jawaban salah skor dikurangi 5 point
	9. Menampilkan skor akhir
Kondisi Akhir	Aplikasi menampilkan skor akhir

3. Skenario *Use Case* Info Cara Menggunakan

Tabel 3.11 di bawah ini adalah *Skenario Use Case* info cara menggunakan pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.11 Skenario *Use Case* Info Cara Menggunakan

Identifikasi	
Nomor	UCS-03
Nama	Info Cara Menggunakan

Tujuan	Menampilkan info cara menggunakan aplikasi
Deskripsi	Proses untuk menampilkan info cara menggunakan aplikasi
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Pengguna berada pada menu utama
Aksi Aktor	**Reaksi Sistem**
1. Pengguna memilih menu cara menggunakan	
	2. Menampilkan info cara menggunakan aplikasi
Kondisi Akhir	Aplikasi menampilkan info cara menggunakan aplikasi

4. Skenario *Use Case* Info Tentang Saya

Tabel 3.12 di bawah ini adalah *Skenario Use Case* info tentang saya pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.12 Skenario *Use Case* info tentang saya

Identifikasi	
Nomor	UCS-04
Nama	Info Tentang Saya
Tujuan	Menampilkan info pembuat aplikasi
Deskripsi	Proses untuk menampilkan info tentang pembuat aplikasi
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Pengguna berada pada menu utama
Aksi Aktor	**Reaksi Sistem**
1. Pengguna memilih menu tentang saya	
	2. Menampilkan info tentang pembuat aplikasi
Kondisi Akhir	Aplikasi menampilkan info tentang pembuat aplikasi

5. Skenario *Use Case* Deteksi Kamera

Tabel 3.13 di bawah ini adalah *Skenario Use Case* deteksi kamera pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.13 Skenario *Use Case* Deteksi Kamera

Identifikasi	
Nomor	UCS-05
Nama	Deteksi Kamera
Tujuan	Mengaktifkan kamera untuk menangkap gambar
Deskripsi	Proses untuk mengaktifkan kamera
Aktor	Pengguna, Kamera
Skenario Utama	
Kondisi awal	Kamera dalam kondisi non-aktif
Aksi Aktor	**Reaksi Sistem**
	1. Sistem mendeteksi ketersediaan kamera
2. Pengguna mengaktifkan kamera	
	3. Kamera aktif
Kondisi Akhir	Kamera dalam kondisi aktif

6. Skenario *Use Case Tracking Marker*

Tabel 3.14 di bawah ini adalah *Skenario Use Case tracking marker* pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.14 Skenario *Use Case* Tracking Marker

Identifikasi	
Nomor	UCS-06
Nama	Tracking Marker
Tujuan	Mendeteksi keberadaan marker
Deskripsi	System mendeteksi keberadaan marker melalui gambar yang ditangkap oleh kamera
Aktor	Pengguna, Kamera , Marker
Skenario Utama	
Kondisi awal	Kamera terdeteksi
Aksi Aktor	**Reaksi Sistem**
1. Pengguna mengarahkan marker ke kamera	

	2. Sistem memproses masukan gambar yang ditangkap oleh kamera
	3. Sistem mencocokan pola
	4. Marker terdeteksi
Kondisi Akhir	Marker terdeteksi

7. Skenario *Use Case* Render Objek

Tabel 3.15 di bawah ini adalah *Skenario Use Case render objek* pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.15 Skenario *Use Case* Render Objek

Identifikasi	
Nomor	UCS-07
Nama	Render Objek
Tujuan	Menampilkan Objek di monitor
Deskripsi	Proses untuk menampilkan objek di monitor
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Marker terdeteksi
Aksi Aktor	**Reaksi Sistem**
1. Pengguna mengarahkan marker ke kamera	
	2. Sistem memproses masukan gambar yang ditangkap oleh kamera
	3. Sistem mencocokan pola
	4. Marker terdeteksi
	5. Objek tampil di monitor
Kondisi Akhir	Objek tampil di monitor

8. Skenario *Use Case* Zoom in

Tabel 3.16 di bawah ini adalah *Skenario Use Case zoom in* pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.16 Skenario *Use Case* Render Objek Zoom in

Identifikasi	
Nomor	UCS-08
Nama	Zoom in
Tujuan	Memperbesar ukuran objek
Deskripsi	Proses untuk memperbesar ukuran objek
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Marker terdeteksi
Aksi Aktor	**Reaksi Sistem**
1. Pengguna mengarahkan marker ke kamera	2. Marker terdeteksi oleh kamera
	3. Objek tampil di monitor
4. Pengguna menutup marker zoom in	
	5. Objek bertambah besar selama marker zoom in ditutup
Kondisi Akhir	Ukuran objek menjadi lebih besar dari pada ukuran sebelumnya

9. Skenario *Use Case* Zoom out

Tabel 3.17 di bawah ini adalah *Skenario Use Case zoom out* pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.17 Skenario *Use Case* Render Objek Zoom in

Identifikasi	
Nomor	UCS-09
Nama	Zoom out
Tujuan	Memperkecil ukuran objek
Deskripsi	Proses untuk memperkecil ukuran objek
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Marker terdeteksi

Aksi Aktor	Reaksi Sistem
1. Pengguna mengarahkan marker ke kamera	2. Marker terdeteksi oleh kamera
	3. Objek tampil di monitor
4. Pengguna menutup marker zoom out	
	5. Objek bertambah kecil selama marker zoom out ditutup
Kondisi Akhir	Ukuran objek menjadi lebih kecil dari pada ukuran sebelumnya

10. Skenario *Use Case* Rotate

Tabel 3.18 di bawah ini adalah *Skenario Use Case Rotate* pada aplikasi pengenalan rangka dan bagian tumbuhan.

Tabel 3.18 Skenario *Use Case* Render Objek Rotate

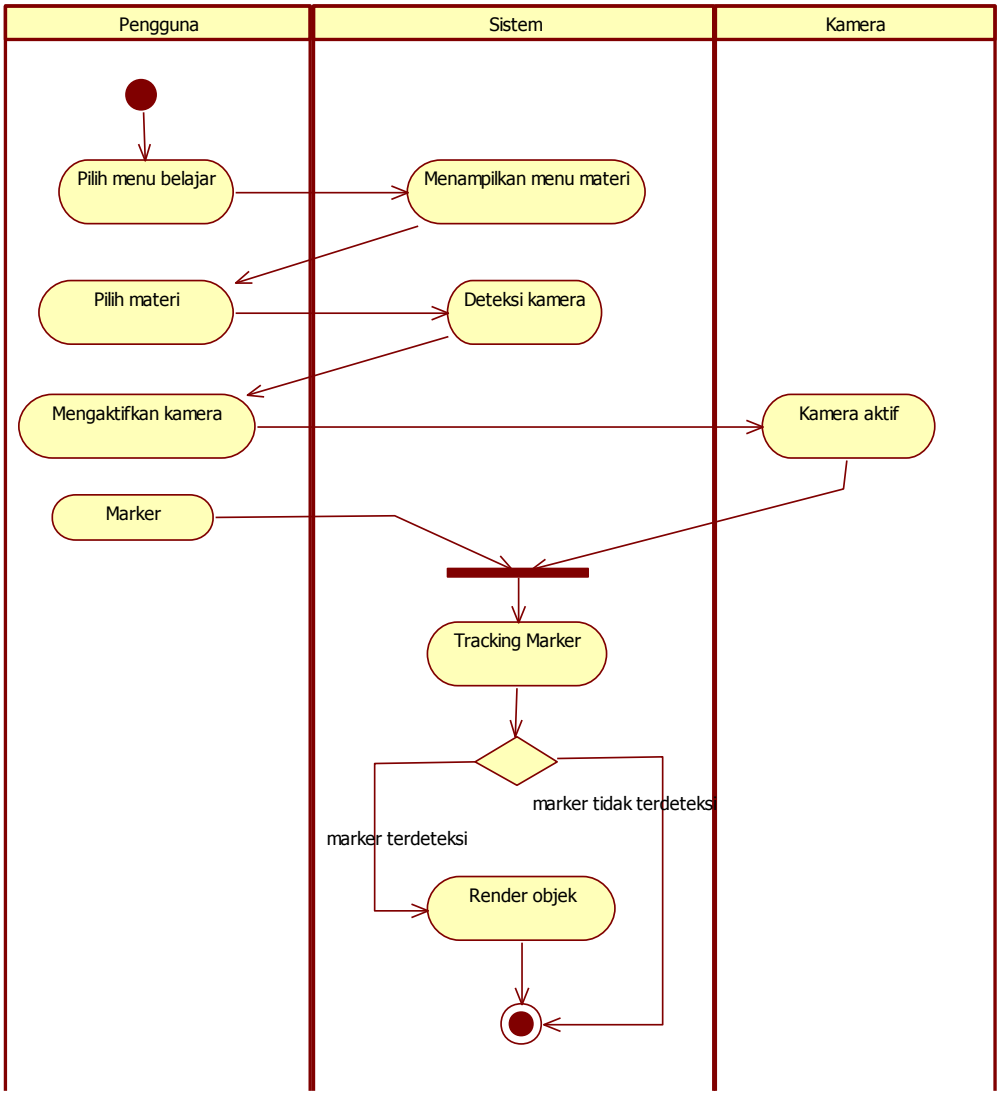
Identifikasi	
Nomor	UCS-10
Nama	Rotate
Tujuan	Memutar Objek
Deskripsi	Proses untuk memutar objek
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Marker terdeteksi
Aksi Aktor	**Reaksi Sistem**
1. Pengguna mengarahkan marker ke kamera	2. Marker terdeteksi oleh kamera
	3. Objek tampil di monitor
4. Pengguna menutup marker rotate	
	5. Objek berputar selama marker zoom out ditutup
Kondisi Akhir	Objek berputar

3.1.8.3 *Activity Diagram*

Activity diagram menggambarkan urutan aktivitas dalam sebuah proses. Struktur diagram ini mirip *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. Berikut ini merupakan *activity diagram* pada sistem yang akan dibangun.

3.1.8.3.1 *Activity Diagram* Penyajian Materi

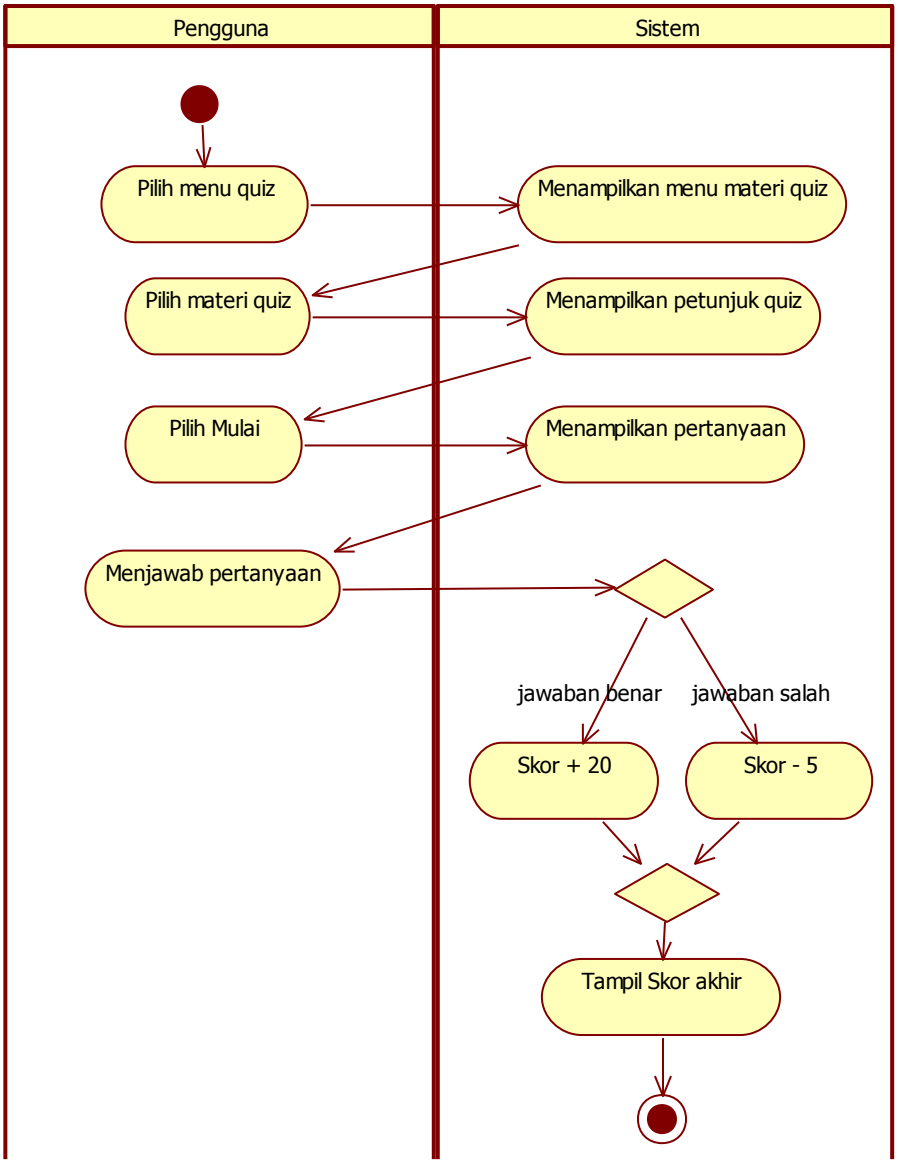
Activity Diagram terkait skenario penyajian materi memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3.23 *Activity Diagram* Penyajian Materi

3.1.8.3.2 *Activity Diagram* Penyajian Quiz

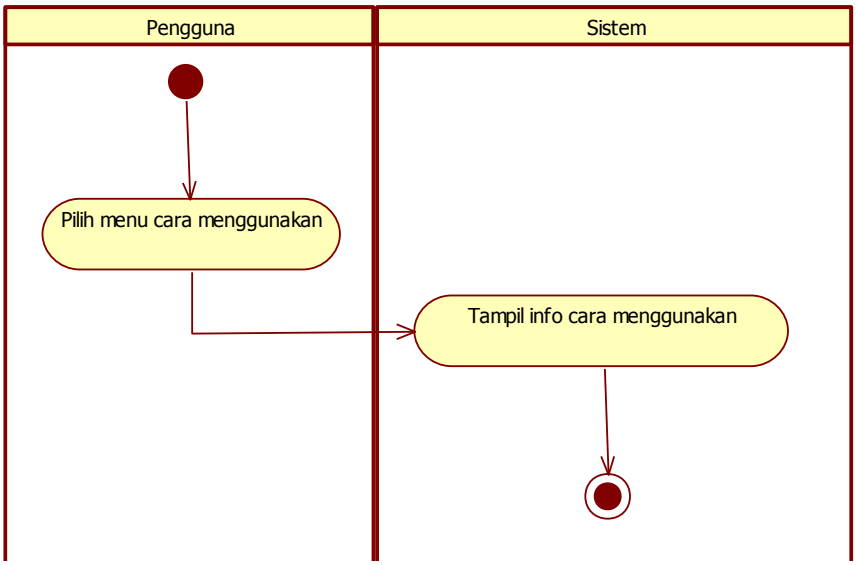
Activity Diagram terkait skenario penyajian quiz memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3.24 *Activity Diagram* Penyajian Quiz

3.1.8.3.3 *Activity Diagram* Info Cara Menggunakan

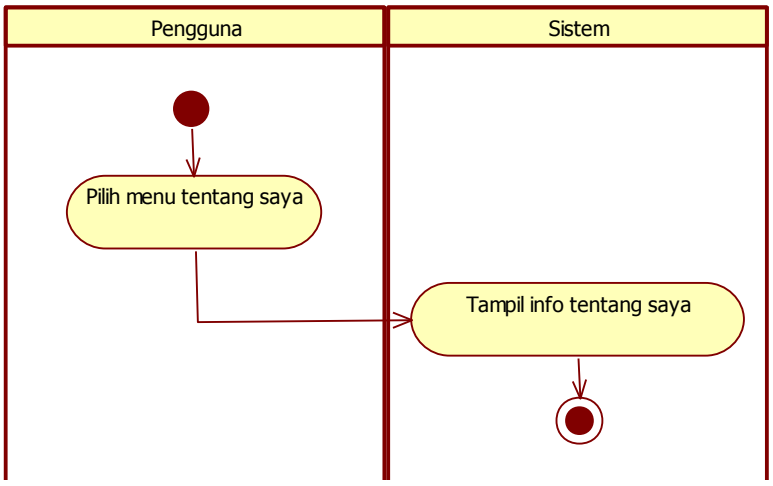
Activity Diagram terkait skenario info cara menggunakan memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3.25 *Activity Diagram* info cara menggunakan

3.1.8.3.4 *Activity Diagram Info* Tentang Saya

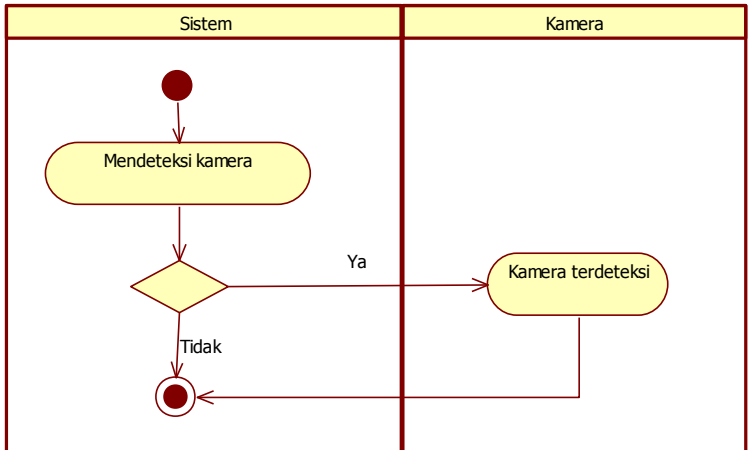
Activity Diagram terkait skenario info tentang saya memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3.26 *Activity Diagram* Info Tentang Saya

3.1.8.3.5 *Activity Diagram* Deteksi Kamera

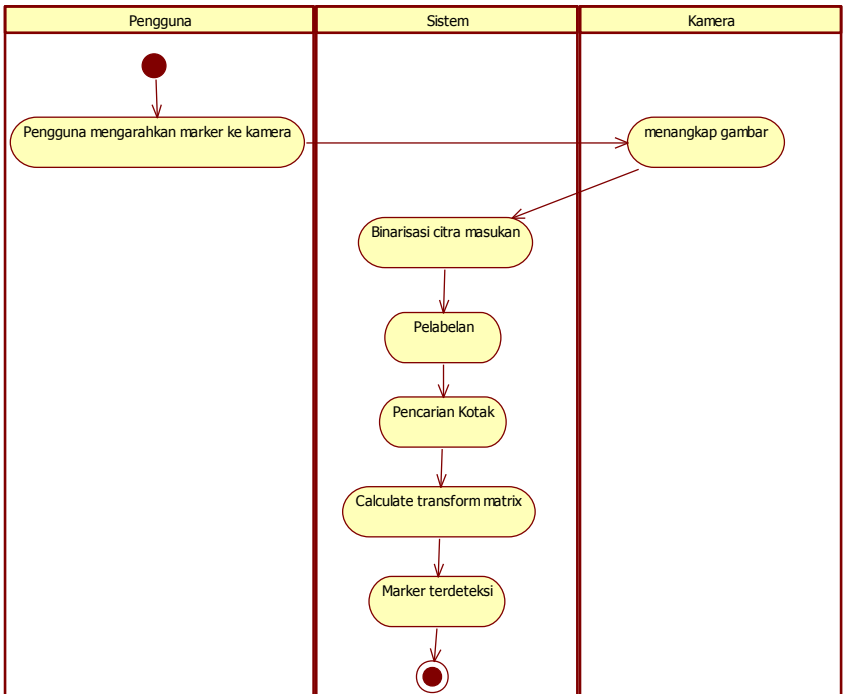
Activity Diagram terkait skenario deteksi kamera memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3.27 *Activity Diagram* Deteksi Kamera

3.1.8.3.6 *Activity Diagram Tracking Marker*

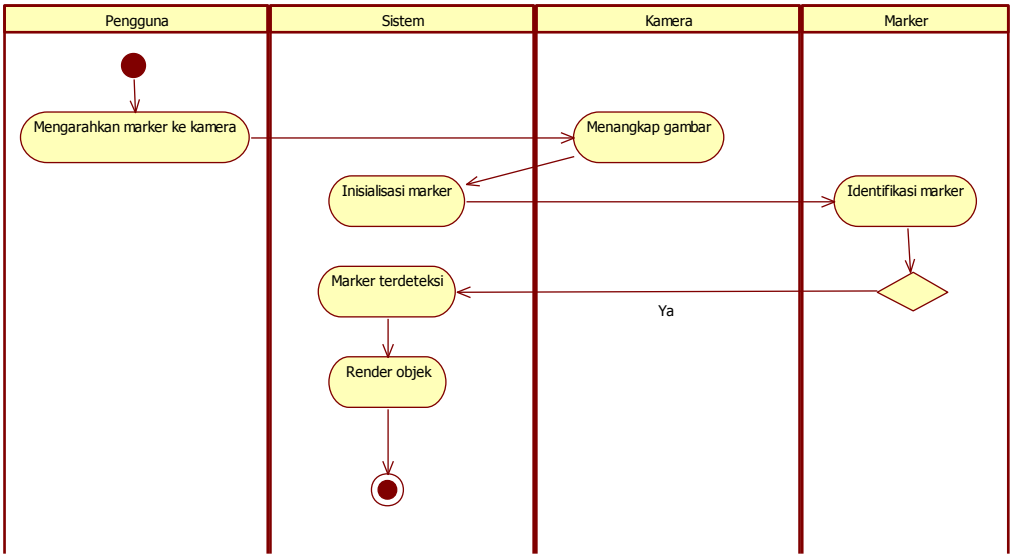
Activity Diagram terkait skenario tracking marker memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3.28 *Activity Diagram* Tracking Marker

3.1.8.3.7 *Activity Diagram Render* Objek

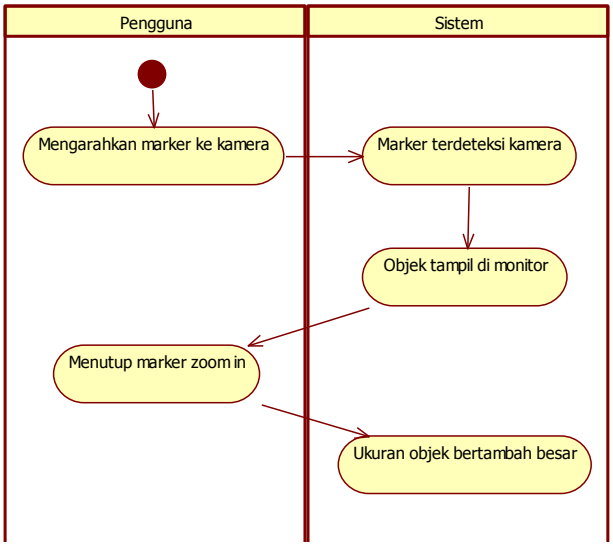
Activity Diagram terkait skenario render objek memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3.29 *Activity Diagram* Render Objek

3.1.8.3.8 *Activity Diagram Zoom in*

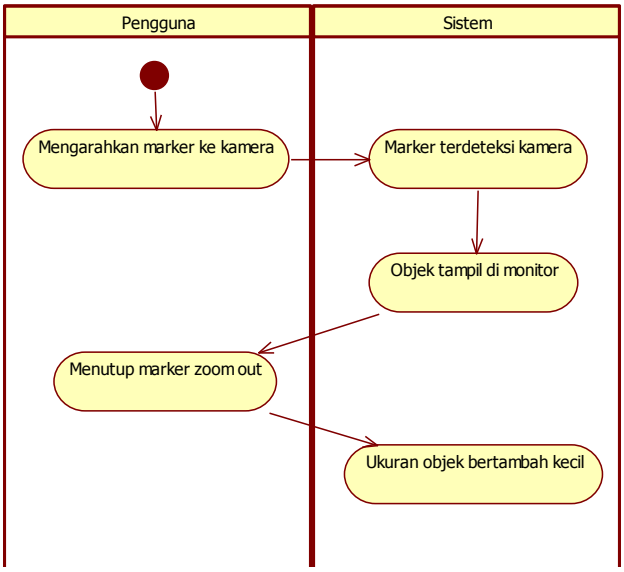
Activity Diagram terkait skenario zoom in memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3.30 *Activity Diagram Zoom in*

3.1.8.3.9 *Activity Diagram Zoom out*

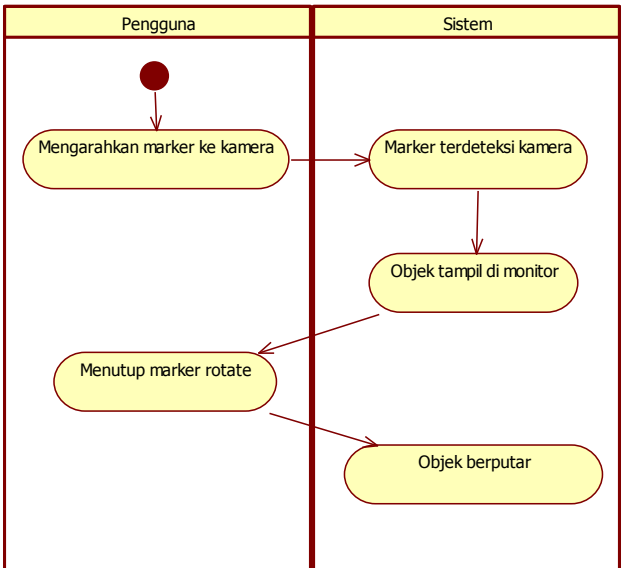
Activity Diagram terkait skenario zoom out memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3.31 *Activity Diagram Zoom in*

3.1.8.3.10 *Activity Diagram Rotate*

Activity Diagram terkait skenario rotate memiliki alur proses seperti digambarkan sebagai berikut :

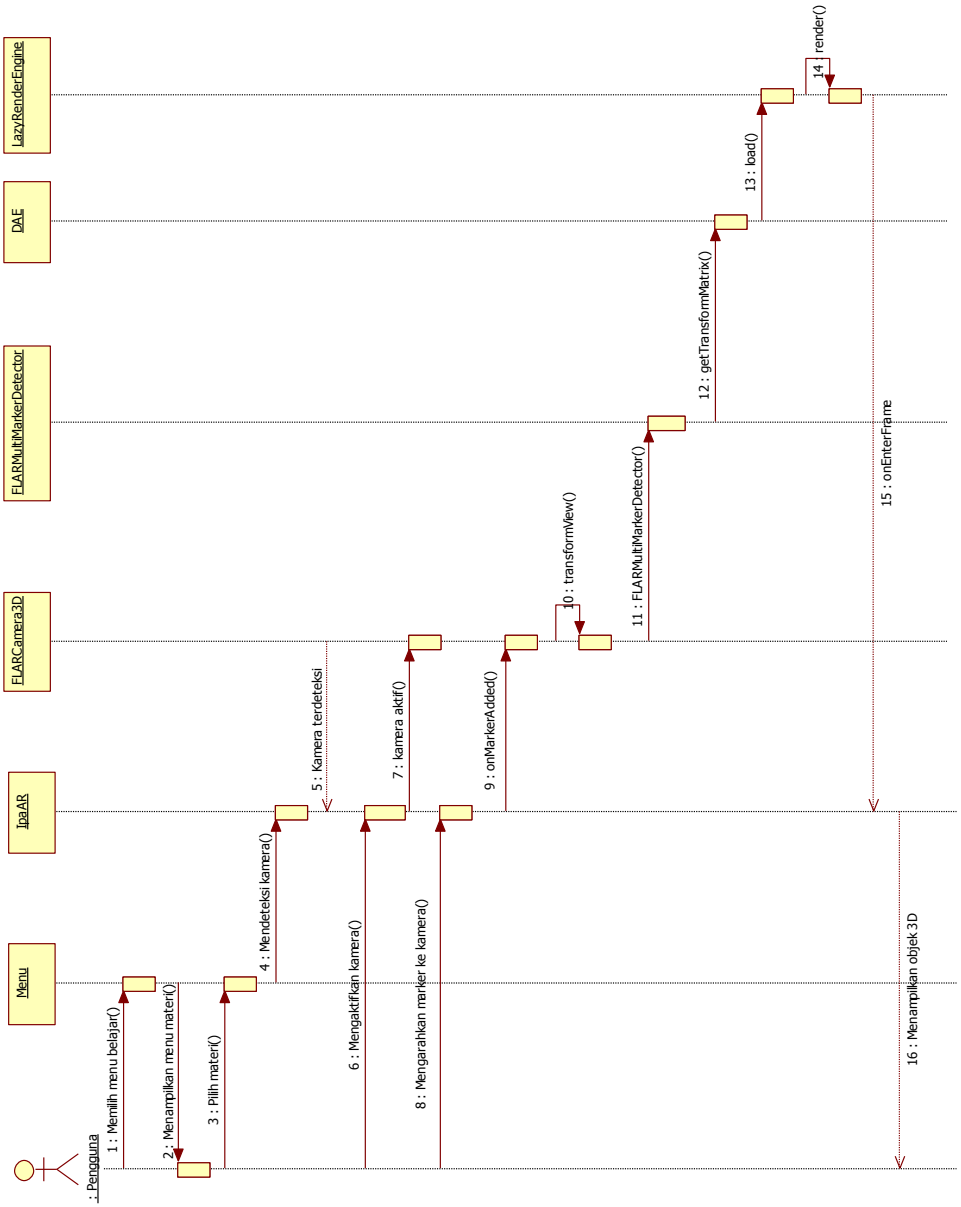


Gambar 3.32 *Activity Diagram Rotate*

3.1.8.4 *Sequence Diagram*

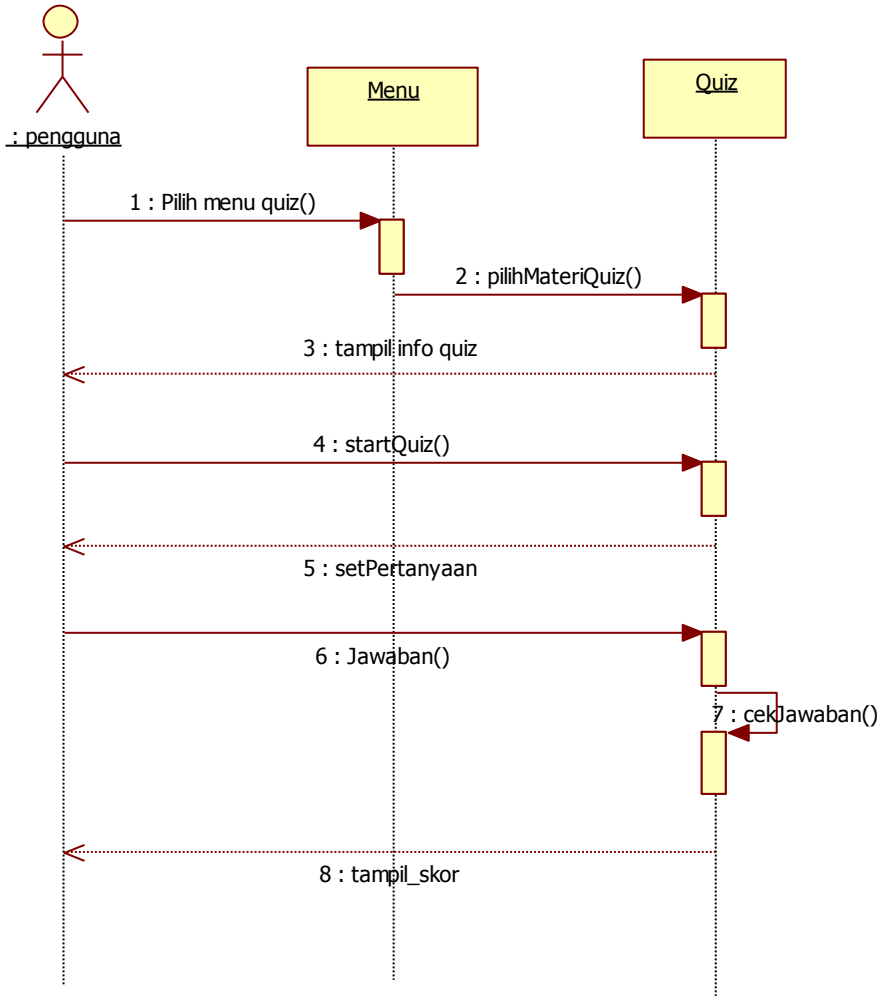
Sequence diagram menjelaskan gambaran interaksi antar objek dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. *Sequence diagram* pada aplikasi ini terdiri dari *Sequence diagram* menu utama, mulai *game* , pengaturan dan instruksi.

1. ***Sequence Diagram* Penyajian Materi**



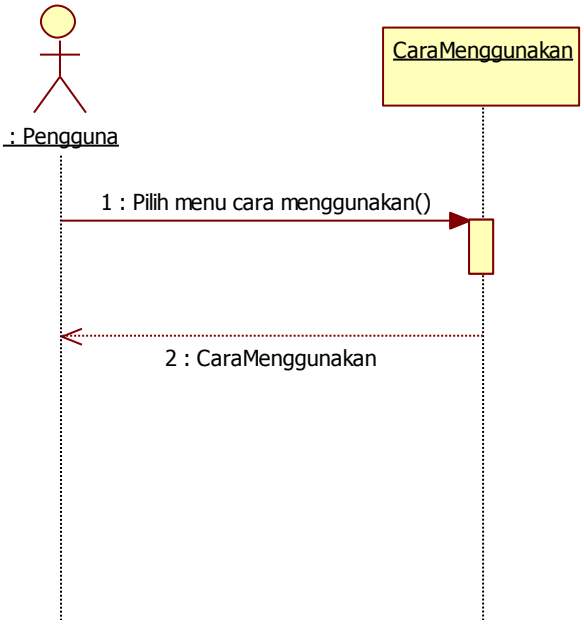
Gambar 3.33 *Sequence Diagram* Penyajian Materi

2. *Sequence Diagram* Penyajian Quiz



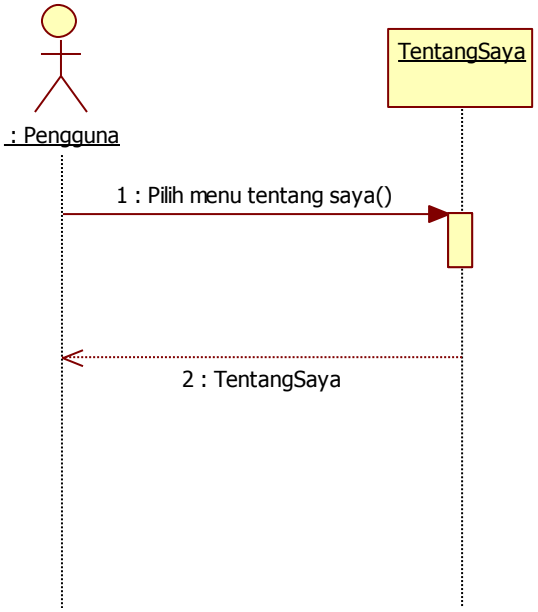
Gambar 3.34 *Sequence Diagram* Penyajian Quiz

3. *Sequence Diagram* Cara Menggunakan



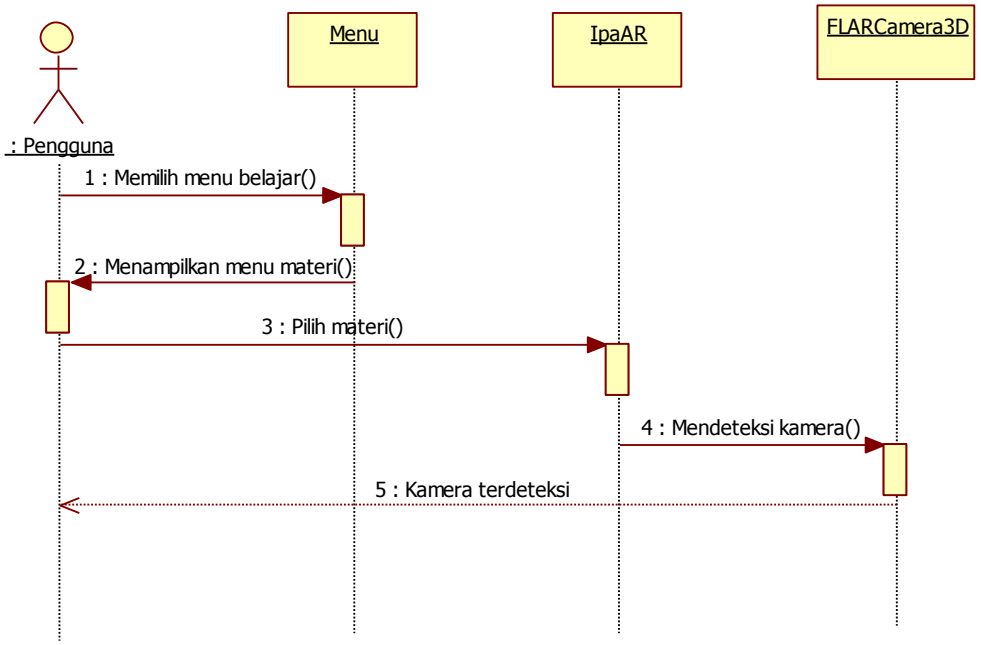
Gambar 3.35 *Sequence Diagram* Cara Menggunakan

4. *Sequence Diagram* Tentang Saya



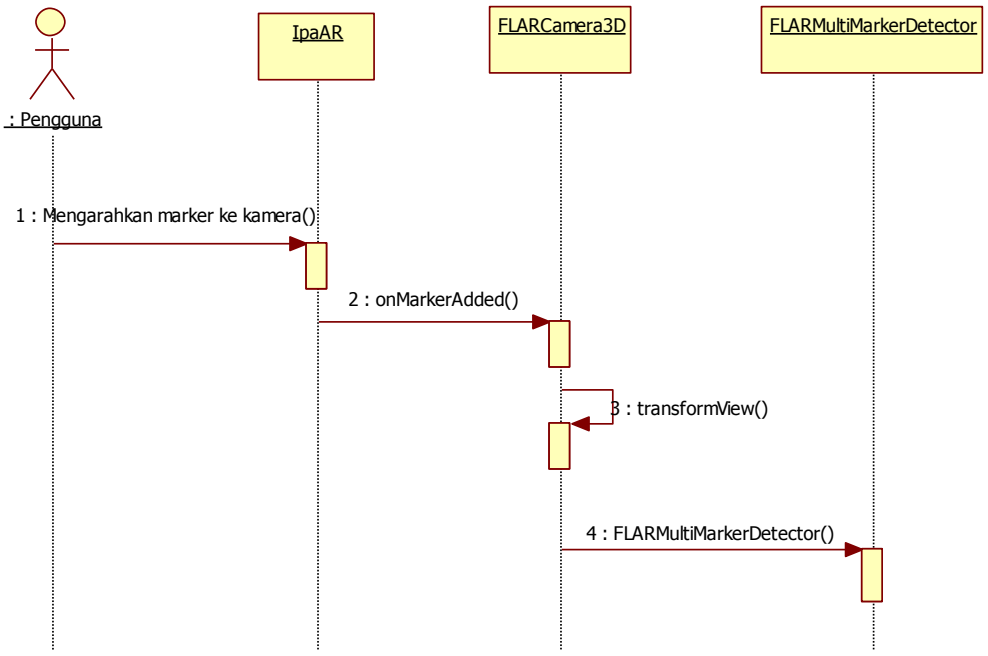
Gambar 3.36 *Sequence Diagram* Tentang Saya

5. *Sequence Diagram* Deteksi Kamera



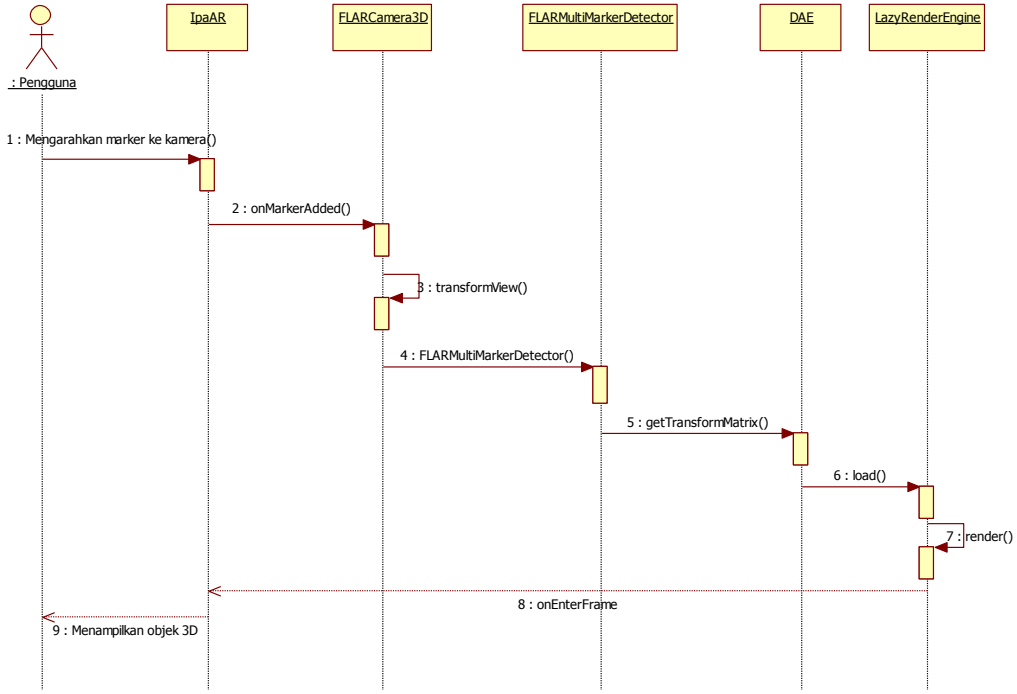
Gambar 3.37 *Sequence Diagram* Deteksi Kamera

6. *Sequence Diagram Tracking Marker*



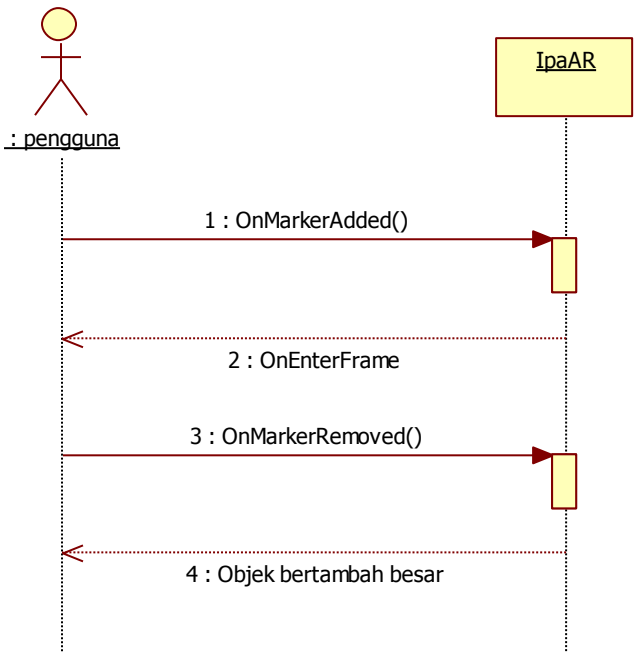
Gambar 3.38 Sequence Diagram Tracking Marker

7. *Sequence Diagram* Render Objek



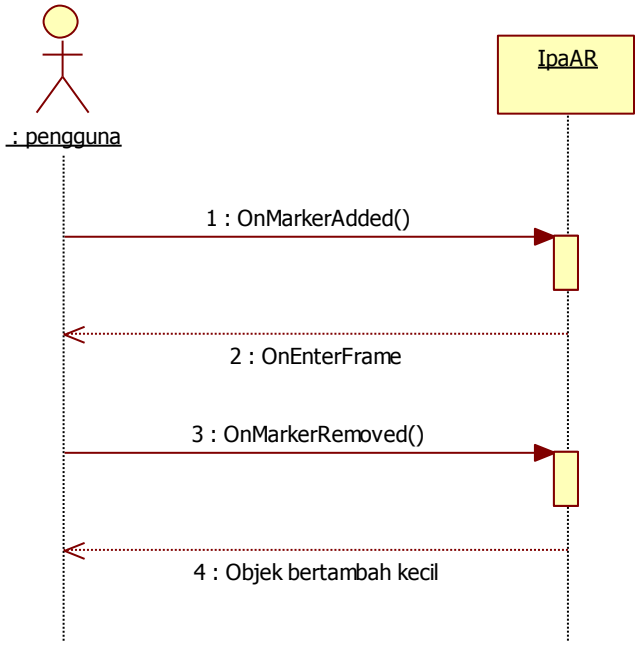
Gambar 3.39 *Sequence Diagram* Render Objek

8. *Sequence Diagram* Zoom in



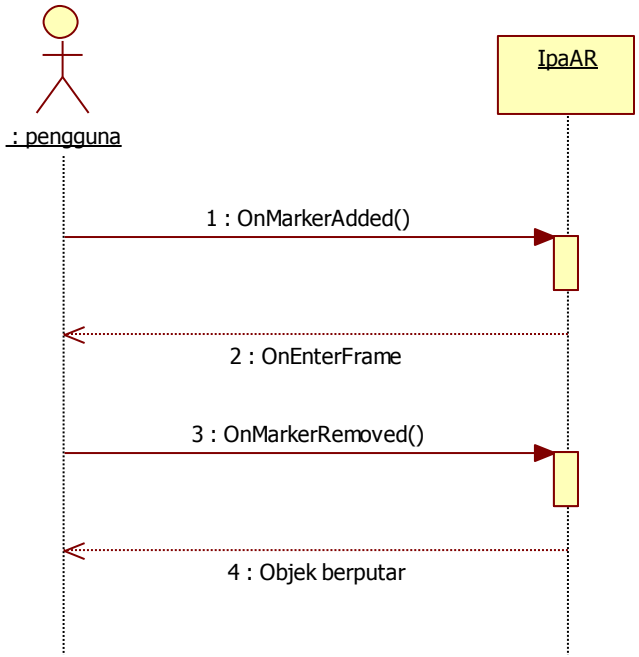
Gambar 3.40 *Sequence Diagram* Zoom in

9. *Sequence Diagram* Zoom out



Gambar 3.41 *Sequence Diagram* Zoom out

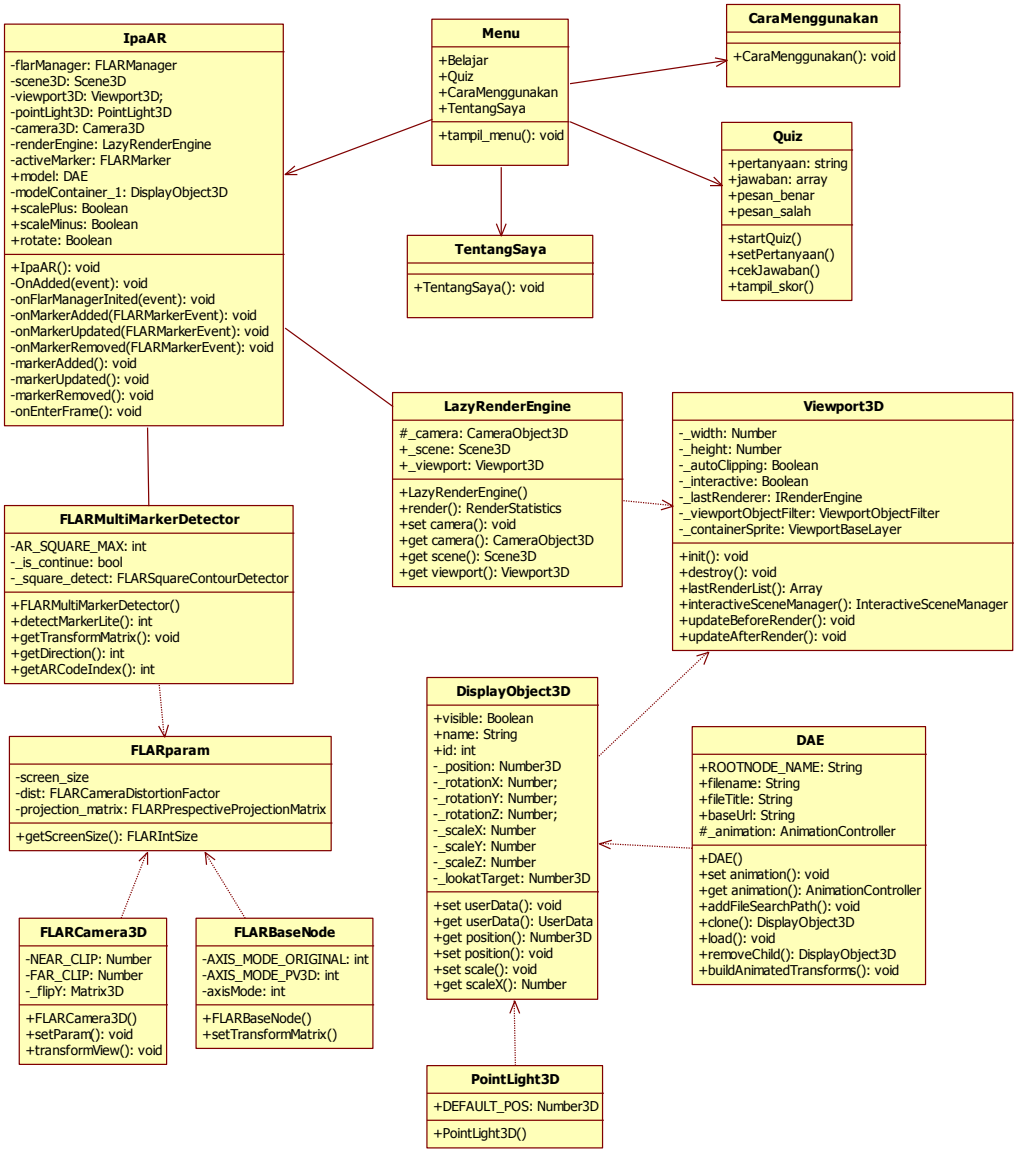
10. *Sequence Diagram* Rotate



Gambar 3.42 *Sequence Diagram* Rotate

3.1.8.5 Class Diagram

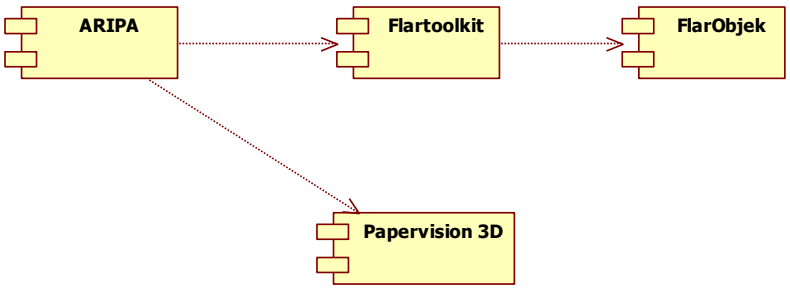
Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*attribut* atau *property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). Berikut adalah *class diagram* dari aplikasi pengenalan rangka manusia dan bagian tumbuhan.



Gambar 3.43 Class Diagram

3.1.8.6 Component Diagram

Component diagram adalah bagian fisik dari sebuah system, karena menetap dikomputer, bukan dibenak para analis. Komponen bisa berupa table, file data, file exe, dokumen dan dan lain-lain. *Component* adalah implementasi dari sebuah class. *Component diagram* rancangan aplikasi pengenalan rangka dan bagian tumbuhan yang akan dibangun dapat dilihat pada gambar dibawah ini.



Gambar 3.44 Component Diagram

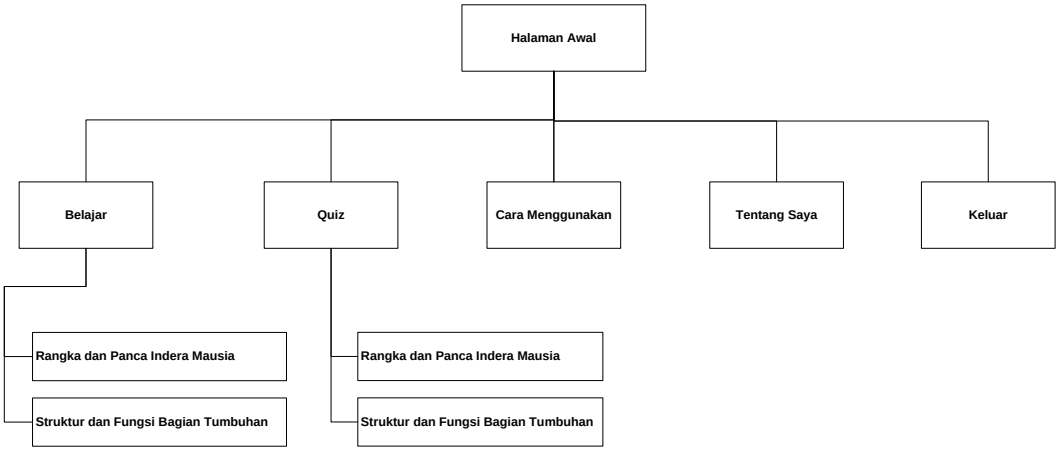
3.2 Perancangan Sistem

Perancangan merupakan bagian dari metodologi pembangunan suatu perangkat lunak yang harus dilakukan setelah melalui tahapan analisis. Untuk dapat melakukan perancangan sistem maka dilakukan beberapa buah langkah yaitu sebagai berikut :

1. Perancangan Struktur *Menu*
2. Perancangan Antarmuka
3. Perancangan Pesan
4. Jaringan Semantik
5. Perancangan *Method*

3.2.1 Perancangan Struktur Menu

Struktur *menu* adalah bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer. Sehingga saat menjalankan program, pengguna tidak mengalami kesulitan dalam memilih *menu-menu* yang diinginkan. Berikut ini perancangan struktur *menu* pada aplikasi pengenalan rangka dan bagian tumbuhan untuk siswa kelas 4 yang ditunjukan pada gambar berikut ini.



Gambar 3.45 Struktur Menu

3.2.2 Perancangan Antarmuka

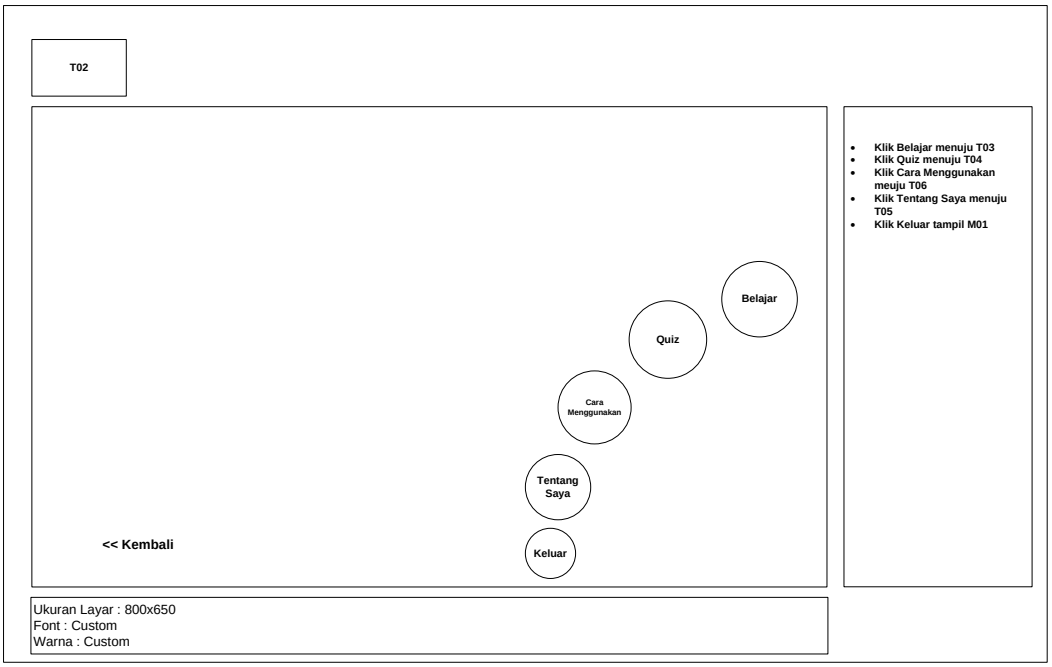
Perancangan antarmuka dibutuhkan untuk mawakili keadaan sebenarnya dari aplikasi yang akan dibangun. Berikut ini contoh perancangan antarmuka yang ada dari aplikasi yang akan dibangun yaitu :

1. Perancangan Antarmuka Halaman Awal



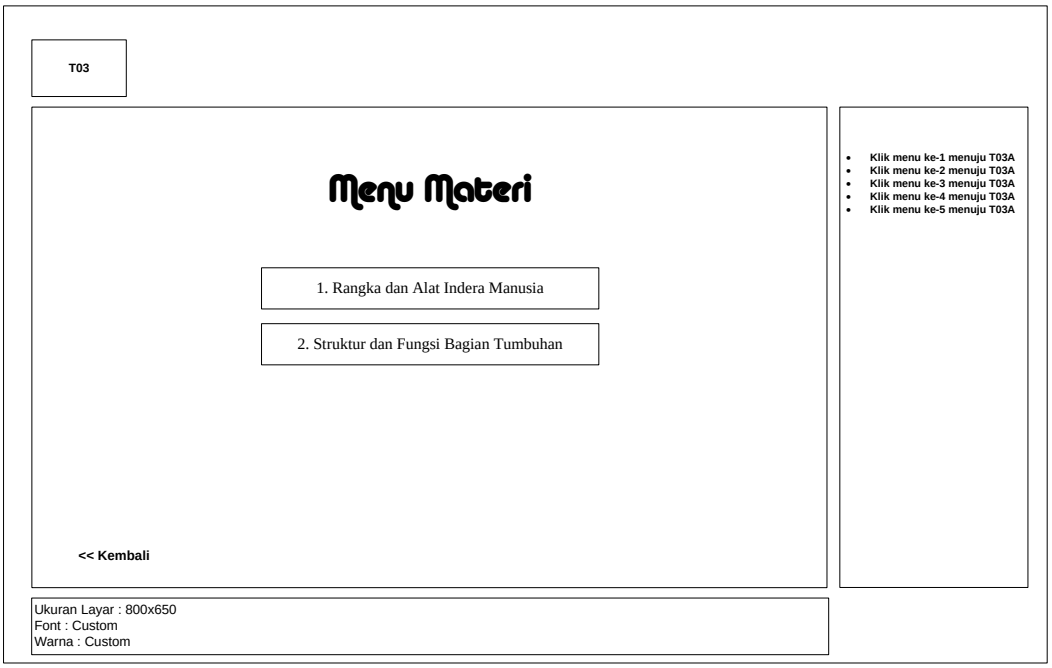
Gambar 3.46 Antarmuka Halaman Awal

2. Perancangan Antarmuka Menu Utama



Gambar 3.47 Antarmuka Menu Utama

3. Perancangan Antarmuka Menu Materi



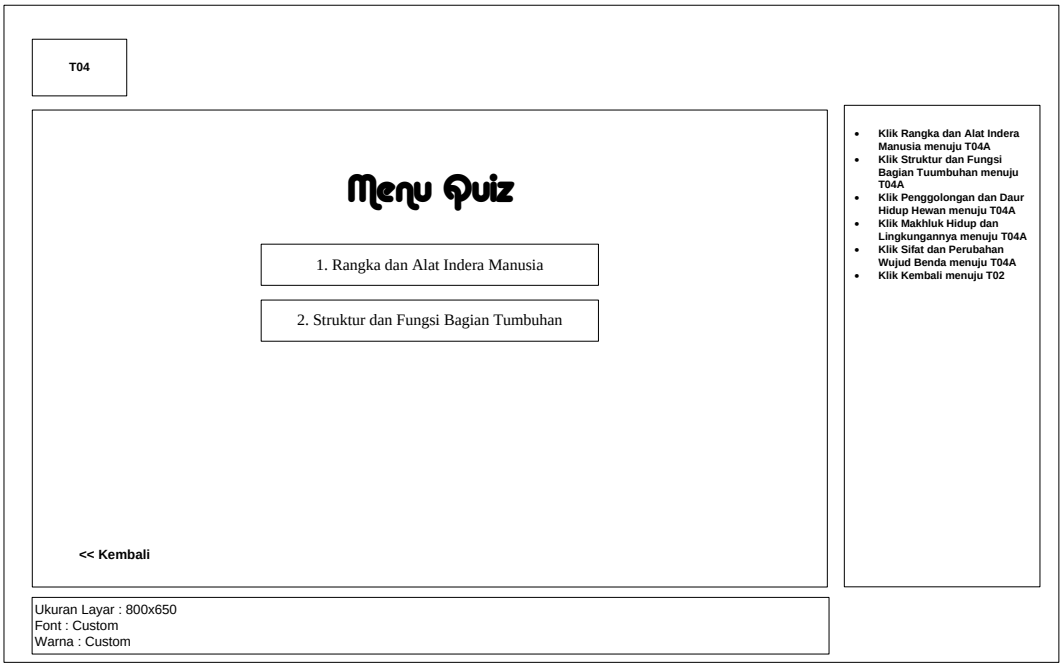
Gambar 3.48 Antarmuka Menu Materi

4. Perancangan Antarmuka Penyajian Augmented Reality



Gambar 3.49 Antarmuka Penyajian Augmented Reality

5. Perancangan Antarmuka Menu Quiz



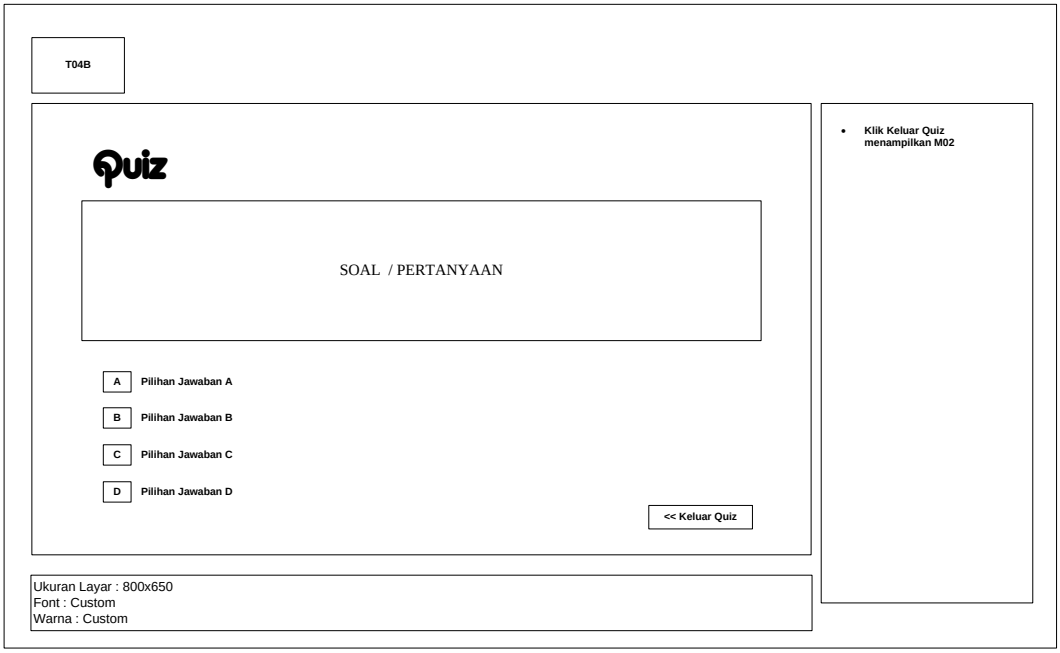
Gambar 3.50 Antarmuka Menu Quiz

6. Perancangan Antarmuka Petunjuk Quiz



Gambar 3.51 Antarmuka Petunjuk Quiz

7. Perancangan Antarmuka Tampilan Quiz



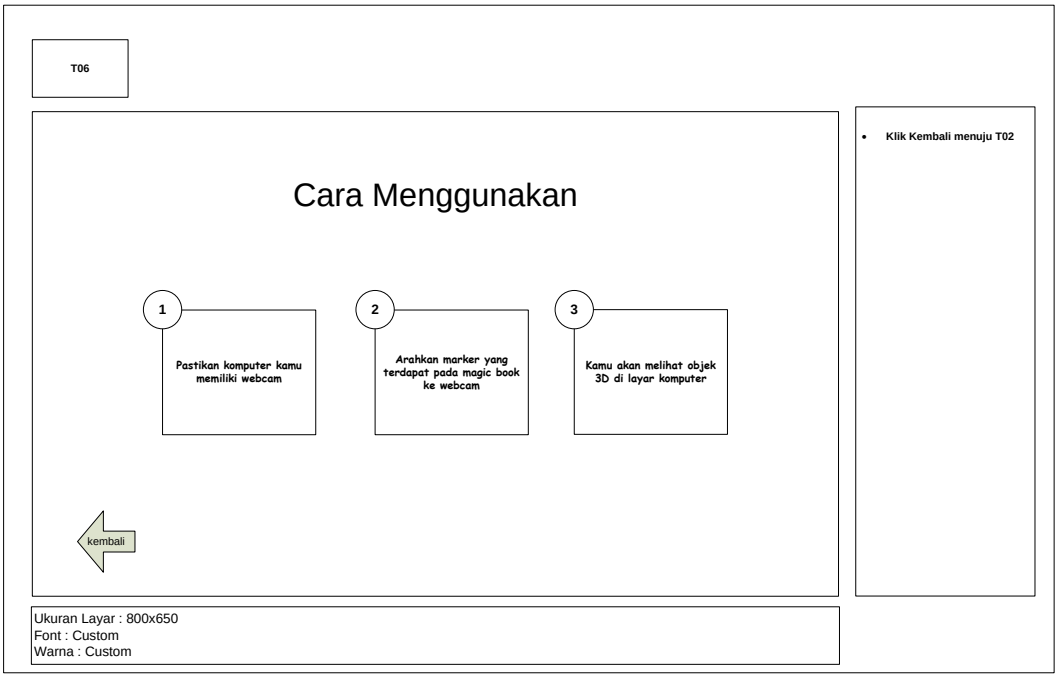
Gambar 3.52 Antarmuka Tampilan Quiz

8. Perancangan Antarmuka Tentang Saya



Gambar 3.53 Antarmuka Tentang Saya

9. Perancangan Antarmuka Cara Menggunakan



Gambar 3.54 Antarmuka Cara Menggunakan

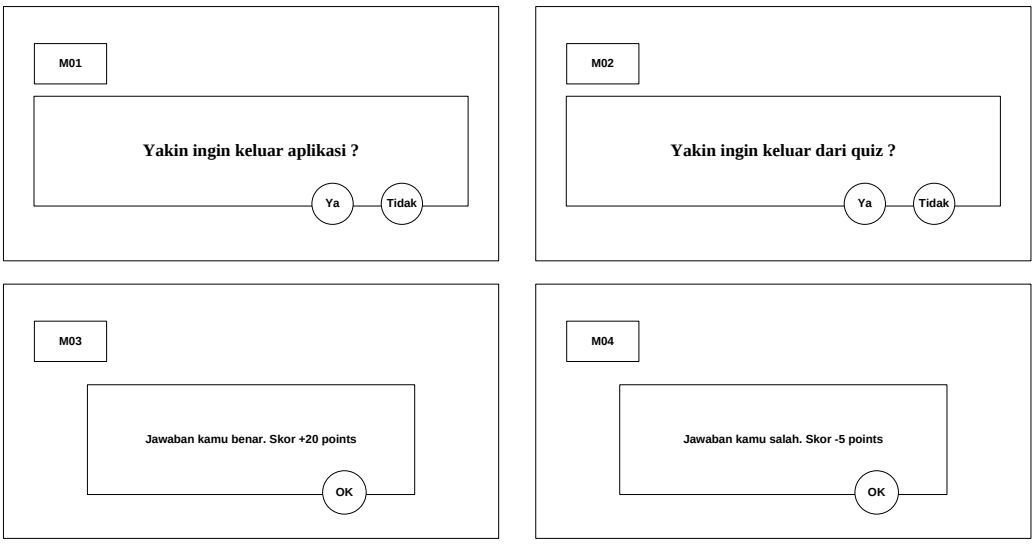
10. Perancangan Antarmuka Skor Akhir



Gambar 3.55 Antarmuka Skor Akhir

3.2.3 Perancangan Pesan

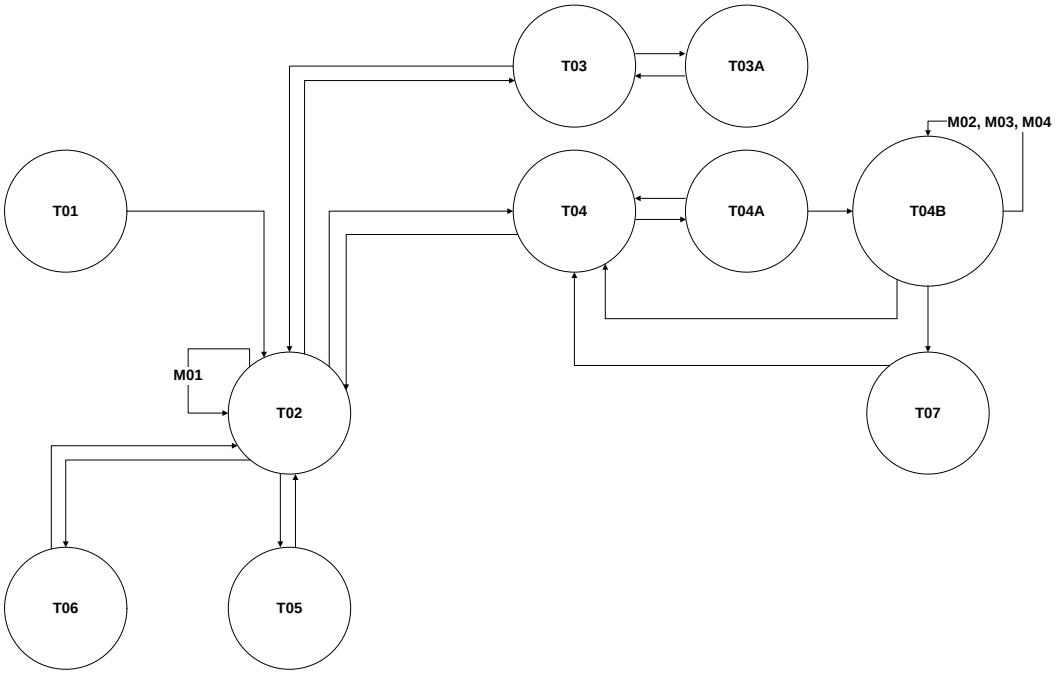
Pada gambar dibawah ini merupakan perancangan pesan yang terdapat pada aplikasi pengenalan rangka dan bagian tumbuhan menggunakan teknologi *augmented reality*. Adapun tampilan pesan yang ada adalah sebagai berikut :



Gambar 3.56 Perancangan Pesan

3.2.4 Jaringan Semantik

Jaringan semantik ini menunjukan form-form yang bisa diakses oleh pengguna seperti ditunjukan pada Gambar 3.23.



Gambar 3.57 Jaringan Semantik

Keterangan :

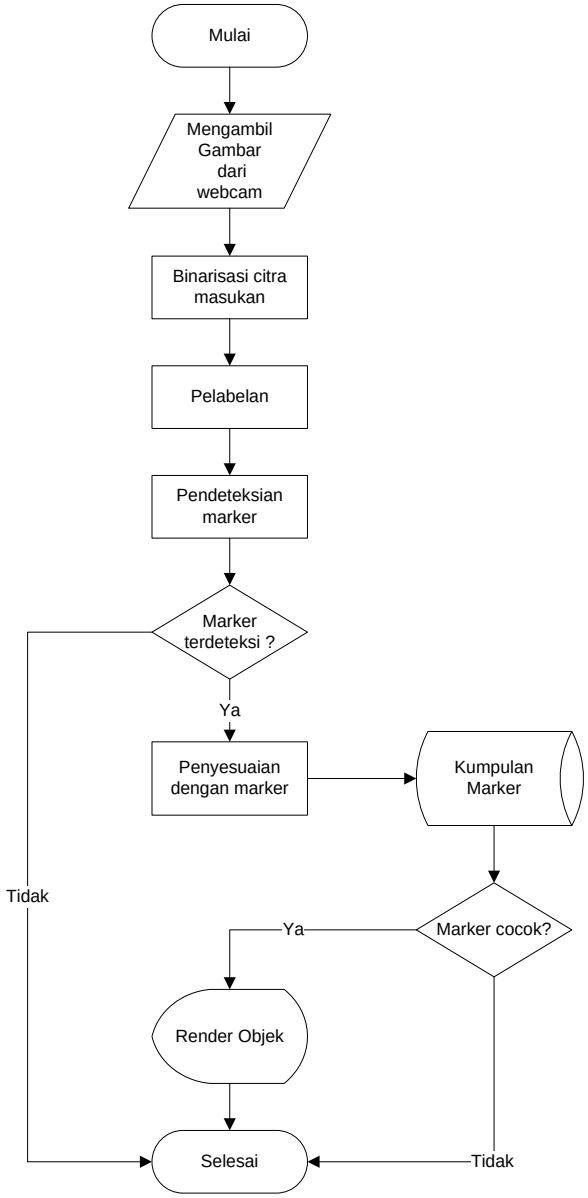
T01 : Halaman Awal
T02 : Menu Utama
T03 : Menu Belajar Materi
T03A : Tampil AR
T04 : Menu Quiz
T04A : Petunjuk Quiz
T04B : Tampilan Quiz
T05 : Tentang Saya
T06 : Cara Menggunakan
T07 : Skor Akhir

M01 : Pesan Konfirmasi Keluar Aplikasi
M02 : Pesan Konfirmasi Keluar Quiz
M03 : Pesan Jawaban Benar
M04 : Pesan Jawaban Salah

3.2.5 Perancangan Method

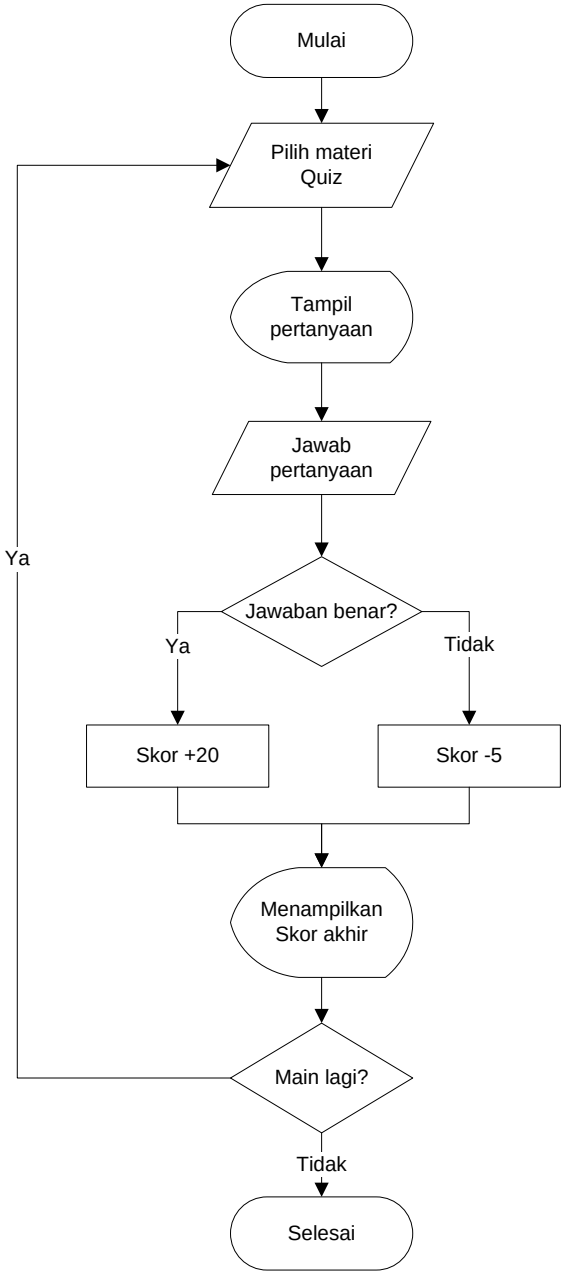
Perancangan komponen method merupakan perancangan yang dibuat setelah perancangan jaringan semantik dan antarmuka. Perancangan ini berfungsi untuk mendeskripsikan method-method yang berada di dalam aplikasi pengenalan rangka dan bagian tumbuhan menggunakan teknologi augmented reality. Adapun method-method yang terdapat dalam aplikasi yang akan dibangun adalah sebagai berikut :

1. Perancangan Method Augmented Reality



Gambar 3.58 Perancangan Method Augmented Reality

2. Perancangan Method Penyajian Quiz



Gambar 3.59 Perancangan Method Penyajian Quiz