BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan proses penguraian konsep ke dalam bagian-bagian yang lebih sederhana, sehingga struktur logisnya menjadi jelas. Tahap analisis sistem adalah tahapan dasar sebelum perancangan untuk memperbaiki sistem yang lama. Dari hasil analisis tersebut dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efisien. Pada tahap analisis sistem ini dibagi menjadi beberapa bagian, yaitu sebagai berikut :

1. Analisis Masalah.
2. Analisis Sistem yang Sedang Berjalan
3. Analisis Algoritma.
4. Analisis Spesifikasi Kebutuhan Perangkat Lunak
5. Analisis Kebutuhan Non Fungsional
6. Analisis Data
7. Analisis Kebutuhan Fungsional

3.1.1 Analisis Masalah

Analisis masalah pada subbab ini adalah penjabaran masalah apa saja yang ada di website Toko Sakinah Kerudung sebelum dibangunnya perangkat lunak jilbab virtual ini. Oleh karena itu analisis masalah yang terdapat di web Toko Sakinah Kerudung meliputi hal-hal sebagai berikut :

1. Belum adanya pemanfaatan maksimal web toko Sakinah Kerudung. Hal tersebut terbukti dengan belum maksimalnya penjualan melalui web dibandingkan penjualan secara langsung.
2. Belum adanya fasilitas yang membantu meningkatkan kepercayaan konsumen terhadap toko Sakinah Kerudung, seperti adanya *fitting room virtual*.
3. Masih kurangnya kepercayaan konsumen terhadap kerudung yang ada di website toko Sakinah Kerudung. Karena keraguan apakah kerudung yang dipilih cocok menurut kriteria konsumen tersebut.

3.1.2 Analisis Sistem Yang Sedang Berjalan

Analisis sistem yang sedang berjalan di toko online Sakinah Kerudung terdiri dari prosedur dan *flowchart*. Adanya analisis sistem sangat penting karena fungsi dari sistem itu sendiri yaitu untuk mengetahui bagaimana sebuah sistem itu berjalan agar sistem yang dibuat menghasilkan keluaran yang diinginkan dan dapat mencapai tujuan yang direncanakan. Prosedur sistem yang sedang berjalan yang terlibat pada sistem yang akan dibangun ialah prosedur pilih barang.

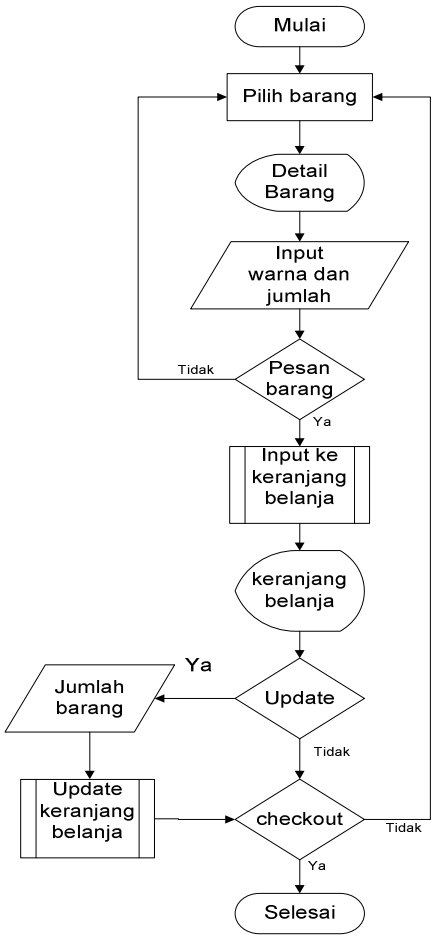
3.1.2.1 Prosedur Pilih Barang

Di toko *online* Sakinah Kerudung untuk melakukan proses pilih barang dijelaskan dalam prosedur sebagai berikut:

1. Konsumen dapat melakukan pemilihan barang sesuai jenis-jenis kerudung yang ada di menu.

2. Ketika konsumen memilih kerudung yang ada maka sistem yang telah ada akan menampilkan detail barang yang berisi harga, ketersediaan, tanggal, *rating*, pilihan warna, diskon, dan *field* jumlah.
3. Sebelum memasukkan ke keranjang belanja konsumen harus mengisi warna dan jumlah yang ingin dipesan.
4. Jika konsumen memilih untuk memesan barang maka data yang telah disi akan masuk ke keranjang belanja. Dan jika tidak jadi memesan maka konsumen dapat kembali memilih barang lain.
5. Setelah ada data yang terinput ke keranjang belanja, konsumen dapat melihat isi dari keranjang belanja miliknya.
6. Konsumen dapat mengupdate data jumlah pesanan yang jadi dipesan.
7. Jika tidak melakukan update konsumen dapat langsung melakukan *checkout* keranjang belanja yang berarti akan mengakhiri proses pemesanan.
8. Jika tidak *checkout* berarti konsumen dapat melanjutkan memilih barang.

Untuk Gambar *flowchart* dari prosedur pemesanan barang dapat dilihat pada gambar di bawah ini :



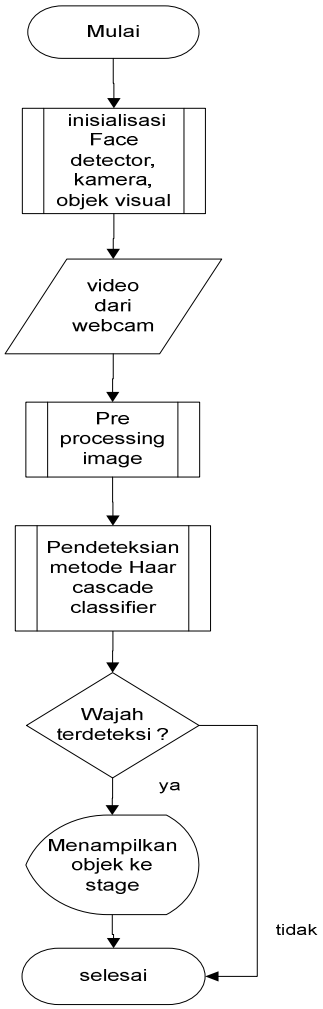
Gambar 3. 1 *FlowChart* Pemesanan Barang

3.1.3 Analisis Algoritma

Secara umum, *augmented reality* (AR) adalah penggabungan antara objek virtual dengan objek nyata. Jadi perangkat lunak jilbab virtual yang dirancang dengan teknologi AR ini, seolah-olah menggabungkan objek virtual dengan objek nyata. Dalam kasus ini objek nyatanya berupa pola wajah dan objek virtualnya berupa model jilbab.

Di sistem ini webcam berperan penting sebagai media visi bagi perangkat lunak untuk mendapatkan video masukan. Kamera mengambil frame-frame video untuk dapat diterima oleh komputer. Komputer memproses citra digital yang

diakuisisi oleh webcam, frame demi frame. Komputer akan mendeteksi pola yang mirip dengan wajah dari setiap frame video tersebut, yang kemudian lokasi wajah dapat ditentukan. Dengan informasi tersebut, objek virtual digabungkan dengan video dari webcam dan merendernya sesuai dengan informasi posisi yang diperoleh dari *face detector* tersebut. Proses tersebut berlangsung secara real-time sehingga model virtual yang tampil di media display akan mengikuti pergerakan wajah. Gambaran umum sistem ini dapat dilihat dalam *flowchart* gambar 3.1. Berikut ini juga akan dijelaskan proses-proses yang terlibat di sistem :



Gambar 3. 2 *FlowChart* Sistem *Augmented Reality*

3.1.3.1 Inisialisasi *Face Detector,* kamera, dan Objek Visual

Di dalam sistem perangkat lunak jilbab virtual ini tahap awal yang harus ditentukan ialah face detector yang akan digunakan, sumber masukan videonya, dan objek visual apa saja yang akan di *load*. Semua hal tersebut tergambar di proses inisialisasi. Inisialisasi ialah proses penentuan awal semua hal yang diperlukan untuk menjalankan proses selanjutnya yaitu *face detector*, objek visual, dan kamera.

Pada bagian inisialisasi ini, objek visual juga diinisialisasi terlebih dahulu karena *loading* objek visual memerlukan waktu yang cukup lama. Objek visual yang akan ditampilkan di *load* terlebih dahulu. Agar perangkat lunak dapat menampilkan objek visual tertentu tanpa mengubah atau membangun ulang aplikasi, diperlukan sebuah konfigurasi untuk menentukan objek visual yang akan di *load*. Contoh konfigurasi yang diperlukan dalam perangkat lunak jilbab virtual itu berisi informasi direktori model jilbab yang digunakan, nama file gambarnya, dan skala yang cocok sehingga penampilan objek bisa lebih proporsional. Barulah setelah itu objek dapat di *load*.

Kamera dan *Face detector* juga merupakan bagian penting pada perangkat lunak ini. *Face detector* akan mengatur hal-hal yang berkaitan dengan pendeteksian. Dalam sistem ini yang berperan sebagai *face detector* ialah *library* Marilena. Sedangkan kamera mulai menginisialisai dari video masukan dari kamera sampai pemprosesan image. Pada perangkat lunak ini hal - hal yang perlu di atur dari *face detector* ini ialah skala yang ingin digunakan dan koordinat posisi

untuk objek visual. Dan untuk kamera akan diatur *frame per rate*, lebar, dan tinggi stage.

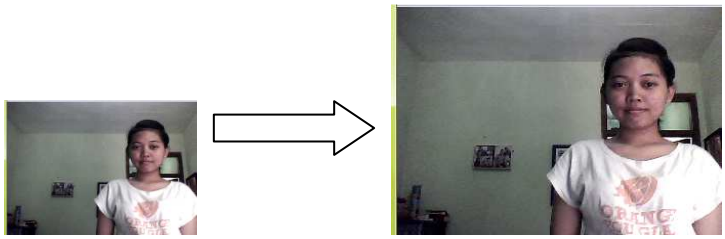
3.1.3.2 *PreProcessing Image*

Pada tahap *preprocessing image* dilakukan proses *scaling* dan *grayscale* untuk menjadi masukan unruk proses selanjutnya dalam metode *Haar Cascade*.

3.1.3.2.1 Tahap *Scaling*

Scaling merupakan proses mengubah ukuran gambar digital menjadi lebih besar ataupun lebih kecil. Untuk sistem akan memanfaatkan *package display object* yang merupakan *package* class miliki flash. *Scaling* dapat dilakukan dengan memanipulasi ukuran tampilan objek dalam dua cara , menggunakan salah satu properti (*width* dan *height*) atau properti *method* skala yaitu scalex dan scaley. Properti skala mewakili ukuran relatif dari tampilan objek dibandingkan dengan ukuran aslinya.

Pada awalnya harus ditentukan dahulu faktor skala untuk membuat layar menampilkan dan memproses image dari kamera sesuai dengan ukuran yang diinginkan. Tahap Scaling ini membutuhkan skala yang tepat karena tahap scaling dilakukan di beberapa proses dan semua nilai skala harus konsisten. Skala yang dirasa tepat dan umum digunakan ialah empat karena menyesuaikan dengan panjang dan lebar layar yang umum digunakan yaitu 500x450, akan tetapi bisa bersifat dinamis dengan mengubah nilai dari variabel faktor skala. Jika faktor skala terlalu besar tidak hanya mempengaruhi terhadap tampilan tetapi juga proses akan menjadi lebih lama. Jika faktor skala terlalu kecil maka penampilan *stage* di flash akan tidak jelas.



Gambar 3. 3 Hasil *Scaling*

3.1.3.2.2 Tahap *Grayscale*

Grayscale merupakan proses untuk mengubah warna gambar menjadi keabu-abuan. Gambar digital yang sudah melalui proses *scaling* akan menjadi masukan dalam proses ini. Proses *grayscale* dilakukan dengan mengubah nilai RGB setiap piksel gambar menjadi 1 nilai yang sama. Dalam perangkat lunak jilbab virtual ini proses pengubahan citra RGB menjadi citra *grayscale* menggunakan sebuah *class* yang dimiliki flash yaitu *ColorMatrixFilter*.

Dalam proses ini menerapkan sebuah efek *grayscale* dengan mengubah semua warna ke setiap ukuran kecerahan yang sama. Acuan atau definisi dari pencahayaan adalah ukuran kecerahan, sehingga dapat mengkonversi warna untuk disetiap *multiplying* setara dengan warna merah , hijau , dan biru disetiap konstanta, dengan nilai masing-masing yaitu 0.3086 , 0.694 , dan 0.0820. Sehingga faktor pencahayaan akan sangat berpengaruh untuk memproses *image*. Nilai-nilai matrik tersebut telah menjadi nilai pasti yang tidak dapat diubah. Berikut ini ialah matrik filter yang digunakan di flash untuk membuat efek *grayscale* :

$$\begin{pmatrix} rc, gc, bc, 0, 0, \\ rc, gc, bc, 0, 0, \\ rc, gc, bc, 0, 0, \\ 0, 0 , 0 , 1, 0 \end{pmatrix}$$

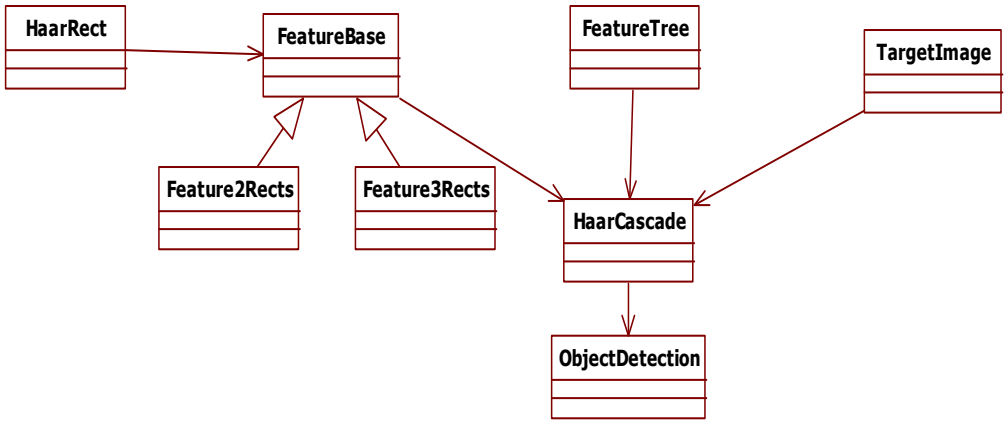
Hasil akhir yang didapatkan dari matrik *grayscale* tersebut dapat dilihat di gambar berikut :



Gambar 3. 4 Hasil *Grayscale*

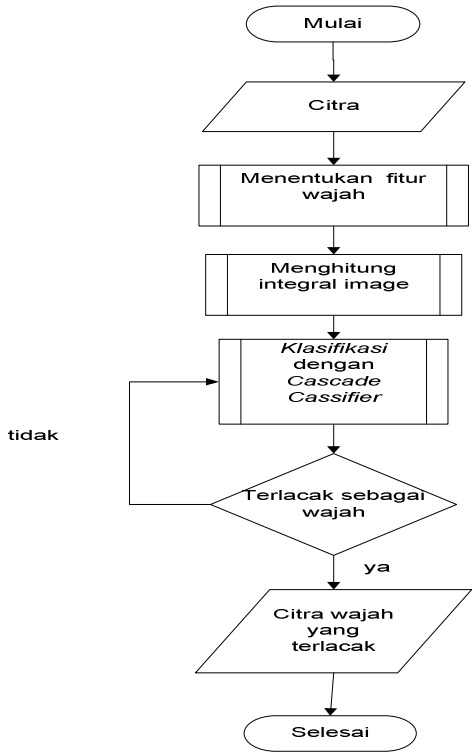
3.1.3.3 Pendeteksian Metode *Haar Cascade Classifier*

Pada penelitian ini digunakan metode *haar cascade classifier* sebagai metode untuk pengenalan pola wajah. Metode ini telah banyak diaplikasikan menggunakan OpenCV, dan sekarang ini telah dapat diaplikasikan dengan flash menggunakan Marilena. Berikut ini *class* di Marilena yang merupakan implementasi *haar cascade classifier* :



Gambar 3. 5 *Class* Marilena

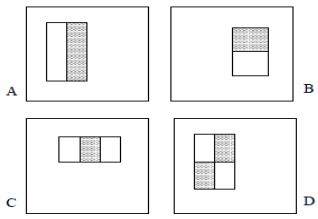
Perangkat lunak ini menggunakan *library* Marilena yang sepenuhnya mengadopsi metode *Haar Cascade Classifier* sebagai pendeteksi pola wajah, berikut ini alur proses metode *haar cascade classifier* yang ada di Marilena:



Gambar 3. 6 metode *haar cascade classifier*

3.1.3.3.1 Proses Menentukan Fitur Wajah

Fitur-fitur ini merupakan gambaran dari wajah manusia yang dikelompokkan berdasarkan sisi yang terang dan sisi yang gelap seperti yang dapat dilihat pada Gambar 3.7. Terdapat tiga jenis fitur berdasarkan jumlah persegi panjang yang terdapat di dalamnya, seperti yang dapat dilihat pada gambar di bawah ini :



Gambar 3. 7 *haar like features* [9]

Di Marilena hanya digunakan dua jenis fitur yaitu fitur (a) dan fitur (c). Fitur tersebut dibuat *class* masing-masing yang bernama Feature2Rects dan Feature3Rects yang merupakan turunan dari *class* FeatureBase. *Class* FeatureBase merupakan *template* yang dapat diturunkan sesuai dengan jenis fitur yang digunakan. Setelah *class* featureBase dapat diturunkan menjadi *class* Feature2Rects yang memiliki dua variabel *HaarRect*. Begitu juga jika menggunakan tiga *haar* maka dibuat class Feature3Rect yang memiliki tiga variabel *HaarRect*.

Setelah melakukan penskalaan sebelumnya kemudian dilakukan pencarian posisi wajah yaitu dengan cara mencari fitur-fitur yang memiliki tingkat pembeda yang tinggi. Fitur yang memiliki batas terbesar antara wajah dan bukan wajah dianggap sebagai fitur terbaik. Contohnya dalam sistem ini ialah pencarian area mata yang menggunakan haar dua fitur seperti yang diperlihatkan gambar 3.8. Area hitam dari fitur mendeteksi bagian mata dan area putihnya mendeteksi bagian kulit.



Gambar 3. 8 contoh pencarian dengan *haar like features*

Berikut ini ialah *class* FeatureBase yang digunakan pada *library* Marilena :

```
public class FeatureBase{
      public  var threshold :Number;
      public  var left_val  :Number;
      public  var right_val :Number;
      public  var next:FeatureBase;

      public function FeatureBase( _th:Number, _lv:Number,
_rv:Number)
            {
                  threshold = _th;
                  left_val  = _lv;
                  right_val = _rv;
            }

      public function getSum( targetImage:TargetImage, x:int,
y:int ):Number{
                  return 0;
      }

      public function setScaleAndWeight(s:Number,w:Number):void{
      }
}
```

3.1.3.3.2 Proses Menghitung *Integral Image*

Untuk memudahkan proses perhitungan nilai dari setiap fitur *Haar* pada setiap lokasi gambar digunakan teknik yang disebut *integral image*. Pada gambar 3.7 dapat dilihat bahwa fitur a dan b terdiri dari dua persegi panjang, sedangkan fitur c terdiri dari tiga persegi panjang dan fitur d empat persegi panjang. Cara menghitung nilai dari fitur ini adalah mengurangkan nilai piksel pada area hitam dengan piksel pada area putih.

Dalam perangkat lunak ini sebagai contoh untuk perhitungan area mata dapat dilihat di gambar berikut :



Gambar 3. 9 Contoh proses deteksi

Di Marilena proses ini melibatkan *class* TargetImage, dalam *class* tersebut dihitung jumlah tiap piksel dilakukan secara iteratif mulai dari ujung kiri atas citra hingga ujung kanan bawah dengan pergeseran sebesar Δx dan Δy. Semakin kecil nilai Δx dan Δy, maka semakin akurat pula proses deteksi. Nilai Δx dan Δy yang sering digunakan adalah satu.

Sebagai contoh perhitungan *integral image* dengan angka-angka piksel *sampel* dapat disimak di dalam gambar tabel matrik di bawah ini :

1	2	3
2	3	4
4	5	1

1	3	6
3	8	15
7	17	25

1	1+2	1+2+3
1+2	1+2+2+3	1+2+3+2+3+4
1+2+4	1+2+2+3+4+5	1+2+3+2+3+4+4+5+1

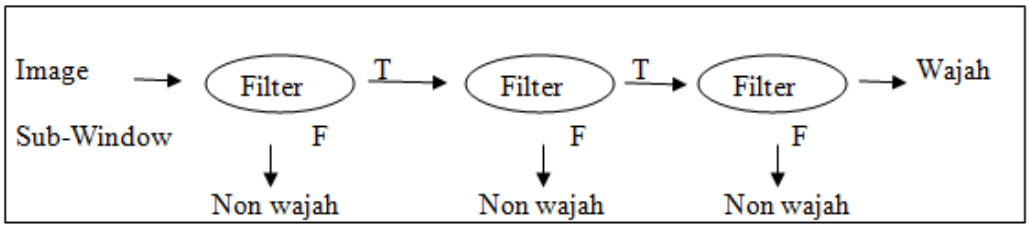
Dari perhitungan *integral image* diatas, maka diperolah nilai integral fitur putih= { 3,8,15 }dan nilai integral fitur hitam = { 7,17,25}. Maka nilai *haar feature* tersebut adalah :

$$\mathbf{f(x) = (7 + 17 + 25) - (3 + 8 + 15) = 23}$$

3.1.3.3.3 Proses Membuat *Cascade Classifier*

Karakteristik metode ini adalah adanya klasifikasi bertingkat. Klasifikasi pada algoritma ini memiliki tingkatan dimana tiap tingkatan mengeluarkan

subcitra yang diyakini bukan wajah. Di bawah ini adalah alur kerja dari klasifikasi bertingkat.



Gambar 3. 10 Proses *Cascade Classifier*

Sesuai dengan yang digambarkan oleh gambar 3.10 maka pencarian akan terus berlangsung seperti gambar tersebut jika belum mendekati citra sempel. Penerapan proses *cascade classifier* di *library* Marilena berada di *class* HaarCascade. Di *class* HaarCascade terdapat nilai *threshold* yang merupakan data hasil uji coba yang bersifat statis yang nantinya akan dijadikan nilai pembanding untuk proses pengklasifikasian wajah. Sedangkan *cascade classifier* ini juga melibatkan *class* FeatureTree yang menampung nilai *threshold* yang ditangkap kamera dan akan dibandingkan dengan nilai *threshold* yang telah ada di *class* HaarCascade. Jika nilai dari hasil perbedaanya di atas dari ambang batas maka fitur tersebut dapat dikatakan ada.

Nilai ambang batas pada setiap filter diatur setiap saat proses filter terjadi, sehingga nilai dari ambang batas bersifat dinamis. Filter pada setiap tingkat telah dilatih untuk mengklasifikasi citra yang telah melalui tahap sebelumnya. Saat proses pengklasifikasian, jika salah satu filter gagal terlewati maka citra dapat dikatakan sebagai daerah yang bukan wajah. Jika citra belum cukup untuk dikatakan sebagai *strong classifier* maka proses diulang sampai bobot terpenuhi dengan menaikkan nilai dari ambang batas. Ketika citra dapat

melewati setiap filter yang ada didalam rantai, maka dapat dikatakan daerah tersebut merupakan wajah. Berikut ini potongan *pseudocode* dari *class* HaarCascade:

```
1.  function run (r: Rectangle) ← integer
2.  (I.S : nilai rectangle dari strong classifier telah
    dihasilkan)
3.  (F.S : menghasilkan classifier terbaik)
4.
5.  Kamus :
6.  x,y,w,h : integer
7.  mean : real
8.  variance_norm_factor : real
9.  features : array of real
10. val, sum, st_th : real
11. i,j : integer
12. feature : FeatureBase
13. tree : FeatureTree
14.
15. Algoritma :

16. x ← r↑.x
17. y ← r↑.y
18. w ← r↑.width
19. h ← r↑.height
20.
21. mean ← targetImage.getSum(x,y,w,h) *  inv_window_area
22. variance_norm_factor ← targetImage.getSum2(x,y,w,h)*
23. inv_window_area - mean*mean
24.
25. if(variance_norm_factor >= 0) then
26.    variance_norm_factor ← Math.sqrt(variance_norm_factor)
27. else
28.    variance_norm_factor ← 1
29. endif
30.
31. while (tree ≠ nil) do
32.    feature ← tree.firstFeature
33.    val ← 0
34.    st_th ← tree.stage_threshold
35.
36.    while (feature  ≠ nil) do
37.          sum ← feature.getSum(targetImage, x, y)
38.
39.          if (sum < feature.threshold * variance_norm_factor)
40.          then
41.                val = val + feature.left_val
42.          else
43.                val = val + feature.right_val
44.          endif
45.
46.          if (val > st_th) then
```

```
47.                 break
48.           endif
49.           feature ← feature.next
50.     endwhile
51.
52.     if (val < st_th) then
53.           return 0
54.     endif
55.     tree ← tree.next
56. endwhile
57. return 1
58. endfunction
```

Kompleksitas potongan *pseudocode* di atas dapat dianalisis dengan metode Big O, berikut tabel analisis Big O :

Tabel 3. 1 Analisis Big O

Pseudocode	**Nilai Big O**
1. x ← r↑.x	O(1)
2. y ← r↑.y	O(1)
3. w ← r↑.width	O(1)
4. h ← r↑.height	O(1)
5. mean ← targetImage.getSum(x,y,w,h) * inv_window_area	O(1)
6. variance_norm_factor ← targetImage.getSum2(x,y,w,h)* inv_window_area - mean*mean	O(1)
7. if(variance_norm_factor >= 0) then	O(1)
8. variance_norm_factor ← Math.sqrt(variance_norm_factor)	O(1)
9. else	
10. variance_norm_factor ← 1	O(1)
11. endif	
12. while (tree ≠ nil) do	O(n)
13. feature ← tree.firstFeature	O(1)
14. val ← 0	O(1)
15. st_th ← tree.stage_threshold	O(1)
16. while (feature ≠ nil) do	O(n)
17. sum ← feature.getSum(targetImage, x, y)	O(1)
18. if (sum < feature.threshold * variance_norm_factor) then	O(1)
19. val = val + feature.left_val	O(1)
20. else	
21. val = val + feature.right_val	O(1)
22. endif	
23. if (val > st_th) then	O(1)
24. break	O(1)
25. endif	
26. feature ← feature.next	O(1)
27. endwhile	
28. if (val < st_th) then	O(1)

29. return 0	O(1)
30. endif	
31. tree ← tree.next	O(1)
32. endwhile	
33. return 1	O(1)
34. endfunction	
Jumlah	$O(n^2)$

Setelah melalui beberapa proses penyaringan akhirnya didapatkan posisi wajah yang fix seperti diperlihatkan gambar 3.10 berikut ini:



Gambar 3. 11 Posisi wajah terdeteksi

3.1.3.4 Proses Menampilkan ke *Stage*

Setelah wajah ditemukan maka, objek visual di *load* atau dimunculkan di posisi wajah, posisi dan rotasi objek visual akan mengikuti *tracker*. Setiap objek akan diatur posisinya sesuai dengan koordinat yang didapat dari deteksi objek. Setelah objek di *load* maka objek akan ditampilkan ke *stage* utama. Berikut contoh kode program untuk me-*load* dan menampilkan ke *stage* :

```
1  private function objekLoad():void
2  {
3      Mask.load(new URLRequest("oakley.png"));
4      addChild(Mask);
5      oakley_th.load(new URLRequest("oakley_th.png"));
6      MCoakley_th.addChild(oakley_th);
7      MCoakley_th.x = 0;
8      MCoakley_th.y = stage.height - 64;
9      MCoakley_th.name = "oakley.png"
10     MCoakley_th.buttonMode = true
11     MCoakley_th.addEventListener(MouseEvent.CLICK, click);}
```

3.1.4 Analisis Spesifikasi Kebutuhan Perangkat Lunak

Analisis spesifikasi kebutuhan perangkat lunak dibagi menjadi dua bagian yaitu SKPL-F (Spesifikasi kebutuhan perangkat lunak fungsional) dan SKPL-NF (Spesifikasi kebutuhan perangkat lunak non-fungsional). Analisis tersebut berisi deskripsi dari kebutuhan perangkat lunak yang akan dibangun baik kebutuhan fungsional maupun kebutuhan non fungsional. Berikut ini adalah tabel dari spesifikasi kebutuhan perangkat lunak yang akan dibangun :

3.1.4.1 Analisis Spesifikasi Kebutuhan Fungsional

Analisis spesifikasi kebutuhan fungsional berisi pernyataan dari layanan sistem yang harus disediakan, bagaimana sistem harus bereaksi terhadap input tertentu dan bagaimana sistem harus berperilaku dalam situasi tertentu. Intinya menjelaskan fungsionalitas dari sistem yang akan dibangun secara detail, berikut ini table spesifikasi kebutuhan fungsional :

Tabel 3. 2 Spesifikasi Kebutuhan Fungsional

Nomor	Spesifikasi Kebutuhan Fungsional
SRS-F-001	Sistem harus bisa terhubung dengan kamera yang terpasang di komputer.
SRS-F-002	Sistem harus bisa mendeteksi posisi wajah yang terdapat di depan kamera.
SRS-F-003	Sistem harus bisa menyediakan pilihan model jilbab lebih dari satu model.
SRS-F-004	Sistem harus bisa menampilkan model jilbab ke posisi wajah yang diketahui di layar.
SRS-F-005	Sistem harus bisa mengambil foto user setelah ataupun sebelum menggunakan jilbab virtual.
SRS-F-006	Sistem harus bisa menyediakan layanan pemesanan jilbab yang sedang dicoba.
SRS-F-007	Sistem harus bisa menyediakan layanan input dan hapus data model jilbab oleh admin.

3.1.4.2 Analisis Spesifikasi Kebutuhan Non-Fungsional

Analisis spesifikasi kebutuhan non fungsional adalah deskripsi mengenai batasan dari layanan dan fungsi-fungsi dari sistem yang akan dibangun. Berikut ialah table spesifikasi kebutuhan non fungsional perangkat lunak yang akan dibangun :

Tabel 3. 3 Spesifikasi Kebutuhan Non-Fungsional

Nomor	Spesifikasi Kebutuhan Non Fungsional
SRS-NF-001	Sistem membutuhkan minimum spesifikasi kamera 1,3 megapiksel.
SRS-NF-002	Sistem dapat mendeteksi wajah dengan baik dengan posisi wajah menghadap tegak ke kamera .
SRS-NF-003	Sistem hanya dapat mendeteksi jarak maksimal 80 centimeter.
SRS-NF-004	Sistem membutuhkan capture minimal kepala sampai dada.
SRS-NF-005	Wajah yang akan dideteksi harus terlihat jelas atau tidak tertutup rambut.
SRS-NF-006	Sistem membutuhkan cahaya yang berada di depan kamera.
SRS-NF-007	Sistem harus bisa dikelola hanya oleh admin.
SRS-NF-008	Objek virtual yang dapat dikelola admin hanyalah yang berekstensi *.PNG .
SRS-NF-009	File jilbab yang akan diupload maksimal berukuran 200 Kb dan thumbnails maksimal berukuran 100 Kb.
SRS-NF-010	Sistem hanya dibangun dalam bentuk *website*.

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah sebuah langkah untuk menganalisis sumber daya manusia yang akan menggunakan perangkat lunak yang dibangun, perangkat keras dan perangkat lunak yang dimiliki sesuai dengan kebutuhan atau permintaan, sehingga dapat ditentukan kompabilitas perangkat lunak yang dibangun terhadap sumber daya yang ada. Analisis kebutuhan non fungsional harus bisa mencakup kebutuhan dan fakta yang ada.

3.1.5.1 Analisis Perangkat Keras

Analisis perangkat keras dimaksudkan untuk mengetahui spesifikasi perangkat keras yang digunakan di Toko Sakinah Kerudung. Perangkat keras yang sedang digunakan untuk mengelola toko adalah sebagai berikut :

1.Processor AMD dengan kecepatan 2.30 GHz

2.RAM 2 Gigabyte

3.Monitor 14 inci

4.VGA on board

5.Koneksi internet 384 Kbps

Perangkat keras yang digunakan untuk pembangunan jilbab virtual ini adalah sebagai berikut :

1. Processor AMD Turion X2 ultra dual core 2.10 GHz
2. RAM 4 gigabyte
3. Monitor 14.1 inci
4. VGA ATI Radeon 3100 Graphics
5. Webcam 2 Megapiksel
6. Koneksi internet 2 Mbps

Rekomendasi spesifikasi minimal perangkat keras untuk mengakses perangkat lunak jilbab virtual di website sakinah kerudung ini adalah sebagai berikut :

1. Processor dengan kecepatan minimal 1.8 GHz
2. RAM minimal 1 Gigabyte
3. Monitor minimal 12 inci

4. VGA *Card* minimal 128 Megabyte
5. Webcam minimal 1,3 Megapiksel
6. Koneksi internet minimal 153 Kbps

Berdasarkan analisis perangkat keras diatas spesifikasi komputer yang ada sudah mencukupi, hanya perlu menambah webcam minimal 1,3 Megapiksel sebagai perangkat keras yang harus disediakan oleh Toko Sakinah Kerudung.

3.1.5.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan di Toko Sakinah Kerudung adalah sebagai berikut :

1. Sistem operasi Windows 7
2. Web Browser Mozilla Firefox

Perangkat lunak yang digunakan dalam pembangunan jilbab virtual ini adalah sebagai berikut :

1. Sistem operasi Windows 7
2. IDE FlashDevelop
3. Library Marilena
4. Flex Builder 4.0
5. Adobe Flash Player 11.1
6. Adobe Flash Professional CS5
7. Adobe Photoshop CS3

Berdasarkan analisis perangkat lunak diatas, rekomendasi minimum perangkat lunak untuk mengakses perangkat lunak jilbab virtual diantaranya sebagai berikut :

1. Adobe Flash Player minimal versi 10.
2. Web Browser Mozilla Firefox minimal versi 4.0 atau Internet Explorer minimal versi 7.0 atau Google Chrome Safari minimal versi 5.0 atau Opera minimal versi 11 .

Berdasarkan analisis perangkat lunak diatas, masih ada kekurangan yang harus diadakan oleh pihak Toko Sakinah Kerudung diantaranya sebagai berikut :

1. Adobe Flash Player minimal versi 10
2. Adobe Photoshop minimal CS3

3.1.5.3 Analisis Pengguna

Selama ini pengunjung website sakinah kerudung berasal dari berbagai kalangan dengan berbagai latar belakang. Sedangkan karakteristik pengguna yang dibutuhkan untuk bisa menggunakan perangkat lunak ini yaitu seperti berikut :

1. Pengguna minimal pernah menggunakan perangkat lunak sejenis.
2. Pengguna tidak memiliki kelainan buta warna.

Sedangkan kemampuan yang harus dimiliki admin untuk mengelola perangkat lunak ini yaitu sebagai berikut :

1. Admin harus memiliki keterampilan menggunakan photoshop.
2. Admin harus mengerti mengkonfigurasi file xml.
3.

3.1.6 Analisis Data

Analisis data yang digunakan di sistem ini ialah pada file xml yang akan digunakan untuk menyimpan data-data objek. Berikut ialah struktur umumnya :

```
1  <?xml version=”1.0″ encoding=”ISO-8859-1″?>
2  <images>
3   <image>
4     <name>        </name>
5     <file>        </file>
6     <thumb>       </thumb>
7   </image>
   </images>
```

Berikut ini ialah deskripsi dari tag-tag xml yang digunakan :

Tabel 3. 4 Deskripsi tag xml

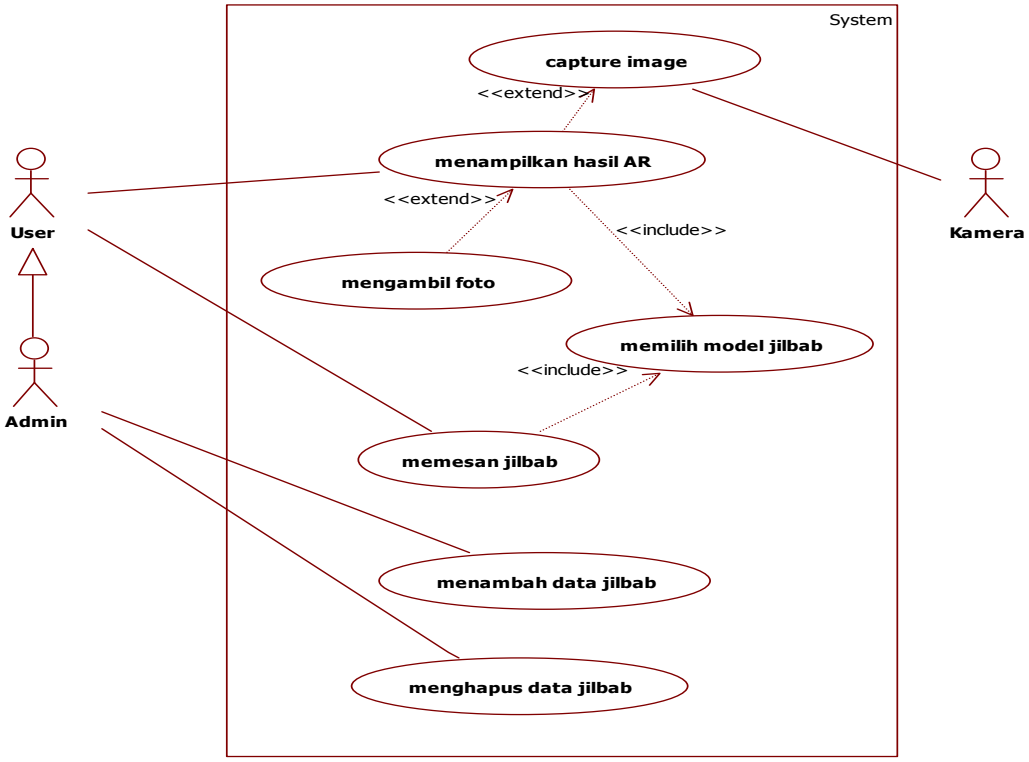
Tag XML	Fungsi
<image> </image>	Merupakan root xml, dimana setiap data akan dimulai dari root ini.
<name> </name>	Merupakan nama jilbab yang bersangkutan
<file> </file>	Merupakan alamat gambar jilbab
<thumb> </thumb>	Merupakan alamat thumbnails jilbab

3.1.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional diperlukan untuk memodelkan sistem. Pemodelan yang digunakan untuk memodelkan perangkat lunak jilbab virtual ini adalah pemodelan berorientasi objek. Perangkat lunak ini dimodelkan menggunakan UML (Unified Modeling Language).

3.1.7.1 *Use Case Diagram*

Didalam sistem hanya terdapat beberapa pengguna yaitu pengunjung web yang dinamakan *user*, admin, dan webcam. Peran aktor-aktor yang ada dapat terlihat pada diagram *Use Case* pada gambar berikut.



Gambar 3. 12 *Use Case* Diagram

3.1.7.1.1 Skenario *Use Case Capture Image*

Use case Capture Image ini menggambarkan keterlibatan aktor kamera yang berperan untuk menangkap *image* yang berada didepan kamera untuk selanjutnya diproses oleh sistem. Berikut ini ialah skenario *use case capture image* :

Tabel 3. 5 Skenario *Use Case Capture Image*

Nama	*Capture image*
Aktor	Kamera
Trigger	User mengaktifkan kamera
Skenario Utama	
Kondisi Awal	Sistem telah hidup
Aksi Aktor	**Reaksi Sistem**
1. Kamera mengirim permintaan koneksi.	
	2. Sistem mengirim respon jawaban *allow* dari user.
3. Kamera mengirim gambar yang ditangkap ke sistem.	
	4. Sistem mengolah gambar dari kamera.

	5. Sistem menampilkan gambar yang telah diolah ke layar.
	6. Sistem melakukan pendeteksian wajah.
Kondisi Akhir	Gambar user beserta posisi wajah dapat ditampilkan ke layar.
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
1. Kamera mengirim permintaan koneksi	
	2. Sistem mengirim respon jawaban *deny* dari user
	3. Kamera mati.
Kondisi Akhir	Sistem tidak dapat menampilkan apapun ke layar

3.1.7.1.2 Skenario *Use Case* Memilih Model Jilbab

Use case memilih model jilbab ini menggambarkan proses dimana *user* harus memilih model jilbab terlebih dahulu sebelum melihat hasil *augmented reality*.

Tabel 3. 6 Skenario *Use case* Memilih Model Jilbab

Nama	Memilih model jilbab	
Aktor	*User*	
Trigger	*Use*r memilih gambar jilbab yang ada di menu jilbab	
Skenario Utama		
Kondisi Awal	Kamera dan sistem telah terhubung	
Aksi Aktor		**Reaksi Sistem**
1. User memilih salah satu gambar jilbab di menu.		
Aksi Aktor		**Reaksi Sistem**
		2. Sistem mencari alamat file gambar
		3. Sistem me*load* gambar
Kondisi Akhir	Sistem berhasil me-*load* gambar ke layar	

3.1.7.1.3 Skenario *Use case* Menampilkan Hasil AR

Use case menampilkan hasil AR ini me-*include* dari *use case* memilih model jilbab karena setiap *user* ingin melihat hasil AR, *user* harus memilih model jilbab terlebih dahulu.

Tabel 3. 7 Skenario *Use case* Menampilkan Hasil AR

Nama	Menampilkan hasil AR	
Aktor	*User*	
Trigger	*User* menghadap tegak ke kamera	
Skenario Utama		
Kondisi Awal	Model jilbab yang dipilih user telah berhasil di-*load*	
Aksi Aktor		**Reaksi Sistem**
1. User menghadap tegak ke kamera.		
		2. Sistem menentukan koordinat jilbab sesuai posisi wajah.
3. User melakukan pergerakan.		
		3. Sistem memindahkan jilbab sesuai posisi wajah yang terbaca.
Kondisi Akhir	User dapat melihat model jilbab yang telah dikenakan di layar.	
Skenario Alternatif		
Aksi Aktor		**Reaksi Sistem**
1. Posisi wajah user tidak menghadap kamera		
		2. Sistem tidak dapat menempatkan jilbab di wajah.
Kondisi Akhir	Gambar jilbab tidak dapat mengikuti posisi wajah.	

3.1.7.1.4 Skenario *Use Case* Mengambil Foto

Use case mengambil foto dapat dijalankan baik sebelum *use case* memilih jilbab dilakukan ataupun sesudahnya.

Tabel 3. 8 Skenario *Use Case* Mengambil Foto

Nama	Mengambil foto	
Aktor	*User*	
Trigger	*User* memilih menu foto	
Skenario Utama		
Kondisi Awal	Sistem dapat menampilkan gambar dari kamera baik yang telah ada jilbab virtual maupun belum.	
Aksi Aktor		**Reaksi Sistem**
1. User memilih menu foto		
		2. Sistem menyalin gambar piksel di layar
		3, Sistem menampungnya sementara
Kondisi Akhir	Sistem dapat mengambil gambar user secara *real time* baik setelah menggunakan jilbab virtual ataupun belum.	

3.1.7.1.5 Skenario *Use Case* Memesan Jilbab

Use case memesan jilbab juga me-*include use case* memilih model jilbab karena sistem harus tahu data model jilbab yang akan dimasukan ke keranjang belanja.

Tabel 3. 9 Skenario *Use Case* Memesan Jilbab

Nama	Memesan jilbab	
Aktor	*User*	
Trigger	*User* memilih menu memesan jilbab	
Skenario Utama		
Kondisi Awal	*User* telah memilih jilbab virtual yang dicobanya	
Aksi Aktor		**Reaksi Sistem**
1. User memilih menu pesan		
		2. Sistem mengambil data jilbab tersebut untuk di kirim ke table keranjang belanja
Kondisi Akhir	Keranjang belanja user telah terisi item jilbab yang dicoba user.	

3.1.7.1.6 Skenario *Use Case* Menambah Model Jilbab

Use case ini berisi proses dimana admin dapat menambah data jilbab yang ingin ditampilkan.

Tabel 3. 10 Skenario *Use Case* Menambah Model Jilbab

Nama	Menambah data jilbab	
Aktor	Admin	
Trigger	Admin membuka form input data jilbab	
Skenario Utama		
Kondisi Awal	Admin telah melakukan login ke website	
Aksi Aktor		**Reaksi Sistem**
1. Admin mengupload model jilbab dengan size lebih kecil dari 900Kb		
		2. Sistem mencek size gambar
		3. Sistem menyalin gambar jilbab ke server.
4. Admin mengupload gambar *thumbnails* dengan size lebih kecil dari 500 Kb		

	5. Sistem mencek size gambar
	6. Sistem menyalin gambar thumbnails ke server
7. Admin mengisi nama jilbab	
	8. Sistem menampung data
9. Admin menekan tombol simpan	
	10. Sistem menyimpan data nama, alamat model jilbab, dan thumbnails ke xml.
Kondisi Akhir	Sistem dapat menyimpan data baru ke xml
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
1. Admin mengupload model jilbab dengan size lebih dari 900Kb	
Aksi Aktor	**Reaksi Sistem**
	2. Sistem mencek size gambar
	3. Sistem memberikan peringatan upload gagal.
4. Admin mengupload gambar *thumbnails* dengan size lebih dari 900Kb	
	5. Sistem mencek size gambar
	6. Sistem memberikan peringatan upload gagal.
7. Admin mengisi nama jilbab	
	8. Sistem menampung nama jilbab
9. Admin menekan tombol simpan	
	10. Sistem memberikan pesan proses simpan gagal
Kondisi Akhir	Admin gagal menambahkan data ke xml

3.1.7.1.7 Skenario *Use Case* Menghapus Data Jilbab

Use case ini berisi proses dimana admin dapat menghapus data jilbab yang diinginkan. Berikut skenario proses menghapus data jilbab :

Tabel 3. 11 Skenario *Use Case* Menghapus Data Jilbab

Nama	Menghapus data jilbab
Aktor	Admin
Trigger	Admin membuka form hapus data jilbab
Skenario Utama	
Kondisi Awal	Admin telah melakukan login ke website

Aksi Aktor	Reaksi Sistem
1. Admin memilih nama jilbab	
Aksi Aktor	**Reaksi Sistem**
	2. Sistem menampilkan *thumbnails* jilbab yang dipilih
Aksi Aktor	**Reaksi Sistem**
3. Admin menekan tombol hapus	
	4. Sistem menghapus data xml yang dipilih.
Kondisi Akhir	Sistem berhasil menghapus data di xml

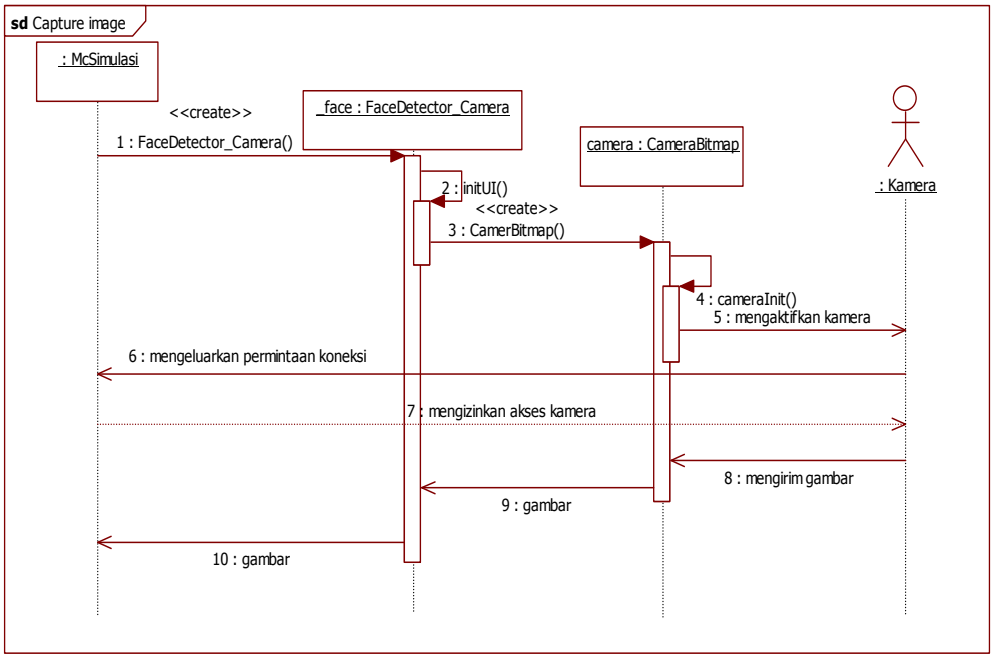
3.1.7.2 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Di bawah ini merupakan *sequence* dari *use case* dan skenario yang sebelumnya telah dijelaskan.

3.1.7.2.1 *Sequence Capture Image*

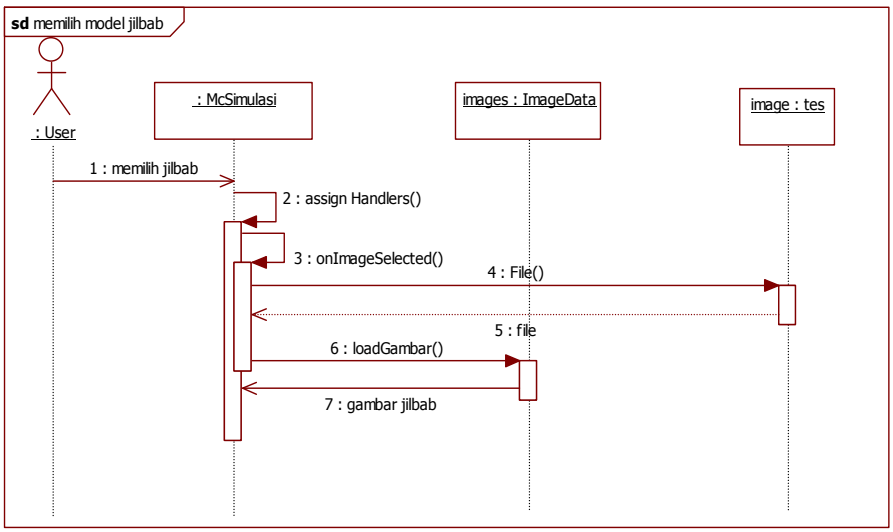
Sequence capture image menggambarkan cara berkomunikasi antar kelas untuk mewujudkan proses *capture image*. Berikut ini gambar *sequence capture image* :



Gambar 3. 13 *Sequence Capture Image*

3.1.7.2.2 *Sequence* Memilih Model Jilbab

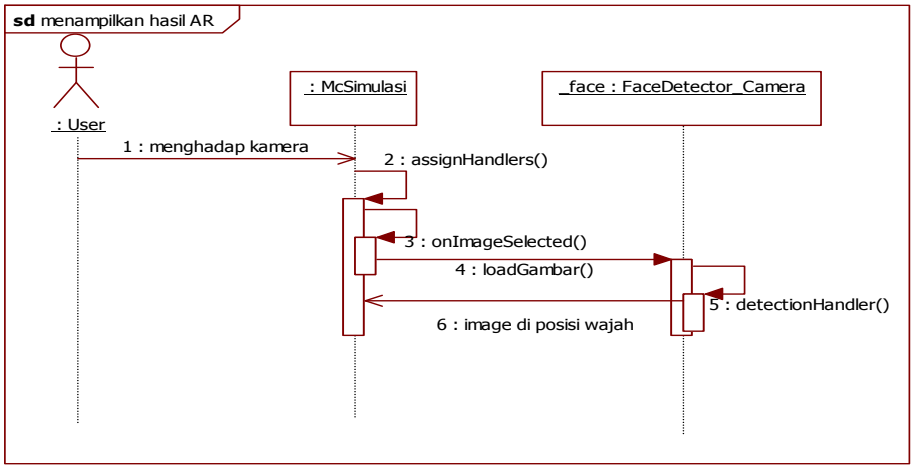
Sequence memilih model jilbab menggambarkan cara berkomunikasi antar kelas untuk mewujudkan proses dimana user memilih model jilbab. Berikut ini gambar *sequence* memilih model jilbab :



Gambar 3. 14 Sequence Memilih Model Jilbab

3.1.7.2.3 *Sequence* Menampilkan Hasil AR

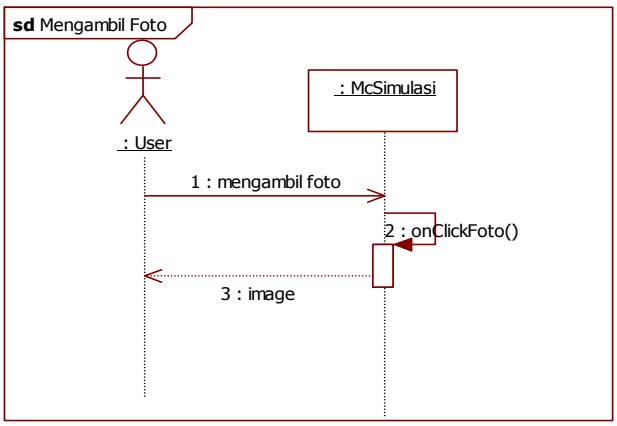
Sequence menampilkan hasil AR menggambarkan cara berkomunikasi antar kelas untuk mewujudkan proses tersebut. Berikut ini gambar *sequence* menampilkan hasil AR:



Gambar 3. 15 *Sequence* Menampilkan Hasil AR

3.1.7.2.4 *Sequence* Mengambil Foto

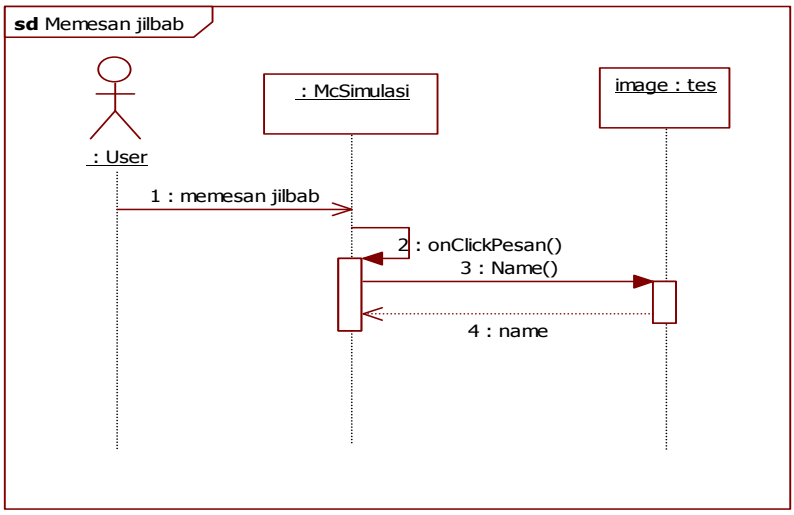
Sequence mengambil foto menggambarkan cara berkomunikasi antar kelas untuk mewujudkan proses tersebut. Berikut ini gambar *sequence* mengambil foto :



Gambar 3. 16 *Sequence* Mengambil Foto

3.1.7.2.5 *Sequence* Memesan Jilbab

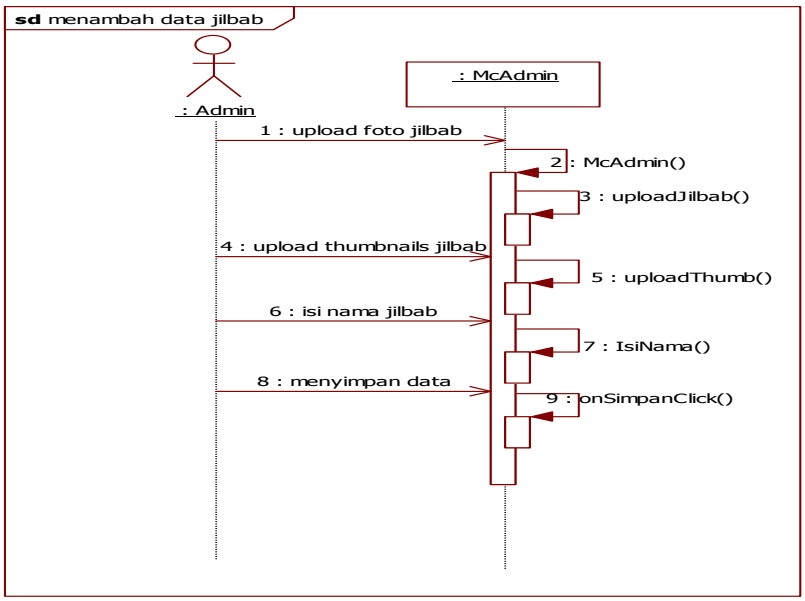
Sequence memesan jilbab menggambarkan cara berkomunikasi antar kelas untuk mewujudkan proses tersebut. Berikut ini gambar *sequence* memesan jilbab :



Gambar 3. 17 *Sequence* memesan jilbab

3.1.7.2.6 *Sequence* Menambah Data

Sequence menambah data menggambarkan cara berkomunikasi antar kelas untuk mewujudkan proses tersebut. Berikut ini gambar *sequence* menambah data :

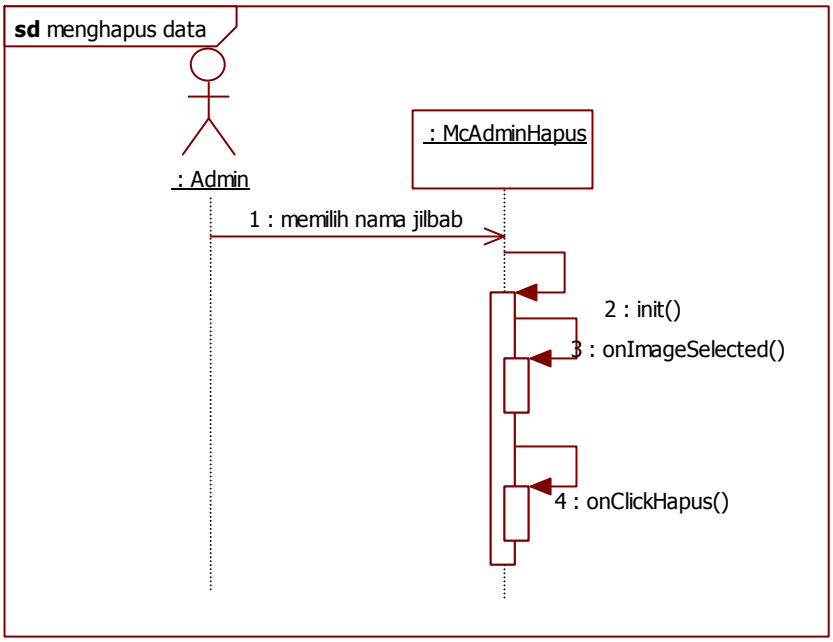


Gambar 3. 18 *Sequence* menambah data

3.1.7.2.7 *Sequence* Menghapus Data

Sequence menghapus data menggambarkan cara berkomunikasi antar kelas untuk mewujudkan proses tersebut. Berikut ini gambar *sequence* menghapus data:

:



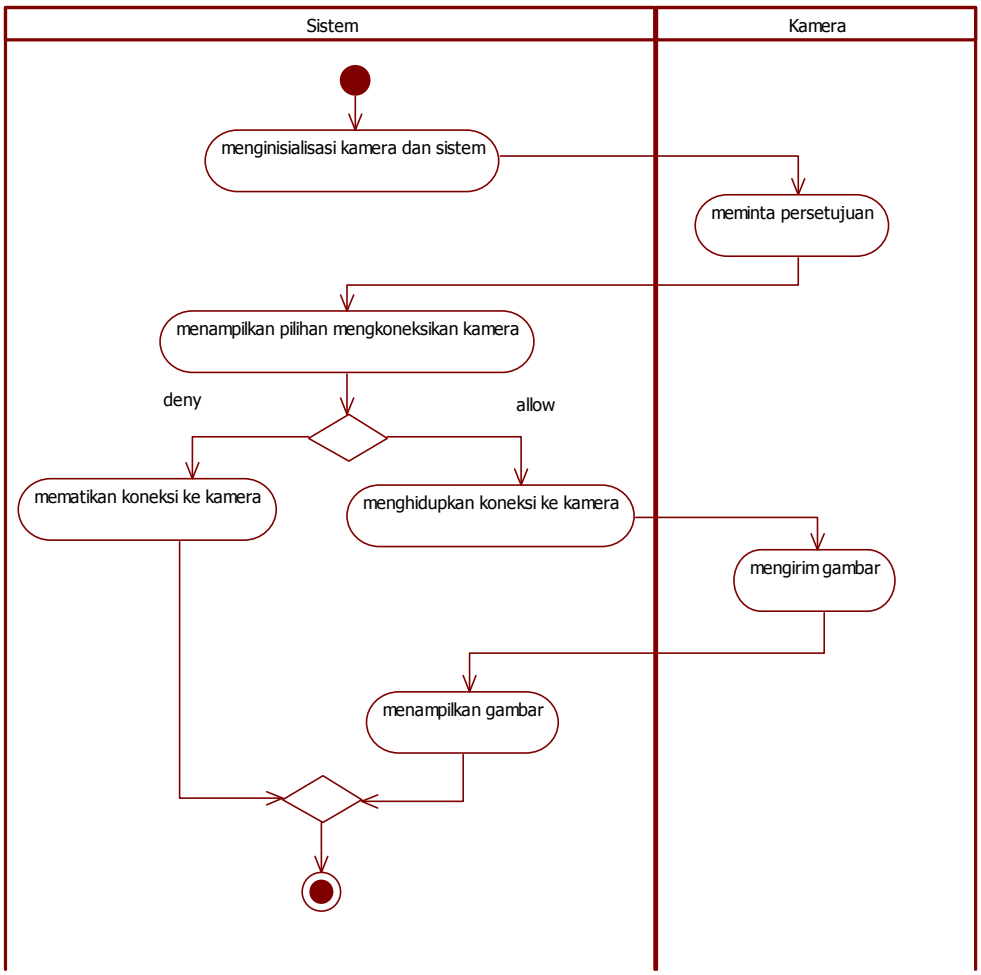
Gambar 3. 19 *Sequence* menghapus data

3.1.7.3 *Activity Diagram*

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

3.1.7.3.1 *Activity Diagram Capture Image*

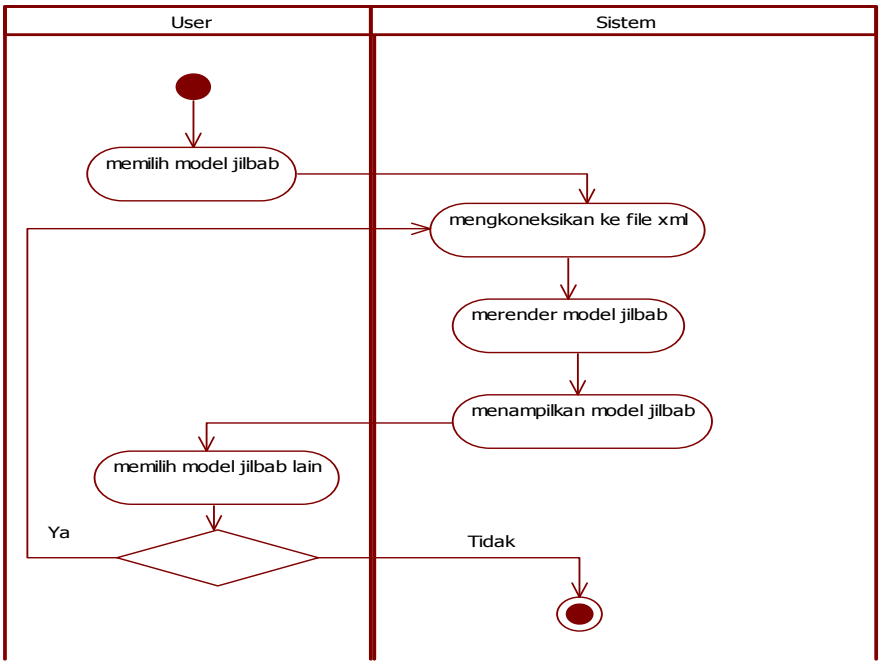
Proses *capture image* memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 20 ***Activity Diagram Capture Image***

3.1.7.3.2 *Activity Diagram* Memilih Model Jilbab

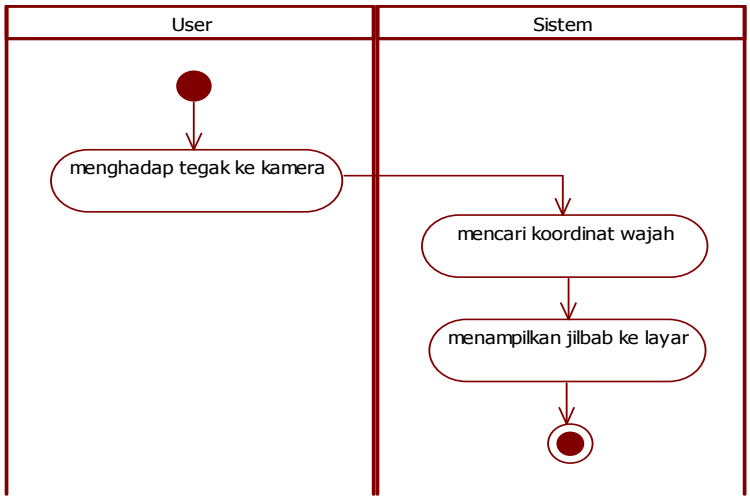
Proses memilih model jilbab memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 21 *Activity Diagram* Memilih Model Jilbab

3.1.7.3.3 *Activity Diagram* Menampilkan Hasil AR

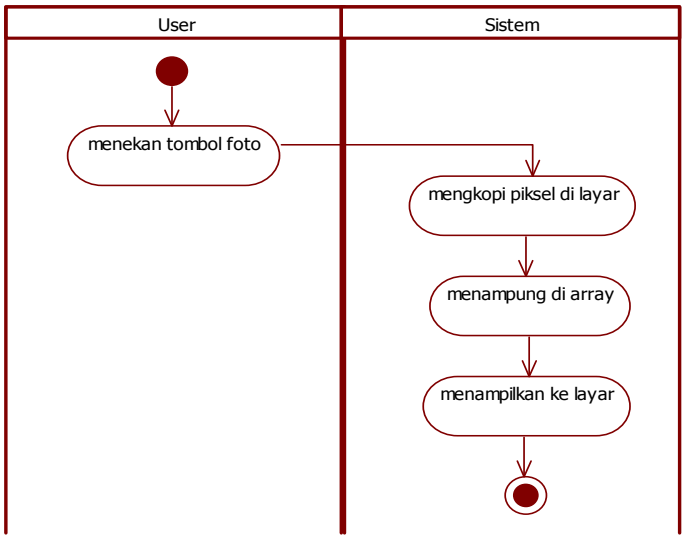
Proses menampilkan hasil AR memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 22 *Activity Diagram* Menampilkan Hasil AR

3.1.7.3.4 *Activity Diagram* Mengambil Foto

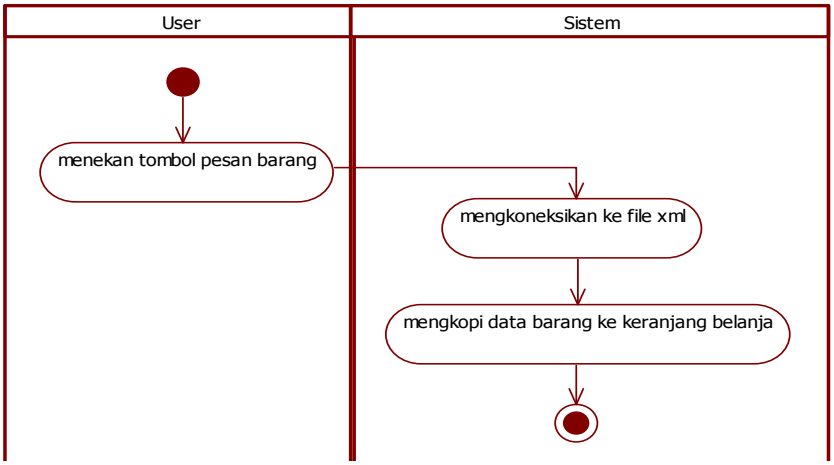
Proses mengambil foto memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 23 *Activity Diagram* Mengambil Foto

3.1.7.3.5 *Activity Diagram* Memesan Jilbab

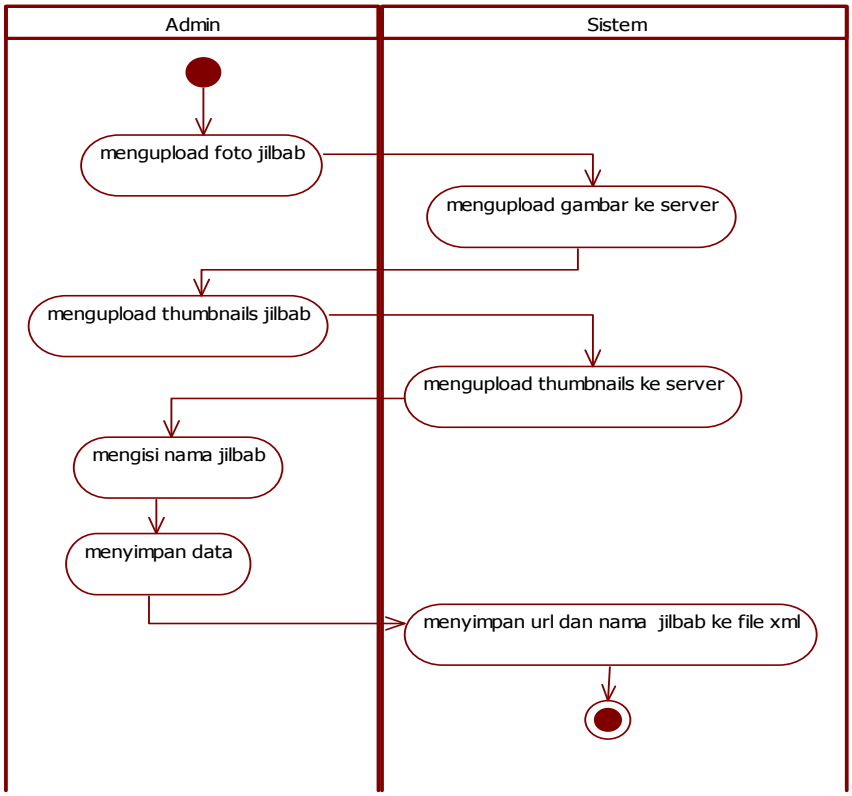
Proses memesan jilbab memiliki alur proses seperti digambarkan sebagai berikut:



Gambar 3. 24 *Activity Diagram* Memesan Jilbab

3.1.7.3.6 *Activity Diagram* Menambah Model Jilbab

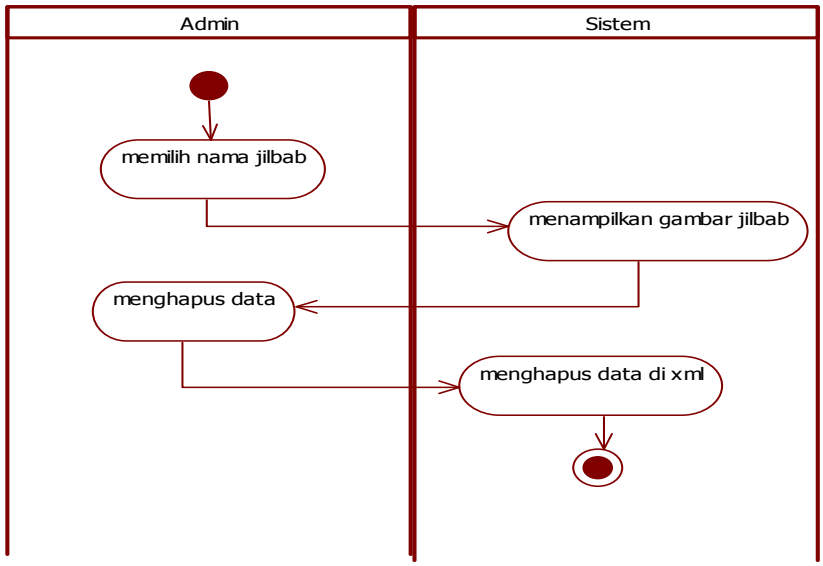
Proses menambah model jilbab memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 25 *Activity Diagram* Menambah Model Jilbab

3.1.7.3.7 *Activity Diagram* Menghapus Model Jilbab

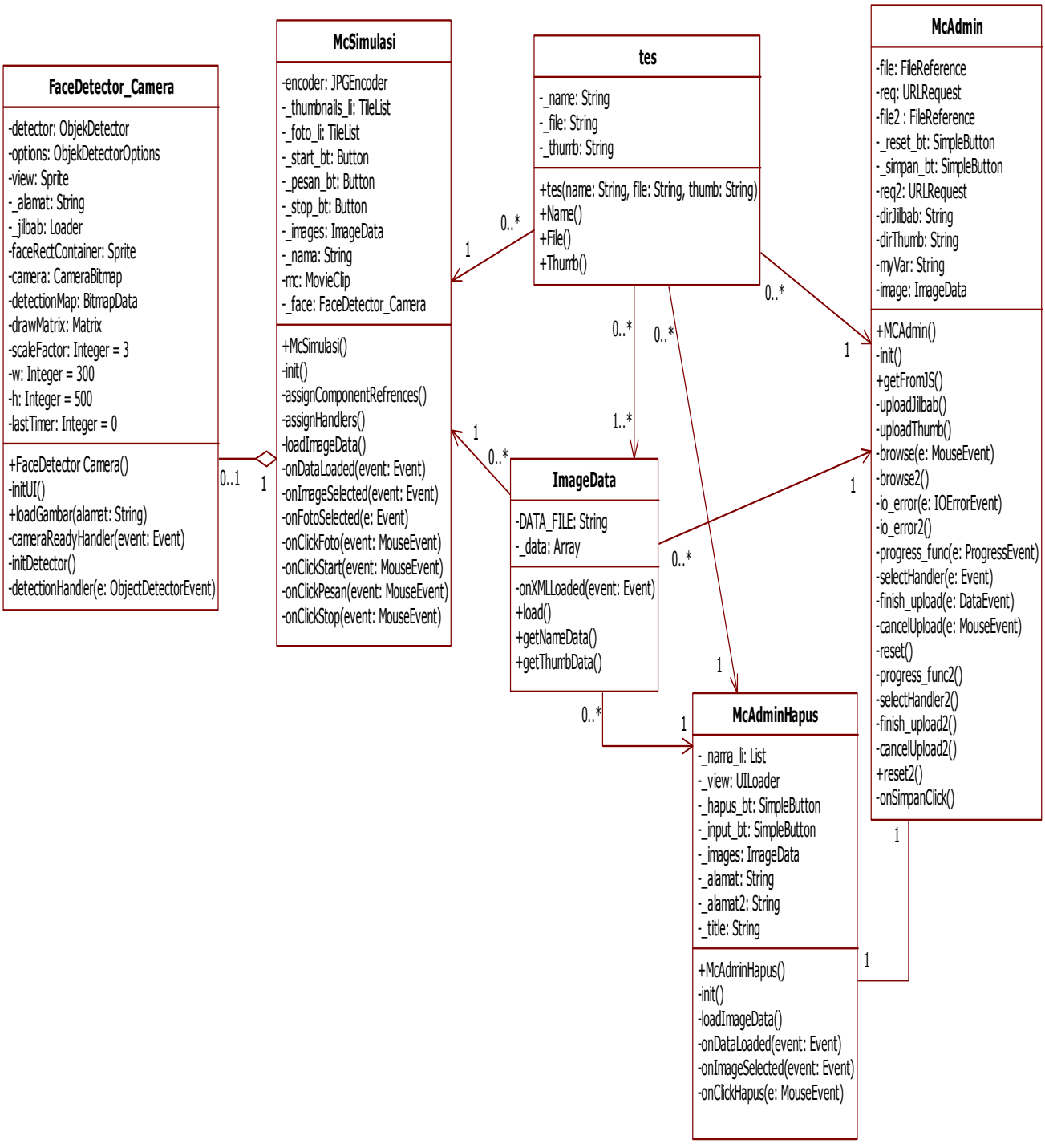
Proses menghapus model jilbab memiliki alur proses seperti digambarkan sebagai berikut :



Gambar 3. 26 *Activity Diagram* Menghapus Model Jilbab

3.1.7.4 *Class Diagram*

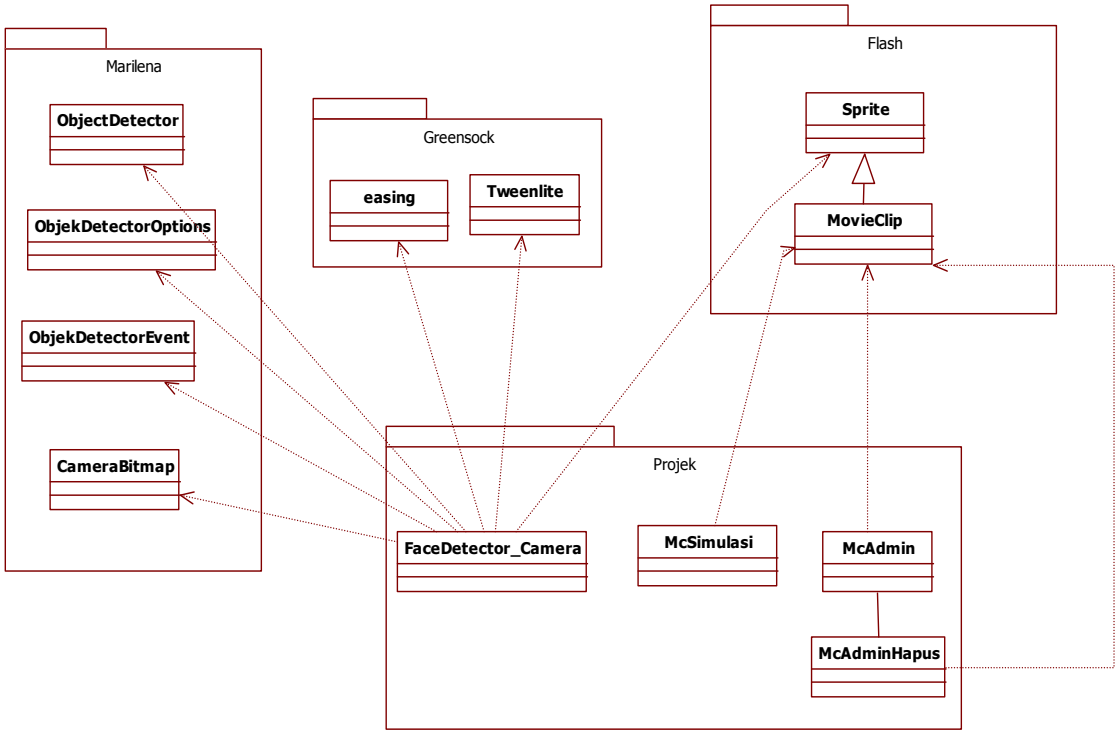
Class Diagram menggambarkan apa saja kelas yang ada dan keterkaitan antar kelas yang akan ada di perangkat lunak jilbab virtual ini. Hubungan tersebut tergambar pada gambar 3.22 sebagai berikut :



Gambar 3. 27 Class Diagram

3.1.7.5 *Package Diagram*

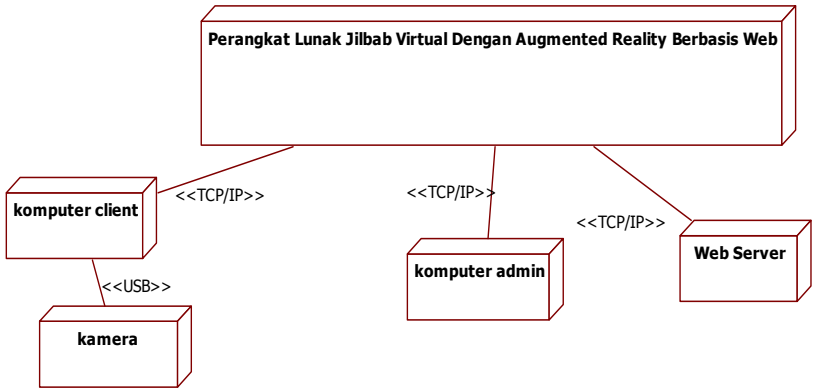
Package Diagram adalah diagram yang menggambarkan cara mengelola *class-class* yang dibuat dan keterkaitannya. *Package* juga mengelompokkan *class-class* sesuai dengan kategori tertentu, berikut ini gambar diagram *package* :



Gambar 3. 28 *Package Diagram*

3.1.7.6 *Deployment Diagram*

Deployment diagram menggambarkan bagaimana *software* dan *hardware* bekerja sama. Dalam gambar 3.29 digambarkan *deployment diagram* dimana di dalamnya terlihat keterkaitan komponen server, admin, dan user. Berikut ini gambar *deployment diagram*.



Gambar 3. 29 *Deployment Diagram*

3.2 Perancangan Sistem

Perancangan merupakan bagian dari metodologi pembangunan suatu perangkat lunak yang harus dilakukan setelah melalui tahapan analisis. Pada bagian ini akan dijelaskan perancangan sistem yang dimaksudkan untuk menggambarkan perbedaan antara sistem yang sedang berjalan dengan sistem yang diusulkan. Perancangan sistem ini menggunakan pendekatan sistem terkomputerisasi. Langkah-langkah yang dilakukan dalam tahapan perancangan sistem ini adalah sebagai berikut :

1. Perancangan Arsitektural Perangkat Lunak
2. Perancangan Struktur Menu
3. Perancangan Antarmuka Perangkat Lunak
4. Perancangan Pesan
5. Perancangan Prosedural

3.2.1 Perancangan Arsitektur Perangkat Lunak

Perancangan arsitektural perangkat lunak merupakan gambaran bagaimana komponen fungsional perangkat lunak disusun, diorganisasi, dan distrukturkan. Sistem perangkat lunak jilbab virtual ini memanfaatkan sistem eksternal yang telah ada yaitu website penjualan di toko Sakinah. Perangkat lunak di toko tersebut telah memiliki server aplikasi maupun database yang baik sehingga mendukung perangkat lunak jilbab virtual untuk di *embedded* ke dalam sistem penjualan tersebut.

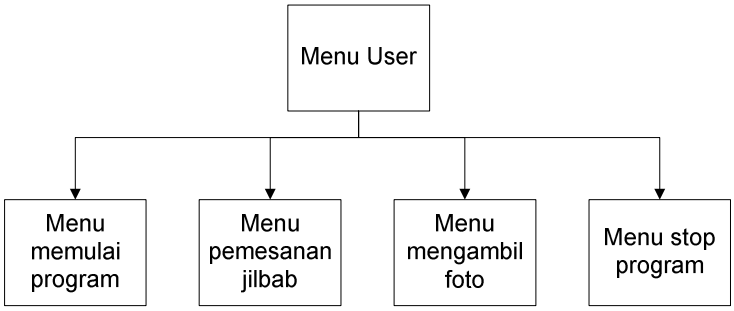
Perangkat lunak jilbab virtual ini terdiri dari dua bagian yaitu *user* dan admin. Untuk bagian *user* akan di *embedded* di sebuah halaman tersendiri yang bisa digunakan oleh setiap pengunjung web tersebut. Sedangkan untuk bagian admin akan di *embedded* ke sebuah halaman *session* admin sehingga hanya dapat diakses oleh admin yang telah melakukan *login* di website.

3.2.2 Perancangan Struktur Menu

Struktur menu dirancang sesuai dengan level pengguna sistem. Terdapat dua pengguna perangkat ini yaitu user dan admin. Adapun struktur menunya adalah sebagai berikut :

3.2.2.1 Struktur Menu User

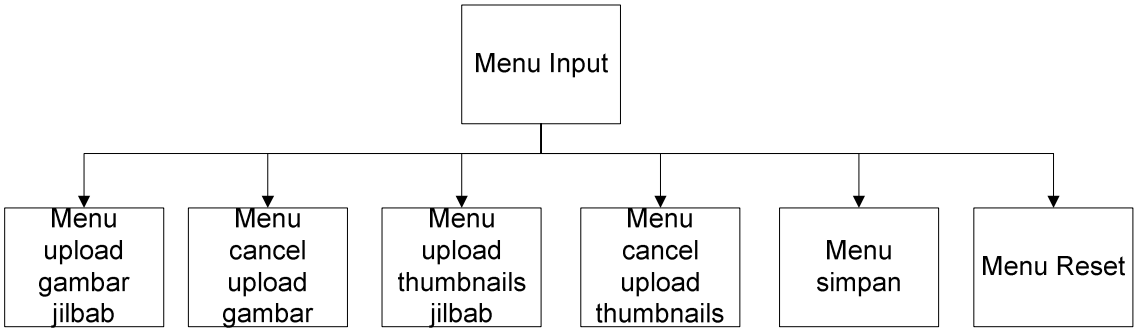
Struktur menu user hanya memiliki satu halaman yang didalamnya terdapat empat menu. Untuk lebih jelasnya struktur menu pada user akan ada pada gambar sebagai berikut :



Gambar 3. 30 Struktur menu user

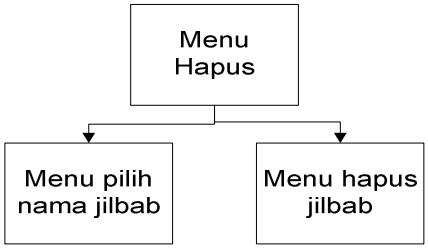
3.2.2.2 Struktur Menu Admin

Untuk lebih jelasnya struktur menu pada admin akan ada pada gambar sebagai berikut :



Gambar 3. 31 Struktur Menu Input

Sedangkan untuk struktur menu admin untuk menghapus model jilbab ialah sebagai berikut :



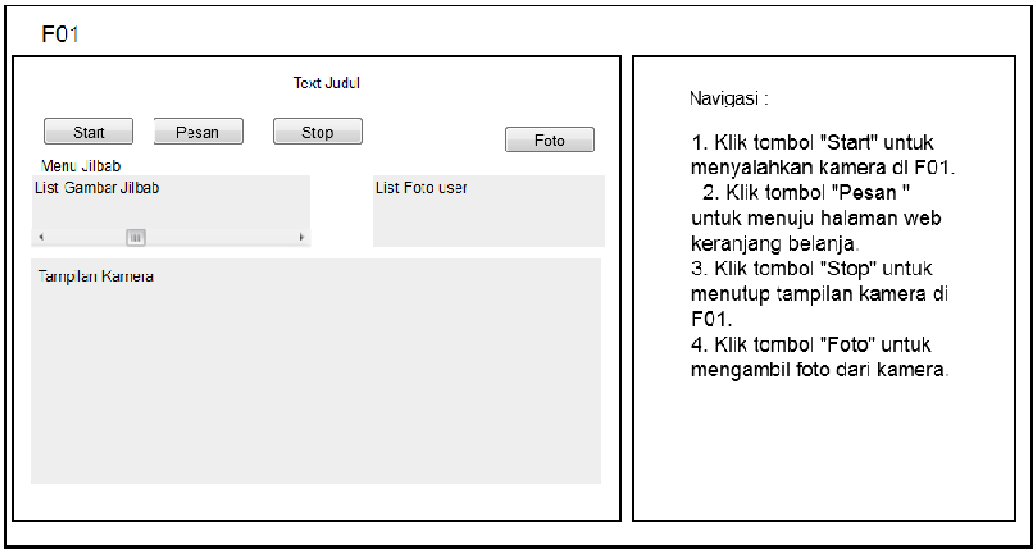
Gambar 3. 32 Struktur Menu Hapus

3.2.3 Perancangan Antarmuka Perangkat Lunak

Interface atau antarmuka merupakan tampilan dari suatu perangkat lunak yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara perangkat lunak dengan *user*. Sistem yang akan dibangun diharapkan menyediakan *interface* yang mudah dipahami dan digunakan oleh *user*.

3.2.3.1 Perancangan Antarmuka *User*

Tampilan antarmuka *user* adalah tampilan untuk user mencoba jilbab virtual. Untuk lebih jelasnya akan ada pada gambar dibawah ini :



Gambar 3. 33 Antarmuka *User*

3.2.3.2 Perancangan Antarmuka Admin

Tampilan antarmuka admin adalah tampilan untuk admin mengelola model jilbab. Antarmuka admin sendiri memiliki dua antarmuka yaitu untuk menambah data dan menghapus data.

3.2.3.2.1 Perancangan Antarmuka Menambah Model Jilbab

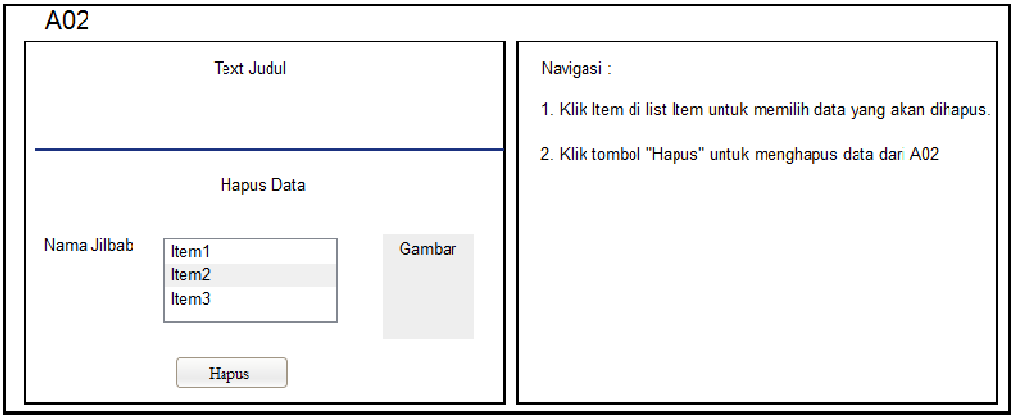
Tampilan antarmuka menambah model jilbab adalah tampilan admin untuk menambah model jilbab di perangkat lunak jilbab virtual. Untuk lebih jelasnya akan ada pada gambar dibawah ini :



Gambar 3. 34 Antarmuka Menambah Model Jilbab

3.2.3.2.2 Perancangan Antarmuka Menghapus Model Jilbab

Tampilan antarmuka menghapus model jilbab adalah tampilan admin untuk menghapus model jilbab dari perangkat lunak jilbab virtual. Untuk lebih jelasnya akan ada pada gambar dibawah ini :



Gambar 3. 35 Antarmuka Menghapus Model Jilbab

3.2.4 Perancangan Pesan

Pada tabel di bawah ini merupakan perancangan pesan yang terdapat diperangkat lunak jilbab virtual ini. Adapun tampilan pesan yang ada adalah sebagai berikut :

Tabel 3. 12 Perancangan Pesan

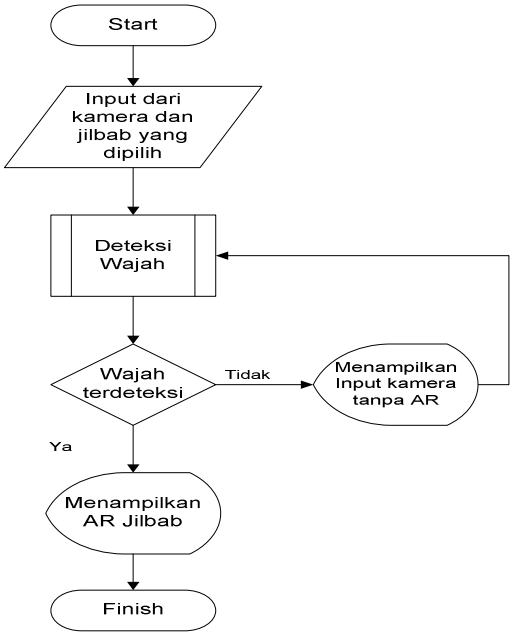
No	Text Pesan	Keterangan
P01	File tidak dapat diupload !	Pesan kesalahan di A01
P02	Upload sukses	Pesan berhasil di A01
P03	Proses simpan berhasil	Pesan berhasil di A01
P04	Proses simpan gagal !	Pesan kesalahan di A01
P05	Proses hapus berhasil	Pesan berhasil di A02
P06	Proses hapus gagal !	Pesan kesalahan di A02

3.2.5 Perancangan Prosedural

Sebagai langkah terakhir dalam perancangan yaitu perancangan prosedural yang akan di implementasikan ke dalam sistem. Prosedur ini akan digunakan sebagai algoritma dasar dalam mengkodekan prosedur yang ada. Adapun perancangan prosedural untuk pembangunan sistem perangkat lunak jilbab virtual ini adalah sebagai berikut :

3.2.5.1 Prosedural Penggunaan Jilbab Virtual

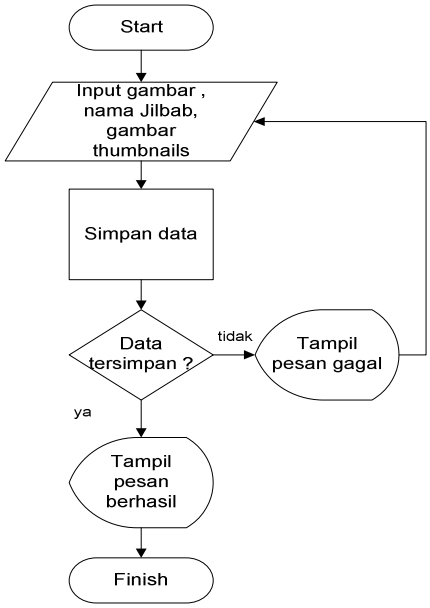
Untuk lebih jelasnya tentang prosedur penggunaan jilbab virtual dapat dilihat pada gambar sebagai berikut :



Gambar 3. 36 Prosedural Penggunaan Jilbab Virtual

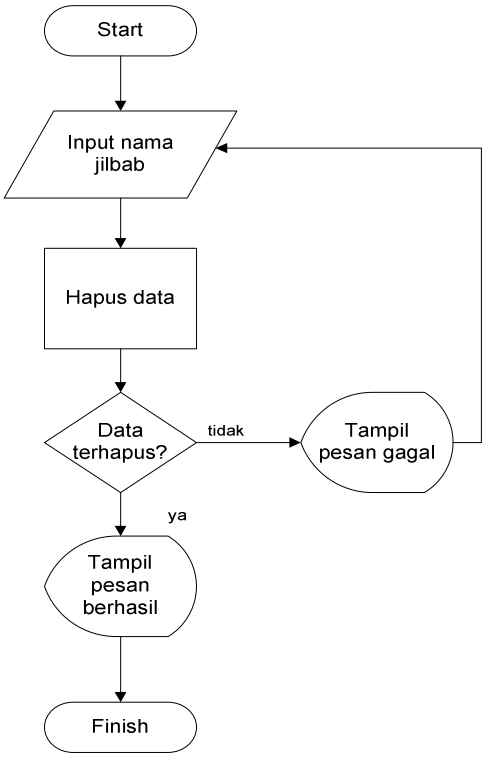
3.2.5.2 Prosedural Menambah Model Jilbab

Untuk lebih jelasnya tentang prosedur menambah model jilbab dapat dilihat pada gambar sebagai berikut :



Gambar 3. 37 Prosedural Menambah Model Jilbab

3.2.5.3 Prosedural Menghapus Model Jilbab



Gambar 3. 38 Prosedural Menghapus Model Jilbab