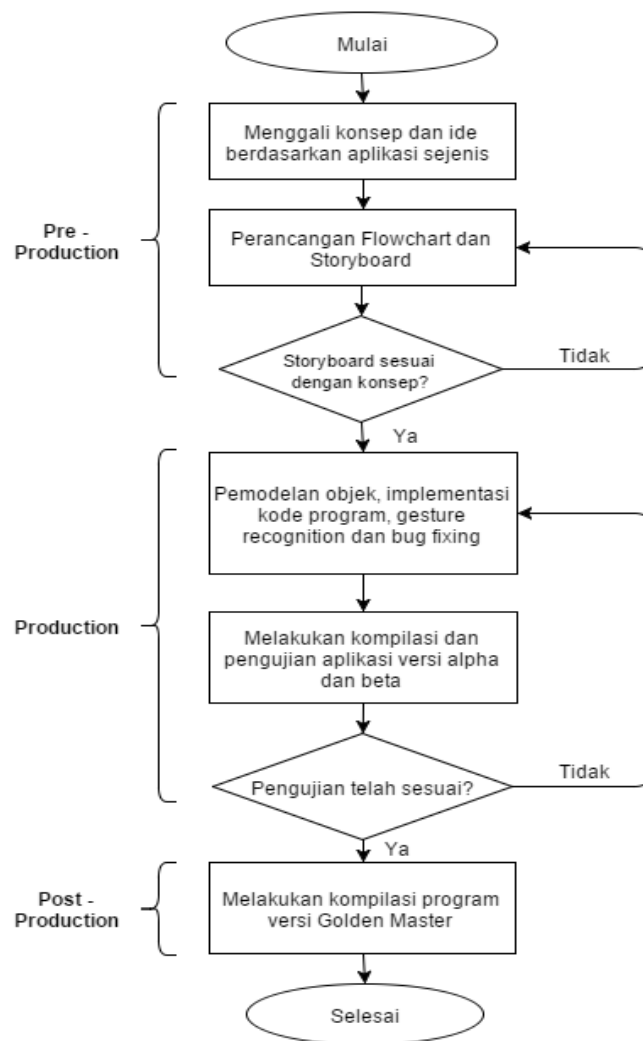


BAB III

PERANCANGAN APLIKASI

3.1 Kerangka Penelitian

Pada kerangka penelitian penulis menjabarkan perancangan aplikasi yang akan dibuat berdasarkan pada metode pengembangan aplikasi multimedia yaitu *Multimedia Development Life Cycle* (MDLC).



Gambar 3.1 Kerangka Penelitian

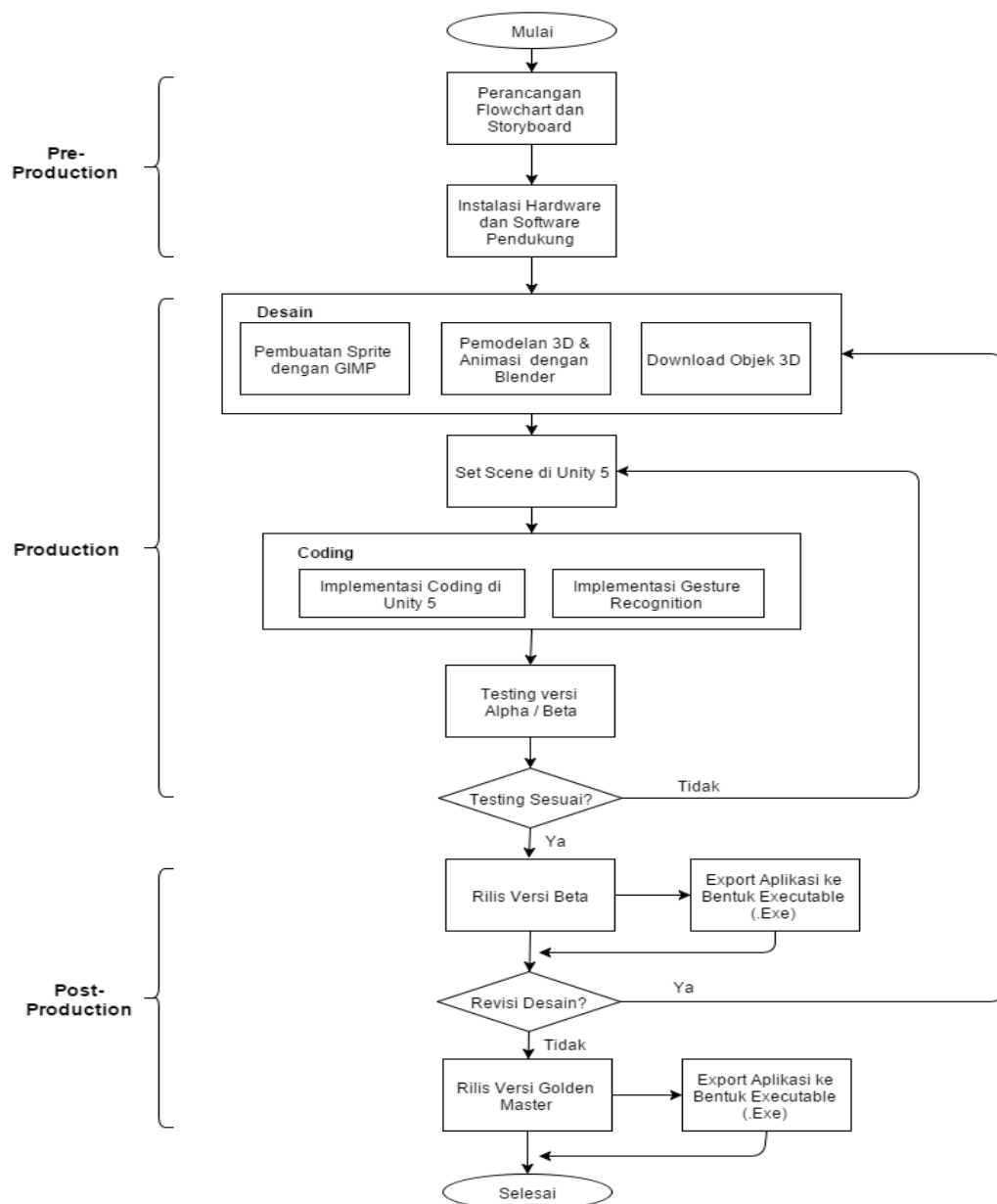
Dari flowchart pada Gambar 3.1 dapat dijelaskan tahap-tahap perancangan sebagai berikut:

1. Kerangka penelitian diawali dengan tahap pra produksi. Konsep dan ide digali dengan mengacu pada aplikasi-aplikasi pembanding yang sudah ada.
2. Perancangan flowchart dan storyboard aplikasi diperlukan untuk menyampaikan konsep aplikasi multimedia secara jelas, terstruktur dan lebih akurat.
3. Setelah perancangan storyboard telah sesuai dengan desain konsep, maka selanjutnya penulis melakukan implementasi kode, graphic design, 3D modelling dan animasi untuk mendukung aplikasi multimedia.
4. Pengujian pertama dilakukan dengan merilis atau mengkompilasi program versi alpha. Kompilasi program versi alpha lalu dilanjutkan versi beta. Pengujian aplikasi versi alpha dan beta disesuaikan dengan standar konsep yang telah ditentukan sebelumnya.
5. Setelah pengujian versi alpha dan beta sesuai, dilakukan revisi desain akhir untuk menuju versi golden master yaitu versi final yang siap untuk digunakan.

3.2 Tahap Perancangan

3.2.1 Perancangan Aplikasi

Pada tahap ini penulis menjabarkan perancangan aplikasi yang akan dibuat juga berdasarkan metode MDLC. Tahap ini merupakan penjelasan teknis dari kerangka penelitian yang digambarkan pada flowchart berikut :



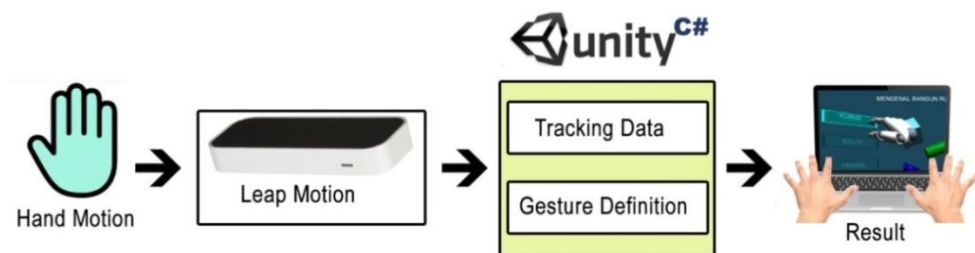
Gambar 3.2 Flowchart Tahap Perancangan

Dari flowchart pada gambar 3.2 dapat diuraikan sebagai berikut :

1. Perancangan Storyboard adalah tahap pembuatan gambar sketsa perencanaan untuk menunjukkan secara visual bagaimana tampilan dan detail properti setiap level atau stage.
2. Instalasi Hardware dan *Software* pendukung yaitu pemasangan *Leap Motion* dan penginstalan SDK serta *software* lain pendukung pembuatan aplikasi.
3. Desain Grafis adalah proses pembuatan image 2D dengan menggunakan *software* GIMP untuk keperluan perancangan *button*, *sprite* dan *texture*.
4. Pada tahap ini dilakukan perancangan 3D model untuk bangun ruang dan objek pendukung lain serta animasi yang dibuat dengan Blender.
5. Set Scene adalah tahap perancangan yang dimulai dengan mendesain interface pada scene Unity. Interface didesain dengan menyiapkan objek-objek game yang diletakan pada folder Assets.
6. Implementasi kode adalah penerapan algoritma aplikasi kedalam bahasa pemrograman C#. Debugging & Bug Fixing pada tahap ini adalah melakukan pengecekan efesiensi kode serta perbaikan jika diperlukan.
7. *Hand Gesture Recognition* adalah metode pengenalan gestur atau gerakan tangan yang diperoleh dari data *hand motion tracking* lalu dikonversi menjadi perintah-perintah tertentu untuk menggantikan input mouse maupun keyboard pada komputer
8. Revisi Desain dilakukan jika desain belum sesuai dengan storyboard serta mengutamakan aspek interface yang user friendly.

3.2.2 Hand Gesture Recognition dengan *Leap Motion*

Hand Gesture Recognition dengan menggunakan *Leap Motion* memiliki beberapa keuntungan diantaranya adalah tidak perlu menggunakan marker . Dengan demikian data *hand motion tracking* lebih mudah diolah dan menghasilkan tingkat efisiensi dan ketepatan yang tinggi.



Gambar 3.3 *Hand Motion Tracking* dengan *Leap Motion*

Keterangan :

1. *Leap Motion* dipasang ke PC melalui kabel USB. Jika lampu indikator berwarna hijau, artinya perangkat telah aktif sehingga *tracking data* dapat dilakukan.
2. Citra tangan dan gerakannya (*hand motion*) akan direkam oleh *Leap Motion Device* sehingga menghasilkan *tracking data* berupa posisi tangan pada sumbu acuan x,y,z. Karena *motion tracking* bersifat *realtime*, maka gerakan *Virtual Hands* yang ada pada aplikasi akan mengikuti gerakan tangan.
3. Unity dengan bahasa pemrograman C# akan menyimpan kode perintah dan logika untuk mengenal gerakan tangan (*Gesture Definition*). Kumpulan logika inilah yang dinamakan dengan metode sintaks.
4. Hasilnya, user dapat mengontrol objek, maupun menu navigasi di dalam aplikasi yang dapat digunakan untuk mempelajari bangun

ruang. Kontrol yang dihasilkan sesuai definisi gestur yang diterjemahkan.

3.3 Spesifikasi Hardware

Dalam pembangunan dan implementasi aplikasi diperlukan perangkat keras sebagai berikut :

3.3.1 PC

Dalam pembuatan aplikasi multimedia berbasis 3D, diperlukan PC dengan spesifikasi yang mumpuni untuk menangani *motion tracking* secara *real-time* dengan performa yang baik.



Gambar 3.4 PC yang digunakan dalam pembuatan aplikasi

Tabel 3.1 Spesifikasi Hardware PC

Notebook Series	Asus A43SM-VX034D
Processor	Intel Core i5-2450M
Graphic Card	NVIDIA GeForce GT 630M, 2 GB DDR3
RAM	8 GB, DDR3
Operating System	Windows 8.1 Enterprise 64-bit
Layar	14" HD Color Shine (LED) , res (1366 x 768)
Drive Optik	Super-Multi DVD
Hard Disk Drive	500 GB (5400 rpm)
Audio	Altec Lansing Speaker

3.3.2 *Leap Motion Controller*

A. Hardware



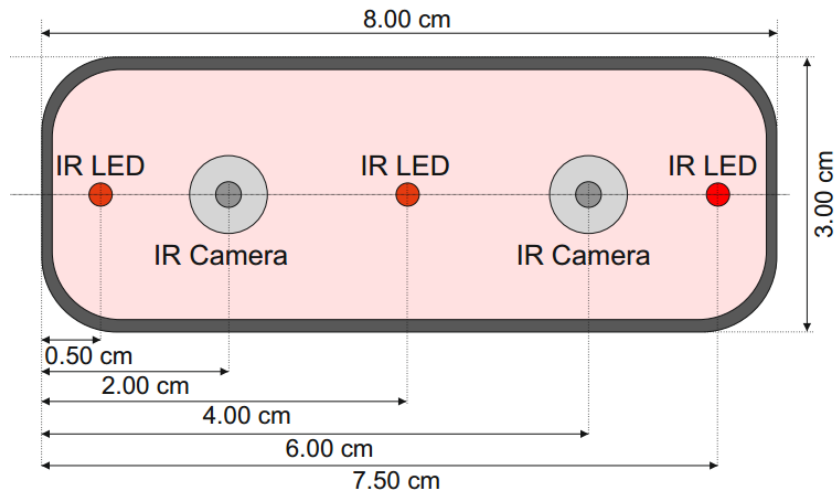
Gambar 3.5 Perangkat *Leap Motion Controller*



Gambar 3.6 Ilustrasi Komponen *Leap Motion Controller*

Sumber: <http://leapmotion.com>

B. Spesifikasi



Gambar 3.7 Sketsa struktur perangkat *Leap Motion*

Sumber: <http://leapmotion.com>

Dimensi (P x L x T) : 7,5 x 3,0 x 1,1 cm

Koneksi Peripheral : USB 3.0

Lampu LED : 3 Buah

Sensor : 2 Kamera Infra Red Monokromatik

Leap Motion berukuran relatif kecil, mudah untuk dibawa-bawa dan tidak memerlukan tempat yang luas. Hanya saja untuk membuat posisi tubuh nyaman dalam menggunakannya, sebaiknya perangkat diletakkan ditempat yang datar dan sejajar dengan keyboard komputer. Sedangkan fitur kamera inframerah membuat *Leap Motion Controller* dapat melakukan deteksi terhadap citra tangan maupun benda-benda lain walau dalam keadaan tidak ada cahaya.

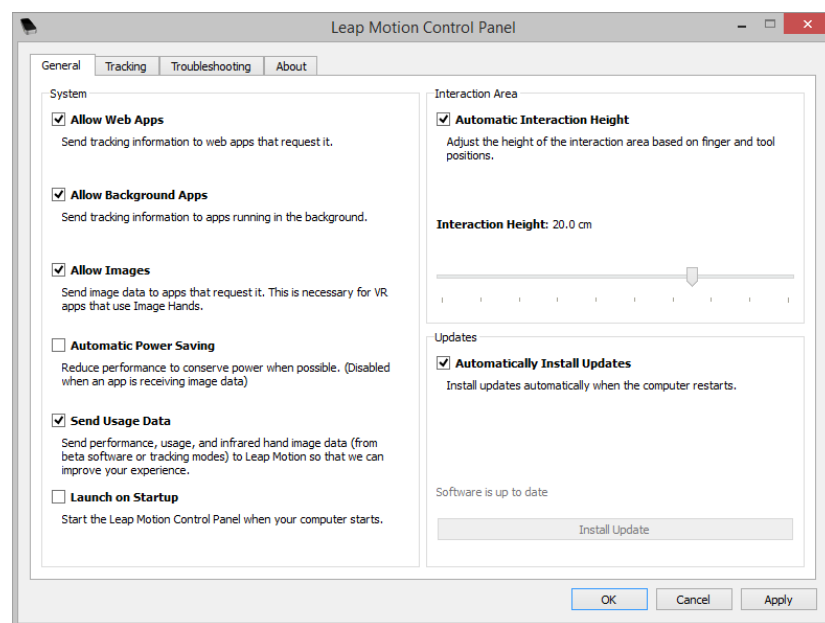
3.4 Spesifikasi Software

Software yang digunakan dalam pembuatan aplikasi mencakup SDK dari *Leap Motion*, Unity 5 sebagai game engine, GIMP sebagai image editor, dan Blender untuk pemodelan objek 3D dan animasi.

3.4.1 Leap Motion Core Services (SDK)

Leap Motion Core Services adalah paket *software* yang termasuk dalam *Leap Motion* SDK meliputi :

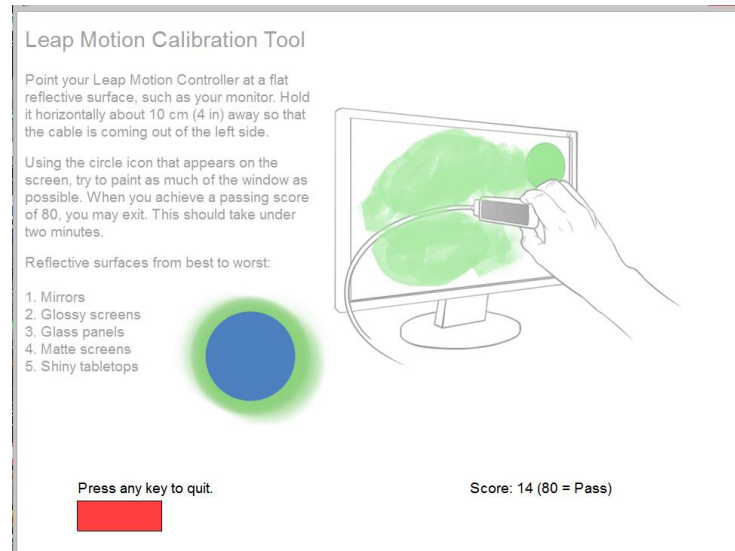
A. LeapControlPanel.exe



Gambar 3.8 *Leap Motion Control Panel*

Leap Motion Control Panel adalah modul aplikasi pengaturan sensor *Leap Motion*. User dapat melakukan pengaturan *feature permission*, area interaksi, *tracking* dan *troubleshooting*.

B. Recalibrate.exe

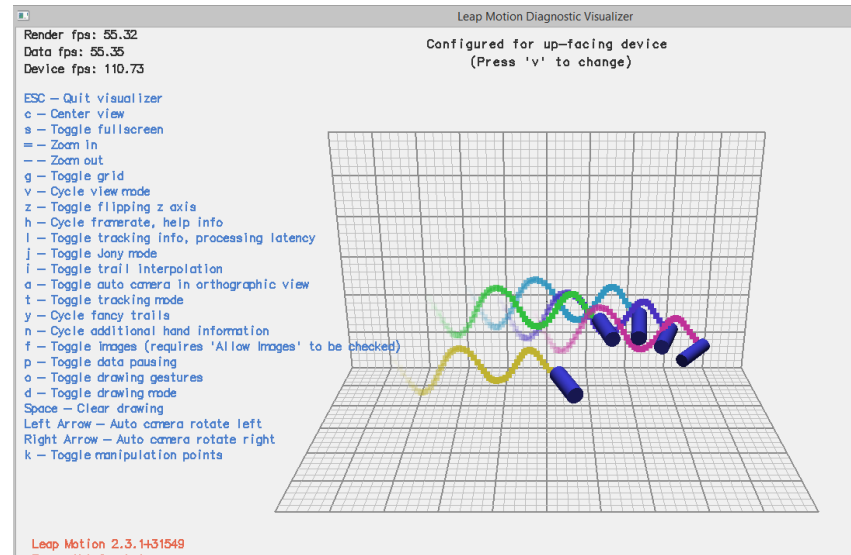


Gambar 3.9 *Leap Motion Calibration Tool*

Leap Motion Calibration Tool digunakan untuk melakukan kalibrasi terhadap sensor supaya sensor meninjau ulang kepekaan dan sensitifitasnya. Tools ini berguna jika user merasa perangkat sensor tidak menginterpretasikan gerakan tangan dengan baik.

Untuk melakukan proses recalibrate , user dapat membuka lalu memposisikan *Leap Motion* sejajar menghadap layar komputer. Setelah itu, lakukan gerakan seperti “mewarnai” keseluruhan layar hingga *score* mencapai angka 80.

C. VisualizerApp.exe



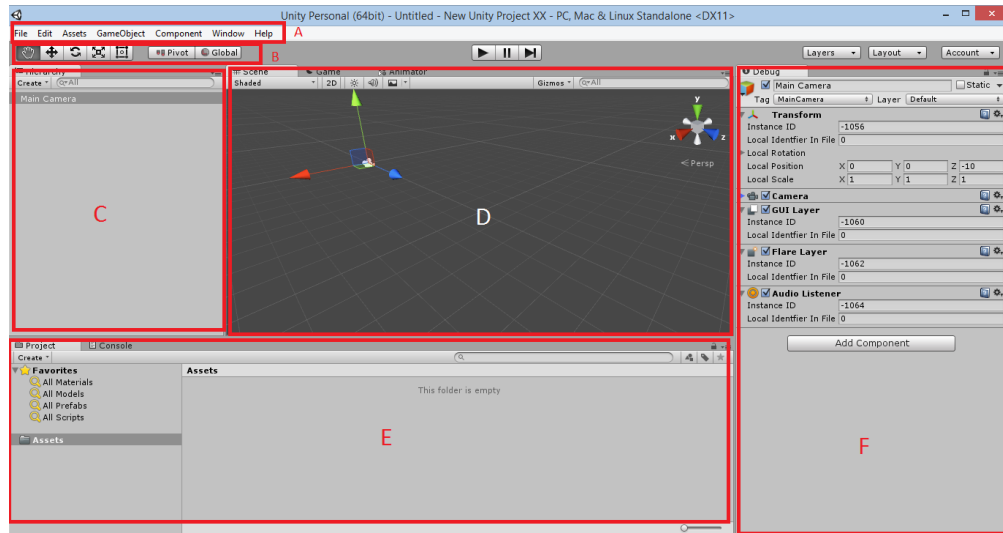
Gambar 3.10 *Leap Motion* Diagnostic Visualizer

Leap Motion Diagnostic Visualizer adalah aplikasi yang disiapkan oleh vendor untuk para pengembang guna memberikan informasi tentang proses hand motion tracking yang dikerjakan oleh perangkat *Leap Motion*. Dengan aplikasi ini, pengembang dapat mengetahui data-data yang dihasilkan dari proses tracking seperti *render fps*, *data fps*, *device fps*, *skeleton*, koordinat tangan secara *real time*, output kamera infrared dan informasi lain yang sudah disajikan cukup menarik secara visual.

3.4.2 Unity 3D Versi 5.1.1f1

Unity adalah game engine lintas platform yang berisi paket-paket tools yang dapat digunakan sebagai aplikasi untuk membuat game atau aplikasi multimedia lain baik dalam bentuk 2D maupun 3D. Salah satu fitur Unity untuk mempermudah programmer dalam membuat aplikasi / game adalah Component. Unity telah menyediakan paket-paket umum non-script maupun script untuk mengembangkan game 2D maupun 3D.

A. Workspace Unity 5



Gambar 3.11 Workspace Unity 5

- A. Menu Bar :
Berisi menu utama pada Unity. Terdiri dari File, Edit, Assets, GameObject, Component, Window dan Help.
- B. Tool Bar
 1. Zoom Tool = Memperdekat/jauh penampakan
 2. Move Tool = Memindahkan/menggeser objek 3D
 3. Rotate Tool = Memutar objek 3D
 4. Scale Tool = Memperbesar/memperkecil ukuran objek 3D
- C. Hierarchy View:
Susunan objek-objek pada game yaitu objek parent dan child
- D. Game Window
Set *world game* atau tampilan game.
- E. Project View
Kumpulan objek-objek yang dapat dipakai dalam game. Mulai dari script, model 3D, sprite, texture, prefab dan sebagainya Umumnya secara default ada pada folder Assets.
- F. Debug / Inspector

Pemperlihatkan proses debugging game secara dinamis dan nilai (value) variabel yang ada pada game.

B. Kelebihan dan Kekurangan Unity 5

Unity 5 memiliki beberapa kelebihan dibandingkan game engine lainnya diantaranya adalah ;

- Unity 3D dapat digunakan untuk melakukan development terhadap aplikasi sederhana hingga yang kompleksitasnya tinggi.
- Unity 3D mendukung bahasa pemrograman C# yang sudah mengacu pada pola pemrograman berorientasi objek.
- Manajemen file yang baik terlihat pada sistem file pada unity yang dapat dikelompokkan kedalam folder dan sub folder.
- Memiliki asset store dengan developer yang cukup banyak aktif melalui forum online.
- Fitur ekspor ke banyak platform OS seperti Windows, android, linux dan dapat pula dimainkan lewat web browser dengan tambahan plugin unity web player.
- Konversi kode menjadi GUI (Graphic User Interface)

Namun ada beberapa kekurangan yaitu:

- Beberapa assets seperti Shader, texture, model hanya tersedia jika membayar lisensi Unity Pro.
- Unity 3D tidak dapat melakukan modelling atau desain objek. Proses modelling atau desain dilakukan di *software* lain seperti blender atau 3DS MAX.

3.4.3 Blender 2.68a

Walaupun blender bersifat open source, namun *software* ini memiliki beberapa keunggulan yang tidak kalah dari *software* berbayar lainnya. *Software* ini dapat diunduh dan digunakan secara gratis pada website resminya di www.blender.org. Blender selalu mengeluarkan versi terbarunya dalam jangka waktu yang berdekatan. Setiap versi baru yang dikeluarkan, selalu ada tools baru yang ditambahkan beserta bug-bug yang sudah diperbaiki. Perkembangan Blender sangat pesat karena banyak developer yang terlibat dalam pengembangan Blender. Para developer ini tergabung dalam suatu komunitas terbuka yang anggotanya bebas dan tersebar di seluruh dunia.

Keunggulan lain dari Blender adalah kesederhanaannya. Kesederhanaan *software* ini membuat Blender tidak memerlukan spesifikasi komputer yang tinggi untuk sekedar menjalankannya. File instalasinya yang kecil membuat blender juga dapat diinstall dan dijalankan secara portable. Sekitar 40 - 60 Mb sebelum diinstall dan 150 Mb setelah di install. Ukuran file yang kecil ini jauh melampaui *software* sejenis yang file instalasinya membutuhkan ruang hardisk sekitar 2 hingga 3 Gb.

3.4.4 GIMP 2.8

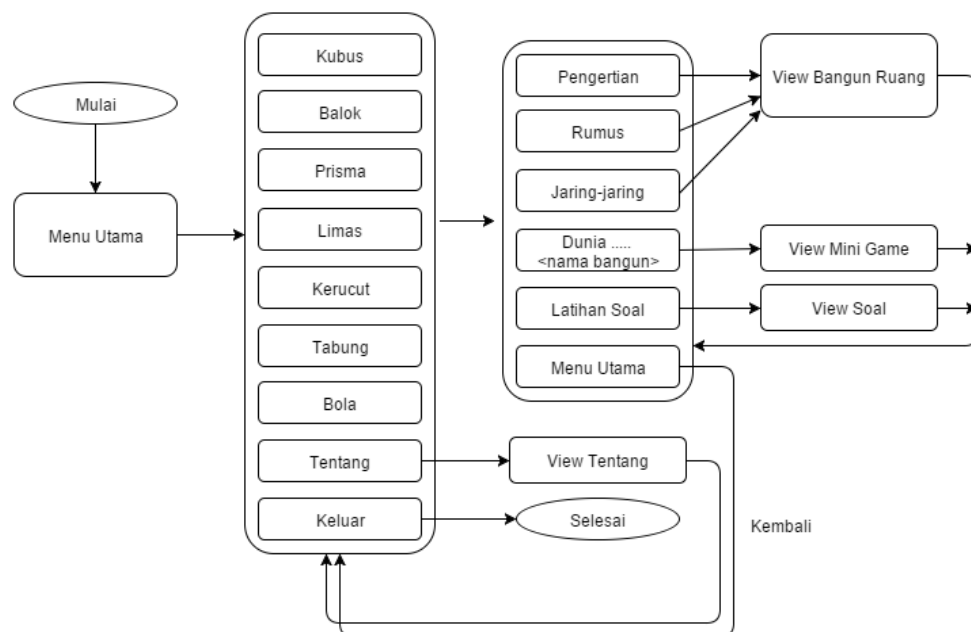
Gimp adalah sebuah aplikasi pengolah gambar raster atau bitmap dan program manipulasi gambar yang bisa berjalan di semua platform sistem operasi. Gimp merupakan singkatan dari GNU Manipulation Program, aplikasi ini bersifat Free Open Source *Software*, sehingga kita dapat mendapatkan aplikasi ini secara gratis tanpa harus membayar lisensi dari si pembuat program. Gimp sangat cocok digunakan untuk *image manipulation tasks*, termasuk *photo retouching*, *image composition*, dan *image construction* ^[23].

3.5 Tahap Pre-Production

Tahap Pre-Production adalah tahap awal pada siklus pengembangan aplikasi multimedia berbasis metode MDLC. Pada tahap ini, dilakukan penggambaran konsep aplikasi yang telah digali sebelumnya melalui flowchart dan storyboard.

3.5.1 Perancangan Flowchart Aplikasi

Flowchart aplikasi menggambarkan alur program Pengenalan Bangun Ruang. berupa menu-menu dan *view* (*scene*).

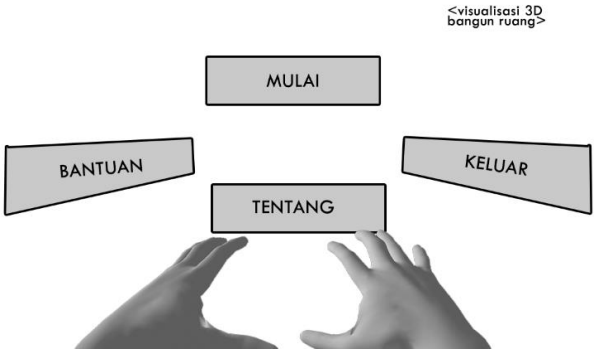



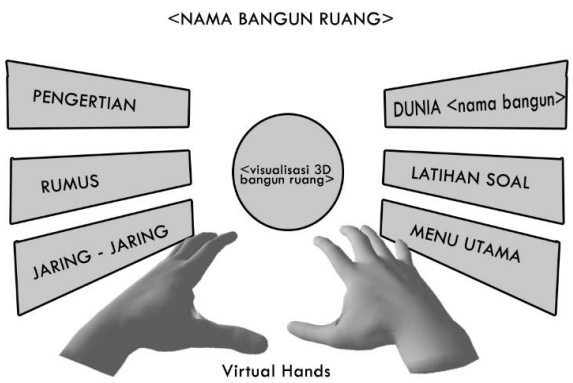
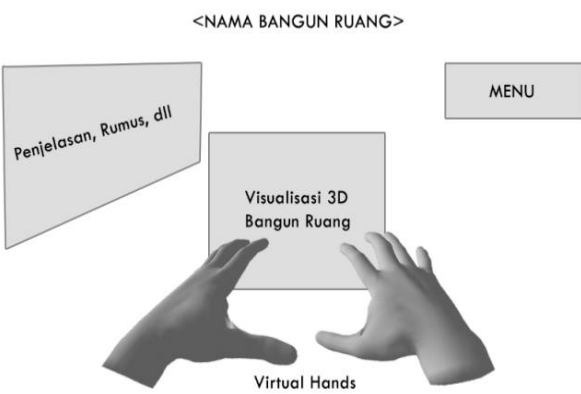
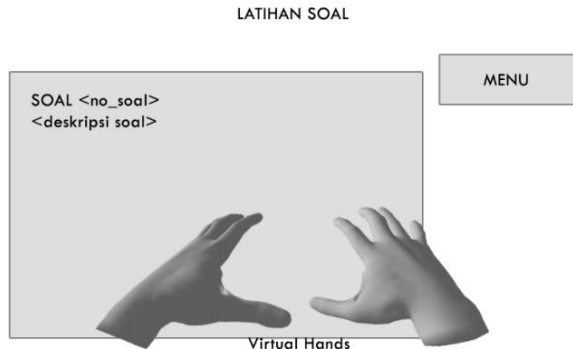
Gambar 3.12 Flowchart Aplikasi

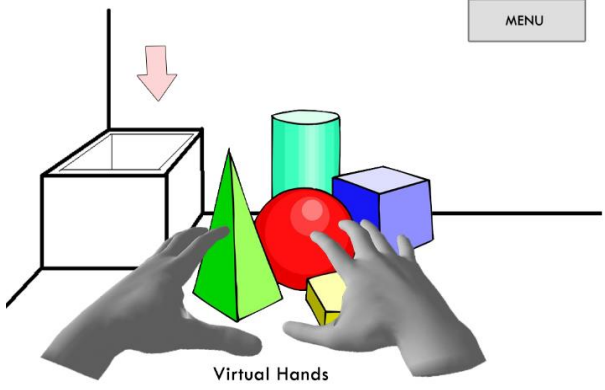
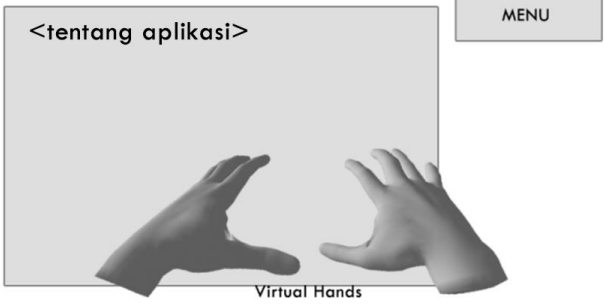

Pada scene menu utama terdapat beberapa menu pilihan yaitu tipe bangun ruang yang disajikan meliputi kubus, balok, prisma, limas, kerucut, tabung dan bola. Sedangkan pada menu bangun ruang, user dapat memilih sajian materi berupa pengertian, rumus, jaring-jaring dan latihan soal.

3.5.2 Perancangan Storyboard Aplikasi

Storyboard aplikasi pengenalan bangun ruang terdiri dari 6 *scene* yang menjadi antarmuka utama. Setiap *scene* mengandung beberapa objek yang mendukung navigasi seperti *virtual button* dan juga *virtual hands*.

Scene	Keterangan
<p>scene 01</p> 	<p>MENU UTAMA</p> <p>Animasi : Bangun ruang berputar</p> <p>Gestur : <i>Key Tap</i></p> <p>Suara : audiobg.ogg</p> <p>User dapat memilih menu untuk memilih menu dengan menyentuh button melalui <i>virtual hand</i>.</p>
<p>scene 02</p>  <p>sumber: http://i.imgur.com/3KqRqMp.png</p>	<p>MINI GAME 1</p> <p>Animasi : Robot Terbang</p> <p>Gesture : <i>Fly Control</i></p> <p>Suara : JetSound.mp3</p> <p>User ditugaskan untuk melakukan eksplorasi di dalam sebuah kota, dimana user harus mencari stasiun pengenalan bangun ruang. Setiap stasiun, memiliki sebuah komputer canggih yang menjelaskan tentang sebuah bangun ruang.</p>

<p>scene 03</p> 	<p>STASIUN BANGUN RUANG</p> <p>Animasi : Bangun ruang berputar</p> <p>6 buah 3D button</p> <p>Gestur : <i>Key Tap, Circle</i></p> <p>User dapat memilih menu dengan menyentuh button melalui virtual hand.</p>
<p>Scene 04</p> 	<p>PENGENALAN BANGUN RUANG</p> <p>Animasi : Bangun ruang terbuka, Bangun ruang tertutup, Bangun ruang berputar</p> <p>Gestur : <i>Key Tap, Swipe (left/right), Push, Pull, Close Fist</i></p> <p>Fitur : <i>Zoom out, Zoom in, Rotate, Translate</i></p> <p>User dapat melakukan kontrol terhadap bangun ruang dengan menggunakan gestur</p>
<p>scene 05</p> 	<p>LATIHAN SOAL</p> <p>3 buah Sprite Soal 1 buah 3D Button</p> <p>Gestur : <i>Key Tap, Swipe</i></p> <p>Menunjukkan soal-soal yang berhubungan dengan bangun ruang terkait.</p>

<p>scene 06</p>  <p>Virtual Hands</p>	<p>MINI GAME 2</p> <p>1 Buah 3D button</p> <p>Gesture : Pinch, Release, Throw</p> <p>Fitur : Physics</p> <p>Pemain memasukkan bangun ruang yang benar ke dalam kotak. Jika benar maka akan mendapat score.</p>
<p>Scene 07</p> <p>Halaman Tentang</p>  <p>Virtual Hands</p>	<p>TENTANG (CREDIT)</p> <p>Menampilkan Informasi tentang aplikasi.</p> <p>1 buah 3D Button 1 buah sprite Tentang</p> <p>Gestur : <i>Key Tap</i></p> <p>User dapat melihat informasi aplikasi dan dapat kembali ke menu utama.</p>
<p>Scene 08</p> <p>Halaman Tentang</p>  <p>Virtual Hands</p>	<p>BANTUAN</p> <p>Menampilkan bantuan penggunaan aplikasi.</p> <p>1 buah 3D Button 1 buah sprite Tentang</p> <p>Gestur : <i>Key Tap</i></p> <p>User dapat melihat info bantuan dan dapat kembali ke menu utama.</p>

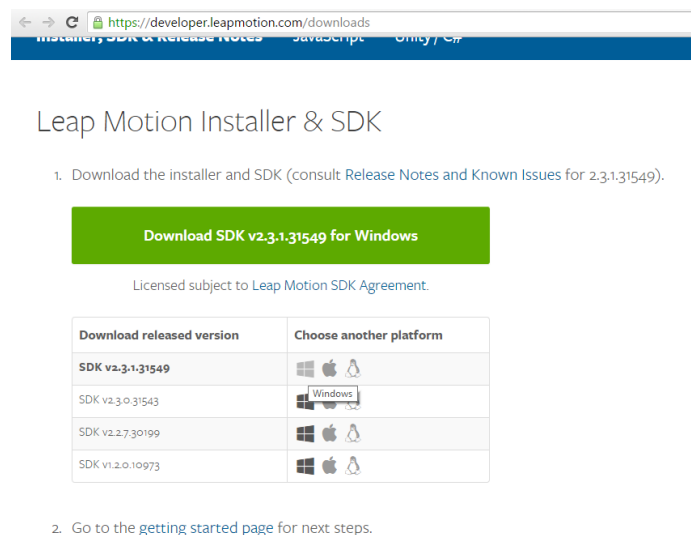
3.6 Tahap Produksi

3.6.1 Instalasi *Leap Motion Software Development Kit (SDK)*

Leap Motion Software Development Kit (SDK) diperlukan bagi pengembang aplikasi *Leap Motion* untuk mengetahui kinerja *Leap Motion* serta pustaka-pustaka (*libraries*) kode yang diperlukan. Pada penulisan ini penulis menggunakan SDK Versi 2.3.1+31549. SDK juga dapat digunakan oleh pengembang untuk menganalisa sensor *Leap Motion* dengan berbagai tools dan viewer yang ada di dalamnya.

Untuk mendownload SDK terbaru, terlebih dahulu login pada situs resmi *Leap Motion* untuk mendownload SDK. Jika belum memiliki ID, maka diperlukan registrasi yang dapat dilakukan secara gratis.

Jika sudah berhasil login dengan akun yang terdaftar sebelumnya, maka SDK dapat didownload di halaman <https://developer.leapmotion.com/downloads>



Gambar 3.13 Halaman Download *Leap Motion* SDK

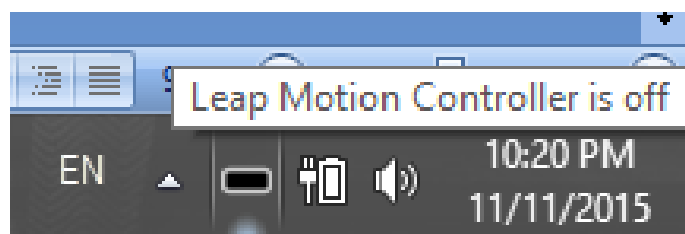
Terdapat 3 versi platform OS yang tersedia yaitu Windows, Mac, dan Linux. Pada penelitian ini penulis menggunakan versi windows.

Setelah diinstal, akan ada dua buah shortcut program pada desktop yaitu *Leap Motion App Home* dan *Leap Motion Control Panel*.



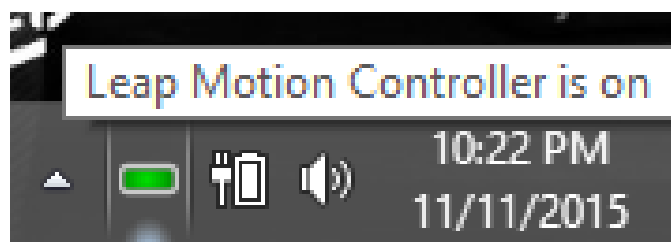
Gambar 3.14 Program SDK yang telah diinstall

Untuk memulai perancangan aplikasi, perlu diketahui terlebih dahulu apakah *controller device* berjalan dengan baik atau tidak. Jalankan *Leap Motion Control Panel* maka ikon *Leap Motion* akan muncul di pojok kanan taskbar. Jika status pada ikon belum menyala hijau artinya fungsi tracking pada *Leap Motion* belum berjalan.



Gambar 3.15 Status *tracking* pada posisi *off*

Untuk menjalankan fungsi tracking maka klik kanan pada icon, lalu pilih *resume tracking*.

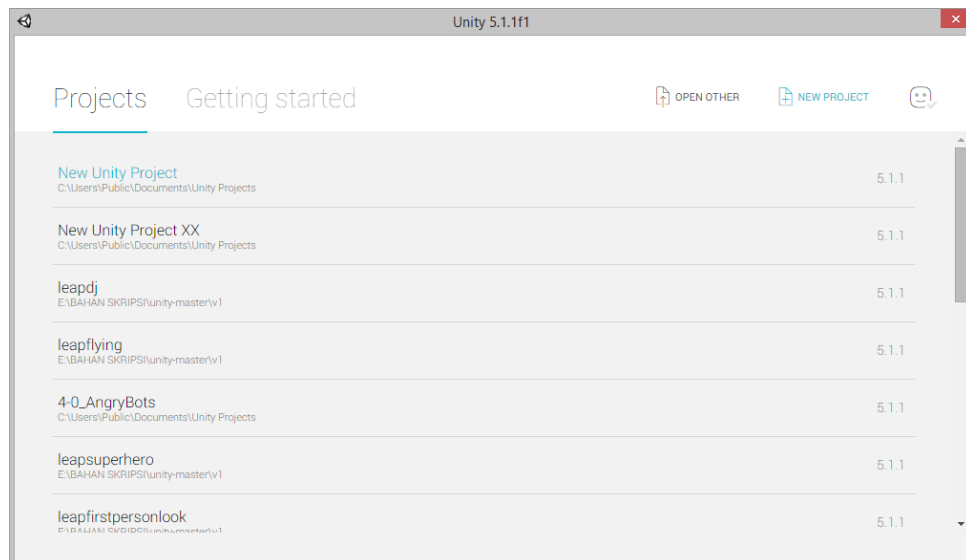


Gambar 3.16 Status *tracking* pada posisi *on*

Buka Visualizer (VisualizerApp.exe) dengan klik kanan pada icon, lalu pilih Visualizer. Maka, jika *Leap Motion* berhasil terkoneksi dengan baik, tracking gerakan tangan secara real-time akan terlihat pada layar.

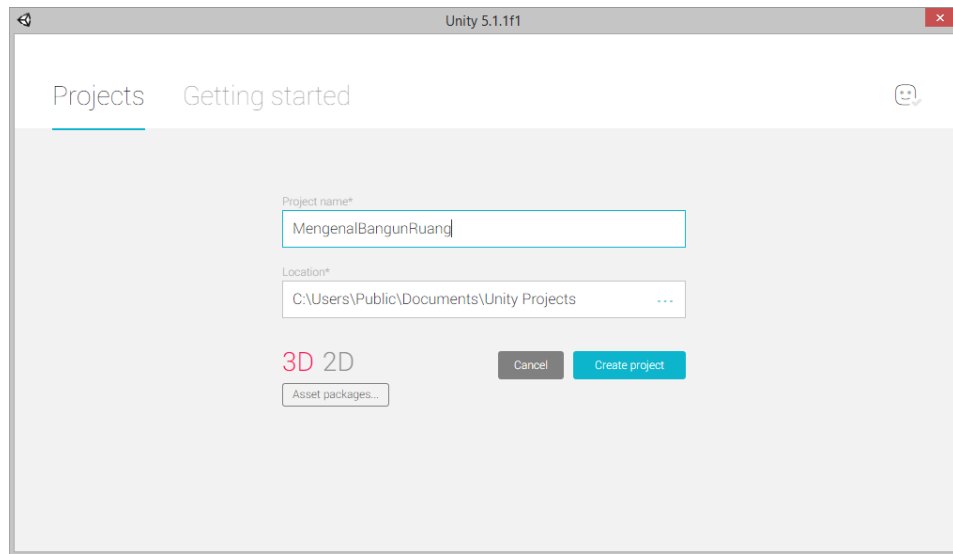
3.6.2 Membuat Project Baru

Setelah aplikasi game engine Unity dibuka, maka akan muncul jendela berisi project-project yang terakhir dibuka. Selain itu terdapat menu Open Other untuk membuka project lainnya dan menu New Project untuk membuat project baru. Pilih menu New Project.



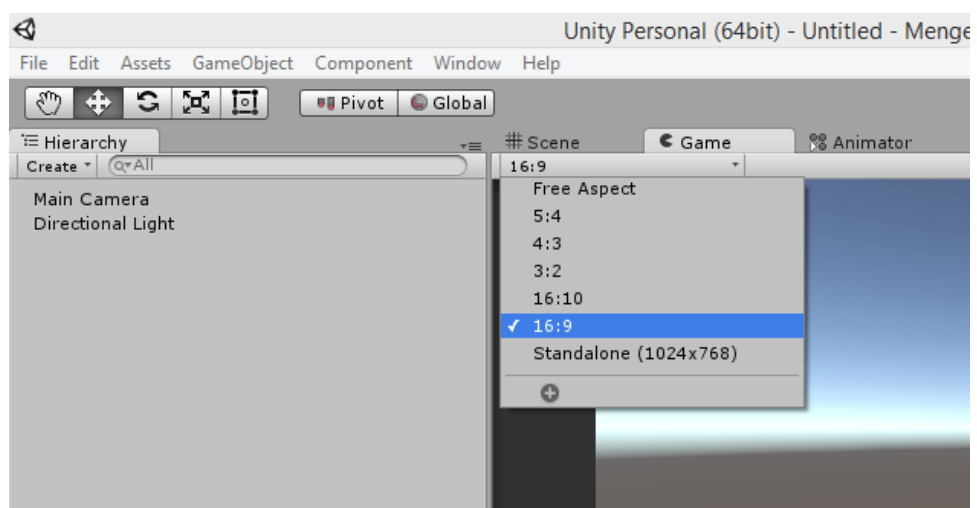
Gambar 3.17 Jendela Awal New Project

Setelah itu, project diberi nama, dan dipastikan bahwa asset package bertipe 3D. Lokasi penyimpanan project juga ditentukan. Dalam hal ini penulis menyimpan project Unity pada folder path C:\Users\Public\Documents\Unity Projects .



Gambar 3.18 Penentuan Lokasi Penyimpanan Project

Maka secara default, Unity akan membuatkan sebuah scene baru yang dapat langsung dipergunakan untuk merancang aplikasi. Scene berarti satu tampilan layar yang berisi sekumpulan obyek game yang aktif. Scene dapat digunakan untuk membuat menu, level, dan lain-lain. Untuk mendapatkan resolusi yang baik pada layar bertipe widescreen, tentukan rasio layar pada game dengan pengaturan 16:9.

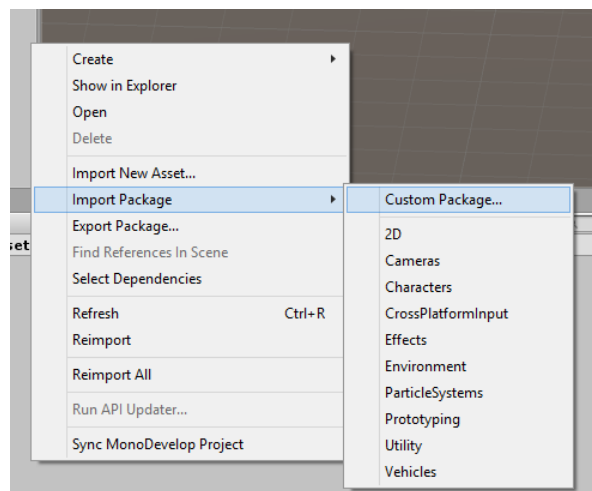


Gambar 3.19 Pengaturan Rasio Layar Aplikasi

3.6.3 Import *Leap Motion* Package

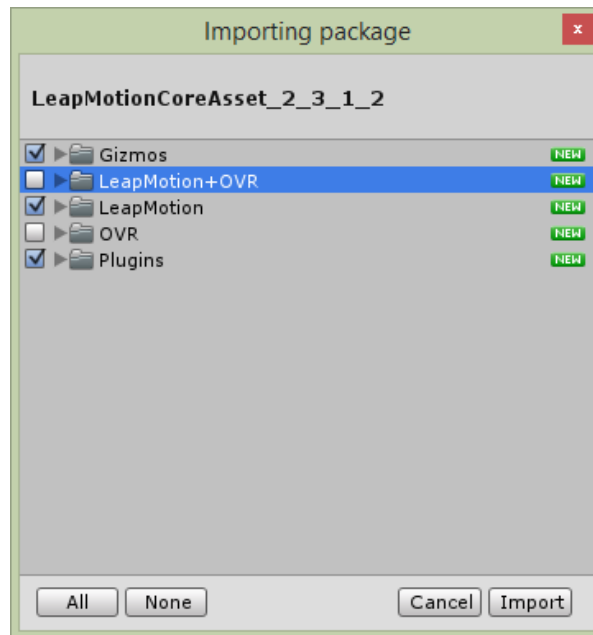
Paket Unity yang diperlukan dalam perancangan aplikasi Pengenalan Bangun Ruang dengan *Leap Motion Controller* adalah paket resmi dari *Leap Motion* yaitu LeapMotionCoreAsset versi 2.312.

Untuk menerapkan Custom Package, klik kanan pada project view lalu Import Package → Custom Package seperti yang tertera pada gambar berikut ;



Gambar 3.20 Proses Import Custom Package

Pada pilihan pada saat import package, paket OVR tidak perlu dicentang karena OVR adalah plugin untuk perangkat Oculus Virtual Reality.

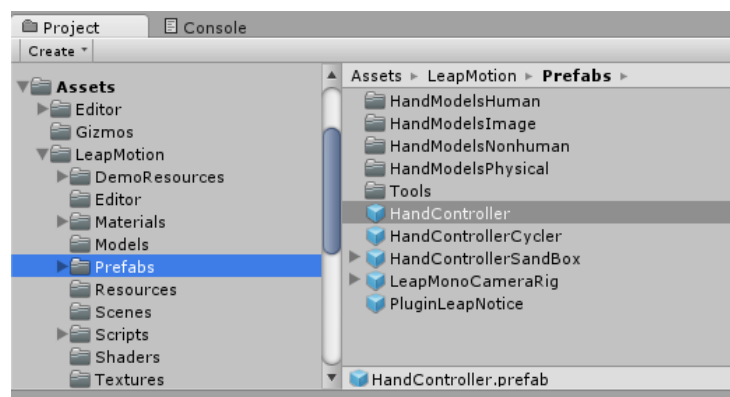


Gambar 3.21 Import *Leap Motion* Asset Package

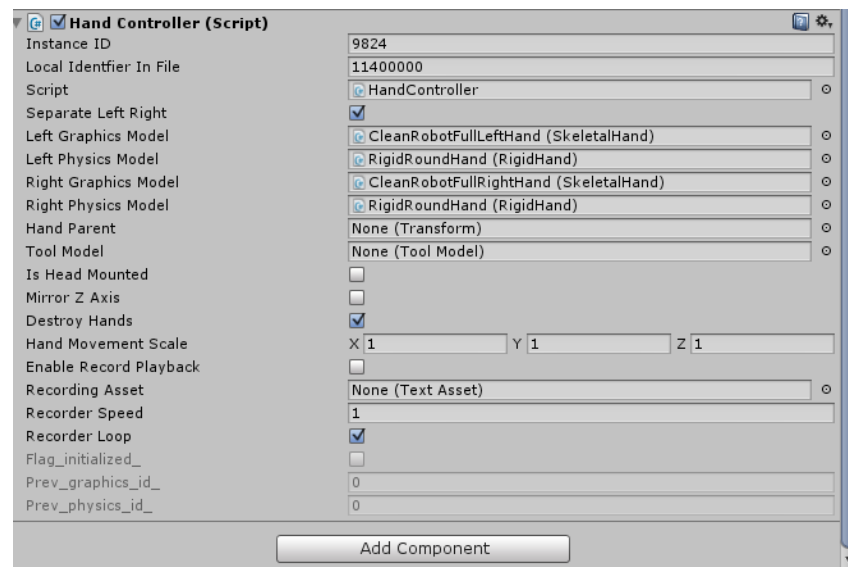
3.6.4 Set Scene pada Unity 5

Setelah package dari *Leap Motion* diimport, objek-objek pada scene disusun berdasarkan spesifikasi desain yang telah dirancang pada storyboard.

Komponen utama adalah *virtual hands* yang merupakan UI controller berbentuk tangan manusia. Untuk mengimplementasikan virtual hands, paket *Leap Motion* telah menyediakan prefab *HandController*.

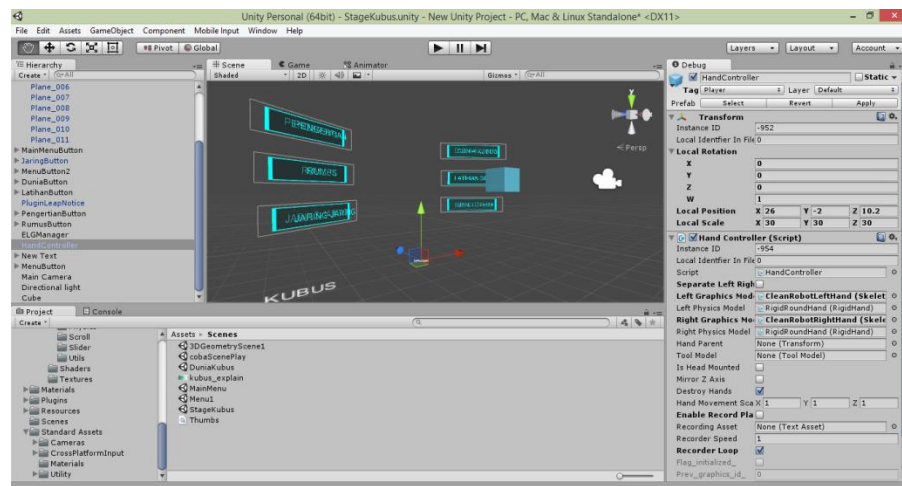


Gambar 3.22 Prefab Hand Controller



Gambar 3.23 Variabel-variabel Hand Controller

Untuk menambahkan virtual button, drag kedalam workspace prefab virtual button yang terletak pada folder path Assets\LeapMotion\Widgets\Prefab dengan nama prefab DemoToggleButton. Setelah itu, button diatur pada posisi di atas handController dan disusun berdasarkan rancangan.



Gambar 3.24 Objek disusun sesuai Storyboard

Struktur objek pada sebuah virtual button.

<Objek>Button

-- Button

---- Canvas

----- OnGraphics

----- Button-Square-Text-ON

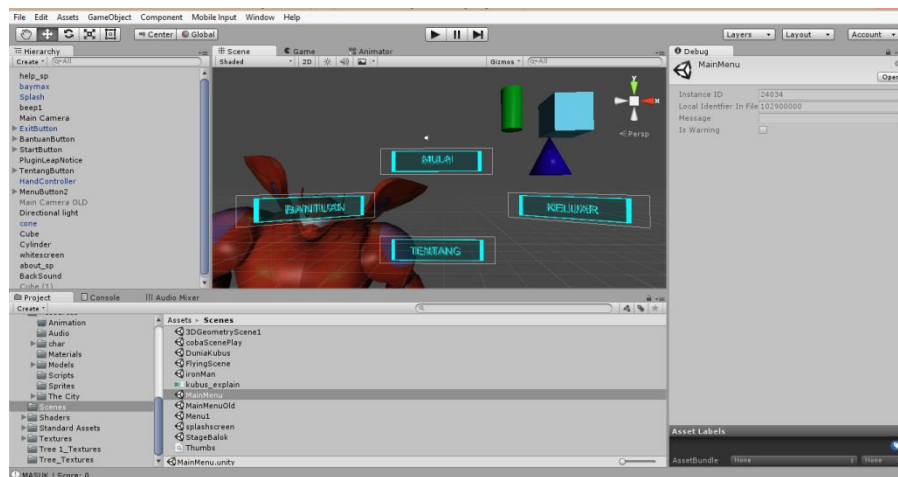
----- OffGraphics

----- Button-Square-Text-OFF

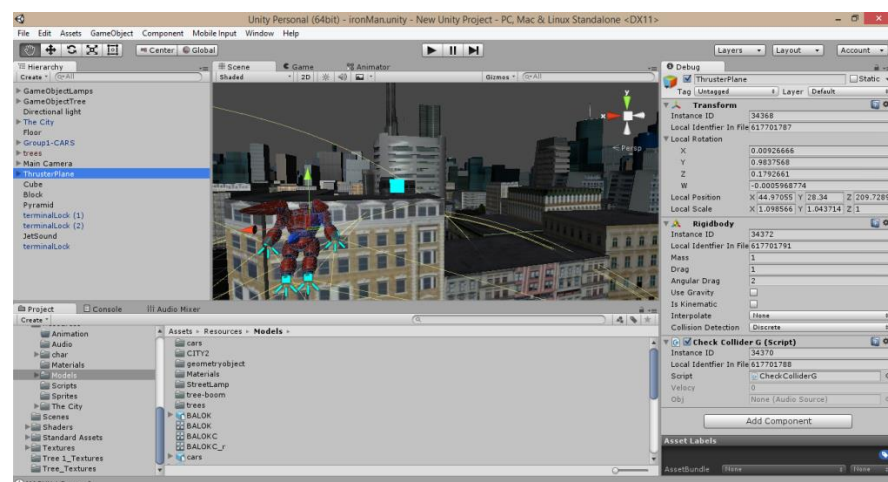
----- BotGraphics

----- Button_Panel_Bottom

----- Button_Plane_Bottom



Gambar 3.25 Susunan Button pada Main Menu

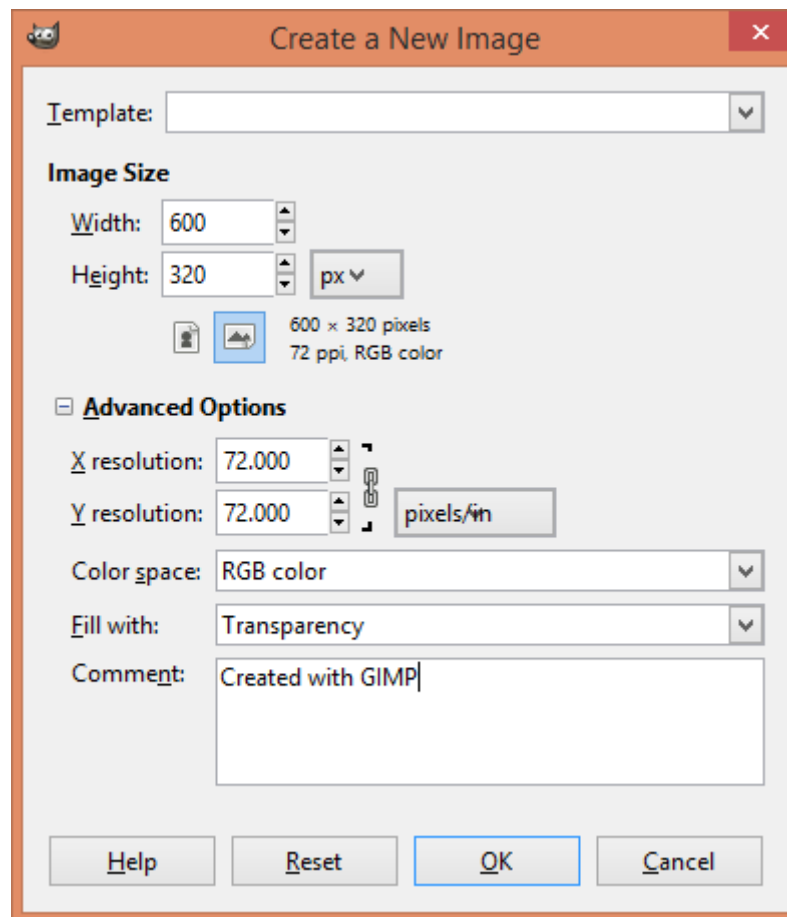


Gambar 3.26 Susunan Desain Robot dan Kota

3.5.5 Perancangan Sprite

Sprite adalah objek 2D atau sebuah image yang dapat dimanipulasi kedalam ruang 3D. Pada tugas akhir ini, penulis merancang sprite dengan menggunakan GIMP versi 2.8. Sprite yang dirancang adalah image untuk kebutuhan penyajian materi bangun ruang.

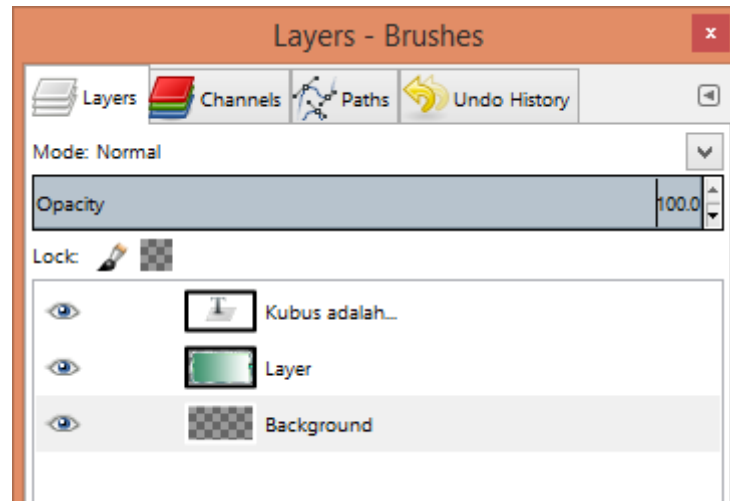
Setelah GIMP dibuka, klik menu File → New. Pengaturan awal project dibuat seperti yang ada pada gambar berikut ;



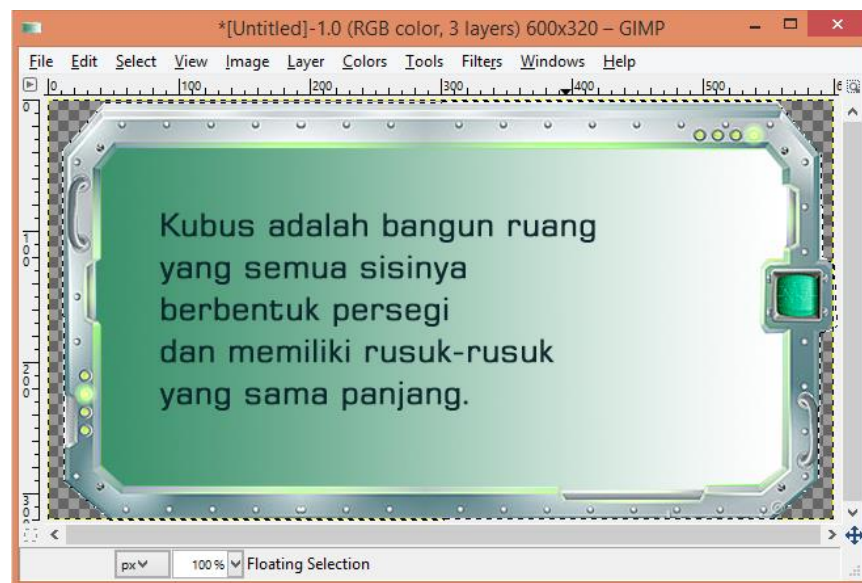
Gambar 3.27 Pengaturan Image pada GIMP

Layer 2 ditambahkan dengan cara menekan tombol Ctrl+Shift+n. Pada Layer 2 inilah background disisipkan. Sedangkan untuk

menambahkan text, pilih toolbox Text lalu ketikan materi yang akan disajikan. Maka akan dihasilkan susunan layer seperti berikut ;



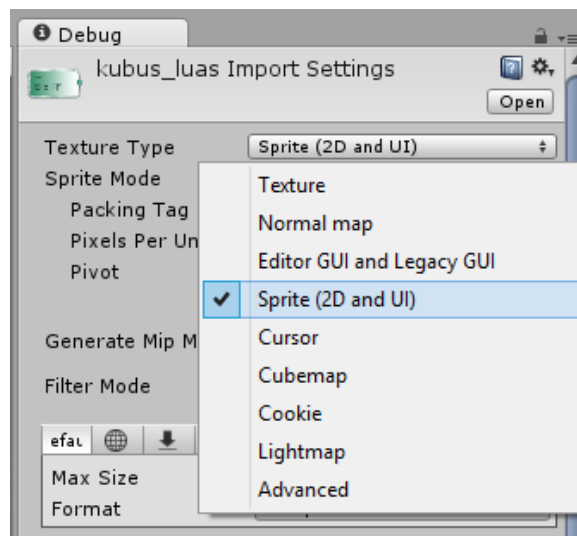
Gambar 3.28 Susunan Layer pada GIMP



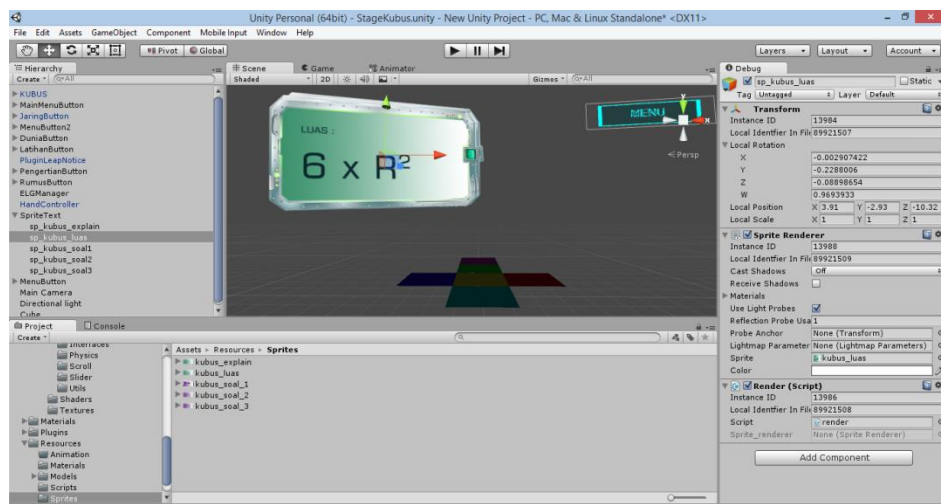
Gambar 3.29 Perancangan Sprite pada GIMP

Setelah selesai, hasil disimpan dengan format .png di dalam folder Asset/Resource/Sprite.

Jika sebuah image ingin dijadikan sprite, maka Unity perlu mengetahui tipe sebuah image. Untuk mengubah tipe image menjadi sprite maka pada panel debug, Texture Type diubah menjadi Sprite (2D and UI)



Gambar 3.30 Mengubah Tipe Image

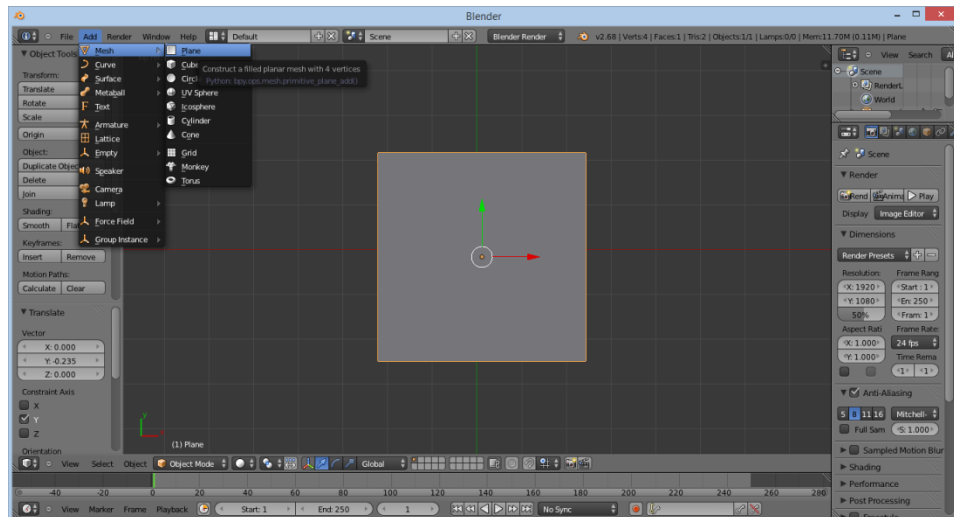


Gambar 3.31 Penyusunan Sprite pada Scene

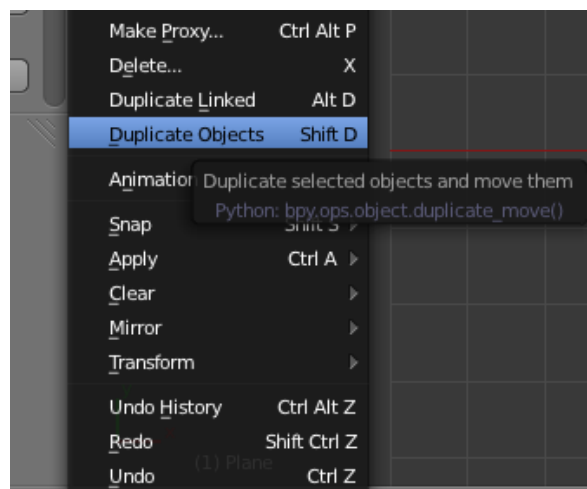
3.5.6 Pemodelan Objek 3D dan Animasi

Pemodelan dan animasi objek bangun ruang dibuat dengan menggunakan blender 2.68a. Kelebihan Blender adalah dapat langsung diintegrasikan dengan Unity 5 tanpa harus melakukan ekspor ke format 3D tertentu. Dengan kata lain, Unity 5 dapat melakukan import format default blender yaitu format *.blend*.

Buka blender. Untuk menambahkan bangun datar, pilih menu add → Mesh→ Plane.

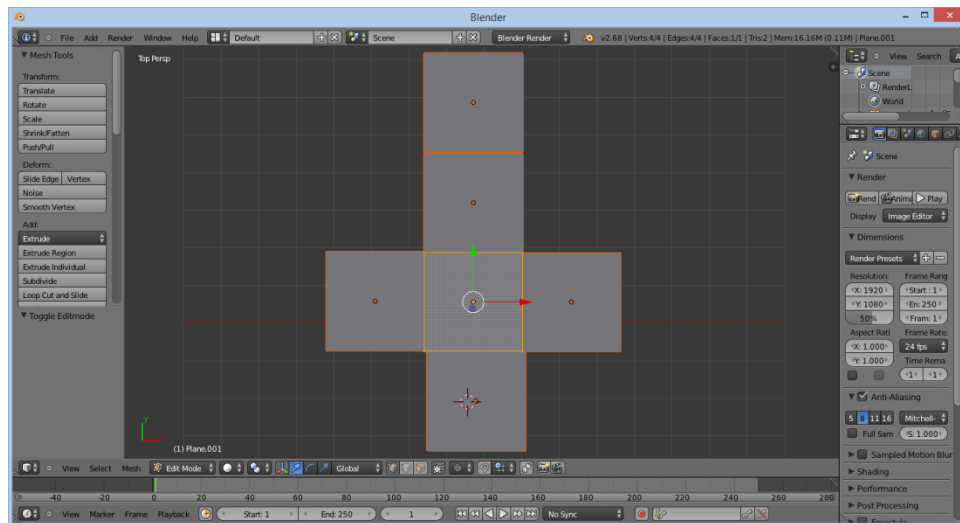


Gambar 3.32 Menambahkan objek Plane



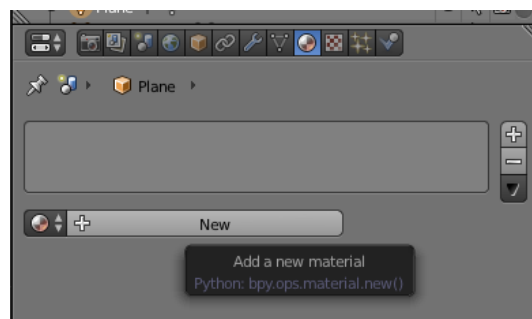
Gambar 3.33 Tahap Duplikasi objek

Objek persegi (plane) diduplikat dan dibentuk hingga membentuk jaring-jaring bangun tertentu. Dalam hal ini jaring-jaring kubus.



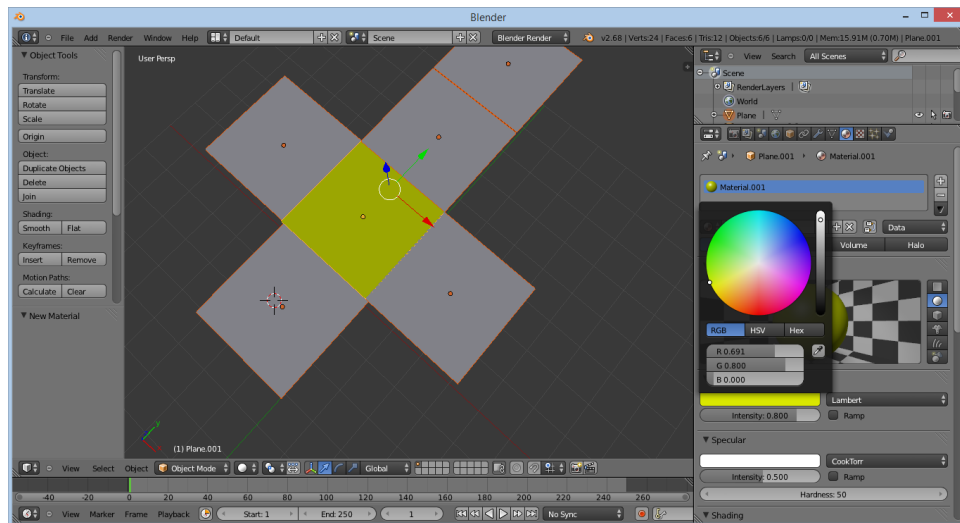
Gambar 3.34 Desain Jaring-jaring Kubus

Setelah jaring-jaring bangun ruang terbentuk, maka dilakukan penambahan material untuk memberikan warna pada setiap bidang mesh. Pada Material, pilih tombol.New.



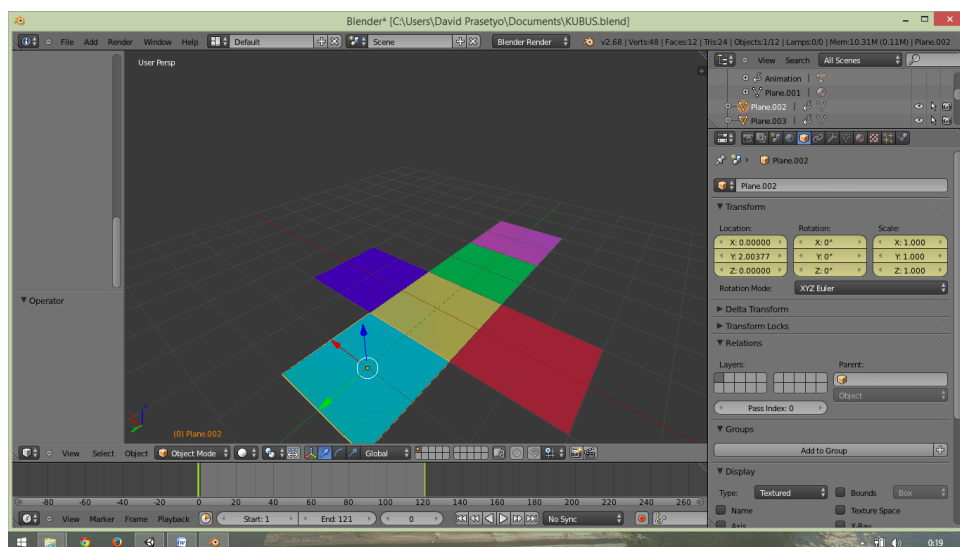
Gambar 3.35 Penambahan Texture

Pada bagian ini , dapat ditentukan warna material yang diinginkan melalui color picker. Setiap surface dapat diberikan warna atau material texture yang berbeda-beda.



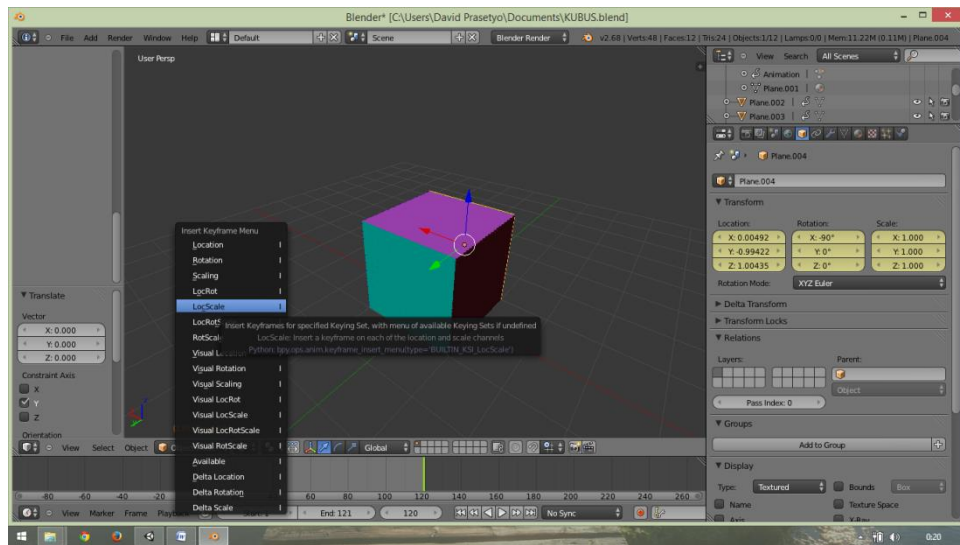
Gambar 3.36 Proses Pewarnaan Texture

Setelah semua bidang diwarnai maka hasilnya seperti gambar berikut



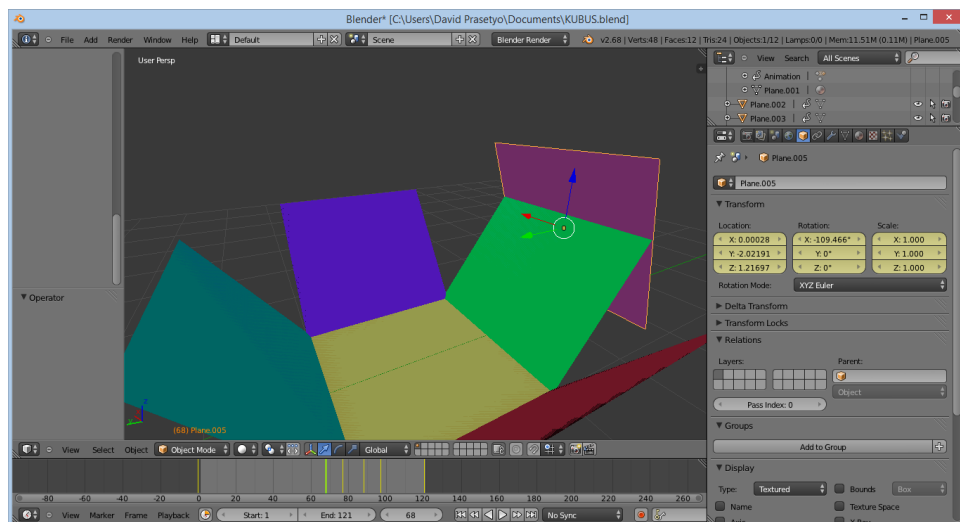
Gambar 3.37 Objek bangun Ruang Bertekstur

Pada proses penganimasian objek, penulis melakukan penentuan titik-titik frame untuk meletakkan keyframe. Pada kasus berikut ditentukan banyak frame dalam satu durasi animasi adalah 120 frame.



Gambar 3.38 Proses penambahan keyframe

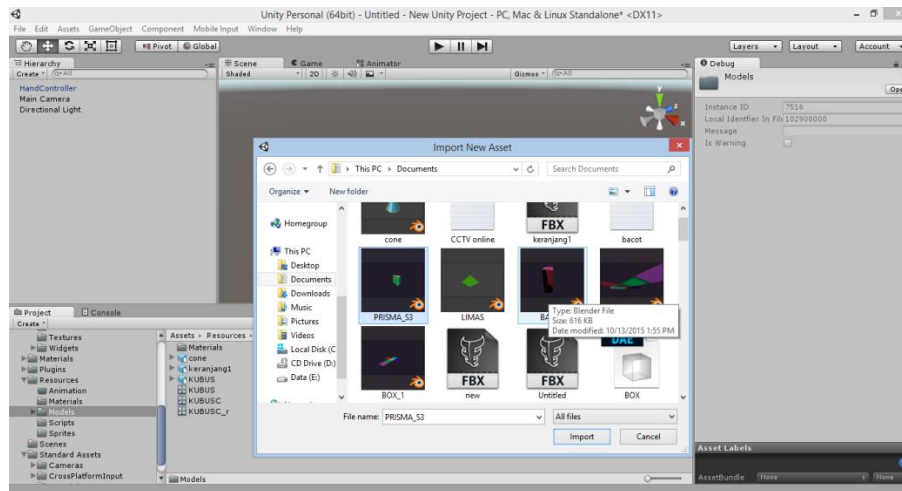
Pada frame ke-x posisi direkam dalam insert keyframe. Mode keyframe adalah locrotscale (location, rotation, scale).



Gambar 3.39 Animasi Bangun Ruang

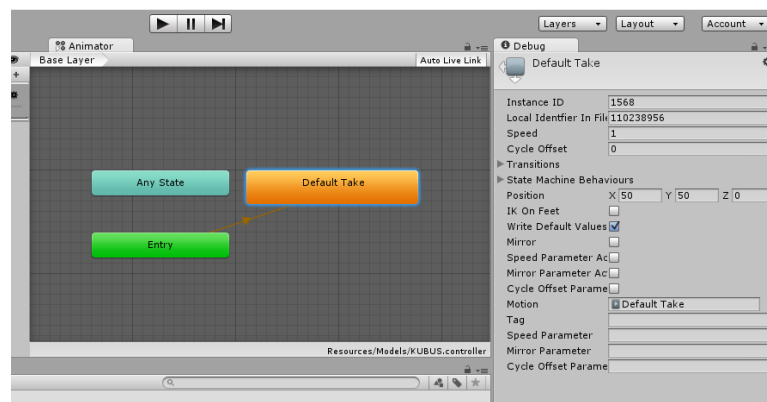
3.5.7 Proses Import Model ke Unity

Pada folder path ../Asset/Resource/Models import, klik kanan lalu Import New Asset. Pilih model yang telah dibuat pada blender dengan format *.blend* untuk diimport kedalam workspace Unity.



Gambar 3.40 Proses Import Blender Model ke Unity

Setelah animasi diekstrak, pada tab animator State Default Take nilai speed diisi 1 untuk menghasilkan gerakan animasi aslinya yaitu gerak menutup dari jaring-jaring menjadi bangun ruang. Setelah itu animasi diduplikat dan diberi nilai speed -1 untuk menghasilkan gerakan terbalik yaitu bangun ruang menjadi jaring-jaring.

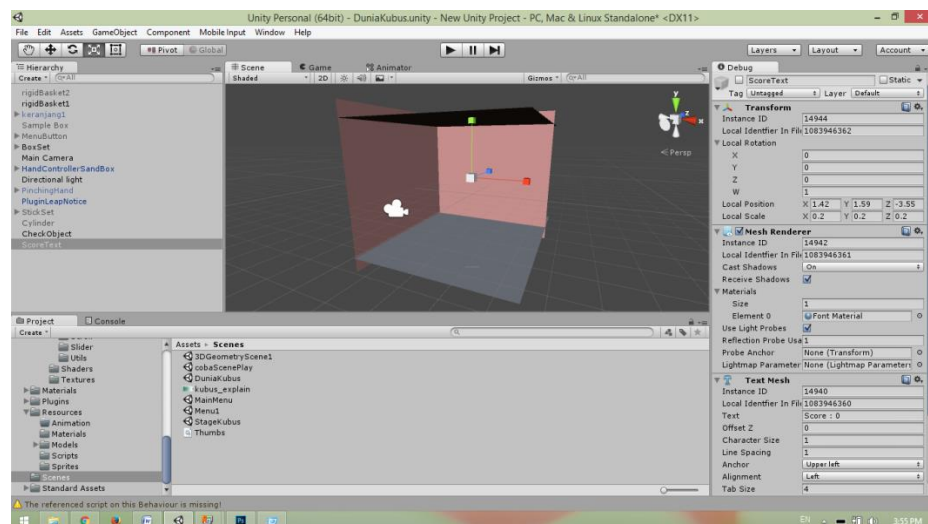


Gambar 3.41 Animator pada Unity

3.5.8 Perancangan Scene Mini Game

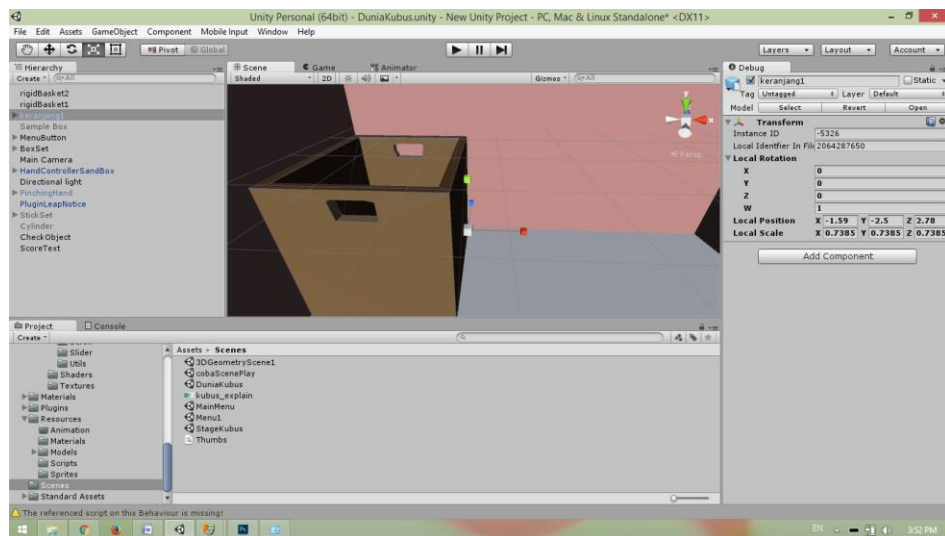
Scene baru dibuat dengan cara pilih File → New Scene. Untuk mencegah scene tidak tersimpan, maka dapat langsung dilakukan penyimpanan scene dengan memilih menu File → Save Scene As. Simpan pada path folder ../Assets/Resource/Scene.

Arena permainan dibuat terlebih dahulu untuk membatasi pemain dan objek-objek supaya tidak keluar dari arena. Yang dibuat dari 6 buah objek persegi (plane) dengan penambahan beberapa *component* yaitu Mesh collider, Mesh Renderer dan Box Collider. Objek Hand Controller juga diimport untuk menampilkan fitur virtual hand.



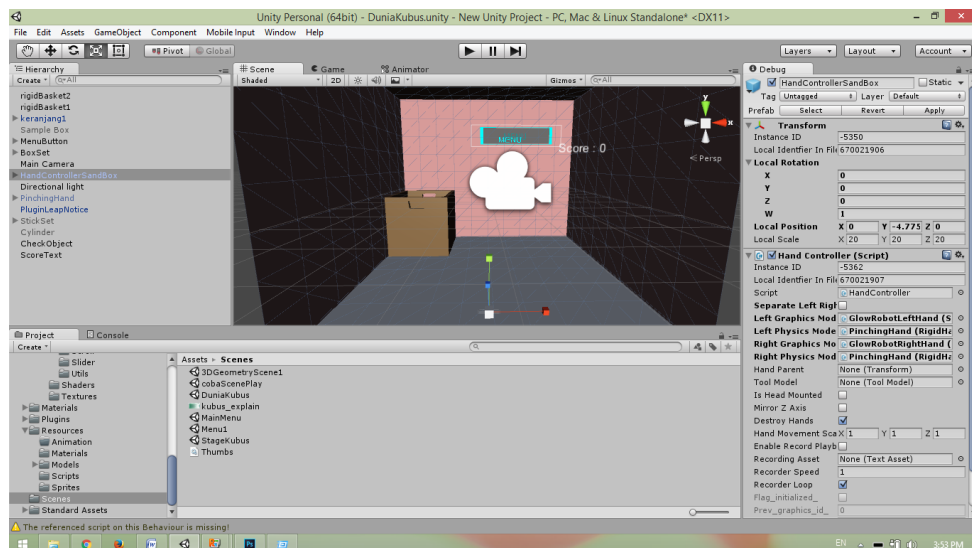
Gambar 3.42 Set Arena Mini Game

Setelah arena siap, penulis menambahkan objek kardus atau keranjang untuk meletakkan objek bangun ruang yang benar. Objek untuk karakter robot dan desain kota juga didownload. Objek didapatkan secara gratis dan bebas lisensi dengan mendownloadnya di <http://3dwarehouse.sketchup.com>



Gambar 3.43 Penambahan Objek Keranjang

Untuk membuat GUI seperti menu dapat menggunakan GUIText dan GUITexture. GUITexture adalah objek 2D yang menampilkan sebuah texture 2D pada layar. Texture ini dapat digunakan untuk menampilkan logo, button, virtual joystick, dan sebagainya.



Gambar 3.44 Penambahan Objek 3D Button

Langkah selanjutnya adalah mengimport objek-objek 3D lain yang berhubungan dengan bangun ruang yang di sekitar kita kedalam scene. Seperti objek dadu, rubik, cone, bola, kayu dan lain-lain. Nantinya ditambahkan script *grabable objet* supaya objek dapat dipilih dan diangkat oleh *virtual hands*. Tambahkan juga property rigidbody pada setiap objek supaya fisik objek dapat terdeteksi.

3.5.9 Implementasi Gesture Recognition

Pengenalan gestur tangan (*hand gesture recognition*) pada aplikasi dibuat menjadi 10 variasi tipe gestur yaitu; *Tap, Swipe, Clap, Double Outward Swipe, Circular, Close Fist, Pinch, Release, Push dan Pull*. Tipe-tipe gestur tersebut dijabarkan sebagai berikut ;

1. Swipe Gesture

Seperti dijabarkan oleh Pedro Miguel dalam jurnal berjudul *Analysis and Evaluation of Gesture Recognition using LeapMotion* ^[16] , Gestur swipe dapat dikenal dengan mendefinisikan gerakan tangan secara vertikal maupun horisontal menggunakan jari telunjuk maupun telapak tangan.

$$H = |D_x| > |D_y| \quad (3.1)$$

maka implementasi kodenya adalah sebagai berikut:

```

if(gestype == "gTYPESWIPE") {
    switch(tipe){
        case 1 :
            if(gesture.Position.x < 0){
                spinLeft = true ;
                spinRight = false ;
            }
            if(gesture.Position.x > 0){
                spinLeft = false ;
                spinRight = true ;
            }
            break;
    }
}

```

2. Key / Button Tap Gesture

```
private void CheckTrigger()
{
    float scale = transform.lossyScale.z;
    m_localTriggerDistance = triggerDistance / scale;
    m_localCushionThickness = Mathf.Clamp(cushionThickness / scale, 0.0f,
    m_localTriggerDistance - 0.001f);
    if (m_isPressed == false)
    {
        if (transform.localPosition.z > m_localTriggerDistance)
        {
            m_isPressed = true;
            buttonPressed();
        }
    }
    else if (m_isPressed == true)
    {
        if (transform.localPosition.z < (m_localTriggerDistance - m_localCushionThickness))
        {
            m_isPressed = false;
            buttonReleased();
        }
    }
}
```

3. Clap Gesture

```
if(clapGestureRegistered) {
    if(mFrame.Hands.Count == 2) {
        bool leftHandSwipeIn = (!PalmIsHorizontal(mFrame.Hands.Leftmost)) && mFrame.
            Hands.Leftmost.PalmVelocity.x > EasyLeapGesture.MinClapVelocity;
        bool rightHandSwipeIn = (!PalmIsHorizontal(mFrame.Hands.Rightmost)) && mFrame.
            Hands.Rightmost.PalmVelocity.x < -EasyLeapGesture.MinClapVelocity;
        if(leftHandSwipeIn && rightHandSwipeIn) {
            if(mFrame.Hands[0].StabilizedPalmPosition.DistanceTo(mFrame.Hands[1].
                StabilizedPalmPosition) < EasyLeapGesture.MaxPalmClapDistance)
                ClapRecognised(EasyLeapGestureState.STATESTART);
            } else ClapRecognised(EasyLeapGestureState.STATEINVALID);
        }
    }
}
```

4. Double Outward Swipe Gesture

```
if(doubleOutwardsSwipeGestureRegistered && leftHandSwipeOut &&
    rightHandSwipeOut) {
    if(mFrame.Hands[0].StabilizedPalmPosition.DistanceTo(mFrame.Hands[1].
        StabilizedPalmPosition) > EasyLeapGesture.MaxPalmDistance)
        DoubleOutwardsSwipeRecognised(EasyLeapGestureState.STATESTART);
    } else DoubleOutwardsSwipeRecognised(EasyLeapGestureState.STATEINVALID);
}
```

5. Push Gesture

```
if(pushGestureRegistered) {
    if(mFrame.Hands[0].PalmVelocity.y < -EasyLeapGesture.MinPushPullVelocity)
        PushGestureRecognised(EasyLeapGestureState.STATESTART);
    else PushGestureRecognised(EasyLeapGestureState.STATEINVALID);
}
```

6. Pull Gesture

```

if(pullGestureRegistered) {
    if(mFrame.Hands[0].PalmVelocity.y > EasyLeapGesture.MinPushPullVelocity)
        PullGestureRecognised(EasyLeapGestureState.STATESTART);
    else PullGestureRecognised(EasyLeapGestureState.STATEINVALID);
}

private void PullGestureRecognised(EasyLeapGestureState state) {
    if(state == EasyLeapGestureState.STATEINVALID || Time.time <
        pullRestTime + EasyLeapGesture.PullRecoveryTime) {
        gestureList.Remove(-(int)EasyLeapGestureType.PULL);
        return;
    }
    pullRestTime = Time.time;
    RecordNewGesture(-(int)EasyLeapGestureType.PULL, EasyLeapGestureState.
        STATESTART, EasyLeapGestureState.STATEUPDATE, EasyLeapGestureType.
        PULL, -1, mFrame.Hands[0].StabilizedPalmPosition);
}

```

7. Circular Gesture

```

String clockwiseness;
if (circle.Pointable.Direction.AngleTo (circle.Normal) <= Math.PI / 2) {
    clockwiseness = "clockwise";
} else {
    clockwiseness = "counterclockwise";
}

```

8. Fly Control

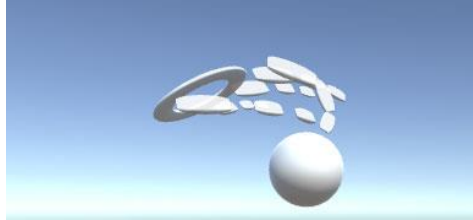
```

void FixedUpdate () {
    Frame frame = m_leapController.Frame();
    if (frame.Hands.Count >= 2) {
        Hand leftHand = GetLeftMostHand(frame);
        Hand rightHand = GetRightMostHand(frame);
        Vector3 avgPalmForward = (frame.Hands[0].Direction.ToUnity() + frame.Hands[1].Direction.ToUnity()) * 0.5f;
        Vector3 handDiff = leftHand.PalmPosition.ToUnityScaled() - rightHand.PalmPosition.ToUnityScaled();
        Vector3 newRot = transform.parent.localRotation.eulerAngles;
        newRot.z = -handDiff.y * 20.0f;
        newRot.y += handDiff.z * 3.0f - newRot.z * 0.03f * transform.parent.GetComponent<Rigidbody>().velocity.magnitude;
        newRot.x = -(avgPalmForward.y - 0.1f) * 100.0f;
        if(forceMult < -3){
            forceMult = -4.0f;
            StartCoroutine (MyCoroutine ());
        }else{
            forceMult = 10.0f;
        }
        if (frame.Fingers.Count < 3) {
            forceMult = -3.0f;
        }
        transform.parent.localRotation = Quaternion.Slerp(transform.parent.localRotation, Quaternion.Euler(newRot), 0.1f);
        transform.parent.GetComponent<Rigidbody>().velocity = transform.parent.forward * forceMult;
    }
}

```

9. Pinch Gesture :

Hansaem Lee, Junseok Park, pada jurnal berjudul *Hand Gesture Recognition in Multi-space of 2D/3D* ^[24] .



Gambar 3.45 Pinch Gesture

Sumber: *Leap Motion Documentation*

Metode pengenalan gestur pinch pada bidang 2D. Definisi pinch dan release atau gerakan mencubit (mencapit) dan melepaskannya secara 2 Dimensi dijabarkan dengan algoritma sebagai berikut:

- (1) titik sentuhan harus berada di sekeliling objek interaktif.
- (2) jarak setiap ujung jari yang menjepit objek harus lebih kecil daripada nilai threshold. Nilai threshold adalah jarak yang ditentukan dari lebar objek interaktif.

Data jari membutuhkan koordinat dari setiap jari yang pertamakali menjangkau bidang sentuh ($finger_1_{x_in}$, $finger_1_{y_in}$), ($finger_2_{x_in}$, $finger_2_{y_in}$), \dots , ($finger_n_{x_in}$, $finger_n_{y_in}$),. maka nilai maksimum dan minimum jarak jari dijabarkan sebagai berikut:

$$\begin{aligned}
 max_x &= max(finger_1_{xin}, finger_1_{xout}, \dots, finger_n_{xin}, finger_n_{xout}) \\
 min_x &= min(finger_1_{xin}, finger_1_{xout}, \dots, finger_n_{xin}, finger_n_{xout}) \\
 max_y &= max(finger_1_{yin}, finger_1_{yout}, \dots, finger_n_{yin}, finger_n_{yout}) \\
 min_y &= min(finger_1_{yin}, finger_1_{yout}, \dots, finger_n_{yin}, finger_n_{yout})
 \end{aligned} \tag{3.2}$$

pada saat itu, saat titik tengah objek mereferensi ke titik interaksi / *interaction point* ($Object_x$, $Object_y$)

$$\begin{aligned}
 min_x &\leq Object_x \leq max_x \\
 min_y &\leq Object_y \leq max_y
 \end{aligned} \tag{3.3}$$

Saat formula (3.3) terpenuhi, maka kondisi (1) juga terpenuhi. Untuk memenuhi kondisi (2), nilai pergerakan jarak jari haruslah melebihi nilai treshold. Untuk menghitung nilai dari jarak dinamis maka digunakan metode Euclidean :

$$distance_{finger_n} = \sqrt{finger_n_{xin} + finger_n_{xout}} \quad (3.4)$$

Sehinga gestur 'pinch' dinyatakan berhasil jika jumlah dari jarak jari lebih kecil atau sama dengan nilai threshold :

$$distance_{finger_1} + \dots + distance_{finger_1} \leq Threshold \quad (3.5)$$

maka implementasi kodenya adalah sebagai berikut:

```
protected PinchState GetNewPinchState() {
    HandModel hand_model = GetComponent<HandModel>();
    Hand leap_hand = hand_model.GetLeapHand();
    Vector leap_thumb_tip = leap_hand.Fingers[0].TipPosition;
    float closest_distance = Mathf.Infinity;
    for (int i = 1; i < HandModel.NUM_FINGERS; ++i) {
        Finger finger = leap_hand.Fingers[i];

        for (int j = 0; j < FingerModel.NUM_BONES; ++j) {
            Vector leap_joint_position = finger.Bone((Bone.BoneType)j).NextJoint;

            float thumb_tip_distance = leap_joint_position.DistanceTo(leap_thumb_tip);
            closest_distance = Mathf.Min(closest_distance, thumb_tip_distance);
        }
    }
    float proximal_length = leap_hand.Fingers[0].Bone(Bone.BoneType.TYPE_PROXIMAL).Length;
    float trigger_distance = proximal_length * grabTriggerDistance;
    float release_distance = proximal_length * releaseTriggerDistance;
    if (closest_distance <= trigger_distance)
        return PinchState.kPinched;
    if (closest_distance <= release_distance && pinch_state_ != PinchState.kReleased &&
        !ObjectReleaseBreak(current_pinch_position_))
        return PinchState.kReleasing;
    return PinchState.kReleased;
}
```

10. Release Gesture

Sedangkan untuk gestur release (melepaskan) merupakan kebalikan operator kondisi dari gestur mencapit ;

$$\sum(RDistance) \geq Threshold_{distance} \quad (3.6)$$

$$\sum(angle) \geq Threshold_{angle}$$

maka implementasi kodenya adalah sebagai berikut ;

```
if (closest_distance <= release_distance && pinch_state_ != PinchState.kReleased &&
    !ObjectReleaseBreak(current_pinch_position_))
    return PinchState.kReleasing;
```

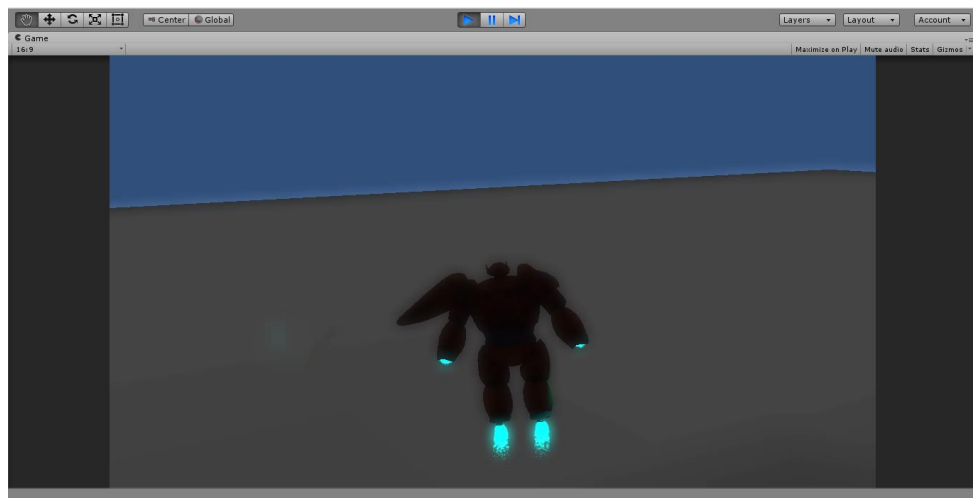
3.5.9 Hasil Kompilasi Versi Alpha & Beta

Pada hasil rilis versi alpha , prototip aplikasi telah berjalan baik. Interface aplikasi telah didesain sesuai dengan storyboard. Pada menu utama user dapat memilih menu bangun ruang.



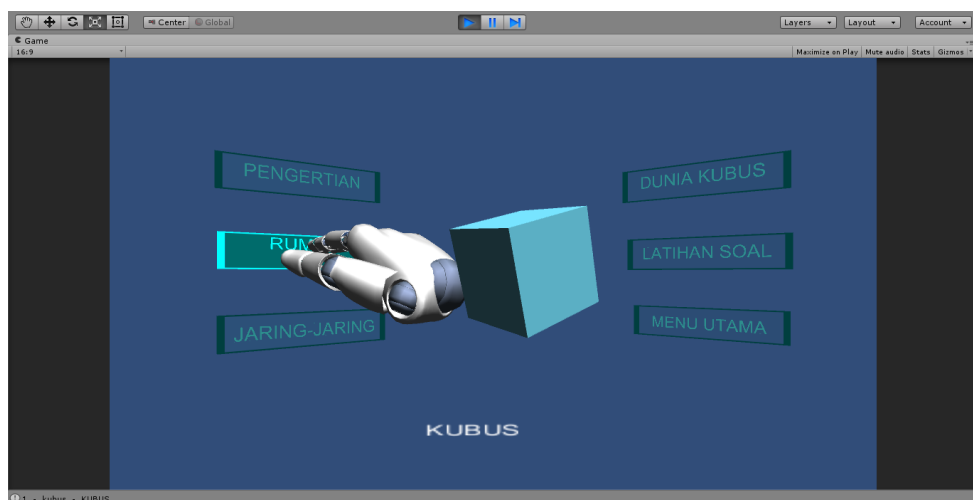
Gambar 3.46 Tampilan Menu Utama

Di dalam scene mini game 1 , karakter robot diberikan kemampuan untuk terbang dan dikendalikan dengan gestur *fly control* . Nantinya karakter ini akan mampu untuk menyusuri kota untuk mencari titik-titik adatu stasiun pengenalan bangun ruang



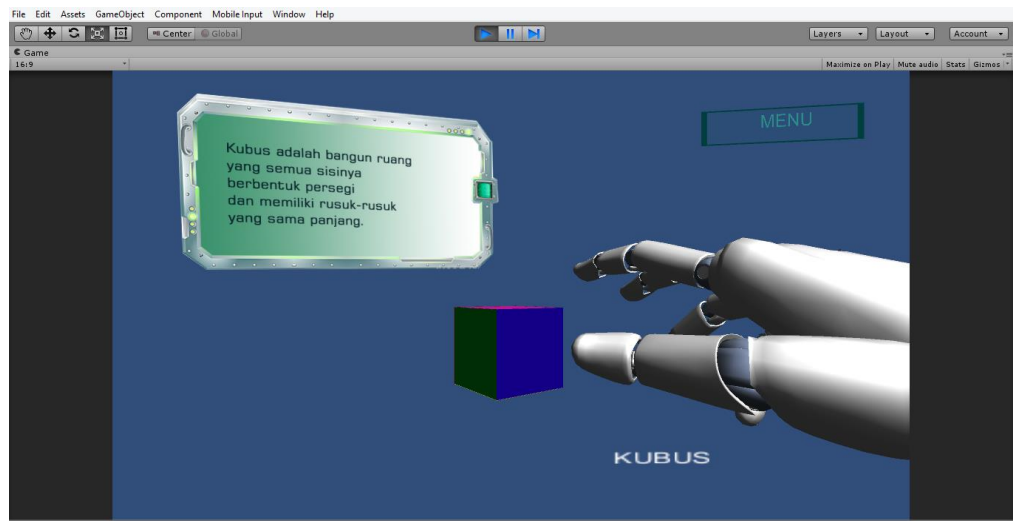
Gambar 4.47 Scene Mini Game 1

Sedangkan pada menu bangun ruang, user dapat memilih materi yang akan disajikan dengan menggunakan *key tap gesture*.



Gambar 3.48 Tampilan Menu Bangun Ruang (1)

Implementasi gestur *swipe*, *clap double outward swipe*, sudah berjalan dengan baik. Animasi bangun ruang terbuka atau tertutup telah diterapkan.



Gambar 3.49 Tampilan Materi Bangun Ruang (2)



Gambar 3.50 Tampilan Latihan Soal

Scene mini game pada gambar berikut sudah memasuki versi beta, user memilih bangun ruang yang sesuai dengan materi yang disajikan. Tugas user adalah memasukkannya ke dalam kotak. Jika benar, maka akan mendapat nilai atau *score*.



Gambar 3.51 Scene Mini Game 2