BAB 3
ANALISIS DAN PERANCANGAN SISTEM

Analisis sistem merupakan proses penguraian konsep yang utuh kedalam bagian-bagian yang lebih sederhana. Analisis pada sistem yang dibutuhkan untuk implementasi sistem untuk mendapatkan hasil yang baik meliputi analisis masalah, analisis permainan sejenis, analsis sistem yang akan dibangun, analisis arsitektur sistem, analisis metode yang digunakan, analisis permainan yang aka dibangun, analisis kebutuhan non-fungsional, analisis kebutuhan fungsional, Sedangkan untuk tahapan perancangan sistem membahas tentang perancangan sistem dan perancangan antarmuka yang akan dibangun.

3.1. Analisis Sistem

Analisis Sistem adalah suatu cara penguraian sistem yang akan dibangun. Tahapan punguraian yang dilakukan dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan tujuan untuk megidentifikasikan dan mengevaluasikan permasalahan pada sistem yang lama sehingga nantinya dapat dilakukannya perbaikan. Analisis dapat juga diartikan sebagai penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru. Dalam membangun sebuah sistem sangat perlu dilakukan tahapan tentang analisis sistem yang akan dibangun.

3.1.1. Analisis Masalah

Permainan yang menggunakan alat kendali berbentuk fisik memiliki kekurangan yaitu membuat pemain cepat merasa bosan karena pemain tidak dapat berinteraksi langsung dengan permainan, serta pemain kurang aktif dalam bergerak. Salah satu permainan yang akan dibangun yang berjenis *arcade* adalah permainan *Augmented Wall*. Cara bermain *Augmented Wall* adalah pemain harus berusaha mendapatkan skor tertinggi dengan mengandalkan konsentrasi yang tinggi serta bergerak dengan cepat menyentuh objek-objek yang ada pada permainan. Untuk memainkan permainan jenis ini pada komputer biasanya menggunakan alat kendali berbentuk fisik yang berupa *mouse*, *keyboard*, *joystick*, dan *gamepad*. Sedangkan permainan berjenis *arcade* seperti *Augmented Wall* lebih menyenangkan apabila

permain berinteraksi langsung dengan permainan tanpa menggunakan alat kendali berbentuk fisik serta tidak hanya terdiam saat memainkannya. Dengan kekurangan yang telah disebutkan dapat disimpulkan untuk permainan *Augmented Wall* bahwa interaksi menggunakan alat kendali berbentuk fisik membuat pemain merasa bosan dan kurang interaktif.

Permainan *Augmented Wall* dibangun tidak menggunakan alat kendali berbentuk fisik. Dengan pengolahan citra memungkinkan pemain untuk berinteraksi langsung dengan permainan, pemain lebih aktif bergerak, serta memberikan pengalaman yang menarik karena melibatkan gerakan tubuh pemain saat memainkannya contohnya dengan menggerakan tangan. Terlebih saat memainkan permainan sejenis *Augmented Wall*.

3.1.2. Analisis Permainan Sejenis

Analisis permainan sejenis merupakan tahap analisis terhadap permainan yang telah dibuat. Tujuan dari analisis ini yaitu untuk membandingkan konsep dari permainan yang sudah ada untuk kemudian dikembangkan sesuai dengan kebutuhan dari permainan yang akan dibangun untuk kemudian dikembangkan sesuai dengan kebutuhan dari permainan yang akan dibangun. Ada dua permainan sejenis yang akan dianalisis diantaranya permainan *Kung Fu Panda Dojo Mojo* dan *Speed of Light*.

3.1.2.1. *Kung Fu Panda Dojo Mojo*

Permainan *Kung Fu Panda Dojo Mojo* merupakan permainan *arcade* yang menggunakan sebuah alat untuk mengendalikan permainan. *Kung Fu Panda Dojo Mojo* dikenalkan di Indonesia oleh *Timezone*. Permainan ini dikembangkan untuk *console* khusus yang hanya bisa digunakan untuk permainan *Kung Fu Panda Dojo Mojo*. Selain itu dalam permainan ini pemain diajak untuk memukul dan menangkap target yang cukup cepat untuk bisa melanjutkan ke tahap berikutnya. Logo dari permainan *Kung Fu Panda Dojo Mojo* dapat dilihat pada Gambar 3.1 Logo *Kung Fu Panda Dojo Mojo* berikut.

Gambar 3.1 Logo *Kung Fu Panda Dojo Mojo*

3.1.2.1.1. Cara Bermain *Kung Fu Panda Dojo Mojo*

Cara bermain permainan ini adalah dengan cara menekan tombol sesuai dengan posisi objek yang muncul pada layar. Permainan *Kung Fu Panda Dojo Mojo* memiliki beberapa antarmuka, yaitu antarmuka utama, antarmuka *gameplay*, dan antarmuka *result score.*

1. Antarmuka Utama

   Pada antarmuka utama terdapat perintah menekan tombol untuk memulai permainan. Antarmuka utama dapat dilihat pada Gambar 3.2 Antarmuka Utama berikut.

Gambar 3.2 Antarmuka Utama

2. Antarmuka *Gameplay*

Pada antarmuka *gameplay* terdapat beberapa tampilan, yaitu target dan waktu yang tersisa. Antarmuka *gameplay* dapat dilihat pada Gambar 3.3 Antarmuka *Gameplay* berikut.

Gambar 3.3 Antarmuka *Gameplay*

3. Antarmuka *Result Score*

Pada antarmuka *result score* terdapat tampilan skor akhir yang telah didapatkan oleh pemain. Antarmuka *result score* dapat dilihat pada gambar berikut.

Gambar 3.4 Antarmuka *Result Score*

3.1.2.1.2. Komponen *Kung Fu Panda Dojo Mojo*

Permainan *Kung Fu Panda Dojo Mojo* memiliki beberapa komponen diantaranya grafis, tombol, dan fungsi permainan yang dapat dilihat pada Tabel 3.1 Komponen berikut.

Tabel 3.1 Komponen *Kung Fu Panda Dojo Mojo*

No	Komponen	Keterangan
1	Grafis	Permainan ini menggunakan grafis 2 dimensi.
2	Tombol	Permainan ini memiliki 6 tombol yang digunakan untuk mengendalikan permainan. Berikut tombol beserta fungsinya yang diurutkan dari kiri ke kanan: 1. Mengklik objek pada bagian kiri atas. 2. Mengklik objek pada bagian kiri bawah. 3. Mengklik objek pada bagian bawah kiri. 4. Mengklik objek pada bagian bawah kanan. 5. Mengklik objek pada bagian kanan bawah. 6. Mengklik objek pada bagian kanan atas.
3	Fungsi Permainan	Permainan ini berguna untuk menghibur dan mengajak pemain untuk bergerak serta dapat melatih daya konsentrasi pemain.

3.1.2.1.3. Hasil Analisis *Kung Fu Panda Dojo Mojo*

Hasil analisis permainan *Kung Fu Panda Dojo Mojo* adalah sebagai berikut:

1. Permainan ini merupakan permainan berjenis *arcade* yang hanya bisa dimainkan oleh satu orang pemain.
2. Permainan ini sangat mudah untuk dapat dimainkan oleh semua umur, mulai dari anak kecil sampai orang tua.

3.1.2.2. *Speed of Light*

Permainan *Speed of Light* merupakan permainan *arcade* yang tombol-tombol yang bisa ditekan untuk bermain. *Speed of Light* merupakan salah satu permainan yang ada di *Timezone*. Permainan ini dikembangkan untuk *console* khusus yang hanya bisa digunakan untuk permainan *Speed of Light*. Selain itu dalam permainan

ini pemain mengajak untuk bergerak cepat dan tidak panik untuk mendapatkan skor tertinggi dalam waktu yang telah ditentukan. Logo dari permainan *Speed of Light* dapat dilihat pada Gambar 3.5 Logo *Speed of Light* berikut.

Gambar 3.5 Logo *Speed of Light*

3.1.2.2.1. Cara Bermain *Speed of Light*

Cara bermain permainan *Speed of Light* adalah dengan menekan tombol yang menyala secepat mungkin. Sebenarnya untuk bermain cukup mudah, namun memerlukan konsentrasi dan kecepatan tangan. Pada permainan ini untuk mendapatkan skor cukup dengan menekan tombol yang menyala. Permainan ini hanya memiliki antar muka untuk menampilkan skor dan waktu yang tersisa.

3.1.2.2.2. Komponen *Speed of Light*

Permainan *Speed of Light* memiliki beberapa komponen diantaranya grafis, tombol, dan fungsi permainan yang dapat dilihat pada Tabel 3.2 Komponen *Speed of Light* berikut.

Tabel 3.2 Komponen *Speed of Light*

No	Komponen	Keterangan
1	Tombol	Permainan ini memiliki tombol-tombol yang berguna untuk mendapatkan skor dalam permainan.

2	Fungsi Permainan	Permainan ini berguna untuk menghibur dan mengajak pemain untuk bergerak serta dapat melatih daya konsentrasi pemain.

3.1.2.2.3. Hasil Analisis *Speed of Light*

Hasil analisis permainan *Speed of Light* adalah sebagai berikut:

1. Permainan ini merupakan permainan berjenis *arcade* yang hanya bisa dimainkan oleh satu atau dua orang pemain.
2. Permainan ini memiliki cara bermain yang mengharuskan permain menggerakan tangan dan bahkan bisa sampai menggerakan tubuh untuk menekan tombol.
3. Dengan cara bermain yang mengharuskan bergerak bisa memanfaatkan permainan ini untuk sarana berolahraga.

3.1.2.3. Perbandingan Komponen pada Permainan Sejenis

Perbandingan komponen pada permainan sejenis adalah cara membandingkan komponen yang dimiliki oleh masing-masing permainan yang telah dianalisis dengan tujuan dapat membantu dalam pembangunan permainan. Perbandingan komponen permainan sejenis dapat dilihat pada Tabel 3.3 Perbandingan Komponen Permainan Sejenis berikut.

Tabel 3.3 Perbandingan Komponen Permainan Sejenis

No	Komponen	Permainan *Kung Fu Panda Dojo Mojo*	Permainan *Speed of Light*
1	Antarmuka	Memiliki 3 antarmuka.	-
2	Grafis	Permainan ini menggunakan grafis 2 dimensi.	-
3	Tombol	Permainan ini memiliki 6 tombol yang digunakan untuk mengendalikan permainan. Berikut tombol beserta fungsinya yang diurutkan dari kiri ke kanan:	Permainan ini memiliki tombol-tombol yang berguna untuk mendapatkan skor dalam permainan.

		1. Mengklik objek pada bagian kiri atas. 2. Mengklik objek pada bagian kiri bawah. 3. Mengklik objek pada bagian bawah kiri. 4. Mengklik objek pada bagian bawah kanan. 5. Mengklik objek pada bagian kanan bawah. 6. Mengklik objek pada bagian kanan atas.	
4	Fungsi Permainan	Permainan ini berguna untuk menghibur dan mengajak pemain untuk bergerak serta dapat melatih daya konsentrasi pemain.	Permainan ini berguna untuk menghibur dan mengajak pemain untuk bergerak serta dapat melatih daya konsentrasi pemain.

3.1.2.4. Kesimpulan Hasil Analisis Permainan Sejenis

Dari hasil analisis permainan sejenis terdapat beberapa perbedaan pada komponen permainan yang dapat dikembahkan dan digunakan dalam pembangunan permainan. Berikut adalah kesimpulan hasil analisis permainan sejenis:

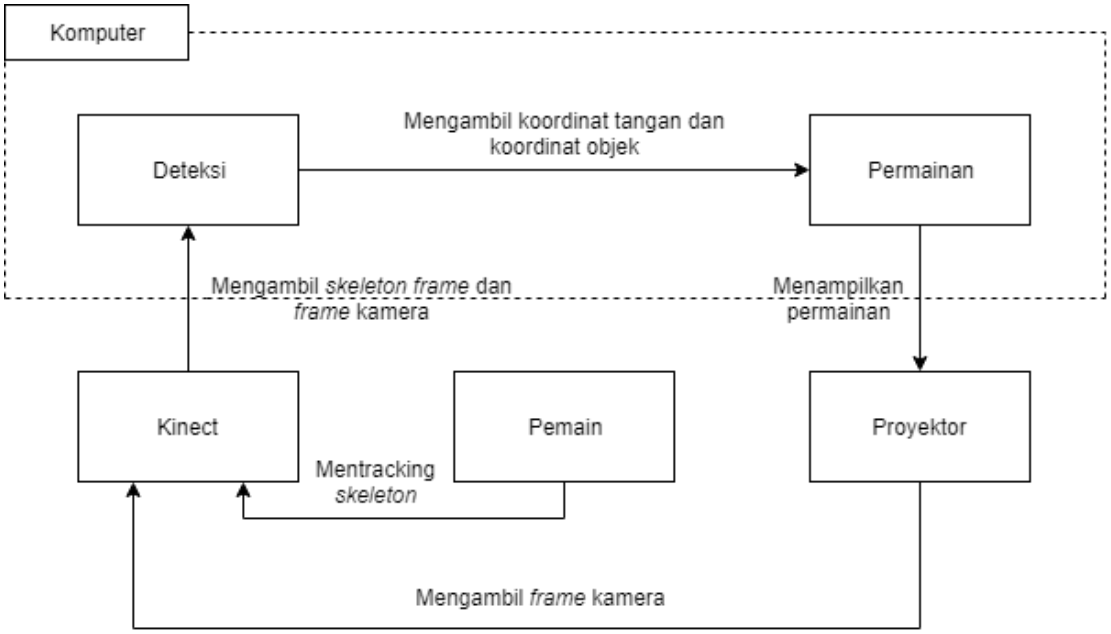
1. Permainan *Kung Fu Panda Dojo Mojo* merupakan permainan berjenis *arcade* yang hanya dapat dimainkan oleh satu pemain. Permainan ini memiliki 6 tombol yang digunakan untuk mengendalikan permainan dan posisi yang dapat dikendalikan dari permainan itu terbatas hanya ada 6 posisi. Konsep permainan ini memberikan ide untuk menambah posisi yang dapat dikendalikan dalam permainan.
2. Permainan *Speed of Light* merupakan permainan berjenis *arcade* juga, akan tetapi permainan ini bisa dimainkan oleh satu atau dua orang pemain secara bersamaan. Dari segi *gameplay* permainan ini cukup monoton karena tidak

memiliki tampilan grafis. Namun, secara konsep permainan ini cukup menarik karena mengandalkan konsentrasi yang penuh dari pemain dan kecepatan tangan.

Dari kedua konsep permainan *Kung Fu Panda Dojo Mojo* dan *Speed of Light* dibangun dengan tujuan untuk menghibur, melatih konsentrasi pemain, dan membuat pemain bergerak saat bermain. Kedua permainan ini mempunyai kelebihan masing-masing. Permainan *Kung Fu Panda Dojo Mojo* dari *gamplay*-nya mempunyai beberapa *stage* yang harus di selesaikan dan pada setiap *stage*-nya kecepatan serta targetnya bertambah, dan permainan *gameplay Speed of Light* mempunyai keunikan dari banyaknya tombol yang harus diawasi sehingga pemain memerlukan konsentrasi yang tinggi. Dengan konsep yang terdapat pada kedua permainan tersebut dapat dikembangkan permainan yang menghibur dan melatih konsentrasi pemain. Untuk memberikan pengalaman yang berbeda dan pemain merasa lebih masuk kedalam permainan maka akan memanfaatkan pengolahan citra untuk mengendalikan permainan yang akan dikembangkan.

3.1.3. Analisis Sitem yang akan Dibangun

Dari masalah yang ada, solusi dengan membangun sebuah sistem yang dapat mengendalikan permainan berjenis *arcade* yang menggunakan gerakan tangan manusia sebagai alat kendali permainan dengan memanfaatkan *sekeletal tracking* yang ada pada *Kinect* dan *color detection* yang diambil pada *camera RGB Kinect*, mendeteksi pergerakan tangan pemain dan mendeteksi objek-objek yang ada pada permainan. Untuk pembangunan sistem membutuhkan dua *library*, yaitu *Kinect SDK* yang digunakan untuk mendeteksi pergerakan tangan pemain memanfaatkan metode *skeletal tracking* dan *OpenCV* dapat digunakan untuk mendeteksi pergerakan objek pada permainan dengan metode *color detection*. Analisis sistem yang dibangun dapat dilihat pada Gambar 3.6 Analisis Sistem yang akan Dibangun.



Gambar 3.6 Analisis Sistem yang akan Dibangun

Dari Gambar 3.6 Analisis Sistem yang akan Dibangun dapat diamati bahwa setiap bagian penyusun sistem saling terhubung dan memiliki fungsinya masing masing. Bagian komputer berfungsi untuk menjalankan permainan dan mendeteksi gerakan tangan dan objek-objek pada permainan yang ditampilkan oleh proyektor. Berikut adalah uraian dari masing-masing komponen blok sistem pada Gambar 3.6 Analisis Sistem yang akan Dibangun.

1. Blok Komputer

   Blok komputer mempunyai dua kompenen yaitu deteksi dan permainan. Deteksi berfungsi untuk mendeteksi pergerakan tangan dan objek-objek yang ada pada permainan, sedangkan permainan merupakan permainan yang akan ditampilkan pada proyektor dan memproses hasil dari deteksi sebagai masukkan permainan.

   a. Deteksi

      Deteksi digunakan untuk memproses data *skeleton frame* dan *color frame* yang dikirimkan oleh Kinect. Kemudian deteksi akan meneruskan ke permainan yang nantinya akan di proses oleh permainan.

   b. Permainan

Permainan merupakan permainan yang dibangun dan digunakan untuk memproses masukkan dari deteksi. Permainan akan menampilkannya pada proyektor.

2. *Kinect*

   Kinect digunakan untuk mengambil data *skeleton frame* dan gambar yang ditampilkan oleh proyektor. *Kinect* mengambil *skeletorn frame* dan gambar terus menerus selama deteksi berjalan dan meneruskan data ke deteksi.

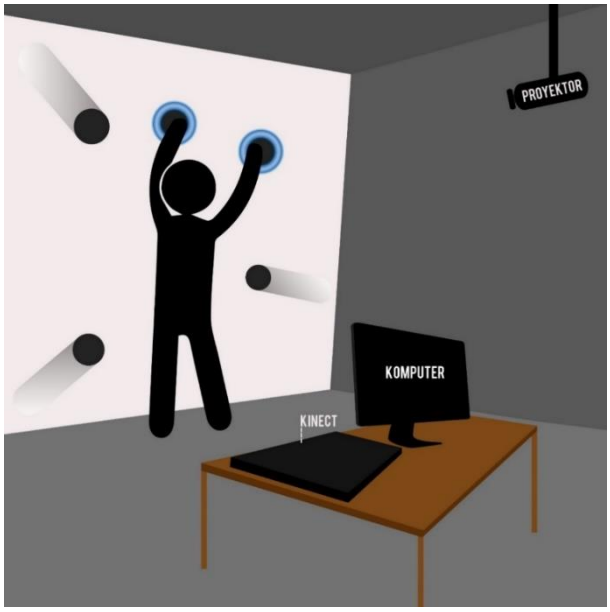
3. Pemain

   Pemain merupakan orang yang memainkan permainan. Pemain pada saat bermain nantinya akan menghadap ke dinding yang telah ada tampilan permainan yang ditampilkan oleh proyektor.

4. Proyektor

   Proyektor digunakan untuk menampilkan permainan. Proyektor menampilkan permainan ke dinding yang berukuran 3,2x2,4 meter.

Adapun simulasi posisi pemain dan alat pada saat menggunakan sistem. Dapat dilihat pada Gambar 3.7 Simulasi Saat Bermain berikut:



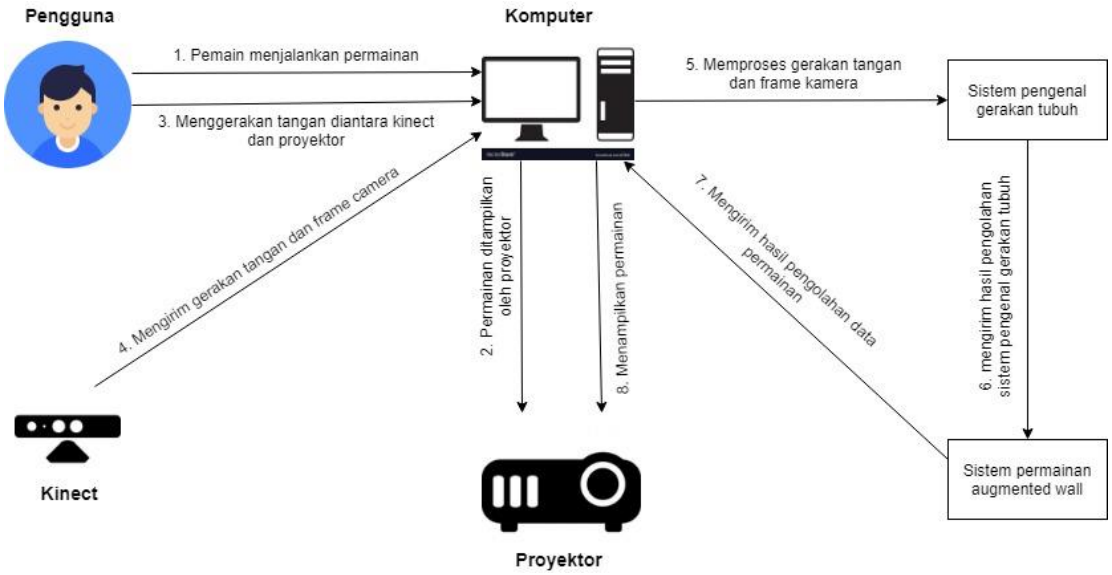
Gambar 3.7 Simulasi Saat Bermain

Berikut ini merupakan penjelasan mengenai simulasi saat menggunakan sistem yang akan dibangun :

1. Proyektor menampilkan permainan pada dinding yang berukuran 3,2x2,4 meter.
2. Pemain harus diantara gambar yang ditampilkan oleh proyektor.
3. Kinect harus bisa menangkap seluruh gambar yang ditampilkan oleh proyektor.

3.1.4. Analisis Arsitektur Sistem

Arsitektur sistem yang akan dibangun terdiri dari tiga bagian, yaitu mengenai analisis sistem alat kendali permainan dan sistem permainan *Augmented Wall*. Arsitektur sistem yang aka dibangun dapat dilihat pada Gambar 3.8 Arsitektur Sistem berikut.



Gambar 3.8 Arsitektur Sistem

Berikut ini merupakan uraian dari Arsitektur sistem:

1. Pemain menjalankan permainan pada komputer.
2. Proyektor menampilkan permainan pada dinding berukuran 3,2x2,4 meter.
3. Pemain berdiri antara *Kinect* dan proyektor.
4. Pemain menghadap ke arah dinding yang ada tampilan permainan dan menggerakan tangan dan.
5. *Kinect* mendeteksi pergerakan tangan pemain dan objek-objek yang ada pada permainan.

6. Pergerakan pemain menjadi masukkan untuk *Kinect* dan menghasilkan keluaran *skeleton frame*.
7. Tampilan permainan yang ditampilkan oleh menjadi masukkan untuk *Kinect* dan menghasilkan *color frame*.
8. Komputer menerima masukkan dari *Kinect* dan memproses *skeleton frame* dengan metode *skeleton tracking* dan *color frame* dengan metode *color detection* sebegai masukkan untuk permainan.
9. *Skeleton tracking* menghasil koordinat tangan pemain.
10. *Color tracking* menghasil koordinat objek pada permain.
11. Pada antarmuka utama dan antarmuka skor akhir koordinat tangan berfungsi sebagai kursor, sedangkan pada antarmuka *gameplay* koordinat tangan akan di cek apakah koordinat tersebut berada masuk didalam koordinat objek atau tidak. Apabila masuk, koordinat tangan akan digunakan sebagai fungsi klik.
12. Setelah proses diatas berhasil dilakukan maka sistem sudah bisa digunakan untuk mengendalikan permainan.

3.1.5. Analisis Metode

Analisis ini bertujuan untuk menggambarkan kebutuhan sistem untuk dapat mendeteksi gerakan pemain dan objek-objek yang ada pada permainan. Dalam penelitian ini metode yang digunakan adalah metode *skeleton tracking* dan metode *color detection*.

3.1.5.1. Analisis *Skeleton Tracking*

Analisis *skeleton tracking* menggambarkan kebutuhan sistem untuk dapat menyediakan fungsi yang dapat mendeteksi pergerakan tangan pemain dalam permainan *Augmented Wall*. Proses s*keleton tracking* ini memanfaatkan *Kinect* dan *Kinect SDK*. Prinsip kerja *skeleton tracking* adalah mendeteksi pergerakan tubuh manusia yang tertangkap. Pergerakan tubuh yang tertangkap kemudian dipetakan dalam *skeleton frame.* Dalam *skeleton frame* berisi posisi-posisi sendi utama pada manusia. Namun dalam permainan *Augmented Wall* hanya akan menggunakan dua buah posisi, yaitu posisi tangan kanan dan tangan kiri. Posisi kedua tangan itu akan digunakan untuk keperluan antarmuka utama dan *gameplay* permainan. Gambaran

skeleton tracking dapat dilihat pada Gambar 3.9 *Flowchart* Metode *Skeleton Tracking* berikut.

Gambar 3.9 *Flowchart* Metode *Skeleton Tracking*

3.1.5.1.1. Mendeteksi Posisi Sendi Utama Tubuh Manusia

Pada proses ini mendapatkan masukkan berupa *skeleton frame*. *Skeleton frame* tersebut kemudian akan dipetakan oleh *Kinect SDK*. Apabila *skeleton frame* sudah dipetakan maka siap digunakan untuk proses selanjutnya.

Mendeteksi posisi sendi utama tubuh manusia memanfaatkan fungsi KinectSensor.SkeletonFrameReady yang ada pada *Kinect SDK*. Fungsi SkeletonFrameReady menghasilkan posisi sendi utama tubuh manusia. Untuk proses ini dapat menggunakan perintah berikut.

```
KinectSensor.SkeletonFrameReady += SensorSkeletonFrameReady;
```

Keterangan dari perintah diatas Convert dapat dilihat pada tabel berikut.

Tabel 3.4 Keterangan Perintah Konversi Warna

Komponen	Deskripsi
SensorSkeletonFrameReady	Merupakan fungsi untuk mengolah data apabila ada *skeleton frame* yang terdeteksi

3.1.5.1.2. Mendapatkan Posisi Tangan Kanan dan Tangan Kiri

Pada proses ini mendapatkan masukkan berupa *skeleton frame*. Masukkan tersebut kemudia diolah dalam fungsi SensorSkeletonFrameReady yang berisikan perintah untuk mendapatkan posisi tangan. Untuk mendapatkan posisi tangan kanan dan tangan kiri dalam fungsi SensorSkeletonFrameReady dapat menggunakan perintah berikut.

```
// Tangan kanan
Skeleton firstSkel = (from trackskeleton in SumSkeletons where
trackskeleton.TrackingState == SkeletonTrackingState.Tracked
select trackskeleton).FirstOrDefault();

// Tangan Kiri
Skeleton secondSkel = (from trackskeleton in SumSkeletons
where trackskeleton.TrackingState == SkeletonTrackingState.Tracked select
trackskeleton).FirstOrDefault();

SkeletonPoint TanganKanan = firstSkel.Joints[JointType.HandRight].Position

SkeletonPoint TanganKiri = firstSkel.Joints[JointType.HandLeft].Position
```

3.1.5.1.3. Konversi Posisi Tangan ke *Pixels*

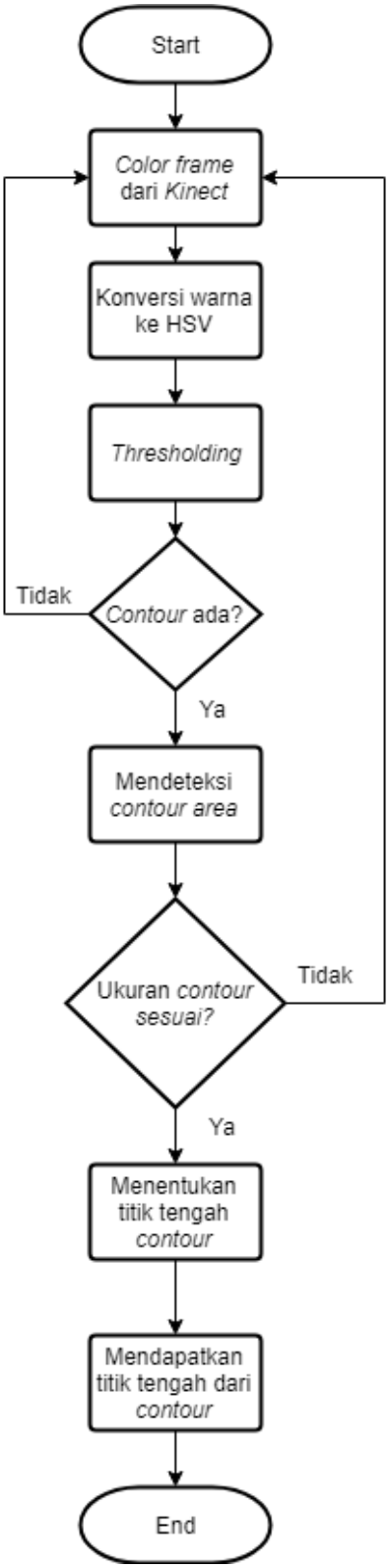
Proses ini sangat diperlukan supaya posisi tangan bisa dirubah menjadi koordinat yang bisa digunakan untuk permainan. Pada proses sebelumnya posisi tangan telah didapatkan, namun posisinya itu masih bertipe data SkeletonPoint. Untuk dapat menggunakan posisi tangan sebagai masukkan dalam permainan, maka perlu dirubah dulu menjadi koordinat dalam bentuk *pixels* menggunakan fungsi PosisiSkala yang telah dibuat. Hasil dari fungsi ini ada koordinat tangan bertipe data Point yang sudah bisa digunakan untuk sebagai masukkan untuk permainan. Untuk mendapatkan koordinat tangan dalam bentuk koordinat *pixels* bisa menggunakan fungsi berikut.

```
Point _TanganKiri =
PosisiSkala(firstSkel.Joints[JointType.HandLeft].Position)
Point _TanganKiri =
PosisiSkala(firstSkel.Joints[JointType.HandLeft].Position)

private static Point PosisiSkala(SkeletonPoint skeletonPoint) {
  var depthPoint =
KinectSensor.CoordinateMapper.MapSkeletonPointToDepthPoint(skeletonPoint,
                    DepthImageFormat.Resolution640x480Fps30);
  return new Point(depthPoint.X, depthPoint.Y);
}
```

3.1.5.2. Analisis Metode *Color Detection*

Analisis *color detection* menggambarkan kebutuhan sistem untuk dapat menyediakan sebuah fungsi pendeteksi objek dan mendeteksi pergerakan objek pada permainan *Augmented Wall*. Proses deteksi objek-objek pada permainan menggunakan metode *color detection* dengan memanfaatkan beberapa fungsi dari *OpenCV*. Masukkan untuk *color detection* ini adalah *color frame* yang didapatkan dari *Kinect* yang akan menghasilkan koordinat dari setiap objek pada permainan. Untuk melakukan deteksi objek pada permainan ada beberapa tahapan, yaitu konversi warna *color frame* ke HSV, *thresholding*, mendeteksi *contour area*, dan menentukan titik tengah *contour*. Gambaran dari proses *color detection* yang akan dibangun dapat dilihat pada Gambar 3.10 *Flowchart* Proses *Color Detection* berikut.



Gambar 3.10 *Flowchart* Proses *Color Detection*

3.1.5.2.1. Konversi Warna

Pada proses ini mendapatkan masukkan berupa *color frame* dengan tiga komponen warna *Red, Green, Blue* (RGB). *Color frame* tersebut kemudian akan di konversi menjadi citra dengan komponen warna *Hue, Saturation, Value* (HSV). Pemilihan warna HSV ini dikarenakan citra dengan warna HSV tidak terlalu terpengaruh oleh cahaya.

Konversi warna ke HSV memanfaatkan fungsi Convert yang ada pada *OpenCV*. Fungsi Convert menghasilkan citra dengan warna HSV. Untuk proses konversi warna dapat menggunakan perintah berikut.

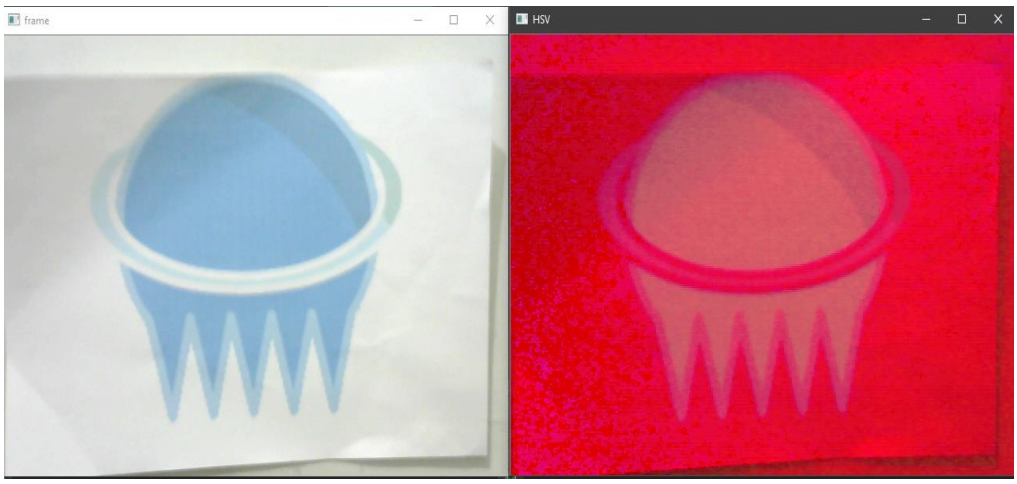
```
Image<HSV, Byte> HsvFrame = ColorFrame.Convert<Hsv, Byte>();
```

Keterangan dari perintah diatas Convert dapat dilihat pada tabel berikut.

Tabel 3.5 Keterangan Perintah Konversi Warna

Komponen	Deskripsi
Image<HSV, Byte>	Merupakan tipe data hasil dari fungsi Convert
ColorFrame	Merupakan citra masukkan yang diterima dari *Kinect*.
<Hsv, Byte>	Merupakan warna yang dinginkan untuk konversi

Berikut dari konversi warna bisa dilihat pada Gambar 3.11 Konversi Warna berikut.



Gambar 3.11 Konversi Warna

3.1.5.2.2. *Thresholding*

Pada proses ini menerima masukkan citra dengan warna HSV. Dari citra tersebut kemudian warna dipisahkan menjadi dua berdasarkan batas warna yang telah ditentukan (pada penelitian ini menggunakan warna biru untuk *threshold*). Nilai intensitas citra yang sama dengan batas warna akan diubah menjadi 1 (putih) sedangkan nilai intensitas citra yang tidak sama dengan batas warna akan diubah menjadi 0 (hitam).

Thresholding ini memanfaatkan fungsi InRange. Fungsi InRange menghasilkan citra dengan dua komponen warna hitam dan putih. Untuk proses *thresholding* dapat menggunakan perintah berikut.

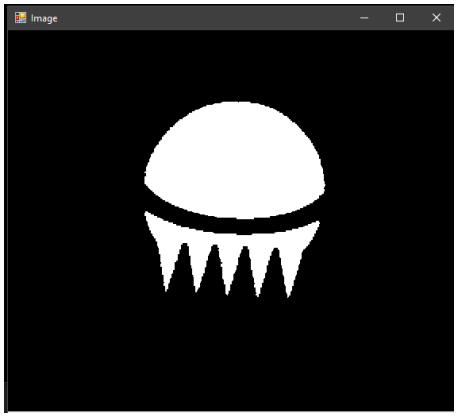
```
Image<Gray, Byte> Thresholding = HsvFrame.InRange(ColorLower, ColorUpper);
```

Keterangan dari perintah diatas InRange dapat dilihat pada tabel berikut.

Tabel 3.6 Keterangan Perintah *Thresholding*

Komponen	Deskripsi
Image<Gray, Byte>	Merupakan tipe data hasil dari fungsi InRange
HsvFrame	Merupakan citra masukkan yang diterima dari hasil konversi warna.
ColorLower	Merupakan batas bawah warna yang dinginkan
ColorUpper	Merupakan batas atas warna yang dinginkan

Berikut hasil dari *thresholding* dapat dilihat pada Gambar 3.12 *Thresholding* berikut.



Gambar 3.12 *Thresholding*

3.1.5.2.3. Mendeteksi *Contour Area*

Proses ini menerima masukkan berupa citra yang telah di *thresholding*. Citra tersebut kemudian dideteksi ukuran dari *contour* yang ada. Untuk mendeteksi *contour area* memanfaatkan fungsi FindContours yang ada pada *OpenCV*. Apabila ukuran *contour area* sesuai dengan yang telah ditentukan maka akan masuk ke proses selanjutnya, yaitu mencari titik tengah dari *contour area*. Untuk proses mendeteksi *contour area* dapat menggunakan perintah berikut.

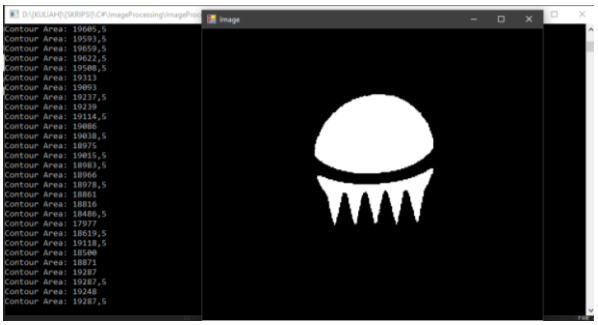
```
FindContours(Thresholding, Contours, null, RetrType.External,
ChainApproxMethod.ChainApproxNone);
```

Keterangan dari perintah diatas FindContours dapat dilihat pada tabel berikut.

Tabel 3.7 Keterangan Perintah *Thresholding*

Komponen	Deskripsi
Thresholding	Merupakan citra hasil dari *thresholding*
Contours	Merupakan hasil *contours* yang terdeteksi oleh fungsi FindContours
Null	Merupakan *hierarcy*, namun pada sistem ini tidak diperlukan jadi diisi dengan null
RetrType.External	Merupakan perintah untuk mendapatkan bagian luar dari *contour*
ChainApproxMethod.ChainApproxNone	Merupakan perintah untuk *translate* semua *point* dari kode yang berurutan menjadi sebuah *point*

Berikut hasil *contour area* yang dapat dilanjutkan ke proses berikutnya dapat dilihat pada Gambar 3.13 Mendeteksi *Contour Area* berikut.



Gambar 3.13 Mendeteksi *Contour Area*

3.1.5.2.4. Menentukan Titik Tengah *Contour Area*

Contour yang didapat pada proses sebelumnya digunakan untuk mendapatkan koordinat objek dengan memanfaatkan fungsi Moments yang ada pada *OpenCV*. Hasil dari proses ini adalah koordinat dari setiap *contour* objek yang terdeteksi. Untuk proses mendeteksi *contour area* dapat menggunakan perintah berikut.

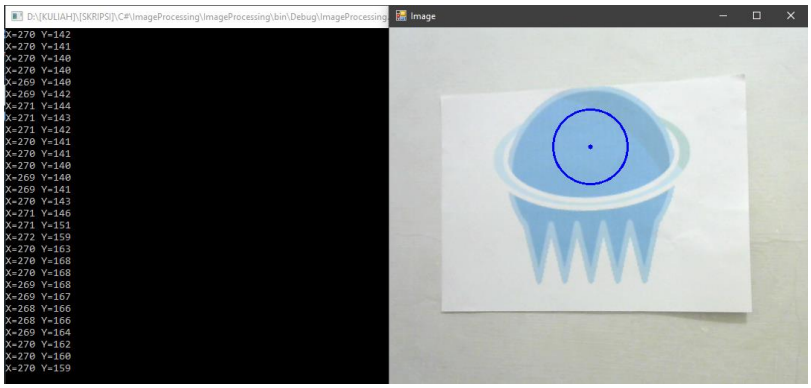
```
MCvMoments Moment = Moments(Contours);
int X = (int) (Moment.M10 / Moment.M00);
int Y = (int) (Moment.M01 / Moment.M00);
```

Keterangan dari perintah diatas Moments dapat dilihat pada tabel berikut.

Tabel 3.8 Keterangan Perintah *Thresholding*

Komponen	Deskripsi
McvMoments	Merupakan tipe data untuk menampung titik tengah pada Contours
Moment	Merupakan tipe data untuk menampung titik tengah pada Contours
Int	Merupakan tipe data bilangan bulat
M10 / M00 / M01 / M00	Merupakan *spatial moment*

Berikut hasil titik tengah dari *contour* yang dapat dilihat pada Gambar 3.14 Menentukan Titik Tengah *Contour Area* berikut.



Gambar 3.14 Menentukan Titik Tengah *Contour Area*

3.1.5.3. Cara Menggunakan Metode *Skeleton Tracking* dan Metode *Color Detection*

Hasil dari penggunaan metode *skeleton tracking* dan *color detection* menghasil dua buah fungsi, yaitu fungsi kursor dan fungsi klik. Berikut merupakan penjelasan dari setiap fungsi:

1. Fungsi Kursor

Fungsi kursor merupakan fungsional yang dihasilkan oleh metode *skeleton tracking* untuk menggerakan kursor dalam permainan menggunakan pergerakan tangan pemain. Fungsi ini digunakan ketika berada pada antarmuka utama dan antarmuka skor akhir permainan *Augmented Wall*.

2. Fungsi Klik

Fungsi klik merupakan fungsional yang dihasilkan oleh metode *skeleton tracking* dan *color detection* untuk mengklik objek dalam permainan apabila tangan pemain menutupi objek dalam permainan. Untuk mengetahui apakah posisi tangan pemain sudah menutupi objek menggunakan rumus geometri sebagai berikut:

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

Dengan:

x1 = Posisi x tangan

x2 = Posisi x objek

y1 = Posisi y tangan

y2 = Posisi y objek

Setelah mendapatkan hasil perhitungan d, kemudian menentukan nilai radius (r) yang berasal dari setengah dari lebar objek. Apabila nilai d <= r, maka itu berarti posisi tangan pemain menutupi objek. Fungsi ini digunakan ketika berada pada antarmuka *gameplay* permainan *Augmented Wall*.

3.1.6. Analisis Permainan yang akan Dibangun

Permainan *Augmented Wall* yang akan dibangun dalam grafis 2 dimensi, permainan ini mempunyai cara bermain yang mirip dengan permainan Kung Fu Panda Dojo Mojo di Timezone, namun yang membedakan adalah cara

mengendalikan permainannya. Berikut adalah fitur yang ditawarkan oleh permainan ini :

1. *Single player*
2. Permainan dengan grafis 2 dimensi.
3. Pengguna menggunakan gerakan tangan untuk berinteraksi dengan permainan.

3.1.7.1. Deskripsi Konsep Permainan

Permainan *Augmented Wall* ini yang dibangun bersifat *single player*. Permainan diharapkan mempunyai pengalaman bermain yang baru. Tampilan utama permainan disediakan untuk kalibrasi tangan dari pemain. Setelah kedua tangan pemain terdeteksi maka permainan akan di mulai. Pada saat bermain terdapat antarmuka skor untuk melihat berapa banyak skor yang didapatkan oleh pemain dan antarmuka waktu untuk menampilkan waktu yang tersisa pada setiap *stage*. *Augmented Wall* mempunyai sepuluh *stage* yang dapat diselesaikan oleh pemain untuk mendapatkan skor tertinggi. Pada setiap *stage*-nya jumlah objek yang harus disentuh dan munculnya setiap objek akan semakin cepat. Untuk dapat memainkan permainan ini pengguna harus menggerakan tangannya. Pada Tabel 3.9 Deskripsi Konsep Permainan akan menjelaskan deskripsi konsep dari permainan.

Tabel 3.9 Deskripsi Konsep Permainan

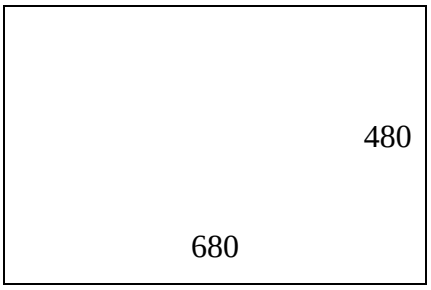
Unsur yang dibangun	Deskripsi
Judul	Permainan *Augmented Wall*
Audio	Audio yang digunakan merupakan musik agar pemain tidak merasa bosan dan efek-efek ketika dalam *gameplay*
Grafis	Menggunakan grafis 2 dimensi
Gambar	Gambar yang digunakan merupakan gambar yang permainan *arcade*
Animasi	Animasi yang digunakan pada permainan ini merupakan animasi sederhana, seperti memperbesar dan memperkecil pada

	tombol navigasi. Animasi tambahan ada saat permainan dimulai.

Untuk menghasilkan permainan *arcade* yang sesuai dengan tujuan utama yaitu dapat mengimplementasikan pengolahan citra, dibutuhkan konsep secara visual untuk mendukung tampilan dan fungsionalitas pada permainan yang terdapat di dalam permainan. Berikut adalah konsep visual *Augmented Wall*:

1. Layar

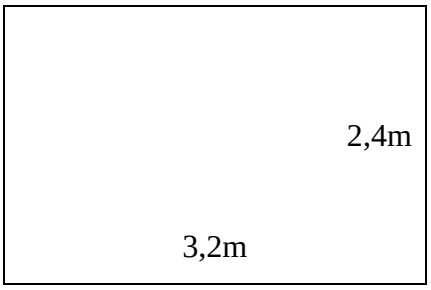
   Resolusi permainan harus dengan ukuran 640px×480px (*fullscreen*) agar sesuai dengan ukuran frame kamera yang ditangkap oleh Kinect.



Gambar 3.15 Resolusi Permainan

2. Tampilan Proyektor

   Tampilan permainan pada proyektor berukuran 2,4m×3,2m.



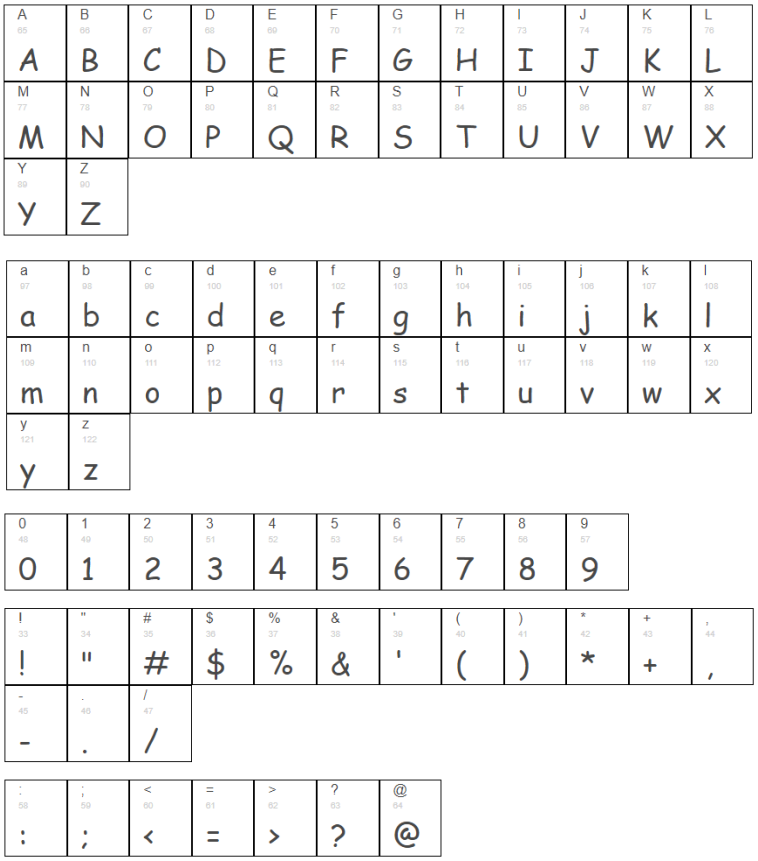
Gambar 3.16 Tampilan Proyektor

3. Tata Letak

   Layout atau tata letak merupakan bagian dari desain visual yang penting sebab layout yang tertata rapi dapat sebagai sarana untuk menyampaikan informasi kepada pengguna agar tidak terjadi kesalahan penyampaian informasi.

4. Huruf

Perancangan huruf yang digunakan dalam permainan ini merupakan Comic Sans MS Reguler Huruf ini digunakan untuk judul, tampilan skor, dan navigasi.

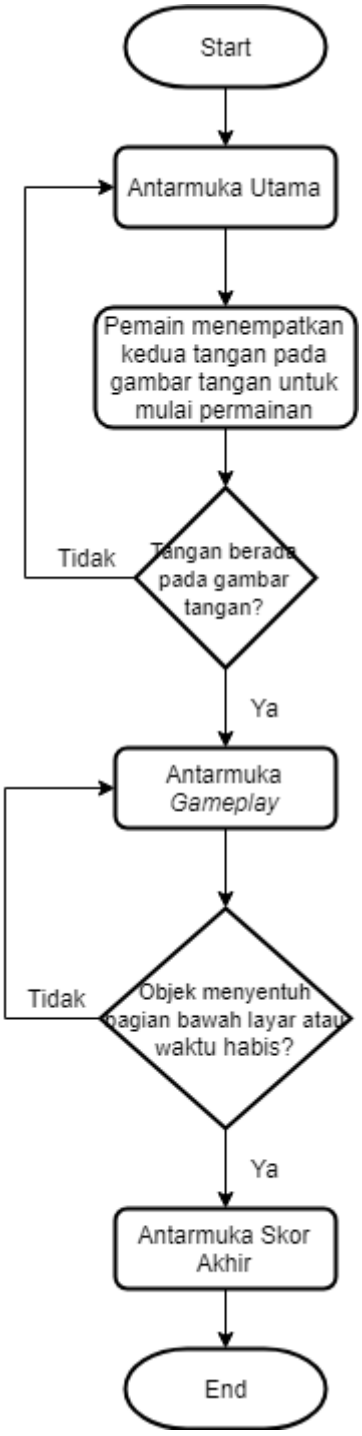


Gambar 3.17 Huruf Comic Sans MS Reguler

3.1.7.2. *Gameplay* Permainan *Augmented Wall*

Permainan *Augmented Wall* ini yang dibangun bersifat *single player*. Tampilan utama permainan disediakan untuk membaca tangan dari pemain. Pada saat di tampilan utama kedua tangan pemain terdeteksi maka permainan akan di mulai. Setelah selesai mendeteksi kedua tangan maka akan muncul tampilan *gameplay* dengan waktu permainan 20 detik untuk setiap *stage*, pemain diminta untuk bisa menekan tombol sebanyak yang telah ditentukan pada setiap *stage*. Jika pemain berhasil menekan tombol maka akan diberikan skor 1. Setelah selesai *gameplay* akan muncul tampilan skor akhir yang berisi skor pemain dan setelah 5

detik akan kembali ke tampilan utama. *Gameplay* permainan *Augmented Wall* dapat dilihat pada Gambar 3.18 *Gameplay* Permainan *Augmented Wall* berikut:



Gambar 3.18 *Gameplay* Permainan *Augmented Wall*

3.1.7.3. Menghitung Skor

Pada setiap permainan harus bisa menyentuh semua objek yang keluar pada layar permainan, akan mendapatkan penilaian berupa angka. Berikut tabel yang dapat menjelaskan perhitungan skor.

Tabel 3.10 Skor Permainan

Aksi yang dilakukan	Skor
Menyentuh objek	1
Tidak menyentuh objek	0

3.1.7.4. *Staging*

Staging disini menjelaskan setiap *stage* yang ada pada permainan. Permainan mempunyai 10 *stage* dengan waktu 20 detik. Pada setiap *stage* jumlah target skor dan kecepatanya berbeda. Kecepatan merupakan perpindahan posisi objek pada setiap *frame*. *Staging* dapat dilihat pada Tabel 3.11 *Staging* berikut.

Tabel 3.11 *Staging*

Stage	Target Skor	Kecepatan
1	10	2
2	15	3
3	20	4
4	25	5
5	30	6
6	35	6
7	40	6
8	45	6
9	50	6
10	55	6

3.1.8. Analisis Kebutuhan Non Fungsional

Analisis ini bertujuan untuk menentukan pengguna dari sistem yang akan dibuat dan mempertimbangkan spesifikasi kebutuhan yang diperlukan. Pada analisis mencakup analisis kebutuhan perangkat keras dan kebutuhan lunak.

3.1.8.1. Analisis Kebutuhan Perangkat Keras

Dalam menjalankan suatu sistem diperlukan perangkat keras yang dapat mendukung proses kerja dari sistem itu sendiri. Pada dasarnya sistem ini dapat dijalankan hanya pada sistem operasi *Windows* (64-bit) yang menggunakan *Kinect*

dan proyektor. Spesifikasi perangkat keras yang dibutuhkan untuk menjalankan sistem dapat dilihat pada Tabel 3.12 Spesifikasi Minimum.

Tabel 3.12 Spesifikasi Minimum Perangkat Keras

No	Spesifikasi	Keterangan
1	*Processor*	Intel Core i3-4005U 1.70 GHz
2	USB	*Dedicated* USB 3.0 BUS
3	RAM	4 GB RAM
4	VGA	NVIDIA GeForce 820M 2GB
5	*Kinect*	*Kinect* versi 1
6	Proyektor	*Brightness*: 3500 ANSI Lumens *Contrast Ratio*: 4000:1 *Lamp*: 170 W, 5000H • *Maximum Ration*: Analog: UXGA (1600 x 1200), Horizontal: 15 – 100 Khz Vertical: 50 – 120 Hz

3.1.8.2. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak merupakan hal yang terpenting dalam mendukung kinerja sebuah sistem. Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar dapat saling berinteraksi di antara keduanya. Perangkat lunak minimum yang dibutuhkan untuk menjalankan sistem dapat dilihat pada Tabel 3.13 Tabel Minimum Spesifikasi Perangkat Lunak berikut.

Tabel 3.13 Tabel Minimum Spesifikasi Perangkat Lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 8 64-bit
2	Program	Kinect SDK versi 1.8
		OpenCV

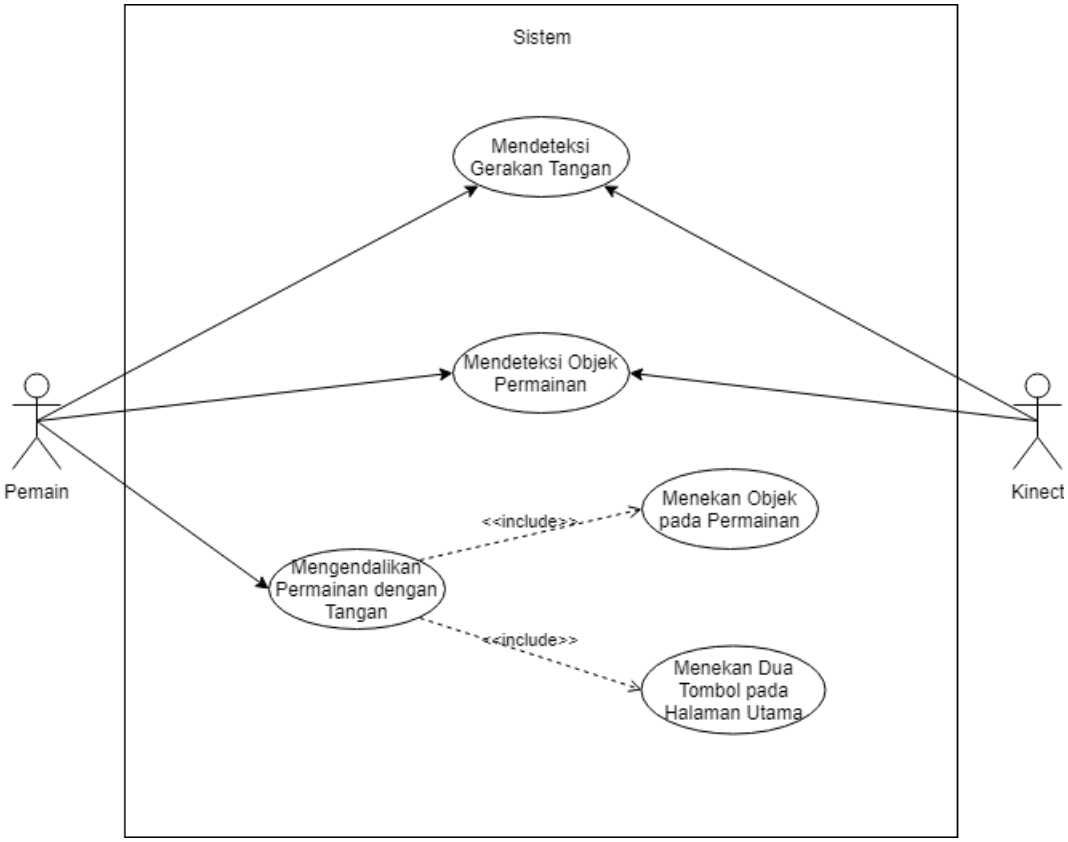
3.1.9. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan perancangan dari beberapa elemen terpisah kedalam satu kesatuan yang utuh dan berfungsi. Alat bantu yang

digunakan untuk menggambarkan sistem secara umum yang akan dibangun yaitu dengan model *use case* dan perancangan antarmuka. Untuk menjelaskan bagaimana suatu masukan di proses pada sistem. Pemodelan sistem dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain *Use Case*, *Use Case Scenario*, *Activity* Diagram, *Sequence* Diagram dan *Class* Diagram.

3.1.8.1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk perilaku sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem aplikasi dan apa saja yang dapat dilakukan menggunakan fungsi – fungsi itu. Dalam *use case* diagram akan membahas tentang identifikasi aktor, identifikasi *use cas*, dan sekanario *use case*. *Use case* diagram dapat dilihat pada Gambar 3.19 *Use Case* Diagram berikut :



Gambar 3.19 *Use Case* Diagram

Dalam sistem ini terdapat dua aktor dalam use case ini, berikut adalah penjelasnnya.

3.2.4.1. Identifikasi Aktor

Aktor yaitu sesorang atau sesuatu seperti perangkat atau sistem lain yang mengakses *use case* dengan berperan sebagai pengguna yang akan menggunakan sistem. Identifikasi aktor dapat dilihap pada Tabel 3.14 Identifikasi Aktor berikut :

Tabel 3.14 Identifikasi Aktor

Aktor	Keterangan
Pengguna	Merupakan aktor dari sistem yang akan dibangun atau pengguna sistem.
Kinect	Merupakan aktor untuk mengambil gerakan tangan dan frame dari kamera.

3.2.4.2. Identifikasi *Use Case*

Identifikasi *use case* dapat dilihat pada Tabel 3.15 Identifikasi *Use Case* berikut :

Tabel 3.15 Identifikasi *Use Case*

No	Use Case	Keterangan
1	Mendeteksi Gerakan Tangan	Fungsional sebagai navigasi di dalam permainan.
2	Mendeteksi Objek pada Permainan	Fungsional untuk mengambil frame kamera dari Kinect yang menghasilkan koordinat setiap objek.
3	Mengendalikan Permainan dengan Tangan	Funsional untuk menekan tombol saat permainan dimulai.
4	Menekan Dua Tombol pada Halaman Utama	Fungsional untuk masuk ke *gameplay*
5	Menekan Objek pada Permainan	Fungsional untuk mendapatkan skor pada saat bermain

3.1.8.2. *Use Case Scenario*

Tabel 3.16 *Use Case Scenario* Mendeteksi Gerakan Tangan

Skenario	
Nomor	1
Nama	Mendeteksi gerakan tangan
Tujuan	Mendeteksi pergerakan tangan
Deskripsi	Menangkap masukkan dari tangan pemain
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada pada tampilan utama
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan antarmuka permainan
2. Menggerakan tangan	
	3. Mendeteksi pergerakan tangan

Tabel 3.17 *Use Case Scenario* Mendeteksi Objek Permainan

Skenario	
Nomor	2
Nama	Mendeteksi objek permainan
Tujuan	Mendeteksi pergerakan objek pada permainan
Deskripsi	Mendeteksi objek dari antarmuka permainan yang ditampilkan proyektor
Aktor	Kinect
Skenario Utama	
Kondisi Awal	Pengguna berada pada tampilan *gameplay*
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan antarmuka utama *gameplay* permainan
2. Menangkap antarmuka permainan yang ditampilkan oleh proyektor	
	3. Mendeteksi objek dari antarmuka permainan yang ditampilkan proyektor

Tabel 3.18 *Use Case Scenario* Mengendalikan Permainan dengan Tangan

Skenario	
Nomor	3
Nama	Mengendalikan permainan dengan tangan
Tujuan	Menampilkan antarmuka dari permainan
Deskripsi	Menampilkan antarmuka dari permainan dan menangkap masukkan dari tangan pemain
Aktor	Pemain & Kinect
Skenario Utama	
Kondisi Awal	Pengguna berada saat bermain
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan antarmuka utama permainan
2. Menempatkan kedua tangan pada tempat yang telah ditentukan	
	3. Menampilkan antarmuka *gameplay* permainan
	4. Mendeteksi pergerakan tangan
5. Menangkap tampilan permainan yang ditampilkan oleh proyektor	
	6. Mendeteksi pergerakan objek pada permainan
	7. Memerika posisi tangan dan objek yang dideteksi
	8. Menampilkan antarmuka skor akhir permainan

Tabel 3.19 *Use Case Scenario* Menyentuh Objek Pada Permainan

Skenario	
Nomor	4
Nama	Menyentuh objek pada permainan
Tujuan	Mendeteksi gerakan tangan dan gerakan objek pada permainan pada saat di antarmuka *gameplay*
Deskripsi	Menangkap masukkan dari tangan dan *frame* kamera dari Kinect

Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pengguna berada saat bermain
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan antarmuka *gameplay* permainan
2. Menggerakan tangan	
	3. Mendeteksi pergerakan tangan
4. Menangkap tampilan permainan yang ditampilkan oleh proyektor	
	5. Mendeteksi pergerakan objek pada permainan
	6. Memerika posisi tangan dan objek yang dideteksi

Tabel 3.20 *Use Case Scenario* Dua Tombol Pada Permainan

Skenario	
Nomor	5
Nama	Mendeteksi gerakan tangan
Tujuan	Memastikan tangan sudah terdeteksi
Deskripsi	Menangkap masukkan dari tangan pemain
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pengguna berada pada tampilan utama
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan antarmuka utama
2. Menempatkan kedua tangan pada lingkaran yang telah ditentukan	
	3. Mendeteksi pergerakan tangan
	4. Menyelesaikan antarmuka utama

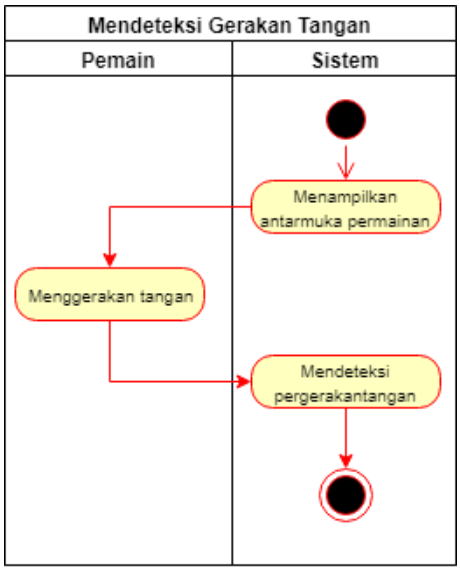
3.1.8.3. *Activity Diagram*

Activity Diagram adalah sebuah tahapan yang lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Setelah

sebelumnya dibuat *use case scenario* untuk memaparkan tahapan aktivitas yang ada dalam suatu *use case*, selanjutnya dibuat activity diagram untuk menggambarkan tahapan aktivitas tersebut.

1. *Activity Diagram* Mendeteksi Gerakan Tangan

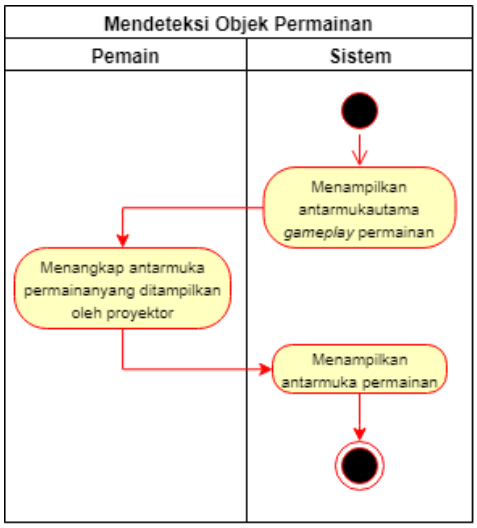
Activity Diagram Mendeteksi Gerakan Tangan dapat dilihat pada Gambar 3.20 *Activity Diagram* Mendeteksi Gerakan Tangan berikut,



Gambar 3.20 *Activity Diagram* Mendeteksi Gerakan Tangan

2. *Activity Diagram* Mendeteksi Objek pada Permainan

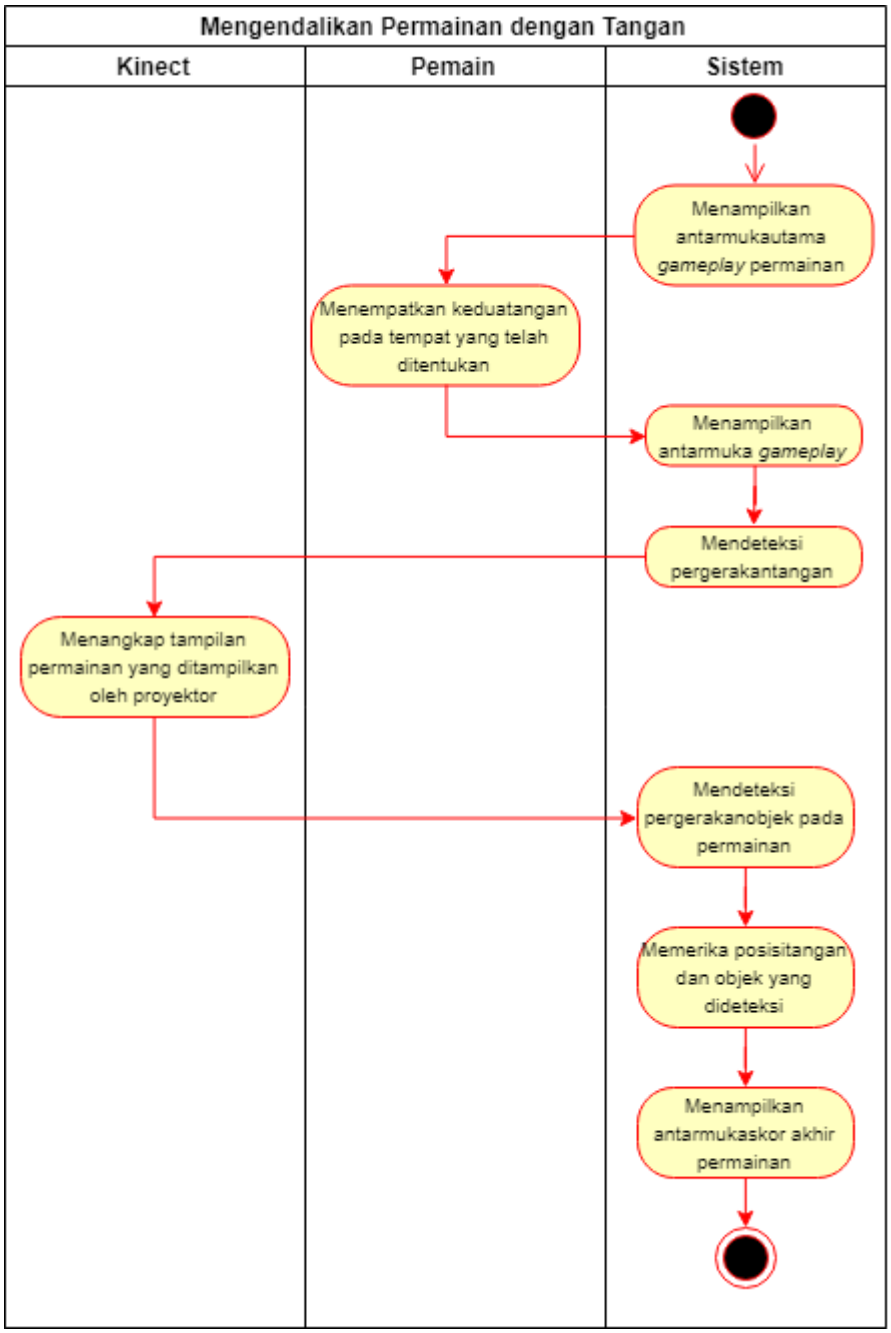
Activity Diagram Mendeteksi Objek pada Permainan dapat dilihat pada Gambar 3.21 *Activity Diagram* Mendeteksi Objek pada Permainan berikut.



Gambar 3.21 *Activity Diagram* Mendeteksi Objek pada Permainan

3. *Activity Diagram* Mengendalikan Permainan dengan Tangan

Activity Diagram Mengendalikan Permainan dengan Tangan dapat dilihat pada Gambar 3.22 *Activity Diagram* Mengendalikan Permainan dengan Tangan berikut.



Gambar 3.22 *Activity Diagram* Mengendalikan Permainan dengan Tangan

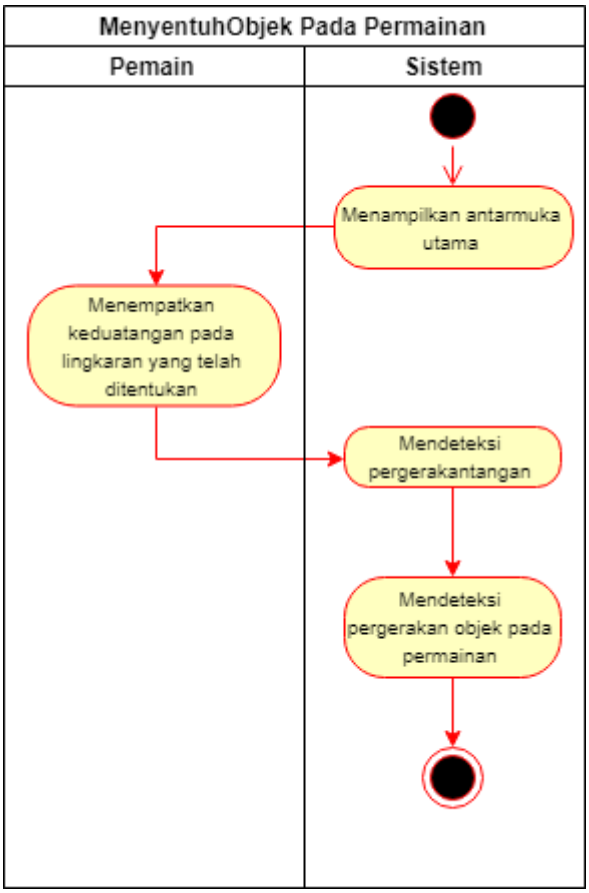
4. *Activity Diagram* Menekan Objek pada Permainan

Activity Diagram Menekan Objek pada Permainan dapat dilihat pada Gambar 3.23 *Activity Diagram* Menekan Objek pada Permainan berikut.

Gambar 3.23 *Activity Diagram* Menekan Objek pada Permainan

5. *Activity Diagram* Menekan Dua Tombol pada Halaman Utama

Activity Diagram Menekan Dua Tombol pada Halaman Utama dapat dilihat pada Gambar 3.24 *Activity Diagram* Menekan Dua Tombol pada Halaman Utama berikut.



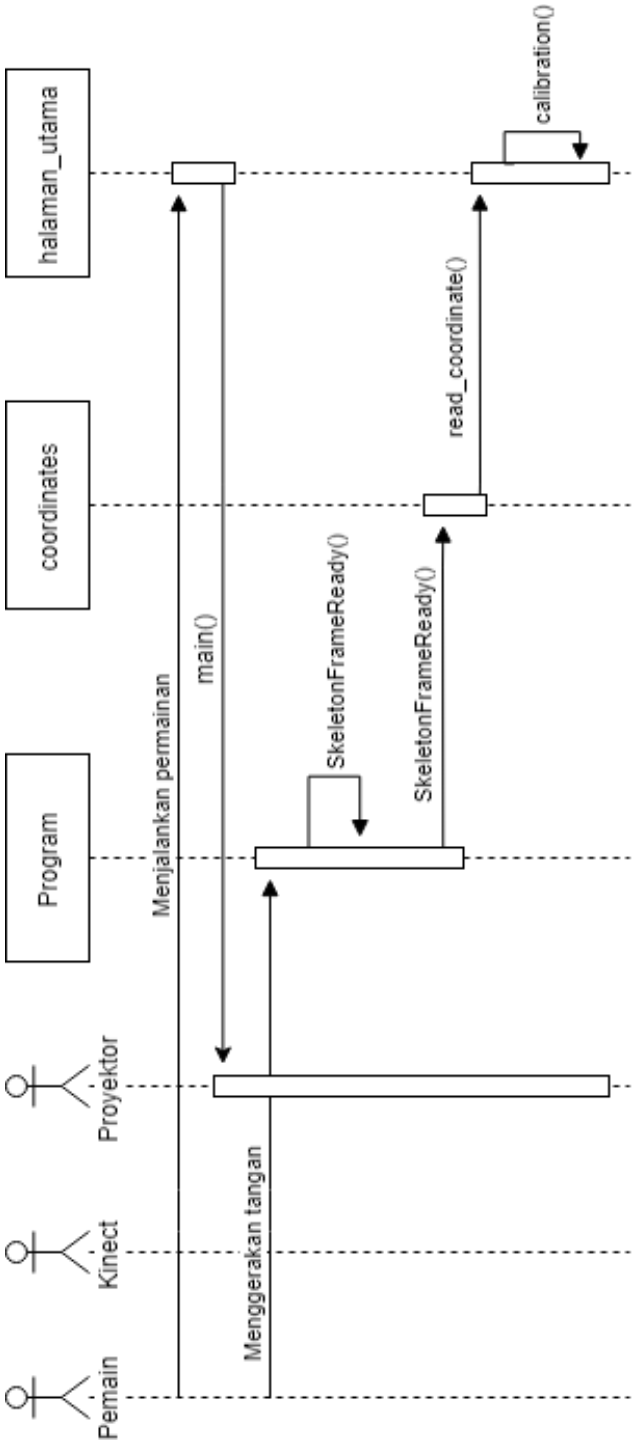
Gambar 3.24 *Activity Diagram* Menekan Dua Tombol pada Halaman Utama

3.1.8.4. *Sequence Diagram*

Diagram sequence dibuat bertujuan untuk menggambarkan interaksi antar objek pada *use case*. Berikut ini diagram sequence yang digunakan dalam membangun sistem pengenal gerakan tangan sebagai alat kendali permainan *Augmented Wall*.

1. *Squence Diagram* Mendeteksi Gerakan Tangan

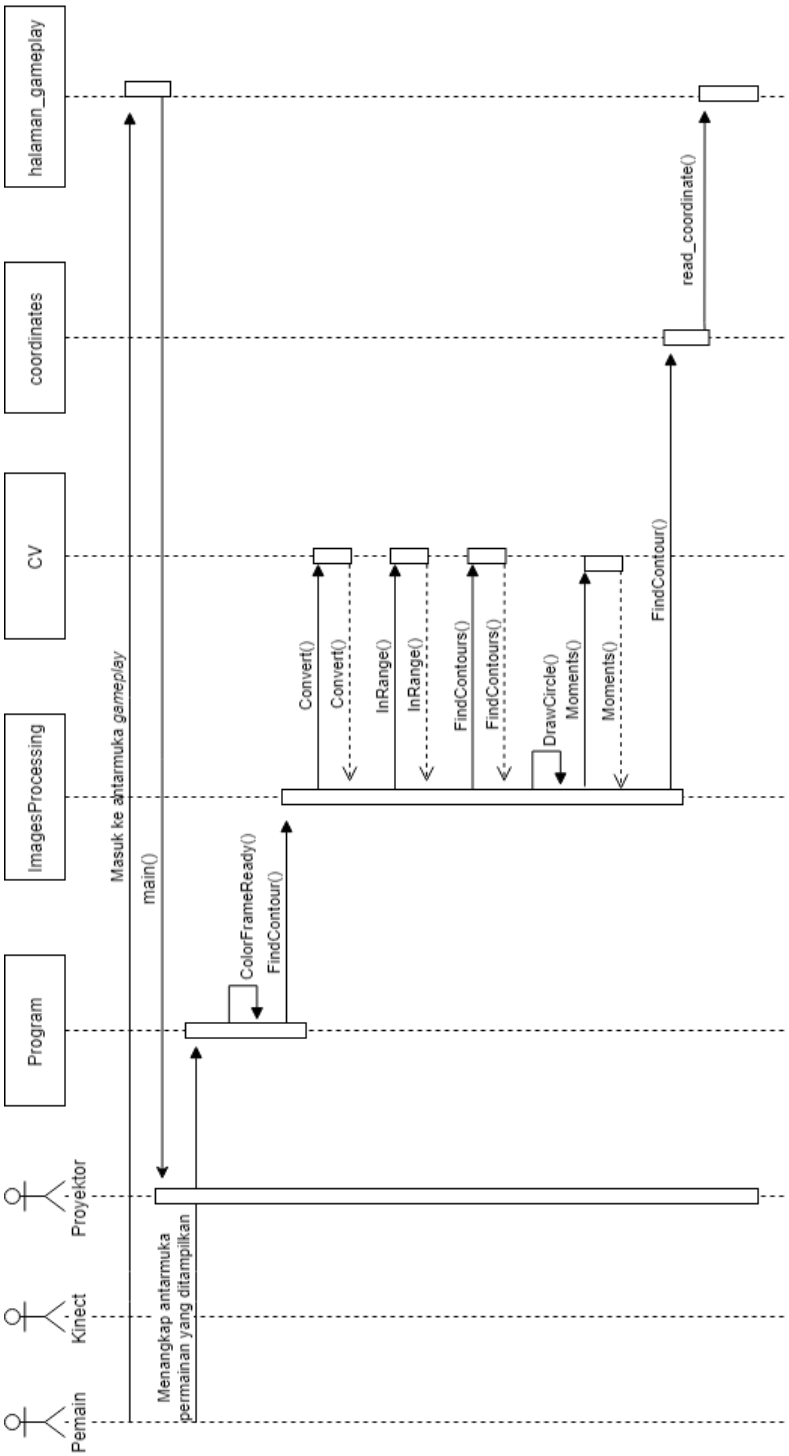
Squence Diagram Mendeteksi Gerakan Tangan dapat dilihat pada Gambar 3.25 *Squence Diagram* Mendeteksi Gerakan Tangan berikut.



Gambar 3.25 *Squence Diagram* Mendeteksi Gerakan Tangan

2. *Squence Diagram* Mendeteksi Objek pada Permainan

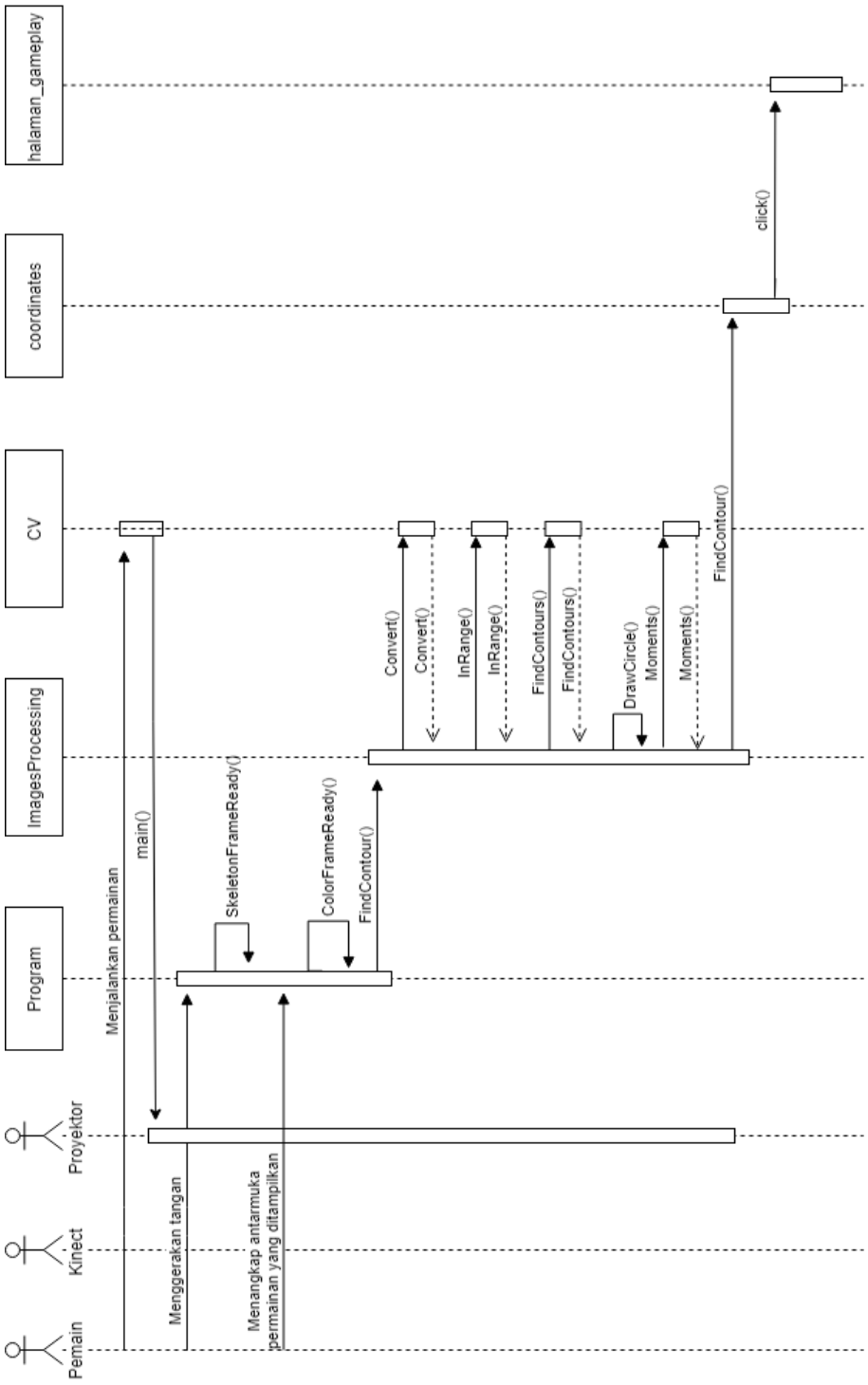
Squence Diagram Mendeteksi Objek pada Permainan dapat dilihat pada Gambar 3.26 *Squence Diagram* Mendeteksi Objek pada Permainan berikut.



Gambar 3.26 *Squence Diagram* Mendeteksi Objek pada Permainan

3. *Squence Diagram* Mengendalikan Permainan dengan Tangan

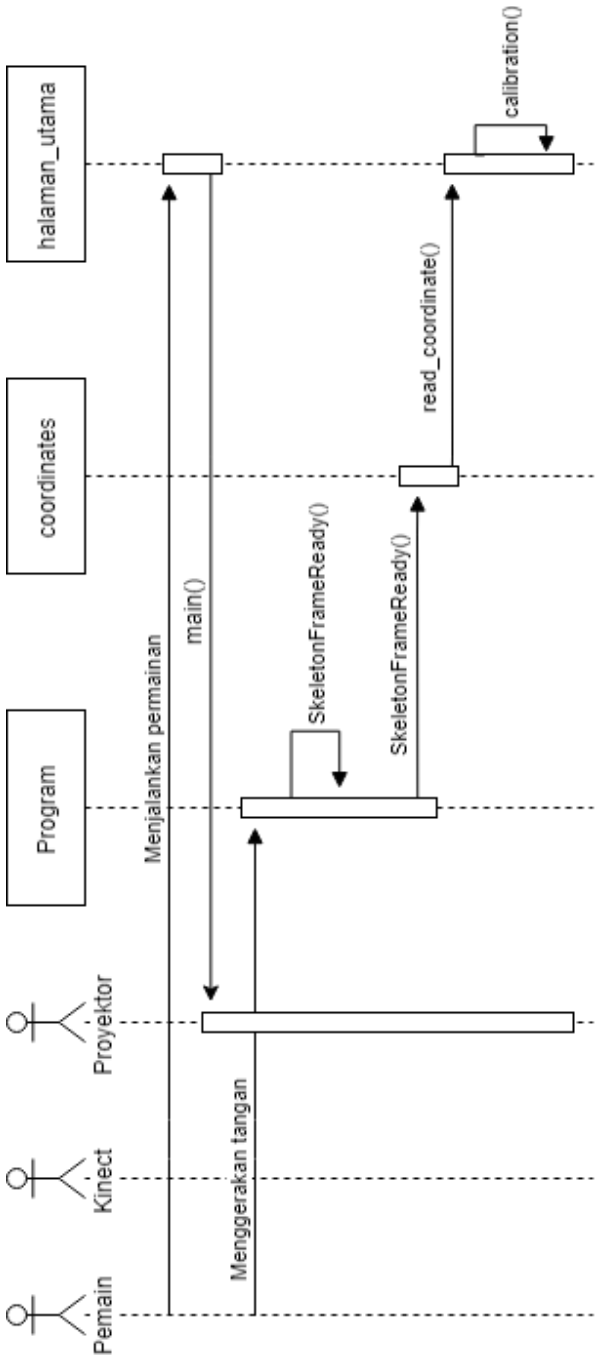
Squence Diagram Mengendalikan Permainan dengan Tangan dapat dilihat pada Gambar 3.27 *Squence Diagram* Mengendalikan Permainan dengan Tangan berikut.



Gambar 3.27 *Squence Diagram* Mengendalikan Permainan dengan Tangan

4. *Squence Diagram* Menekan Dua Tombol pada Halaman Utama

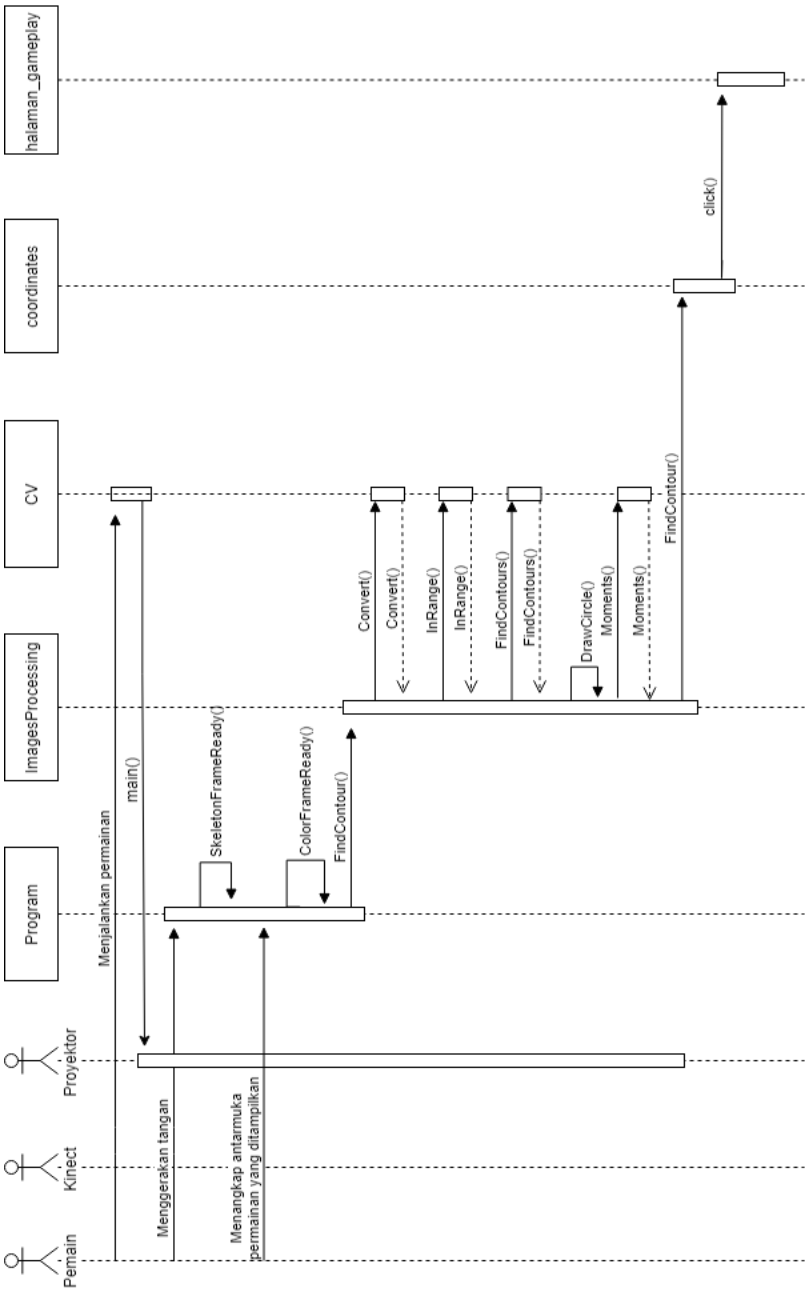
Squence Diagram Menekan Dua Tombol pada Halaman Utama pada Permainan dapat dilihat pada Gambar 3.28 *Squence Diagram* Menekan Dua Tombol pada Halaman Utama berikut.



Gambar 3.28 *Squence Diagram* Menekan Dua Tombol pada Halaman Utama

5. *Squence Diagram* Menekan Objek pada Permainan

Squence Diagram Menekan Objek pada Permainan dapat dilihat pada Gambar 3.29 *Squence Diagram* Menekan Objek pada Permainan berikut.

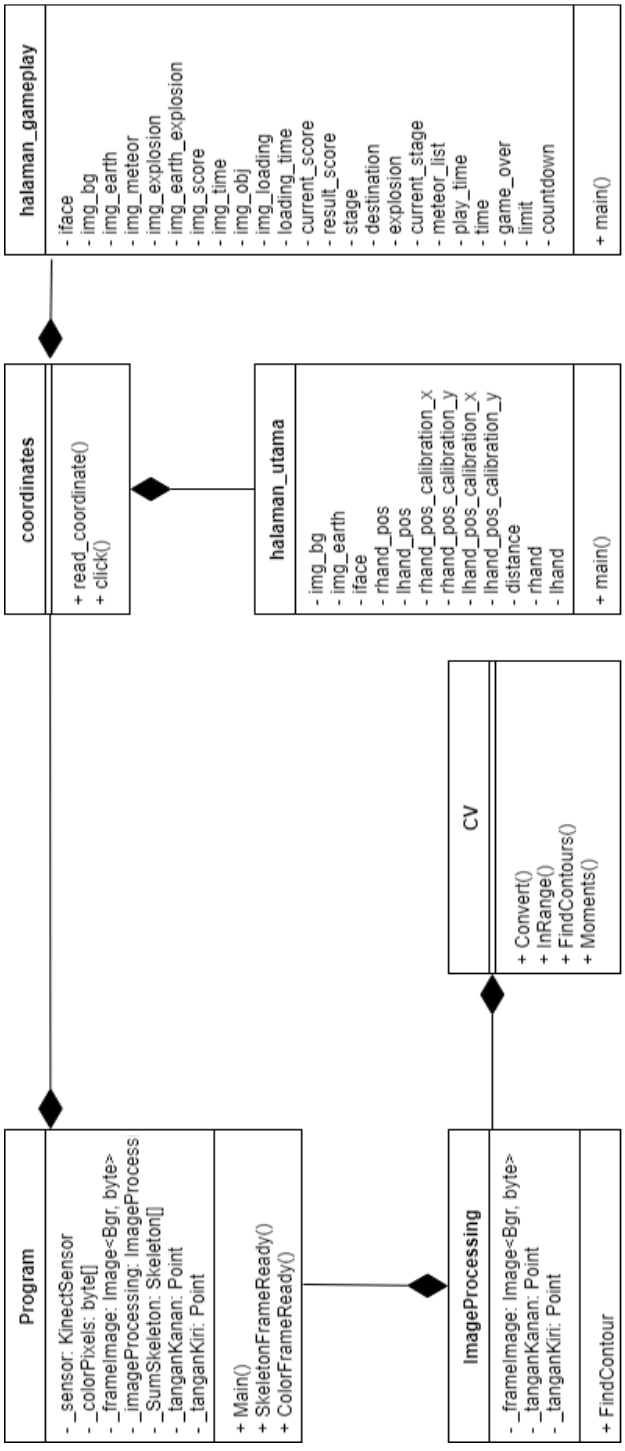


Gambar 3.29 *Squence Diagram* Menekan Objek pada Permainan

3.1.8.5. *Class Diagram*

Class diagram menggambarkan struktur dan hubungan antar objek-objek yang akan dibuat untuk membangun sistem. Kelas masing-masing memiliki atribut

dan metode atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Metode atau operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas. *Class diagram* bisa dilihat pada Gambar 3.30 *Class Diagram* berikut.



Gambar 3.30 *Class Diagram*

3.2. Perancangan Sistem

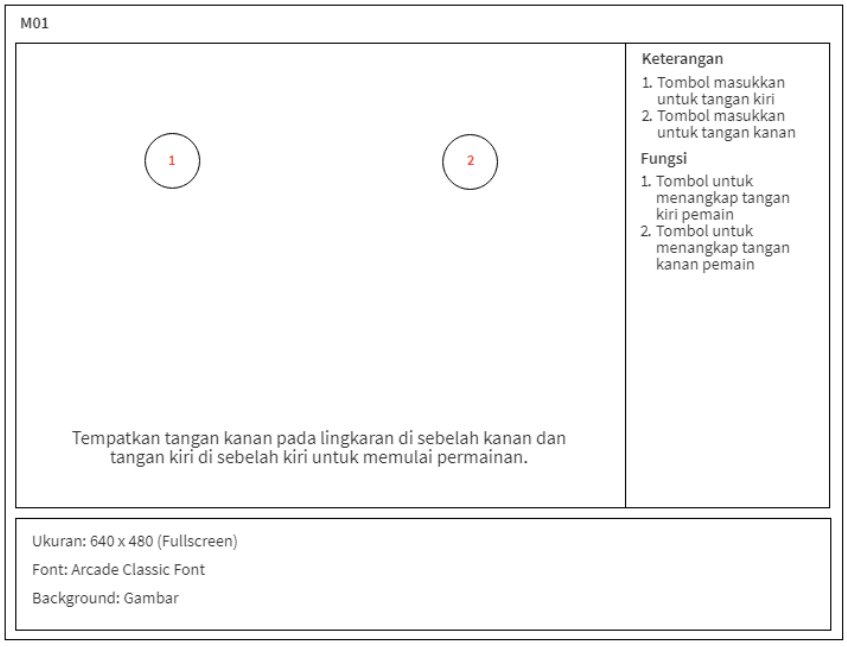
Perancangan sistem untuk menspesifikasi aspek-aspek teknik yang menjadi solusi dalam perencanaan. Pada tahap ini perancangan akan didefinisikan secara detail untuk mengatasi masalah-masalah yang lebih teknis, berkaitan dengan kegiatan implementasi seperti perancangan antarmuka.

3.2.1. Perancangan Antarmuka Sistem

Tahap perancangan antarmuka bertujuan untuk memberikan gambaran tentang permainan yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan serta akan memudahkan dalam pembuatan. Perancangan antarmuka ini hanya perancangan untuk antarmuka permainan karena sistem alat kendali tidak mempunyai antarmuka. Berikut ini gambaran perancangan antarmuka yang ada dari aplikasi yang akan dibangun yaitu:

1. Halaman Antarmuka Utama

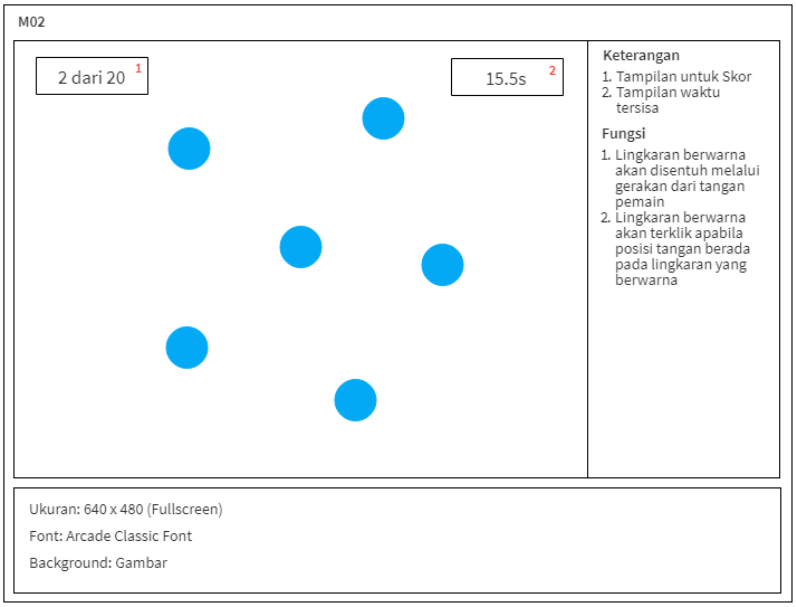
   Perancangan antarmuka utama dapat dilihat pada Gambar 3.31 Antarmuka Menu Utama berikut.



Gambar 3.31 Antarmuka Menu Utama

2. Halaman Antarmuka *Gameplay*

Perancangan antarmuka *gameplay* dapat dilihat pada Gambar 3.32 Antramuka *Gameplay* berikut.



Gambar 3.32 Antramuka *Gameplay*

3. Halaman Antarmuka Skor Akhir

Perancangan antarmuka skor akhir dapat dilihat pada Gambar 3.33 Antarmuka Skor Akhir berikut.

Gambar 3.33 Antarmuka Skor Akhir