BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem informasi 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.

3.1.1 Analisis Masalah

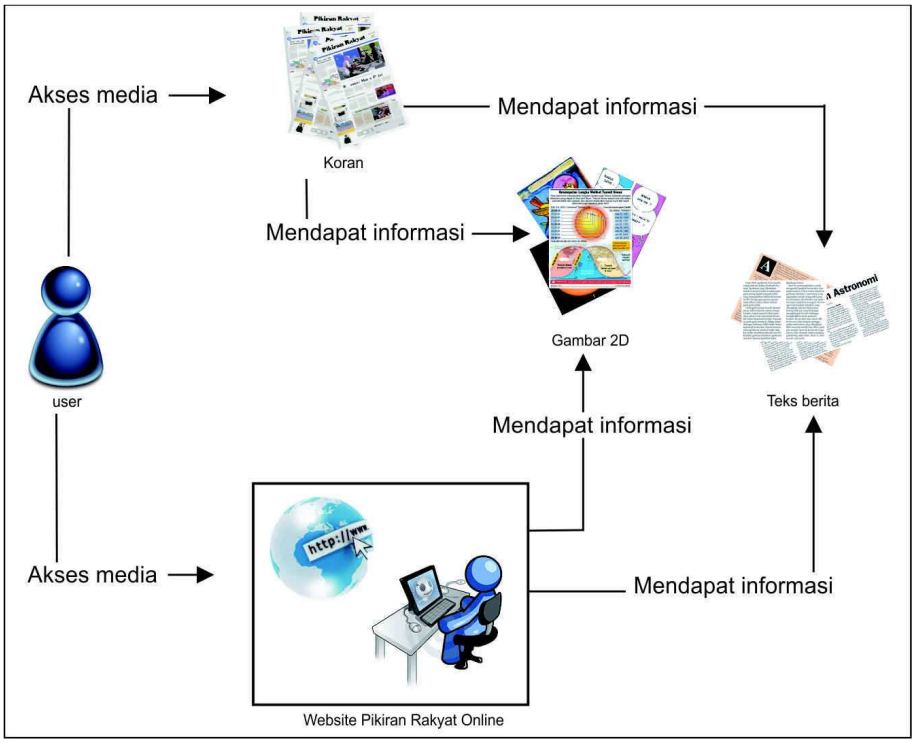
Semakin berkembangnya teknologi informasi, terutama internet sebagai sarana pemenuhan kebutuhan informasi bagi masyarakat dan penyampaian informasi melalui koran berupa teks dan gambar 2D, menjadi masalah yang dihadapi PT. Pikiran Rakyat Bandung dalam beberapa tahun terakhir ini. Hal itu juga menjadi alasan bagi para pembaca untuk tidak memilih koran sebagai tempat pencari informasi dan lebih memilih media online atau menonton televisi.

Analisis perubahan media habit dalam mengkonsumsi media di era teknologi informasi terjadi penurunan eksistensi koran terhadap pembaca dengan perkembangan teknologi informasi pada saat ini [Barina Indah Puspaningrum FE UI 2010].

Keunggulan dari teknologi *augmented reality* dalam segi menampilkan suatu informasi dapat diterapkan pada koran untuk dapat memberikan solusi dari permasalahan yang ada. Dengan demikian, tujuan dari pembangunan aplikasi yaitu meningkatkan minat membaca koran kepada masyarakat dengan memberikan fasilitas tambahan teknologi *augmented reality* berupa penerapan metode *markerless* pada gambar dikoran dan membuat aplikasi berbasis web pada Pikiran Rakyat *Online* untuk menjalankan aplikasi tersebut. Dengan menerapkan teknologi ini, penyampaian informasi yang didapat akan lebih menarik dan interaktif yaitu berupa video atau animasi 3D. Sistem yang dibuat merupakan program untuk mendeteksi sebuah *markerless* sekaligus menapilkan objek tiga dimensi. Objek yang di buat berupa video bergerak atau animasi dan menerapkan sebagian dari objek dengan metode *markerless* pada gambar dikoran, agar pembaca lebih berinteraksi dengan koran dan memaksimalkan pemberitaan tersebut.

3.1.2 Analisis Sistem Yang Berjalan

Analisis Sistem atau analisis proses adalah tahapan yang memberi gambaran tentang sistem yang sedang berjalan sekarang. Analisis ini bertujuan untuk memberi gambaran yang lebih detail bagaimana cara kerja dari sistem yang sedang berjalan. Prosedur pada proses media yang sedang berjalan sekarang adalah sebagai berikut:



Gambar 3. 1 Alur sistem yang sedang berjalan

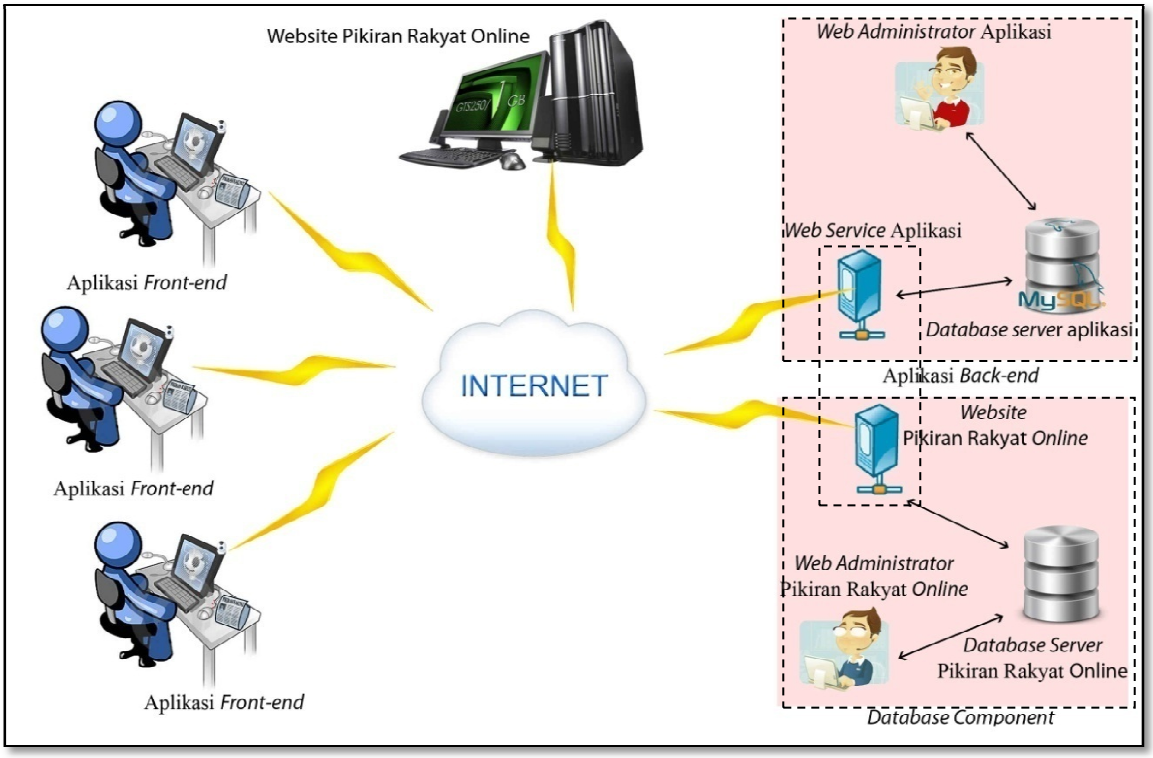
1. User disini adalah pembaca koran atau *user frontend web* pikiran rakyat *online* yang ingin mengetahui informasi berita apa saja yang ada pada koran dan *web* pikiran rakyat *online*.
2. Terdapat 2 media yang tersedia untuk mengetahui informasi yang akan didapat, diantaranya :
    1) Koran : Media ini cukup mudah untuk mengetahui informasi berita tersebut, penyampaian informasi pada koran berupa teks dan gambar 2D.
    2) *Web* pikiran rakyat *online* : Media ini terkoneksi dengan internet,

dan untuk mengakses web sendiri diperlukan perangkat hardware. Pada *web* pikiran rakyat *online* terdapat informasi yang terbaru dan e-*paper* yang sesuai dengan koran cetak.

Dari gambaran prosedur media yang didapat untuk mengakses informasi pada koran dan *web* pikiran rakyat *online,* teknologi *augmented reality* bisa dijadikan sebagai media alternatif sehingga konten berita menjadi lebih menarik dan interaktif.

3.1.3 Analisis Arsitektur Sistem

Arsitektur fisik sistem yang dibangun menggunakan model *tree tier.* Pada arsitektur ini terdiri dari komponen aplikasi *ARNewspaper (front-end Component), web admin* aplikasi, *web service* dan *database server* aplikasi (*back-end Component)* dan *web admin* pikiran rakyat *online*, *web service* pikiran rakyat *online* dan *database server* pikiran rakyat *online (database Component).* Arsitektur sistem dalam pembangunan aplikasi ini ditunjukan pada gambar 3.2



Gambar 3. 2 Arsitektur Sistem

Front-end adalah aplikasi koran AR yang dijalankan melalui komputer atau *notebook* dengan fasilitas *webcam*. Aplikasi ini berfungsi sebagai antarmuka antara pembaca atau *user frontend* dengan sistem.

Database component, Pada bagian ini berisi informasi data yang menampilkan antarmuka pengguna dengan sistem, tapi *end-user* tidak dapat melihat atau berhubungan langsung dengan database. Fungsi utama komponen *database* adalah menyimpan data Pikiran Rakyat *Online* dan melakukan *cross server* untuk mengambil data pada *server* aplikasi.

Back-end adalah aplikasi pendukung yang berfungsi sebagai penyuplai atau sumber data pada aplikasi *frontend*. Aplikasi ini terdiri dari tiga subbagian utama yaitu *web administrator* aplikasi, *web service* aplikasi, dan *database server* aplikasi. *Web administrator* aplikasi

merupakan web yang berfungsi sebagai pengolah sumber basis data yang terdapat pada database server yang dikelola oleh *user backend*. *Web service* aplikasi merupakan web yang berfungsi sebagai media pertukaran data antar *database server* dan aplikasi. *Database server* aplikasi merupakan aplikasi yang berfungsi sebagai sumber data (*data resources*) aplikasi.

3.1.4 Analisis Kebutuhan AR Toolkit/Library

Toolkit dan library untuk AR telah banyak dikembangkan. Perbandingan beberapa toolkit dan library untuk AR dapat dilihat pada Tabel 3.1

Tabel 3. 1 Perbandingan toolkit dan library untuk AR

Parameter		AR Toolkit/Library			
		AR Toolkit	**Flartoolkit**	**D'Fusion Studio**	**In2AR**
1.	Lisensi	Gratis	Gratis	Gratis	Gratis
2.	Dukungan bahasa pemrograman	C	Action script	LUA scripting	Action Script
3.	Dukungan marker	Berbingkai hitam	Berbingkai hitam	Hampir semua jenis gambar	Hampir semua jenis gambar
4.	Authoring/non authoring	Non authoring	Non authoring	Authoring	Non authoring
5.	Format model 3 dimensi yang dapat ditampilkan	wrl	.dae, .obj, .3ds	scenetta	.dae, .obj, .3ds
6.	Dukungan platform	Multiplatform	Multiplatform	Multiplatform	Multiplatform
7	Distribusi kepada pengguna	Harus install ARToolkit di komputer pengguna	.swf	Menggunakan DFusion @HomeDesktopPlayer	.swf

Kelebihan IN2AR jika dibandingkan dengan toolkit atau library lain adalah:

1. Selain menyediakan lisensi berbayar (2212,50 EURO per proyek), juga terdapat lisensi gratis. Perbedaan kedua lisensi tersebut adalah

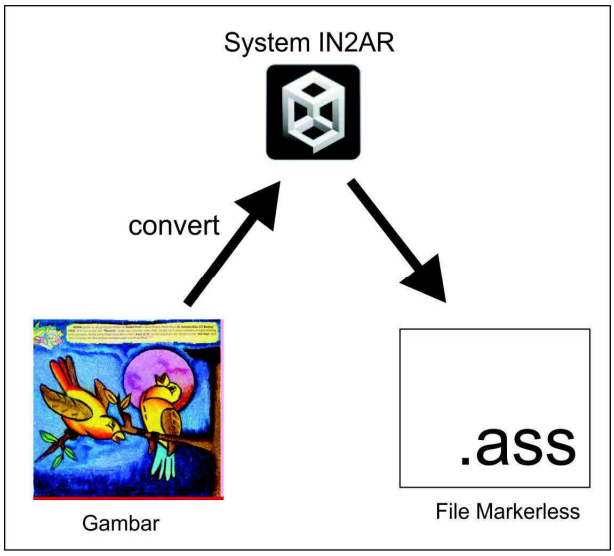
bahwa aplikasi AR yang dibuat dengan lisensi gratis harus selalu menyertakan logo IN2AR di dalam aplikasi.

2. Dukungan pengembang di forum IN2AR sangat kuat. Pengembang aktif menjawab pertanyaan seputar teknis IN2AR dari *user frontend*.
3. IN2AR dapat mengenali hampir semua jenis gambar sebagai marker. Berbeda dengan ARToolkit dan FLARToolkit yang hanya mengenali gambar dengan bingkai hitam sebagai marker.
4. Distribusi aplikasi yang dibuat dengan IN2AR cukup mudah. Aplikasi didistribusikan menggunakan file .swf yang dapat ditampilkan di halaman web. *User frontend* yang memiliki web browser dengan flash player dan komputer webcam serta terkoneksi internet akan dapat mengakses aplikasi AR tersebut.

Selain memiliki beberapa kelebihan, IN2AR juga memiliki kelemahan yaitu:

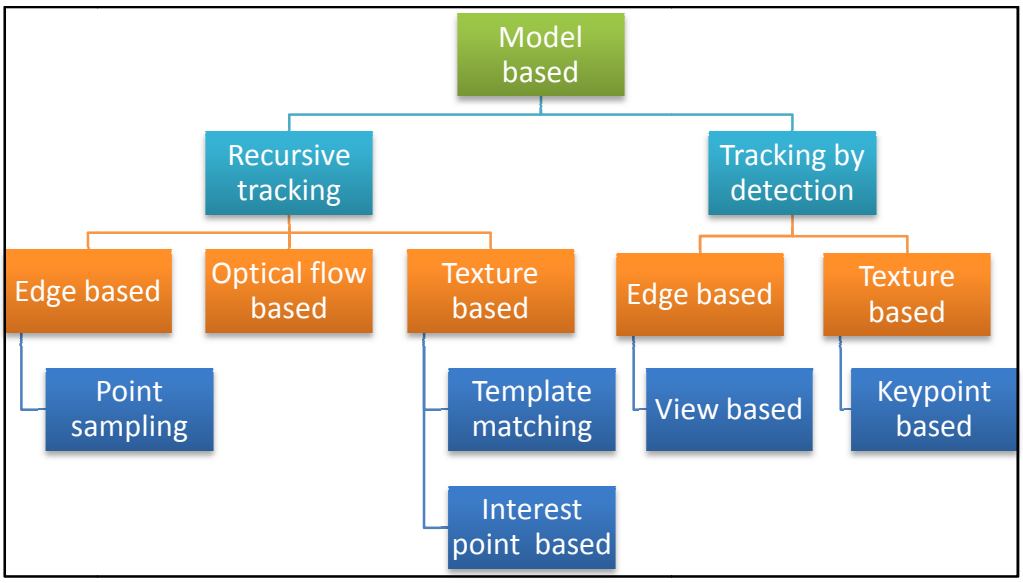
1. Pengguna harus mengenal bahasa pemrograman, terutama bahasa pemrograman Action Script 3. Hal ini karena segala macam perubahan dalam penggunaan IN2AR adalah dengan cara mengetikkan kode dalam bahasa pemrograman Action Script 3. IN2AR tidak memiliki dokumentasi yang lengkap dan sistematis. Pengguna IN2AR harus melakukan eksplorasi sendiri untuk dapat menggunakan IN2AR sesuai kebutuhan. Aktivitas di forum IN2AR akan sangat membantu jika pengguna mengalami kesulitan.

2. Pembuatan marker untuk lisensi gratis harus melalui tim IN2AR. Gambar yang akan dijadikan marker harus dikirim terlebih dahulu ke tim IN2AR untuk dibuatkan file .ass.



Gambar 3. 3 Proses Pembentukan *marker*

3.1.5 Analisis *Markerless* AR *Tracking System*



Gambar 3. 4 *Model based online monocular markerless AR taxonomy*

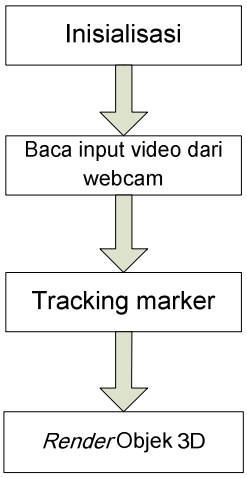
Mengingat sifat pelacakannya, teknik berbasis model (*model based techniques*) dapat diklasifikasikan dalam 2 rublik (Gambar 3.4) : Pertama, pelacakan rekursif (*Recursive tracking*), dimana pada saat sebelum objek dimunculkan terlebih dahulu dilakukan perkiraan untuk menghitung pelacakan yang terjadi dan kedua, pelacakan dengan pendeteksian (*Tracking by detection*), pada klasifikasi ini dilakukan perhitungan pelacakan tanpa perkiraan sebelumnya, yang memungkinkan otomatis inisialisasi dan pemulihan dari kegagalan.

Pelacakan rekursif menjadi teknik model yang digunakan oleh pihak IN2AR untuk melakukan teknik *markerless ar tracking system.*

3.1.6 Analisis Metode Terhadap Kasus

Analisis metode terhadap kasus merupakan analisis yang mendeskripsikan bagaimana proses *augmented reality* IN2AR dari awal inisialisasi, *tracking marker,* sampai dengan proses *rendering* objek 3D dengan metode *markerless* yang diterapkan oleh IN2AR.

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 (*markerless*). Sistem *Augmented Reality Display* yang digunakan adalah teknik *spatial display* (pembahasan pada subbab 2.7.3.3) dengan *screen display* (bisa menggunakan monitor ataupun proyektor).



Gambar 3. 5 Diagram alir sistem *Augmented Reality*

Secara keseluruhan, proses sistem *augmented reality* dapat digambarkan dengan diagram alir pada gambar 3.5. Aplikasi melakukan inisialisasi terlebih dahulu sebelum melakukan *tracking marker*, marker dideteksi dari masukan video webcam, jika marker terdeteksi maka objek 3D di-*render* .

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

1. Inisialisasi
2. *Tracking marker*
3. *Rendering* Objek 3D

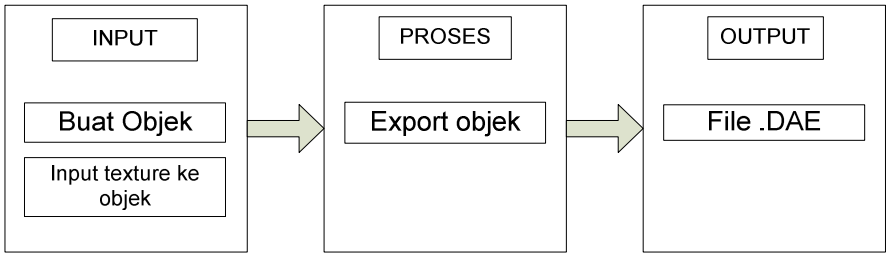
3.1.6.1 Inisialisasi

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.

3.1.6.1.1 Inisialisasi Model 3D

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.

1. Proses Pembentukan Objek



Gambar 3. 6 Proses pembentukan data objek 3D



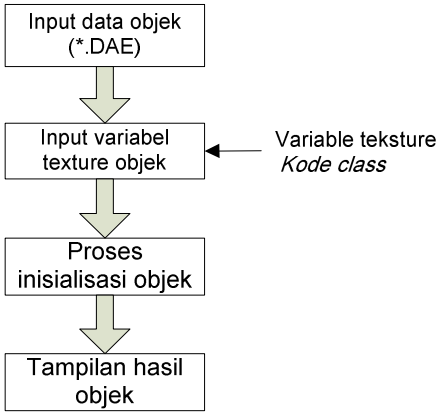
Gambar 3. 7 Proses export objek

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.

2. Proses Input Data Objek



Gambar 3. 8 Proses Input Data Objek

1) Input data objek

Proses pertama yaitu menambahkan data objek .DAE ke dalam *library* IN2AR. File .DAE ini berfungsi untuk memanggil bentuk objek yang sudah diexport.

2) Input variabel teksur objek

Proses ini berfungsi untuk menambahkan tekstur pada objek, tekstur pada objek tidak bisa digunakan atau muncul sebelum ada penambahan variabel dan pengaturan kode yang ditambahkan pada *function init3Dmodel* agar tekstur tersebut sesuai dengan model yang telah dibuat.

1. Penambahan variabel teksture.

   Penambahan variabel teksture disesuaikan dengan jumlah dan nama material id pada file .DAE yang digunakan pada objek.

2. Penambahan kode *class* variabel teksture.

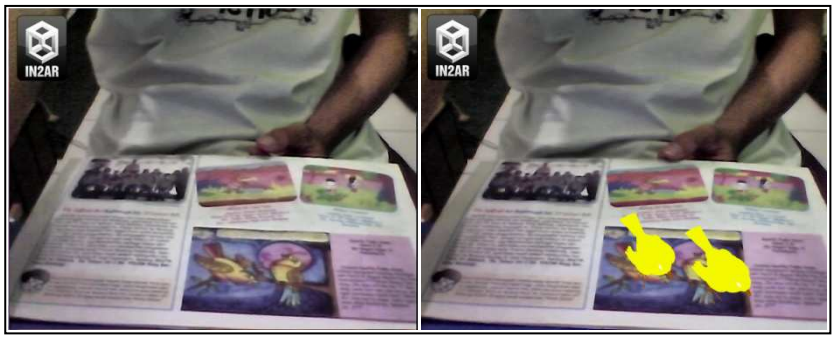
   Penambahan kode *class* pada gambar atau tekstur yang ditambah disesuaikan dengan variabel teksture.

3) Proses inisialisasi objek

Proses ini dilakukan pada saat objek di-*compile* menjadi aplikasi dan data objek beserta teksturnya objek akan di inisialisasi.

4) Tampilan hasil objek

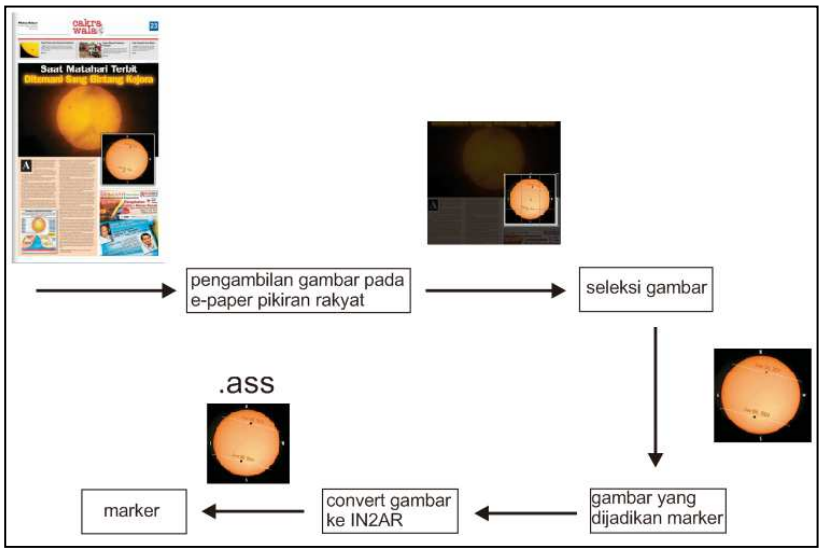
Proses yang terakhir adalah menampilkan hasil objek yang telah melalui proses inisialisasi, dimana pada saat objek tampil disesuaikan dengan *marker* yang ada.



Gambar 3. 9 Tampilan hasil objek

3.1.6.1.2 *Marker* Yang Digunakan

Marker yang digunakan pada analisis ini didapatkan dari hasil gambar pada koran yang sudah ada, ini disebabkan karena aplikasi ini masih bersifat sementara belum di resmikan. Namun kedepanya setelah aplikasi diresmikan *marker* yang digunakan akan sesuai dengan rublik dan informasi yang akan ditampilkan oleh pihak pembuat berita. Adapun proses pengambilan *marker* adalah sebagai berikut :

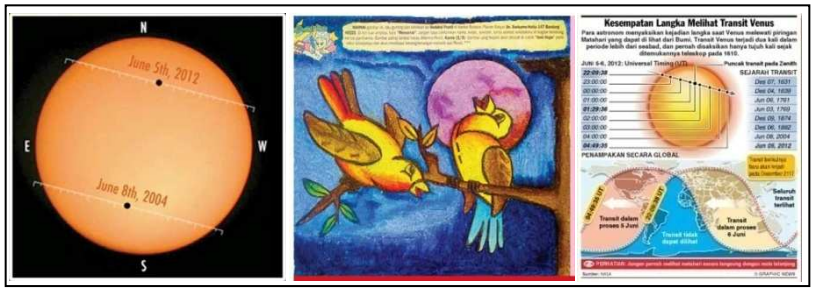


Gambar 3. 10 Proses pembentukan marker

Proses pembuatan gambar *marker* menjadi *marker* dilakukan oleh pihak IN2AR dengan cara mengirimkan gambar yang akan di-*convert,* melalui *marker engine* yang disediakan oleh pihak IN2AR. Setelah gambar di *convert* menghasilkan file dengan format .ass. File tersebut kemudian dijadikan masukan pada *coding* untuk mendeteksi gambar yang dijadikan marker.

Spesikasi pola *marker* :

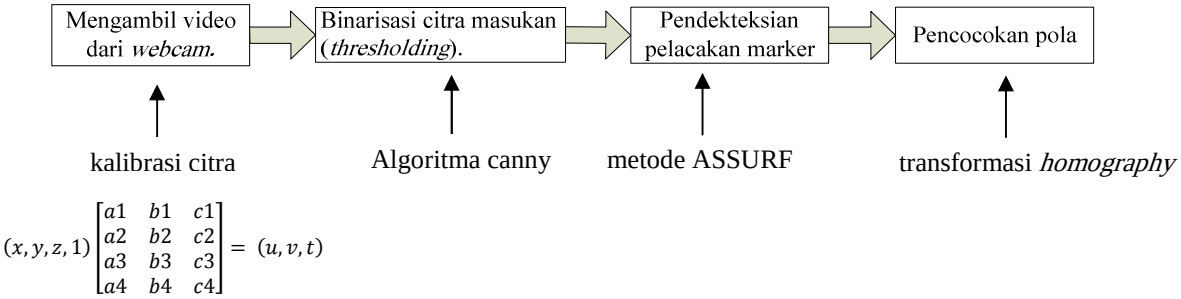
1. Pola *marker* minimum harus memiliki lebar 550 pixels
2. Format gambar yang dikirimkan .jpg



Gambar 3. 11 *Marker* yang digunakan

3.1.6.2 *Tracking marker*

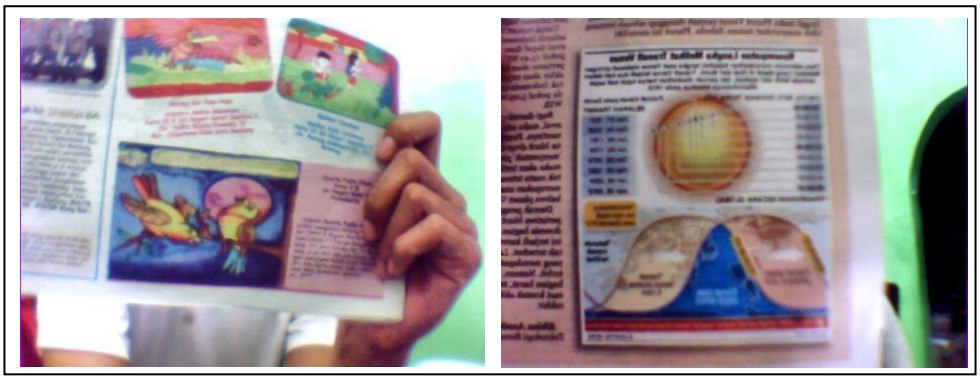
IN2AR memiliki kemampuan untuk mendeteksi gambar dan menghitung posisi gambar tersebut menggunakan webcam standar. Informasi posisi yang didapatkan akan dipergunakan untuk menempatkan objek atau model tiga dimensi atau video ke dalam posisi gambar atau *markerless*. Ada empat langkah, dalam proses kerja *Tracking marker* IN2AR.



Gambar 3. 12 Proses kerja *Tracking marker* IN2AR

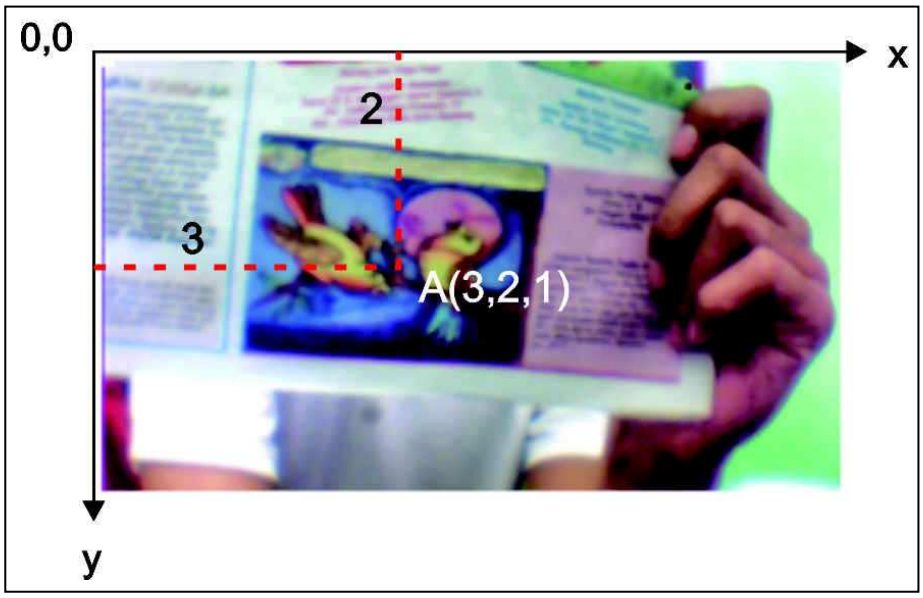
3.1.6.2.1 Mengambil video dari *webcam*

Mendapatkan masukan video dari sebuah *webcam* adalah langkah awal yang harus dilakukan, seperti yang ditunjukan gambar 3.13 .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.. Tujuan dari kalibrasi *webcam* adalah untuk menghitung tingkat distorsi dari sebuah lensa *webcam* yang digunakan agar citra yang dihasilkan mendekati citra ideal.



Gambar 3. 13 Mengambil citra dari *webcam*

Berikut ini adalah proses kalibrasi citra realtime :



Gambar 3. 14 Menentukan koordinat-koordinat citra

Misalkan koordinat citra A(3,5,1), kemudian konversi koordinat-koordinat citra ke dalam dunia nyata dengan matriks 3 x 4 seperti berikut :

$$(3,5,1,1)\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ 1 & 1 & 1\end{bmatrix} = (u,v,t)$$

Jika dilanjutkan maka akan diperoleh :

u = 3.1 + 5.0 + 1.0 + 1 = 4

v = 3.0 + 5.1 + 1.0 + 1 = 6

t = 3.0 + 5.0 + 1.1 + 1 = 2

Maka koordinat citra adalah : (U,V) dimana U = 4/2 = 2 , V = 6/2 =3, jadi koordinat citra pada komputer yaitu (2,3)

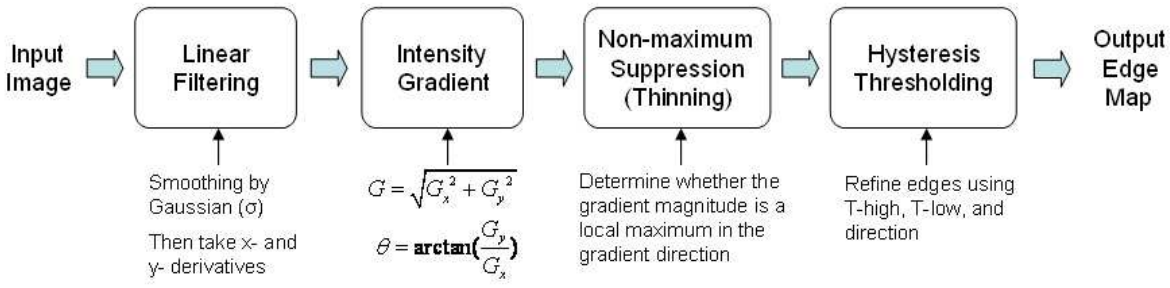
3.1.6.2.2 Binarisasi citra masukan (*thresholding*).

Langkah pertama pada aplikasi visi komputer yang terletak pada deteksi tepi adalah untuk men-*threshold* sumber citra atau disebut juga binarisasi seperti yang ditunjukkan gambar 3.12.



Gambar 3. 15 *Thresholding*

Algoritma yang digunakan pada proses ini adalah algoritma Canny, dimana ada beberapa tahapan yang dilaluinya untuk menemukan hasil yang optimal (Gambar 3.16).



Gambar 3. 16 Tahapan Algoritma Canny

Fungsi dari proses ini adalah untuk membantu sistem agar dapat mengenali bentuk segi empat dan pola marker pada citra yang diterima. Proses tahapannya akan dijelaskan sebagai berikut :

Langkah 1 : Penghalusan citra dengan filter *Gaussian.*

Misal I [i,j] menunjukan suatu citra pada proses kalibrasi, G [i,j;σ] adalah filter penghalus Gaussian dimana σ adalah penyebaran Gaussian dan pengendali derajat penghalus. Hasil konvulusi pada I [i,j] dengan G [i,j;σ] diperoleh suatu array pada penghalusan data sebagai :

$$S\,[i,j] = G\,[i,j;\sigma] \otimes I\,[i,j] \qquad (1)$$

Dimana

$$G(x) = \frac{1}{\sqrt{2\cdot\pi}\cdot\sigma} \cdot e^{-x^2/2\sigma^2} \qquad (2)$$

Psuedo code filter *Gaussian*

```
Function filter_Gaussian
{IS : objek gambar dan nilai koordinat citra,serta elemen
matriks sudah terdefinisi }
{FS : objek gambar diperhalus dengan filter Gaussian  }
Kamus :
p                    : array of real
b                    : Tbitmap
b2                   : Tbitmap
sigma                : real

Procedure gauss
{IS : titik koordinat citra sudah terdefinisi }
{FS : mencari nilai citra dari penghalusan gaussian }
Kamus :
   sigma        : real
   i            : integer
   sum, w       : real
   norm         : real
   mid, ksize   : integer
   ker          : array of real

Algoritma
   norm <- 1 / (sigma x sqtr(2 x pi))
   ediv <- 0,5 (sigma x sigma)
   ksize <- 1 + 2 x round (3 x sigma)
   if ksize < 3 then ksize <- 3
   if ((ksize dan 1) <> 1) then ksize <- ksize + 1
   setlength(ker, ksize)
   mid <- ksize div 2
   sum <- 0
```

```
   for i <- 0 to ksize - 1 do begin
   w <- norm x exp( -( i - mid ) x ( i - mid ) x ediv
   )
   ker[i] <- w
   sum <- sum + w
   Endfor
   for i <- 0 to high( ker ) do
   ker[i] <- ker[i] / sum
   Endfor
Endprocedure
```

Langkah 2 : Menghitung Besar dan Arah Gradien

Gradien array S [i,j] pada persamaan 1 yang diperhalus digunakan untuk menghasilkan turunan parsial x dan y yaitu P [i,j] dan Q [i,j] berturut-turut dengan operator sobel :

P [i,j] ~ (S[i,j +1] – S[i,j] + S[i+1,j+1] – S[i+1,j])/2 (3)

Q [i,j] ~ (S[i,j]- S[i+1,j] + S[i,j+1] – S [i+1,j+1)/2 (4)

Turunan parsial x dan y dihitung dengan rata-rata *finite-difference* lebih dari 2x2 persegi. Dari rumus standar untuk konversi rectangular-to-polar, besar dan arah gradien dapat dihitung sebagai :

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2} \quad (5)$$

Langkah 3 : *Non-maxima Suppresion*

Algoritma dimulai dengan membandingkan nilai-nilai matriks tetangga sebelah kiri dan kanan, atas atau bawah dan secara diagonal untuk

mencari titik yang berada ditengahnya berdasarkan besarnya nilai magnitude yang didapat pada persamaan 5 (M[i,j] biasanya nilai berkisar dari 0-3). Jika besar nilai titik tengahnya (pusat) lebih kecil dari kedua nilai tetangganya maka diset nol. Jika sebaliknya maka nilainya diganti.

$$\begin{bmatrix} 3 & 2 & 1 \\ 0 & \bullet & 0 \\ 1 & 2 & 3 \end{bmatrix}$$

Dimana

$$\theta[i,j] = \arctan(Q[i,j], P[i,j]) \quad (6)$$

Fungsi arctan (x,y) disini mengambil dua argumen dan membangkitkan suatu sudut.

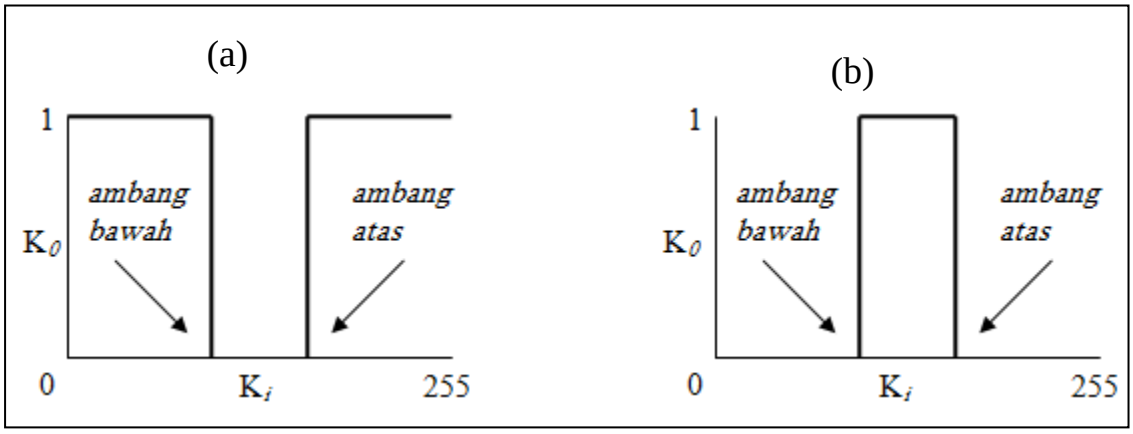
Langkah 4 : Thresholding

Pada operasi pengambangan ini mengunakan pengambangan ganda. berikut persamaanya :

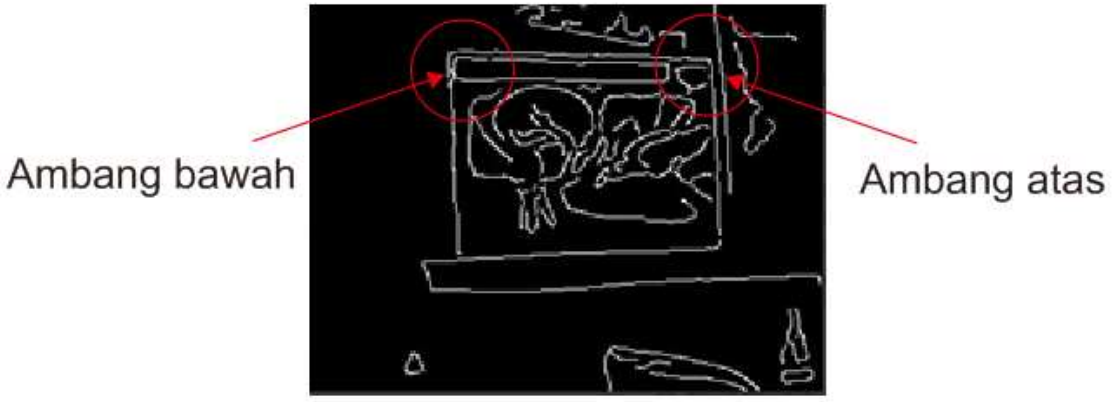
$$K_0 = \begin{cases} 0, \text{ jika } \textit{ambang bawah} <= K_i <= \textit{ambang atas} \\ 1, \text{ lainnya} \end{cases} \quad (7)$$

atau

$$K_0 = \begin{cases} 1, \text{ jika } \textit{ambang bawah} <= K_i <= \textit{ambang atas} \\ 0, \text{ lainnya} \end{cases} \quad (8)$$



Gambar 3. 17 (a) persamaan 7 (b) persamaan 8



Gambar 3. 18 Pengembangan ganda (*thresholding ganda)*

Psuedo code algoritma canny

```
function canny
{IS : objek gambar dan nilai koordinat citra,serta elemen matriks
sudah terdefinisi }
{FS : objek gambar dirubah menjadi vektor ciri }

Procedure mencari_titik_koordinat_citra
  {IS : titik koordinat citra belum diketahui}
  {FS : menemukan titik koordinat citra}
Kamus :
    b               : TBitmap {citra}
    x, y            : integer
    i, j            : integer
    pr              : ParrRGB
    br              : Tbitmap
    thrlow, thrhigh : real
    mag             : array of array of real
Algoritma :
   hasil <- ( x >= 0 ) and ( y >= 0 ) and ( x < b.Width ) and ( y <
   b.Height ) {mencari koordinat citra}
   for hasil <- false
   if not edge[y][x] then exit
   edge[y][x] <- false
   hasil <- mag[y][x] >= thrhigh
   for j <- -1 to 1 do
   for i <- -1 to 1 do
   if ok( x + i, y + j ) then
   hasil <- trace( x + i, y + j ) {jika hasil rekursi menghasilkan
    true}{terhubung dengan piksel yang nilainya lebih besar dari hi-
    threshold}
       if hasil then
    pr <- br.Scanline[y]
    pr[x] <- rgb_hitam  {menampilkan citra baru berwarna putih}
    Endfor
    Endfor
    Endfor
Endprocedure

Procedure penghalusan_dengan_gaussian
    {IS : titik koordinat citra sudah terdefinisi}
    {FS : mencari nilai citra dari penghalusan gaussian }
Kamus :
```

```
    b               : TBitmap {citra}
    bg              : TBitmap
    j               : integer
    p               : array of PArrRGB
    mag, gx, gy     : array of array of real
    edge            : array of array of boolean
 Algoritma :
    bg <- gaussian(b,1)
    hasil <- citra_create(b.Width, b.Higth) {membuat citra(bitmap)
    baru}
    setlength( p, b.Height )
    for j <- 0 to high( p ) do
    p[j] <- bg.ScanLine[j]
    setlength( gx, b.Height )
    for j <- 0 to high( gx ) do
    setlength( gx[j], b.Width )
    setlength( gy, b.Height )
    for j <- 0 to high( gx ) do
    setlength( gy[j], b.Width )
    setlength( mag, b.Height )
    for j <- 0 to high( mag ) do
    setlength( mag[j], b.Width )
    setlength( edge, b.Height )
    for j <- 0 to high( edge ) do
    setlength( edge[j], b.Width )

Procedure  menghitung_hitung_gradien_gx_dan_gy
    {IS : gradien gx dan gy sudah terdefinisi}
    {FS : mencari nilai gradien menggunakan operator sobel}
Kamus :
    i, j            : integer
    mag, gx, gy     : array of array of real
    p               : array of PArrRGB
Algoritma :
    for j <- 1 to b.Height - 2 do begin
    for i <- 1 to b.Width - 2 do begin
    gy[j][i] <- ( p[j + 1][i - 1].r - p[j - 1][i - 1].r ) + 2 x ( p[j
    + 1][i].r - p[j - 1][i].r ) + ( p[j + 1][i + 1].r - p[j - 1][i +
    1].r )
    gx[j][i] <- ( p[j - 1][i + 1].r - p[j - 1][i - 1].r )
    + 2 x ( p[j][i + 1].r - p[j][i - 1].r ) + ( p[j + 1][i + 1].r -
```

```
    p[j + 1][i - 1].r )
    mag[j][i] <- sqrt( gx[j][i] * gx[j][i] + gy[j][i] * gy[j][i] )
    Endfor
    Endfor
Endprocedure

Procedure  menghitung_non-maximal_suppression
    {IS : gradien gx dan gy sudah terdefinisi}
    {FS : mencari nilai non-maximal suppression}
Kamus :
    i, j            : integer
    dx, dy          : integer
    mag, gx, gy     : array of array of real
    edge            : array of array of boolean
Algoritma :
    for j <- 1 to b.Height - 2 do begin
    for i <- 1 to b.Width - 2 do begin
    dir <- arctan2( gy[j][i], gx[j][i] )
    dx <- round( cos( dir ) )
    dy <- round( sin( dir ) )
    edge[j][i] <- ( mag[j][i] >= mag[j + dy][i + dx] ) and ( mag[j][i]
    >= mag[j - dy][i - dx] )
    if edge[j][i] then begin
    edge[j + dy][i + dx] <- false
    edge[j - dy][i - dx] <- false
    Endfor
    Endfor
Endprocedure

Procedure menghitung_nilai_thresholding
    {IS : nilai on-maximal suppression sudah terdefinisi}
    {FS : mencari nilai hysteresis thresholding }
Kamus :
    i, j            : integer
    thrlow, thrhigh : real
    br              : Tbitmap
    edge            : array of array of boolean
Algoritma :
    thrlow <- 10
    thrhigh <- 90
```

```
    br <- hasil
    for j <- 1 to b.Height - 2 do begin
    for i <- 1 to b.Width - 2 do begin
    if edge[j][i] then trace( i, j ){ melakukan trace edge-linking }
    Endfor
    Endfor
Endprocedure
```

3.1.6.2.3 Pendeteksian pelacakan marker.

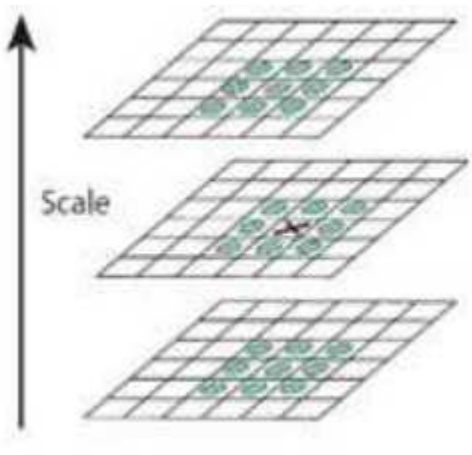
Langkah berikutnya dari IN2AR adalah menemukan area yang berdampingan dalam citra yang di-*treshold*. Area yang berdampingan diberi tanda sebagai persegi (*marker outline*). menggunakan metode ASSURF dengan algoritma utamanya algoritma SURF (*Speeded Up Robust Feature*). Proses tahapannya akan dijelaskan sebagai berikut :

Langkah 1 : Menentukan *keypoint*

Proses pertama yaitu menentukan *keypoint* ketika *treshold* dinaikan, jumlah *keypoint* yang terdeteksi lebih kecil begitupun sebaliknya.

Langkah 2 : *Non-maxima Suppresion*

Proses ini dilakukan untuk mencari sekumpulan calon *keypoint* dengan membandingkan tiap-tiap *pixel* gambar pada *scale space* dengan 26 tetangga. 26 tetangga pixel itu terdiri atas 8 titik di scale asli dan 9 titik di tiap-tiap scale diatas dan dibawahnya. Proses inilah yang menghasilkan keypoint dari suatu gambar. Gambar 3.19 menunjukkan *non-maxima suppresion.*



Gambar 3. 19 Non-maxima suppression

Langkah 3 : Proses mencari lokasi keypoint

Proses terakhir yaitu proses mencari lokasi keypoint menggunakan interpolasi data yang dekat dengan keypoint hasil proses sebelumnya. Ini dilakukan dengan mencocokan quadratic 3D yang diajukan oleh Brown $\mathrm{H}(x, y, \sigma)$ adalah determinan Hessian, didefinisikan sebagai berikut :

$$\mathrm{H(x)} = \mathrm{H} + \frac{\partial \mathrm{H}^{\mathrm{T}}}{\partial \mathrm{x}}\mathrm{x} + \frac{1}{2}\mathrm{x}^{\mathrm{T}}\frac{\partial^2 \mathrm{H}}{\partial x^2}x \qquad (9)$$

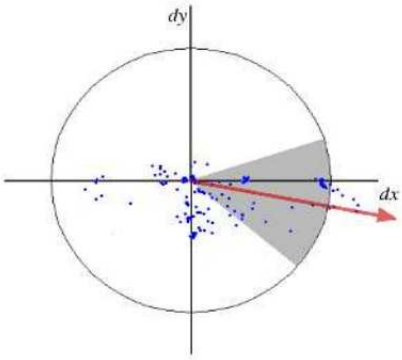
Lokasi ekstrim yang diinterpolasi, $\hat{x} = (x, y, \sigma)$, ditemukan dengan mencari turunan dari fungsi diatas dan diberi nilai nol, sehingga:

$$\hat{x} = -\frac{\partial^2 \mathrm{H}}{\partial x^2}\frac{\partial \mathrm{H}}{\partial x} \qquad (10)$$

Langkah 4 : Proses ekstraksi deskriptor

Dilakukan dengan membuat daerah kotak persegi disekitar *keypoint*, dimana *keypoint* sebagai pusat dari daerah kotak persegi tersebut,

dan orientasinya disekitar orientasi yang ditentukan dan ditunjukkan pada Gambar 3.20.



Gambar 3. 20 Jendela pergeseran orientasi

Kemudian, respon wavelet *dx* dan *dy* dijumlahkan untuk setiap sub-region. Hal ini akan memberikan informasi tentang polar dari perubahan intesitas, dan juga akan dihasilkan jumlah nilai absolut dari respon |dx| dan |dy|. Masing-masing sub-region mempunyai 4 dimensi deskriptor vektor v, yaitu dx ,dy, |dx| dan |dy|. Untuk 4 x 4 sub-region, maka panjang vektor deskriptornya berjumlah 64.

$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right) \quad (11)$$

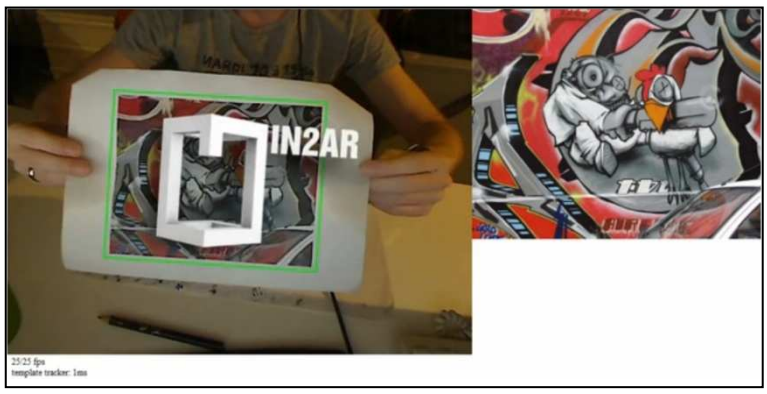
Sehingga yang akan ditampilkan pada layar hanyalah area yang memiliki bentuk segi empat dan pola-pola gambar yang sudah di-*labelling* (Gambar 3.21).



Gambar 3. 21 Pendeteksian pelacakan marker (*marker detection tracking)*

3.1.6.2.4 Pencocokan pola.

Setelah semua area persegi dan pola-pola gambar ditandai, IN2AR menganalisa citra yang berada di dalam persegi dan membandingkan polanya dengan sekumpulan pola yang telah ditentukan (pencocokan pola). IN2AR mengekstrak pola didalam persegi menggunakan transformasi *homography*. IN2AR memberikan memberikan nilai *confidance* kepada pola yang cocok, jika kecocokannya di atas nilai yang telah ditentukan maka polanya dinyatakan cocok (Gambar 3.22)



Gambar 3. 22 Pencocokan pola

Psuedo code transformasi *homography* IN2AR

```
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 ]) )
```

3.1.6.3 *Rendering* Objek 3D

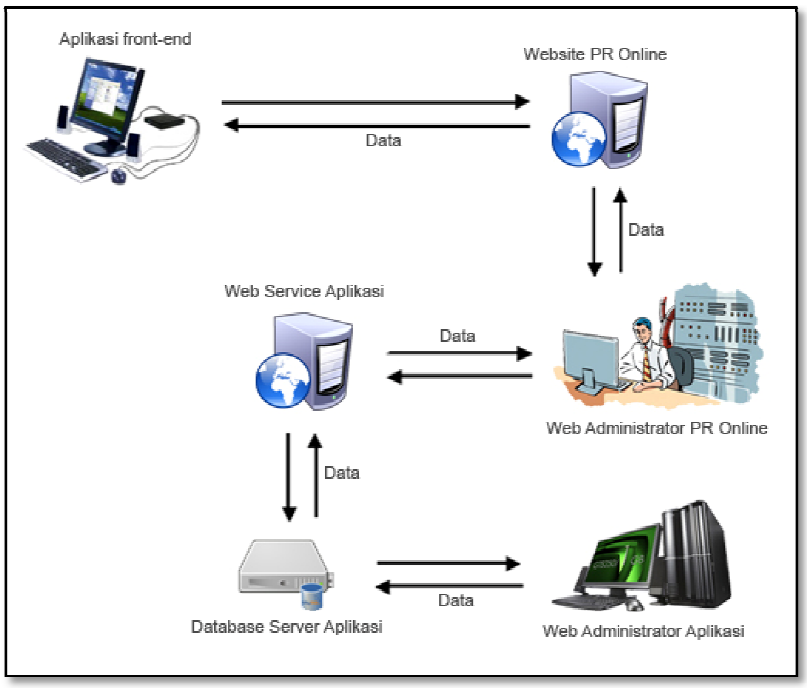
Transformasi matriks yang dikalkulasikan di step sebelumnya yang digunakan IN2AR dan menampilkan objek yang sesuai dengan sebuah library 3D, seperti yang ditunjukkan gambar 3.23. IN2AR menyertakan kelas pendukung yang mengkonversikan transformasi matriks IN2AR ke setiap kelas matriks internal library 3D tersebut.



Gambar 3. 23 *Render* Objek 3D

3.1.7 Analisis Alur Data Sistem

Analisis ini berfungsi untuk menggambarkan secara rinci bagaimana sistem bekerja.



Gambar 3. 24 Alur Data Sistem

Analisis ini meliputi :

1. Aplikasi *front-end*

   Aplikasi dalam hal ini adalah adalah analisis fungsionalitas aplikasi untuk mengakses data dari database server melalui website PR *online*.

2. *Website* PR *Online*

   Website pada penelitian ini berfungsi sebagai *middle application* atau perantara aplikasi dengan *database server*. Cara kerja *website* ini yaitu mengambil tampilan data aplikasi dari *database server* aplikasi dan kemudian mengkonversi data tersebut ke *website* pr *online* agar

dapat di tampilkan ke situs PR *Online* yang diakses oleh *user front-end.*

3. *Web Administrator* PR *Online*

   Web Administrator PR *Online* disini bertugas melakukan konfigurasi *cross server* antara *web service* aplikasi dengan *website* pr *online.* komunikasi data antar aplikasi dengan *database server* menggunakan *parsing* data dari web service yang bertipe XML.

4. *Web Service* Aplikasi

   Web Servive Aplikasi disini menampilkan tampilan aplikasi sesuai data yang ada pada database server aplikasi yang nantinya akan ditampilkan pada situs PR *Online*.

5. *Database Server* Aplikasi

   Pada analisis ini database server aplikasi menampung data aplikasi yang akan dibuat oleh *web administrator* aplikasi yang kemudian akan digunakan sesuai dengan keperluan pada *web service* pr aplikasi.

6. *Web Administrator* Aplikasi

   Web Administrator Aplikasi disini bertugas untuk membuat objek yang akan ditampilkan pada aplikasi berupa animasi atau video sesuai dengan informasi yang ada pada koran.

3.1.8 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitik-beratkan pada properti perilaku yang dimiliki oleh

sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.8.1 Analisis Perangkat Lunak

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

1) Sistem operasi Windows 7 Ultimate
2) FlashDevelop Version 4.0.0 RC3
3) Autodesk 3ds Max 2012
4) Adobe Flash CS5
5) Adobe Dreamweaver CS5

3.1.8.2 Analisis Perangkat Keras

Komputer dan *wabcam* terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan tugas tertentu. Pembangunan aplikasi *ARNewspaper*, perangkat keras yang digunakan adalah sebagai berikut :

1) *Processor* AMD A6-3400M 1,4Ghz
2) RAM 4GB
3) *Harddisk* 750GB
4) *Mouse* dan *Keyboard*
5) Webcam 30 fps 5 Mega Pixel
6) Printer Canon IP1980
7) Koran Pikiran Rakyat

3.1.8.3 Analisis Pengguna User

Analisis pengguna dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat dalam proses penggunaan aplikasi *ARNewspaper* sehingga dapat diketahui hak akses masing – masing pengguna dapat dilihat pada tabel 3.2.

Pengguna sistem dibagi menjadi dua bagian, yaitu :

1 Pengguna aplikasi *ARNewspaper* (*user frontend*) yaitu para pembaca koran Pikiran Rakyat. Dalam menggunakan sistem ini, pengguna diharuskan terhubung dengan koneksi jaringan internet dan memiliki *webcam* untuk bisa menggunakan aplikasi ini.

2 Admin (*user backend*) sebagai pengelola konten mempunyai kapabilitas dalam mengelola *database* seperti menambah, mengubah, dan menghapus data melalui web administrator.

Tabel 3. 2 Tabel Karakteristik Pengguna

Pengguna	Hak Akses	Hak Akses
User Frontend	Mengakses data informasi *ARNewspaper* melalui aplikasi *front-end*	Melihat Informasi
User backend	Mengakses, mengelola aplikasi *back-end*	Edit, Tambah, Hapus rublik dan data, Menampilkan data.

3.1.8.4 Analisis Pengkodean

Kode merupakan penyajian dalam mengklasifikasikan data sehingga mudah dalam proses memasukan ke dalam sistem. Penggunaan kode digunakan untuk mengidentifikasi data.



Pengurutan berdasarkan kode rublik , tanggal terbit, dan judul sesuai dengan masukan yang dilakukan diawal.

Misalkan kode data “220120607PLA” terbagi dari 3 kode dari masing- masing field .

1. “2” = merupakan id rublik cakrawala.
2. “20120607” = berasal dari nama field tanggal terbit koran dengan format “yyyy/mm/dd”.
3. “PLA” = diambil dari nama field judul planet venus, sama seperti kode pertama yang di *substr* tetapi diambil 3 huruf didepan dan di *uppercase.*

3.1.9 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis kebutuhan fungsional ini meliputi analisis kebutuhan sistem, analisis kebutuhan konten, dan pemodelan sistem.

3.1.9.1 Analisis Kebutuhan Sistem

Dalam pembangunan aplikasi perancangan sistem, sebelumnya dilakukan analisis kebutuhan sistem agar sistem yang dibangun sesuai dengan yang diharapkan atau dibutuhkan pengguna. Adapun kebutuhan sistem yang diperlukan antara lain :

1 Menampilkan informasi berita animasi atau video pada *display user frontend*.
2 Informasi berita animasi atau video yang diinformasikan jelas.
3 Memberikan informasi visual mengenai berita yang terkait dengan bagian atau rublik koran.
4 Informasi mengenai berita animasi atau video yang sudah diterapkan teknologi *augmented reality* segera ditampilkan.

3.1.9.2 Analisis Kebutuhan Konten

Analisis ini bertujuan untuk menganalisis konten yang diperlukan agar memudahkan perancangan informasi setelah sistem yang dibutuhkan

telah diketahui. Adapun analisis kebutuhan konten dengan kriteria sebagai berikut :

1. Konten Video

   Menampilkan konten video berupa video informasi yang sesuai dengan berita.

2. Konten Animasi

   Menampilkan konten animasi berupa objek 3D.

3.1.9.3 Pemodelan Sistem

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

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 melalui sebuah cerita bagaimana sebauh sistem dipakai. *Use case* terdiri dari tiga bagian yaitu identifikasi *actor*, identifikasi *use case*, dan *scenario use case*.

1.1 Identifikasi Aktor

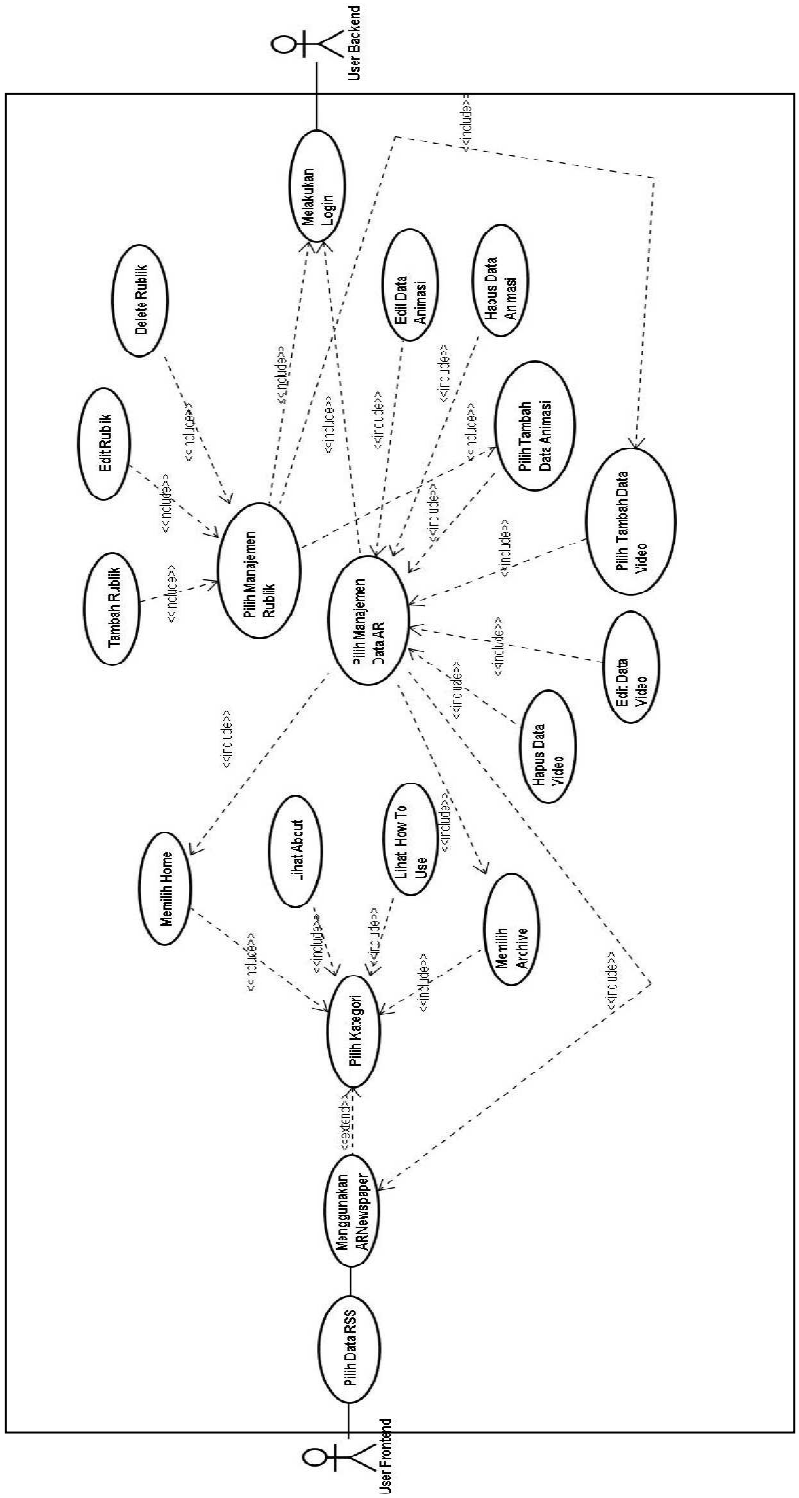
Aktor adalah abstraksi dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem. Berikut adalah aktor yang berperan dalam menjalankan sistem yang dibangun.

Tabel 3. 3 Tabel *Actor Use Case*

No	Aktor	Deskripsi
A-01	*User Frontend*	Merupakan *actor* atau pengguna akhir dari aplikasi yang dibangun, aplikasi *frontend* (*ARNewspaper*)
A-02	*User Backend*	Merupakan *actor* yang berperan mengelola data informasi dalam *database server* melalui *web administrator*.

Penggambaran *use case diagram* sistem dibagi menjadi 2 bagian yaitu *use case diagram frontend* dan *use case diagram backend*. Kedua *use case diagram* menunjukkan bagaimana interaksi antara aktor dan sistem, dimana terdapat dua aktor yang berperan dan sistem yang terdiri dari aplikasi *frontend* dan *backend*. Aktor *user fronted* yaitu yang menjalankan aplikasi *frontend*. Sedangkan aktor *user backend* berperan mengelola aplikasi *backend* pada *web administator*.

1.2 *Use Case Diagram*



Gambar 3. 25 *Use Case Diagram*

1.3 Identifikasi *Use Case Diagram*

Berikut tabel identifikasi use case yang terdapat dalam sistem :

Tabel 3. 4 Identifikasi *Use Case Diagram*

Use Case* Aplikasi *Frontend		
No	***Use Case***	**Deskripsi**
UCF-01	Menggunakan AR*Newspaper*	Fungsionalitas untuk langsung menjalankan dan menampilkan aplikasi AR*Newspaper* dengan menggunakan wabcam pc/laptop.
UCF-02	Pilih Rublik	Fungsionalitas memungkinkan *user frontend* memilih rublik yang ada pada Pikiran Rakyat *Online.*
UCF-03	Melihat Home	Fungsionalitas untuk menampilakan halaman home aplikasi.
UCF-04	Lihat *About*	Fungsionalitas untuk menampilkan About tentang aplikasi dan sedikit penjelasan tentang teknologi *augmented reality.*
UCF-05	Lihat *How to use*	Fungsionalitas untuk menampilkan cara penggunaan aplikasi.
UCF-06	Memilih *Archive*	Fungsionalitas untuk menampilkan dan memilih *archive* / data sesuai dengan tanggal penerbitan koran.
UCF-07	Pilih Data RSS	Fungsionalitas *user frontend* memilih data RSS yang ada pada Pikiran Rakyat *Online.*
Use Case* Aplikasi *Backend		
No	***Use Case***	**Deskripsi**
UCB-01	Melakukan *Login*	Fungsionalitas untuk proses autentifikasi hak akses kepada *user backend*
UCB-02	Pilih Manajemen Rublik	Fungsionalitas yang memungkinkan *user backend* dapat mengelola data rublik.
UCB-03	Tambah Rublik	Fungsionalitas untuk menambahkan data rublik.
UCB-04	*Edit* Rublik	Fungsionalitas untuk mengubah atau *update* data rublik.
UCB-05	Hapus Rublik	Fungsionalitas untuk menghapus data rublik.
UCB-06	Pilih Manajemen Data AR	Fungsionalitas yang memungkinkan *user backend* dapat mengelola data ar.
UCB-07	Pilih Tambah Data Animasi	Fungsionalitas untuk menambahkan data ar animasi.
UCB-08	*Edit* Data Animasi	Fungsionalitas untuk mengubah atau *update* data ar animasi.
UCB-09	Hapus Data Animasi	Fungsionalitas untuk menghapus data ar animasi.
UCB-10	Pilih Tambah Data	Fungsionalitas untuk menambahkan data ar.

	Video	
UCB-11	*Edit* Data Video	Fungsionalitas untuk mengubah atau *update* data ar.
UCB-12	Hapus Data Video	Fungsionalitas untuk menghapus data ar.

1.4 Skenario *Use Case*

1.4.1 Skenario *Use Case* Aplikasi *Frontend*

Skenario proses-proses yang terdapat dalam *use case diagram* aplikasi *frontend* dapat dilihat pada tabel-tabel berikut :

Tabel 3. 5 Skenario *Use Case* Menggunakan AR*Newspaper*

Identifikasi	
Nama Use Case	Menggunakan AR*Newspaper*
Aktor	*User Frontend*
Tujuan	Menggunakan aplikasi AR*Newspaper*.
Keadaan awal	Sistem menampilkan menu *rublik*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan halaman menu aplikasi
2. Menyiapkan *webcam* dan koran yang akan di AR kan dan menekan *allow* yang ditampilkan sistem	
	3. Setelah menekan *allow* maka sistem akan menampilkan informasi pada display sesuai dengan *marker* dan tanggal koran diterbitkan.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1." *Flash Player not detected, Click below to download the latest Adobe Flash Player.*"
2. Menekan tombol "*Download*"	
	3. Sistem akan terminate.
Kondisi Akhir	Menampilkan informasi sesuai yang dipilih.

Tabel 3. 6 Skenario Pilih Rublik

Identifikasi	
Nama Use Case	Pilih Rublik
Aktor	*User Frontend*
Tujuan	Melihat konten berdasarkan rublik
Keadaan awal	Sistem menampilkan menu rublik
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan jenis-jenis rublik (Home, About, How to use, Archive).
2. Memilih jenis rublik yang ditampilkan sistem.	
	3. Jika memilih salah satu dari jenis rublik, Maka informasi yang ditampilkan adalah halaman web rublik yang dipilih.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1.Menampilkan pesan ” *Sorry, a connection Error occured!. Please check if you are connected to the Internet. “*
2. Menekan tombol “*Download*”	
	3. Sistem akan terminate.
Kondisi Akhir	Menampilkan informasi sesuai yang dipilih.

Tabel 3. 7 Skenario *Use Case* Melihat *Home*

Identifikasi	
Nama Use Case	Melihat *Home*
Aktor	*User Frontend*
Tujuan	Menampilkan tampilan awal/*home* aplikasi
Keadaan Awal	Sistem menampilkan konten rublik.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan tombol “Home”.	
	2. Menampilkan konten menu yang dapat dipilih untuk ditampilkan.
Keadaan Akhir	Sistem menampilkan menu aplikasi.

Tabel 3. 8 Skenario *Use Case* lihat *About*

Identifikasi	
Nama Use Case	Lihat *About*
Aktor	*User Frontend*
Tujuan	Menampilkan informasi tentang aplikasi dan teknologi *augmented reality.*
Keadaan Awal	Sistem menampilkan konten rublik.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan tombol “About”.	
	2. Menampilkan informasi about.
Keadaan Akhir	Sistem menampilkan menu about.

Tabel 3. 9 Skenario *Use Case* lihat *How To Use*

Identifikasi	
Nama Use Case	Lihat *How to use*
Aktor	*User Frontend*
Tujuan	Menampilkan informasi tentang penggunaan aplikasi.
Keadaan Awal	Sistem menampilkan konten rublik.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan tombol “*How To Use*”.	
	2. Menampilkan Informasi bagaimana penggunaan menggunakan aplikasi yang baik dan benar.
Keadaan Akhir	Sistem menampilkan menu *How To Use*.

Tabel 3. 10 Skenario *Use case* memilih *Archive*

Identifikasi	
Nama Use Case	Memilih *Archive*
Aktor	*User Frontend*
Tujuan	Melihat detail konten arsip yang dipilih.
Keadaan Awal	Sistem menampilkan konten rublik.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan tombol “Archive”.	
	2. Menampilkan optional *Archive* yang dapat dipilih untuk ditampilkan.

3. Memilih salah satu konten *archive.*	
	4. Menampilkan aplikasi dari konten *archive* yang dipilih.
Skenario Gagal	
Aksi Aktor	**Reaksi sistem**
	1.Muncul pesan "*There is no application information available. Please try again."*
2. Memilih konten yang tersedia.	
	3. Sistem terminate
Keadaan Akhir	Aplikasi menampilkan informasi yang dipilih.

Tabel 3. 11 Sekenario *Use Case* Pilih Data RSS

Identifikasi	
Nama Use Case	Pilih Data RSS
Aktor	*User Frontend*
Tujuan	Menampilkan konten yang dipilih *user frontend*.
Keadaan Awal	Sistem menampilkan situs pikiran-rakyat.com .
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih konten yang ada di Pikiran-rakyat.com.	
	2. Menampilkan konten yang dipilih *user frontend*.
Keadaan Akhir	Sistem menampilkan konten yang dipilih *user frontend*.

1.4.2 Skenario *Use Case* Aplikasi *Backend*

Skenario proses-proses yang terdapat dalam *use case diagram* aplikasi *backend* dapat dilihat pada tabel-tabel berikut :

Tabel 3. 12 Skenario *Use Case* Melakukan *Login*

Identifikasi	
Nama Use Case	Melakukan *Login*
Aktor	*User Backend*
Tujuan	Masuk kedalam sistem
Keadaan Awal	Sistem menampilkan halaman login admin
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**

1. Mengisi *username* dan *password* pada halaman web *login* admin	
	2. Sistem melakukan autentikasi *username* dan *password* yang diiunputkan.
	3. Bila sesuai maka menampilan halaman web menu utama.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan pesan "Ada kesalahan pada username atau password."
2. Melakukan input *username* dan *password* pada halaman web login admin	
Kondisi Akhir	Halaman menu utama ditampilkan

Tabel 3. 13 Skenario *Use Case* Pilih Manajemen Rublik

Identifikasi	
Nama Use Case	Pilih Manajemen Rublik
Aktor	*User Backend*
Tujuan	Menampilkan data rublik secara keseluruhan serta fungsionalitas pengolahan data
Keadaan Awal	Menampilkan halaman manajemen rublik
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih manajemen rublik pada menu navigator.	
	2. Menampilkan halaman manajemen rublik.
Kondisi Akhir	Menampilkan data rublik.

Tabel 3. 14 Skenario *Use Case* Tambah Rublik

Identifikasi	
Nama Use Case	Tambah Rublik
Aktor	*User Backend*
Tujuan	Menambah data Rublik
Keadaan Awal	Sistem menampilkan halaman pengelohan rublik.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih *Tambah* pada pilihan halaman Rublik	
	2. Menampilkan halaman pengisian tambah rublik.
3. Mengisi *field-field* pada	

pengisian tambah *rublik*.	
	4. Melakukan proses tambah data pada basis data.
	5. Menyimpan hasil tambah data.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan pesan kesalahan sesuai dengan *field-field field* pada pengisian tambah rublik.
2. Melengkapi *field-field* pada pengisian tambah rublik	
Kondisi Akhir	Data rublik bertambah sesuai dengan input data.

Tabel 3. 15 Skenario *Use Case* Edit Rublik

Identifikasi	
Nama Use Case	*Edit* Rublik
Aktor	*User Backend*
Tujuan	Mengubah data rublik
Keadaan Awal	Sistem menampilan halaman pengolahan rublik
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mengisi *field-field* pada halaman ubah rublik	
	2. Melakukan proses ubah data pada basis data dan menyimpannya.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	1. Menampilkan pesan kesalahan sesuai dengan *field-field field* pada pengisian edit rublik.
2. Melengkapi *field-field* pada pengisian edit rublik	
Keadaan Akhir	Perubahan data rublik sesuai dengan ubah data.

Tabel 3. 16 Skenario *Use Case* Hapus Rublik

Identifikasi	
Nama Use Case	Hapus Rublik
Aktor	*User backend*
Tujuan	Menghapus data rublik
Keadaan Awal	Sistem menampilkan halaman pengolahan rublik
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih hapus pada kolom aksi tabel Data	

	2. Menampilkan pesan konfirmasi “Apakah Anda Yakin Menghapus Rublik dengan id <id_rublik>”
3. Menyetujui penghapusan dengan memilih atau menekan tombol “OK”	
	4. Menghapus data rublik.
	5. Menyimpan hasil perubahan data rublik
Kondisi Akhir	Perubahan data rublik sesuai dengan data yang dihapus.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	1. Menampilkan pesan kesalahan ”Data tidak bisa dihapus”.
3. Menyetujui pesan kesalahan dengan memilih atau menekan tombol “OK”	
Keadaan Akhir	Kembali ke halaman pengolahan rublik

Tabel 3. 17 Skenario *Use Case* Pilih Manajemen Data AR

Identifikasi	
Nama Use Case	Pilih Manajemen Data AR
Aktor	*User Backend*
Tujuan	Menampilkan tabel data ar
Keadaan Awal	Sistem menampilkan halaman data ar.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih Manajemen data pada menu navigator.	
	2. Menampilkan halaman manajemen data.
Kondisi Akhir	Menampilkan halaman manajemen data.

Tabel 3. 18 Skenario *Use Case* Pilih Tambah Data Animasi

Identifikasi	
Nama Use Case	Pilih Tambah Data AR Animasi
Aktor	*User Backend*
Tujuan	Menambah data ar animasi
Keadaan Awal	Sistem menampilkan halaman pengelohan data animasi.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**

1. Memilih *Tambah* pada pilihan halaman data	
	2. Menampilkan halaman pengisian tambah data.
3. Mengisi *field-field* pada pengisian tambah data.	
	4. Melakukan proses tambah data pada basis data.
	5. Menyimpan hasil tambah data.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan pesan kesalahan sesuai dengan *field-field field* pada pengisian tambah data animasi.
2. Melengkapi *field-field* pada pengisian tambah data	
Kondisi Akhir	Data ar bertambah sesuai dengan input data.

Tabel 3. 19 Skenario *Use Case Edit* Data Animasi

Identifikasi	
Nama Use Case	*Edit* Data Animasi
Aktor	*User Backend*
Tujuan	Mengubah data ar animasi
Keadaan Awal	Sistem menampilan halaman pengolahan data
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih “Edit” pada kolom aksi tabel data	
	2. Mempilkan *field-field* data
3. Mengisi *field-field* pada halaman ubah data ar	
	4. Melakukan proses ubah data pada basis data dan menyimpannya.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan pesan kesalahan sesuai dengan *field-field field* pada pengisian edit data animasi.
2. Melengkapi *field-field* pada pengisian edit data	
Keadaan Akhir	Perubahan data ar sesuai dengan ubah data.

Tabel 3. 20 Skenario *Use Case Hapus* Data Animasi

Identifikasi	
Nama Use Case	*Hapus* Data Animasi

Aktor	*User backend*
Tujuan	Menghapus data ar animasi
Keadaan Awal	Sistem menampilkan halaman pengolahan data
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih hapus pada kolom aksi tabel Data	
	2. Menampilkan pesan konfirmasi "Apakah Anda Yakin Menghapus Data <type_ar> dengan id <id_data>"
3. Menyetujui penghapusan dengan memilih atau menekan tombol "OK"	
	4. Menghapus data ar.
	5. Menyimpan hasil perubahan data ar
Kondisi Akhir	Perubahan data ar sesuai dengan data yang dihapus.

Tabel 3. 21 Skenario *Use Case* Pilih Tambah Data Video

Identifikasi	
Nama Use Case	Pilih Tambah Data Video
Aktor	*User Backend*
Tujuan	Menambah data ar video
Keadaan Awal	Sistem menampilkan halaman pengelohan data video.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih *Tambah* pada pilihan halaman data	
	2. Menampilkan halaman pengisian tambah data.
3. Mengisi *field-field* pada pengisian tambah data.	
	4. Melakukan proses tambah data pada basis data.
	5. Menyimpan hasil tambah data.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan pesan kesalahan sesuai dengan *field-field field* pada pengisian tambah data video.
2. Melengkapi *field-field* pada pengisian tambah data	
Kondisi Akhir	Data ar bertambah sesuai dengan input data.

Tabel 3. 22 Skenario *Use Case Edit* Data Video

Identifikasi	
Nama Use Case	*Edit* Data Video
Aktor	*User Backend*
Tujuan	Mengubah data ar video
Keadaan Awal	Sistem menampilan halaman pengolahan data
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Memilih "Edit" pada kolom aksi tabel data	
	2. Menampilkan *field-field* data
3. Mengisi *field-field* pada halaman ubah data ar	
	4. Melakukan proses ubah data pada basis data dan menyimpannya.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan pesan kesalahan sesuai dengan *field-field field* pada pengisian edit data video.
2. Melengkapi *field-field* pada pengisian edit data	
Keadaan Akhir	Perubahan data ar sesuai dengan ubah data.

Tabel 3. 23 Skenario *Use Case Hapus* Data Video

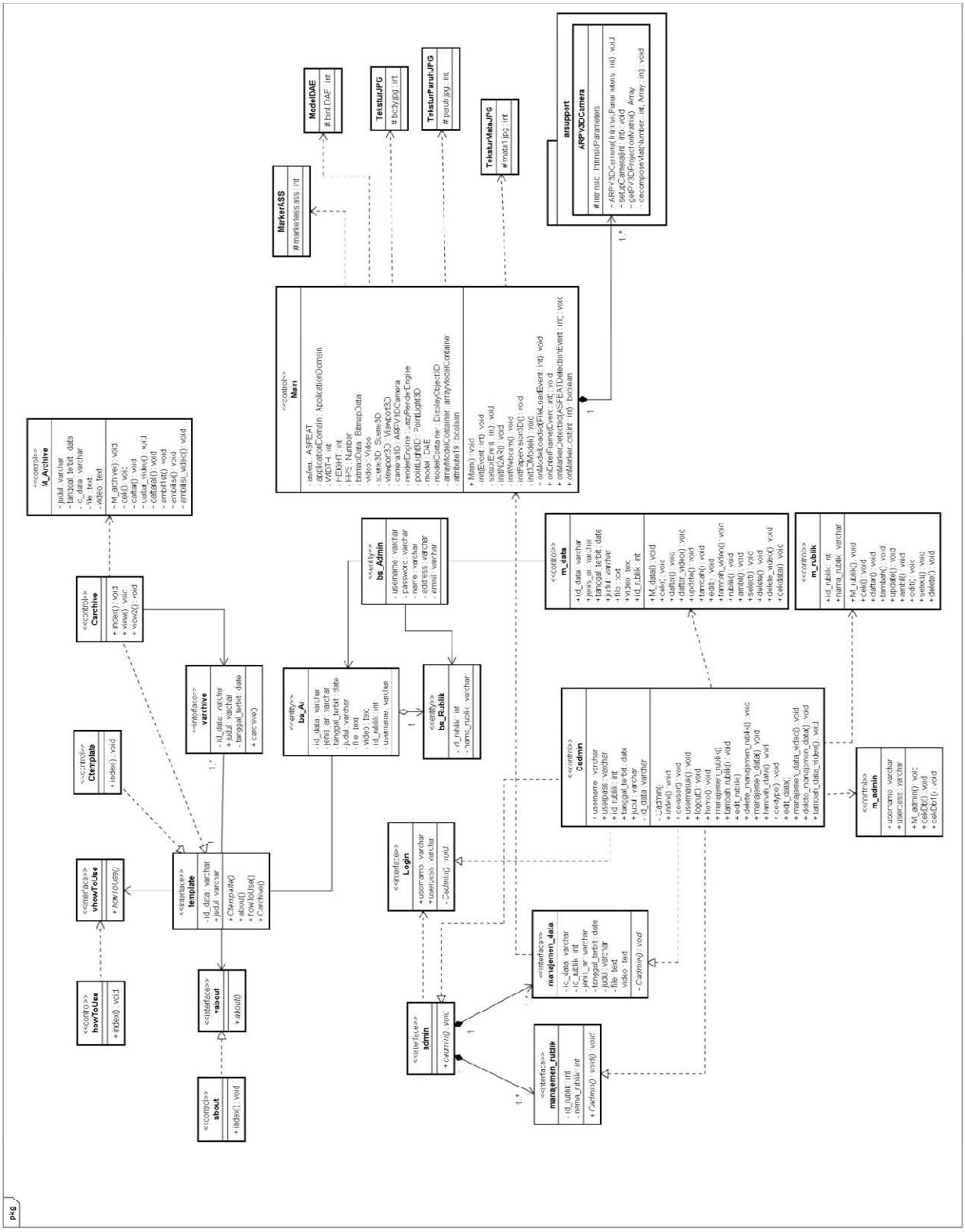
Identifikasi	
Nama Use Case	*Hapus* Data Video
Aktor	*User backend*
Tujuan	Menghapus data ar video
Keadaan Awal	Sistem menampilkan halaman pengolahan data
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih gambar silang pada kolom aksi tabel Data	
	2. Menampilkan pesan konfirmasi "Apakah Anda Yakin Menghapus Data <type_ar> dengan id <id_data>*".*
3. Menyetujui penghapusan dengan memilih atau menekan tombol "OK"	
	4. Menghapus data ar.
	5. Menyimpan hasil perubahan data ar
Kondisi Akhir	Perubahan data ar sesuai dengan data yang dihapus.

2. *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 sistem aplikasi *ARNewspaper*.

2.1 *Class Diagram ARNewspaper*



Gambar 3. 26 *Class Diagram ARNewspaper*

Tabel 3. 24 Deskripsi *Class Diagram ARNewspaper*

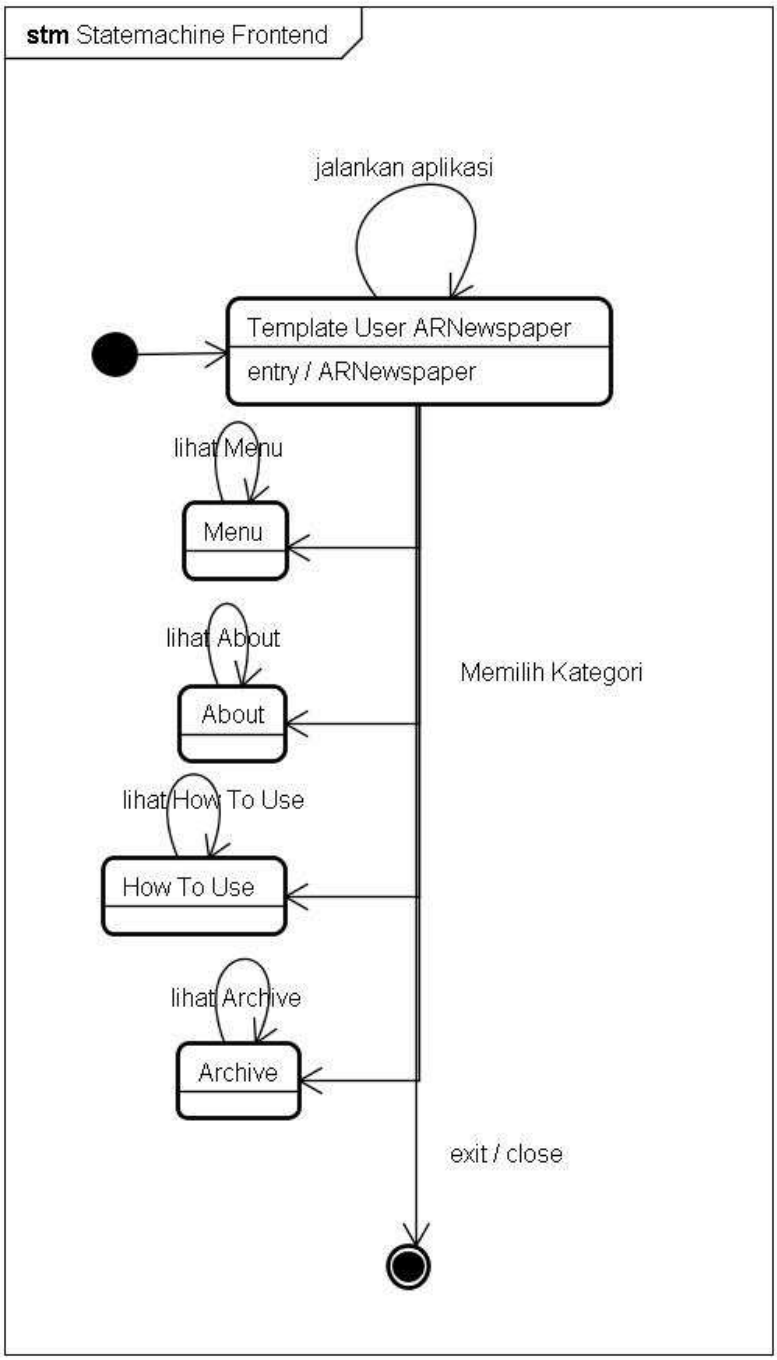
Backend Application		
Class	**Jenis Kelas**	**Deskripsi**
Manajemen_rublik	*Interface*	*Class* yang berisikan tampilan data rublik
Manajemen_data	*Interface*	*Class* yang berisikan tampilan data ar
Admin	*Interface*	Class yang berisikan *method-method* untuk fungsionalitas tampilan utama aplikasi yaitu menampilkan Home
Login	*Interface*	*Class* yang berisikan tampilan *login web*
Cadmin	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface*
M_admin	*Control*	*Class* yang berisikan *method* untuk mengontrol *class control* Cadmin
M_rublik	*Control*	*Class* yang berisikan *method* untuk mengontrol *class control* Cadmin
M_data	*Control*	*Class* yang berisikan *method* untuk mengontrol *class control* Cadmin
Main	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class* paket fileAR
ARPV3DCamera	*Control*	Class yang mengaktifkan fungsionalitas kamera pada *webcam* untuk ditampilkan.
MarkerASS	*Control*	Class yang menginisialisasikan fungsionalitas marker untuk ditampilkan.
ModelDAE	*Control*	Class yang menginisialisasikan fungsionalitas model bentuk untuk ditampilkan.
TeksturJPG	*Control*	Class yang menginisialisasikan fungsionalitas tekstur pada tubuh objek untuk ditampilkan.
TeksturParuhJPG	*Control*	Class yang menginisialisasikan fungsionalitas tekstur pada paruh objek untuk ditampilkan.
TeksturMataJPG	*Control*	Class yang menginisialisasikan fungsionalitas tekstur pada mata objek untuk ditampilkan.
bs_Admin	*Entity*	Class yang berisikan data admin
bs_AR	*Entity*	Class yang berisikan data ar
bs_Rublik	*Entity*	Class yang berisikan data rublik
Frontend Application		

Class	Jenis Class	Deskripsi
Template	*Interface*	Class yang berisikan *method-method* untuk fungsionalitas tampilan utama aplikasi yaitu menampilkan Home
vabout	*Interface*	Class yang berisikan *method* untuk fungsionalitas tampilan halaman *about*
vhowToUse	*Interface*	Class yang berisikan *method* untuk fungsionalitas tampilan halaman *how to use*
varchive	*Interface*	Class yang berisikan *method* untuk fungsionalitas tampilan halaman *archive*
Ctemplate	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface template*
about	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface about*
howToUse	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface howToUse*
Carchive	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface varchive*
M_archive	*Control*	*Class* yang berisikan *method* untuk mengontrol *class control* Archive

3. *State machine Digram*

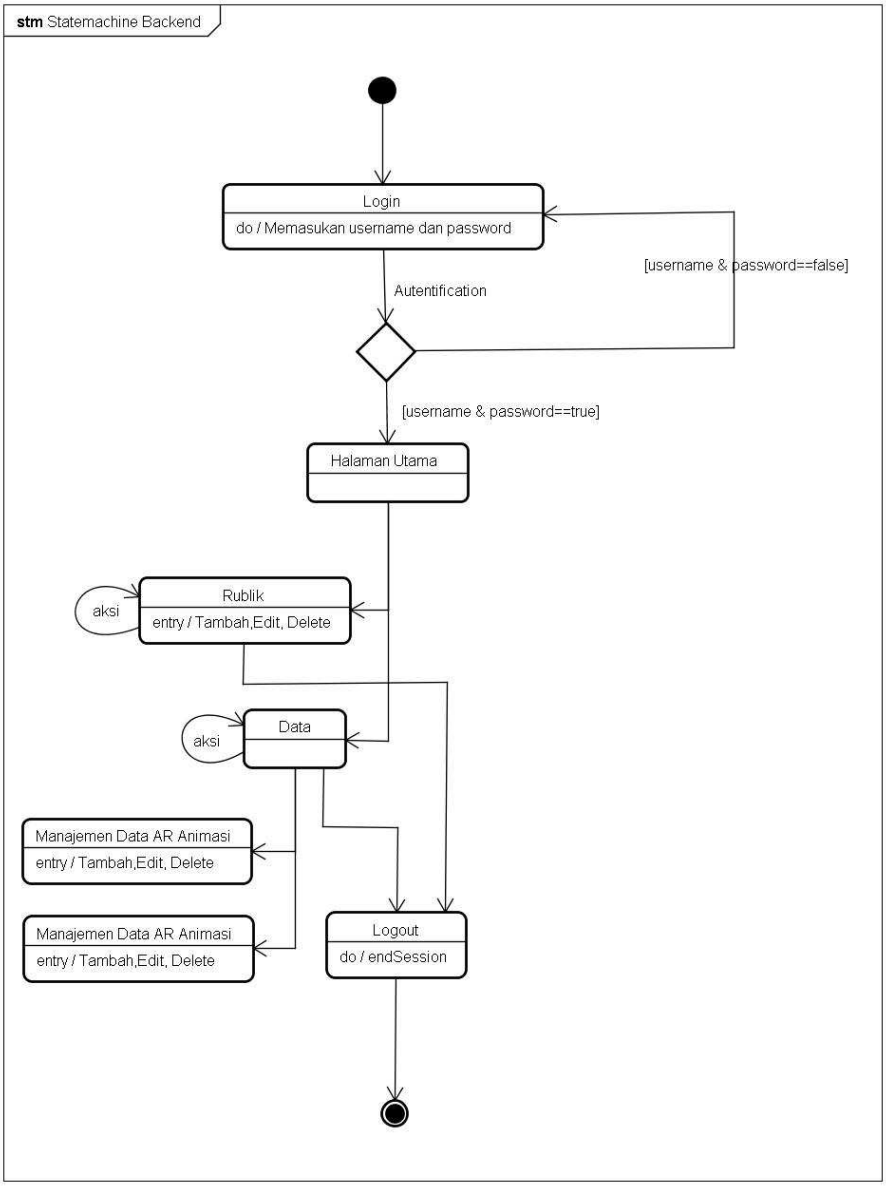
State machine diagram mengekspresikan perilaku sebagai sebuah progresif (perkembangan) melalui sekumpulan *state*, *triggered by events*, dan *action* yang mungkin terjadi. *State machine diagram* juga biasa dikenal sebagai *behavioural state diagram. State machine diagram* biasanya digunakan untuk mendeskripsikan perilaku sebuah *object*. Namun, juga dapat digunakan pada elemen yang besar dari sistem apapun.

3.1 *State Machine Diagram* Aplikasi *Frontend*



Gambar 3. 27 *State Machine* Diagram Aplikasi *Frontend*

3.2 *Statemachine Diagram* Aplikasi *Backend*

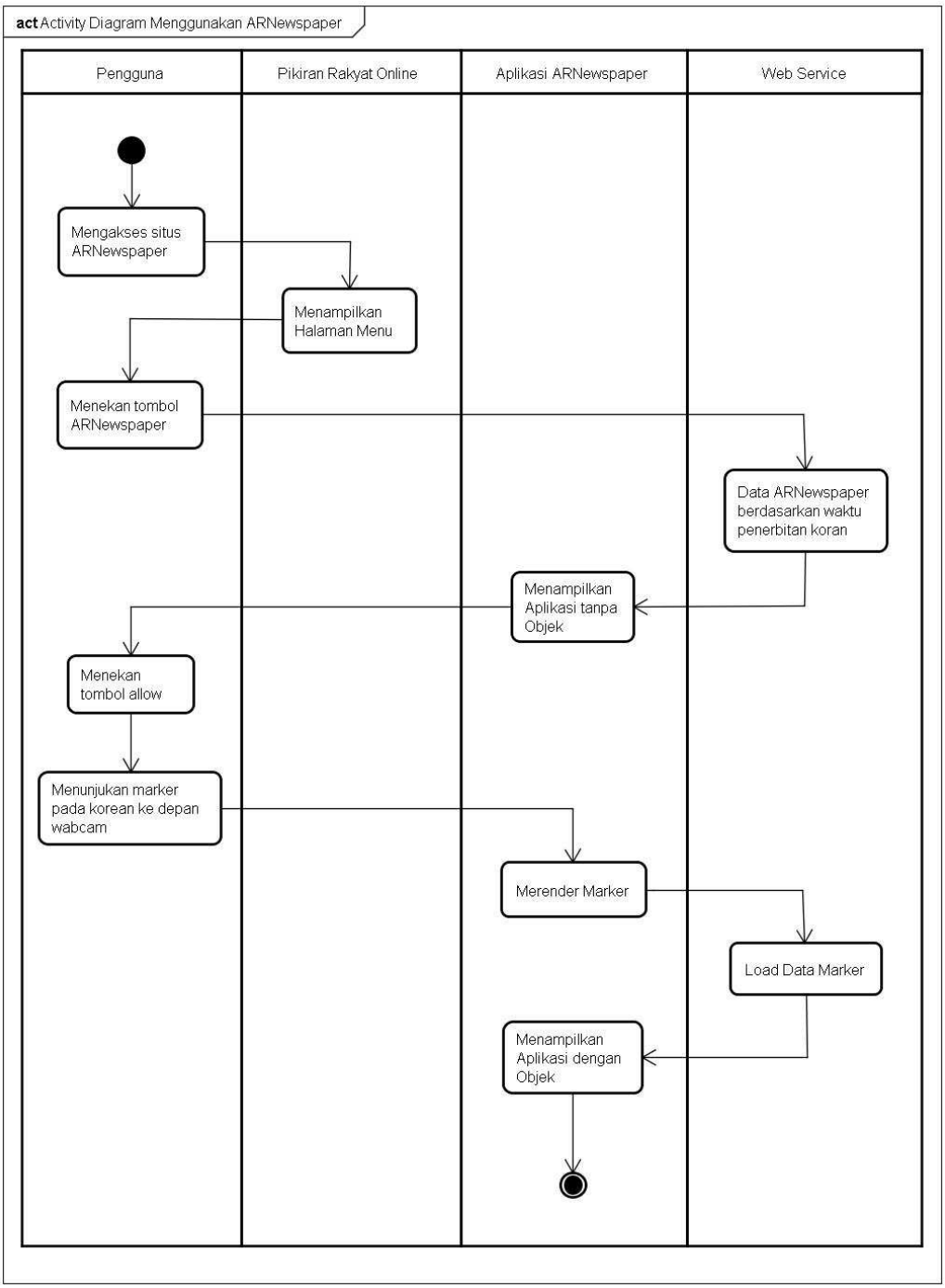


Gambar 3. 28 *State Machine* Diagram Aplikasi *Backend*

4. *Activity Diagram*

Diagram activity menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing alir 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 pemodelan aspek dinamis dari sistem.*

4.1 *Activity Diagram* Menggunakan *ARNewspaper*

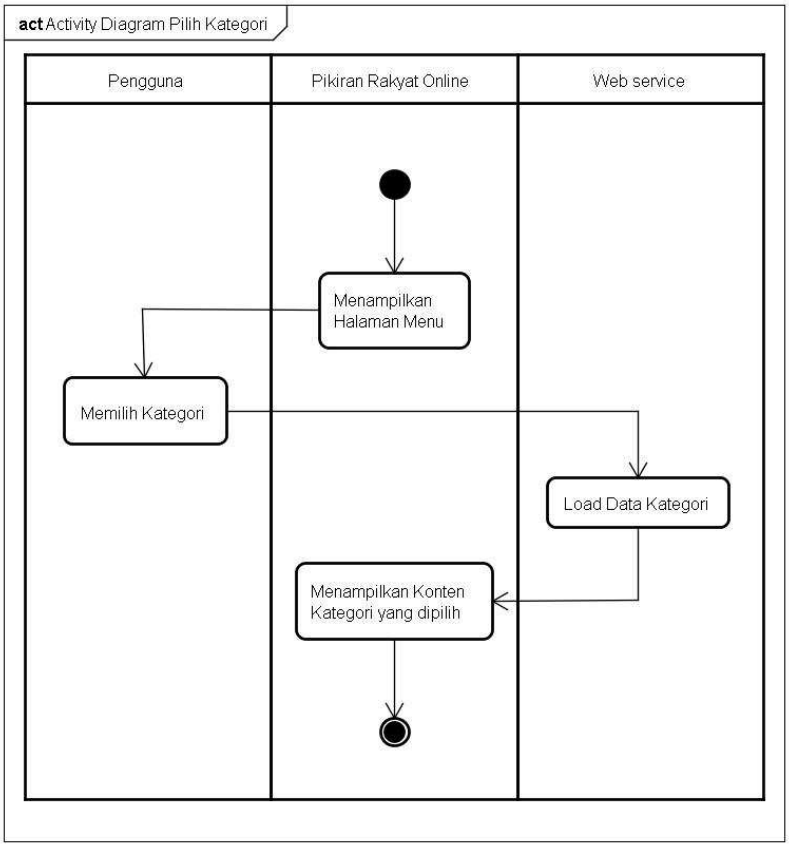


Gambar 3. 29 *Activity Diagram* Menggunakan *ARNewspaper*

Gambar 3.29**,** menggambarkan aktivitas yang terjadi pada aplikasi saat *user frontend* pertama kali menggunakan aplikasi. Dimana pada saat

penggunaan menjalankan aplikasi *user frontend* harus menunjukan gambar koran yang sudah diimplementasikan metode markerless ke depan wabcam agar objek yang ada pada gambar tersebut akan muncul.

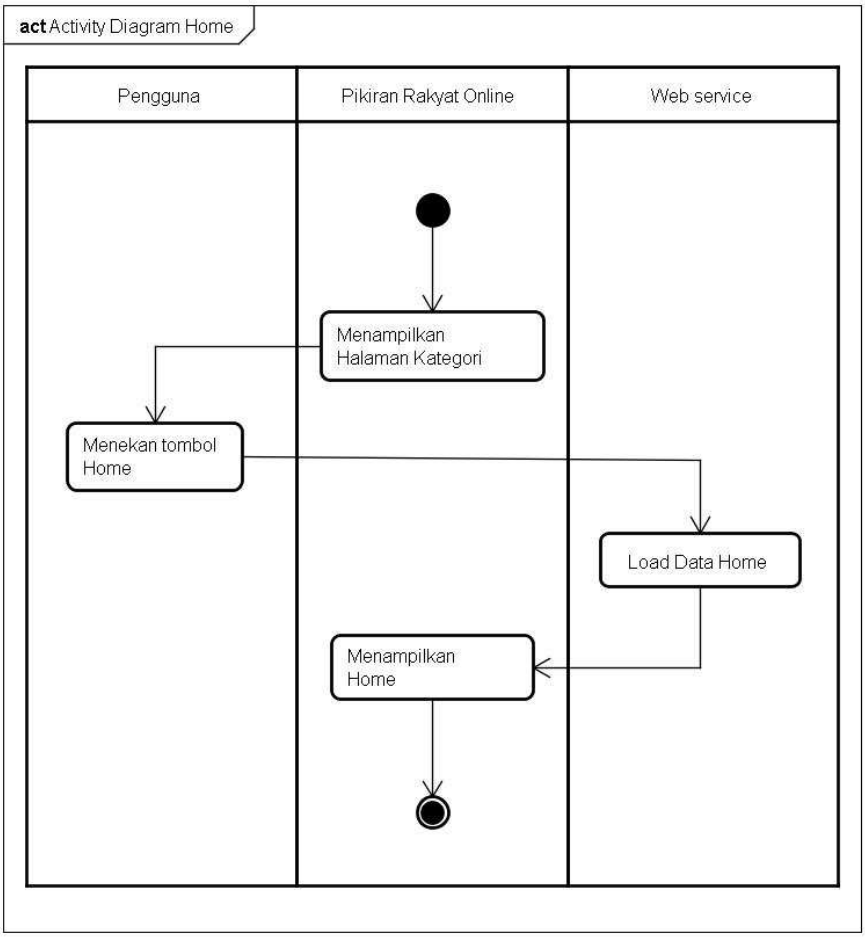
4.2 *Activity Diagram* Pilih Kategori



Gambar 3. 30 *Activity* Diagram Pilih Rublik

Gambar 3.30**,** menggambarkan aktivitas ketika *user frontend* memilih rublik. *User frontend* memilih salah satu rublik, selanjutnya sistem akan me-*load* data pada server (*web service*) untuk menampilkan konten yang sesuai dengan jenis rublik yang dipilih.

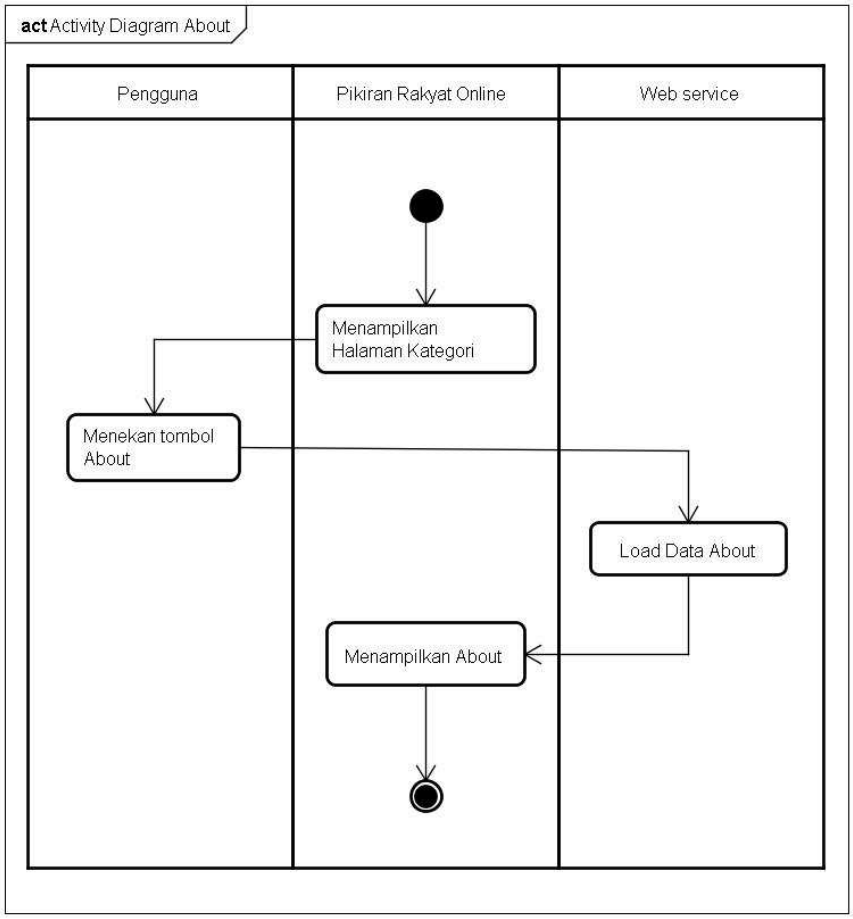
4.3 *Activity Diagram* Home



Gambar 3. 31 *Activity Diagram* Home

Gambar 3.31**,** menggambarkan aktivitas ketika *user frontend* memilih home. *User frontend* menekan tombol home, selanjutnya sistem akan me-*load* data home pada server (*web service*) untuk menampilkan rublik home. Pada rublik home ini menjadi halaman utama pada saat membuka situs Pikiran Rakyat Online. Home ini berisi tentang applikasi ARNewspaper yang bisa langsung digunakan sesuai dengan hari pada saat situs dibuka.

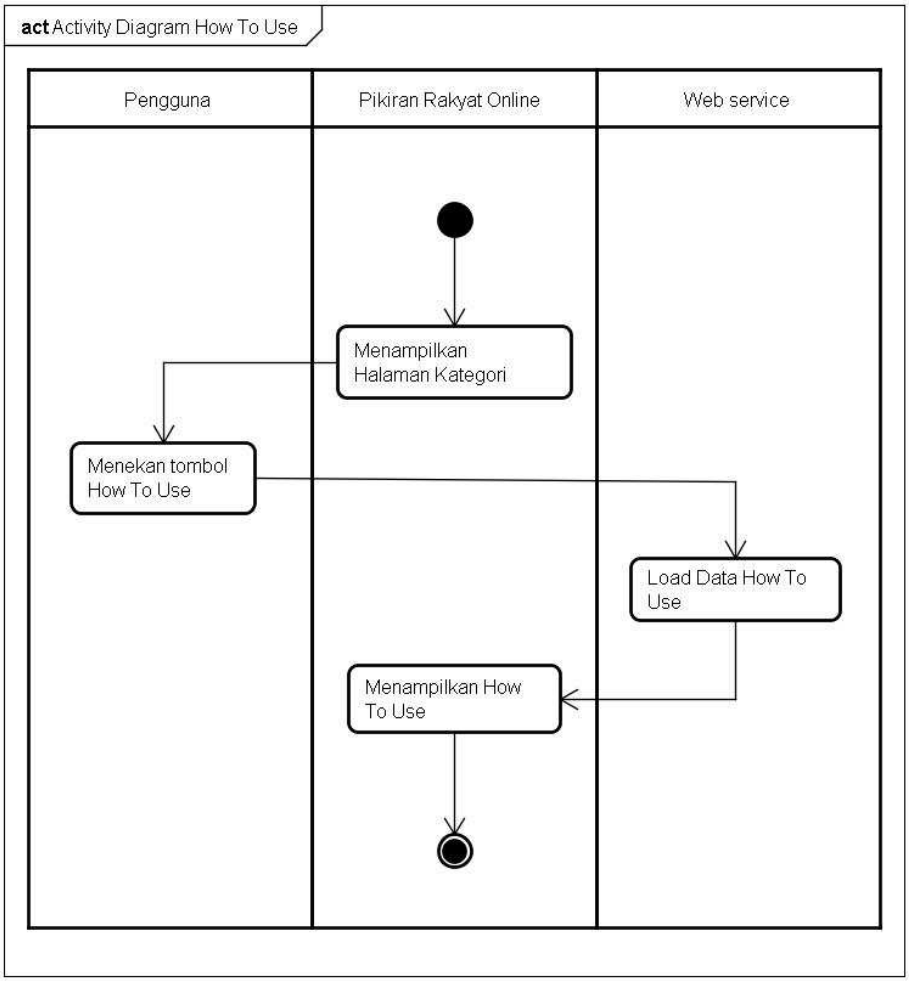
4.4 *Activity Diagram* Lihat *About*



Gambar 3. 32 *Activity Diagram* Lihat *About*

Gambar 3. 32, menggambarkan aktivitas ketika *user frontend* memilih about. *User frontend* menekan tombol *about*, selanjutnya sistem akan me-*load* data about pada server (*web service*) untuk menampilkan rublik about. Pada rublik about berisikan halaman informasi ringkas tentang aplikasi dan teknolologi *augmented reality* yang digunakan.

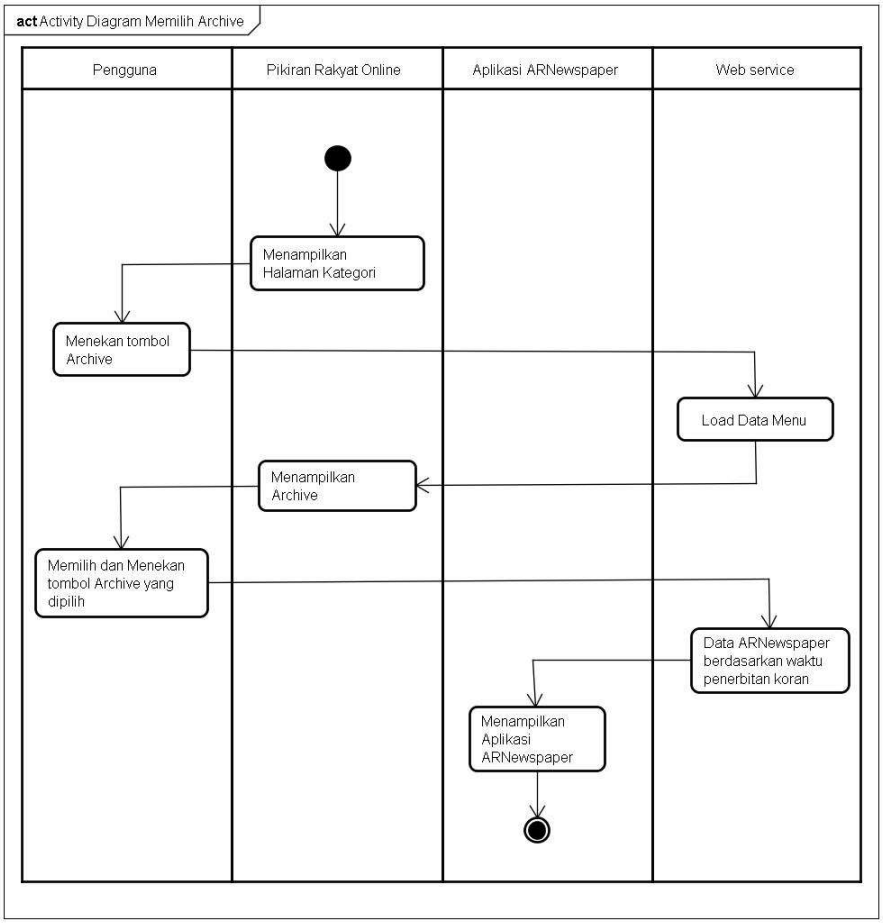
4.5 *Activity Diagram* Lihat *How To Use*



Gambar 3. 33 *Activity Diagram* Lihat *How To Use*

Gambar 3. 33, menggambarkan aktivitas ketika *user frontend* memilih *how to use*. *User frontend* menekan tombol *how to use*, selanjutnya sistem akan me-*load* data about pada server (*web service*) untuk menampilkan rublik *How To Use*. Pada halaman ini menjelaskan tentang tata cara penggunaan aplikasi secara baik dan benar.

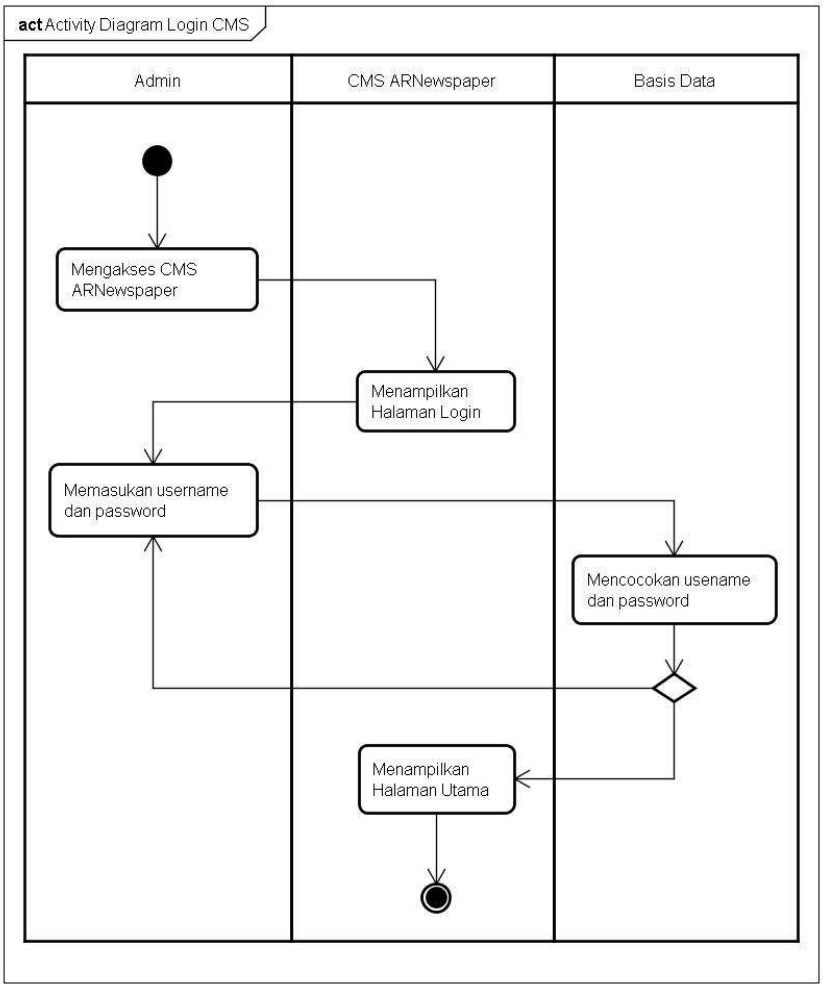
4.6 *Activity Diagram* Memilih Archive



Gambar 3. 34 *Activity Diagram* Memilih *Archive*

Gambar 3. 34, menggambarkan aktivitas ketika *user frontend* memilih *archive*. *User frontend* menekan tombol *archive*, selanjutnya sistem akan me-*load* data *archive* pada server (*web service*) untuk menampilkan rublik *Archive*, kemudian *user frontend* dapat memilih *archive* sesuai dengan tanggal terbitnya koran yang dimiliki. Setelah proses pemilihan *archive* dengan menekan tombol *archive* yang dipilih kemudia sistem akan me-*load* data untuk di tampilkannya aplikasi *ARNewspaper.*

4.7 *Activity Diagram Login* CMS

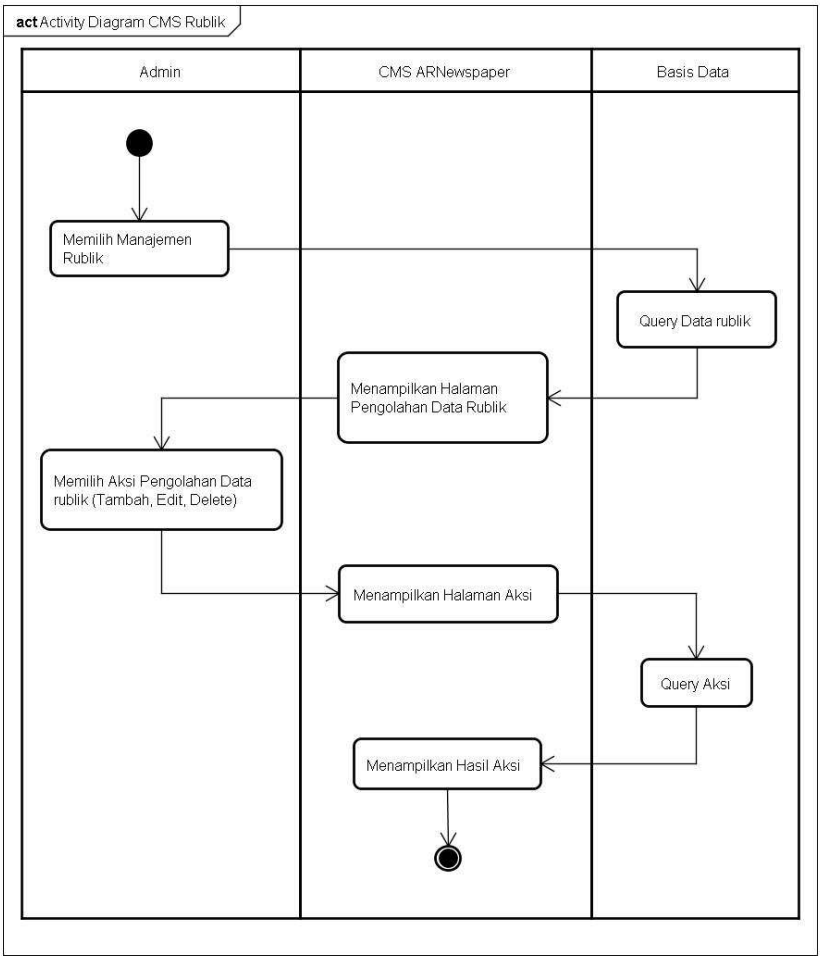


Gambar 3. 35 *Activity Diagram Login* CMS

Gambar 3.35, menggambarkan aktivitas login untuk masuk ke halaman utama CMS. *User backend* mengakses halaman CMS, kemudian CMS akan menampilkan halaman login, kemudian *user backend* melakukan input *username* dan *password*. Server akan melakukan verifikasi, jika *username* dan *password* tidak sesuai maka *user backend* akan diarahkan kembali ke halaman login untuk melakukan input

username dan *password* yang valid. Jika *username* dan *password* yang di input valid, maka akan ditampilkan halaman utama web CMS.

4.8 *Activity Diagram* CMS Rublik

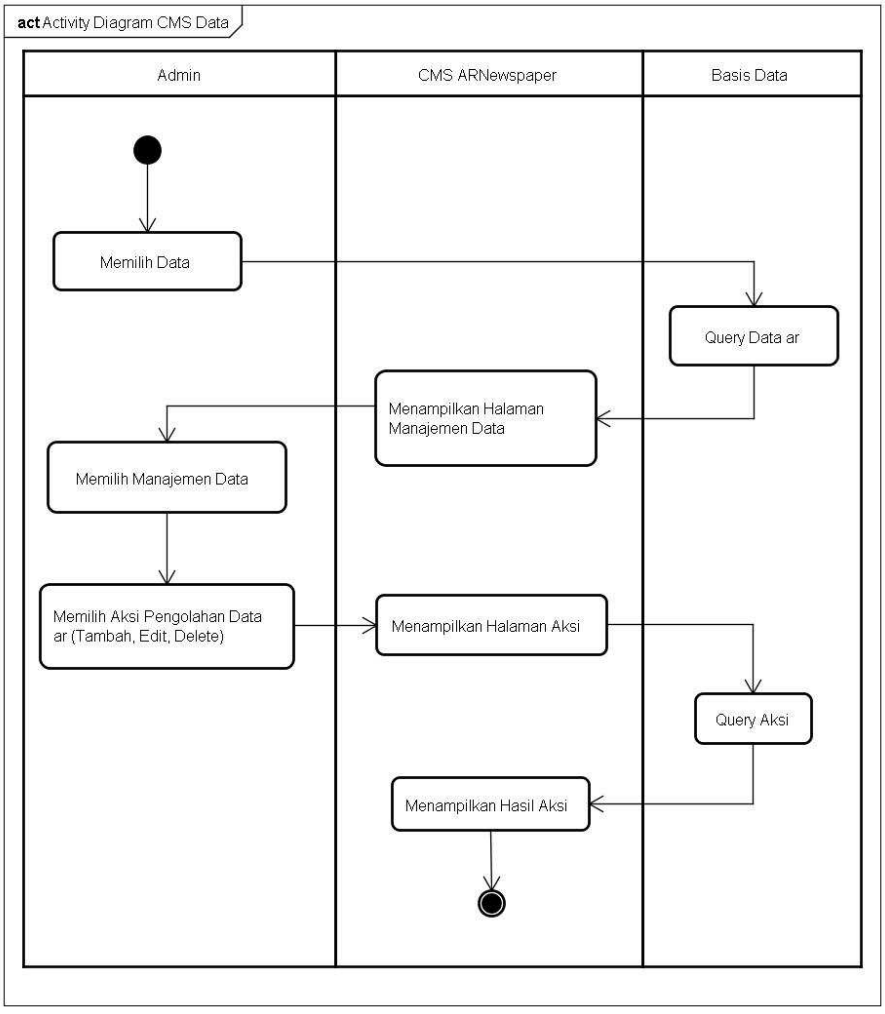


Gambar 3. 36 *Activity Diagram* CMS Rublik

Gambar 3. 36, menggambarkan aktivitas pengolahan data rublik. Proses ini berjalan setelah *user backend* berhasil melakukan login. *User backend* memilih menu rublik, selanjutnya CMS akan menampilkan halaman rublik. *User backend* dapat mengelola data rublik dengan

menggunakan aksi Tambah, Ubah, atau Hapus. CMS akan menampilkan halaman berdasarkan aksi yang dilakukan *user backend*, saat bersamaan *query* akan dipanggil dan dieksekusi pada database server kemudian hasilnya ditampilkan pada web CMS

4.9 *Activity Diagram* CMS Data



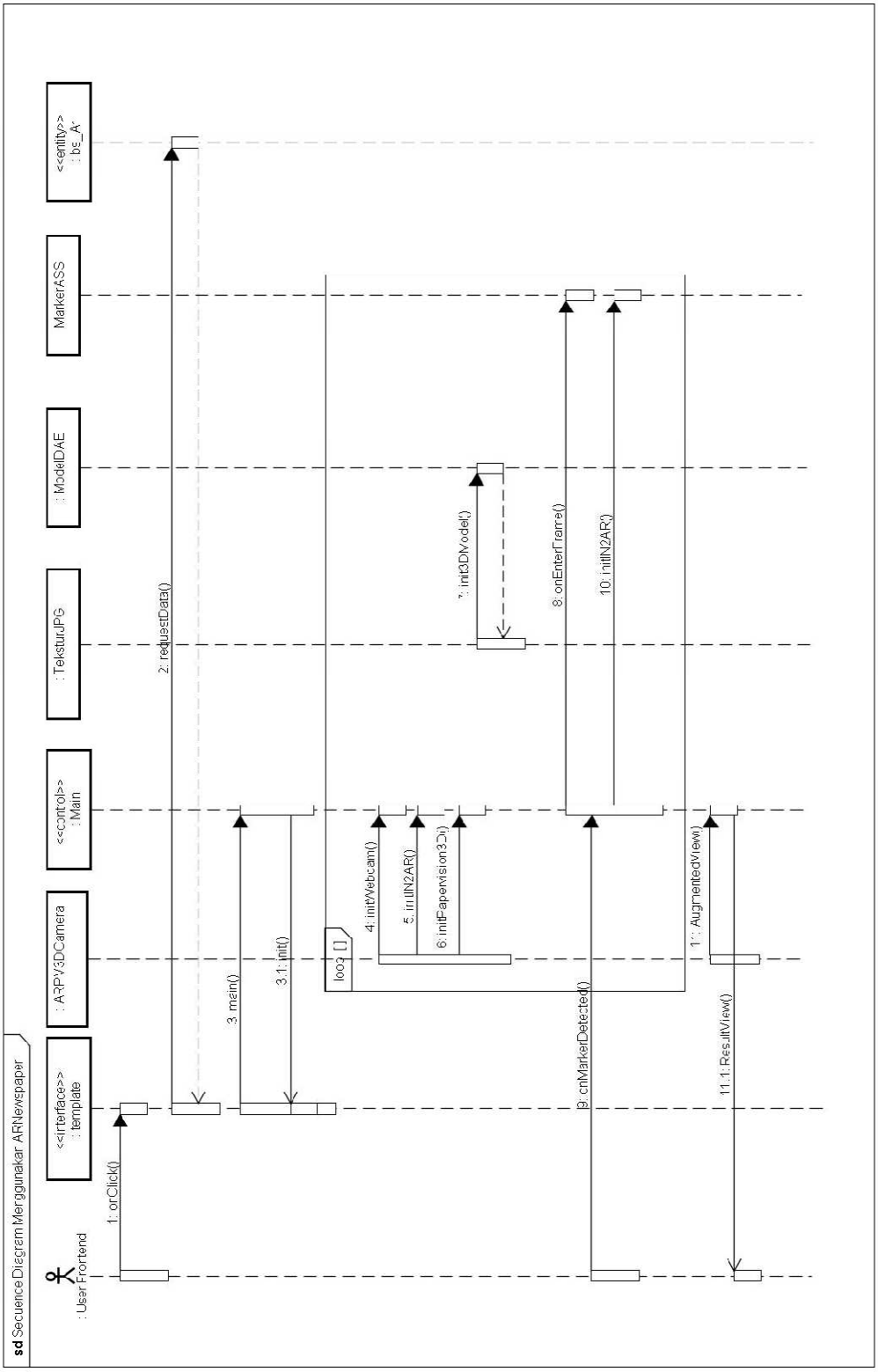
Gambar 3. 37 *Activity Diagram* CMS Data

Gambar 3. 37, menggambarkan aktivitas pengolahan data ar. Proses ini berjalan setelah *user backend* berhasil melakukan login. *User backend* memilih menu data, selanjutnya CMS akan menampilkan halaman data ar manajemen yang ada. Kemudian *User backend* memilih manajemen data dan *user backend* dapat mengelola data ar dengan menggunakan aksi Tambah, Ubah, atau Hapus. CMS akan menampilkan halaman berdasarkan aksi yang dilakukan *user backend*, saat bersamaan *query* akan dipanggil dan dieksekusi pada database server kemudian hasilnya ditampilkan pada web CMS.

5. *Sequence Diagram*

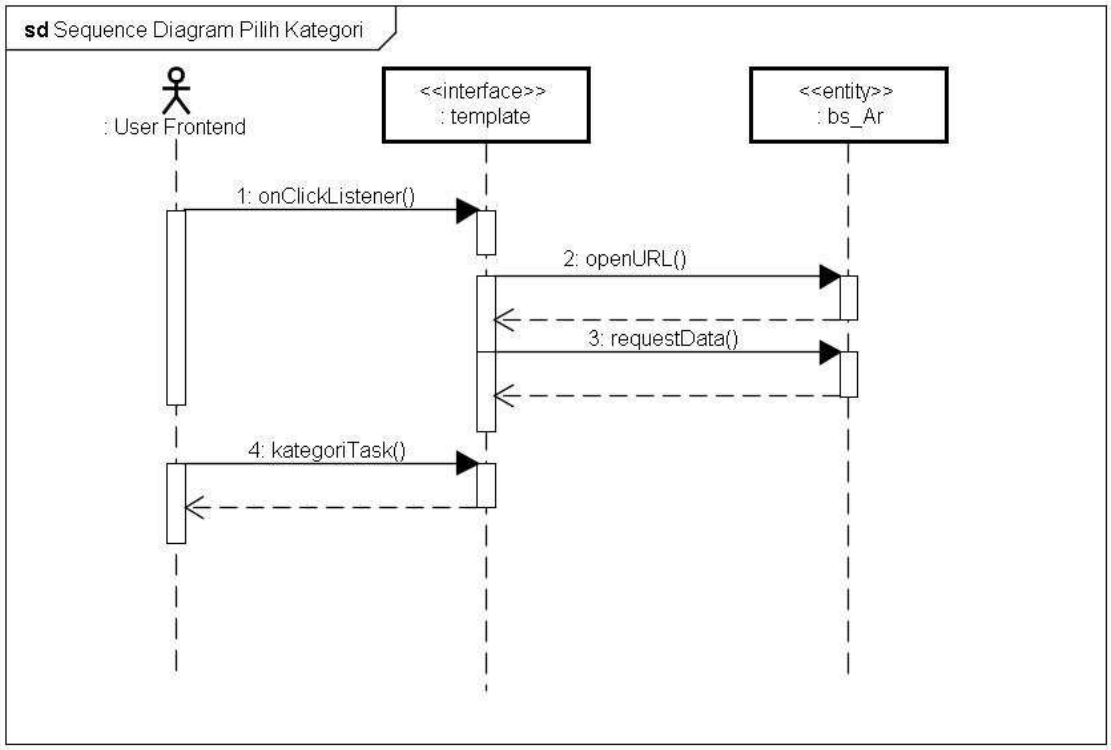
Sequence diagram menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam use case. Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progres vertikal.

5.1 *Sequence Diagram* Menggunakan *ARNewspaper*



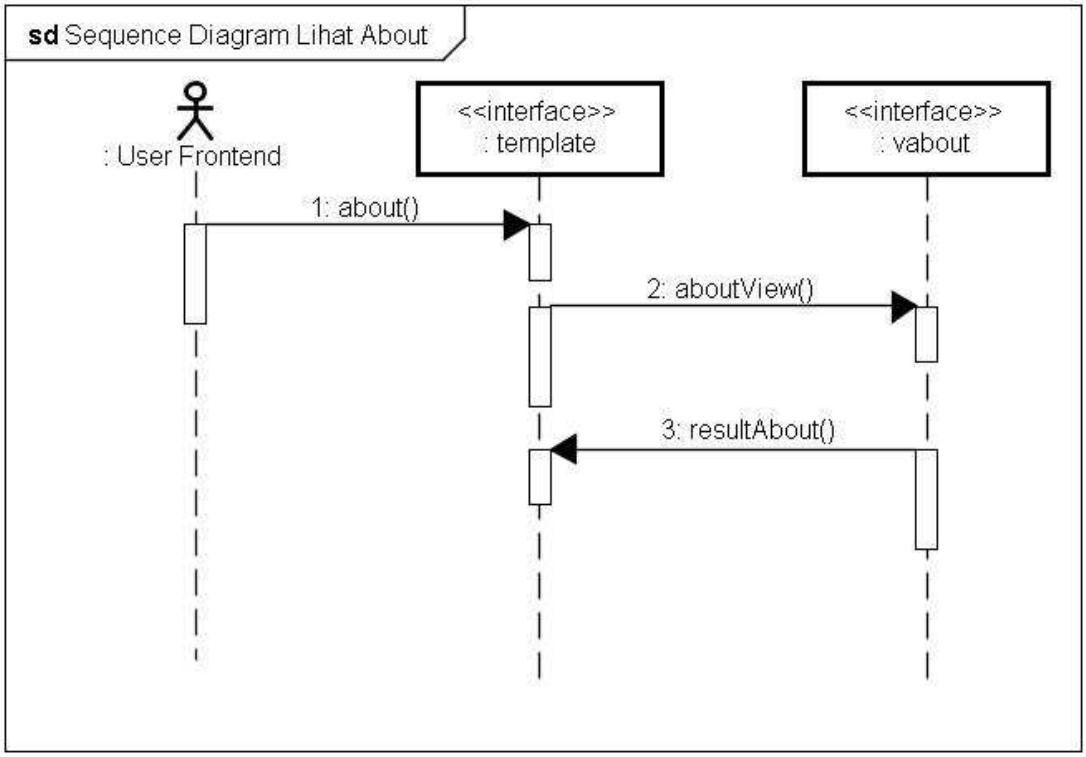
Gambar 3. 38 *Sequence Diagram* Menggunakan *ARNewspaper*

5.2 *Sequence Diagram* Pilih Kategori



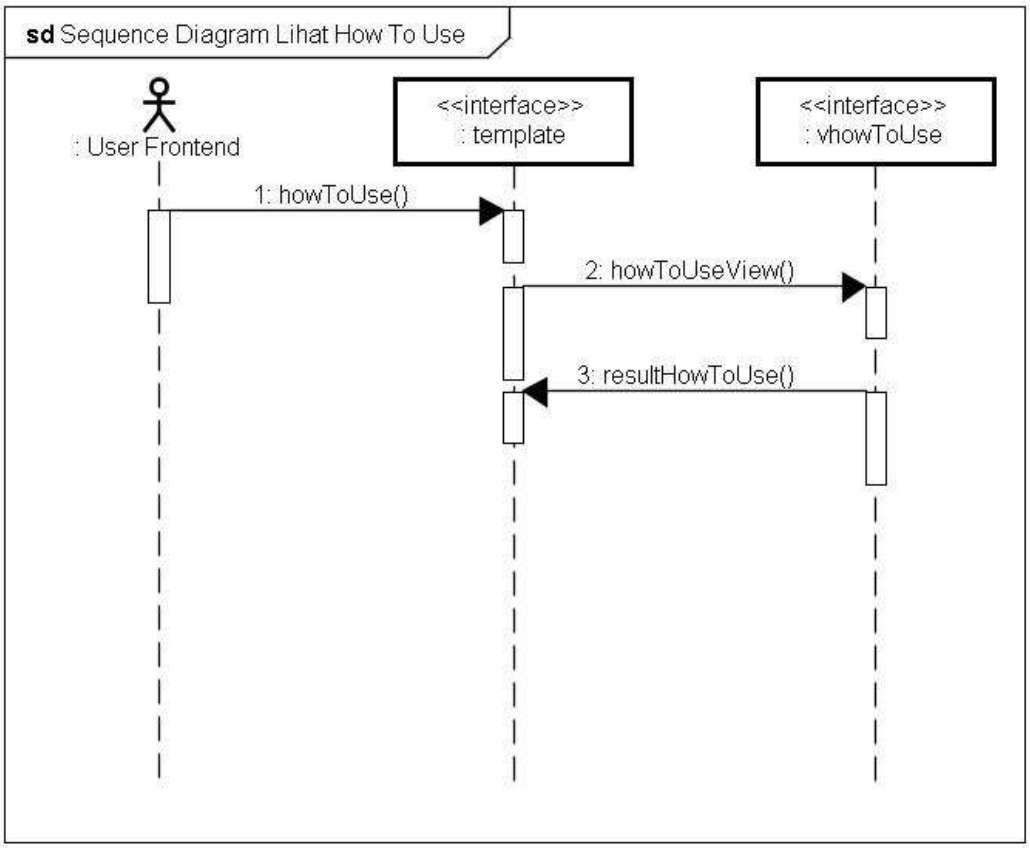
Gambar 3. 39 *Sequence Diagram* Pilih Kategori

5.3 *Sequence Diagram* Lihat *About*



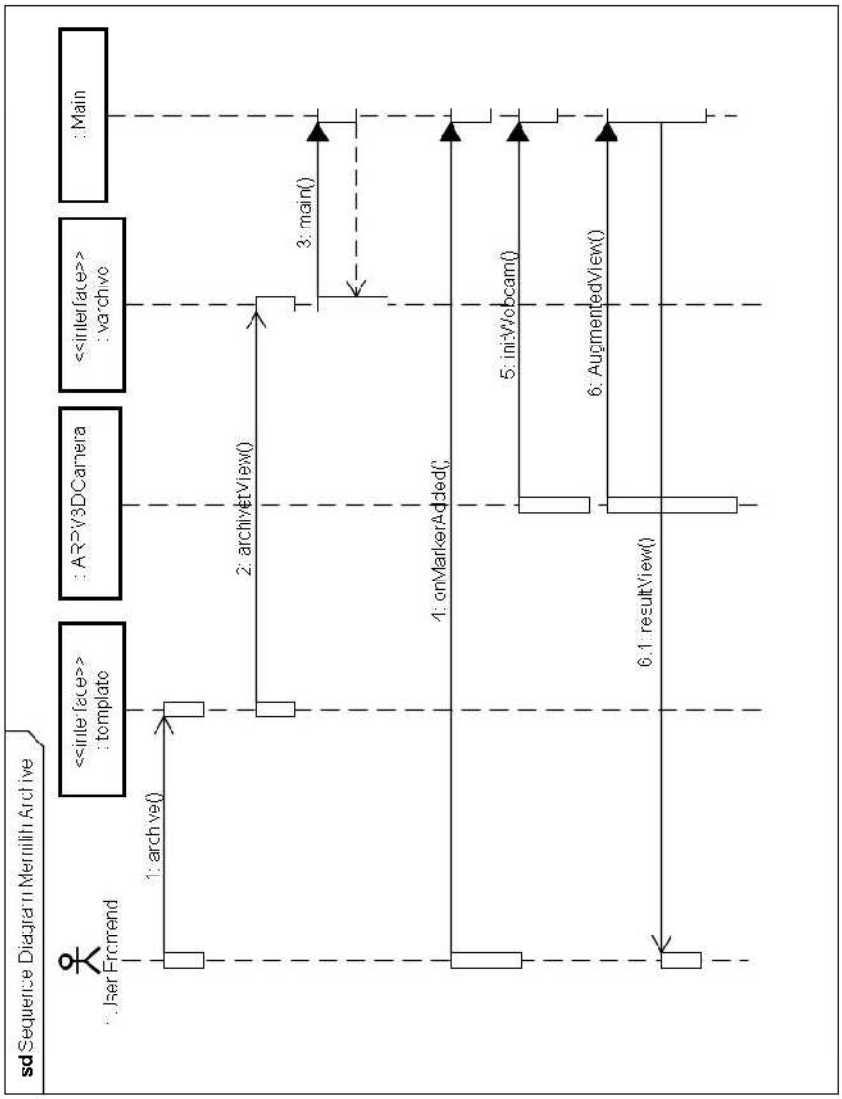
Gambar 3. 40 *Sequence Diagram* Lihat *About*

5.4 *Sequence Diagram* Lihat *How To Use*



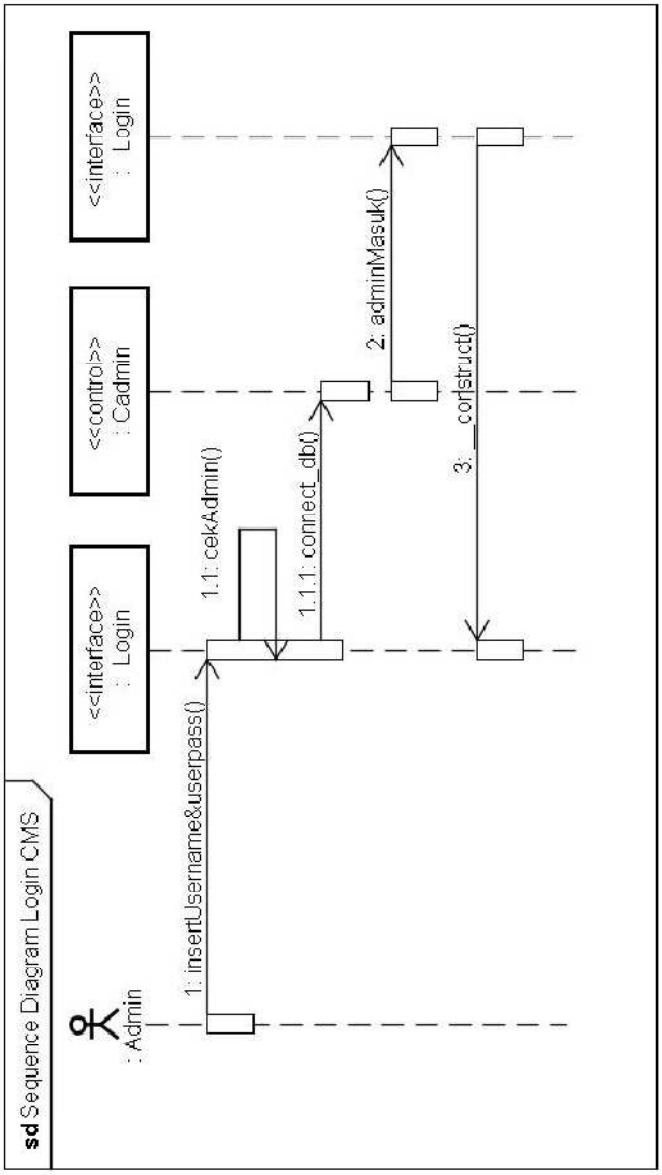
Gambar 3. 41 *Sequence Diagram* Lihat *How To Use*

5.5 *Sequence Diagram* Memilih Archive



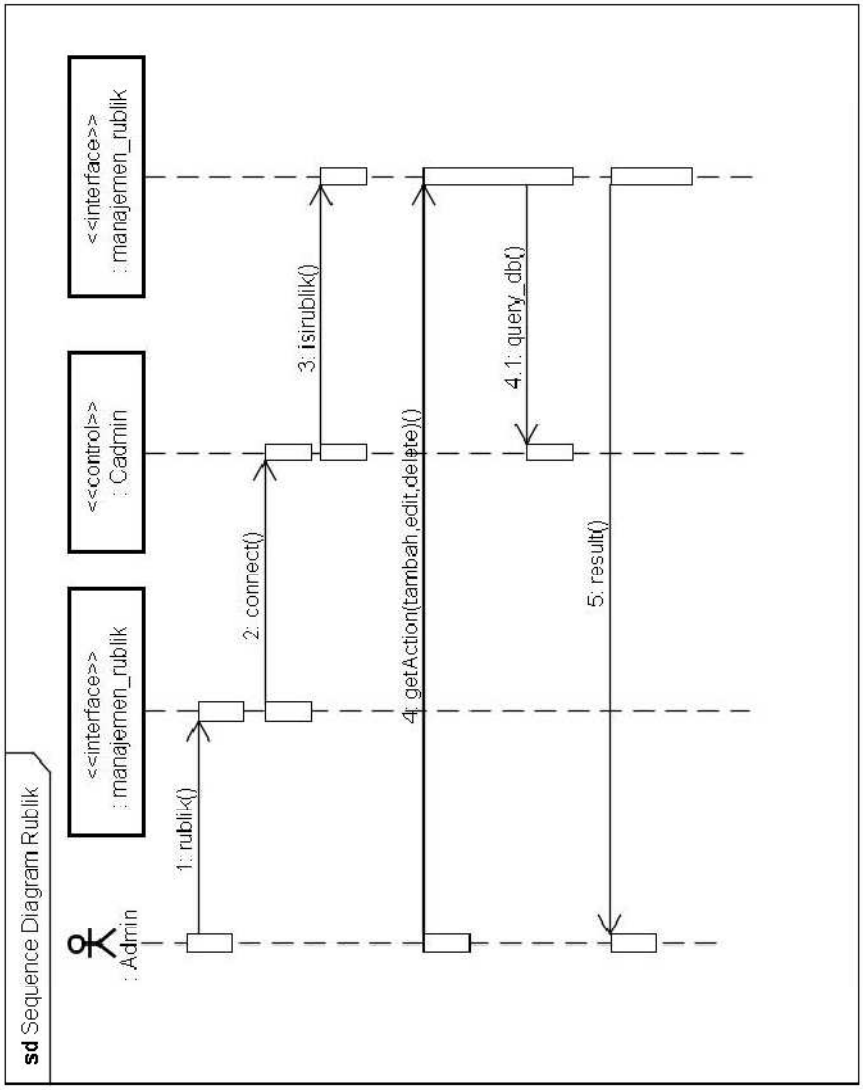
Gambar 3. 42 *Sequence Diagram* Memilih *Archive*

5.6 *Sequence Diagram* Login CMS



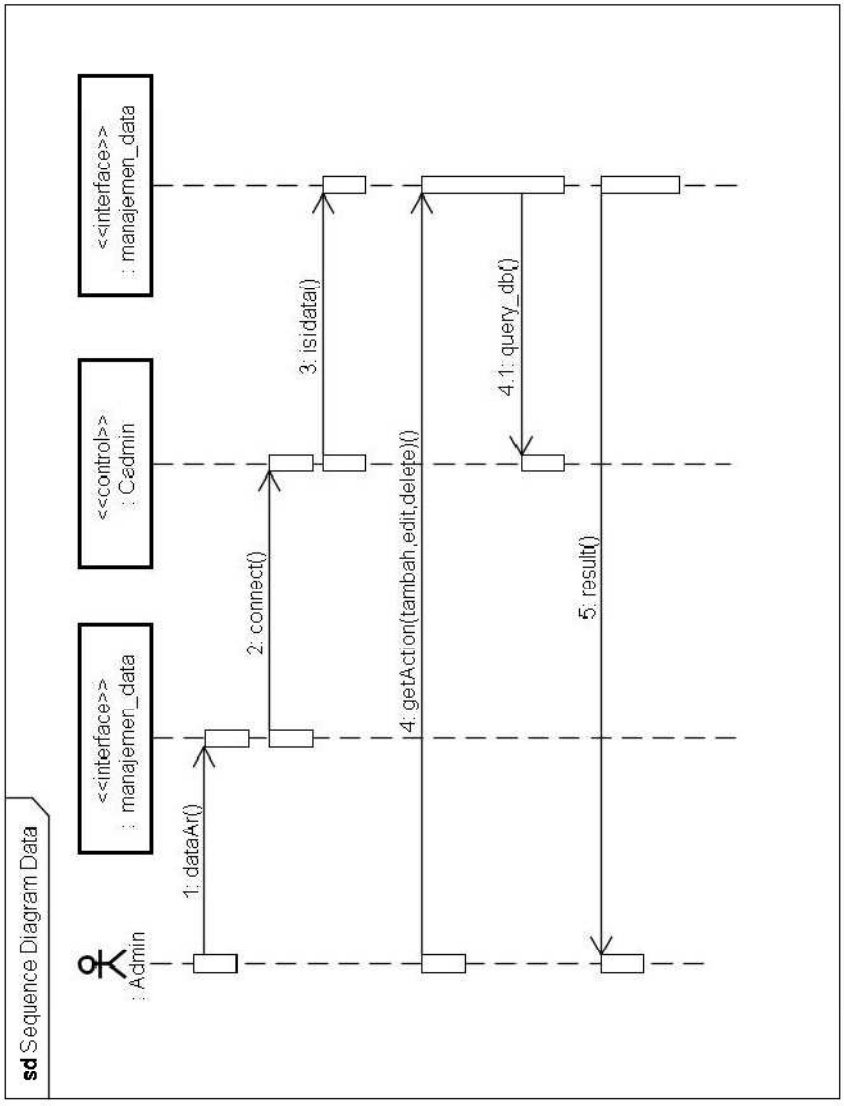
Gambar 3. 43 *Sequence Diagram Login* CMS

5.7 *Sequence Diagram* Rublik



Gambar 3. 44 *Sequence Diagram* Rublik

5.8 *Sequence Diagram* Data

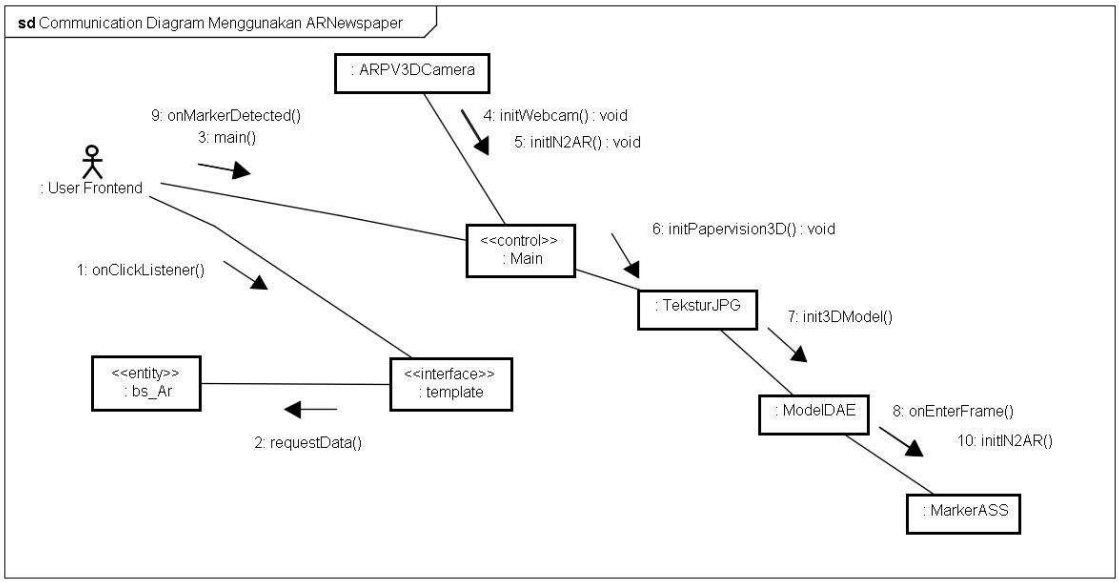


Gambar 3. 45 *Sequence Diagram* Data

6. *Communication Diagram*

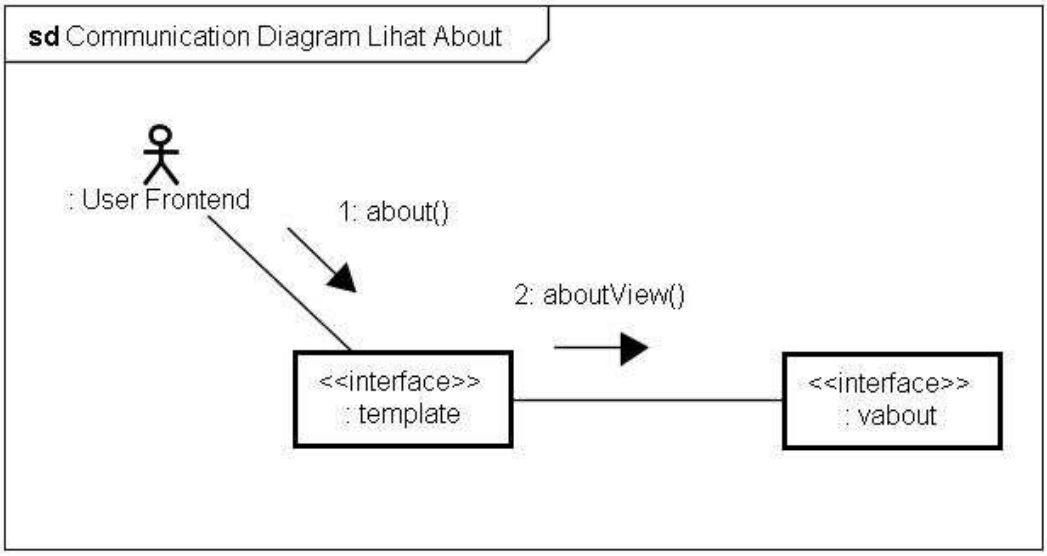
Diagram Komunikasi juga digunakan untuk memodelkan perilaku dinamis dari *use case*. Bila dibandingkan dengan Diagram Sequence, Diagram Komunikasi lebih terfokus pada menampilkan kolaborasi dari objek bukan urutan waktu. Commonication diagram merupakan pengembangan dari *collaboration diagram* pada UML versi sebelumnya.

6.1 *Communication Diagram* Menggunakan *ARNewspaper*.



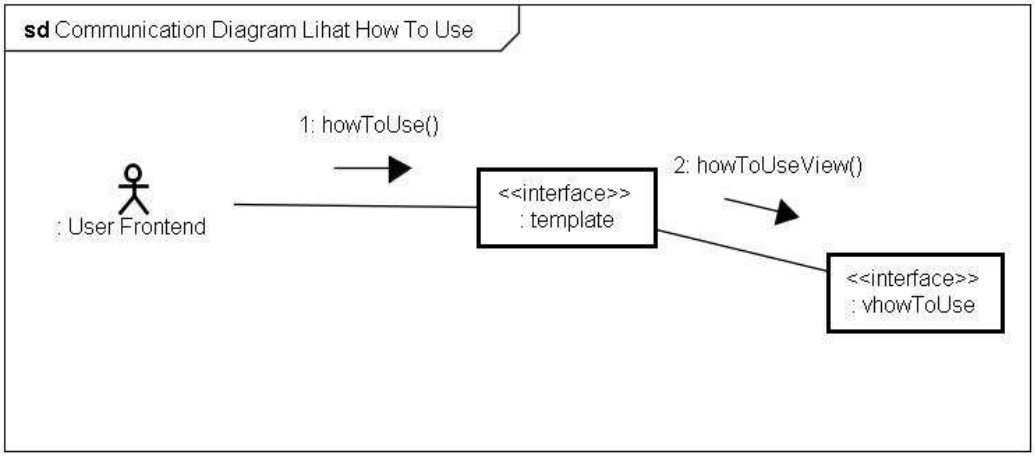
Gambar 3. 46 *Communication Diagram* Menggunakan *ARNewspaper*

6.2 *Communication Diagram* Lihat *About*



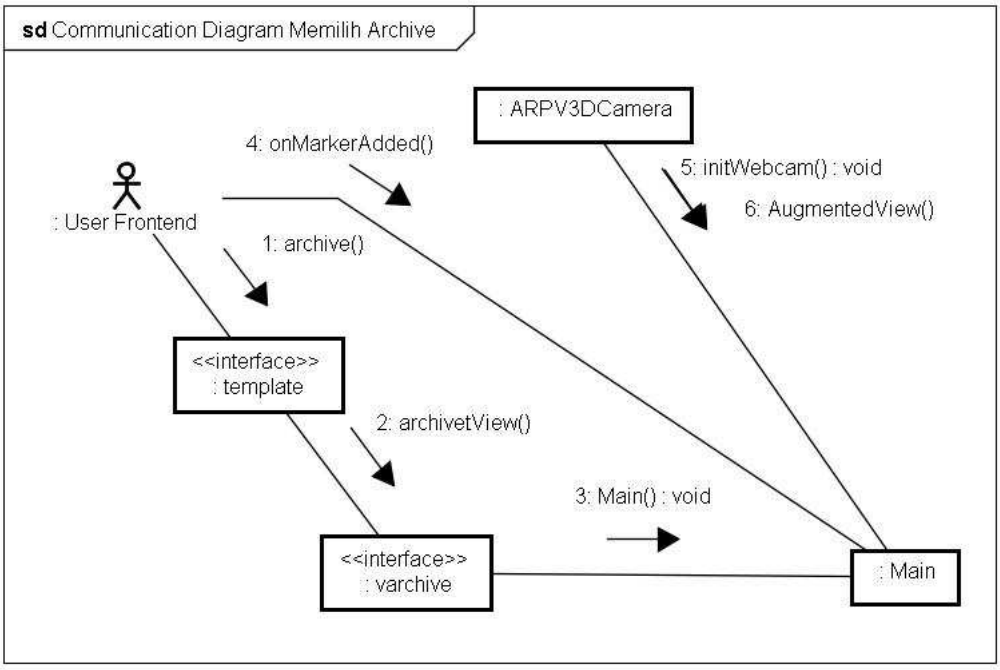
Gambar 3. 47 *Communication Diagram* Lihat *About*

6.3 *Communication Diagram* Lihat *How To Use*



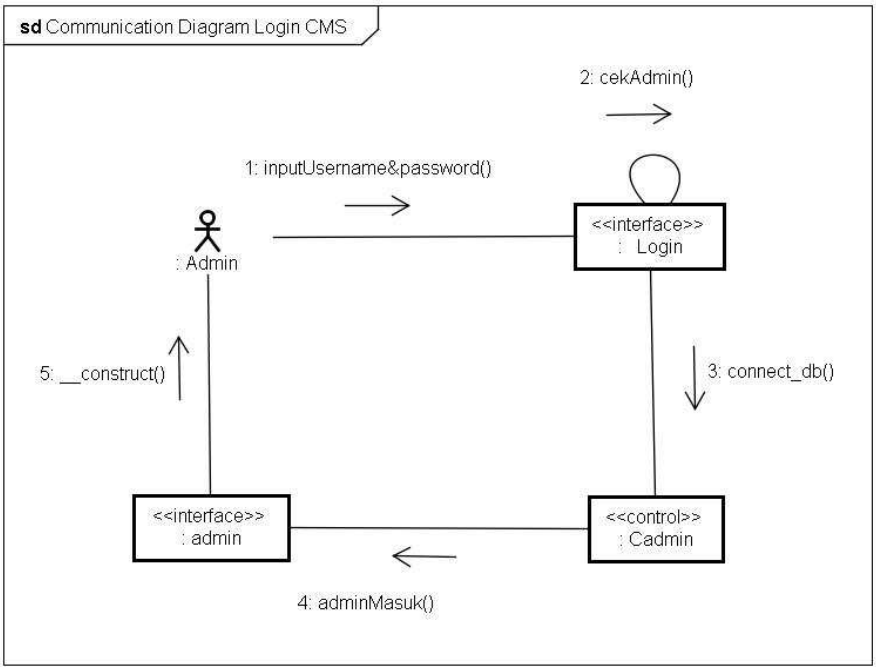
Gambar 3. 48 *Communication Diagram* Lihat *How To Use*

6.4 *Communication Diagram* Memilih *Archive*



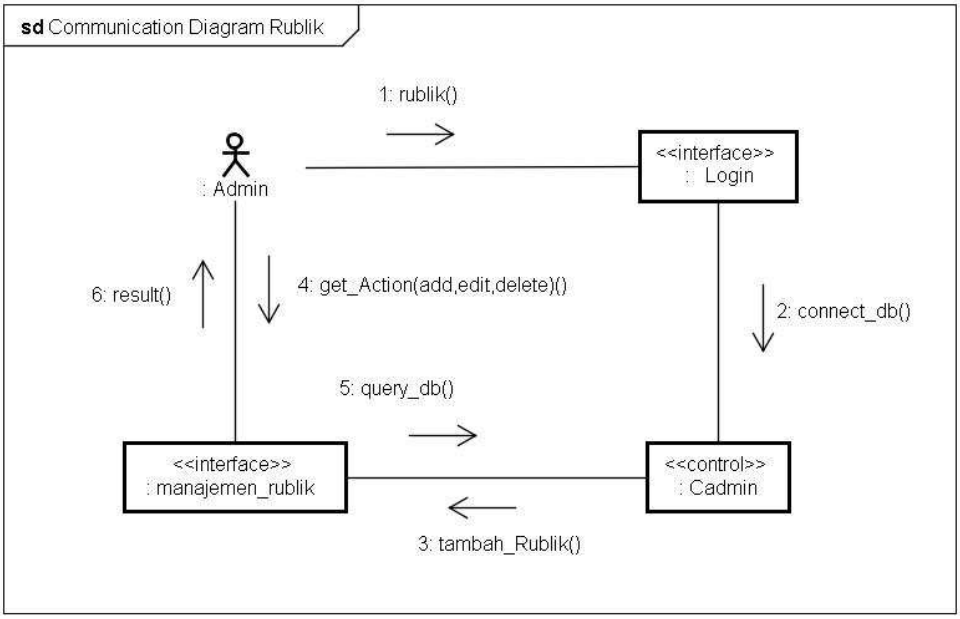
Gambar 3. 49 *Communication Diagram* Memilih *Archive*

6.5 *Communication Diagram* Login CMS



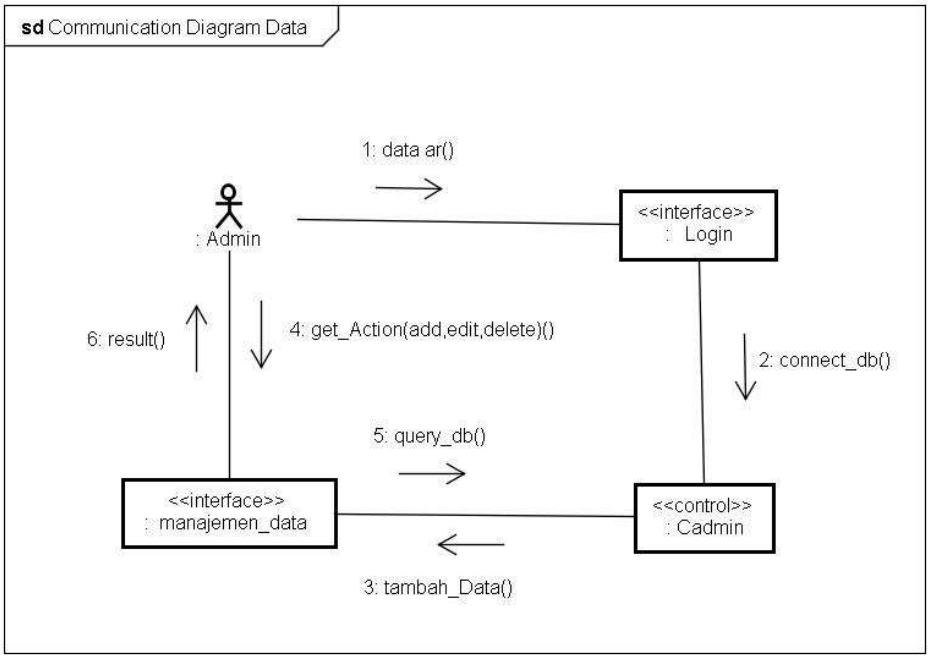
Gambar 3. 50 *Communication Diagram Login* CMS

6.6 *Communication Diagram* Rublik



Gambar 3. 51 *Communication Diagram* Rublik

6.7 *Communication Diagram* Data



Gambar 3. 52 *Communication* Diagram Data

3.2 Perancangan Sistem

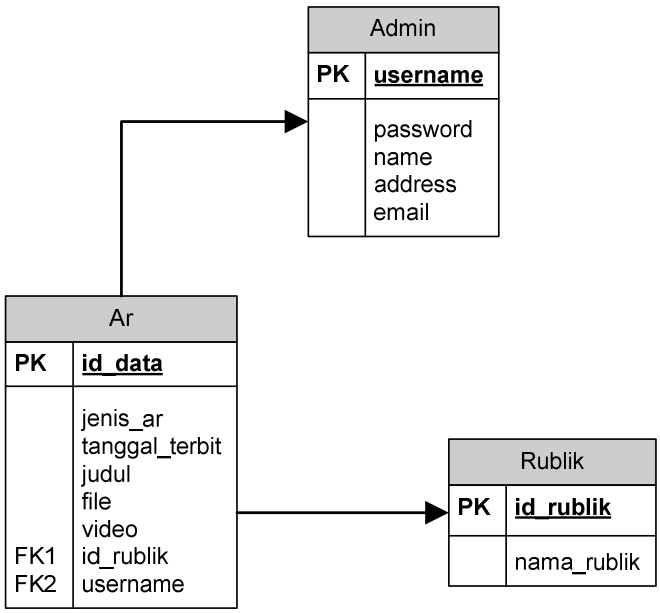
Perancangan akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.

3.2.1 Perancangan Basis Data

Perancangan basis data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Perancangan basis data terbagi menjadi dua yaitu skema relasi dan perancangan struktur data.

3.2.1.1 Skema Relasi

Skema relasi atau relasi antar tabel merupakan gabungan antar file yang mempunyai kunci utama yang sama, sehingga file-file tersebut menjadi satu kesatuan yang dihubungkan oleh field kunci (*primary key*). Pada proses ini elemen-elemen data dikelompokkan menjadi satu file database beserta entitas dan hubungan.



Gambar 3. 53 Skema Relasi

3.2.1.2 Struktur Tabel

1) Tabel Admin

Tabel 3. 25 Struktur Tabel Admin

Nama Field	Tipe Data	Panjang	Keterangan
username	Varchar	20	Primary Key
password	Varchar	50	
name	Varchar	50	
address	Varchar	50	
email	Varchar	40	

2) Tabel Rublik

Tabel 3. 26 Struktur Tabel Rublik

Nama Field	Tipe Data	Panjang	Keterangan
id_rublik	Integer	auto increment	Primary Key
nama_rublik	Varchar	50	

3) Tabel Ar

Tabel 3. 27 Struktur Tabel Ar

Nama Field	Tipe Data	Panjang	Keterangan
Id_data	Varchar	30	Primary Key
jenis_ar	Varchar	50	
tanggal_terbit	Date		
judul	Varchar	50	
file	Text		
video	Text		
id_rublik	Integer	auto increment	Foreign Key, reference(T.rublik(id_rublik))
username	Varchar	20	Foreign Key, reference(T.admin(username))

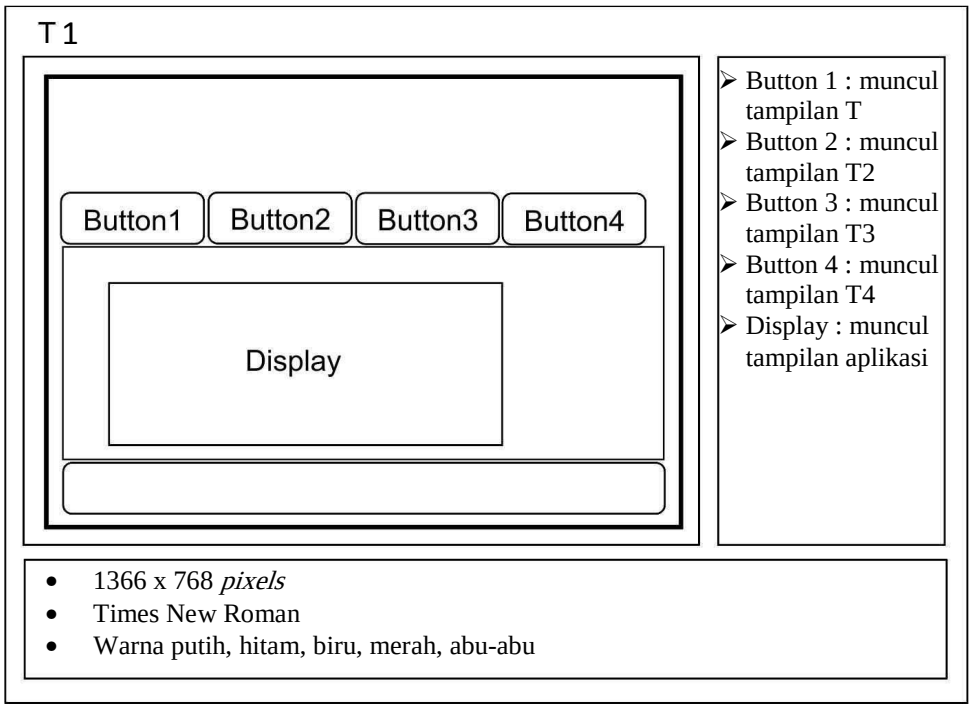
3.2.2 Perancangan Grafis dan Antarmuka

Tahap perancangan desain bertujuan untuk mencari bentuk yang optimal dari aplikasi yang akan dibangun dengan pertimbangan faktor-faktor permasalahan dan kebutuhan yang ada pada sistem seperti yang telah ditetapkan pada tahap analisis. Dalam tahap ini upaya yang dilakukan yaitu dengan cara mengkombinasikan penggunaan teknologi perangkat keras dan perangkat lunak yang tepat sehingga diperoleh desain yang optimal dan mudah di implementasikan.

Perancangan grafis dan antarmuka terdiri dari dua bagian utama yaitu desain grafis dan antarmuka aplikasi *frontend* dan *backend.*

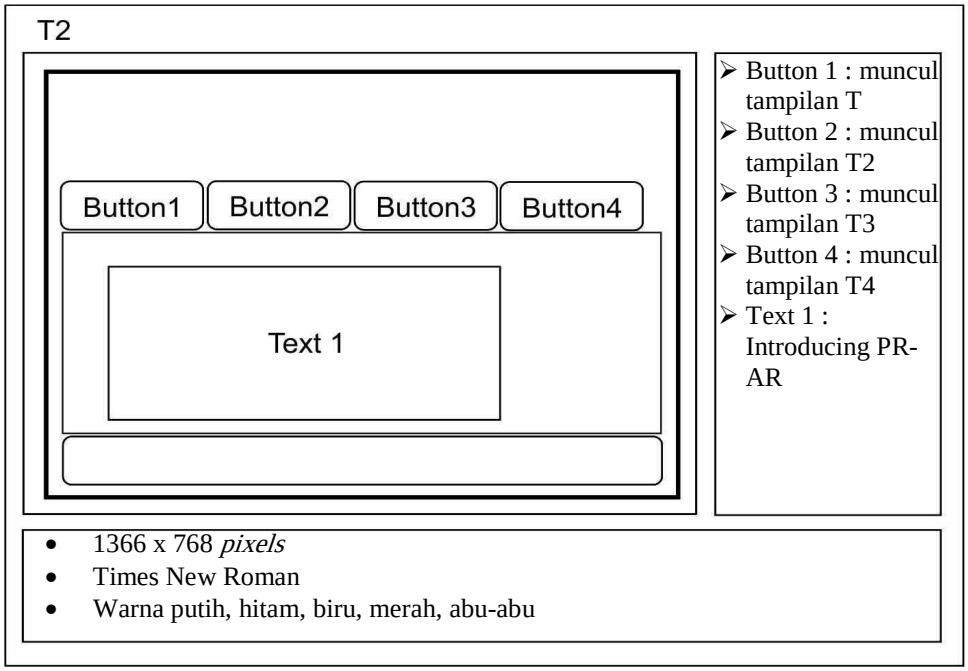
3.2.2.1 Perancangan Antarmuka Aplikasi *Frontend*

1) Perancangan Antarmuka *Menu*



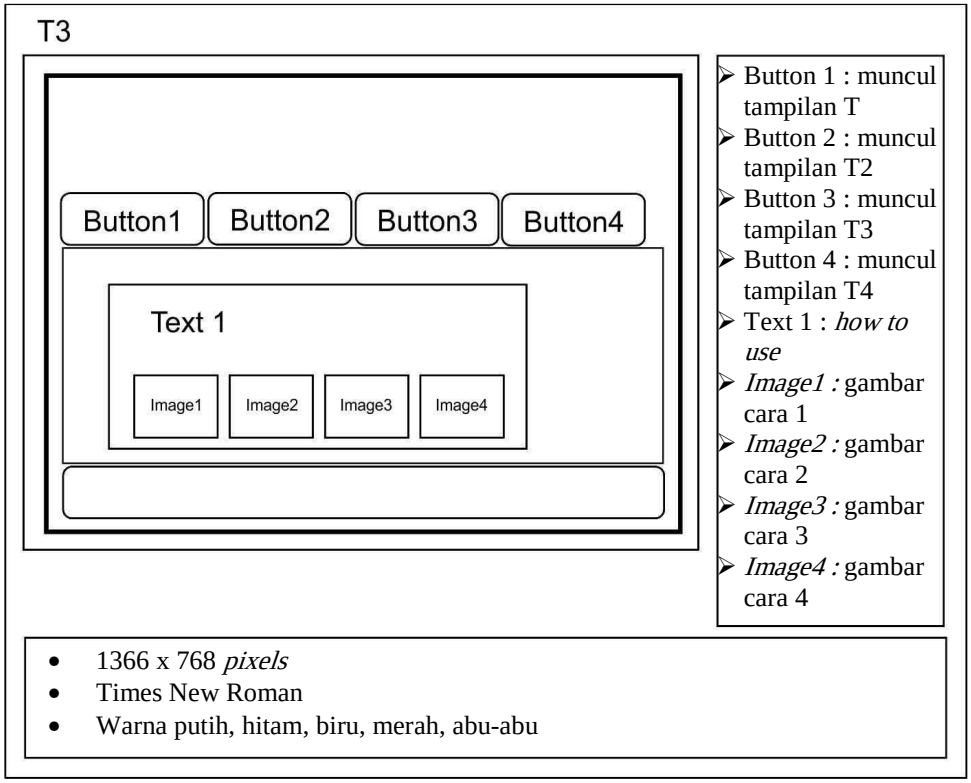
Gambar 3. 54 perancangan Antarmuka Aplikasi *Frontend*

2) Perancangan Antarmuka *About*



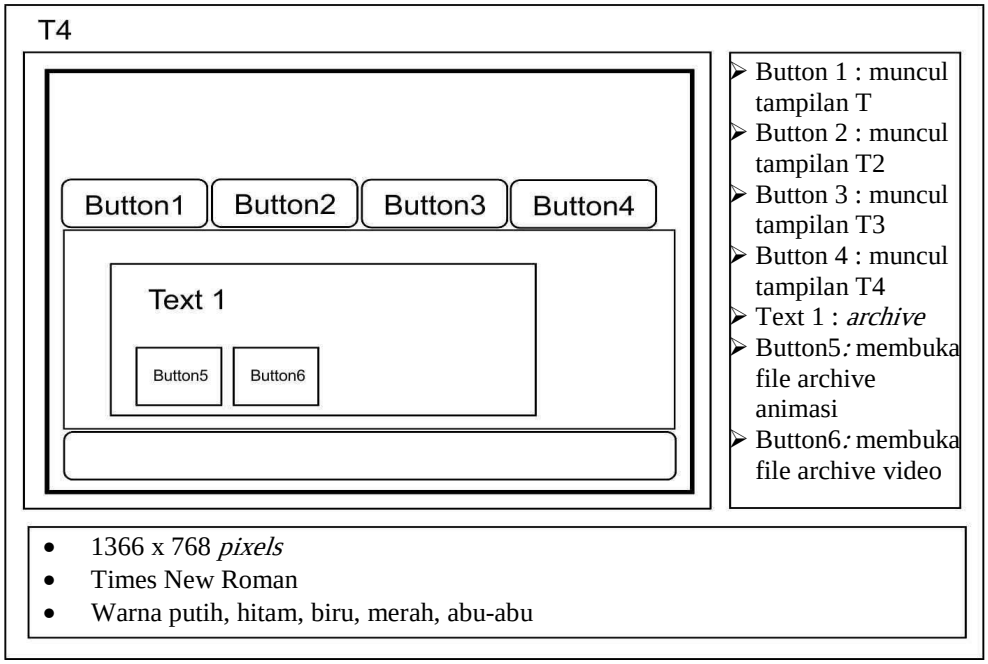
Gambar 3. 55 Perancangan Antarmuka *About*

3) Perancangan Antarmuka *How To Use*



Gambar 3. 56 Perancangan Antarmuka *How To Use*

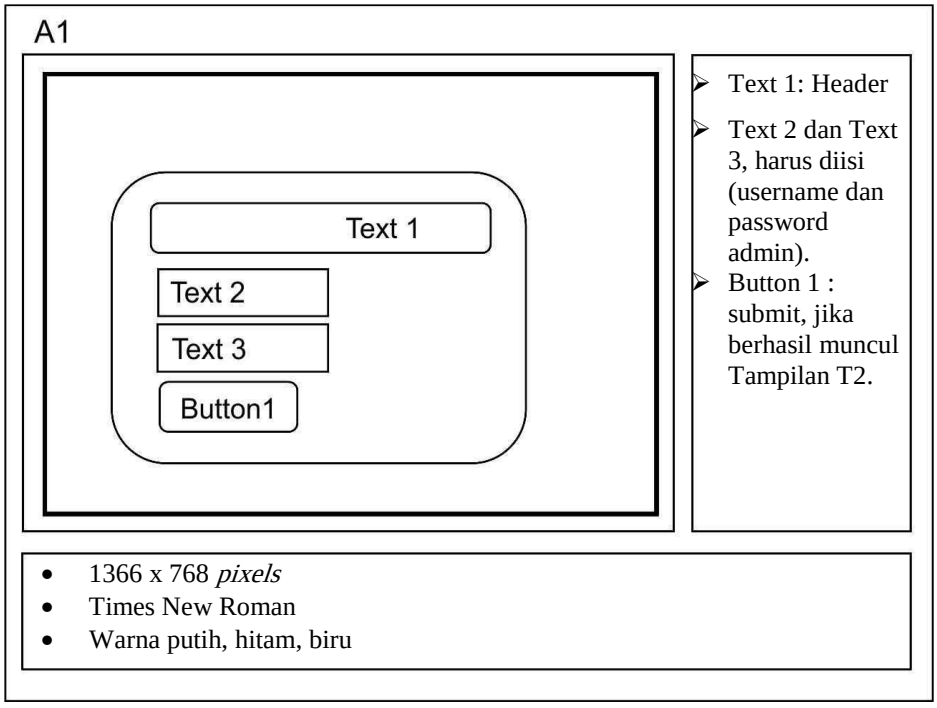
4) Perancangan Antarmuka *Archive*



Gambar 3. 57 Perancangan Antarmuka *Archive*

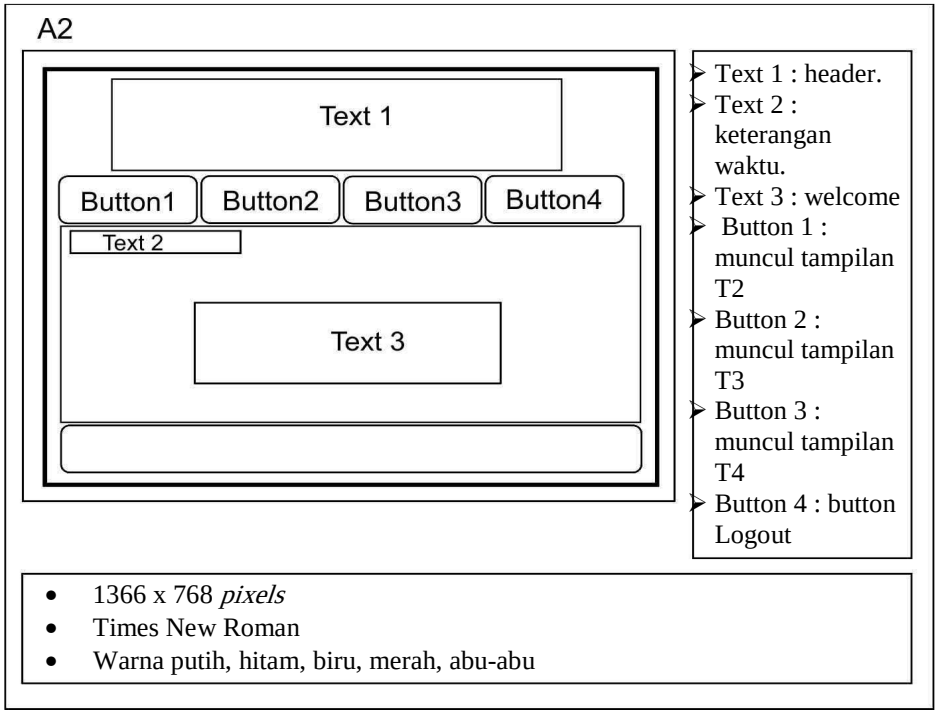
3.2.2.2 Perancangan Antarmuka Aplikasi *Backend*

1) Perancangan Antarmuka *Login* CMS *ARNewspaper*



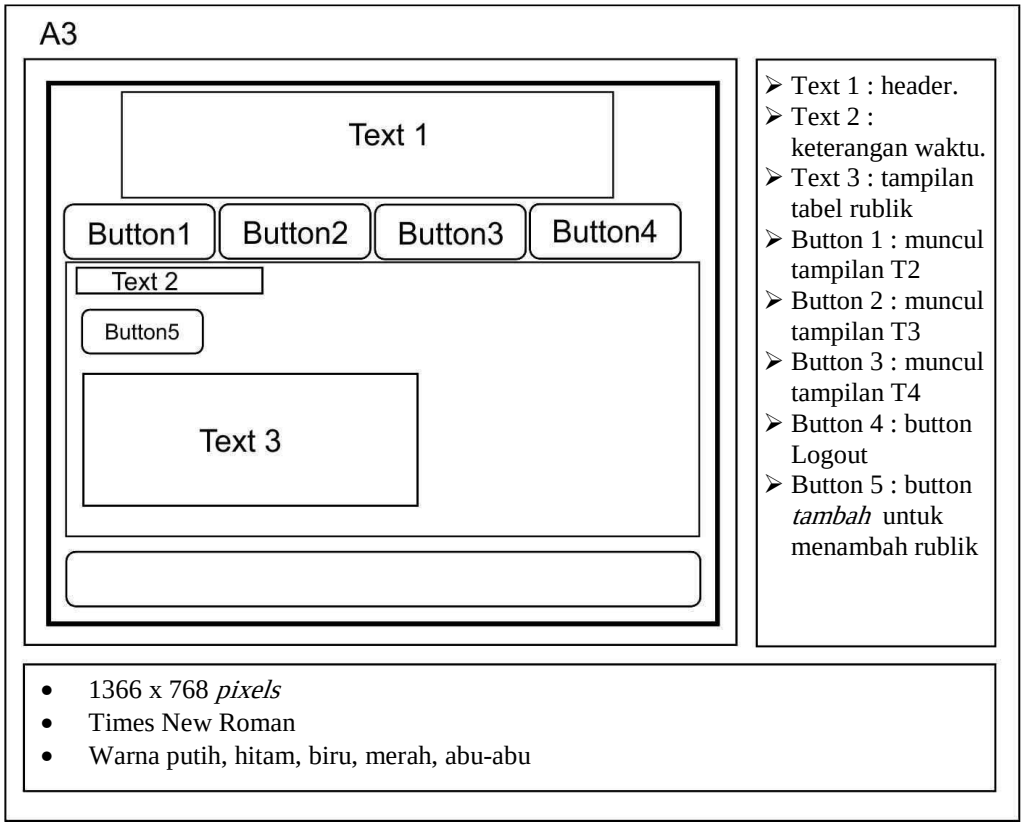
Gambar 3. 58 Perancangan Antarmuka Login CMS ARNewspaper

2) Perancangan Antarmuka Halaman Utama CMS



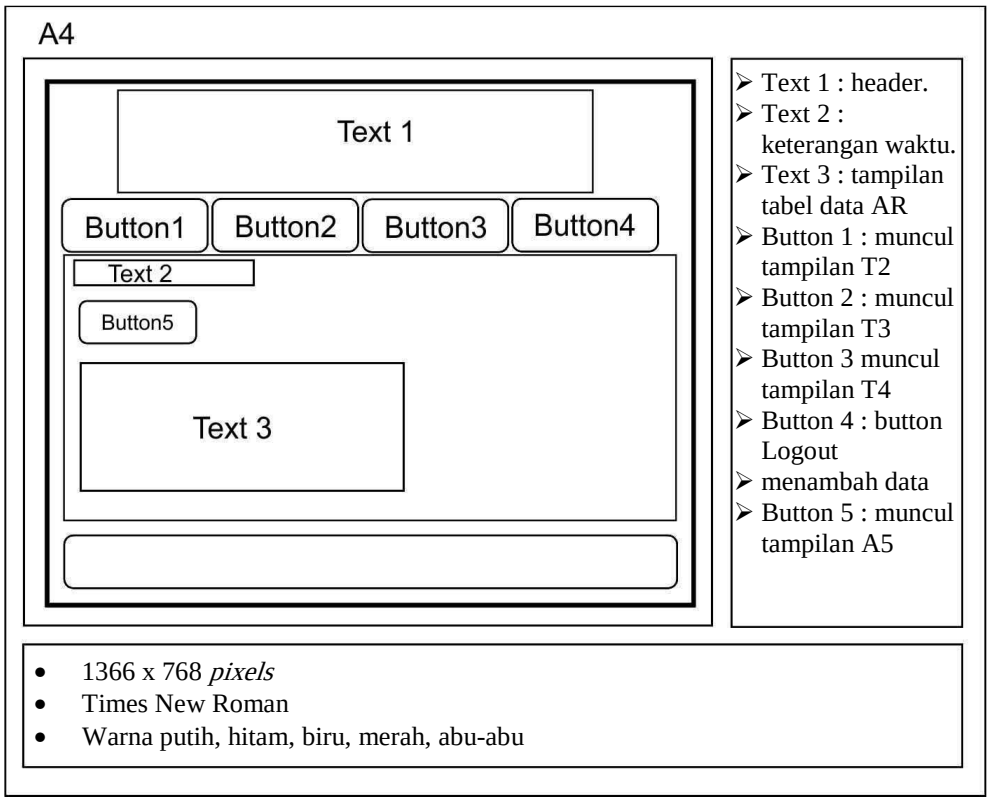
Gambar 3. 59 Perancangan Antarmuka Halaman Utama CMS

3) Perancangan Antarmuka Rublik



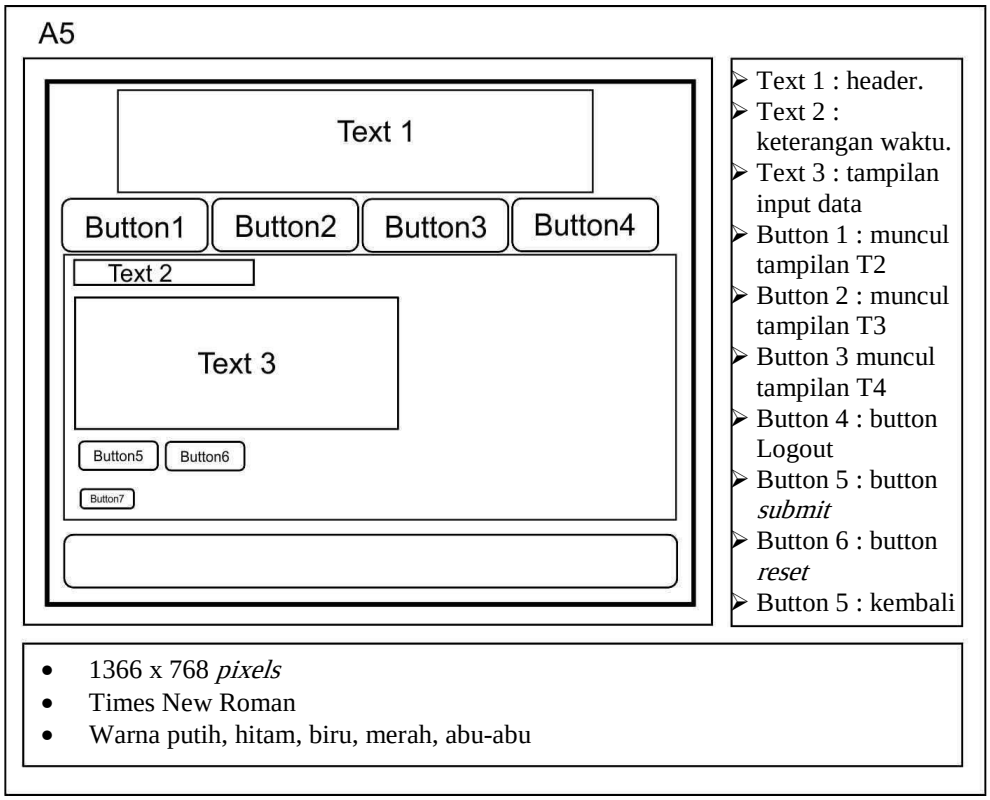
Gambar 3. 60 Perancangan Antarmuka Rublik

4) Perancangan Antarmuka Data



Gambar 3. 61 Perancangan Antarmuka Data

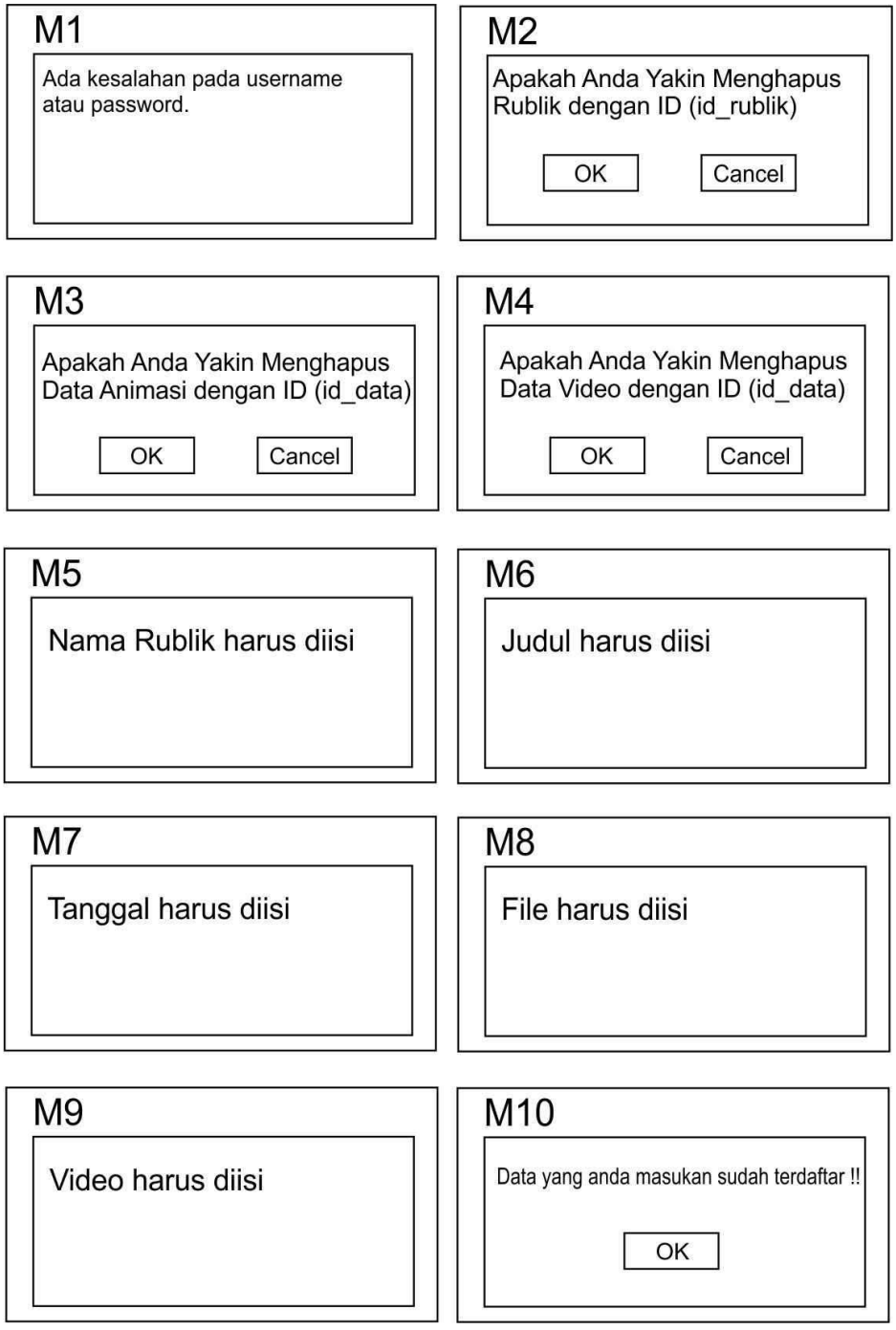
5) Perancangan Antarmuka Manajemen Tambah Data AR



Gambar 3. 62 Perancangan Antarmuka Manajemen Tambah Data AR Animasi

3.2.2.3 Perancangan Pesan

Pada gambar dibawah ini merupakan perancangan pesan yang terdapat pada aplikasi AR*Newspaper* . Adapun tampilan pesan yang ada adalah sebagai berikut :

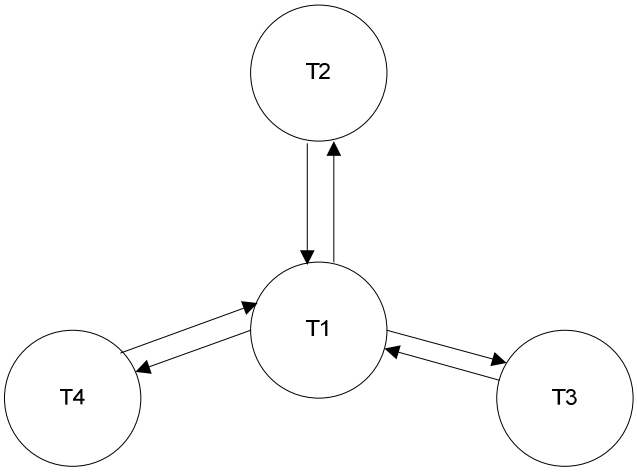


3.2.2.4 Jaringan Semantik

Jaringan semantic sistem yang akan dibangun dibedakan sesuai dengan jenis penggunanya yaitu *user frontend*, *user backend*. Jaringan semantiknya dapat dilihat pada gambar dibawah ini :

1. Jaringan Semantik *User Frontend*

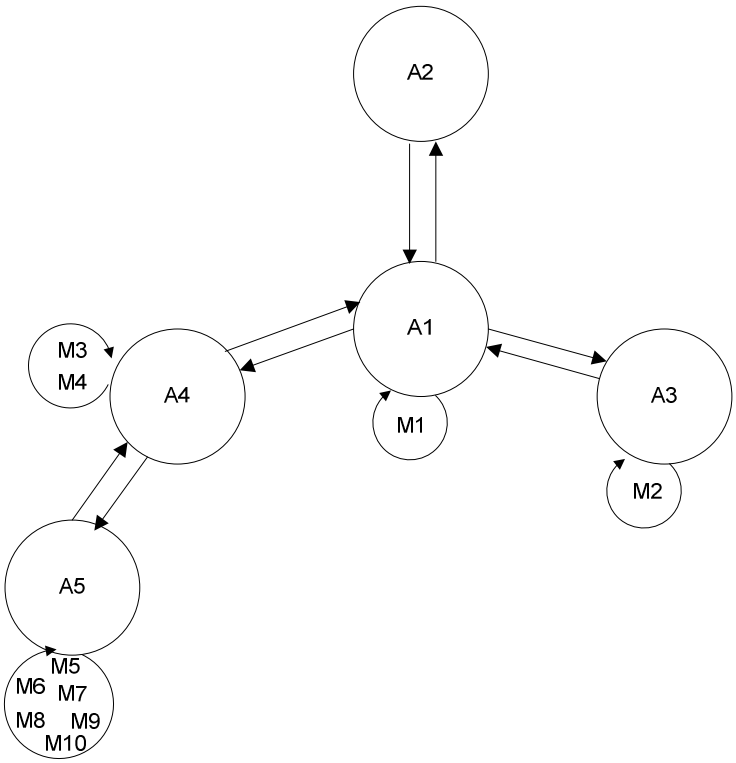
Jaringan semantik *user frontend* dapat dilihat pada Gambar 3.64 :



Gambar 3. 63 Jaringan Semantik *User frontend*

2. Jaringan Semantik *User backend*

Jaringan semantik *user backend* dapat dilihat pada Gambar 3.65 :



Gambar 3. 64 Jaringan Semantik Admin