Pembangunan Aplikasi Virtual Mirror Eyeglasses Menggunakan Teknologi Augmented Reality

Zaid Arham
10107478
Teknik Informatika
Universitas Komputer Indonesia



Prezi
Fullscreen

Latar Belakang

Banyak konsumen mencoba kacamata sehingga membuat kacamata framenya menjadi rusak/cacat.

Konsumen yang ingin mencoba-coba kacamata harus datang ke lokasi toko.

Kurangnya layanan yang mampu menarik minat konsumen.



Prezi
Fullscreen

Identifikasi Masalah

Bagaimana cara merancang dan membangun aplikasi virtual mirror eyeglasses yang mampu memudahkan para konsumen untuk mencoba kacamata secara online dan meningkatkan minat konsumen dengan media yang lebih interaktif

Maksud

Membangun aplikasi Virtual Mirror Eyeglasses yang interaktif dengan menggunakan teknologi Augmented Reality pada Hero Optikal

Tujuan

- Untuk lebih praktis sehingga memudahkan konsumen yang ingin mencoba-coba kacamata sebelum atau tanpa harus membeli karena dapat dicoba secara online.
- Menyediakan layanan kepada konsumen berupa aplikasi Virtual Mirror Eyeglasses secara online yang dapat menghemat waktu konsumen tanpa harus datang langsung ke toko.
- Sebagai sarana promosi dalam menarik minat konsumen.

Prezi

Batasan Masalah

- Penelitian menggunakan personal computer atau notebook yang memiliki fasilitas kamera webcam baik internal maupun eksternal.
- Aplikasi yang dibangun berbasis web
- Menggunakan bahasa pemrograman Action Script 3.0 dengan software pembangun Adobe Flash CS5.5 Professional, Beyond Reality Face SDK sebagai library AR, dan untuk library 3D nya menggunakan Flare3D, serta 3D modeling nya menggunakan 3D Studio Max.
- Menggunakan teknik face detection
- Hanya dapat mendeteksi satu wajah pada saat bersamaan.
- Tidak menyediakan proses transaksi.
- Jarak maksimal wajah terhadap kamera tidak melebihi 1(satu) meter.

Prezi

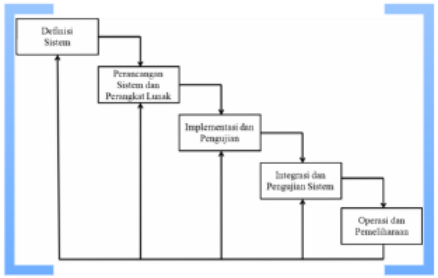
Metodologi penelitian

Analisis Deskriptif

Tahap pengumpulan data

- Studi Literatur.
- Observasi
- Wawancara

Tahap pembuatan perangkat lunak.



Prezi

Definisi

Augmented Reality

Versi Ronald Azuma (1997),

Penggabungan dunia nyata dan virtual

Berjalan secara interaktif dalam waktu nyata (realtime)

Terdapat integrasi antarbenda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata





Prezi

DemProgram

Library pendukung AR

Beyond Reality Face

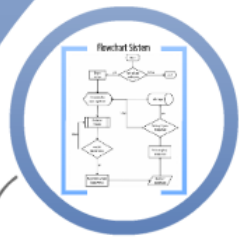
Merupakan sebuah SDK untuk pembangunan aplikasi AR khususnya menggunakan wajah sebagai marker dalam tracking-nya pada platform Adobe Flash

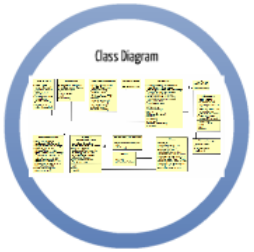
Flare 3D

Salah satu plugin 3D dan dapat membuat model 3D di Platform flash atau mengimport model 3d yang dibuat dengan software pemodelan

Prezi

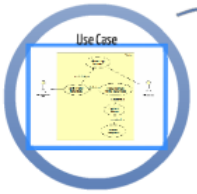
Analisis Sistem

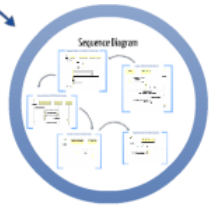








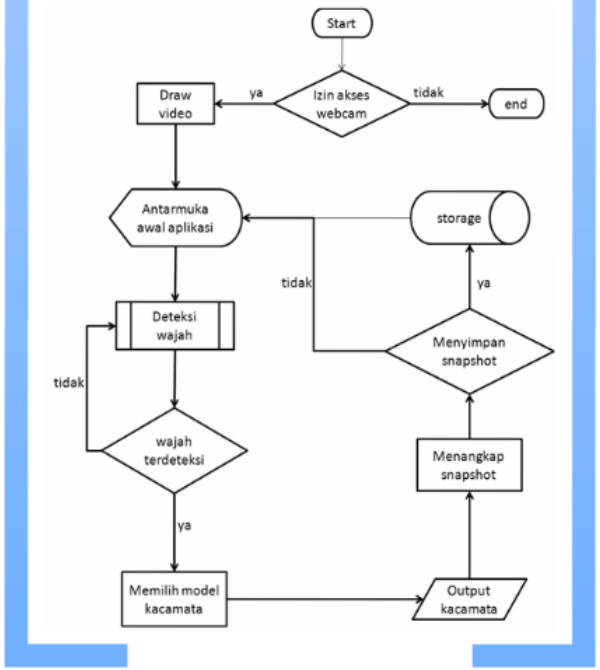




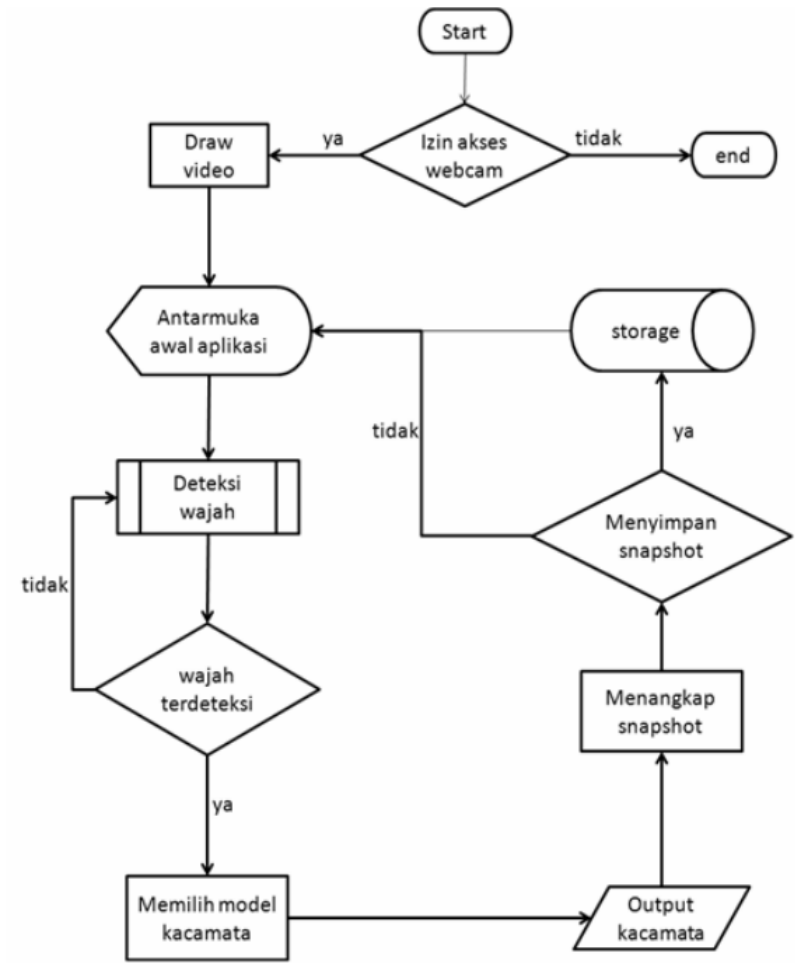


DemProgram
Prezi

Flowchart Sistem

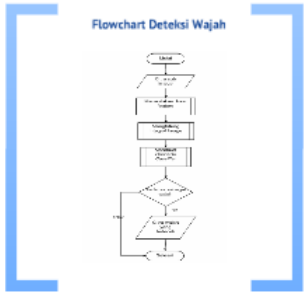


Flowchart Sistem



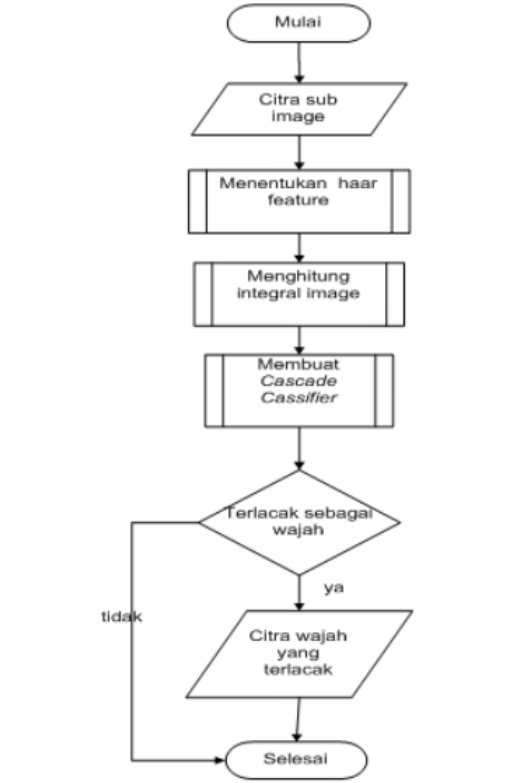
Metode Face Detection

Metode yang digunakan dalam pendeteksian wajah menggunakan metode Haar Cascades

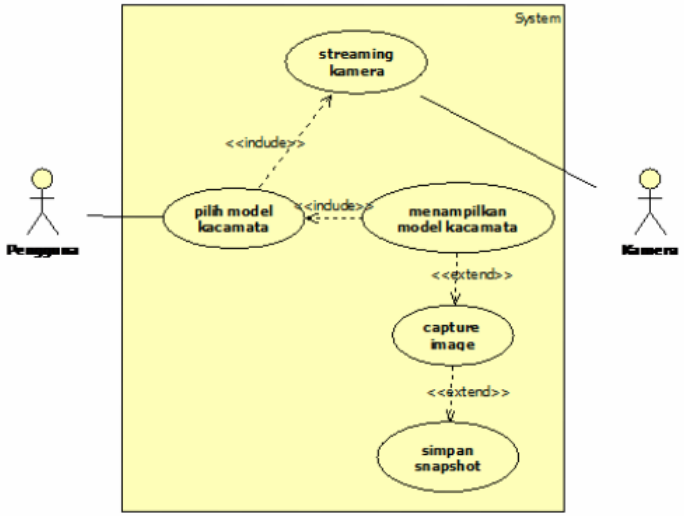


Prezi

Flowchart Deteksi Wajah

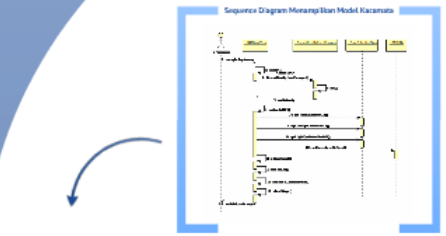


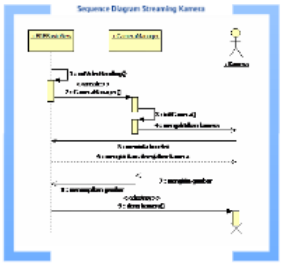
Use Case

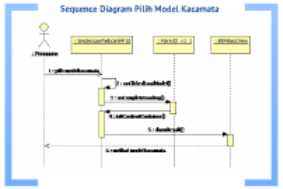


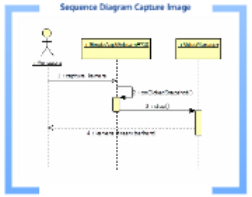
Prezi

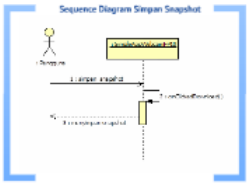
Sequence Diagram





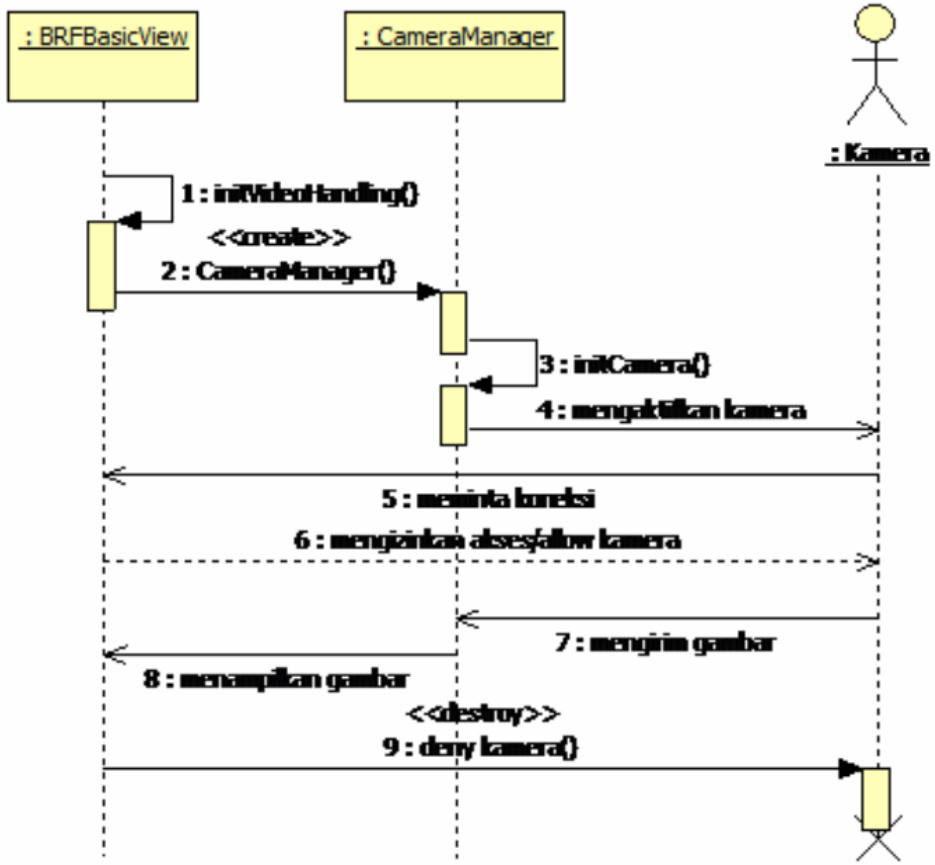






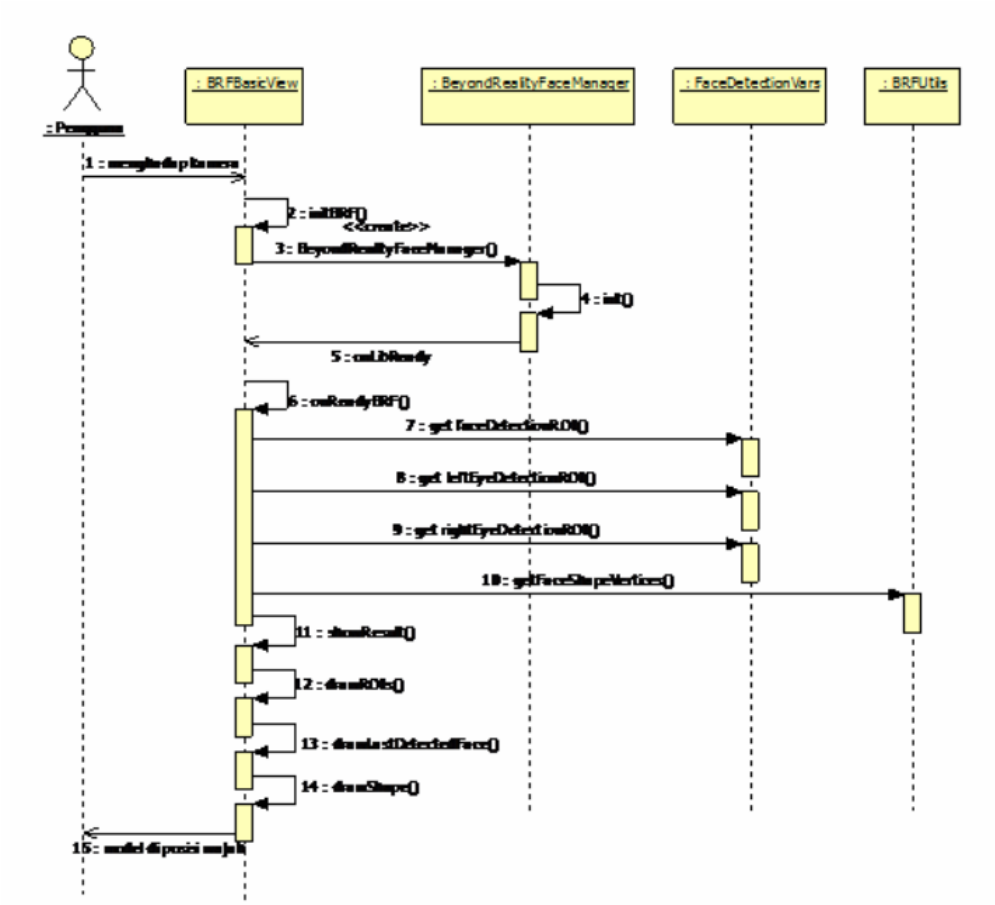
Prezi

Sequence Diagram Streaming Kamera

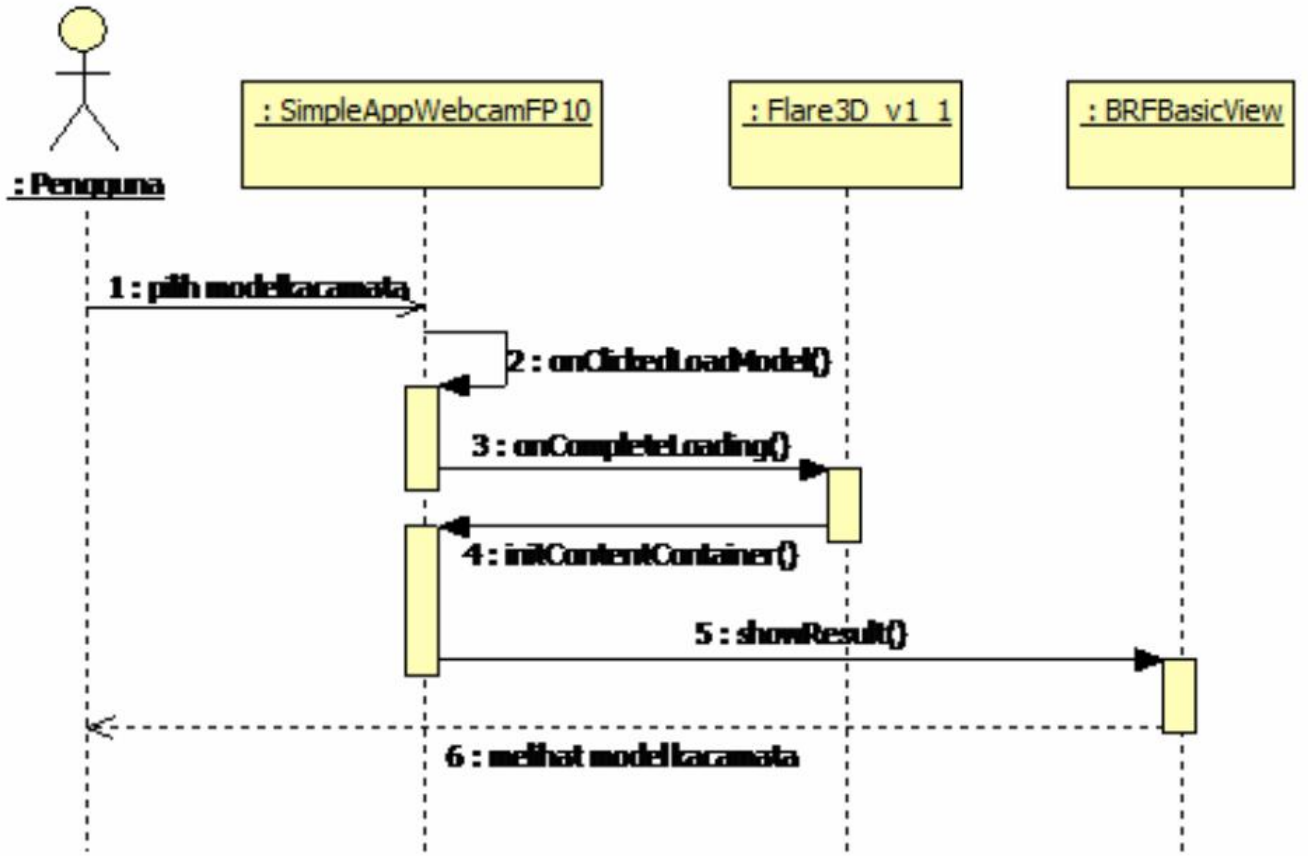


Prezi

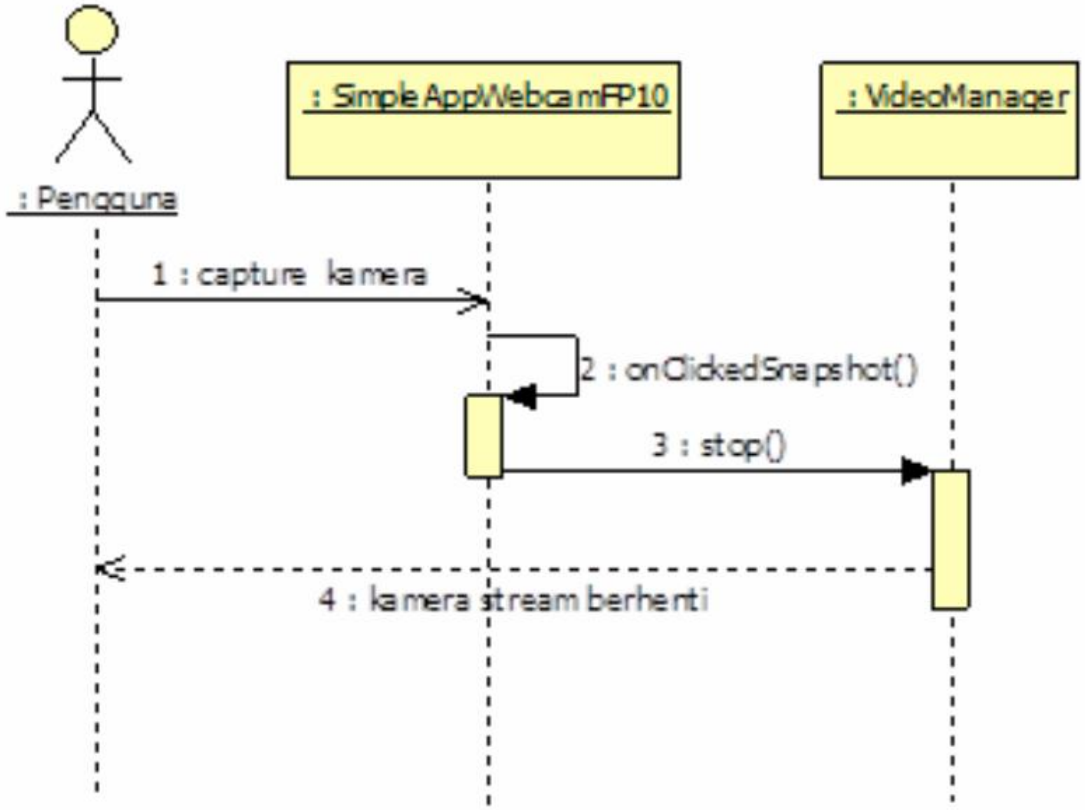
Sequence Diagram Menampilkan Model Kacamata



Sequence Diagram Pilih Model Kacamata

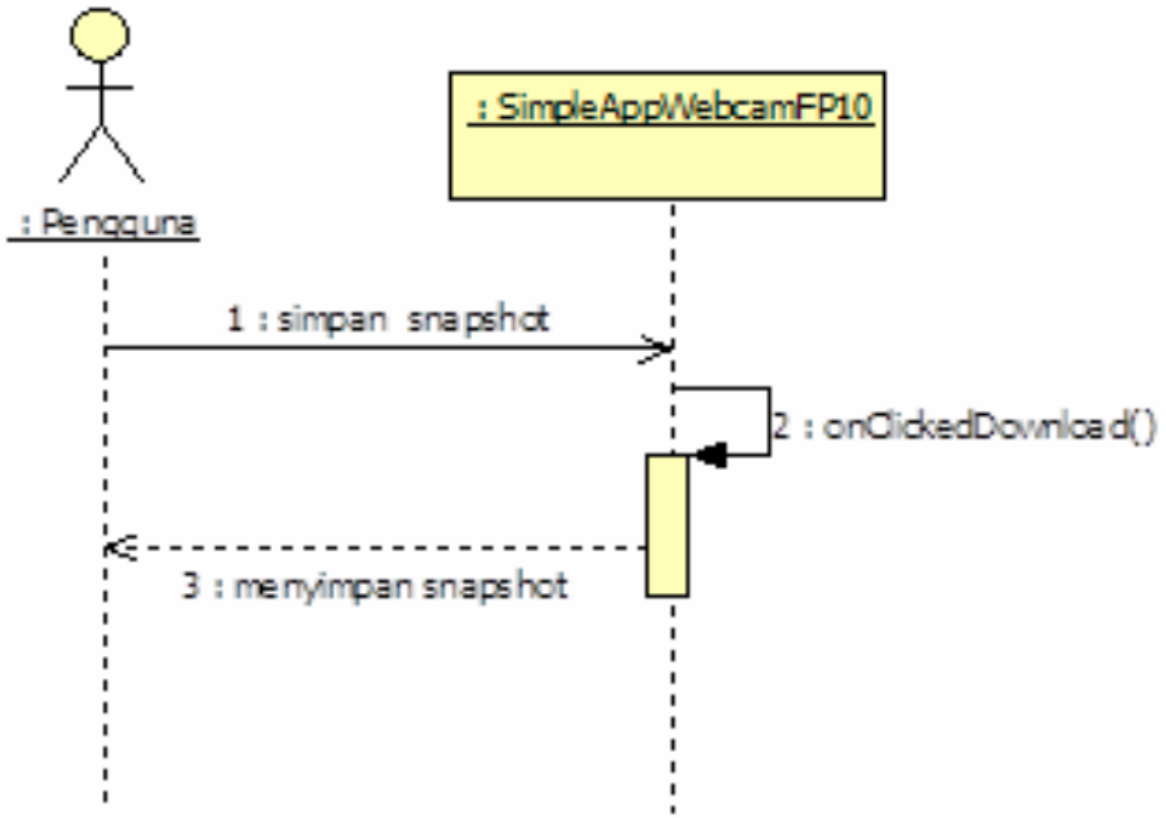


Sequence Diagram Capture Image

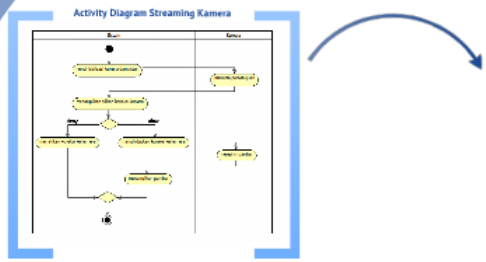


Prezi

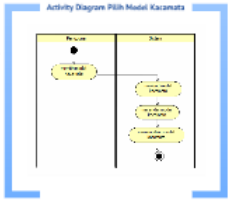
Sequence Diagram Simpan Snapshot

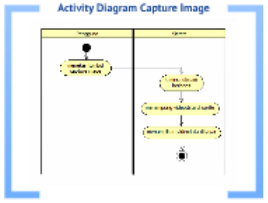


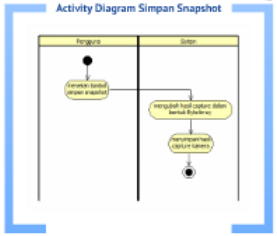
Activity Diagram





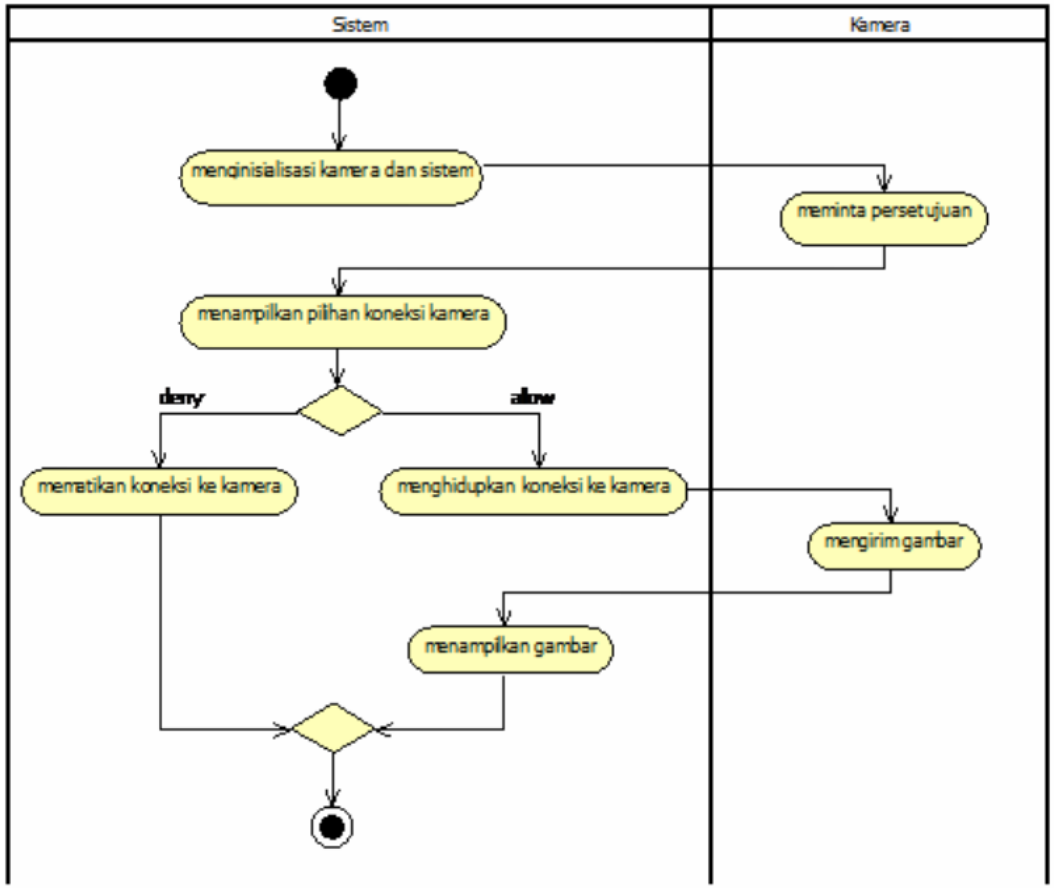






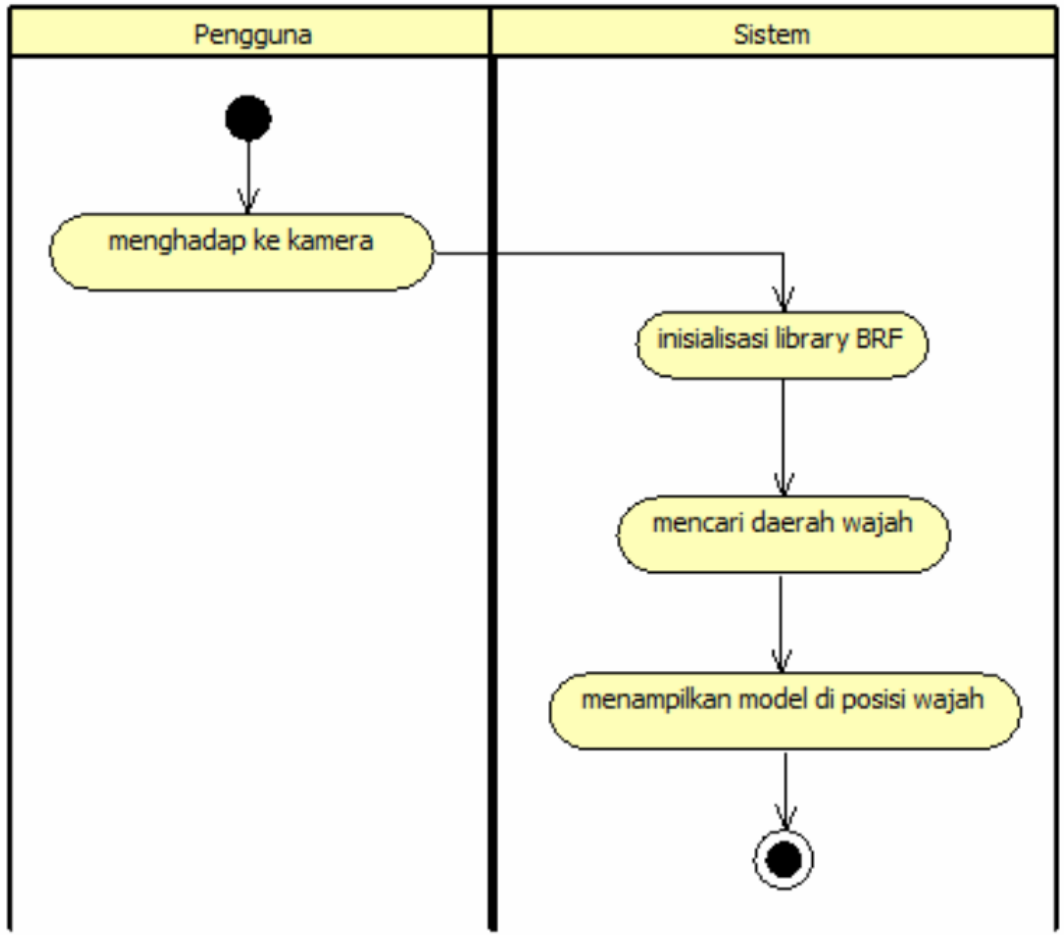
Prezi

Activity Diagram Streaming Kamera



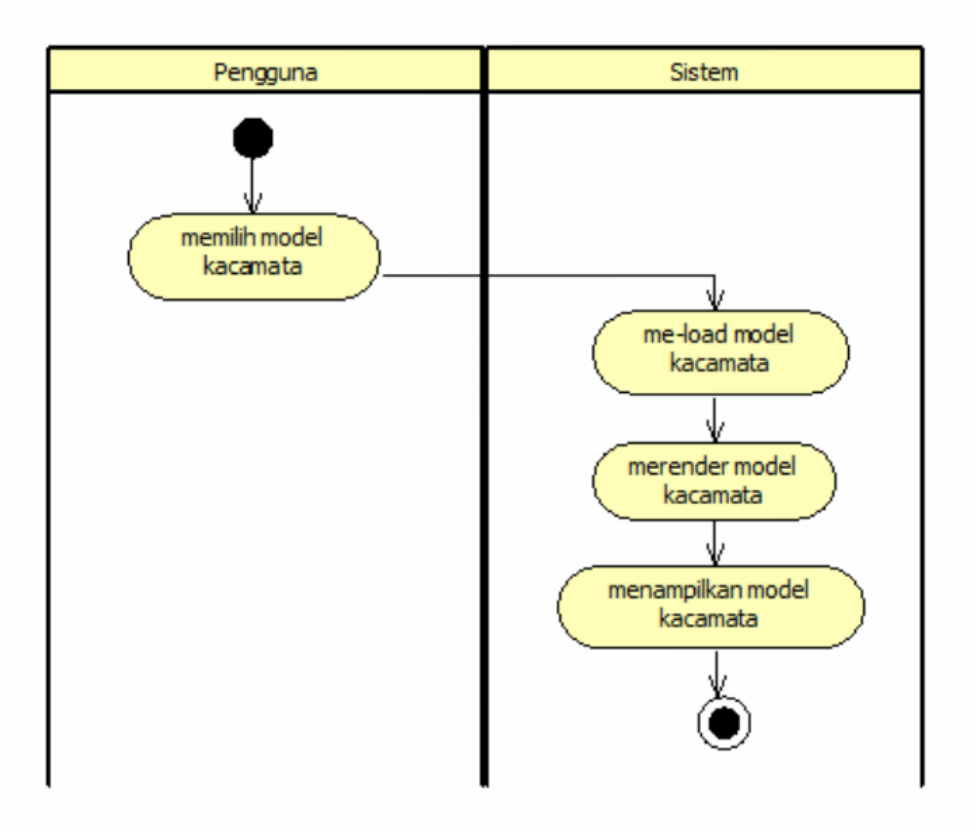
Prezi

Activity Diagram Menampilkan Model Kacamata

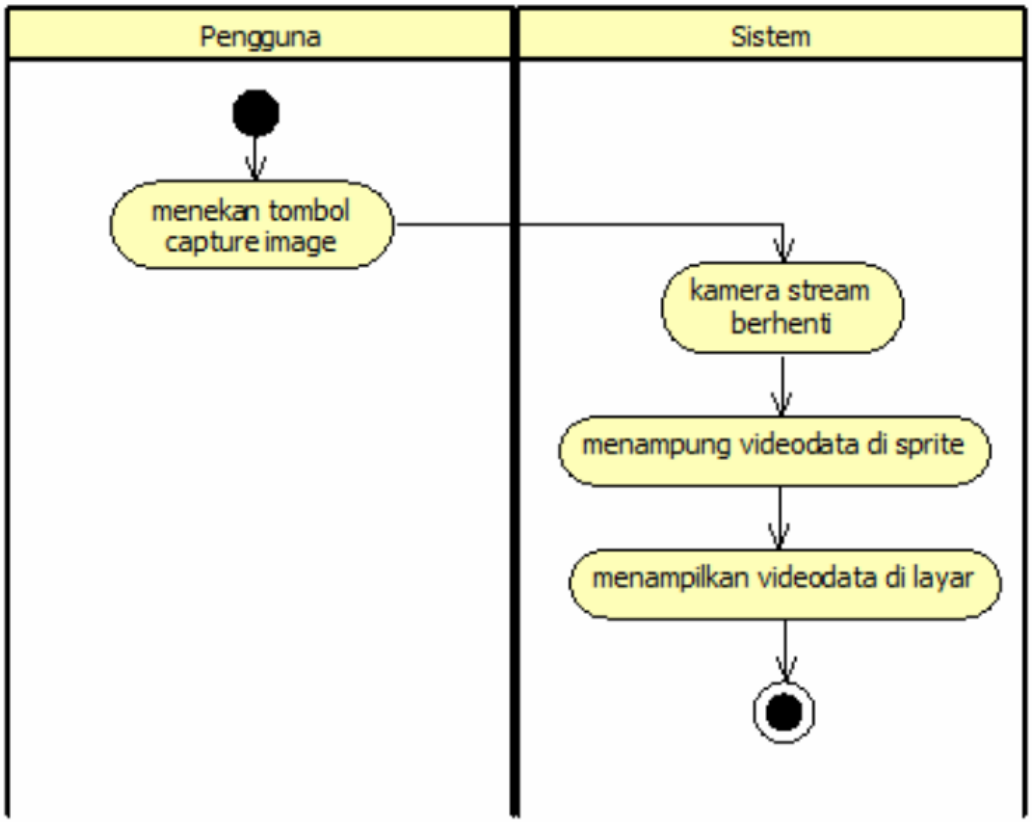


Prezi

Activity Diagram Pilih Model Kacamata

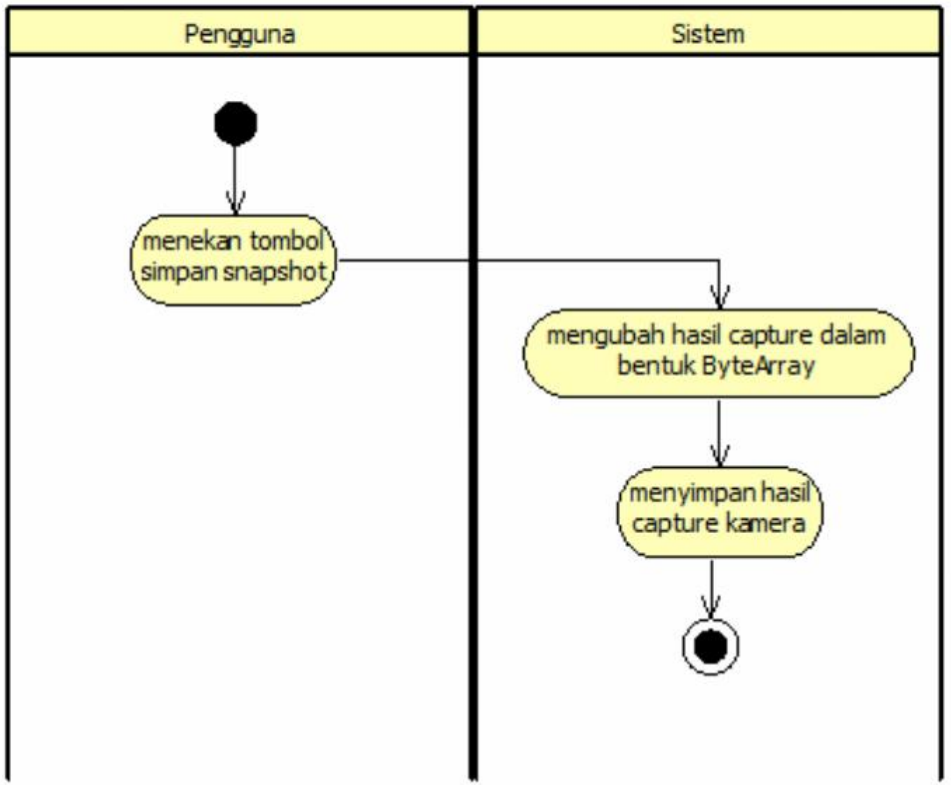


Activity Diagram Capture Image

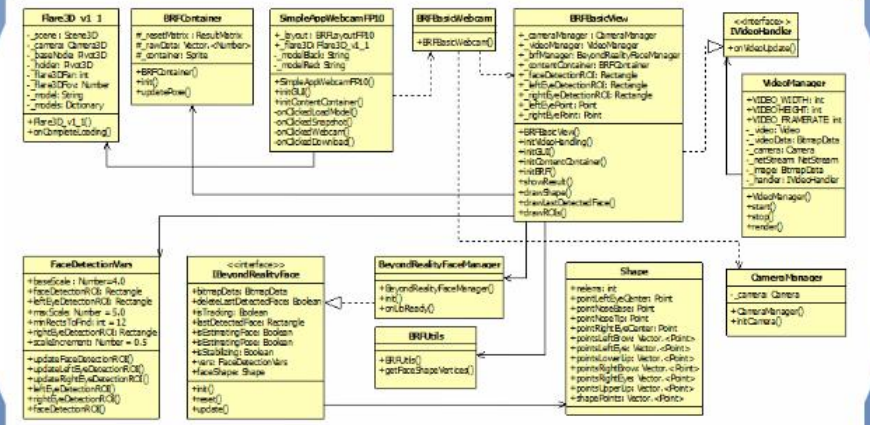


Prezi

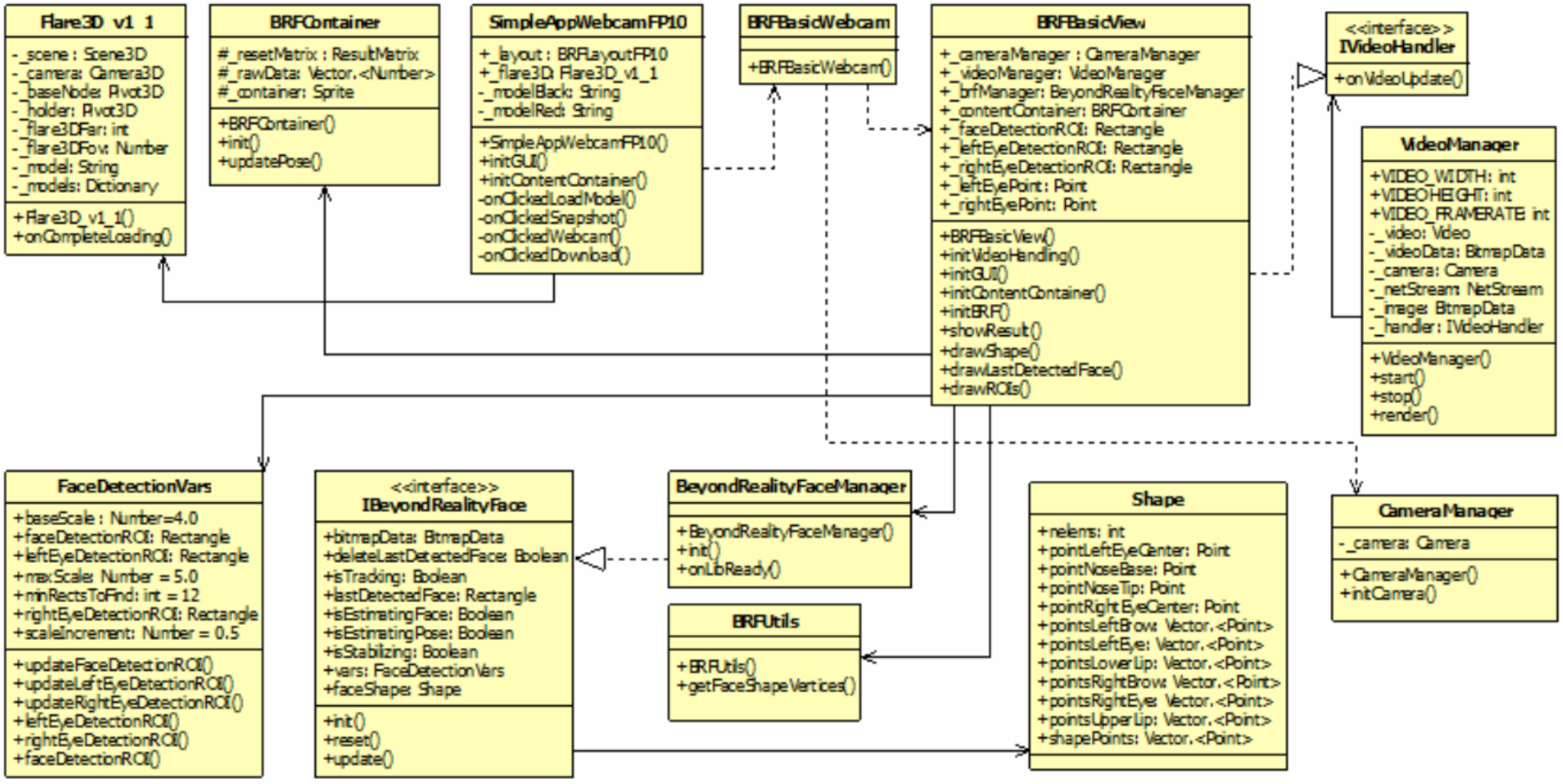
Activity Diagram Simpan Snapshot



Class Diagram



Prezi


Flare3D v1 1
-_scene : Scene3D
-_camera: Camera3D
-_baseNode: Pivot3D
-_holder: Pivot3D
-_flare3DFar: int
-_flare3DFov: Number
-_model: String
-_models: Dictionary
+Flare3D_v1_1()
+onCompleteLoading()
BRFContainer
#_resetMatrix : ResultMatrix
#_rawData: Vector.<Number>
#_container: Sprite
+BRFContainer()
+init()
+updatePose()
SimpleAppWebcamFP10
+_layout : BRFLayoutFP10
+_flare3D: Flare3D_v1_1
-_modelBlack: String
-_modelRed: String
+SimpleAppWebcamFP10()
+initGUI()
+initContentContainer()
-onClickedLoadModel()
-onClickedSnapshot()
-onClickedWebcam()
-onClickedDownload()
BRFBasicWebcam
+BRFBasicWebcam()
BRFBasicView
+_cameraManager : CameraManager
+_videoManager: VideoManager
+_brfManager: BeyondRealityFaceManager
+_contentContainer: BRFContainer
+_faceDetectionROI: Rectangle
+_leftEyeDetectionROI: Rectangle
+_rightEyeDetectionROI: Rectangle
+_leftEyePoint: Point
+_rightEyePoint: Point
+BRFBasicView()
+initVideoHandling()
+initGUI()
+initContentContainer()
+initBRF()
+showResult()
+drawShape()
+drawLastDetectedFace()
+drawROIs()
<<interface>>
IVideoHandler
+onVideoUpdate()
VideoManager
+VIDEO_WIDTH: int
+VIDEO_HEIGHT: int
+VIDEO_FRAMERATE: int
-_video: Video
-_videoData: BitmapData
-_camera: Camera
-_netStream: NetStream
-_image: BitmapData
-_handler: IVideoHandler
+VideoManager()
+start()
+stop()
+render()
FaceDetectionVars
+baseScale : Number=4.0
+faceDetectionROI: Rectangle
+leftEyeDetectionROI: Rectangle
+maxScale: Number = 5.0
+minRectsToFind: int = 12
+rightEyeDetectionROI: Rectangle
+scaleIncrement: Number = 0.5
+updateFaceDetectionROI()
+updateLeftEyeDetectionROI()
+updateRightEyeDetectionROI()
+leftEyeDetectionROI()
+rightEyeDetectionROI()
+faceDetectionROI()
<<interface>>
IBeyondRealityFace
+bitmapData: BitmapData
+deleteLastDetectedFace: Boolean
+isTracking: Boolean
+lastDetectedFace: Rectangle
+isEstimatingFace: Boolean
+isEstimatingPose: Boolean
+isStabilizing: Boolean
+vars: FaceDetectionVars
+faceShape: Shape
+init()
+reset()
+update()
BeyondRealityFaceManager
+BeyondRealityFaceManager()
+init()
+onLibReady()
BRFUtils
+BRFUtils()
+getFaceShapeVertices()
Shape
+nelems: int
+pointLeftEyeCenter: Point
+pointNoseBase: Point
+pointNoseTip: Point
+pointRightEyeCenter: Point
+pointsLeftBrow: Vector.<Point>
+pointsLeftEye: Vector.<Point>
+pointsLowerLip: Vector.<Point>
+pointsRightBrow: Vector.<Point>
+pointsRightEye: Vector.<Point>
+pointsUpperLip: Vector.<Point>
+shapePoints: Vector.<Point>
CameraManager
-_camera: Camera
+CameraManager()
+initCamera()

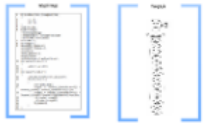
PREZI

Pengujian Whitebox

Cascade Classifier



Integral Image



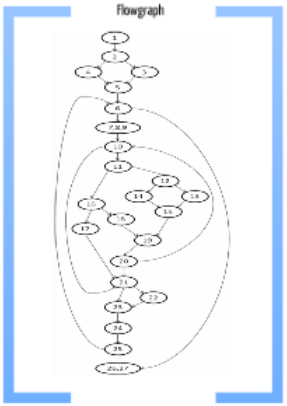
Prezi

Cascade Classifier

Cascade Classifier



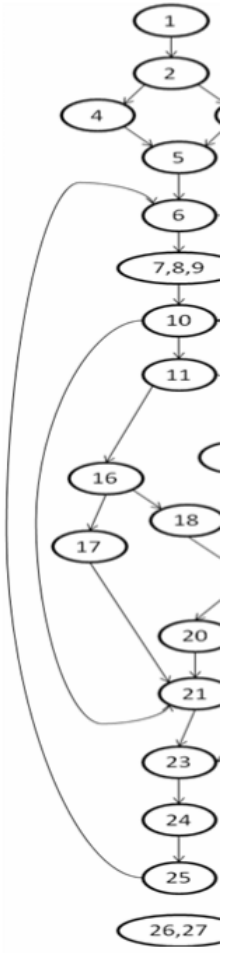
Flowgraph



Cascade Classifier

```
1.  public function run( r:Rectangle ):int
    {
        var x:int = r.x;
        var y:int = r.y;
        var w:int = r.width;
        var h:int = r.height;
        var mean:Number =
    targetImage.getSum(x,y,w,h) * inv_window_area;
        var variance_norm_factor:Number =
    targetImage.getSum2(x,y,w,h)* inv_window_area -
    mean*mean;
2.      if( variance_norm_factor >= 0 ){
3.          variance_norm_factor=
    Math.sqrt(variance_norm_factor);
4.      } else {
            variance_norm_factor= 1;
5.      }
        var features:Array;
        var val:Number = 0
        var sum:Number = 0;
        var feature:FeatureBase;
        var i:int=0;
        var j:int=0;
        var st_th:Number = 0;
        var tree:FeatureTree = firstTree;
6.      while ( tree != null )
        {
7.          feature   = tree.firstFeature
8.          val       = 0;
9.          st_th     = tree.stage_threshold;

10.         while ( feature != null )
            {
11.             sum  = feature.getSum( targetImage, x, y
    );
12.             if (sum < feature.threshold *
    variance_norm_factor){
13.                 val += feature.left_val;
14.             }else{
                    val += feature.right_val;
15.             }
16.             if( val > st_th ){
                    // left_val, right_val are always
    plus
17.                 break;
18.             }
19.             feature = feature.next;
20.         }
21.         if( val < st_th ){
22.             return 0;
23.         }
24.         tree = tree.next;
25.     }
26.     return 1;
27. }
```



Classifier

```
le ):int


        =
* inv_window_area;
ctor:Number =
)* inv_window_area -

tor >= 0 ){
_factor=
tor);

_factor= 1;


e;


firstTree;

firstFeature

stage_threshold;

!= null)

feature.getSum( targetImage, x, y

< feature.threshold *

val += feature.left_val;

val += feature.right_val;

st_th ){
// left_val, right_val are always

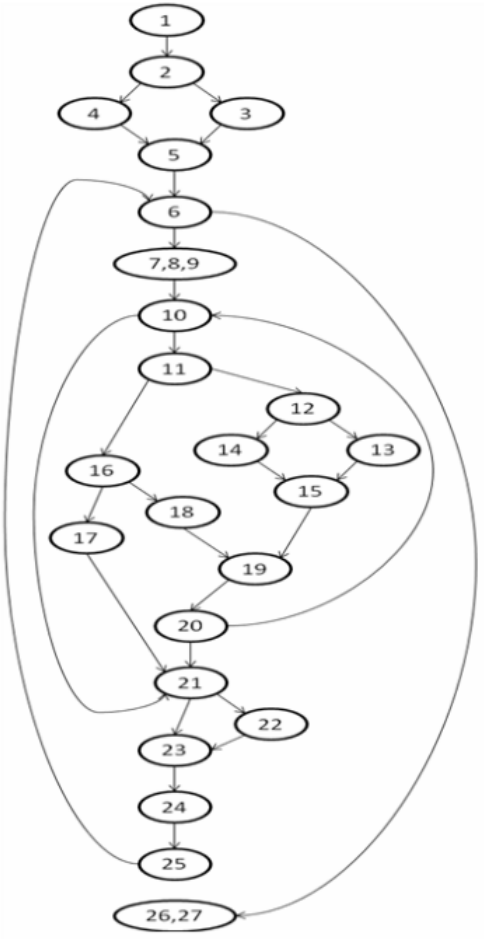
break;

= feature.next;

{
;

;
```

Flowgraph



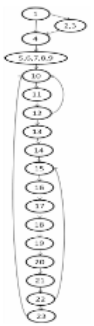
PREZI

Integral Image

Integral Image

```
1   if( !b.width()+1]*sizi ) !b.height()+1]*sizi )
    {
2       _ii = [];
3       _ii2 = [];
4   }
5   width = b.width;
6   height = b.height;
    // build IntegralImages
    // IntegralImage is 1 size large than image
    // ii[ 0 for the 1st row,col,rows
7   iiw = width + 1;
8   iih = height + 1;
9   var single[1 :Number = 0;
    var single[2 :Number = 0;
    var index:int;
    var ha_index:int = 1;
    var pix:Number;
    var ha:ByteArray = b.getPixels( b.rect );
10  for( var i:int=0; i<iiw; i++ )
    {
11      _ii[ i ] = _ii[ i ] = 0;
12  }
13  for( var j:int=1; j<iih; j++ )
    {
14      _ii[ index = ii[*iiw) ] = _ii[index] = 0;
15      for( i=1; i<iiw; i++ )
        {
16          pix = ha[ha_index];
17          single[1]   =   Number(_ii[int(index-iiw+1)]) +
Number(_ii[index]) - Number(_ii[int(index-iiw)]) + pix;
18          single[2]   = Number(_ii2[int(index-iiw+1)]) +
Number(_ii2[index]) - Number(_ii2[int(index-iiw)]) + pix*pix;
19          _ii[ ++index ] = single[1];
20          _ii2[ index ] = single[2];
21          ha_index+=4;
22      }
23  }
```

Flowgraph



Prezi

Integral Image

```
1   if( (b.width+1)!=iiw || (b.height+1)!=iih )
    {
2           _ii  = [];
3           _ii2 = [];
4   }
5   width = b.width;
6   height = b.height;
    // build IntegralImages
    // IntegralImage is 1 size larger than image
    // all 0 for the 1st row,column
7   iiw = width + 1;
8   iih = height + 1;
9   var singleII  :Number = 0;
    var singleII2 :Number = 0;
    var index:int;
    var ba_index:int = 1;
    var pix:Number;
    var ba:ByteArray = b.getPixels( b.rect );
10  for( var i:int=0; i<iiw; i++ )
    {
11          _ii2[ i ] = _ii[  i ] = 0;
12  }
13  for( var j:int=1; j<iih; j++ )
    {
14          _ii2[ index = int(j*iiw) ] = _ii[index] = 0;
15          for( i=1; i<iiw; i++ )
            {
16                  pix = ba[ba_index];
17                  singleII    =   Number(_ii[int(index-iiw+1)])  +
    Number(_ii[index]) - Number(_ii[int(index-iiw)]) + pix;
18                  singleII2   =   Number(_ii2[int(index-iiw+1)])  +
    Number(_ii2[index]) - Number(_ii2[int(index-iiw)]) + pix*pix ;
19                  _ii[ ++index ] = singleII;
20                  _ii2[ index ] = singleII2;
21                  ba_index+=4;
22          }
23  }
```

ge

```
iih )

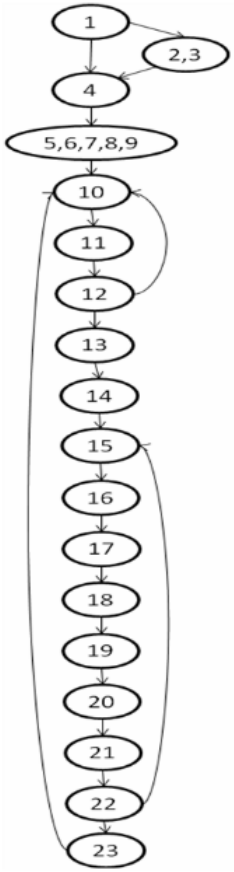

image


ct );


i[index] = 0;


er(_ii[int(index-iiw+1)])  +
t(index-iiw)]) + pix;
(_ii2[int(index-iiw+1)])  +
int(index-iiw)]) + pix*pix ;
eII;
I2;
```

Flowgraph





Prezi