BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

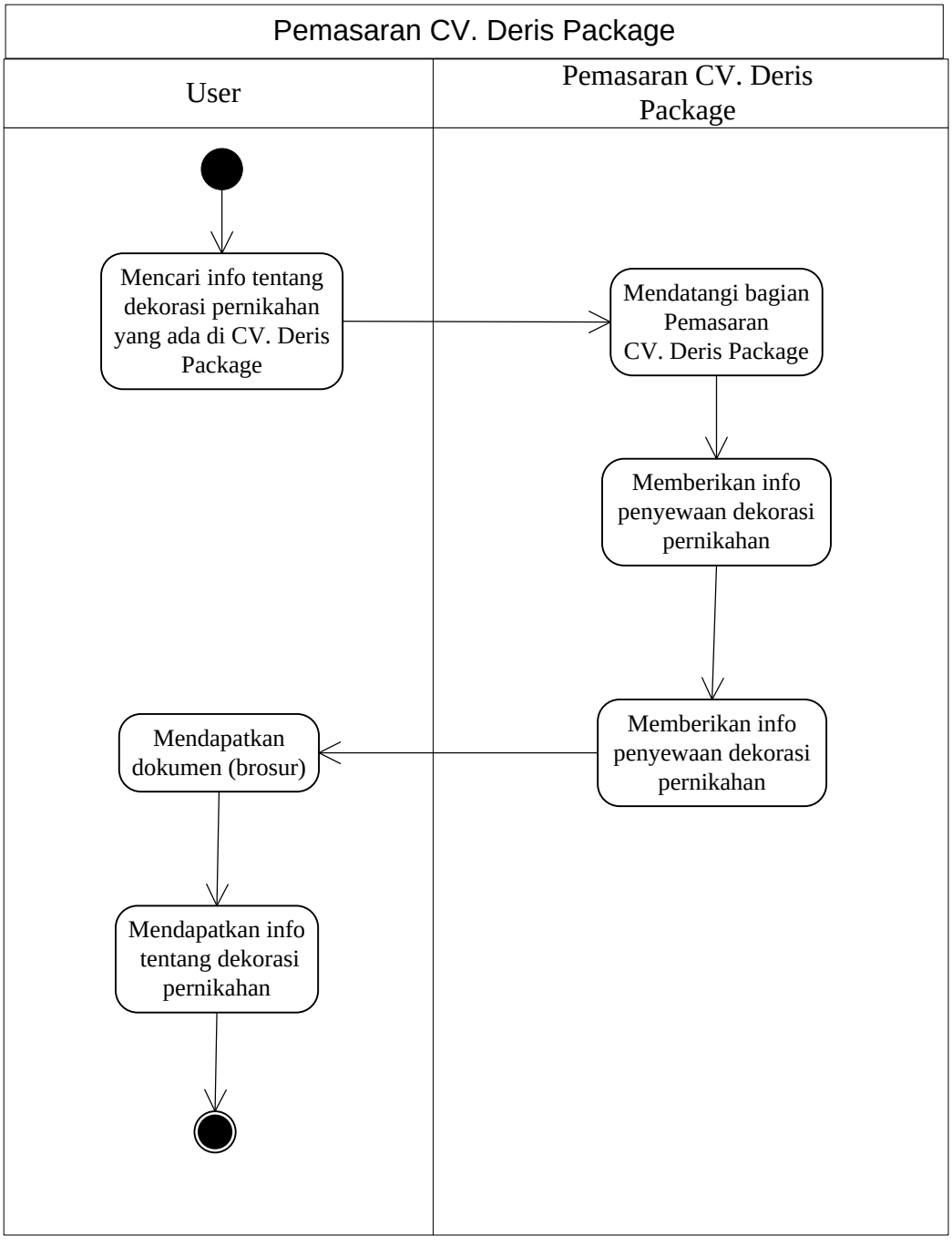
Analisis sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan solusi atau perbaikan. Dari hasil analisis tersebut dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efisien. Sistem yang dibuat merupakan aplikasi untuk mendeteksi *marker* dan menampilkan objek tiga dimensi yang telah dibuat dengan menggunakan *software* tiga dimensi (3D Max). Objek yang dibuat merupakan dekorasi pernikahan, seolah-olah pengguna berinteraksi langsung dengan objek virtual dalam dunia nyata yang disajikan dalam bentuk brosur.

3.1.1 Analisis Sistem Yang berjalan

Salah satu tahapan analisis sistem yaitu tahapan yang memberi gambaran tentang sistem yang sedang berjalan saat ini. Analisis ini bertujuan untuk memberikan gambaran bagaimana cara kerja dari sistem yang sedang berjalan. Prosedur yang sedang berjalan saat ini yaitu proses informasi pernikahan. Proses informasi dekorasi pernikahan di Deris *Package* adalah sebagai berikut :

1. User atau konsumen mencari info tentang dekorasi pernikahan yang ada di CV. Deris *Package* melalui blog.
2. Setelah mendapatkan informasi melalui blog, lalu konsumen mendatangi tempat bagian pemasaran CV. Deris *Package*.
3. Konsumen akan mendapatkan info dekorasi pernikahan dari brosur dan petugas.
4. Konsumen akan mendapatkan info dekorasi pernikahan berupa teks dan gambar dari brosur yang diberikan oleh petugas.

5. Konsumen mendapatkan info dekorasi pernikahan berupa teks dan gambar dari brosur yang diberikan oleh petugas



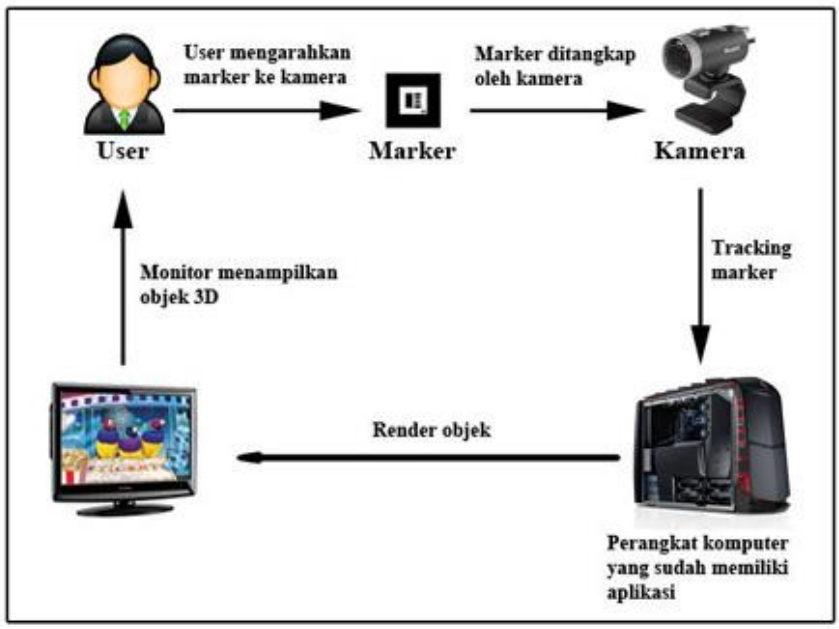
Gambar 3.1 Informasi Sistem Pemasaran Yang Berjalan

3.1.2 Analisis Masalah

Analisis masalah adalah langkah awal dari analisis sistem yang diperlukan untuk mengetahui permasalahan apa saja yang terjadi didalam sistem yang telah berjalan. Berdasarkan analisis sistem yang dilakukan dengan cara mengamati pemasaran yang telah dilakukan selama ini adalah pihak pemasaran hanya memberikan informasi dan gambaran dekorasi pernikahan yang akan disewakan kepada para calon penyewa dalam bentuk foto satu arah yang terdapat pada brosur maupun blog.

3.1.3 Analisis Arsitektur Sistem

Pada arsitektur aplikasi yang akan dibangun terdiri dari beberapa komponen, yaitu : *user* adalah pengguna yang menggunakan aplikasi brosur berbasis 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. Gambaran arsitektur sistem dapat dilihat pada gambar 3.2



Gambar 3.2 Arsitektur Sistem

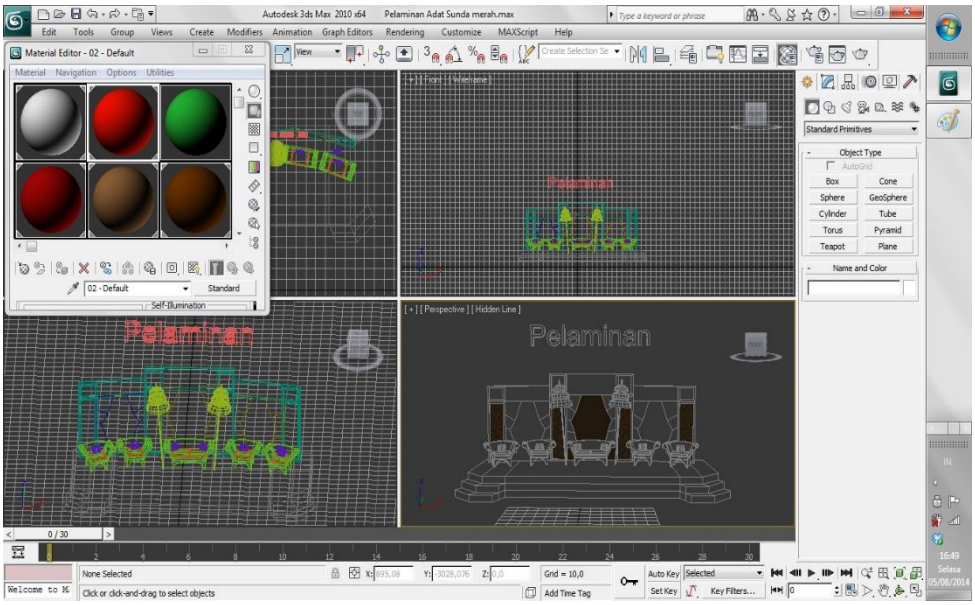
3.1.3.1 Perancangan Model Adat Dekorasi Pernikahan

3.1.3.1.1 Konsep Perancangan

Dalam pembuatan konsep perancangan objek dekorasi pernikahannya menggunakan *software* Autodesk 3ds Max 2010. Autodesk 3ds Max 2010 adalah *software* pembuatan model animasi tiga dimensi, rendering dan pembuatan grafis pada game dan animasi. Autodesk 3ds Max 2010 banyak digunakan oleh para praktisi dalam bisnis periklanan. Software ini banyak ragamnya, sesuai dengan ketersediaan fasilitas yang disediakan untuk memudahkan pengguna. Discreet 3DS Max merupakan software tiga dimensi yang dapat membuat objek tiga dimensi tampak realistis.

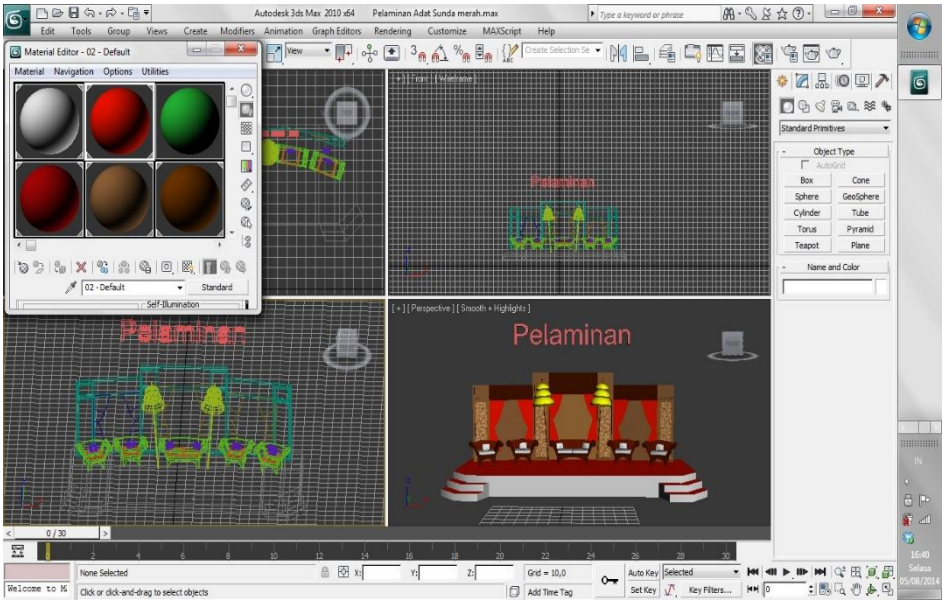
3.1.3.1.2 Teknis Perancangan

Berikut ini adalah tahap demi tahap teknis pengerjaan model objek dekorasi pernikahan 3D:



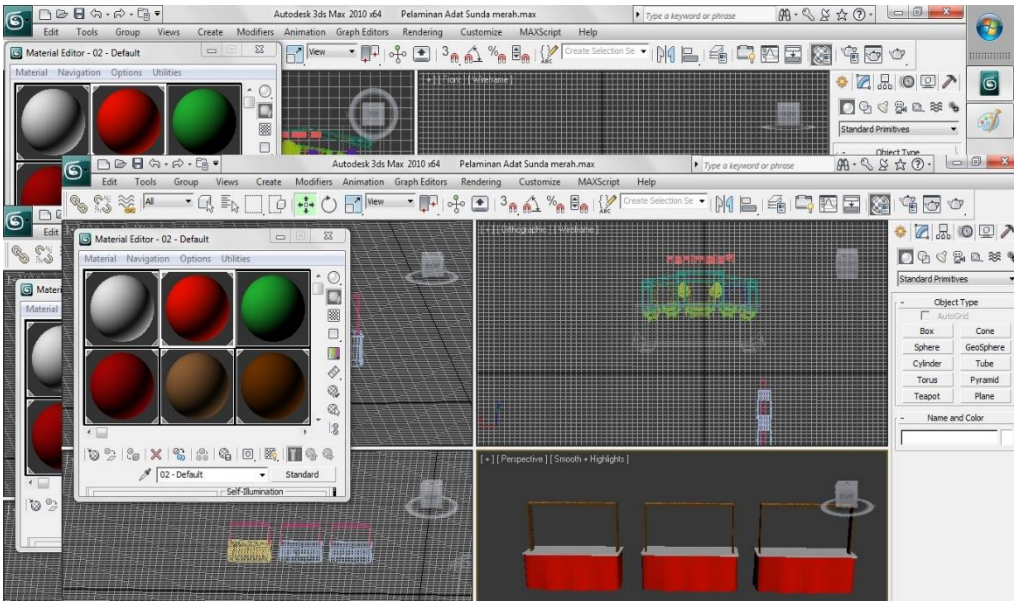
Gambar 3.3 Proses Awal Pembuatan Objek Dekorasi Sebelum Diberi Warna

model objek 3D adat dekorasi yang telah dibuat maka harus diteliti dahulu sebelum objek tersebut di kasih warna agar tidak terdapat kesalahan pada design dekorasi yang telah dibuat.

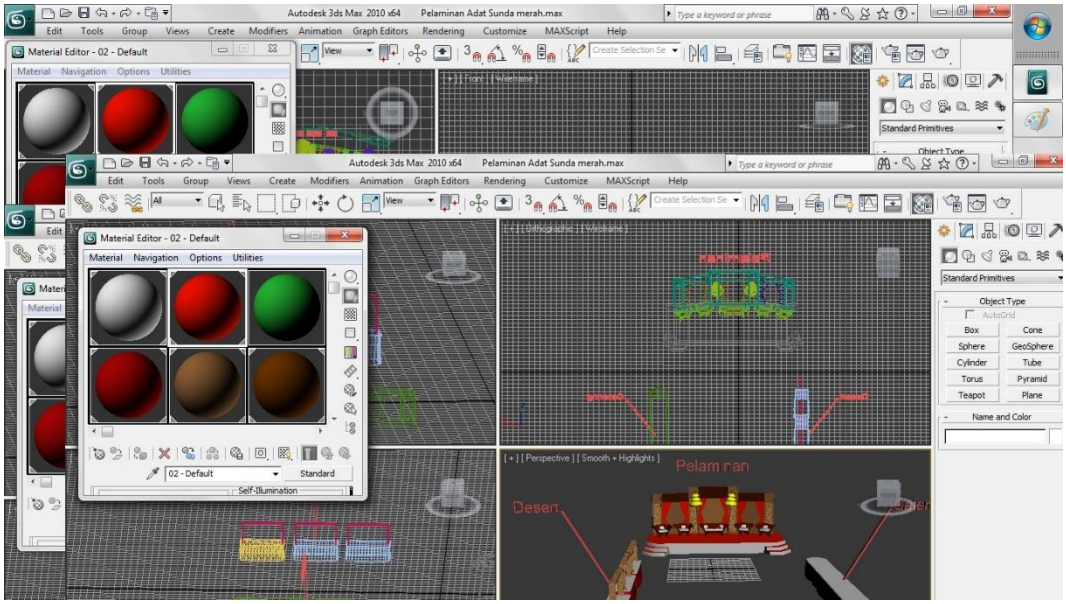


Gambar 3.4 Proses Pembuatan Objek Dekorasi Sesudah Diberi Warna

Setelah dipastikan model 3D dekorasi pernikahan tidak ada kesalahan langkah selanjutnya adalah menambahkan meja catering pada objek dekorasi sebelum nya.

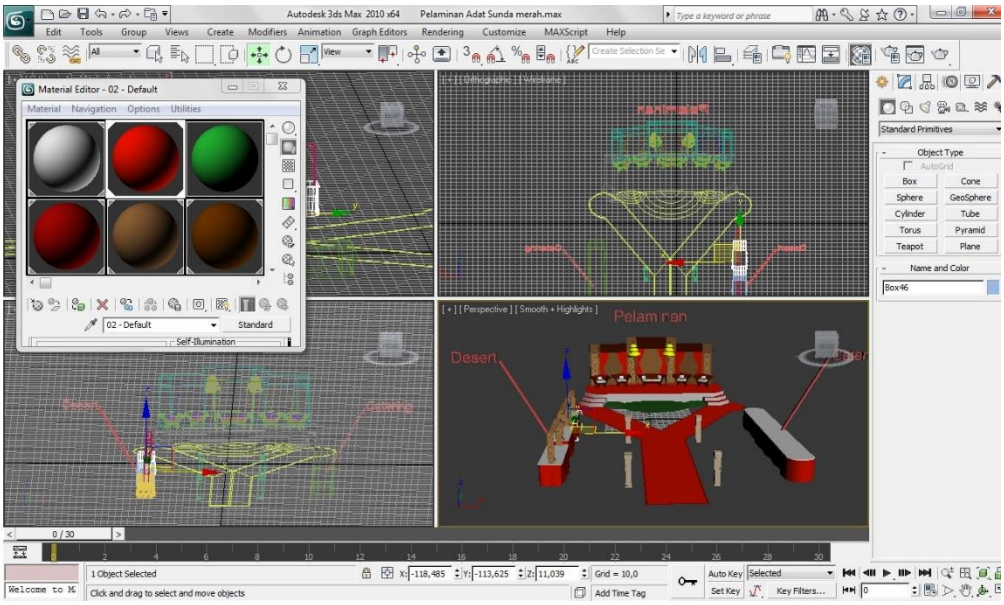


Gambar 3.5 Pembuatan Objek Meja Catering



Gambar 3.6 Hasil Objek Dekorasi Pernikahan dan Meja Catering

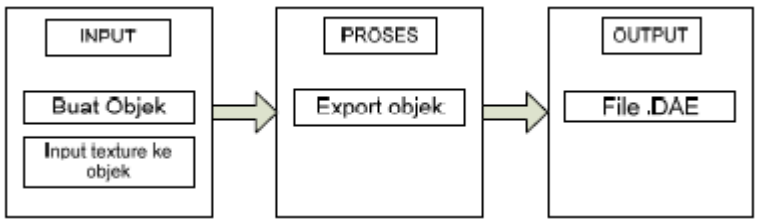
Selanjutnya masukan beberapa objek tambahan untuk dekorasi pernikahannya, karpet, dan ornamen-ornamen lainnya.



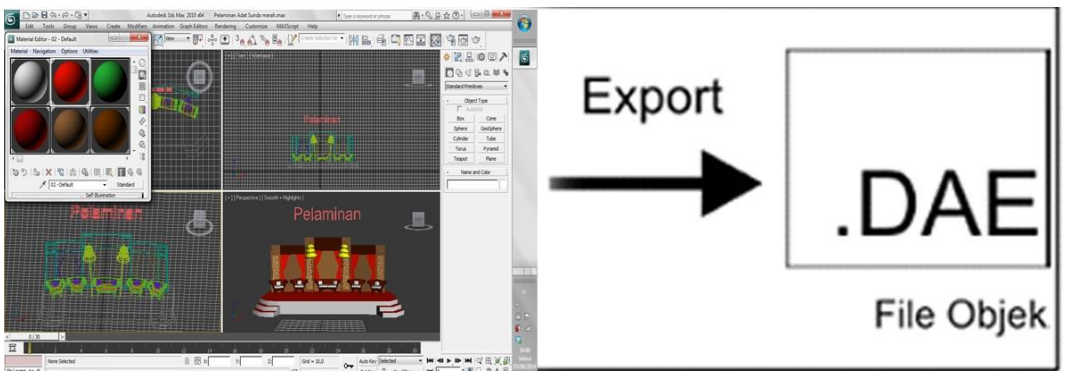
Gambar 3.7 Hasil Akhir Objek Dekorasi Pernikahan

3.1.3.2 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 aka di load sesuai dengan pola marker yang diteksi.



Gambar 3.8 Proses Pembentukan Data Objek 3D menurut Tomohiko Koyama [8]



Gambar 3.9 Proses Export Objek

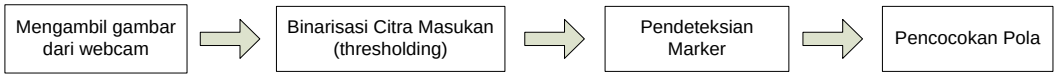
Dalam proses permodelan objek terdiri dari 3 langkah.

1. Menyesuaikan objek 3D dengan animasi atau benuk yang akan dibuat.
2. Memasukan teksture sesuai dengan objek 3D
3. Mengexport objek yang sudah dirancangan 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 menginstall aplikasi Open COLLADA agar objek dapat ditampilkan dan sesuai dengan yang dibuat.

3.1.3.3 Tracking Marker

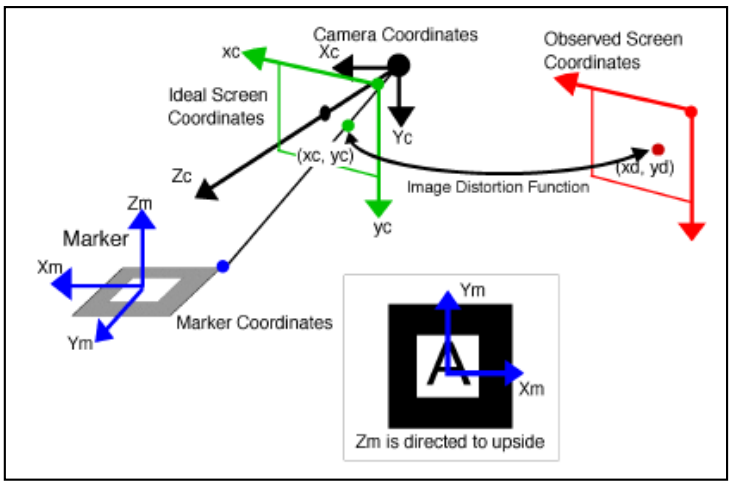
FLARToolkit memiliki kemampuan untuk mendeteksi gambar dan menghitung posisi gambar tersebut menggunakan webcam standar. Informasi posisi yang didapatkan akan dipergunakan untuk menempatkan objek atau model tiga dimensi atau video ke dalam posisi marker. Ada empat langkah, dalam proses kerja Tracking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



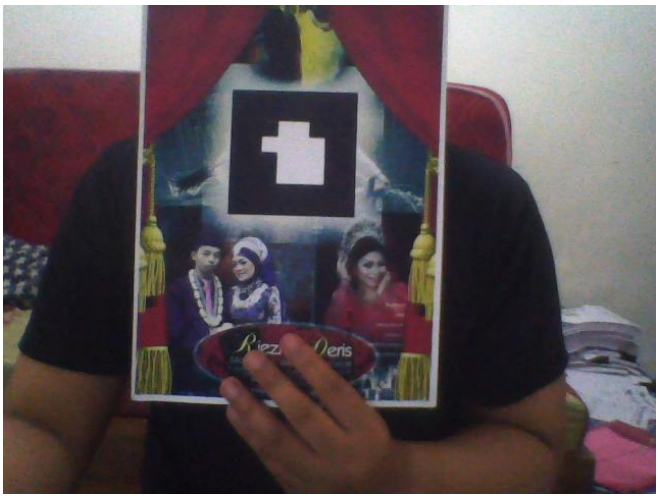
Gambar 3.10 Proses kerja Tracking Marker FLARToolKit menurut Galih Rakacitra [15]

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 aan digunakan untuk tahap berikutnya. Ilustrasi sistem koordinat lingkungan AR dapat dilihat pada gambar 3.4.



Gambar 3.11 Sistem Koordinat Lingkungan AR



Gambar 3.12 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:

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

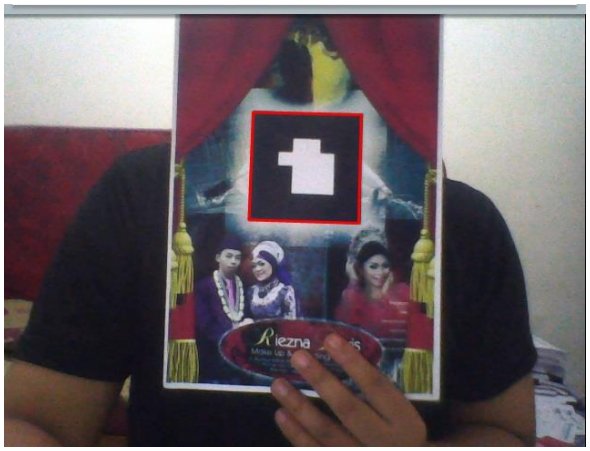
yang dalam hal ini, $fg(i, j)$ adalah citra hitam-putih, $fB(i, j)$ adalah citra biner, dan T adalah nilai ambang yang dispesifikasikan. Dengan operasi pengambangan tersebut, objek dibuat berwarna gelap (1 atau hitam) sedangkan latar belakang berwarna terang (0 atau putih).



Gambar 3.13 Binarisasi citra masukan

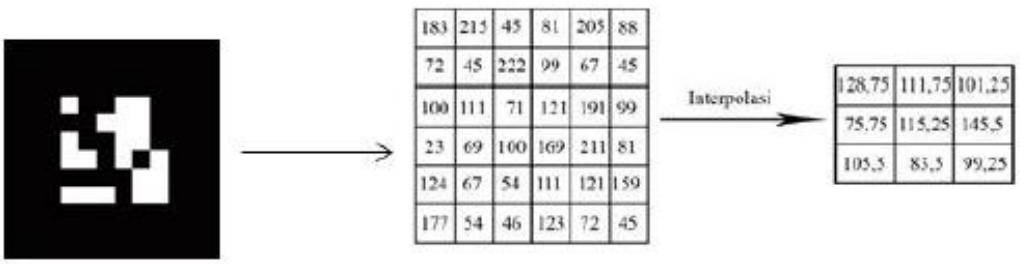
3. Pendeteksian Pelacakan Marker

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



Gambar 3.14 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 scale x dan scale y. Sebagai contoh perhitungan *matriks image* dengan angka-angka piksel *sampel* dapat dilihat di dalam gambar tabel matrik di bawah ini :



Gambar 3.15 Cara menentukan hitungan marker

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

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. FLARToolkit sudah memberikan nilai confidance kepada pola yang 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 ]) )
```

Pattern resoluton = 64, *Pattern To Border Ratio X* = 50, *Pattern To Border Ratio X* = 50, *Min Confidence* = 0,5 jika kecocokannya di atas nilai yang telah ditentukan maka polanya dinyatakan cocok.

3.1.3.4 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 3.16 Render Objek 3D

3.1.4 Analisis Metode

Occlusion adalah hubungan antara suatu benda dengan benda lain jika kita lihat dari suatu sudut pandang. Hal ini tentunya mengurangi informasi antar objek dalam lingkungan 3D, karena jika dilihat dari satu sudut pandang maka lingkungan 3D akan diproyeksikan kepada suatu bidang sehingga seolah-olah menjadi

lingkungan 2D. Pengurangan dimensi ini menyebabkan informasi interaksi antar objek seperti keadaan bersinggungan atau beririsan.

Occlusion Based Detection juga berguna untuk mendeteksi adanya suatu objek yang saling bertabrakan atau menghalangi, perancangan aplikasi ini menggunakan obyek *kubus* yang berfungsi sebagai virtual button. Teknik *occlusion based detection* pada aplikasi yang diterapkan para marker dan objek virtual button. Secara sederhana *occlusion bassed detection* hanya mendefinisikan keadaan dimana suatu marker tidak terdeteksi karena tertutup oleh benda lain.

Akan tetapi ada beberapa persyaratan bahwa objek tersebut mengalami *occlusion* yaitu dimana persamaan 1 dan 2 terpenuhi

$$O2x - batas \leq O1\ x \leq + batas \qquad (1)$$

$$O2y - batas \leq O1\ y \leq + batas \qquad (2)$$

Keterangan :

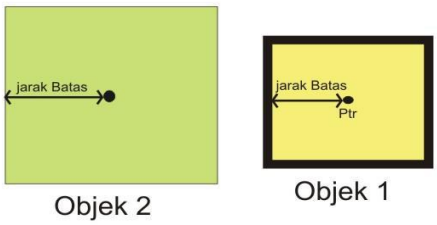
O2x – batas = setengah bagian dari objek O2 pada sumbu x

O2y – batas = setengah bagian dari objek O2 pada sumbu y

O1x – batas = setengah bagian dari objek O1 pada sumbu x

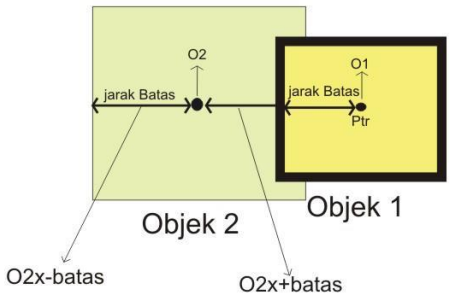
O1y – batas = setengah bagian dari objek O1 pada sumbu y

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 3.5. Misalkan titik tengah objek1 diwakili oleh matrik O1(x,y) dan objek 2 diwakilili oleh matrik O2(x,y).



Gambar 3.17 Objek matrik O1(x,y) dan O2 (x,y)

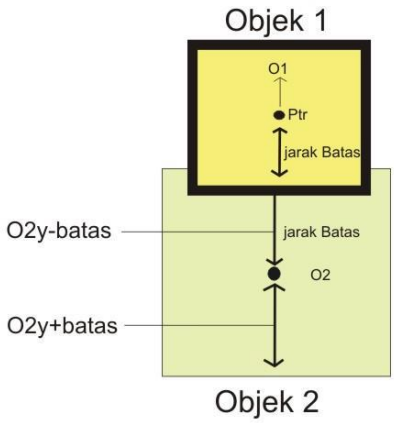
Untuk persamaan (1) tidak terjadi *event* jika :



Gambar 3.18 Objek matrik O1 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.*

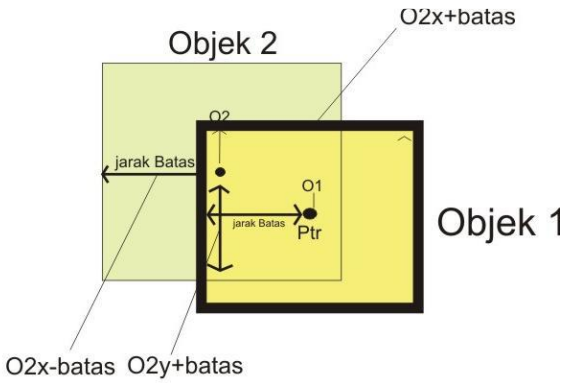
Untuk persamaan (2) tidak terjadi *event* jika :



Gambar 3.19 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 3.20 Objek matrik O1 (x,y) yang memenuhi persamaan (1) dan (2)

Dari Gambar 3.11 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).

3.1.5 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 brosur berbasis *Augmented Reality* di CV. Deris *Package* ini meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan memakai aplikasi.

3.1.5.1 Analisis Kebutuhan Perangkat Keras

Analisis Perangkat keras atau *hardware* merupakan salah satu hal yang penting karena tanpa *hardware* yang memenuhi syarat, aplikasi yang akan dibuat tidak akan dapat berjalan. Berikut spesifikasi standar perangkar keras yang dapat dipergunakan untuk membangun aplikasi brosur berbasis *Augmented Reality* ini dan spesifikasi perangkat keras yang dapat dipergunakan untuk menjalankan aplikasi brosur berbasis *Augmented Reality* ini yaitu :

Tabel 3.2 Kebutuhan Perangkat Keras Pengembang

No	Perangkat Keras	Spesifikasi
1	Processor	Intel Core Duo 2.27 GHz
2	Monitor	14 inch
3	Graphic Card	Internal 829 MB
4	Memori	RAM 2 GB
5	Hard disk drive	Free Space 500 MB
6	Webcam	3 MP
7	Keyboard dan Mouse	Standar

Pembangunan aplikasi yang akan dibangun membutuhkan spesifikasi perangkat keras, spesifikasi minimum perangkat keras untuk menjalankan aplikasi ini dapat dilihat pada Tabel 3.2 Kebutuhan Perangkat Keras Pengguna

Tabel 3.3 Kebutuhan Perangkat Keras Pengguna

No	Perangkat Keras	Spesifikasi
1	Processor	Processor dengan kecepatan 1.8 Ghz
2	Graphic Card	VGA 512 MB
3	Memori	RAM 1 GB
4	Hard disk drive	Free Space 60 MB
5	Webcam	Minimal 0.3 MP
6	Keyboard dan Mouse	Standar

3.1.5.2 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 dibutuhkan untuk membangun aplikasi brosur berbasis *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
3	Tool Desain	Adobe Photoshop CS5
4	Tool Compiler	Adobe Flash Player 10
5	Tool Compiler	Adobe Flash Builder 4

Untuk dapat menggunakan aplikasi ini, perangkat lunak yang dibutuhkan oleh pengguna yaitu tool compiler Adobe Flash Player 10.

3.1.5.3 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 brosur berbasis *Augmented Reality* ini antara lain pengguna dapat mengerti dan memahami komputer sehingga dapat menggunakan aplikasi yang akan dibangun. Aplikasi ini dapat digunakan oleh umum yang memahami komputer.

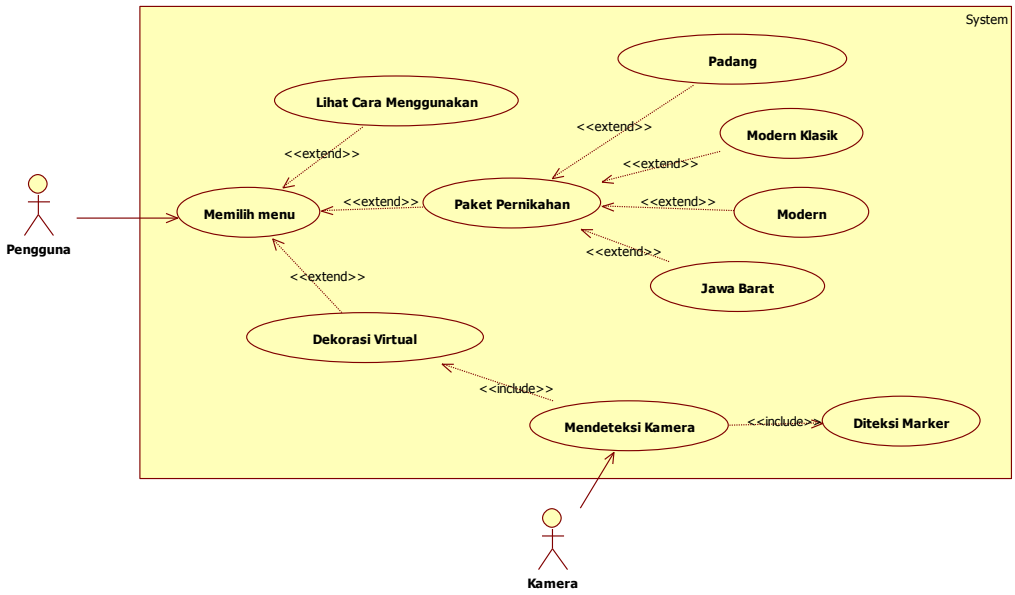
3.1.6 Analisis Kebutuhan fungsional

Analisis kebutuhan fungsional menggambarkan kebutuhan sistem yang akan dibangun pada aplikasi brosur berbasis *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*).

3.1.6.1 Use Case Diagram

Digunakan untuk memodelkan atau menggambarkan batasan sistem dan fungsi-fungsi utamanya. Mendiskripsikan fungsi dari sebuah sistem dari perspektif pengguna, *use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri.

Didalam sistem terdapat pengguna yaitu pemakai aplikasi. Peran aktor yang ada dapat terlihat pada diagram *Use Case* pada gambar berikut.



Gambar 3.21 *Use Case Diagram*

Use Case memilih menu ini menggambarkan proses dimana *user* harus memilih menu terlebih dahulu sebelum menggunakan aplikasi.

3.1.6.1.1 Skenario *Use Case* Memilih Menu

Use case memilih menu ini menggambarkan proses dimana *User* memilih menu ketika sedang menggunakan aplikasi.

Tabel 3.5 Skenario *Use Case* Memilih Menu

Nama	Memilih Menu
Aktor	Pengguna
Trigger	Pengguna memilih menu yang akan dilihat atau digunakan.
Skenario Utama	
Kondisi Awal	Sistem Cara Menggunakan dan Dekorasi Virtual
Aksi Aktor	**Reaksi Sistem**
	1. Sistem menampilkan utama
2. Memilih menu yang diinginkan.	
	3. Menampilkan menu yang dipilih oleh pengguna.
Kondisi Akhir	Menampilkan informasi sesuai dengan menu yang dipilih.

3.1.6.1.2 Skenario *Use Case* Lihat Cara menggunakan

Use case memilih menu Lihat cara menggunakan ini menggambarkan proses dimana *User* memilih menu cara menggunakan ketika sedang menggunakan aplikasi.

Tabel 3.6 Skenario *Use Case* Lihat Cara Menggunakan

Nama	Lihat Cara Menggunakan
Aktor	Pengguna
Trigger	Menampilkan informasi tentang penggunaan aplikasi dan apa saja yang

	dibutuhkan untuk menjalankan aplikasi.
Skenario Utama	
Kondisi Awal	Sistem menampilkan menu utama.
Aksi Aktor	**Reaksi Sistem**
	1. Sistem menampilkan menu utama
2. Aktor memilih menu Cara Menggunakan.	
	3. Sistem menampilkan informasi Cara Menggunakan.
Kondisi Akhir	Sistem menampilkan menu Cara Menggunakan.

3.1.6.1.3Skenario *Use Case* Memilih Dekorasi Virtual

Use case memilih Dekorasi virtual ini menggambarkan proses dimana user memilih dekorasi pernikahan yang diinginkan.

Tabel 3.7 Skenario *Use Case* Memilih Dekorasi Virtual

Nama	Memilih Dekorasi Virtual
Aktor	Pengguna,
Trigger	Pengguna memilih adat dekorasi.
Skenario Utama	
Kondisi Awal	Komputer atau laptop memiliki kamera.
Aksi Aktor	**Reaksi Sistem**
	1. Sistem menampilkan menu utama
2. aktor mengarahkan marker	
	3. Sistem menampilkan dekorasi adat yang di pilih oleh aktor.
Kondisi Akhir	Sistem memilih menu dekorasi adat.

3.1.6.1.4 Skenario *Use Case* Paket Pernikahan

Use case Lihat Paket Pernikahan ini menggambarkan proses dimana user memilih Lihat Paket Pernikahan yang diinginkan.

Tabel 3.8 Skenario *Use Case* Paket Pernikahan

Nama	Memilih Lihat Paket Pernikahan	
Aktor	Pengguna,	
Trigger	Pengguna memilih paket pernikahan adat dekorasi sesuai adat yang dipilih	
Skenario Utama		
Kondisi Awal	Komputer atau laptop memiliki kamera.	
Aksi Aktor		**Reaksi Sistem**
		1. Sistem menampilkan menu paket pernikahan
2. Aktor memilih paket pernikahan sesuai dengan adat yang telah dipilih sebelumnya.		
		3. Sistem menampilkan menu paket pernikahan yang di pilih oleh pengguna.
Kondisi Akhir	Sistem menampilkan menu lihat paket pernikahan	

3.1.6.1.5 Skenario *Use Case* Menggunakan Dekoravi Virtual

Use Case menggunakan dekorasi virtual ini menggambarkan proses dimana *user* memilih cara menggunakan Dekorasi virtual ketika sedang menggunakan aplikasi.

Tabel 3.9 Skenario *Use Case* Menggunakan Dekorasi Virtual

Nama	Menggunakan Dekorasi Virtual
Aktor	Pengguna

Trigger	Menggunakan aplikasi Dekoraasi Virtual
Skenario Utama	
Kondisi Awal	Sistem menampilkan menu utama.
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu dekorasi virtual pada adat yang telah dipilih	
	2. Sistem mendeteksi kamera
3. Aktor menekan tombol allow	
4. Sistem menampilkan aplikasi tanpa objek	
5. Aktor menunjukan marker ke depan webcam	
	6. Sistem menampilkan aplikasi dengan objek
Kondisi Akhir	Menampilkan informasi sesuai menu yang dipilih.
Skenario Alternatif	
Kondisi Awal	Sistem menampilkan menu utama
Aksi aktor	**Reaksi sistem**
1. Aktor memilih menu dekorasi virtual pada adat yang telah dipilih	
	2. Sistem mendeteksi kamera
3. Aktor menekan tombol deny	
Kondisi Akhir	Kamera tidak terditeksi

3.1.6.1.6 Skenario *Use Case* Mendeteksi Kamera

Use Case mendeteksi kamera ini menggambarkan proses dimana *user* harus mendeteksi kamera terlebih dahulu sebelum menggunakan aplikasi.

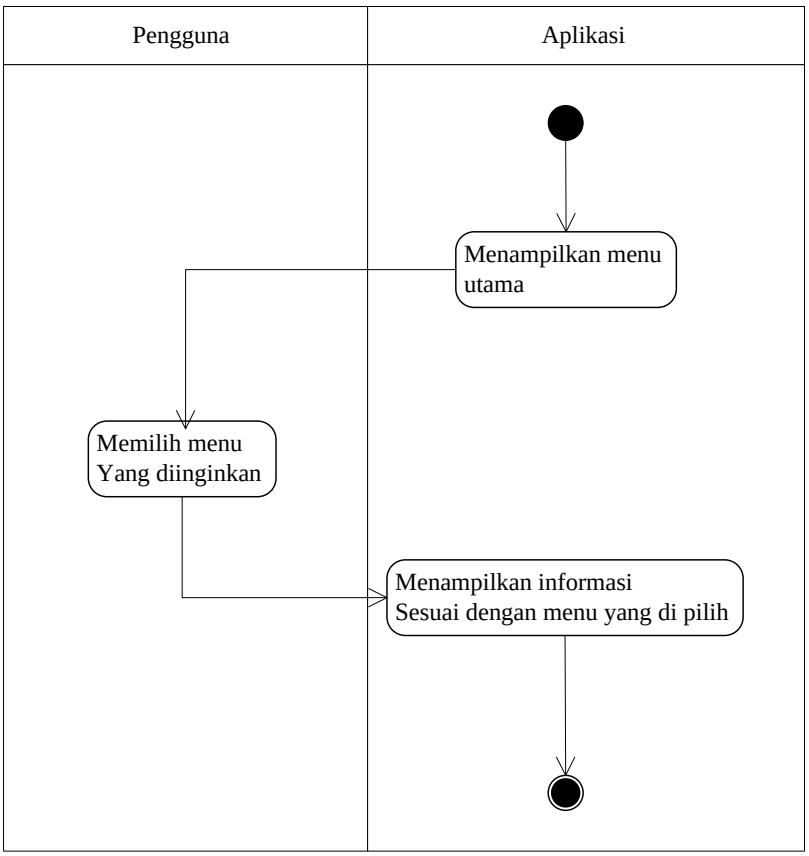
Tabel 3.10 Skenario *Use Case* Mendeteksi Kamera

Nama	Mendeteksi kamera
Aktor	Pengguna, Kamera
Trigger	Pengguna mendeteksi kamera sebelum menggunakan aplikasi.
Skenario Utama	
Kondisi Awal	Komputer atau laptop memiliki kamera.
Aksi Aktor	**Reaksi Sistem**
1. Pengguna memasang kamera.	
	2. Sistem mendeteksi ketersediaan kamera.
Kondisi Akhir	Kamera terdeteksi dan aplikasi siap untuk digunakan.
Skenario Alternatif	
Kondisi Awal	Komputer atau laptop memiliki kamera.
Aksi Aktor	**Reaksi Sistem**
1. Sistem mendeteksi ketersediaan kamera	
Kondisi Akhir	Kamera tidak terdeteksi dan aplikasi tidak dapat dilanjutkan.

3.1.6.2.1 *Activity Diagram*

Diagram *activity* menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing akir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Penggambaran *activity* diagram memiliki kemiripan dengan *flowchart* diagram. *Activity* diagram memodelkan *event-event* yang terjadi pada *Use Case* dan digunakan untuk permodelan aspek dinamis dari sistem.

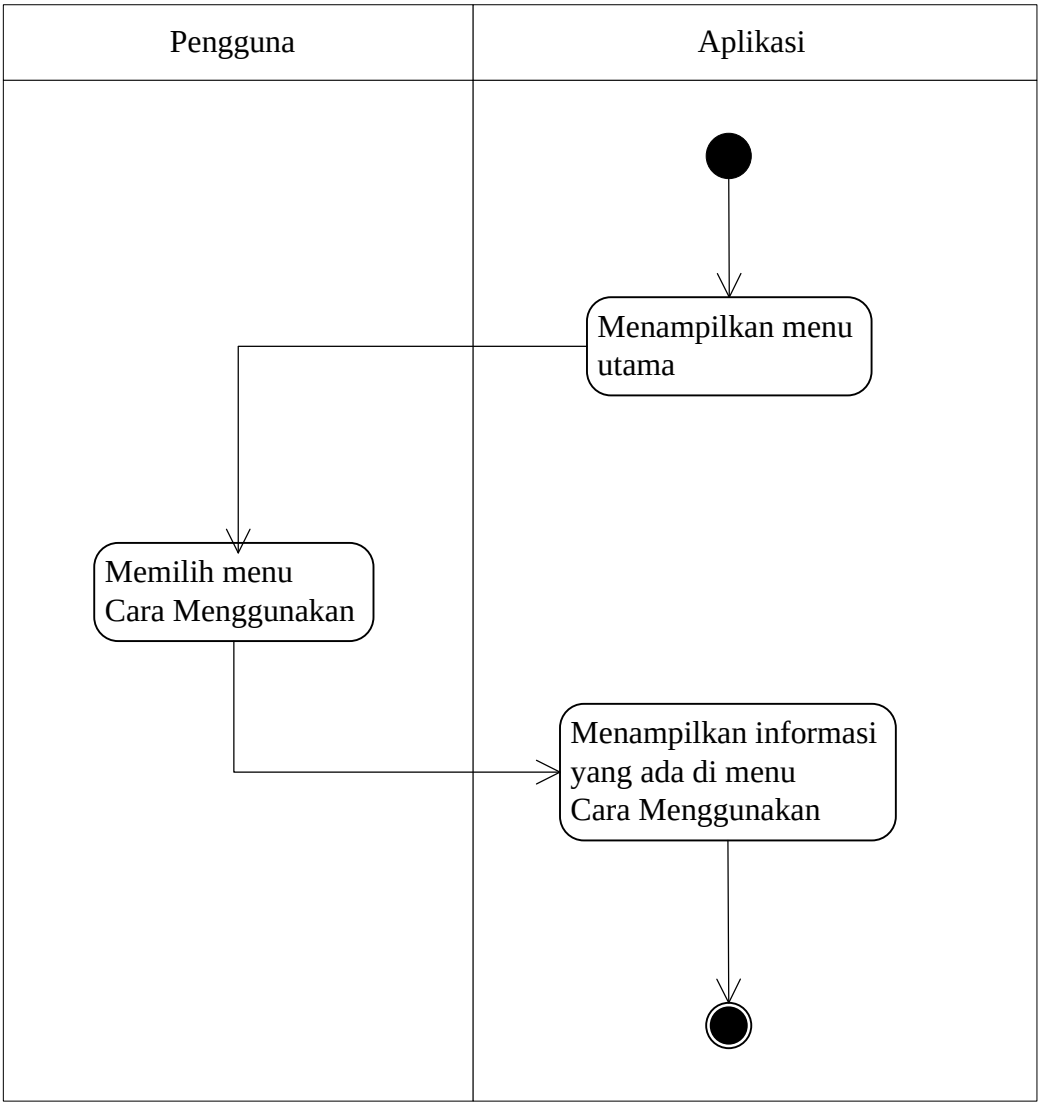
3.1.6.2.1.1 *Activity Diagram* Memilih Menu



Gambar 3.22 *Activity Diagram* memilih menu

Gambar 3.11, menggambarkan aktivitas ketika pengguna memilih menu. Pengguna memilih salah satu menu, selanjutnya sistem akan menampilkan konten yang sesuai dengan jenis menu yang dipilih.

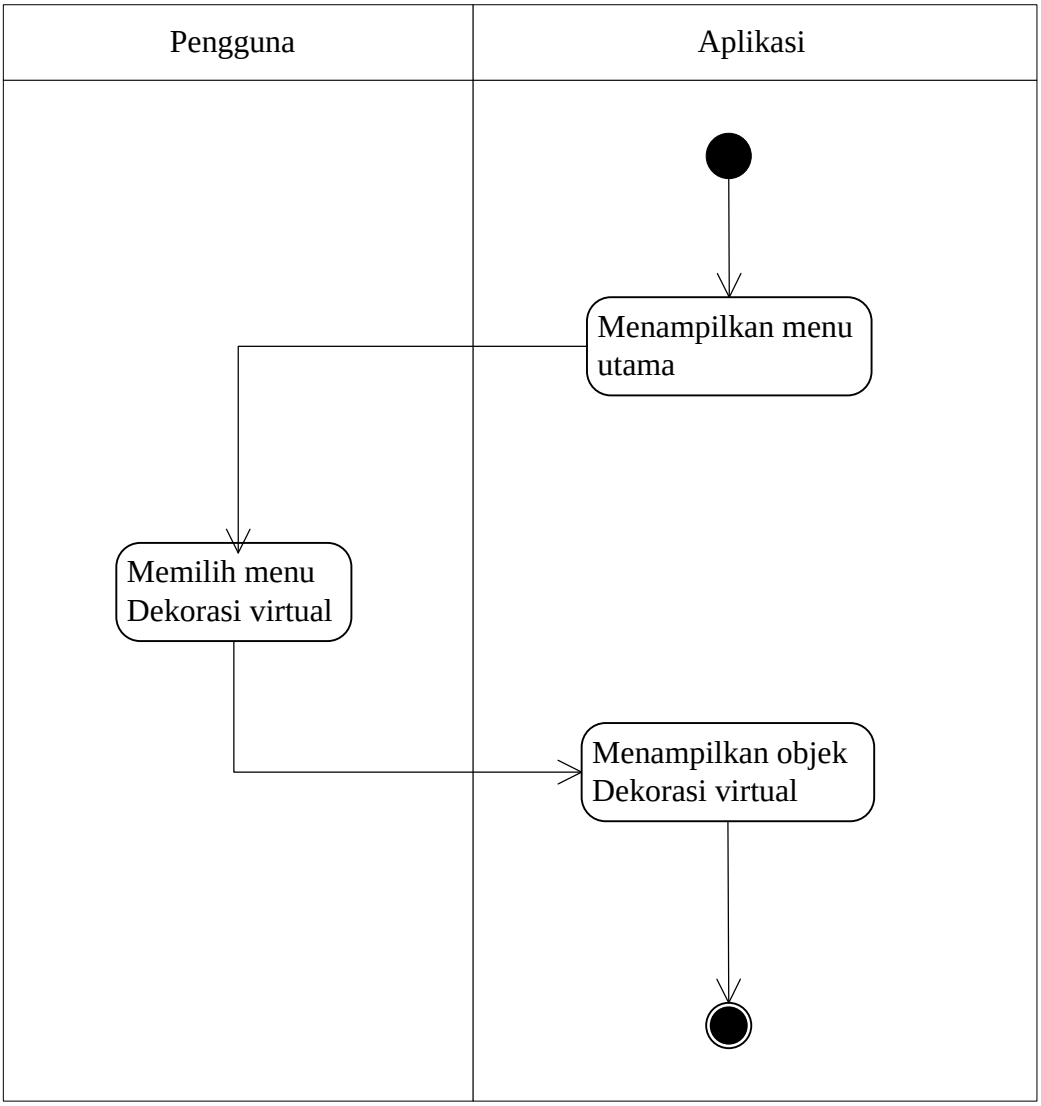
3.1.6.2.1.2 *Activity Diagram* Lihat Cara Menggunakan



Gambar 3.23 *Activity Diagram* Lihat Cara Menggunakan

Gambar 3.12 menggambarkan aktivitas ketika pengguna memilih menu cara menggunakan. Pengguna menekan tombol cara menggunakan, selanjutnya sistem akan menampilkan menu cara menggunakan. Pada menu ini menjelaskan tentang tata cara penggunaan aplikasi dan apa saja yang dibutuhkan untuk menjalankan aplikasi.

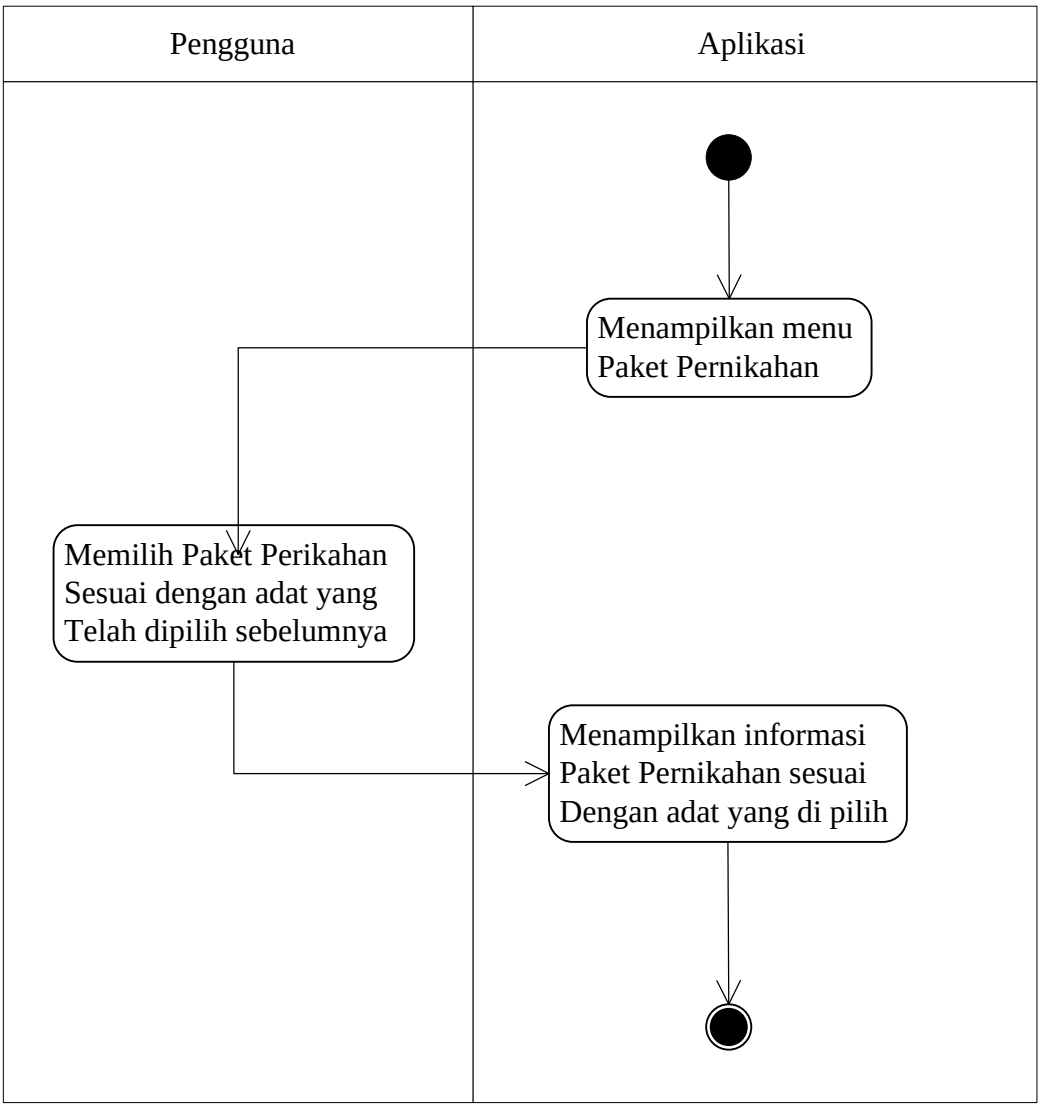
3.1.6.2.1.3 *Activity Diagram* Dekorasi Virtual



Gambar 3.24 *Activity Diagram* Dekorasi virtual

Gambar 3.13 menggambarkan aktivitas ketika pengguna memilih menu dekorasi virtual. Pengguna menekan tombol memilih dekorasi virtual, selanjutnya sistem akan menampilkan adat dekorasi yang dipilih oleh pengguna.

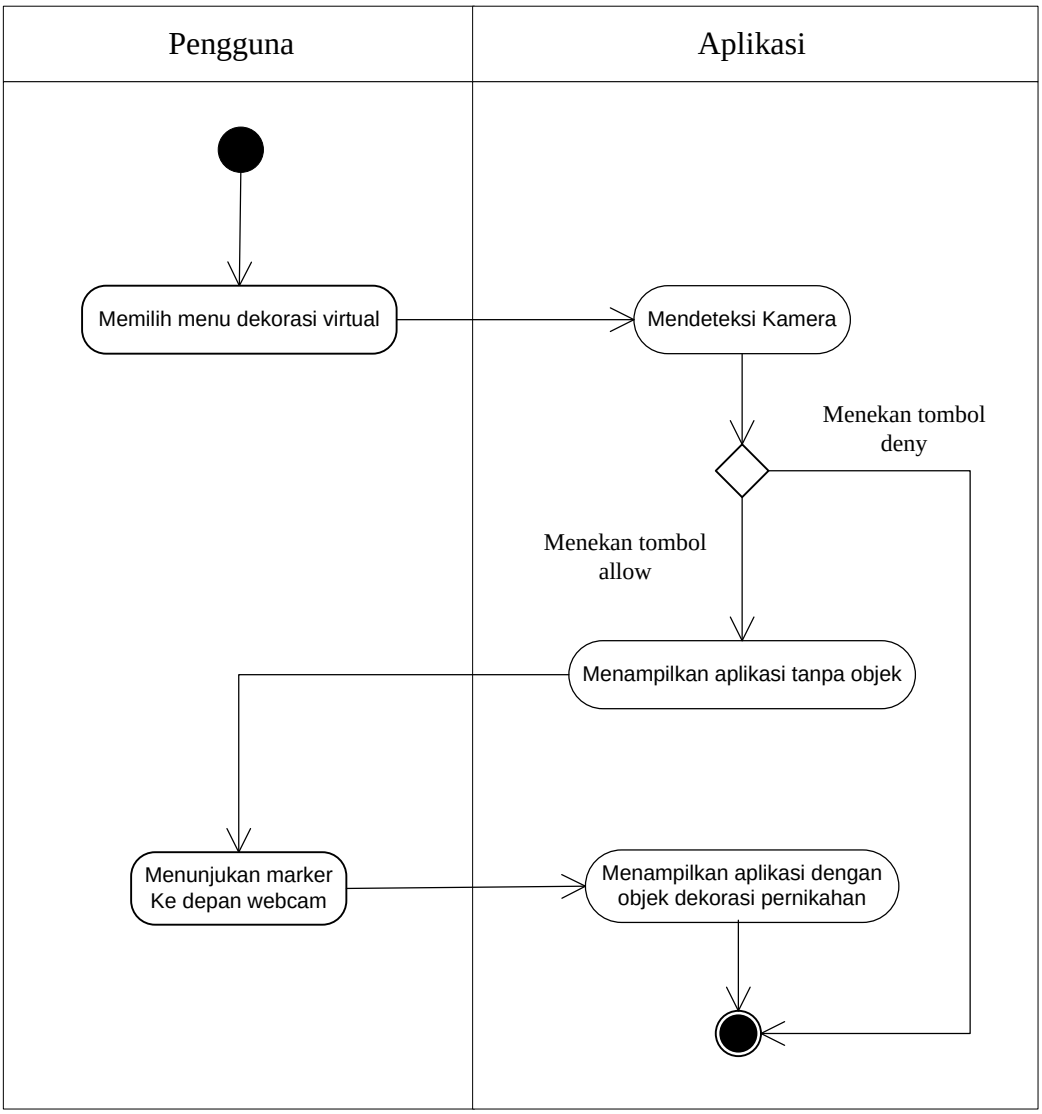
3.1.6.2.1.4 *Activity Diagram* Paket Pernikahan



Gambar 3.25 *Activity Diagram* Paket Pernikahan

Gambar 3.15, menggambarkan aktivitas ketika pengguna memilih menu lihat paket pernikahan, Selanjutnya sistem akan menampilkan konten yang sesuai dengan jenis menu adat yang dipilih

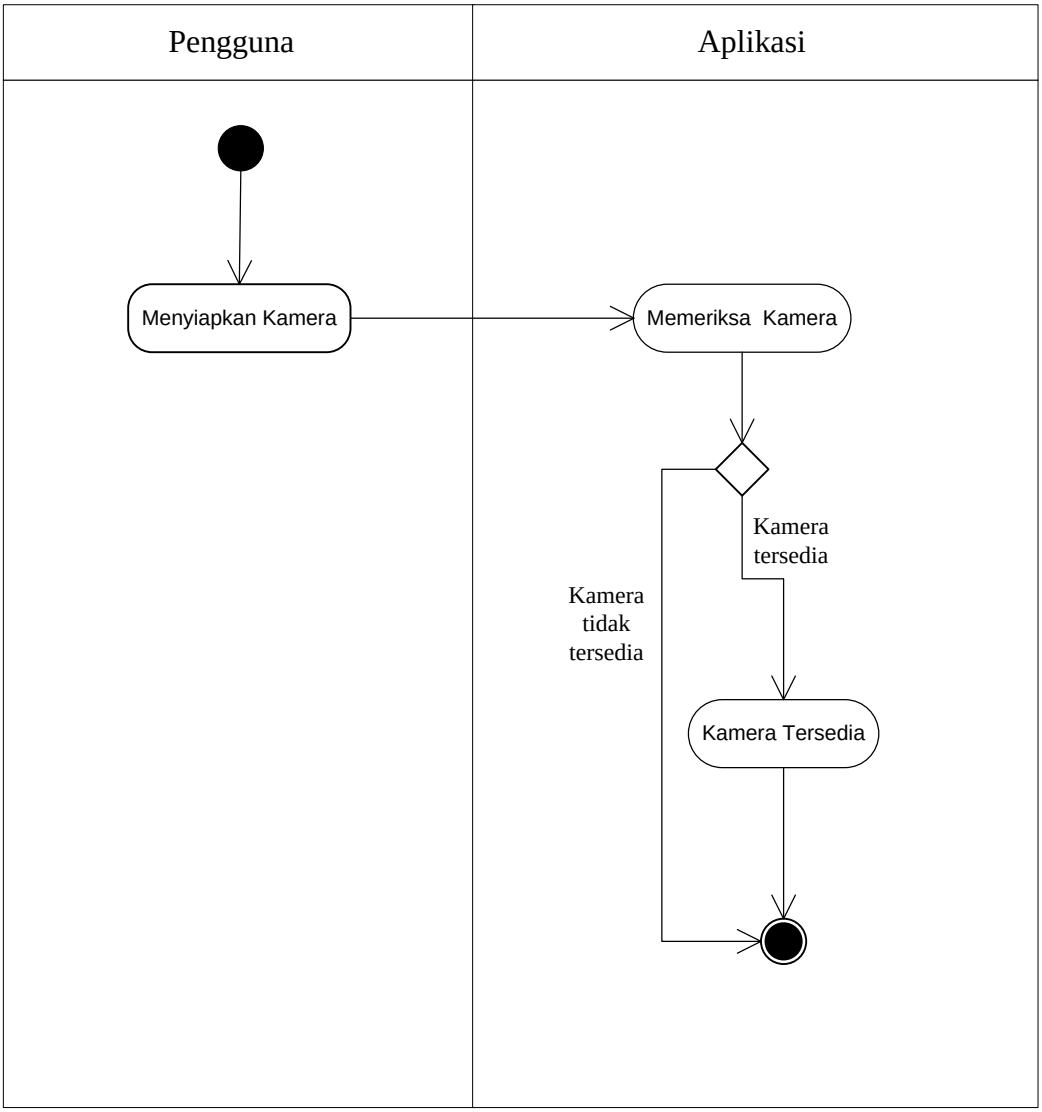
3.1.6.2.1.5 *Activity Diagram* Menggunakan Dekorasi Virtual



Gambar 3.26 *Activity Diagram* Menggunakan Dekorasi Virtual

Gambar 3.16 menggunakan aktivitas yang terjadi pada aplikasi saat pengguna memilih aplikasi Dekorasi Virtual. Dimana pada saat menjalankan aplikasi pengguna harus menunjukan atau marker ke depan *webcam* agar marker tersebut memunculkan objek.

3.1.6.2.1.6 *Activity Diagram* Mendeteksi Kamera



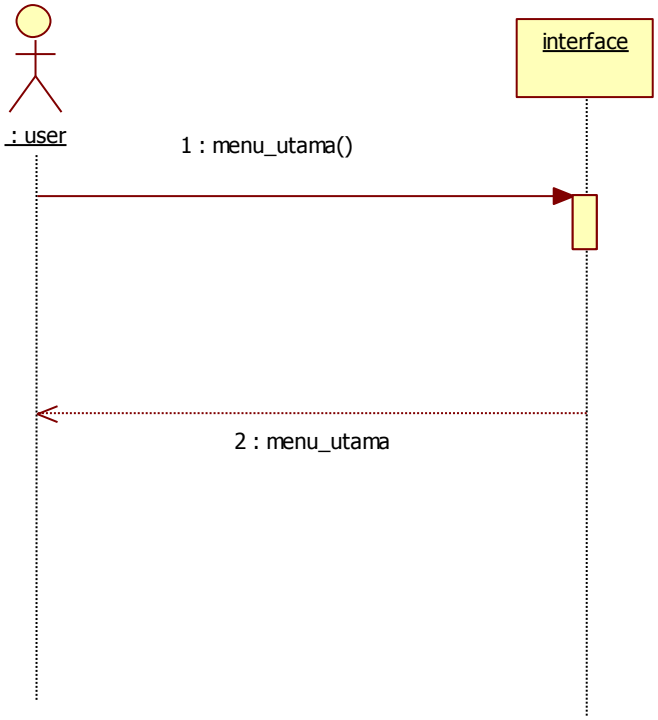
Gambar 3.27 *Activity Diagram* Mendeteksi Kamera

Gambar 3.27, menggambarkan aktivitas yang terjado pada aplikasi saat pengguna mendeteksi kamera. Dimana pada saat menjalankan aplikasi pengguna harus mendeteksi ada atau tidaknya kamera yang terpasang pada komputer atau laptop.

3.1.6.3 *Sequence* Diagram

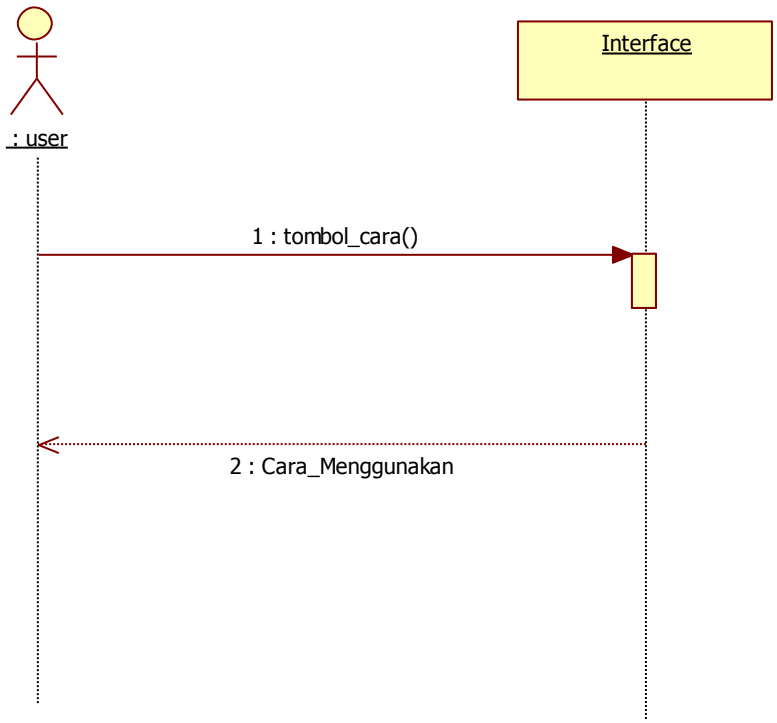
Pada sub bab ini penulis akan menjelaskan sistem dalam bentuk penggambaran *Sequence* Diagram. *Sequence* diagram berbeda dengan diagram sebelum-sebelumnya yang telah dibahas, pada *sequence* diagram dimensi vertikal menjelaskan tentang waktu yang sedang terjadi sedangkan dimensi horizontal menjelaskan objek yang sedang berhubungan.

3.1.6.3.1 *Sequence* Diagram Memilih Menu



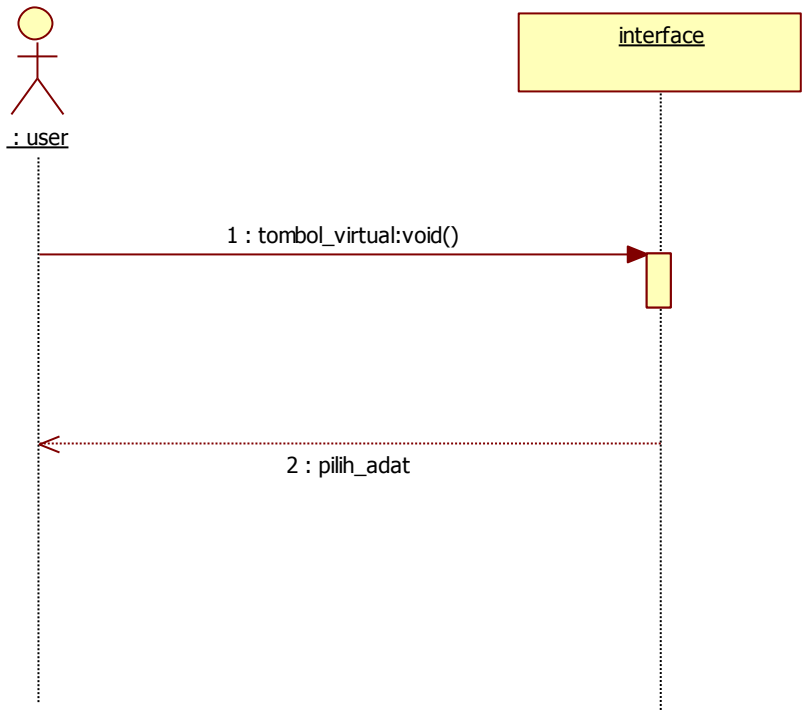
Gambar 3.28 *Sequence* Diagram Memilih Menu

3.1.6.3.2 *Sequence* Diagram Lihat Cara Menggunakan



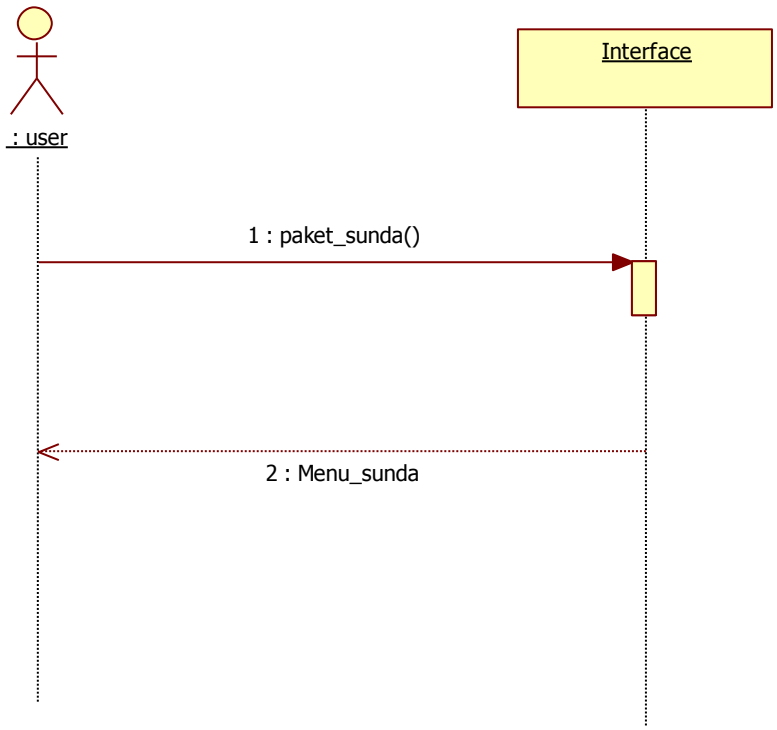
Gambar 3.29 *Sequence* Diagram Lihat Cara Menggunakan

3.1.6.3.3 *Sequence* Diagaram Memilih Dekorasi Virtual



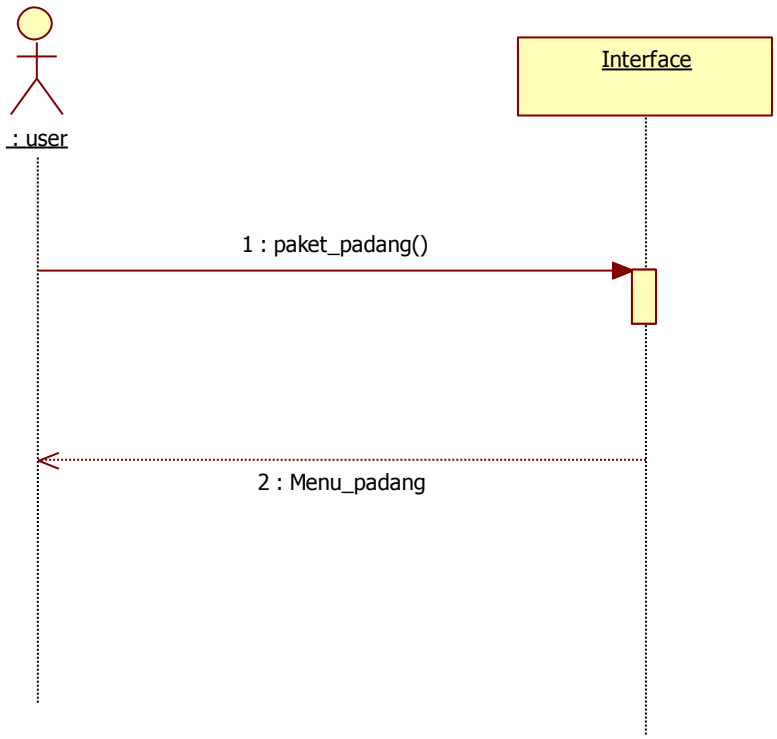
Gambar 3.30 *Sequence Diagram* Memilih Dekorasi Virtual

3.1.6.3.4 *Sequence* Diagram Paket Pernikahan Jabar



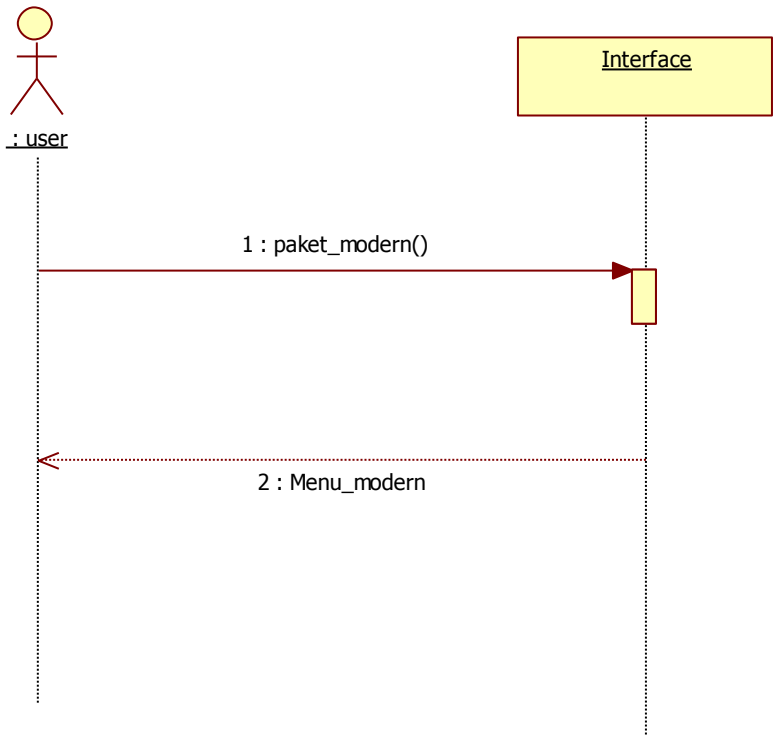
Gambar 3.31 *Sequence* Diagram Paket Pernikahan Jabar

3.1.6.3.5 *Sequence* Diagram Paket Pernikahan Padang



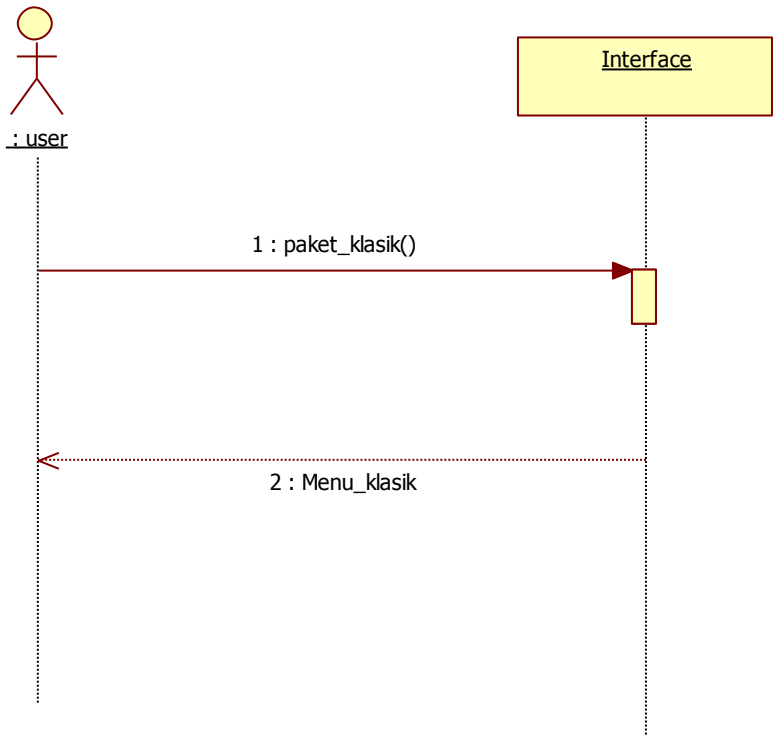
Gambar 3.32 *Sequence* Diagram Paket Pernikahan Padang

3.1.6.3.6 *Sequence* Diagram Paket Pernikahan Modern



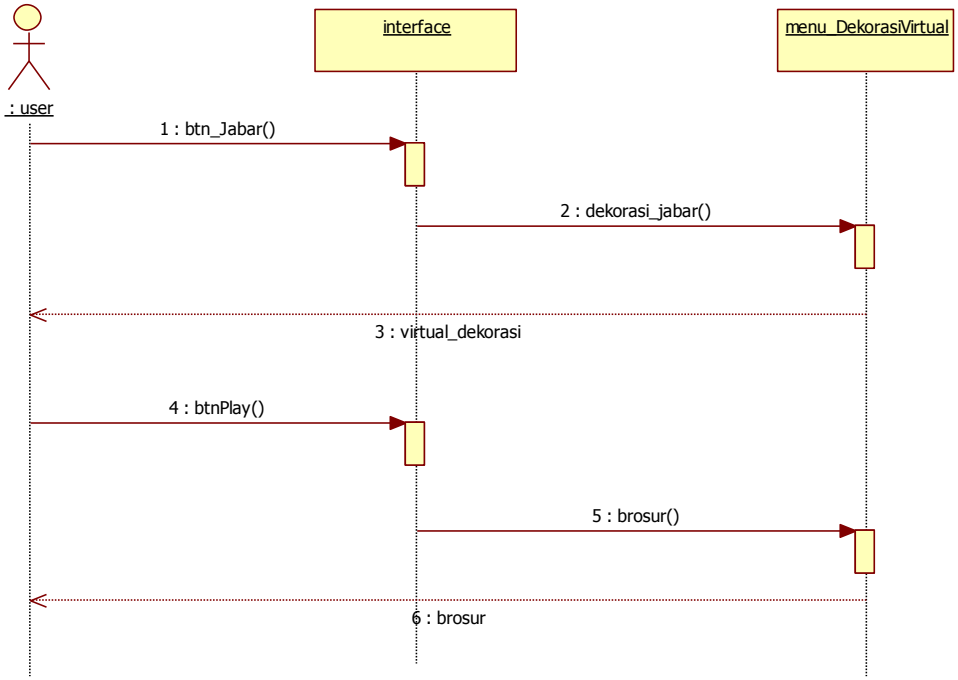
Gambar 3.33 *Sequence* Diagram Lihat Paket Pernikahan Modern

3.1.6.3.7 *Sequence* Diagram Paket Pernikahan Modern Klasik



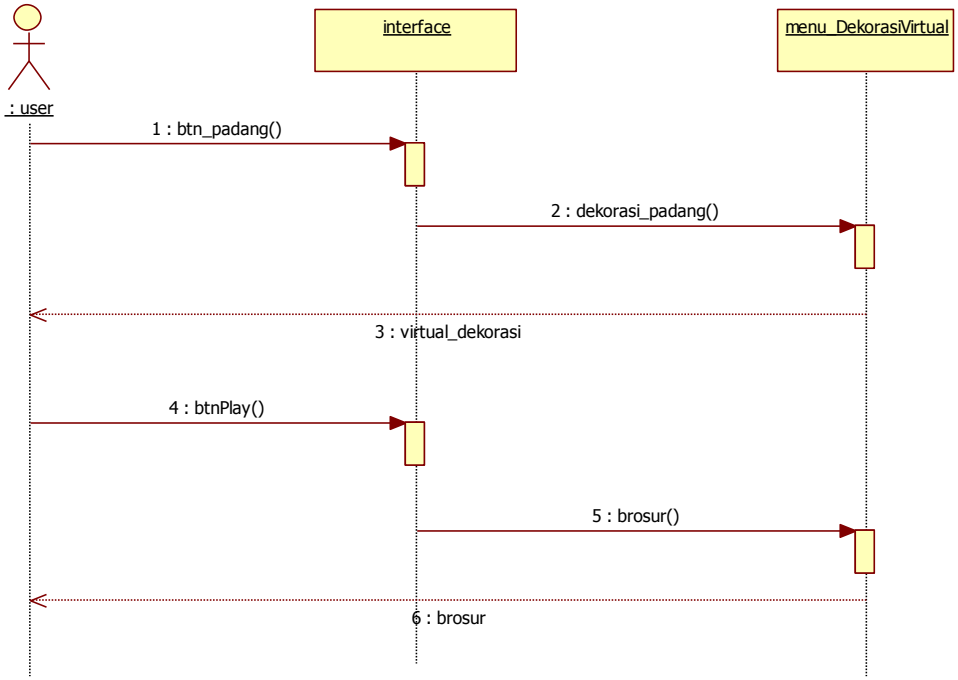
Gambar 3.34 *Sequence* Diagram Paket Pernikahan Modern Klasik

3.1.6.3.8 *Sequence* Diagram Menggunakan Dekorasi Virtual Jabar



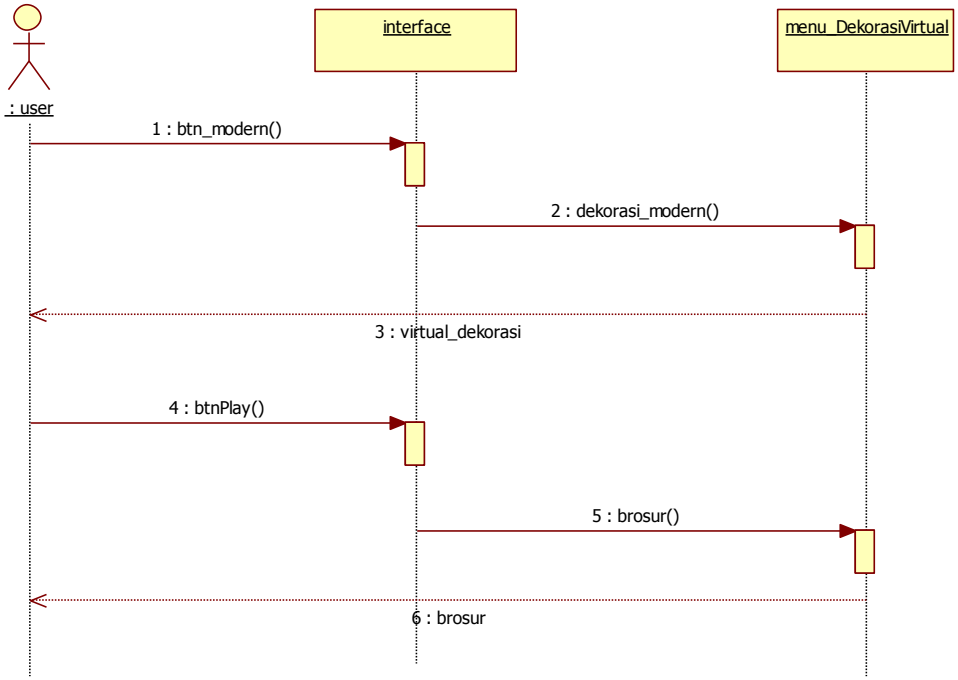
Gambar 3.35 *Sequence* Diagram Menggunakan Dekorasi Virtual Jabar

3.1.6.3.9 *Sequence* Diagram Menggunakan Dekorasi Virtual Padang



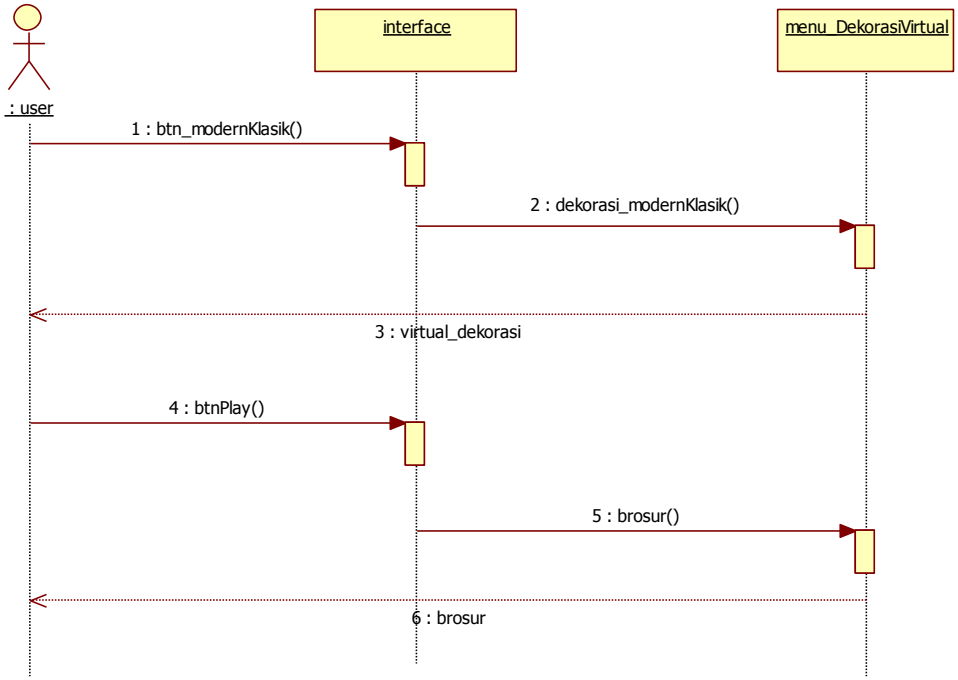
Gambar 3.36 *Sequence* Diagram Menggunakan Dekorasi Virtual Padang

3.1.6.3.10 *Sequence* Diagram Menggunakan Dekorasi Virtual Modern



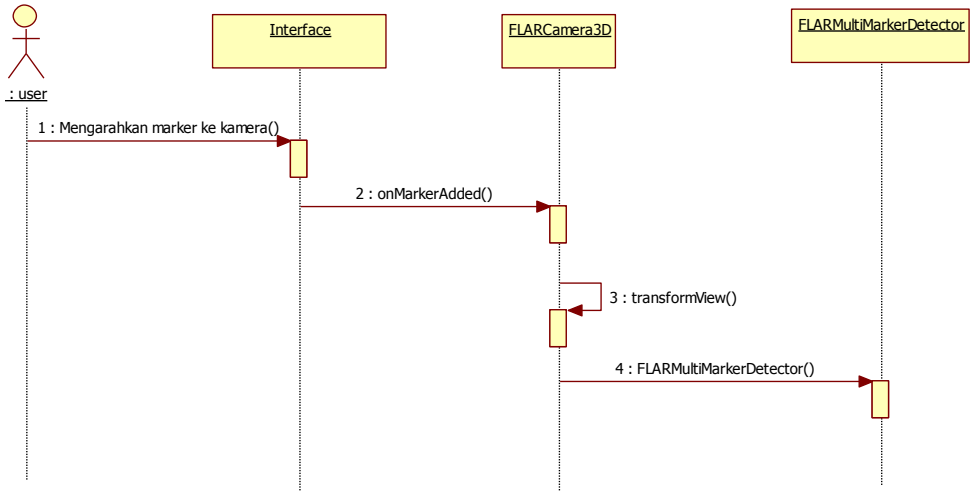
Gambar 3.37 *Sequence* Diagram Menggunakan Dekorasi Virtual Modern

3.1.6.3.11 *Sequence* Diagram Menggunakan Dekorasi Virtual Modern Klasik



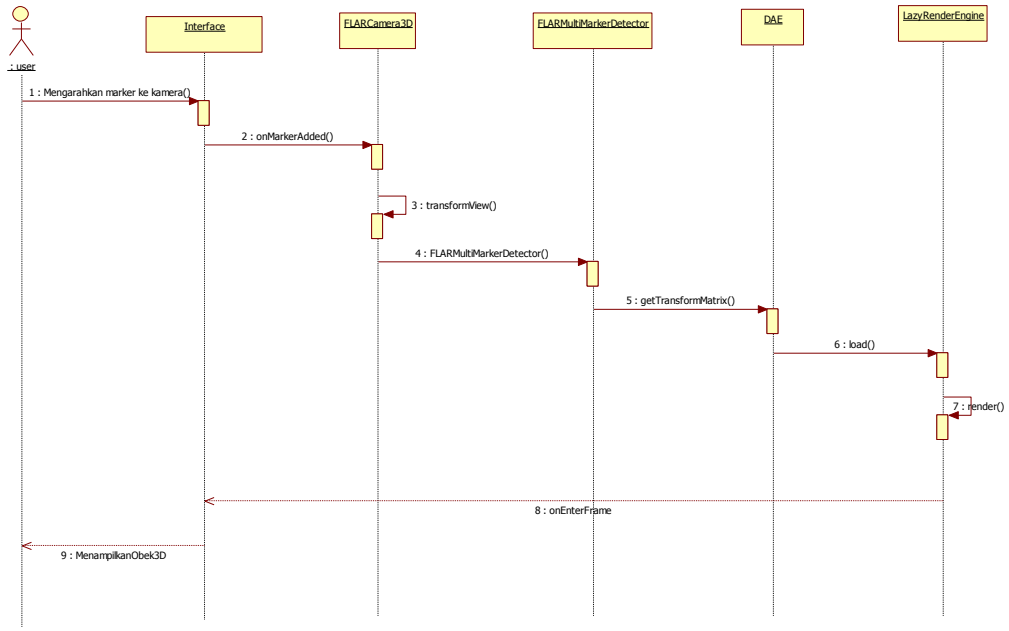
Gambar 3.38 *Sequence* Diagram Menggunakan Dekorasi Virtual Modern Klasik

3.1.6.3.12 *Sequence* Diagram Tracking *Marker*



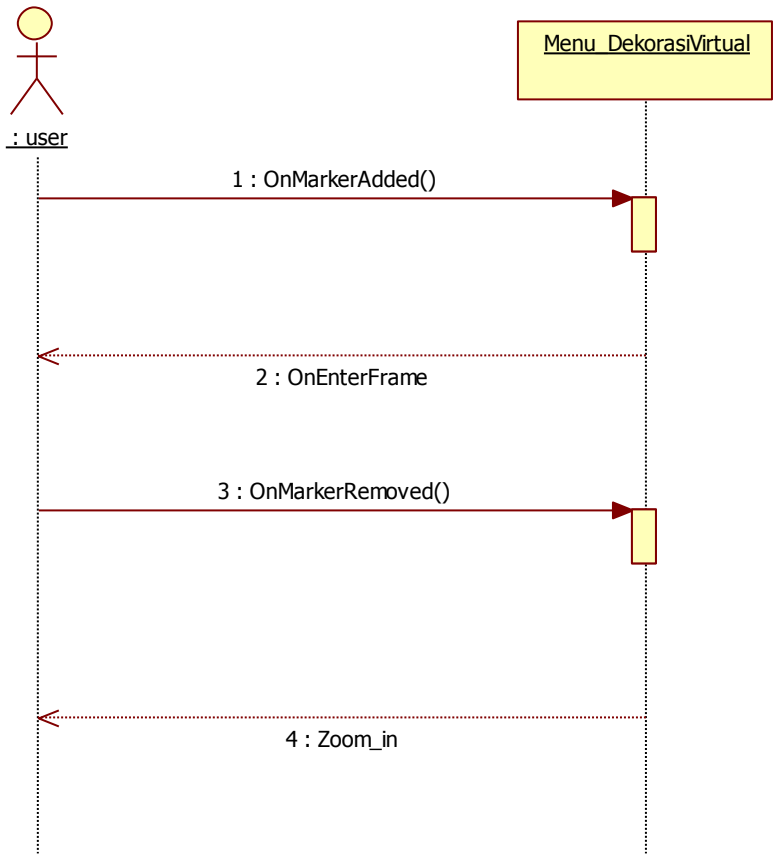
Gambar 3.39 *Sequence* Diagram *Tracking Marker*

3.1.6.3.13 *Sequence* Diagram Render Objek



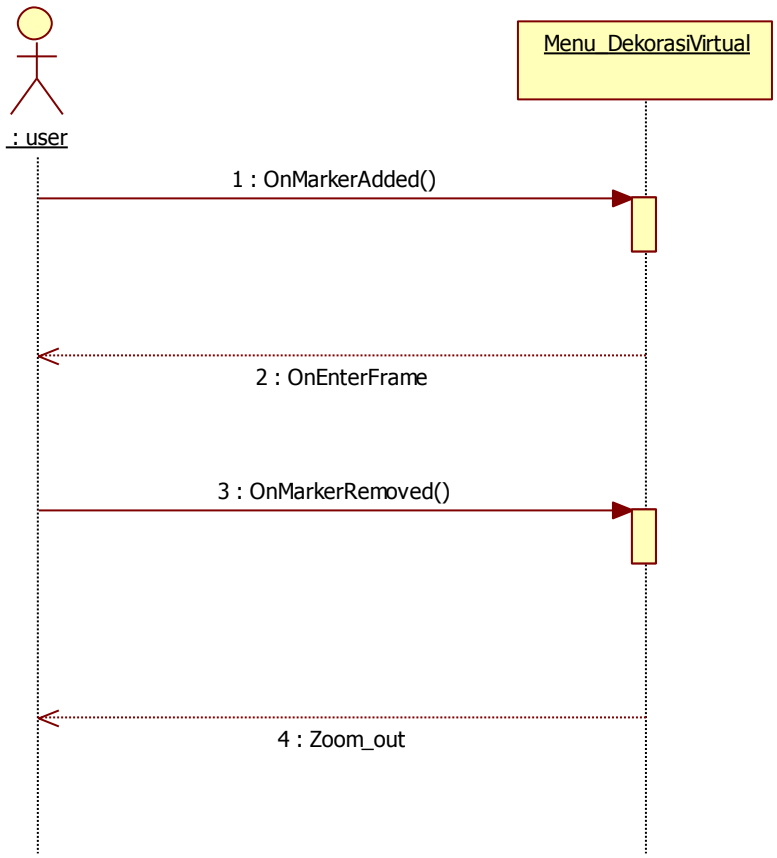
Gambar 3.40 *Sequence* Diagram Render Objek

3.1.6.3.14 *Sequence* Diagram Zoom in



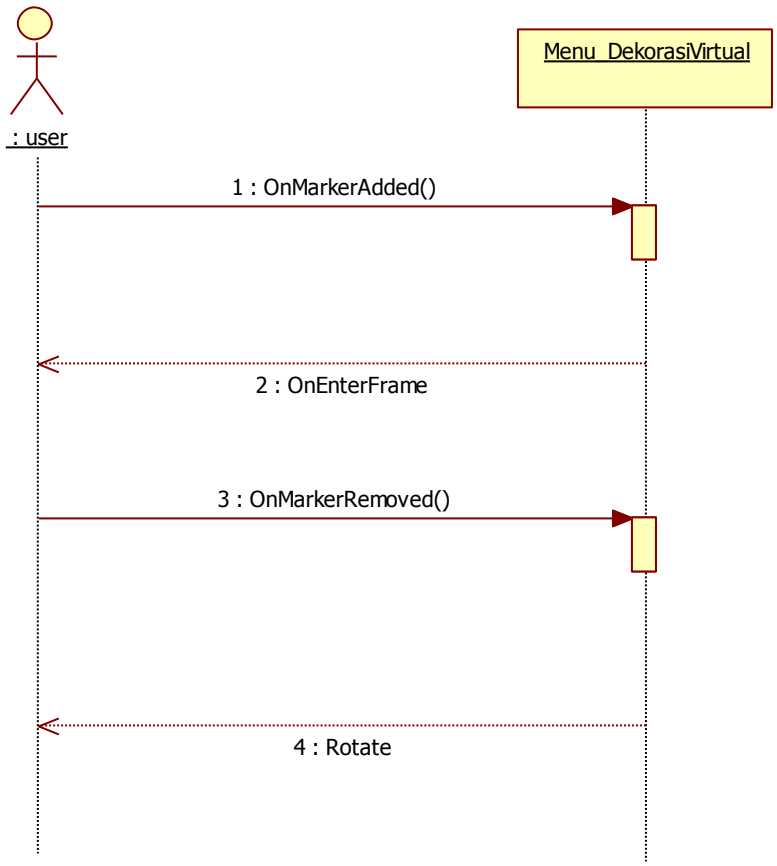
Gambar 3.41 *Sequence* Diagram Zoom in

3.1.6.3.15 *Sequence* Diagram Zoom out



Gambar 3.42 *Sequence* Diagram Zoom out

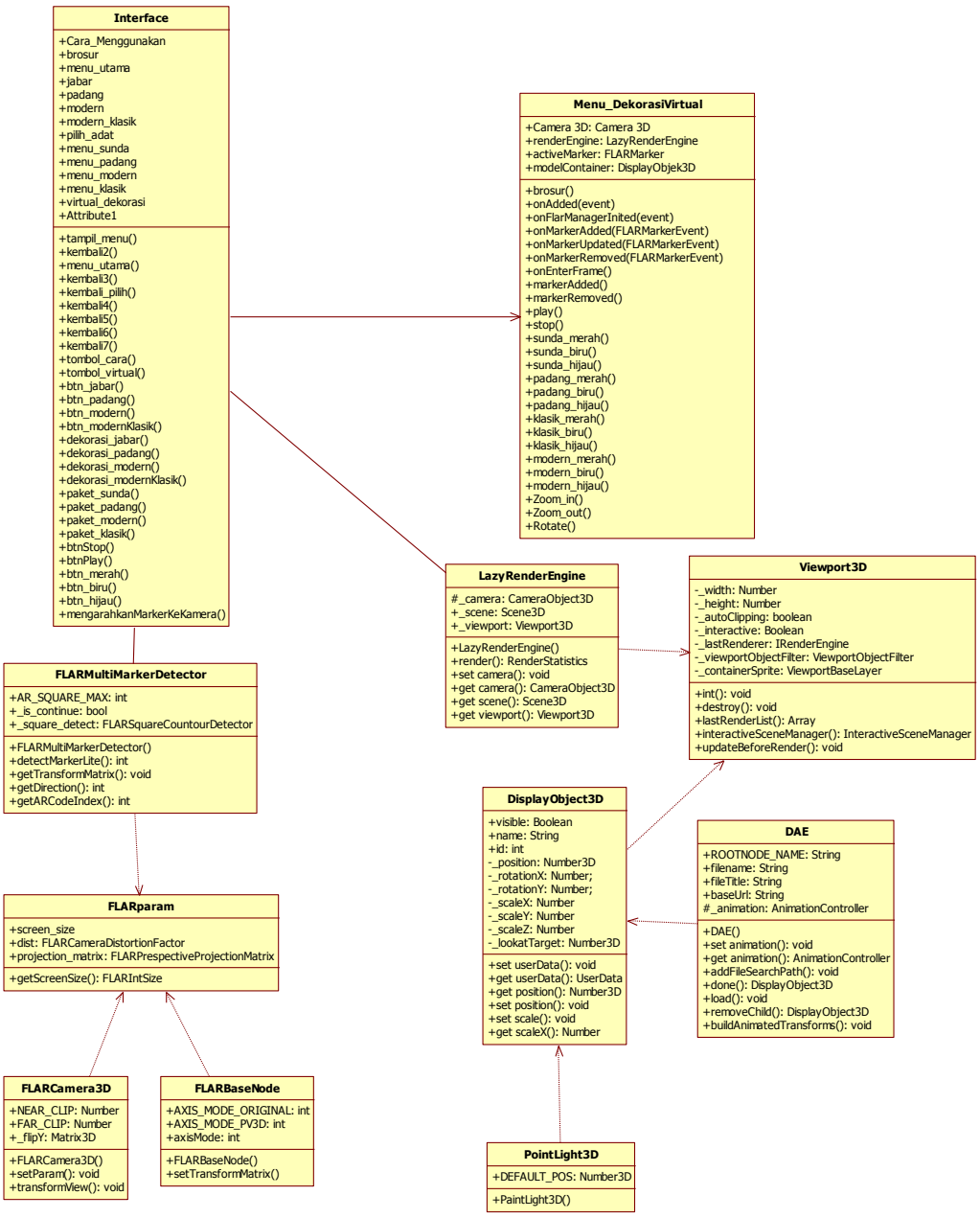
3.1.6.3.16 *Sequence* Diagram Rotate



Gambar 3.43 *Sequence* Diagram Rotate

3.1.6.4 *Class* Diagram

Class Diagram adalah diagram UML *(Unified Modeling Language)* yang menggambarkan kelas-kelas yang berhubungan dengan sistem antara satu dengan yang lain yang berisi dan operasi. Untuk lebih jelasnya dapat dilihat**.**



Gambar 3.44 *Class Diagram*

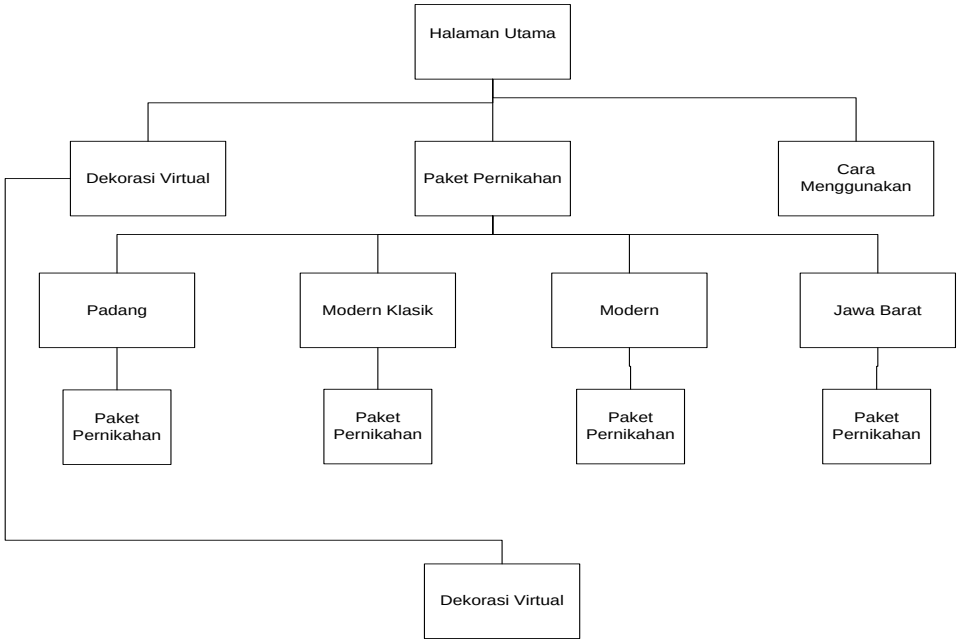
3.2 Perancangan Sistem

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 *Marker*
4. Jaringan Semantik
5. Perancangan *Method*

3.2.1 Perancangan Struktur Menu

Struktur *menu* adalah bentuk umum dari suatu racangan aplikasi untuk memudahkan pengguna dalam menjalankan aplikasi komputer. Sehingga saat menjalankan aplikasi, pengguna tidak mengalami kesulitan dalam memilih *menu-menu* yang diinginkan. Berikut ini perancangan struktur *menu* pada aplikasi brosur AR di CV. Deris *Package* yang ditunjukan pada gambar berikut ini :

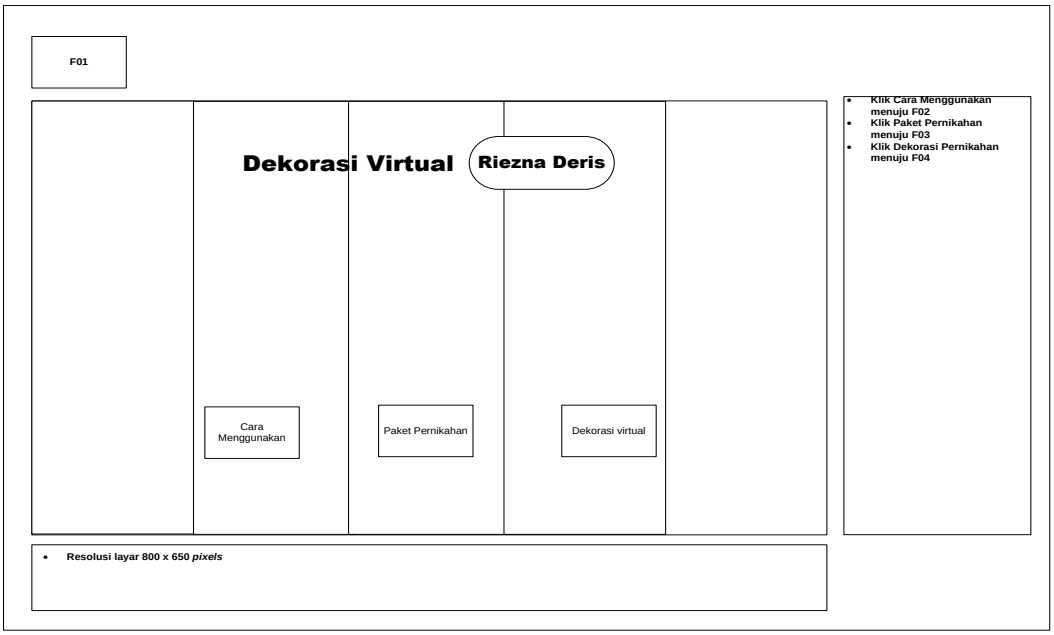


Gambar 3.45 Struktur Menu

3.2.2 Perancangan Antarmuka

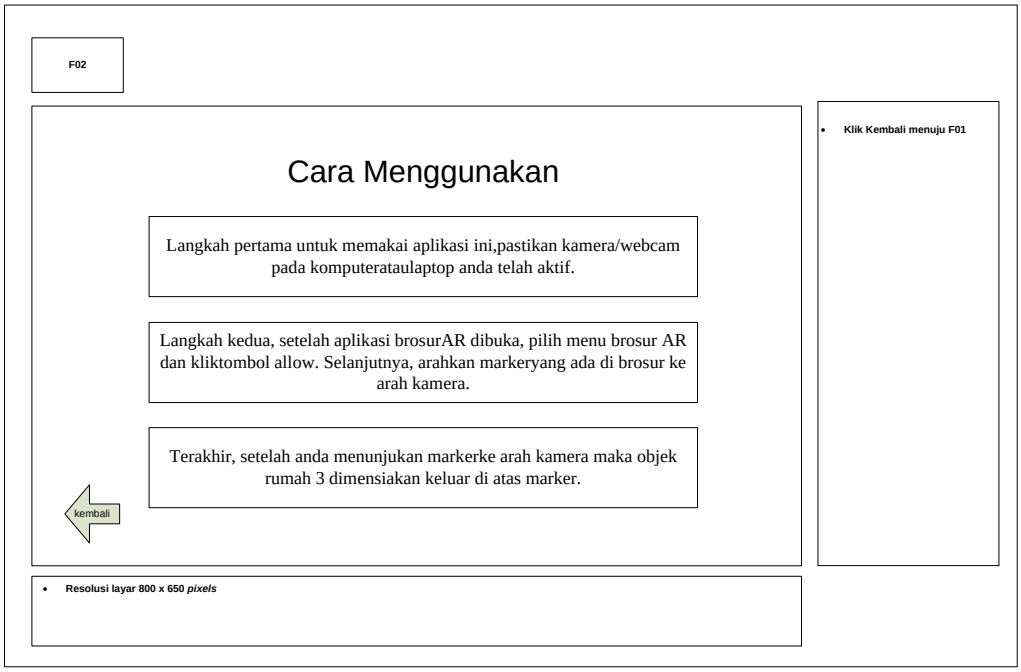
Perancangan antarmuka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun. Berikut ini contoh perancangan antarmuka yang ada dari aplikasi yang akan dibangun yaitu :

1. **Perancangan Antarmuka Halaman Utama**



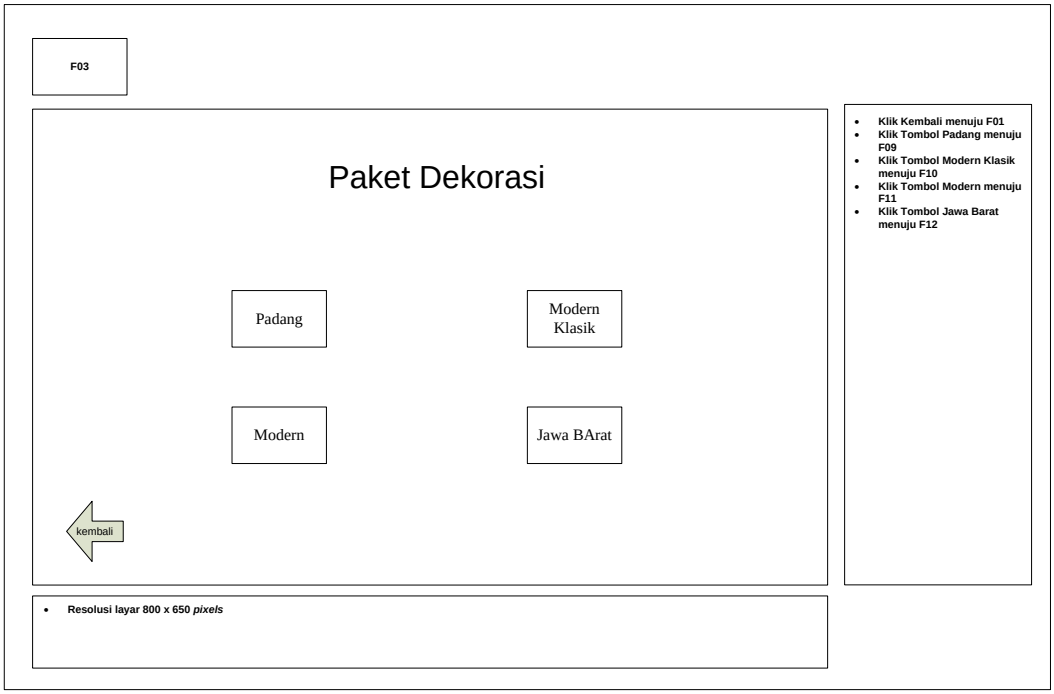
Gambar 3.46 Antarmuka Halaman Utama

2. Perancangan Antarmuka Menu Cara Menggunakan



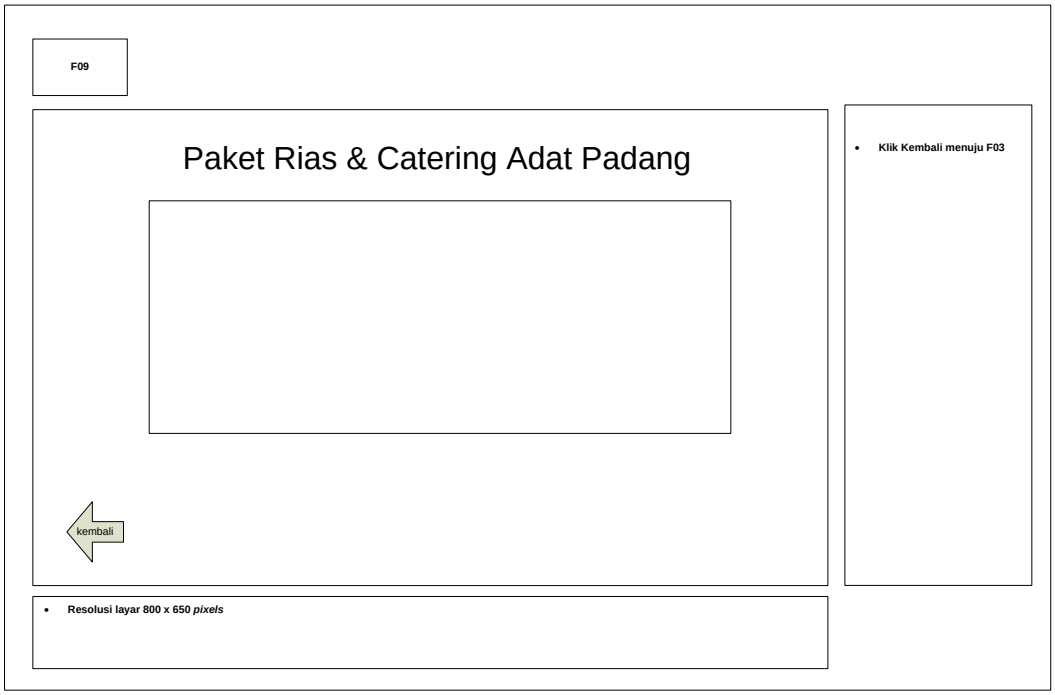
Gambar 3.47 Antarmuka Menu Cara Menggunakan

3. Perancangan Antarmuka Paket Dekorasi



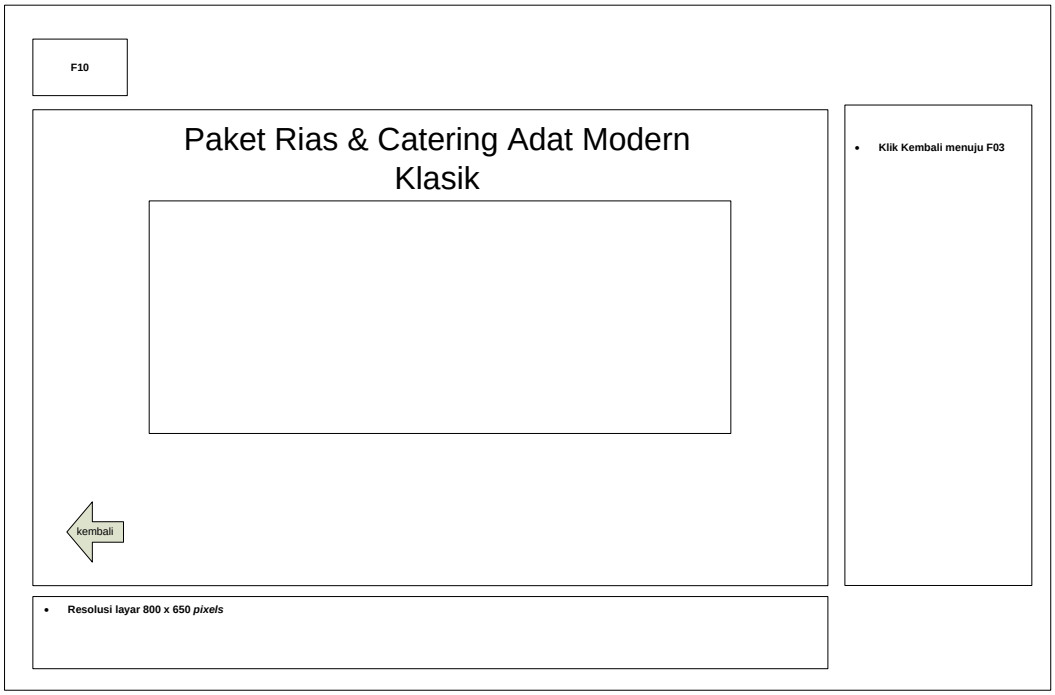
Gambar 3.48 Antarmuka Menu Paket Dekorasi

4. Perancangan Antarmuka Paket Rias dan Catering Adat Padang



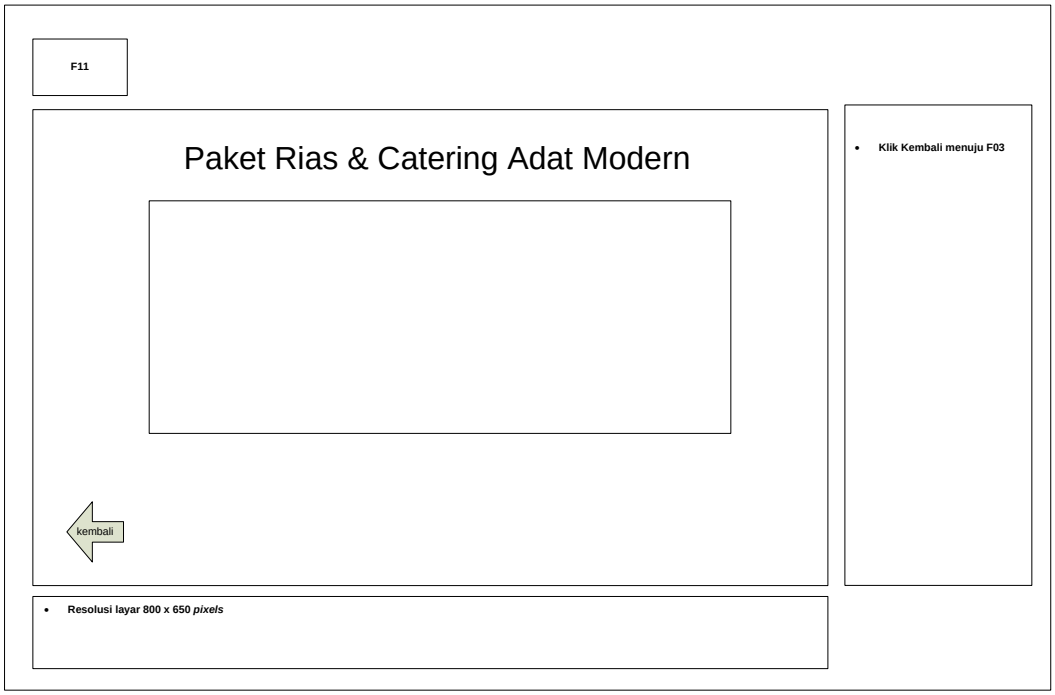
Gambar 3.49 Antarmuka Menu Paket Pernikahan Adat Padang

5. Perancangan Antarmuka Paket Rias dan Catering Adat Modern Klasik



Gambar 3.50 Antarmuka Menu Paket Pernikahan Adat Modern Klasik

6. Perancangan Antarmuka Paket Rias dan Catering Adat Modern



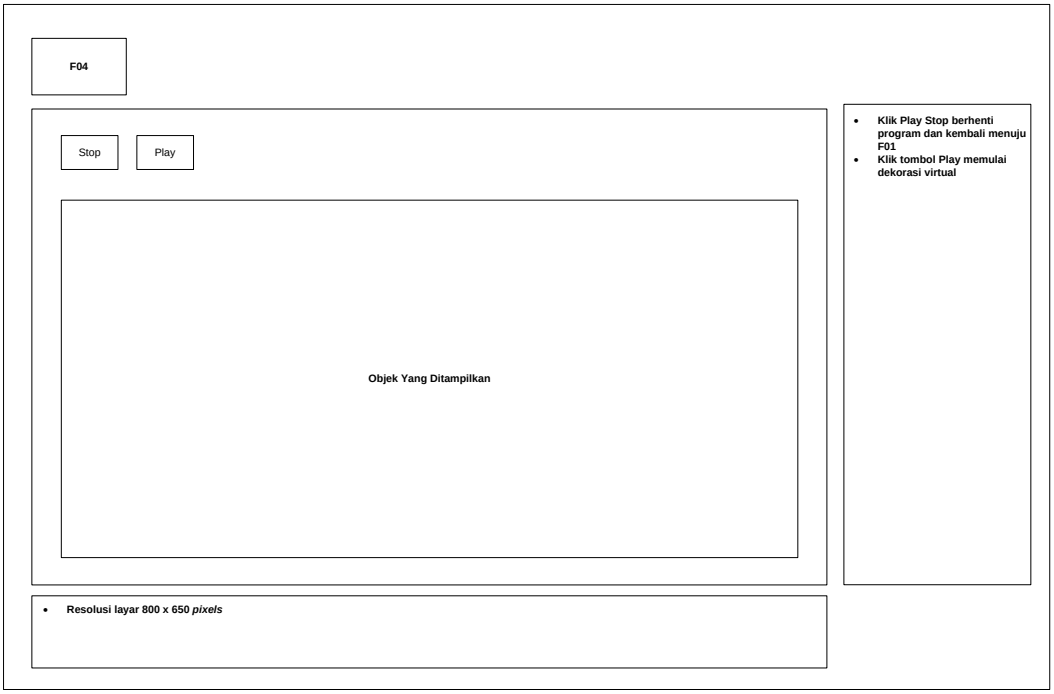
Gambar 3.51 Antarmuka Menu Paket Pernikahan Adat Modern

7. Perancangan Antarmuka Paket Rias dan Catering Adat Jawa Barat



Gambar 3.52 Antarmuka Menu Paket Pernikahan Adat Jawa Barat

8. Perancangan Antarmuka Dekorasi Virtual



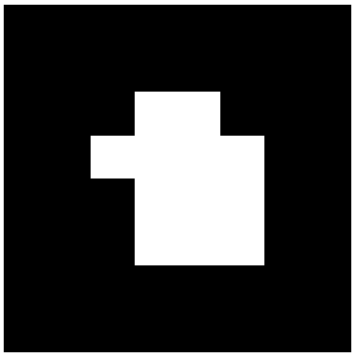
Gambar 3.53 Antarmuka Dekorasi Virtual

3.2.3 Perancangan *Marker*

Marker adalah sebuah pola yang berbentuk kotak hitam putih yang mana akan dikenali oleh kamera yang bersifat real time. *Marker* dapat dibuat sesuai dengan desain atau model yang diinginkan. Untuk membuat *marker*, ada beberapa langkah yang harus dilakukan, diantaranya adalah sebagai berikut :

1. Mempersiapkan Pola *Marker*

Pada tahap ini dimulai dengan membuat pola marker sesuai dengan kebutuhan. Pola Marker ini dapat dibuat dengan menggunaan aplikasi Adobe Photosho. Berikut adalah beberapa pola marker yang telah dibuat :



Gambar 3.54 Pola *Marker*

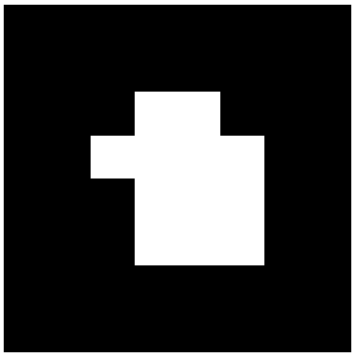
2. Regristrasi *Marker*

Regristrasi *marker* dapat dilakukan dengan menggunakan aplikasi *Marker's Generator Online*



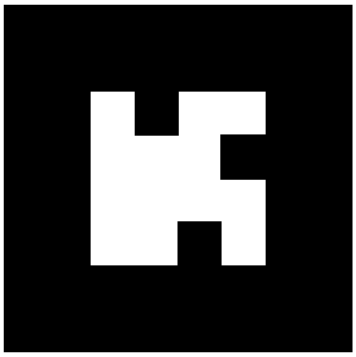
Gambar 3.55 Tampilan *Marker's Generator Online*

Pada pembangunan aplikasi ini penulis membuat beberapa marker. Setiap pattern mewakili 1 *object* sebagai identitas *object* tersebut. Berikut adalah marker yang digunakan dalam pembangunan aplikasi ini :



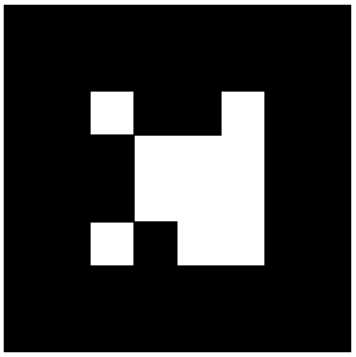
Gambar 3.56 *Marker* 1

Marker 1 Digunakan sebagai identitas objek dekorasi adat padang. Objek dekorasi adat padang akan muncul ketika *webcam* mendeteksi *marker* 1.



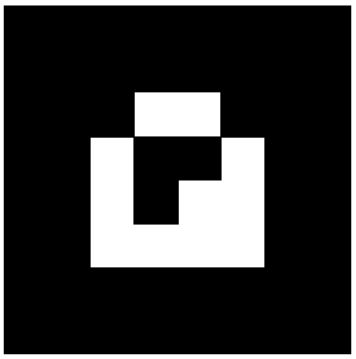
Gambar 3.57 *Marker* 2

Marker 2 Digunakan sebagai identitas objek dekorasi Modern Klasik. Objek dekorasi adat modern klasik akan muncul ketika *webcam* mendeteksi *marker* 2.



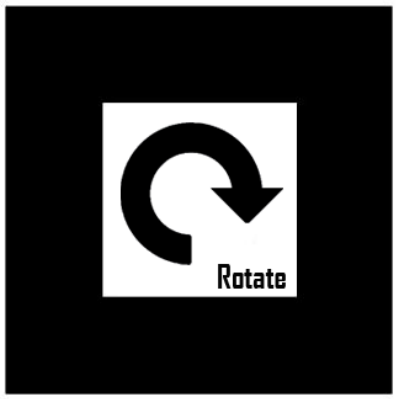
Gambar 3.58 *Marker* 3

Marker 3 Digunakan sebagai identitas objek dekorasi Modern. Objek dekorasi adat modern akan muncul ketika *webcam* mendeteksi *marker* 3.



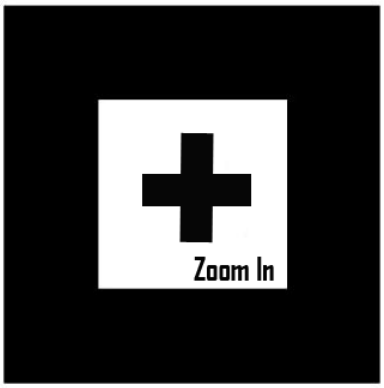
Gambar 3.59 *Marker* 4

Marker 4 Digunakan sebagai identitas objek dekorasi adat Sunda. Objek dekorasi adat sunda akan muncul ketika *webcam* mendeteksi *marker* 4.



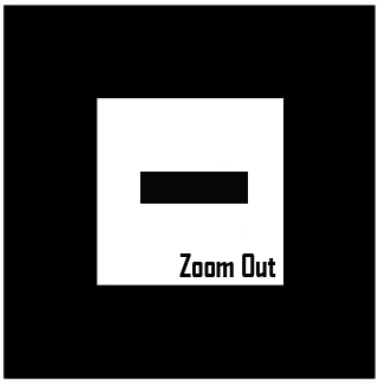
Gambar 3.60 *Marker* 5

Marker 5 digunakan untuk melakukan rotasi terhadap objek adat dekorasi yang sedang terdeteksi.



Gambar 3.61 *Marker* 6

Marker 6 digunakan untuk melakukan *Zoom In* terhadap objek adat dekorasi yang sedang terdeteksi.



Gambar 3.62 *Marker* 7

Marker 7 digunakan untuk melakukan *Zoom Out* terhadap objek adat dekorasi yang sedang terdeteksi.



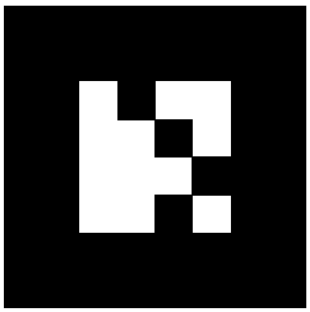
Gambar 3.63 *Marker* 8

Marker 8 digunakan untuk menampilkan objek adat dekorasi dengan tata letak yang berbeda terhadap objek yang sedang terdeteksi.



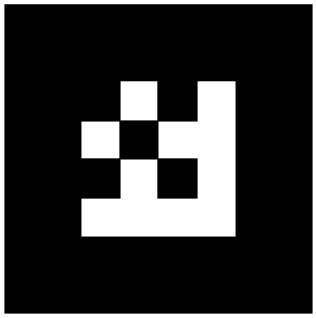
Gambar 3.64 *Marker* 9

Marker 9 digunakan untuk menampilkan objek adat dekorasi dengan tata letak yang berbeda terhadap objek yang sedang terdeteksi.



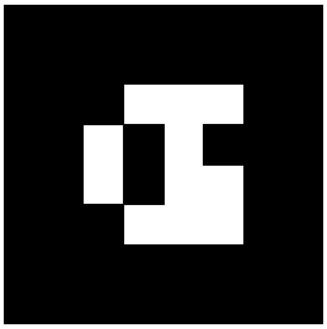
Gambar 3.65 *Marker* 10

Marker 10 digunakan untuk menggantikan warna objek dekorasi menjadi warna merah.



Gambar 3.66 *Marker* 11

Marker 11 digunakan untuk menggantikan warna objek dekorasi menjadi warna biru.

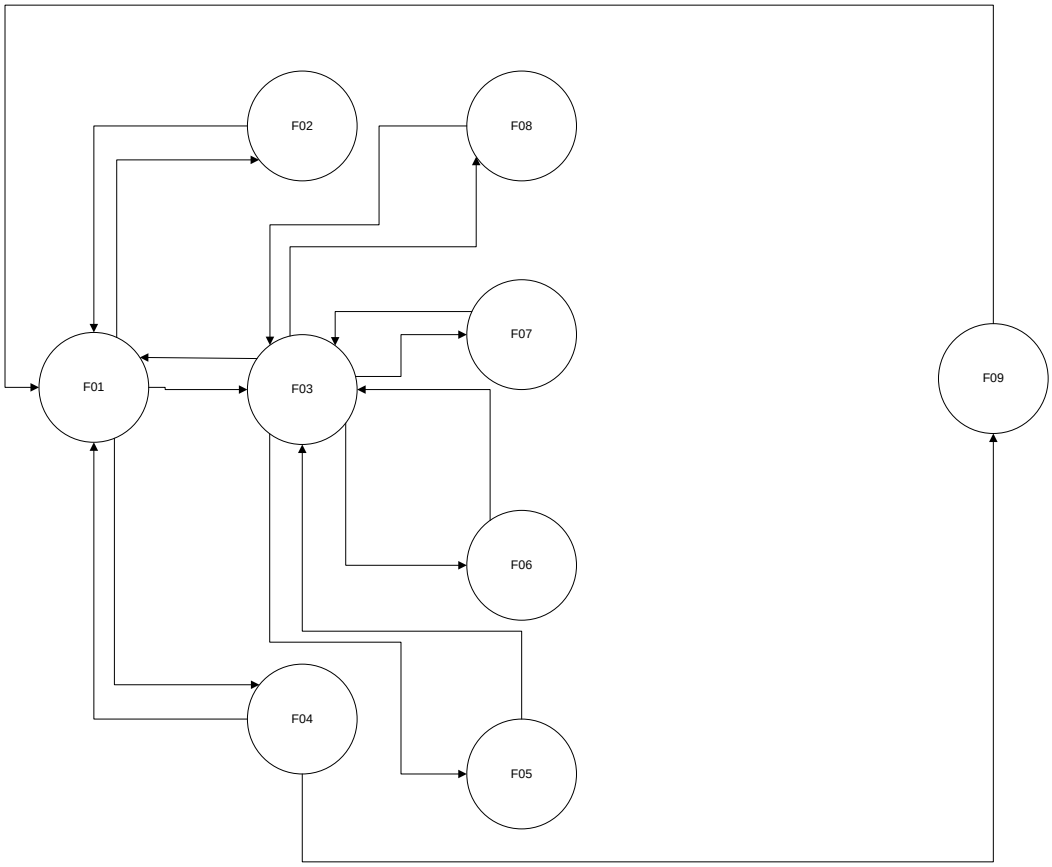


Gambar 3.67 *Marker* 12

Marker 12 digunakan untuk menggantikan warna objek dekorasi menjadi warna hijau.

3.2.4 Jaringan Semantik

Jaringan semantik ini menunjukan form-form yang bisa diakses oleh pengguna seperti ditunjukan pada gambar 3.



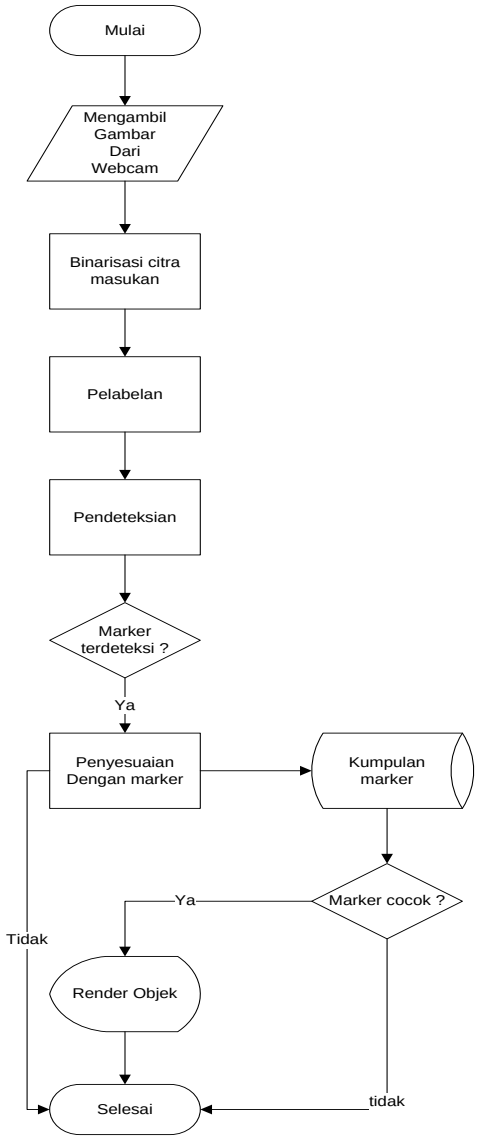
Gambar 3.68 Jaringan Semantik

Keterangan :

F01 : Halaman Utama
F02 : Menu Cara Menggunakan
F03 : Menu Paket Perniahan
F04 : Menu Dekorasi Virtual
F05 : Dekorasi Adat Modern Klasik
F06 : Dekorasi Adat Modern
F07 : Dekorasi Adat Jawa Barat
F08 : Dekorasi Adat Padang
F09 : Dekorasi virtual

3.2.5 Perancangan *Method*

Perancangan *method* merupakan perancangan yang berfungsi untuk mendeskripsikan *method* yang berada di dalam aplikasi brosur menggunakan teknologi *augmented reality*. Adapun *method* yang digunakan dalam aplikasi yang akan dibangun adalah sebagai berikut :



Gambar 3.69 Perancangan *Methond Augmented Reality*