

**GAME SIMULASI KERJA PRAKTEK MAHASISWA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
MENGUNAKAN UNREAL ENGINE 4**

SKRIPSI

Karya Tulis sebagai syarat untuk memperoleh
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi
Universitas Bale Bandung

Disusun oleh:

AGGIA BINTANG RAMADHAN H
NIM. C1A150020



PROGRAM STRATA 1
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
BANDUNG

2019

LEMBAR PERSETUJUAN PEMBIMBING

**GAME SIMULASI KERJA PRAKTEK MAHASISWA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
MENGUNAKAN UNREAL ENGINE 4**

Disusun oleh:

**AGGIA BINTANG RAMADHAN H
NIM. C1A150020**

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Pembimbing 1

Pembimbing 2

Yudi Herdiana, S.T., M.T.
NIK. 04104808008

Rustiyana, S.T., M.T.
NIK. 04104808015

LEMBAR PERSETUJUAN PENGUJI

**GAME SIMULASI KERJA PRAKTEK MAHASISWA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
MENGUNAKAN UNREAL ENGINE 4**

Disusun oleh:

**AGGIA BINTANG RAMADHAN H
NIM. C1A150020**

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Penguji 1

Penguji 2

Denny Rusdianto, S.T., M.Kom.
NIK. 04104808094

Nurul Imamah, S.T., M.T.
NIK. 04104808121

LEMBAR PENGESAHAN PROGRAM STUDI

**GAME SIMULASI KERJA PRAKTEK MAHASISWA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
MENGUNAKAN UNREAL ENGINE 4**

Disusun oleh:

**AGGIA BINTANG RAMADHAN H
NIM. C1A150020**

SKRIPSI ini telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar
SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Mengetahui,
Dekan,

Mengesahkan,
Ketua Program Studi

Yudi Herdiana, S.T., M.T.
NIK. 04104808008

Yaya Suharya, S.Kom., M.T.
NIK. 01043170007

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : AGGIA BINTANG RAMADHAN HERMANTO

NIM : C1A150020

Judul Skripsi : **GAME SIMULASI KERJA PRAKTEK MAHASISWA
FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG
MENGUNAKAN UNREAL ENGINE 4**

Menyatakan dengan sebenarnya bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah laporan maupun kegiatan *programming* yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, saya mencantumkan sumber yang jelas. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG.

Demikian surat pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Baleendah, Agustus 2019

Yang membuat pernyataan,

AGGIA BINTANG
RAMADHAN HERMANTO

NIM. C1A150020

ABSTRACT

The development of computer video game applications is now very mushrooming. The emergence of a variety of software that aims to facilitate the development of video games on free computer devices makes computer video game developer studios more and more. Video games are electronic games that are played using a controller. One type of game that is popular today is the simulation genre game, where this genre simulates what is in the real world.

The research method used is to collect the data needed by observation, literature studies and interviews, then for the development method of this research using the Multimedia Development Life Cycle (MDLC) method in which this method has 6 stages including: concept (Consept), design (Design), collection of materials and data (Material Collecting), Compilation and Manufacturing (Assembly), testing (Testing) and widespread (Distribution).

The result of the video game that will be made is a simulation video game regarding the procedures of the Job Training course at the Information Technology Faculty of the University of Bale Bandung. Where students can play this simulation video game to see what are the procedures in this practical work course. to students and administrative staff at the Information Technology Faculty of Bale Bandung University.

Keywords : *Game, Video Game, Game Simulasi, Simulasi*

ABSTRAK

Perkembangan aplikasi *video game* komputer saat ini sudah sangat menjamur. Munculnya berbagai perangkat lunak yang bertujuan untuk memudahkan pengembangan *video game* pada perangkat komputer yang gratis membuat studio pengembang *video game* komputer semakin banyak. *Video game* merupakan permainan elektronik yang dimainkan dengan menggunakan *controller*. Salah satu jenis *game* yang digemari saat ini adalah *game ber-genre simulation*, dimana *genre* ini mensimulasikan apa yang ada di dunia nyata.

Metode penelitian yang dilakukan yaitu dengan mengumpulkan data yang dibutuhkan dengan cara observasi, studi literatur dan wawancara, selanjutnya untuk metode pengembangan penelitian ini menggunakan metode *Multimedia Development Life Cycle* (MDLC) yang dimana metode ini memiliki 6 tahapan meliputi : konsep (*Concept*), perancangan (*Design*), pengumpulan bahan dan data (*Material Collecting*), Penyusunan dan Pembuatan (*Assembly*), pengujian (*Testing*) dan menyerbar luaskan (*Distribution*).

Hasil dari *video game* yang akan dibuat adalah *video game* simulasi mengenai prosedur mata kuliah Kerja Praktek di Fakultas Teknologi Informasi Universitas Bale Bandung. Dimana mahasiswa dapat memainkan *video game* simulasi ini untuk melihat apa saja prosedur pada mata kuliah kerja praktek ini. kepada mahasiswa dan petugas tata usaha di Fakultas Teknologi Informasi Universitas Bale Bandung.

Kata Kunci : *Games, Video Games, Simulation Games, Simulation*

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah SWT yang telah memberikan penulis kemudahan sehingga penulis dapat menyelesaikan laporan ini dengan tepat waktu. Tanpa pertolongan-Nya tentunya penulis tidak akan sanggup untuk menyelesaikan laporan ini dengan baik. Shalawat serta salam semoga terlimpah curahkan kepada baginda tercinta yaitu Nabi Muhammad SAW yang kita nanti-nantikan syafa'atnya di akhirat nanti.

Penulis mengucapkan syukur kepada Allah SWT atas limpahan nikmat sehat-Nya, baik itu berupa sehat fisik maupun akal pikiran, sehingga penulis mampu untuk menyelesaikan pembuatan laporan skripsi guna memperoleh gelar Sarjana Komputer dengan judul **“Game Simulasi Kerja Praktek Mahasiswa Fakultas Teknologi Informasi Universitas Bale Bandung Menggunakan Unreal Engine 4”**.

Penulis tentu menyadari bahwa laporan ini masih jauh dari kata sempurna dan masih banyak terdapat kesalahan serta kekurangan di dalamnya. Untuk itu, penulis mengharapkan kritik serta saran dari pembaca untuk laporan ini, supaya laporan ini nantinya dapat menjadi laporan yang lebih baik lagi. Kemudian apabila terdapat banyak kesalahan pada laporan ini penulis mohon maaf yang sebesar-besarnya.

Penulis juga mengucapkan terima kasih kepada semua pihak yang terlibat dalam pembuatan laporan ini. Demikian, semoga laporan ini dapat bermanfaat. Terima kasih.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Bandung, Agustus 2019

Penulis

DAFTAR ISI

<i>ABSTRACT</i>	vi
ABSTRAK.....	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan	3
1.5. Metodologi Penelitian	3
1.6. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Landasan Teori.....	5
2.2. Dasar Teori.....	7
2.2.1. Game.....	7
2.2.2. <i>Game Engine</i>	9
2.2.3. Pemrograman Berorientasi Objek (PBO)	12
2.2.4. <i>Finite State Machine</i>	13
2.2.5. <i>Flowmap</i>	15
2.2.6. <i>Unified Modeling Language</i> (UML)	16
2.2.7. Mahasiswa	18
2.2.8. Pemodelan dan Simulasi.....	19
2.2.9. Multimedia.....	19
2.2.10. <i>Multimedia Development Life Cycle</i> (MDLC)	21
2.2.11. Wave Editor	22
2.2.12. SketchUp.....	23
2.2.13. Adobe Photoshop.....	25

2.2.14. Adobe Fuse	33
2.2.15. Adobe Audition	33
2.2.16. Mozilla Firefox	34
2.2.17. Microsoft Visio	34
2.2.18. DirectX	35
BAB III METODOLOGI PENELITIAN.....	37
3.1. Kerangka Pikir/ Skema Penelitian (Diagram).....	37
3.2. Penjelasan Skema Penelitian.....	38
3.2.1. Observasi Awal.....	38
3.2.2. Proses Pembuatan <i>Game</i>	38
3.2.3. Pembuatan Laporan	40
BAB IV ANALISIS DAN PERANCANGAN	41
4.1. Analisis.....	41
4.1.1. Instrumen Penelitian	41
4.1.2. Analisis Sistem	42
4.1.3. Deskripsi <i>Game</i>	44
4.1.4. Analisis Kebutuhan.....	47
4.1.5. Hasil Analisis.....	52
4.2. Perancangan	52
4.2.1. <i>Use Case</i> Diagram	52
4.1.1. <i>Activity</i> Diagram	63
4.1.2. Storyboard.....	69
4.1.3. Perancangan Arsitektur.....	78
BAB V IMPLEMENTASI DAN PENGUJIAN	85
5.1. Implementasi	85
5.1.1. Implementasi Perangkat Keras	85
5.1.2. Implementasi Perangkat Lunak	86
5.1.3. Implementasi Aplikasi	86
5.1.4. Implementasi Antarmuka.....	86
5.2. Pengujian.....	90
5.2.1. Pengujian Alpha.....	90
5.2.2. Pengujian <i>Beta</i>	99

BAB VI KESIMPULAN DAN SARAN	100
6.1. Kesimpulan	100
6.2. Saran.....	100
DAFTAR PUSTAKA	101
LAMPIRAN.....	104

DAFTAR TABEL

Tabel 2.1 Simbol Flowmap	15
Tabel 2.2 Keterangan <i>Tool</i> SketcUp	24
Tabel 2.3 Keterangan <i>Tool</i> Adobe Photoshop	26
Tabel 3.1 Prosedur Kerja Praktek	39
Tabel 4.1 Intrumen Penelitian	41
Tabel 4.2 <i>Character</i>	46
Tabel 4.3 Spesifikasi Perangkat Keras (<i>Hardware</i>) - Pengembang.....	48
Tabel 4.4 Spesifikasi Perangkat Keras (<i>Hardware</i>) – Pengguna.....	48
Tabel 4.5 Spesifikasi Perangkat Lunak (<i>Software</i>) – Pengembang	49
Tabel 4.6 Spesifikasi Perangkat Lunak (<i>Software</i>) – Pengguna	50
Tabel 4.7 Kebutuhan User.....	50
Tabel 4.8 Kebutuhan Data.....	51
Tabel 4.9 Definisi Aktor	53
Tabel 4.10 Definisi <i>Use Case</i>	53
Tabel 4.11 Skenario <i>Use Case</i> Mulai.....	54
Tabel 4.12 Skenario <i>Use Case</i> Credits.....	55
Tabel 4.13 Skenario <i>Use Case</i> Tutup.....	55
Tabel 4.14 Skenario <i>Use Case</i> Petunjuk.....	56
Tabel 4.15 Skenario <i>Use Case</i> Tutup.....	56
Tabel 4.16 Skenario <i>Use Case</i> Keluar	57
Tabel 4.17 Skenario <i>Use Case</i> Pengaturan Player	57
Tabel 4.18 Skenario <i>Use Case</i> Mulai Game	58
Tabel 4.19 Skenario <i>Use Case</i> Tutup.....	59
Tabel 4.20 Skenario <i>Use Case</i> Awal Permainan	59
Tabel 4.21 Skenario <i>Use Case</i> Inventory.....	61
Tabel 4.22 Skenario <i>Use Case</i> Pause Menu	63
Tabel 4.23 <i>Storyboard</i> Misi Pertama	70
Tabel 4.24 <i>Storyboard</i> Misi Kedua.....	70
Tabel 4.25 <i>Storyboard</i> Misi Ketiga.....	71
Tabel 4.26 <i>Storyboard</i> Misi Keempat.....	71

Tabel 4.27 <i>Storyboard</i> Misi Kelima	72
Tabel 4.28 <i>Storyboard</i> Misi Keenam	73
Tabel 4.29 <i>Storyboard</i> Misi Ketujuh	74
Tabel 4.30 <i>Storyboard</i> Misi Kedelapan	74
Tabel 4.31 <i>Storyboard</i> Misi Kesembilan	75
Tabel 4.32 <i>Storyboard</i> Misi Kesepuluh	76
Tabel 4.33 <i>Storyboard</i> Misi Kesebelas	77
Tabel 4.34 <i>Storyboard</i> Misi Keduabelas.....	77
Tabel 4.35 Antarmuka Menu Utama.....	79
Tabel 4.36 Antarmuka Credits	80
Tabel 4.37 Antarmuka Petunjuk	80
Tabel 4.38 Antarmuka Pengaturan Character	81
Tabel 4.39 Antarmuka Awal Permainan.....	82
Tabel 4.40 Antarmuka Inventory	82
Tabel 4.41 Antarmuka Pause Menu	83
Tabel 5.1 Spesifikasi Perangkat Keras.....	85
Tabel 5.2 Pengujian Menu Main Menu.....	90
Tabel 5.3 Pengujian Menu Credits.....	91
Tabel 5.4 Pengujian Menu Petunjuk	92
Tabel 5.5 Pengujian Menu Pengaturan Player	92
Tabel 5.6 Pengujian Menu Inventory	93
Tabel 5.7 Pengujian Menu Inventory – Klik Gambar Makanan.....	94
Tabel 5.8 Pengujian Menu Inventory – Klik Gambar Minuman	94
Tabel 5.9 Pengujian Menu Inventory – Klik Gambar Kertas	95
Tabel 5.10 Pengujian Menu Pause Menu.....	95
Tabel 5.11 Pengujian Kontrol Pemain	96
Tabel 5.12 Pengujian Kontrol Pemain – Interaksi Player	97
Tabel 5.13 Pengujian Lainnya.....	98

DAFTAR GAMBAR

Gambar 2.1 Contoh <i>Blueprint Visual Scripting</i>	11
Gambar 2.2 Contoh Penggunaan FSM Pada <i>Player</i>	14
Gambar 2.3 Simbol Pada Usecase Diagram	17
Gambar 2.4 Simbol Activity Diagram	18
Gambar 2.5 Contoh Model 3D Rektorat UNIBBA.....	19
Gambar 2.6 <i>Multimedia Development Life Cycle</i>	22
Gambar 2.7 <i>Tool SketcUp</i>	23
Gambar 3.1 Kerangka Pikir.....	37
Gambar 4.1 Analisis Sitem Yang Berjalan	42
Gambar 4.2 Analisis Sistem Usulan	43
Gambar 4.3 <i>Use Case Diagram</i>	53
Gambar 4.4 Diagram <i>Activity</i> Mulai	64
Gambar 4.5 Diagram <i>Activity</i> Credits	64
Gambar 4.6 Diagram <i>Activity</i> Tutup Credits.....	64
Gambar 4.7 Diagram <i>Activity</i> Petunjuk	65
Gambar 4.8 Diagram <i>Activity</i> Tutup Petunjuk.....	65
Gambar 4.9 Diagram <i>Activity</i> Keluar Game	65
Gambar 4.10 Diagram <i>Activity</i> Pengaturan Player	66
Gambar 4.11 Diagram <i>Activity</i> Mulai Game.....	66
Gambar 4.12 Diagram <i>Activity</i> Tutup Pengaturan Player.....	67
Gambar 4.13 Diagram <i>Activity</i> Awal Permainan.....	67
Gambar 4.14 Diagram <i>Activity</i> Inventory	68
Gambar 4.15 Diagram <i>Activity</i> Pause Menu	69
Gambar 4.16 Struktur Menu	78
Gambar 4.17 Jaringan Semantik	84

DAFTAR LAMPIRAN

Lampiran 1 : Hasil Wawancara – 10 Mahasiswa.....	104
Lampiran 2 : Hasil Wawancara – Mahasiswa Angga Triatna.....	105
Lampiran 3 : Hasil Wawancara – Petugas TU Yusuf Muharam, S.Kom	106
Lampiran 4 : Hasil Wawancara – Mahasiswa Megantari Suhendar	107
Lampiran 5 : Script Program.....	108
Riwayat Hidup Penulis	144

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan aplikasi permainan komputer saat ini sudah sangat menjamur. Munculnya berbagai perangkat lunak yang bertujuan untuk memudahkan pengembangan permainan pada perangkat komputer yang gratis membuat studio pengembang permainan komputer semakin banyak. Bermain *game* juga merupakan salah satu sarana pembelajaran, dimana pemain atau biasa disebut *player* bisa mempelajari apa yang ada dalam *game* yang dimainkan-nya. *Game Simulation* merupakan jenis *game* yang mengutamakan pemain agar menikmati simulasi yang disajikan, jenis *game* ini juga bisa digunakan untuk belajar, misal-nya simulasi mengemudi.

Fakultas Teknologi Informasi merupakan salah satu Fakultas yang ada di Universitas Bale Bandung (UNIBBA). Di Fakultas Teknologi Informasi ini pada semester 5 ada mata kuliah kerja praktek. Dimana mata kuliah ini merupakan salah satu matakuliah yang wajib di lakukan mahasiswa Fakultas Teknologi Informasi untuk melakukan tahap selanjutnya yaitu skripsi.

Dalam mata kuliah kerja praktek Fakultas Teknologi Informasi Universitas Bale Bandung ada beberapa prosedur yang mahasiswa harus lakukan, antara lain mendaftarkan diri sebagai peserta, menyerahkan proposal, melakukan kerja praktek-nya, dan lain-lain sampai akhirnya dinyatakan lulus dari mata kuliah Kerja Praktek.

Dari adanya beberapa prosedur tersebut. Penyusun akan mensimulasikan prosedur dari mata kuliah kerja praktek ini kedalam sebuah *video game*. Dimana nanti mahasiswa bisa memainkan *video game* tersebut untuk mengetahui prosedur mata kuliah kerja praktek.

Oleh karena itu dengan adanya *game* simulasi ini diharapkan dapat membantu mahasiswa Fakultas Teknologi Informasi Universitas Bale Bandung untuk mengetahui prosedur mengenai mata kuliah kerja praktek secara jelas. Maka dari

uraian tersebut penyusun mengambil judul “Game Simulasi Kerja Praktek Mahasiswa Fakultas Teknologi Informasi Universitas Bale Bandung Menggunakan Unreal Engine 4”.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka rumusan masalah pada penelitian ini adalah bagaimana mensimulasikan prosedur kerja praktek Fakultas Teknologi Informasi Universitas Bale Bandung Kedalam *video game*?

1.3. Batasan Masalah

Berdasarkan rumusan masalah diambil beberapa batasan masalah pada penelitian ini antara lain :

1. Membuat *game* simulasi kerja praktek mahasiswa Fakultas Teknologi Informasi Universitas Bale Bandung yang dimainkan secara *single player*.
2. *Software Engine* yang digunakan *Unreal Engine 4* (UE4).
3. Bahasa yang digunakan *Blueprints Visual Scripting* (BVS).
4. Adapun prosedur kerja praktek di *video game ini* antara lain :
 - a. Daftar Kerja Praktek.
 - b. Membuat Proposal.
 - c. Membayar Mata Kuliah Kerja Praktek.
 - d. Menyerahkan Proposal Beserta FotoCopy Pembayaran.
 - e. Melakukan Kerja Praktek.
 - f. Membuat Laporan Kerja Praktek.
 - g. Melakukan Bimbingan Kerja Praktek.
 - h. Menyerahkan Laporan Kerja Praktek.
 - i. Melakukan Sidang Kerja Praktek.

1.4. Tujuan

Tujuan dari penelitian ini adalah untuk menunjukkan prosedur dari matakuliah kerja praktek secara visual kepada mahasiswa Fakultas Teknologi Informasi Universitas Bale Bandung.

1.5. Metodologi Penelitian

1.5.1. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan pada penelitian ini adalah sebagai berikut :

1. Studi Literatur
2. Observasi
3. Wawancara

1.5.2. Metode Pengembangan Aplikasi

Metode pengembangan yang dipakai adalah metode *Multimedia Development Life Cycle (MDLC)*, dimana MDLC ini memiliki 6 tahap antara lain :

1. Konsep (*Consept*) : Membuat Konsep *Video Game*.
2. Perancangan (*Design*) : Mendesain Alur Cerita, Character, Dll.
3. Pengumpulan Bahan dan Data (*Material Collecting*) : Mengumpulkan Data Yang Diperlukan.
4. Penyusunan dan Pembuatan (*Assembly*) : Menyusun dan Membuat *Video Game*.
5. Pengujian (*Testing*) : Menguji *Video Game* Apakah Sudah Sesuai Dengan Desain.
6. Menyerbar Luaskan (*Distribution*) : Membuat File *.Exe *Video Game*.

1.6. Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan penelitian ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Dalam bab ini diuraikan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penelitian.

BAB II : TINJAUAN PUSTAKA

Dalam bab ini diuraikan mengenai teori yang berhubungan dengan penelitian ini, seperti pengertian Pemrograman Berorientasi Objek, Pemodelan dan Simulasi, Multimedia, *Game*, dan hal-hal lain yang terkait dengan pembuatan *game* simulasi.

BAB III : METODOLOGI PENELITIAN

Dalam bab ini diuraikan mengenai kerangka pikir beserta deskripsinya yang digunakan dalam pembuatan *game* simulasi Kerja Praktek Mahasiswa Fakultas Teknologi Infomasi.

BAB IV : ANALISIS DAN PERANCANGAN

Dalam bab ini diuraikan mengenai analisis terhadap masalah dalam pembuatan *game* dan dilanjutkan dengan membuat rancangan mengenai pemecahan masalah dalam aplikasi *game*.

BAB V : IMPLEMENTASI DAN PENGUJIAN

Dalam bab berisi implementasi dan pengujian dari hasil pemecahan masalah dan perancangan yang didapat secara teoritis serta implementasi secara menyeluruh pada alat yang dibangun.

BAB VI : KESIMPULAN DAN SARAN

Dalam Bab ini berisi kesimpulan yang diperoleh dari bab-bab sebelumnya dan saran-saran berisi tentang tindak lanjut atau pengembangan yang dapat dilakukan terhadap sistem yang telah dibuat

BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

Landasan teori ini berisi tentang referensi dari jurnal atau penelitian terdahulu sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

2.2.1. Pengembangan Aplikasi Peta Tiga Dimensi Menggunakan Unreal Engine (Studi Kasus : Jurusan Biologi Fakultas MIPA Institut Teknologi Sepuluh Nopember Surabaya) (Saputra & Samopa, 2012)

Penelitian ini berlatar belakang mengenai pemanfaatan teknologi pengembangan visual 3D sebagai penyampaian informasi yang lebih mengenai suatu bidang, dalam penelitian ini digunakan untuk menunjukkan tata ruang dan penggambaran contoh-contoh kegiatan yang ada pada Jurusan Biologi Fakultas MIPA Institut Teknologi Sepuluh Nopember Surabaya, sehingga nantinya bisa digunakan sebagai saran yang informatif kepada para mahasiswa ataupun calon mahasiswa yang ingin menuntut ilmu di Fakultas MIPA Institut Teknologi Sepuluh Nopember Surabaya, khususnya Jurusan Biologi.

Penelitian ini menggunakan metode yang dibuat oleh si penyusun, yang dilakukan adalah survey lokasi kemudian dilanjutkan dengan pembuatan aplikasi dan pembuatan file *.exe

Kesimpulannya Hasil dari aplikasi ini akan digunakan untuk menunjukkan tata ruang dan penggambaran contoh-contoh kegiatan yang terdapat pada Jurusan Biologi Fakultas MIPA Institut Teknologi Sepuluh Nopember Surabaya secara akurat dan interaktif. Dengan begitu pengguna

akan lebih tertarik dengan produk yang ditawarkan dan pengguna dapat menerima informasi tanpa harus mengasumsikan tata ruang gedung tersebut.

2.2.2. Aplikasi Game Geografi berbasis Multimedia Interaktif (Studi Kasus Siswa Kelas IX SMPN 1 RAO) (Novaliendry, 2013)

Penelitian ini berlatar belakang mengenai bentuk media pembelajaran yang dikemas dalam sebuah *game* edukasi berbasis multimedia interaktif. Penelitian aplikasi Geografi ini dirancang berdasarkan pelajaran yang terkait dengan permukaan bumi atau proses pembelajaran dari studi geografi regionalisasi daerah bumi dalam bentuk peta pengetahuan dan negara.

Penelitian ini menggunakan metode *Multimedia Development Life Cycle* (MDLC), dimana MDLC merupakan tahapan-tahapan yang dilakukan untuk membuat sebuah multimedia. MDLC ini meliputi *concept, design, material collecting, assembly, testing* dan *distribution*.

Kesimpulan perancangan aplikasi *game* geografi ini untuk menunjang media pembelajaran di sekolah dengan cara belajar simulasi melalui *game* edukasi. *Game* edukasi ini dirancang dengan dengan unsur media yang terdiri dari teks, grafis, animasi, dan suara yang disajikan secara interaktif. *Game* edukasi ini merupakan alternatif dan inovasi baru untuk mempermudah proses belajar siswa pada mata pelajaran IPS bidang kajian Geografi Regional.

2.2.3. Video Game Edukatif Terhadap Kemampuan Mengenal Anggota Tubuh Manusia Pada Anak Tunarungu Kelas I (Hirnandin, 2018)

Penelitian ini berlatar belakang mengenai bagaimana cara untuk mengenalkan anggota tubuh manusia terhadap anak tungarungu kelas 1 dan bagaimana pengaruh *game* edukasi ini terhadap anak tungarungu.

Penelitian ini menggunakan metode pendekatan penelitian kuantitatif dikarenakan data yang digunakan ialah variabel bebas (variabel

independen) dan variabel terikat (variabel dependen) dan menguji rumus yang telah dirumuskan terlebih dahulu serta hasil. Metode kuantitatif data penelitian berupa angka-angka dan dianalisis menggunakan statistika, penelitian kuantitatif yang digunakan untuk meneliti populasi atau sampel tertentu, pengumpulan data menggunakan instrumen penelitian, analisis data bersifat kuantitatif dengan tujuan untuk menguji hipotesis yang telah ditetapkan. Penelitian ini menggunakan desain *pre-eksperimen* dan rancangan penelitian *one-group pretest-posttest design*. Karena tidak adanya variabel kontrol dan subjek tidak diambil secara acak selain itu subjek diberikan *pre-testter* lebih dahulu sebelum diberikan perlakuan atau *treatment* kemudian baru dilakukan *post-test*. Hal ini bertujuan untuk membandingkan keadaan sebelum dan sesudah diberi perlakuan.

Kesimpulan hasil penelitian menunjukkan bahwa *video game* edukatif mempunyai pengaruh terhadap kemampuan mengenal anggota tubuh pada anak tunarungu. Hal ini terlihat bahwa kemampuan mengenal-mengenal anggota tubuh pada anak tunarungu mengalami peningkatan, aspek yang dinilai adalah kemampuan mengenal bagian anggota tubuh manusia, mengenal fungsi bagian dari anggota tubuh dan merawat bagian dari anggota tubuh.

2.2. Dasar Teori

Dasar teori ini berisikan beberapa teori yang digunakan untuk membangun aplikasi *game* simulasi ini. Berikut merupakan dasar-dasar teori yang digunakan.

2.2.1. Game

Game adalah sebuah permainan komputer interaktif yang di kendalikan *mikroprosesor*. Sebuah permainan komputer atau *video game* menggunakan satu atau lebih alat *input*, biasanya sebuah tombol atau kombinasi dari *joystick*, sebuah *keyboard* dan *mouse*. (Susilawati, 2014).

Dalam *video game* ada beberapa istilah yang digunakan antara lain *platform* dan *genre* dimana *platform* merupakan istilah untuk menyebutkan alat yang digunakan untuk memainkan sebuah *video game*, sedangkan *genre* merupakan jenis-jenis *video game* berdasarkan kesamaan yang dimiliki *video game* tersebut. Berikut adalah contoh dari *platform* dan *genre* dalam sebuah *video game*.

1. Platform Game

Platform *game* adalah jenis alat yang digunakan untuk memainkan *game*, antara lain :

a. *Arcade Games*.

Arcade Games yaitu yang sering disebut ding-dong di Indonesia, biasanya berada di daerah atau tempat khusus dan memiliki box atau mesin yang memang khusus di design untuk jenis game tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa masuk dan menikmati, seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya tentunya).

b. *PC Games*.

PC Games yaitu *video game* yang dimainkan menggunakan *personal computer*.

c. *Console Games*.

Console Games yaitu *video games* yang dimainkan menggunakan *console* tertentu, seperti playstation 2, playstation 3, xbox 360, nintendo wii dan *console* lainnya.

d. *Handled Games*.

Handled Games yaitu *video games* yang dimainkan di *console* khusus *game* yang dapat dibawa kemana-mana, contoh nintendo ds, sony psp dan nintendo switch.

e. *Mobile Games*.

Mobile Games yaitu *video games* yang dapat dimainkan atau khusus untuk dimainkan pada perangkat *mobile phone* atau pda.

2. Genre Game

Genre game adalah pengelompokan dari beberapa game yang memiliki karakteristik yang sama, antara lain :

a. *Role Playing Game (RPG).*

RPG adalah salah satu *game* yang mengandung unsur *experience* atau *leveling* dalam *gameplay*-nya. Biasanya dalam *game* ini kita memiliki kebebasan untuk menjelajah dunia *game* tersebut.

b. *First Person Shooter (FPS).*

FPS adalah *game* yang dimana pemain disuguhkan *gameplay* tembak-menembak antar *player*.

c. *Strategy.*

Strategy adalah *genre game* yang biasanya *player* harus berfikir untuk memainkan *game* ini.

d. *Simulation.*

Simulation Adalah *genre* yang mementingkan realisme. Segala faktor pada *game* ini sangat diperhatikan agar semirip didunia nyata. Segala nilai, material, referensi, dan faktor lainnya adalah berdasarkan dunia nyata.

e. *Puzzle.*

Puzzle adalah *game* teka-teki dimana pemain di haruskan menyelesaikan teka-teki agar mendapatkan skor dan lanjut ke teka-teki berikutnya.

2.2.2. *Game Engine*

Game Engine merupakan alat atau perangkat lunak (*software*) yang digunakan untuk membuat sebuah *video game* atau permainan elektronik. Dalam dunia *game* ada begitu banyak *game engine* antara lain :

1. Unreal Engine
2. Unity3D
3. Game Maker Studio

4. Construct
5. Godot Engine
6. Ren'Py
7. RPG Maker

Dalam penelitian ini penulis menggunakan *unreal engine* sebagai alat untuk membuat *game* simulasi ini. *Game engine* yang digunakan adalah seri ke 4 dari *Unreal Engine* dan bahasa yang di gunakan dalam pengembangan *game* ini menggunakan *Blueprint Visual Scripting*.

1. Unreal Engine

Unreal Engine dibuat oleh Epic Game. Dimulai debut pada tahun 1998 dengan nama *Unreal Engine 1* yang mempunyai teknik render terpadu, kecerdasan buatan, visibilitas jaringan, dan manajemen system file kedalam engine yang lengkap.

Selanjutnya pada tahun 2002 *Unreal Engine* memperbaharui *software*-nya dan diubah dengan nama *Unreal Engine 2*. *Unreal Engine 2* ini muncul dengan meningkatkan *asset* serta menambahkan dukungan untuk Game Cube dan Xbox, dan ingin bisa juga dipakai untuk console Wii, X360, PS3, PSP dan 3DS.

Selanjutnya pada tahun 2004 *Unreal Engine* kembali memperbaharui *software*-nya dan di ubah dengan nama *Unreal Engine 3*. *Unreal Engine 3* ini dirancang untuk mengambil keuntungan sepenuhnya di program shader dalam DirectX versi 9-11 untuk Windows dan Xbox 360 yang dilengkapi dengan system OpenGL termasuk PS3, OS X, iOS, Android, tahap 3D untuk Adobe Flash Player 11, javascript / WebGL, PSVita dan Wii.

Selanjutnya pada tahun 2012 sampai dengan tahun sekarang *Unreal Engine* kembali memperbaharui *software*-nya dan diubah dengan nama *Unreal Engine 4* dan pada tahun 2019 sekarang *Unreal Engine 4* sampai pada versi 4.23. UE4 ini mempunyai fitur utama yaitu iluminasi global real-time menggunakan *voxel kerucut tracing*, menghilangkan kebutuhan untuk perhitungan tiap penerangan. UE4 juga dilengkapi fitur pengembang baru untuk mengurangi waktu iterasi dan memungkinkan memperbarui langsung

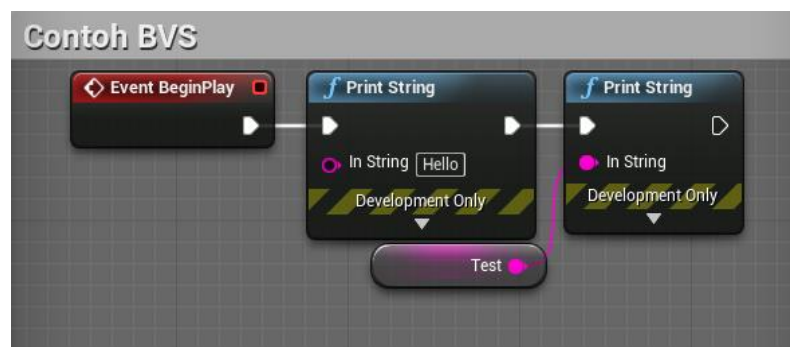
C++ kode. Fitur baru debugger untuk “kismet” (mesin scripting visual pada UE3) memungkinkan developer dapat memvisualisasikan kode saat pengujian. Developer juga dapat melihat sourcecode dan meng-*edit*-nya dalam Visual Studio (Harsatput, 2013).

Untuk pengetahuan lebih dalam mengenai unreal engine 4 silahkan menuju link berikut ini : <https://docs.unrealengine.com/en-US/GettingStarted/index.html>

2. Blueprint Visual Scripting (BVS)

Blueprints Visual Scripting di *Unreal Engine* adalah sistem *scripting gameplay* yang lengkap berdasarkan konsep menggunakan antarmuka yang berbasis *node* untuk membuat elemen *gameplay* dari dalam *Unreal Editor* (Epic Games.Inc, 2004).

Seperti kebanyakan bahasa *scripting* umum, digunakan untuk mendefinisikan kelas *Object - Oriented (OO)* atau objek yang berada dalam *Engine*. Saat menggunakan *UE4*, akan sering menemukan bahwa objek yang didefinisikan menggunakan *Blueprint* adalah bahasa sehari-hari yang disebut sebagai "*Blueprints*".



Gambar 2.1 Contoh *Blueprint Visual Scripting*

Dalam gambar tersebut menjelaskan pada saat pertama kali memulai maka *print string* “Hello” untuk ini bias langsung ditulis oleh developer dan untuk yang kedua menggunakan sebuah variable yang dimana variable tersebut isinya bisa di-*edit* oleh developer atau juga bisa mengambil dari variabel lain.

2.2.3. Pemrograman Berorientasi Objek (PBO)

Pemrograman Berorientasi Objek merupakan pemrograman yang berorientasikan kepada objek. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. (Saputra S. E., 2011).

Menurut (Saputra S. E., 2011), Dalam konsep pemrograman berorientasi objek (PBO) dikenal beberapa istilah umum, yaitu :

1. Objek (*Object*)

Objek merupakan perwujudan dari kelas, setiap objek akan mempunyai atribut dan fungsi yang dimiliki oleh kelas-nya.

2. Kelas (*Class*)

Kelas merupakan model yang didalam nya berisi kumpulan atribut dan fungsi untuk suatu tujuan tertentu.

3. Atribut (*Attribute*)

Atribut dalam sebuah kelas disebut variabel. Atribut dapat memiliki hak akses *private*, *public* maupun *protected*. *Private* artinya hanya dapat di akses secara langsung oleh kelas yang membungkusnya, *public* artinya dapat di akses secara langsung oleh kelas lain di luar kelas yang membungkusnya, *protected* artinya tidak dapat di akses secara langsung oleh kelas lain di luar kelas yang membungkusnya, kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya.

4. Fungsi (*Method*)

Fungsi merupakan prosedur yang dibuat oleh seorang programmer didalam suatu kelas. Dengan kata lain, fungsi pada sebuah kelas hampir sama dengan fungsi atau prosedur pada pemrograman prosedural.

Menurut (Saputra S. E., 2011), Selain istilah umum tersebut ada beberapa konsep pemrograman berorientasi objek yang lain-nya, antara lain :

1. Abstraksi (*Abstraction*)

Abstraksi Adalah suatu cara untuk melihat suatu object dalam bentuk yang lebih sederhana. Contoh-nya adalah komputer, dimana komputer

memiliki berbagai macam komponen elektronik. Dengan adanya abstraksi dapat membuat sistem kompleks tersebut menjadi lebih sederhana dengan membuat subsistem, komputer memiliki subsistem daya, subsistem penyimpanan, subsistem *input/output* dan sebagainya.

2. Pembungkusan (*Encapsulation*)

Pembungkusan merupakan suatu mekanisme untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan interferensi atau penyalahgunaan dari luar sistem dan sekaligus menyederhanakan penggunaan sistem tersebut dan memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak.

3. Pewarisan (*Inheritance*)

Pewarisan adalah proses mewariskan atribut dan fungsi yang ada di dalam kelas kepada kelas lain. Ini seperti membuat *child-class* dimana *child-class* memiliki semua atribut dan fungsi dari *parent* nya.

4. Polimorfism

Polimorfism merupakan konsep yang memungkinkan digunakannya suatu interface yang sama untuk memerintah suatu object agar melakukan suatu tindakan yang mungkin secara prinsip sama tetapi secara proses berbeda. Contoh-nya player menekan tombol di keyboard E dan tindakan yang dilakukannya berbeda-beda tergantung fungsi yang ditetapkan pada objek.

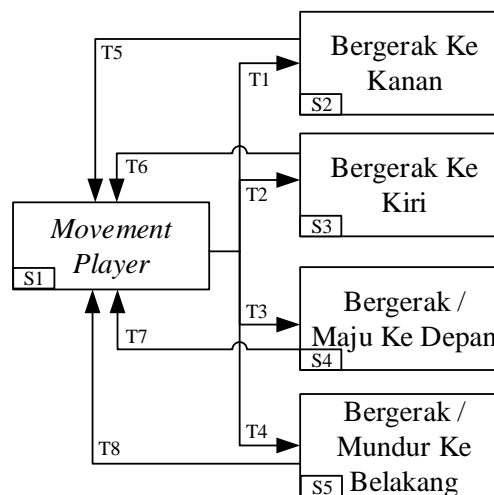
2.2.4. *Finite State Machine*

Finite State Machine (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: State (Keadaan), Event (kejadian) dan action (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu state yang aktif. Sistem dapat beralih atau bertransisi menuju state lain jika mendapatkan masukan atau event tertentu,

baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks (Simanjuntak, 2016).

FSM terdiri dari dua jenis, yaitu FSM ber-*output* dan FSM tidak ber-*output*. FSM tidak ber-*output* digunakan untuk pengenalan bahasa dalam komputer, dengan *input* yang dimasukkan akan diperoleh apakah *input* tersebut dikenal oleh bahasa komputer atau tidak. Salah satu penggunaan FSM tidak ber-*output* adalah program *compiler*, yaitu program untuk memeriksa apakah perintah yang digunakan pengguna benar atau salah. Sementara untuk FSM ber-*output* digunakan untuk merancang mesin atau sistem.

Berikut adalah contoh gambar dan penjelasan yang penyusun gunakan untuk pergerakan *player* :



Gambar 2.2 Contoh Penggunaan FSM Pada *Player*

Dalam gambar tersebut terdapat 5 *state* {S1, S2, S3, S4, S5} yang mungkin terjadi, setiap *state*-nya dapat berpindah *state* jika kondisi terpenuhi. Sebagai contoh *state* S1 dapat berpindah jika kondisi T1/T2/T3/T4 terpenuhi.

2.2.5. Flowmap





Flowmap adalah campuran peta dan *flow chart*, yang menunjukkan pergerakan benda dari satu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan, atau jumlah paket dalam jaringan. *Flowmap* menolong analisis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoprasian (Dianty, 2016).

Menurut (Dianty, 2016), Bila seseorang ingin membuat *flowmap* ada beberapa petunjuk yang harus diperhatikan, seperti :

1. *Flowmap* digambarkan dari halaman atas ke bawah dan kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat di mengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus di tentukan secara jelas.
4. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
5. Lingkup dan range dari aktifitas yang sedang di gambarkan harus di telusuri dengan hati-hati.
6. Gunakan simbol-simbol flowchart yang standar.

Menurut (Iqram, 2013), Adapun simbol-simbol yang biasa digunakan dalam pembuatan flowmap antara lain :

Tabel 2.1 Simbol Flowmap

No	Simbol	Nama	Keterangan
1		<i>Terminator</i>	Awal Atau Akhir Diagram Atau Entitas <i>External</i> .
		Dokumen (<i>Document</i>)	Menunjukkan Dokumen Sebagai Yang Digunakan Untuk Merekam Data Terjadinya Suatu Transaksi.
2		<i>Operasional Manual</i>	Menunjukkan Proses Yang Dikerjakan Secara <i>Manual</i> .
3		Proses	Menunjukkan Kegiatan Proses Dari Operasi <i>program Computer</i> .

4		<i>File</i>	Menunjukkan <i>Input</i> Atau <i>Output</i> Menggunakan <i>Ike</i> .
5		<i>Decision</i>	Menunjukan Pilihan Yang Akan Dikerjakan Atau Keputusan Yang Harus Dibuat Dalam Proses Pengolahan Data
7		<i>Input Manual</i>	Menunjukkan <i>Input</i> Yang Menggunakan <i>On-Line Keyboard</i> .
6		<i>Off Line Storage</i>	Digunakan Untuk Menyimpan Data Secara <i>Manual</i> Dan Sementara, Jika “A” Berarti Disimpan Menurut Abjad, “N” Berarti Disimpan Menurut Nomor Urut Dan Jika “T” Berarti Disimpan Menurut Kronologis Atau Menurut Tanggal.
9		Garis Aliran (<i>Flow Line</i>)	Menunjukan Arus Data Antar Simbol/Proses
9		<i>Conector (On-Page Connector)</i>	Digunakan untuk Penghubung Dalam Satu Halaman
10		<i>Conector (Off-Page Connector)</i>	Digunakan Untuk Penghubung Berbeda Halaman

2.2.6. Unified Modeling Language (UML)




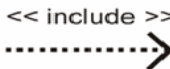
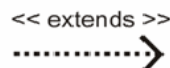

Unified Modeling Language (UML) adalah Bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek).” Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Nugroho, 2010).

Ada 3 diagram yang terdapat pada UML yaitu Usecase Diagram, Class Diagram, dan Activity Diagram. Namun yang digunakan hanya 2, berikut penjelasan dari diagram yang digunakan:

1. Usecase Diagram

Usecase adalah teknik untuk merekam persyaratan fungsional sebuah sistem. Use Case mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Usecase Diagram menampilkan aktor mana yang menggunakan use case mana, use case mana yang memasukkan use case lain dan hubungan antara aktor dan use case .

Berikut ini adalah simbol-simbol yang ada pada Use Case Diagram:

Nama Simbol	Simbol
Aktor	
Use Case	
Association Relationship	
Include Relationship	
Extend Relationship	
Generalisasi Relationship	







Gambar 2.3 Simbol Pada Usecase Diagram

2. Activity diagram

Activity diagram memiliki pengertian yaitu lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. Memiliki struktur diagram yang mirip flowchart atau data flow diagram pada perancangan terstruktur. Memiliki pula manfaat yaitu apabila

kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Dan activity dibuat berdasarkan sebuah atau beberapa use case pada use case diagram (P, 2013)

Berikut ini adalah simbol-simbol pada activity diagram:

Simbol	Keterangan
	Start Point
	End Point
	Activities
	Fork (Percabangan)
	Join (Penggabungan)
	Decision
Swimlane	Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama)

Gambar 2.4 Simbol Activity Diagram

2.2.7. Mahasiswa

Mahasiswa adalah orang yang belajar di perguruan tinggi, baik di universitas, institut atau akademi. Mereka yang terdaftar sebagai murid di perguruan tinggi dapat disebut sebagai mahasiswa. Tetapi pada dasarnya makna mahasiswa tidak sesempit itu. Terdaftar sebagai mahasiswa di sebuah Perguruan Tinggi hanyalah syarat administratif menjadi mahasiswa, tetapi menjadi mahasiswa mengandung pengertian yang lebih luas dari sekedar masalah administratif itu sendiri.

Menyandang gelar mahasiswa merupakan suatu kebanggaan sekaligus tantangan. Betapa tidak, ekspektasi dan tanggung jawab yang diemban oleh mahasiswa begitu besar. Pengertian mahasiswa tidak bisa diartikan kata per kata, Mahasiswa adalah Seorang agen pembawa perubahan. Menjadi seorang yang dapat memberikan solusi bagi permasalahan yang dihadapi oleh suatu masyarakat bangsa di berbagai belahan dunia (Santoso, 2012).

2.2.8. Pemodelan dan Simulasi

Model adalah rencana, representasi, atau deskripsi yang menjelaskan suatu objek, sistem, atau konsep, yang seringkali berupa penyederhanaan atau idealisasi. Bentuknya dapat berupa model fisik (maket), bentuk prototipe, model citra (gambar rancangan, citra komputer) dan rumusan matematis (Wikipedia, 2019).

Pada *video game* ini model yang digunakan berupa model tiga dimensi (3D) dimana model 3D adalah proses mengembangkan matematika representasi dari setiap tiga-dimensi benda (baik benda mati atau hidup) melalui perangkat lunak khusus (Pratama, 2014).



Gambar 2.5 Contoh Model 3D Rektorat UNIBBA

Simulasi adalah suatu proses peniruan dari sesuatu yang nyata beserta keadaan sekelilingnya (*state of affairs*). Aksi melakukan simulasi ini secara umum menggambarkan sifat-sifat karakteristik kunci dari kelakuan sistem fisik atau sistem. Teknik simulasi adalah teknik untuk merepresentasikan atau meniru kondisi real (suatu sistem nyata) dalam bentuk bilangan dan simbol (dengan memanfaatkan program komputer), sehingga menjadi mudah untuk dipelajari (Sitompul, 2014).

2.2.9. Multimedia

Multimedia adalah suatu sarana atau media melalui penggunaan komputer dalam menggabungkan dan menyajikan suara, teks,

, animasi, *audio* dan *video* dengan alat bantu dan koneksi sehingga pengguna dapat bernavigasi, berinteraksi, berkarya dan berkomunikasi (Aisha, 2018).

Menurut (Aisha, 2018), Dalam multimedia terdapat 2 kategori antara lain Multimedia Communication dan Multimedia Content Production. Berikut adalah penjelasan-nya :

1. Multimedia Communication

Multimedia Communication merupakan penggunaan media yang memiliki fungsi mempublikasikan informasi. Media yang digunakan dalam kategori ini adalah TV, Film, Game, Musik, Internet, dan Media Cetak.

2. Multimedia Content Production

Multimedia Content Production adalah penggunaan beberapa media yang berbeda seperti Teks, Animasi, *Audio*, *Video*, gambar (grafik) yang dipadukan untuk menghasilkan produk multimedia seperti musik, game, film dan entertainment.

Menurut (Aisha, 2018), Selain kategori ada juga jenis-jenis multimedia antara lain multimedia inreraktif, multimedia hiperaktif, multimedia linear, dan multimedia kits. Berikut adalah penjelasan-nya :

1. Multimedia Interaktif

Multimedia interaktif adalah multimedia yang dilengkapi dengan alat bantu berupa mouse, keyboard dan lain-lain yang dioperasikan oleh pengguna. Multimedia interaktif dapat menggabungkan beberapa media lain yang terdiri dari teks, video dan lain-lain. Contoh : *Game* dan *Virtual Reality*.

2. Multimedia Linear

Multimedia Liner adalah jenis multimedia yang berjalan lurus. Multimedia linear berlangsung tanpa kontrol navigasi dari pengguna. Contoh : TV dan Film.

3. Multimedia Hiperaktif

Multimedia ini mempunyai struktur dengan unsur terkait yang nantinya dapat diarahkan oleh pengguna melalui link dengan unsur multimedia yang ada. Contoh : World Wide Web dan Web Site.

4. Multimedia Kits

Multimedia ini digunakan sebagai pembelajaran yang melibatkan lebih dari satu jenis media dan diorganisir oleh topik tunggal. Perangkat yang termasuk dalam multimedia kits yaitu CD-ROM dan Audio.

2.2.10. *Multimedia Development Life Cycle (MDLC)*

Menurut Saiful Imran (2014), *Multimedia Development Life Cycle* merupakan tahapan-tahapan yang dilakukan untuk membuat sebuah multimedia. Tahapan MDLC meliputi *consept, design, material collecting, assembly, testing* dan *distribution*, berikut adalah penjelasan beserta gambarnya

1. Konsep (*Consept*)

Tahap dimana menentukan tujuan aplikasi dan siapa pengguna aplikasi (Identifikasi Audience).

2. Perancangan (*Design*)

Tahap dimana pembuat atau pengembang proyek multimedia menjabarkan secara rinci apa yang akan dilakukan dan bagaimana proyek multimedia tersebut akan dibuat. Pembuatan naskah ataupun navigasi serta proses desain lain harus secara lengkap dilakukan. Pada tahap ini akan harus mengetahui bagaimana hasil akhir dari proyek yang akan dikerjakan.

3. Pengumpulan Bahan-Bahan (*Material Collecting*)

Tahap dimana proses untuk pengumpulan segala sesuatu yang dibutuhkan dalam proyek. Mengenai materi yang akan disampaikan, kemudian file-file *multimedia* seperti *audio, video*, dan gambar yang akan dimasukkan dalam penyajian proyek *multimedia* tersebut.

4. Penyusunan dan Pembuatan (*Assembly*)

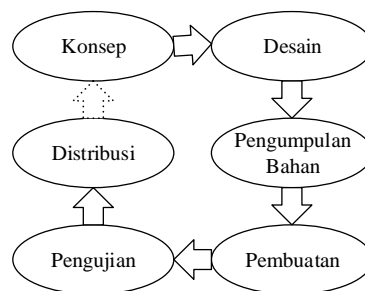
Tahap dimana waktunya proyek *multimedia* diproduksi. Materi-materi serta *file-file multimedia* yang sudah didapat kemudian dirangkai dan disusun sesuai desain. Pada proses ini sangat dibutuhkan kemampuan dari ahli agar mendapatkan hasil yang baik.

5. Pengujian (*Testing*)

Tahap dimana setelah hasil dari proyek *multimedia* jadi, perlu dilakukan uji coba. Uji coba dilakukan dengan menerapkan hasil dari proyek *multimedia* tersebut pada pembelajaran secara minor. Hal ini dimaksudkan agar apa yang telah dibuat sebelumnya memang tepat sebelum dapat diterapkan dalam pembelajaran secara massal.

6. Menyebar Luaskan (*Distribution*)

Tahap dimana penggandaan dan penyebaran hasil kepada pengguna. *Multimedia* perlu dikemas dengan baik sesuai dengan media penyebar luasannya, apakah melalui CD/DVD, download, ataupun media yang lain.



Gambar 2.6 *Multimedia Development Life Cycle*

2.2.11. Wave Editor

Wave Editor adalah aplikasi yang berukuran kecil, gratis dan sederhana sehingga cocok bagi anda yang pemula terhadap proses editing audio. Anda bisa melakukan proses editing dasar seperti pemotongan, duplikasi, menghapus dan menyisipkan suatu bagian dari audio tersebut dengan mudah. Format audio yang didukung adalah MP3, WAV dan WMA dalam proses *audio editing*-nya (Chandra, 2013).

Beberapa operasi pada audio yang disediakan adalah:

1. Potong, duplikat, tambah dan hapus bagian *audio*.
2. Memasukkan suara hening.
3. Normalisasi *volume* suara.
4. Meningkatkan *volume* suara dengan pembatasan puncak.

5. *Fade-in* dan *Fade-out* audio.
6. Simpan bagian *audio* yang dipilih.
7. *Reverse* dan *Invert audio*.
8. Langkah *Undo* dan *Redo* tak terbatas.
9. Mainkan dan hentikan *audio*.
10. *Zoom-in* dan *Zoom-out* grafik *audio*.
11. Simpan file *audio* ke format WAV dan MP3

2.2.12. SketchUp

SketchUp merupakan aplikasi berbasis desain gambar yang mudah dan cukup *powerfull*, dibalik *tool* yang sederhana ternyata *software* ini bisa dibandingkan dengan *software* sejenisnya untuk gambar tiga dimensi seperti desain rumah atau yang lainnya, tidak hanya itu Google SketchUp mempunyai banyak kelebihan dalam hal teknik gambar, begitu cepat, mudah dan efisien, apalagi kalau digabungkan dengan plugin Vray, sejenis *software Rendering* yang paling populer sekarang, hasilnya bisa jauh lebih bagus (Hairi, 2014).

Program ini sangat populer dikalangan pencinta desain 3D, karena program ini gratis, semua orang bisa Man-*download*-nya langsung tanpa harus bayar, akan tetapi *software* Google SketchUp mempunyai dua Versi, pertama Google SketchUp versi biasa dan versi Pro, untuk versi Pro, kita harus membayar lisensinya terlebih dahulu (Hairi, 2014).

Berikut adalah *tool-tool* yang biasa di gunakan pada sketcup.



Gambar 2.7 *Tool* SketcUp

Tabel 2.2 Keterangan *Tool* SketcUp

Keterangan Gambar 2.7 Tool SketchUp		
No	Nama	Fungsi
1	<i>Make Component</i>	Untuk Menggabungkan Objek Satu Kesatuan.
2	<i>Paint Bucket</i>	Untuk Mewarnai Atau Menyisipkan Material Pada Objek.
3	<i>Eraser</i>	Untuk Menghapus Gambar Atau <i>Material</i> .
4	<i>Rectangle</i>	Untuk Menggambar Objek Berbentuk Kotak
5	<i>Line</i>	Untuk Menggambar Garis Lurus.
6	<i>Circle</i>	Untuk Menggambar Objek Bulat.
7	<i>Arc</i>	Untuk Menggambara Setengah Lingkaran.
8	<i>Polygon</i>	Untuk Menggambar Objek Segi Banyak
9	<i>Freehand</i>	Untuk Menggambar Bebas.
10	<i>Move</i>	Untuk Memindahkan Objek.
11	<i>Push/Pull</i>	Untuk Mendorong Atau Mengubah Objek Menjadi 3 Dimensi.
12	<i>Rotate</i>	Untuk Memutar Objek.
13	<i>Follow Me</i>	Untuk Mendorong Objek Yang Disesuaikan.
14	<i>Scale</i>	Untuk Mengubah Ukuran Besar Kecil Objek Yang Di Skala Kan.
15	<i>Offset</i>	Menduplikasi Garis Objek Yang Disesuaikan.
16	<i>Tape Measure Tool</i>	Untuk Mengukur.
17	<i>Dimension</i>	Untuk Memberi Dimensi Pada Objek.
18	<i>Protractor</i>	Untuk Ukur Sudut Miring Konstruksi Dan Menciptakan Entitas Line.
19	<i>Text Tool</i>	Untuk Menyisipkan Tulisan.
20	<i>Axes</i>	Memindahkan Atau Reorientasi Sumbu Menggambar.
21	<i>3D Text</i>	Untuk Membuat Teks 3 Dimensi.

22	<i>Orbit</i>	Untuk Memutar Pandangan Objek
23	<i>Pan</i>	Untuk Memindahkan Pandangan Objek Secara Vertikal Dan Horizontal
24	<i>Zoom</i>	Untuk Memperbesar Atau Memperkecil Pandangan Objek.
25	<i>Zoom Extents</i>	Untuk Memperbesar Objek Satu Layar.
26	<i>Previous</i>	<i>Undo Preview</i>
27	<i>Next</i>	<i>Redo Preview</i>
28	<i>Position Kamera</i>	Posisi Kamera (Pandangan Anda) Pada Ketinggian Tertentu Atau Memeriksa Objek Saling Berhadapan Atau Berjalan Melalui Model.
29	<i>Look Around</i>	<i>Pivot</i> Kamera (Pandangan Anda) Dari Titik <i>Stasioner</i> .
30	<i>Walk</i>	Berjalanlah Melalui (Tur) Model.
31	<i>Section Plane</i>	Untuk Pemotongan Bagian Efek Memungkinkan Anda Untuk Melihat Geometri Dalam Model.

2.2.13. Adobe Photoshop

Adobe Photoshop atau biasa disebut Photoshop adalah perangkat lunak editor buatan Adobe System yang dikhususkan untuk pengeditan foto / gambar dan pembuatan efek. Versi ke delapan software ini disebut dengan nama Photoshop CS (Creative Suite), versi sembilan disebut Adobe Photoshop CS2, versi sepuluh disebut Adobe Photoshop CS3, dan seterusnya, hingga sekarang Adobe System telah meluncurkan Photoshop CC (Creative Cloud) penyempurnaan dari CS (Creative Suite) (Husen, 2017).

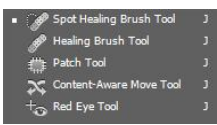


Photoshop tersedia untuk Microsoft Windows, Mac OS X, dan Mac OS; versi 9 ke atas juga dapat digunakan oleh sistem operasi lain seperti Linux dengan bantuan perangkat lunak tertentu seperti Cross Over (Husen, 2017).


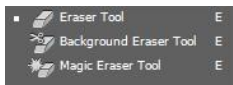
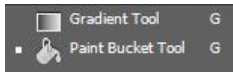
Berikut adalah beberapa fungsi *tool* yang ada di dalam adobe photoshop:




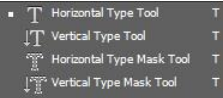
Tabel 2.3 Keterangan *Tool* Adobe Photoshop

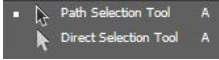
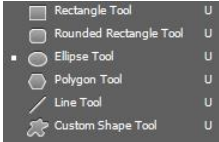
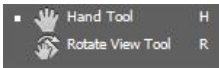
No	Gambar	Nama	Fungsi
1		<i>Move Tool</i>	Untuk Memindahkan Sebuah <i>Object</i> Atau <i>Layer</i> Yang Satu Ke <i>Layer</i> Yang Lainnya.
2		<i>Rectangular Marquee Tool</i>	Untuk Menseleksi Sebuah <i>Object</i> Dalam Bentuk Kotak.
		<i>Elliptical Marquee Tool</i>	Untuk Menseleksi Sebuah <i>Object</i> Dalam Bentuk Lingkaran.
		<i>Single Row Marquee Tool</i>	Untuk Membuat Seleksi Secara <i>Horizontal</i> 1 <i>Pixel</i> .
		<i>Single Column Marquee Tool</i>	Untuk Membuat Seleksi Secara <i>Vertikal</i> 1 <i>Pixel</i> .
3		<i>Lasso Tool</i>	Untuk Membuat Seleksi Secara Bebas.
		<i>Polygonal Lasso Tool</i>	Untuk Membuat Seleksi Berbentuk <i>Poligon</i> .
		<i>Magnetic Lasso Tool</i>	Untuk Membuat Seleksi Secara Otomatis Dengan Mengikuti Warna Yang Serupa.
4		<i>Quick Selection Tool</i>	Untuk Membuat Sebuah Seleksi Secara Otomatis Pada Warna Yang Sama.
		<i>Magic Wand Tool</i>	Untuk Membuat Seleksi Berdasarkan Warna Yang Sama.

5		<i>Crop Tool</i>	Untuk Memotong Sebuah Objek / Gambar.
		<i>Perspective Crop Tool</i>	Untuk Menyempurnakan Gambar Yang Perspective.
		<i>Slice Tool</i>	Untuk Membuat Sebuah Potongan Objek Atau Gambar.
		<i>Slice Select Tool</i>	Untuk Memilih Sebuah Potongan Pada Objek / Gambar.
6		<i>Eyedropper Tool</i>	Untuk Mengambil Contoh Warna Pada Bagian Gambar.
		<i>Color Sampler Tool</i>	Penggunaan Tool Hampir Sama Dengan <i>Eyedropper Tool</i> . Bedanya <i>Color Sampler Tool</i> Dapat Merekam Beberapa Informasi Warna Untuk Di Tampilkan Di Dalam Panel Info. Setelah Itu Data Yang Terekam Dapat Anda Tuangkan Ke Dalam <i>Panel Color</i> Untuk Di Jadikan Warna <i>Foreground</i> Maupun <i>Background</i> Secara <i>Manual</i> .
		<i>Ruler Tool</i>	Untuk Mengukur Objek Pada Kanvas Dan Juga Bisa Digunakan Sebagai Panduan Untuk Memutar Objek Pada Sudut Tertentu.
		<i>Note Tool</i>	Untuk Membuat <i>Notes</i> Sebuah Gambar.

7		<i>Healing Brush Tool</i>	Untuk Memperbaiki Sebuah Objek Atau Gambar Yang Rusak.
		<i>Healing Brush Tool</i>	Untuk Membuat Sebuah Objek Gambar Dengan Sampel Tertentu.
		<i>Patch Tool</i>	Untuk Membuat Sebuah Objek Atau Gambar Dengan Sampel <i>Patern</i> .
		<i>Content-Aware Move Tool</i>	Untuk Memindahkan Objek Suatu Gambar Tanpa Perlu Melakukan <i>Masking</i> , Dan Melakukan Seleksi Secara Detail Pada Objek Suatu Gambar.
		<i>Red Eye Tool</i>	Untuk Menghilangkan Warna Merah Pada Mata Karena Terkena Pantulan Sinar Agar Menjadi Hitam.
8		<i>Brush Tool</i>	Untuk Menggambar/Melukis Gambar Menggunakan Kuas.
		<i>Pencil Tool</i>	Untuk Menggambar/Melukis Gambar Menggunakan Pensil.
		<i>Color Replacement Tool</i>	Untuk Mewarnai Sebuah Gambar Menggunakan Warna Tertentu.
		<i>Mixer Brush Tool</i>	Untuk Membuat Gambar Berdimensi Vektor 3d, Untuk Memperhalus Permukaan Kulit.
9		<i>Clone Stamp Tool</i>	Untuk Membuat Sebuah Gambar Menggunakan Sample

			Gambar Yang Telah Tersedia.
		<i>Pattern Stamp Tool</i>	Untuk Membuat Sebuah Gambar Menggunakan Pola Tertentu.
10		<i>History Brush Tool</i>	Untuk Menghapus Objek/Gambar Menjadi Transparan Mengikuti Warna <i>Background / Layer</i> Di Belakangnya.
		<i>Art History Tool</i>	Untuk Membuat Gambar Secara <i>State History</i> Dari Sebuah Gambar Model Artistik.
11		<i>Eraser Tool</i>	Untuk Menghapus Sebuah Objek Atau Gambar.
		<i>Background Eraser Tool</i>	Untuk Menghapus Sebuah Objek Atau Gambar Yang Akhirnya Menjadi Transparan.
		<i>Magic Eraser Tool</i>	Untuk Menghapus Sebuah Gambar Secara Otomatis Mengikuti Warna Yang Sama Yang Akhirnya Menjadi Transparan.
12		<i>Gradient Tool</i>	Untuk Memberi Warna Sebuah Objek Atau Gambar Secara Gradient.
		<i>Paint Bucket Tool</i>	Untuk Memberi Warna Objek Atau Gambar Yang Dipilih.
13		<i>Blur Tool</i>	Untuk Memberi Efek Kbur Pada Warna Gambar.

		<i>Sharpen Tool</i>	Untuk Memberi Contras Warna Lebih Tajam.
		<i>Smudge Tool</i>	Untuk Menggosok-Gosok Gambar Agar Lebih Halus Dari Hasil Seleksi Yang Tidak Beraturan.
14		<i>Dodge Tool</i>	Untuk Membuat Sebuah Gambar Agar Lebih Jelas/Terang.
		<i>Burn Tool</i>	Untuk Membuat Sebuah Gambar Dengna Efek Lebih Gelap
		<i>Sponge Tool</i>	Untuk Merubah Sebuah Saturasi Pada Area Tertentu Pada Gambar.
15		<i>Pen Tool</i>	Untuk Membuat Sebuah Seleksi Pada Gambar.
		<i>Freeform Pen Tool</i>	Untuk Membuat Sebuah <i>Path</i> Dengan Bentuk Bebas
		<i>Add Anchor Point Tool</i>	Untuk Menambah Sebuah Titik <i>Anchor Point</i> Pada <i>Path</i> .
		<i>Delete Anchor Point Tool</i>	Untuk Menghapus Sebuah Titik <i>Anchor Point</i> Yang Telah Anda Tambahkan Tadi Pada <i>Path</i> .
		<i>Convert Point Tool</i>	Untuk Mengedit Titik <i>Anchor Point</i> Pada <i>Path</i> .
16		<i>Horizontal Type Tool</i>	Untuk Membuat Sebuah Text Secara <i>Horizontal</i> .
		<i>Vertical Type Tool</i>	Untuk Membuat Text/Tulisan Secara <i>Vertical</i> .

		<i>Horizontal Type Mask Tool</i>	Untuk Membuat Seleksi Tulisan/Text Dengan Bentuk Secara <i>Horizontal</i> .
		<i>Vertical Type Mask Tool</i>	Untuk Membuat Seleksi Tulisan/Text Dengan Bentuk Secara <i>Vertical</i> .
17		<i>Path Selection Tool</i>	Untuk Membuat Sebuah <i>Path</i> (Bisa Di Konversi Menjadi Sebuah Seleksi).
		<i>Direct Selection Tool</i>	Untuk Merubah Anchor Point Pada <i>Path</i> .
18		<i>Rectangle Tool</i>	Untuk Membuat Objek Berbentuk Segi 4.
		<i>Rounded Rectangle Tool</i>	Untuk Membuat Objek Bentuk Segi 4 Melengkung.
		<i>Ellipse Tool</i>	Untuk Membuat Objek Dengan Bentuk Lingkaran.
		<i>Polygon Tool</i>	Untuk Membuat Sebuah Objek Dengan Bentuk Segi 3,4, 5 Dan Seterusnya.
		<i>Line Tool</i>	Untuk Membuat Garis Lurus.
		<i>Custom Shape Tool</i>	Untuk Membuat Sebuah Objek Dengan Bentuk Yang Sudah Tersedia Pada Option Bar.
19		<i>Hand Tool</i>	Untuk Memindah/Menggeser Gambar Layaknya Dengan Tangan.
		<i>Rotate View Tool</i>	Untuk Memutar Gambar Hingga 190 ⁰ Atau Sesuai Keinginan.

20		<i>Zoom Tool</i>	Untuk Memperbesar Tampilan Image /Gambar.
21		<i>Switch Foreground And Background Colors</i>	Untuk Mengganti Foreground And Background Ke Depan Atau Ke Belakang.
22		<i>Fungsi Default Foreground And Background Colors</i>	Untuk Mengganti Foreground Menjadi Default.
23		<i>Set Foreground Color And Set Background Color</i>	Untuk Menentukan Warna Pada Seleksi Yang Telah Anda Buat (Pada Warna Foreground). <i>Set Background Color</i> Untuk Menentukan Warna Pada Seleksi Yang Telah Anda Buat (Pada Warna Begroung).
24		<i>Quick Mask</i>	Mempunyai Fungsi Untuk Menjadikan Tampilan Pengeditan Menjadi <i>Quick Mask / RGB</i> .
25		<i>Standard Screen Mode</i>	Untuk Menampilkan Jendela Dengan Mode <i>Standar</i> .
		<i>Full Screen Mode With Menu Bar</i>	Untuk Menampilkan Layar Dengan Mode Layar Penuh Dan <i>Menu Bar</i> .
		<i>Full Screen Mode</i>	Untuk Menampilkan Layar Dengan Mode Layar Penuh Tanpa <i>Menu Bar</i> .

2.2.14. Adobe Fuse

Adobe Fuse CC (Beta) adalah aplikasi desktop untuk membuat karakter manusia 3D yang sangat dapat disesuaikan dalam hitungan menit. Tanpa pengalaman 3D sebelumnya, Anda dapat dengan cepat membuat karakter menggunakan perpustakaan konten 3D berkualitas tinggi, dari wajah dan tubuh hingga pakaian dan tekstur dengan opsi untuk menyesuaikan warna, tekstur, dan bentuk lebih dari 280 atribut. Anda dapat menyimpan karakter Anda ke Creative Cloud Libraries dan membawanya ke ruang kerja 3D di Adobe Photoshop CC untuk membuat komposit foto realistis dengan model 3D (Adobe, 2018).

Kemampuan utamanya adalah kemampuan untuk mengimpor dan mengintegrasikan konten yang dibuat pengguna ke dalam pencipta karakter. Fuse adalah bagian dari rangkaian produk Mixamo dan ditujukan untuk pengembang video game, pengubah video game, dan penggemar 3D (Wikipedia, 2019).

2.2.15. Adobe Audition

Adobe Audition adalah multitrack digital audio recording, editor dan mixer yang udah digunakan dan memiliki berbagai fasilitas pengolahan suara yaitu merekam suara, memperbaiki kualitas suara, menambahkan berbagai efek suara, dan menggabungkan dengan berbagai track suara menjadi satu track, dan menyimpannya dalam berbagai format. Adobe Audition banyak digunakan oleh *musician recording master*, demo cd, produser atau *programing* stasiun radio. Secara umum Adobe Audition memiliki dua lingkungan yaitu *Edit View and Multitrack* (Alvaeyza, 2010).

Edit View sesuai namanya ditujukan terutama untuk menangani editing satu *waveform* saja pada satu saat. Sementara *Multitrack View* dapat menangani beberapa *waveform* sekaligus pada beberapa track. Anda dapat menggunakan kedua lingkungan ini secara bergantian pada tampilan terpisah. (Alvaeyza, 2010)

2.2.16. Mozilla Firefox

Mozilla Firefox merupakan *bwoser* yang dikembangkan oleh Yayasan Mozilla dan ratusan sukarelawan. Firefox dapat dijalankan pada berbagai macam sistem operasi seperti Microsoft Windows, Linux, Mac OS X, dan FreeBSD. Versi stabil dan terbaru Firefox saat ini adalah 62.0, yang dirilis pada 5 September 2018 (Wahyuni, 2011).

Browser merupakan menjelajah dengan mengikuti link di halaman web. *Browser* adalah software/aplikasi/perangkat lunak yang digunakan untuk mengakses/ menampilkan halaman web. Lebih jelasnya *Web browser* adalah aplikasi perangkat lunak digunakan untuk mencari, mengambil dan juga menampilkan informasi di World Wide Web, termasuk halaman Web, gambar, video dan file lainnya. Browser juga mempunyai kemampuan dalam menampilkan kode semantik atau bahasa pemrograman halaman website seperti HTML, Java Script, CSS dan lain-lain menjadi halaman yang mudah dimengerti semua orang (Pengertian, 2013).

2.2.17. Microsoft Visio

Microsoft visio adalah aplikasi yang digunakan untuk merancang suatu model perencanaan, model ini dimanfaatkan untuk kebutuhan *developer* maupun engineering yang didesain untuk berbagai macam kebutuhan. Seperti membuat diagram, *flowchart* (diagram alir), *brainstorm* dan skema jaringan yang dirilis oleh Microsoft Corporation. (Wahyuningrum, 2013).

Berikut adalah Bentuk-Bentuk Diagram pada Microsoft Visio :

1. Diagram Jaringan (*Network Diagram*)

Membuat susunan bentuk jaringan computer atau lainnya sesuai kebutuhan hanya dengan menggabungkan beberapa bentuk kombinasi bentuk dari visio yang sudah tersedia.

2. Flowchart Dasar (*Basic Flowchart*)

Membuat dokumen prosedur, menganalisa proses, menunjukkan alur kerja atau informasi, lagu dan efisiensi biaya, dan banyak lagi.

3. Rencana Denah (*Floor Plan*)

Membuat denah rencana peletakan pintu, jendela, alat alat listrik secara visual untuk bangunan.

4. Bagan struktur Organisasi (*Organization Chart*)

Membentuk bagan kelompok kerja dalam organisasi dan hubungan komunikasi antar departemen.

5. Diagram Basis Data (*Data Base Model Diagram*)

Membentuk model dari data base secara visual dengan penggambaran bentuk skema sesuai aslinya.

6. Diagram Situs Jaringan (*Web site Diagram*)

Menggambar bentuk susunan halaman website secara hierarki dan alur penggunaannya yang dapat dengan mudah di ubah sesuai dengan kebutuhan web saat ini yang dinamis.

7. Diagram Blok (*Block Diagram*)

Melakukan brainstorming, rencana, dan berkomunikasi.

8. Peta Petunjuk (*Directional Maps*)

Peta petunjuk yang dapat menunjukkan arah dengan disertai petunjuk alam seperti pohon, bangunan, sungai dan jalan raya sebagai petunjuknya.

9. Diagram Proses Mesin (*Proccess Enginering Diagram*)

Menunjukkan bagaimana proses dari suatu alat alat dalam perindustrian serta alat-alat yang digunakan seperti mesin, pipa dan penampungnya.

10. Diagram Software (*Software Diagram*)

Membantu mengembangkan tim desain perangkat lunak untuk mengatur tampilan *user interface*.

2.2.18. DirectX

DirectX merupakan sebuah aplikasi yang pada dasarnya untuk menghubungkan sebuah program terhadap hardware. Misalnya untuk mengizinkan game untuk bisa berinteraksi dengan gamepad dan video card.

DirectX merupakan aplikasi yang diciptakan oleh Microsoft yang berisikan Application Programming Interface atau API.

DirectX ini biasanya lebih dikenal di kalangan gamer, karena aplikasi ini merupakan syarat utama untuk menjalankan game di Windows. Jadi, kalau DirectX belum diinstall atau belum diupdate, maka otomatis game tidak bisa jalan dan akan memunculkan notifikasi eror.

DirectX merupakan penghubung interface antara hardware dalam komputer dan sistem operasi, yang merupakan bagian dari Windows API (Application Programming Interface). Dalam penerapannya, ketika developer game ingin memutar file suara, maka mereka hanya perlu memanggil library yang sesuai. Pada waktu game berjalan, ia memanggil DirectX API, yang akan memutar file suara.

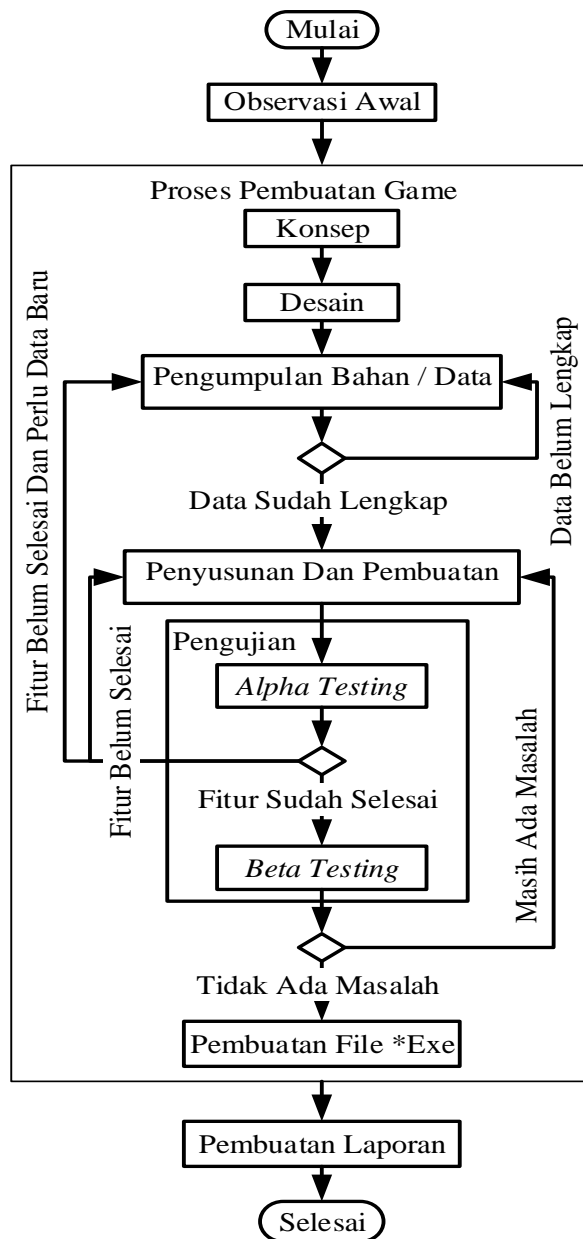
Dengan kata lain developer game tidak perlu tahu jenis sound card apa yang diakan dipakai oleh user, apa yang dapat dilakukannya, atau bagaimana berbicara kepadanya. Microsoft telah menyediakan DirectX, dan pabrikan *sound card* telah menyediakan driver yang berkemampuan DirectX. Jadi *developer* hanya menanyakan suara yang ingin dimainkan, dan ini berlaku untuk semua jenis komputer (Sari, 2019).

BAB III

METODOLOGI PENELITIAN

3.1. Kerangka Pikir/ Skema Penelitian (Diagram)

Kerangka pikir dalam penelitian ini dapat dilihat pada diagram di bawah ini:



Gambar 3.1 Kerangka Pikir

3.2. Penjelasan Skema Penelitian

Berikut adalah penjelasan dari tahapan-tahapan yang ada pada kerangka pikir.

3.2.1. Observasi Awal

Observasi awal dilakukan agar mengetahui apakah perlu dibuatnya sistem baru ini atau tidak, maka penyusun mewawancarai beberapa mahasiswa dimulai dari mahasiswa yang sudah pernah melakukan kerja praktek dan yang lagi mengerjakan kerja praktek. Dari hasil persentasi wawancara kepada 10 mahasiswa didapatkan persentase tertinggi mengenai dibuatnya sistem baru ini, maka penyusun akan membuat sistem baru ini yaitu sebuah *game* simulasi yang didalamnya terdapat prosedur dari kerja praktek mahasiswa Fakultas Teknologi Informasi Universitas Bale Bandung.

3.2.2. Proses Pembuatan *Game*

Dalam proses pembuatan *video game* ini penyusun menggunakan metode MDLC yang dimana ada 6 tahapan antara lain : konsep, desain, pngumpulan bahan, penyusunan dan pembuatan, pengujian dan yang terakhir distribusi, berikut penjabarannya.

1. Konsep

Tahap konsep ini menentukan tujuan dibuatnya aplikasi *game* ini dan siapa penggunanya. Dimana tujuannya adalah untuk menunjukan bagaimana prosedur dari kerja praktek Fakultas Teknologi Informasi Universitas Bale Bandung secara visual kepada mahasiswa-mahasiswa yang ingin mengetahui tentang kerja praktek dan penggunanya yaitu mahasiswa-mahasiswa Fakultas Teknologi Informasi Universitas Bale Bandung.

2. Desain

Agar simulasi *video game* ini sesuai dengan yang dilakukan mahasiswa maka penyusun mewawancarai lagi dua mahasiswa dan satu petugas tata

usaha di Fakultas Teknologi Informasi Universitas Bale Bandung. Untuk hasil wawancara bisa dilihat pada lampiran 2, 3 dan 4.

Dari hasil wawancara dan pengetahuan dari penyusun maka prosedur dari kerja praktek tersebut adalah sebagai berikut :

Tabel 3.1 Prosedur Kerja Praktek

Prosedur	Yang Dilakukan
1	Mendaftarkan Diri Sebagai Peserta Kerja Praktek
2	Membuat Proposal
3	Membayar Mata Kuliah Kerja Praktek
4	Menyerahkan Proposal Beserta Pembayaran
5	Melakukan Survei
6	Menentukan Rencana Kerja Praktek Bersama Permbimbing Lapangan
7	Melakukan Kerja Praktek Selama 6-8 Minggu
8	Membuat Laporan Kerja Praktek
9	Melakukan Proses Pembimbing
10	Menyerahkan Laporan Ke Staf TU
11	Melakukan Sidang Kerja Praktek
12	Dinyatakan Lulus / Tidak

Dari adanya tabel prosedur kerja praktek maka selanjutnya mendesain prosuder kerja praktek tersebut agar sesuai dengan *video game* simulasi ini, tahap ini juga mendesain beberapa karakter untuk *game* baik *main character* maupun *non playable character* dan beberapa objek-objek *game*, beberapa tampilan *user interface game* seperti: *interface inventory*, *interface main menu*, *interface pause menu*, *interface dialog character* dan *interface* lainnya.

3. Pengumpulan Data

Tahap ini adalah tahap dimana mengumpulkan data - data atau bahan - bahan yang akan digunakan / diperlukan dalam pembuatan *game* simulasi ini, seperti: data objek geometri dari Fakultas Teknologi Informasi dan lain lain.

4. Penyusunan Dan Pembuatan

Tahap ini dimulai dari penyusunan gedung-gedung seperti gedung rektorat, bank, fti, rusun dan lain-lain. Setelah itu dimulailah pembuatan modul program (*Scripting*) yang dibutuhkan untuk membuat *video game* ini menarik dan dapat menjabarkan semua prosedur pada matakuliah ini.

5. Pengujian

Tahap ini dimulailah pengujian, pertama adalah *alpha testing*, dimana *alpha testing* dilakukan oleh penyusun sendiri. Setelah melakukan *alphatest* jika masih ada bug maka kembali lagi ke penyusunan dan pembuatan dan jika masih ada bug serta perlu bahan / data baru maka kembali lagi ke pengumpulan bahan / data dan jika tidak ada bug maka lanjut ke *betatest*.

beta testing dimana *beta testing* ini di uji oleh orang yang tidak terkait dengan pembuatan *video game* ini. Setelah melakukan *betatest* jika ada masalah maka kembali lagi ke penyusunan dan pembuatan dan jika tidak maka lanjut ke pembuatan file *.exe.

6. Pembuatan File *Exe

Tahap ini dibuatnya file *.EXE agar bisa di *install* di komputer mahasiswa.

3.2.3. Pembuatan Laporan

Setelah pembuatan *game* selesai maka dibuatlah laporan hasil penelitian.

BAB IV

ANALISIS DAN PERANCANGAN

4.1. Analisis

Analisis yang dilakukan dalam proses penelitian ini meliputi beberapa tahapan yaitu instrumen penelitian, analisis sistem, analisis kebutuhan, hasil analisis dan perancangan.

4.1.1. Instrumen Penelitian

Instrumen Penelitian merupakan alat bantu yang digunakan untuk membantu mengumpulkan data pada sebuah penelitian. Berikut adalah instrument yang digunakan peneliti :

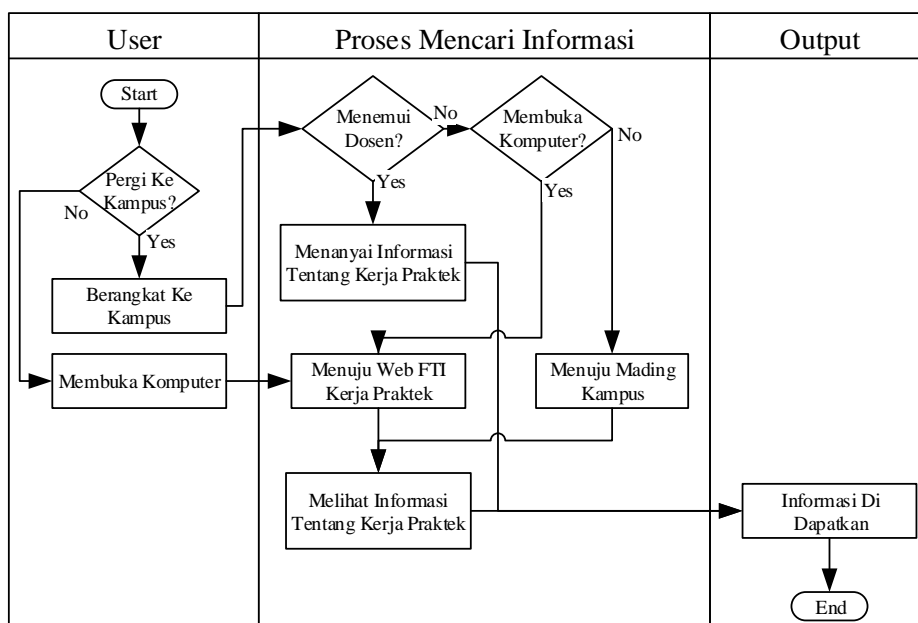
Tabel 4.1 Intrumen Penelitian

No	Tools	Intrumen Penelitian	
1	Perangkat Keras	SmartPhone	
		Spesifikasi	Meuizu M5 (M611H)
			Internal 16GB
			RAM 2GB
			OS
			Android 6.0 (Marshmallow); Flyme 6.3.1.0G
2	Perangkat Lunak	WaveEditor 1.77	Internal
			16GB
3	Lainya	Buku Catatan	
4		Form Wawancara	

4.1.2. Analisis Sistem

Analisis Sistem dilakukan untuk memberikan ide baru dalam pemberian informasi mengenai matakuliah Kerja Praktek Mahasiswa Fakultas Teknologi Informasi Universitas Bale Bandung.

Untuk Sistem yang saat ini berjalan adalah mahasiswa dapat melihat prosedur dari mata kuliah ini dengan cara melihat di mading, melihat dalam website, dan bertanya kepada petugas tata usaha. Dimana semua sistem ini hanya berisikan kata-kata.



Gambar 4.1 Analisis Sitem Yang Berjalan

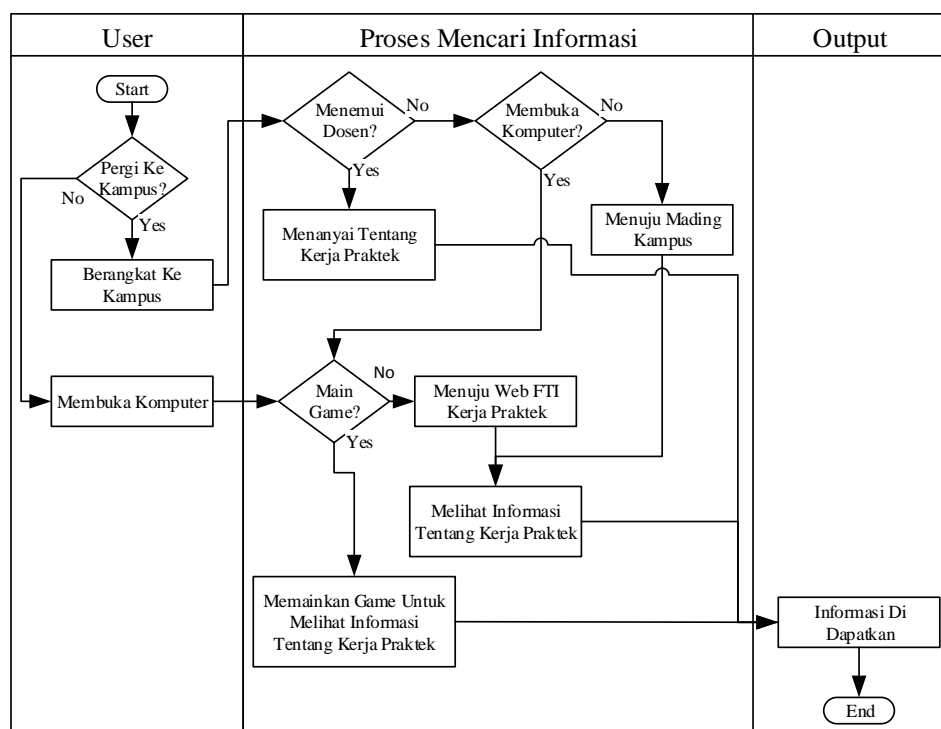
Keterangan Gambar 4.1 :

Pertama user menentukan apakah ingin pergi ke kampus atau tidak, jika tidak maka hanya ada satu pilihan untuk mendapatkan informasi mengenai kerja praktek yaitu dengan membuka komputer dan menuju web fti di bagian kerja praktek.

Jika user pergi ke kampus maka ada 3 pilihan yang bisa dilakukan user untuk mendapatkan informasi yaitu dengan menemui dosen, membuka komputer dan melihat mading. Bila ke dosen maka user harus bertemu dengan

dosen tersebut dan menanyakan informasi mengenai kerja praktek, bila membuka komputer maka user masuk ke web fti bagian kerja praktek dan bila user ingin melihat di mading maka user harus menuju mading tersebut dan melihat informasi mengenai kerja praktek.

Sedangkan untuk sistem usulan penyusun menambahkan sistem baru yaitu sebuah *video game* interaktif, *game* ini juga memiliki kelebihan tersendiri yaitu berbentuk 3D dimana terdapat tata ruang Fakultas Teknologi Informasi Universitas Bale Bandung.



Gambar 4.2 Analisis Sistem Usulan

Keterangan Gambar 4.2 :

Pertama user sama seperti sebelumnya tapi berbeda pada saat membuka komputer, sekarang user memiliki 2 pilihan yaitu dengan membuka web fti atau dengan memainkan game. Pada saat di kampus juga seperti itu ketika user membuka komputer sekarang ada 2 pilihan yaitu dengan membuka web fti atau dengan memainkan game dan untuk yang lain nya sama seperti sistem sebelumnya.

4.1.3.Deskripsi *Game*

Dalam deskripsi *game* berisi pengenalan *game*, *storyline*, *gameplay* dan *character*. Berikut adalah deskripsinya.

1. Pengenalan *Game* Simulasi Kerja Praktek Mahasiswa Fakultas Teknologi Informasi Unveritas Bale Bandung

Game Simulasi Kerja Praktek Mahasiswa Fakultas Teknologi Informasi Unveritas Bale Bandung yang akan dibangun ini bergenre Simulasi yaitu salah satu jenis *game* (*genre*) yang digemari, dimana jenis *game* ini mengutamakan pemain agar menikmati simulasi yang disajikan.

Pada *Game* Simulasi Kerja Praktek Mahasiswa Fakultas Teknologi Informasi Unveritas Bale Bandung ini bahasa komunikasi nantinya akan menggunakan bahasa indonesia dan nantinya pemain diharuskan melakukan beberapa misi agar simulasi ini dapat tercapai dan pemain mendapatkan informasi mengenai mata kuliah ini.

2. Storyline

Jalan cerita *game* simulasi kerja praktek mahasiswa Fakultas Teknik Informatika Universitas Bale Bandung ini diawali dari sang karakter utama / *player* yang sedang di rumah dan ingin mendaftarkan diri sebagai peserta kerja praktek. Disini *player* harus menjalankan misi agar dapat melanjutkan simulasi ini.

Pada saat pertama *player* diharuskan mendaftar online di laptop selanjutnya membuat proposal dan setelah itu melakukan *generate* kode va (*virtual account*). *Player* juga bisa mengambil beberapa makanan yang terdapat di meja makan untuk mengisi energi. Setelah misi selesai maka *player* bisa melanjutkan misi selanjutnya yaitu membayar dengan pergi ke jalan raya.

Sesampainya di jalan raya *player* diharuskan membayar mata kuliah ini dengan cara menggunakan atm yang ada disamping jalanan, setelah selesai maka *player* harus memfotocopy hasil pembayaran tersebut ke tukang foto copy. Dari sini *player* sudah menyelesaikan misi agar bisa lanjut ke tempat selanjutnya yaitu ke gedung Fakultas Teknik Informasi Universitas Bale

Bandung, tapi sebelum masuk ke gedung *player* bisa membeli beberapa minuman atau makanan yang ada di pedagang.

Sesampainya di gedung Fakultas Teknik Informasi Universitas Bale Bandung maka *player* harus menyerahkan proposal berserta fotocopy kwitansi pembayaran mata kuliah ini ke petugas tata usaha atau dalam game ini ke character pa yusuf, selanjutnya *player* bisa melakukan kerja praktek ke tempat yang ada di dalam proposal. Untuk melakukan kerja praktek *player* diharuskan kembali ke jalan raya. Dalam game ini kerja praktek hanya berisikan kata-kata yang penyusun kutip dari web Fakultas Teknik Informasi Universitas Bale Bandung. Setelah itu *player* akan di *transfer* kembali ke rumah.

Sesampainya di rumah *player* di haruskan mencetak laporan, setelah itu *player* di haruskan bimbingan dengan pembimbing di gedung Fakultas Teknik Informasi Universitas Bale Bandung tepatnya pada ruang rapat/sidang. Dalam *game* ini bimbingan di set menjadi 5 kali dimana bimbingan bab 1, bimbingan bab 2, bimbingan bab 3, bimbingan bab 4 dan bimbingan bab 5 setelah itu *player* akan di *transfer* ke rumah lagi.

Setelah sesampainya di rumah *player* di haruskan memfotocopy laporan ini, setelah itu menyerahkan laporan itu ke pada petugas tata usaha atau dalam *game* ini ke character pa yusuf atau pa ojan. Setelah itu *player* akan di *transfer* lagi ke rumah.

Sesampainya di rumah *player* diharuskan melakukan sidang kerja praktek di gedung Fakultas Teknik Informasi Universitas Bale Bandung tepatnya pada ruang rapat/sidang. Setelah sidang maka simulasi ini berakhir.

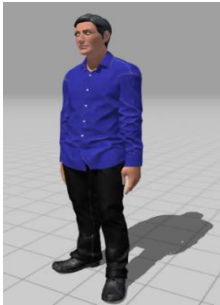



3. *Gameplay*

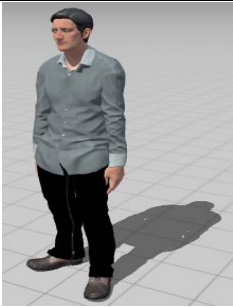


Dalam *Game* Simulasi Kerja Praktek Mahasiswa Fakultas Teknologi Informasi Unveritas Bale Bandung, pemain hanya diharuskan untuk menyelesaikan misi yang ada agar dapat melanjutkan simulasi ini. Dimana setiap misi tidak sulit hanya menggerakkan *player* dan berinterkasi dengan benda sekitar.

4. *Character*

Dalam game ini terdapat beberapa *character*, dimulai dari *character player* dan beberapa *character* pendukung lainnya yang biasa disebut dengan *non playable character*. Berikut adalah penjelasan mengenai setiap *character* yang ada pada *video game* ini.

Tabel 4.2 *Character*

No	Nama	Model Character	Keterangan
1	X		Player Atau Character Utama.
2	Pedagang		Tukang Foto Copy.
3	Pedagang		Penjual Makanan. Menjual : Kue Citron, Roti Bulat, Roti Panjang dan Roti Sandwich.
4	Pedagang		Penjual Minuman. Menjual : Susu, Susu Coklat, Monster Energy dan Pepsi

5	Pa Yusuf		Petugas Di FTI Yang Menerima Proposal, Fotocopy Kwintansi dan Laporan.
6	Pa Ojan		Petugas Di FTI Menerima Penyerahan Laporan.
7	Pembimbing		Salah Satu Pembimbing Yang Ada Di Game Ini.

4.1.4. Analisis Kebutuhan

Analisis kebutuhan untuk aplikasi *game* ini dibagi menjadi lima yaitu kebutuhan perangkat keras (*Hardware*), kebutuhan perangkat lunak (*Software*), kebutuhan sistem, kebutuhan user dan kebutuhan data. Berikut adalah penjelasan mengenai analisis kebutuhan:

1. Analisis Kebutuhan Perangkat Keras (*Hardware*)

Analisis perangkat keras digunakan untuk mengetahui spesifikasi perangkat keras minimum yang dibutuhkan dalam merancang *game* yang dibangun dan spesifikasi minimum bagi pengguna agar dapat menjalankan aplikasi *game* ini dengan lancar dan tidak ada hambatan. Berikut adalah spesifikasi minimum perangkat keras yang dibutuhkan bagi pengembang dan pengguna :

a. Pengembang

Spesifikasi kebutuhan perangkat keras yang dibutuhkan untuk pengembangan *game* yang dibangun adalah sebagai berikut:

Tabel 4.3 Spesifikasi Perangkat Keras (*Hardware*) - Pengembang

No	Nama Perangkat Keras	Spesifikasi
1	Prosesor	Pentium (R) Dual-Core E5700 @ 3.00GHz
2	RAM	DDR3 2x2GB
3	Motherboard	Ampttron G41/G41D3LM LGA775
4	VGA	VGA Card NVIDIA GeForce GTX 640 GDDR5 1GB 256-bit
5	Harddisk	Western Digital 300GB
6	Monitor	Monitor LG 1600 15.6" Resolusi 1366x768
7	Keyboard	Votre KB-2308 USB Standard
8	Mouse	Genius NetScroll 120
9	Speaker	Genius SP-S115
10	Power Suply	PSU GameMax Pure 650W

b. Pengguna

Spesifikasi kebutuhan perangkat keras yang dibutuhkan bagi pengguna untuk menjalankan aplikasi *game* adalah sebagai berikut:

Tabel 4.4 Spesifikasi Perangkat Keras (*Hardware*) – Pengguna

No	Nama Perangkat Keras	Spesifikasi
1	Prosesor	Pentium (R) Dual-Core E5700 @ 3.00GHz
2	RAM	1GB

3	VGA	256MB
4	Harddisk	1GB
5	Monitor	Resolusi 1366x768
6	Keyboard	<i>Standard</i>
7	Mouse	<i>Standard</i>
8	Speaker	<i>Standard</i>

Spesifikasi perangkat keras yang ditentukan untuk pengguna merupakan spesifikasi minimum yang dapat digunakan agar pengguna dapat memainkan *game* yang dibangun dengan baik dan lancar. Semakin rendah spesifikasi perangkat keras yang digunakan oleh pengguna maka proses permainan akan tidak lancar karena beratnya proses grafik dan perhitungan yang dilakukan ketika permainan dijalankan tidak dapat terpenuhi secara maksimal dengan spesifikasi yang rendah.

2. Analisis Kebutuhan Perangkat Lunak (*Software*)

Analisis perangkat lunak digunakan untuk mengetahui spesifikasi perangkat lunak yang dibutuhkan untuk merancang *game* yang akan dibangun. Berikut adalah spesifikasi perangkat lunak yang dibutuhkan bagi pengembang dan pengguna.

a. Pengembang

Spesifikasi kebutuhan perangkat lunak yang dibutuhkan untuk pengembangan *game* yang dibangun adalah sebagai berikut:

Tabel 4.5 Spesifikasi Perangkat Lunak (*Software*) – Pengembang

No	Nama Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 7 Ultimate 64-bit (6.1, Build 7601: Service Pack 1)
2	<i>Tools</i>	Unreal Engine 4.22.3
3		SketchUp 2019

5		Adobe Photoshop CC 2018
6		Adobe Fuse CC 2017
7		Adobe Audition CC 2018
8		Mozilla Firefox 68.0.1 64-bit
9		Microsoft Visio 2013

b. Pengguna

Spesifikasi kebutuhan perangkat lunak yang dibutuhkan bagi pengguna menjalankan aplikasi game adalah sebagai berikut:

Tabel 4.6 Spesifikasi Perangkat Lunak (*Software*) – Pengguna

No	Nama Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 7 dan Versi Selanjutnya
2	Aplikasi	Game Simulasi

3. Analisis Kebutuhan Sistem

Agar *video game* simulasi ini berjalan sesuai prosedur yang ada pada Fakultas Teknologi Informasi Universitas Bale Bandung maka penyusun melakukan wawancara kepada salah satu petugas tata usaha, prosedur hasil wawancara bisa dilihat pada table 3.1.

4. Analisis Kebutuhan *User*

Analisis kebutuhan *user* adalah agar *user* dapat menggunakan aplikasi *game* ini dengan lancar, analisis kebutuhan user adalah sebagai berikut :

Tabel 4.7 Kebutuhan User

No	Kebutuhan User
1	Mengerti Dalam Mengoprasikan Aplikasi
2	Pernah Menginstall Aplikasi Game
3	Dapat Menggunakan Mouse Dan Keyboard

5. Analisis Kebutuhan Data

Setelah melakukan proses analisis kebutuhan sistem, maka analisis selanjutnya adalah analisis kebutuhan data. Analisis ini bertujuan untuk memudahkan dalam perancangan informasi. Adapun analisis kebutuhan data yang diperlukan antara lain :

Tabel 4.8 Kebutuhan Data

No	Data	Jumlah
1	Gambar Bangunan Fakultas Teknologi Informasi Universitas Bale Bandung	4
2	Gambar Bangunan Rektorat Universitas Bale Bandung	3
3	Gambar Bangunan Bank Perkreditan Rakyat Baleendah Rahayu	2
4	Denah Lingkungan Universitas Bale Bandung	1
5	Tata Letak Ruangan Fakultas Teknologi Informasi Universitas Bale Bandung	8
6	<i>Sound Effect</i>	5

Deskripsi Tabel 4.4 Kebutuhan Data:

1. Gambar tampak depan, belakang, samping kiri dan kanan dari Fakultas Teknologi Informasi Universitas Bale Bandung.
2. Gambar tampak depan, samping kanan dan kiri dari rektorat Universitas Bale Bandung.
3. Gambar tampak depan dan samping kanan di lihat dari depan bangunan dari Bank Perkreditan Rakyat Baleendah Rahayu yang ada di wilayah Universitas Bale Bandung.
4. Tata letak dimana rektorat Universitas Bale Bandung, gedung Fakultas Teknologi Informasi Universitas Bale Bandung, warung-warung, dan bank perkreditan rakyat baleendah rahayu berada.
5. Tata letak dimana ruang kelas a, b, c, sidang/rapat, tamu, tata usaha program studi informatika dan tata usaha program studi sistem informasi.

6. Suara Beep ATM, Suara Printer, Suara Berjalan, Suar Pintu Terbuka Dan Tertutup Dan Suara Musik.

4.1.5. Hasil Analisis

Pada hasil analisis dijelaskan kelayakan dari sistem yang akan dibuat, meliputi kelayakan teknis dan kelayakan teknologi:

1. Kelayakan Teknis

Aplikasi ini hanya bisa di jalankan pada platform windows, dimulai dari windows 7 sampai dengan windows terbaru dan dengan spesifikasi yang minimum di harapkan dapat dipakai di semua komputer.

2. Kelayakan Teknologi

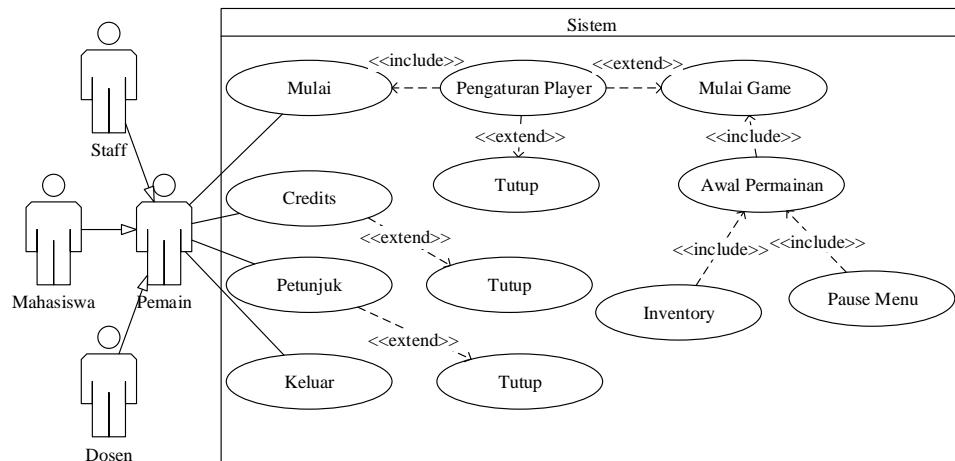
Indrustri *Game* saat ini sudah sangat menjamur. *Game* juga merupakan salah satu teknologi yang sangat berkembang. Menurut (Widiartanto, 2018) potensi industri *game* Indoneisa mencapai Rp 11 Triliun tapi belum tergarap. Dari adanya berbagai survey yang menunjukan betapa bagusnya industri game ini, maka aplikasi ini layak secara teknologi untuk digunakan atau di impelementasikan pada sebuah intitusi.

4.2. Perancangan

Dalam perancangan ini berisi gambaran umum mengenai apa yang akan dikembangkan dalam game, perancangan ini meliputi *Use Case Diagram*, *Activity diagram*, *storyboard*, dan perancanga arsitektur. Berikut adalah perancangan yang akan dibuat.

4.2.1. Use Case Diagram

Penggambaran fungsi *game* berdasarkan interaksi antar aktor dan objek pada sistem digambarkan dengan menggunakan *use case diagram* seperti pada gambar 4.3.



Gambar 4.3 Use Case Diagram

1. Definisi Aktor

Definisi aktor berfungsi menjelaskan peran setiap aktor yang terdapat pada *use case*. Berikut ini adalah definisi aktor.

Tabel 4.9 Definisi Aktor

No	Aktor	Deskripsi
1	Pemain	Pengguna/orang yang memainkan Game ini

2. Definisi Use Case

Definisi *use case* berfungsi untuk menjelaskan tiap-tiap proses dan objek yang terdapat pada *use case*. Berikut ini adalah beberapa definisi *use case*.

Tabel 4.10 Definisi Use Case

No	Use Case	Deskripsi
1	Mulai	Proses Menuju Menu Pengaturan Player
2	Credits	Proses Menuju Tampilan Credits
3	Tutup	Proses Menuju Tampilan Main Menu
4	Petunjuk	Proses Menuju Tampilan Petunjuk
5	Tutup	Proses Menuju Tampilan Main Menu

6	Keluar	Proses Keluar Dari <i>Game</i>
7	Pengaturan Player	Proses Pengaturan Nama, NIM, Tanggal Lahir, Harga Mata Kuliah, Kode Mata Kuliah Dan Tahun Angkatan
8	Awal Permainan	Proses Menuju Tampilan Awal Permainan
9	Tutup	Proses Menuju Tampilan Main Menu
10	Mulai Permainan	Proses Mulai Permainan
11	Inventory	Proses Inventory
12	Pause Menu	Proses Pause Menu

1. Skenario *Use Case*

Skenario *use case* menunjukkan proses apa yang terjadi pada setiap bagian di dalam *use case* tersebut. Berikut ini adalah skenario *use case*.

a. Skenario *Use Case* Mulai

Tabel 4.11 Skenario *Use Case* Mulai

Identifikasi	
Nomor	1
Nama	Mulai
Tujuan	Menampilkan Menu Pengaturan <i>Character</i> .
Deskripsi	Proses Untuk Memulai <i>Game</i>
Aktor	Pemain
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
1. Memilih Menu Mulai	
	2. Masuk Ke Menu Pengaturan <i>Character</i> .
Kondisi Akhir	Masuk Ke Tampilan Pengaturan <i>Character</i> .

b. Skenario *Use Case Credits*Tabel 4.12 Skenario *Use Case Credits*

Identifikasi	
Nomor	2
Nama	Credits
Tujuan	Menampilkan Credits
Deskripsi	Proses Untuk Menuju Tampilan Credits
Aktor	Pemain
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
1. Memilih Menu Credits	
	2. Proses Menampilkan Credits
Kondisi Akhir	Menampilkan Credits

c. Skenario *Use Case Tutup*Tabel 4.13 Skenario *Use Case Tutup*

Identifikasi	
Nomor	3
Nama	Tutup
Tujuan	Menampilkan Main Menu
Deskripsi	Proses Untuk Menampilkan Main Menu
Aktor	Pemain
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
1. Memilih Menu Tutup	

	2. Proses Menampilkan Main Menu
Kondisi Akhir	Menampilkan Main Menu

d. Skenario *Use Case* Petunjuk

Tabel 4.14 Skenario *Use Case* Petunjuk

Identifikasi	
Nomor	4
Nama	Petunjuk
Tujuan	Menampilkan Petunjuk
Deskripsi	Proses Menuju Tampilan Petunjuk
Aktor	Pemain
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
1. Memilih Menu Petunjuk	
	2. Proses Menampilkan Petunjuk
Kondisi Akhir	Menampilkan Petunjuk

e. Skenario *Use Case* Tutup

Tabel 4.15 Skenario *Use Case* Tutup

Identifikasi	
Nomor	5
Nama	Tutup
Tujuan	Menampilkan Main Menu
Deskripsi	Proses Untuk Menampilkan Main Menu
Aktor	Pemain
Skenario Utama	
Kondisi Awal	

Aksi Aktor	Reaksi Sistem
1. Memilih Menu Tutup	
	2. Proses Menampilkan Main Menu
Kondisi Akhir	Menampilkan Main Menu

f. Skenario *Use Case* Mulai

Tabel 4.16 Skenario *Use Case* Keluar

Identifikasi	
Nomor	6
Nama	Keluar
Tujuan	Keluar <i>Game</i>
Deskripsi	Proses Untuk Keluar Dari <i>Game</i>
Aktor	Pemain
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
1. Memilih Menu Keluar	
	2. Proses Keluar Dari <i>Game</i>
Kondisi Akhir	Keluar Dari <i>Game</i>

g. Skenario *Use Case* Pengaturan Player

Tabel 4.17 Skenario *Use Case* Pengaturan Player

Identifikasi	
Nomor	7
Nama	Pengaturan <i>Character</i> .
Tujuan	Menentukan Nama, NIM, Tanggal Lahir, Harga Mata Kuliah Dan Tahun Angkatan
Deskripsi	Proses Pengaturan <i>Character</i>

Aktor	Pemain
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
1. Input Nama, NIM, Tanggal Lahir, Harga Mata Kuliah, Kode Mata Kuliah Dan Tahun Angkatan	
	2. Proses Penginputan Nama, NIM, Tanggal Lahir, Harga Mata Kuliah, Kode Mata Kuliah Dan Tahun Angkatan
Kondisi Akhir	Masuk Ke Tampilan Awal <i>Game</i> .

h. Skenario *Use Case* Mulai Game

Tabel 4.18 Skenario *Use Case* Mulai Game

Identifikasi	
Nomor	8
Nama	Mulai <i>Game</i>
Tujuan	Memulai Permainan
Deskripsi	Proses Untuk Memulai Permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
1. Memilih Pilihan Mulai Game	
	2. Memasukan Variable Yang Telah Di Inputkan Di Skenario Pengaturan Player

	3. Masuk Ke Tampilan Awal <i>Game</i> .
Kondisi Akhir	Masuk Ke Tampilan Awal <i>Game</i> .

i. Skenario *Use Case* Tutup

Tabel 4.19 Skenario *Use Case* Tutup

Identifikasi	
Nomor	9
Nama	Tutup
Tujuan	Menampilkan Main Menu
Deskripsi	Proses Untuk Menuju Ke Main Menu
Aktor	Pemain
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
1. Memilih Menu Tutup	
	2. Masuk Ke Main Menu
Kondisi Akhir	Masuk Ke Tampilan Main Menu

j. Skenario *Use Case* Awal Permainan

Tabel 4.20 Skenario *Use Case* Awal Permainan

Identifikasi	
Nomor	10
Nama	Mulai Permainan
Tujuan	Mulainya Permainan
Deskripsi	Proses Dimana Pemain Memainkan Player
Aktor	Pemain
Skenario Utama	
Kondisi Awal	

Aksi Aktor	Reaksi Sistem
1. Menggerakkan Character Dengan Menekan Tombol Di Key Board	
	2. Jika Menekan Tombol Keyboard “A”, Karakter Bergerak Ke Kiri
	3. Jika Menekan Tombol Keyboard “D”, Karakter Bergerak Ke Kanan
	4. Jika Menekan Tombol Keyboard “S”, Karakter Bergerak/Mundur Ke Belakang
	5. Jika Menekan Tombol Keyboard “W”, Karakter Bergerak/Maju Ke Depan
6. Menekan Tombol “ESC” Di Keyboard	
	7. Menampilkan Menu Pause
8. Menekan Tombol “I” Di Keyboard	
	9. Membuka/Menutup Menu Inventory
10. Menekan Tombol “E” Di Keyboard	
	11. Jika Dekat Dengan Makanan/Minuman Maka Mengambil Makanan/Minuman, Jika Dekat Dengan Pintu Maka Akan Membuka/Menutup Pintu, Jika Dekat Dengan Laptop Maka Akan Membuka Laptop, Jika Dekat Dengan ATM Maka Akan Membuka ATM Dan Jika Dekat

	Dengan Orang Maka Akan Berdialog.
12. Menekan Tombol “Left Ctrl” Di Keyboard	
	13. Player Akan Mengubah Dari Mode Lari Ke Mode Berjalan Atau Sebaliknya.
14. Menekan Tombol “Left Shif” Di Keyboard	
	15. Player Akan Berlari
16. Menggerakan Mouse	
	17. Camera Player Akan Bergerak Sesuai Gerakan Mouse
Kondisi Akhir	Sistem Dalam Permainan

k. Skenario *Use Case* Inventory

Tabel 4.21 Skenario *Use Case* Inventory

Identifikasi	
Nomor	11
Nama	Inventory
Tujuan	Inventory
Deskripsi	Proses Inventory
Aktor	Pemain
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
1. Menekan Tombol “I”	
	2. Membuka/Menutup Inventory

3. Memilih Tombol Gambar Makanan	
	4. Menampilkan Pilihan “Makan Dan Tutup”
5. Memilih Makan	
	6. Menambahkan Energi <i>Character</i>
7. Memilih Tutup	
	8. Kembali Ke Tampilan Awal Inventory
9. Memilih Tombol Gambar Minuman	
	10. Menampilkan Pilihan “Minum Dan Tutup”
11. Memilih Minum	
	12. Menambahkan Energi <i>Character</i>
13. Memilih Tutup	
	14. Kembali Ke Tampilan Awal Inventory
15. Memilih Tombol Gambar Kertas	
	16. Menampilkan Pilihan “Baca Dan Tutup”
17. Memilih Baca	
	18. Menampilkan Isi Bacaan Kertas
19. Memilih Tutup	
	20. Kembali Ke Tampilan Awal Inventory
Kondisi Akhir	Interaksi Inventory

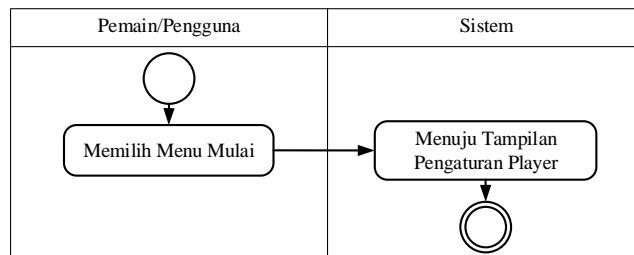
1. Skenario *Use Case* Pause MenuTabel 4.22 Skenario *Use Case* Pause Menu

Identifikasi	
Nomor	12
Nama	Pause Menu
Tujuan	Pause Menu
Deskripsi	Proses Pause Menu
Aktor	Pemain
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
1. Menekan Tombol “ESC” Di Keyboard	
	2. Membuka/Menutup Pause Menu
3. Memilih Button Lanjutkan	
	4. Melanjutkan Permainan
5. Memilih Button Kembali Ke Main Menu	
	6. Proses Menuju Main Menu
7. Memilih Button Keluar	
	8. Proses Keluar Game
Kondisi Akhir	Interaksi Pause Menu

4.1.1. Activity Diagram

Diagram *Activity* menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti pada sistem yang sedang dirancang. Bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Berikut adalah *activity* diagram berdasarkan skenario *use case*.

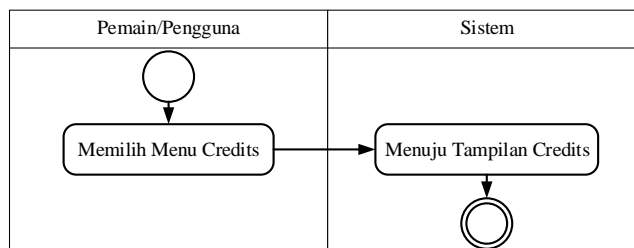
1. Diagram *Activity* Mulai



Gambar 4.4 Diagram *Activity* Mulai

Aktifitas ini pemain menekan tombol mulai yang ada pada tampilan main menu yang akan dilanjutkan ke tampilan pengaturan player.

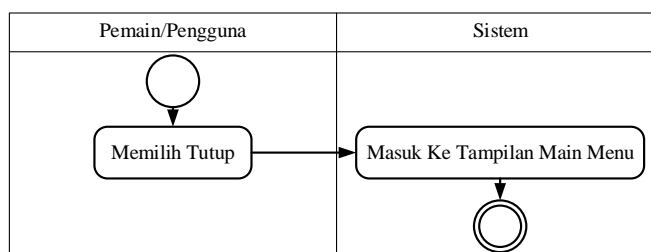
2. Diagram *Activity* Credits



Gambar 4.5 Diagram *Activity* Credits

Aktifitas ini pemain menekan tombol credits yang ada pada tampilan main menu yang akan dilanjutkan ke tampilan credits.

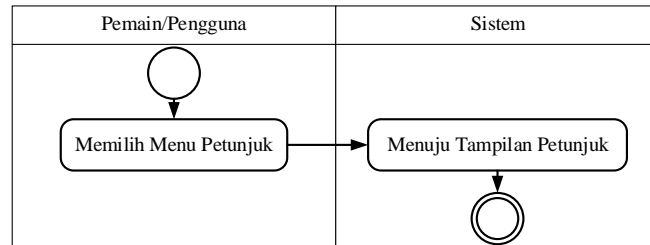
3. Diagram *Activity* Tutup



Gambar 4.6 Diagram *Activity* Tutup Credits

Aktifitas ini pemain menekan tombol tutup yang ada pada tampilan credits yang akan dilanjutkan ke tampilan main menu.

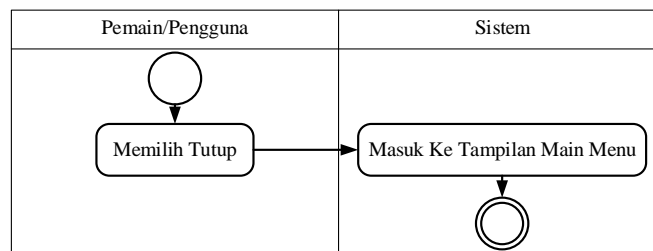
4. Diagram *Activity* Petunjuk



Gambar 4.7 Diagram *Activity* Petunjuk

Aktifitas ini pemain menekan tombol petunjuk yang ada pada tampilan main menu yang akan dilanjutkan ke tampilan petunjuk.

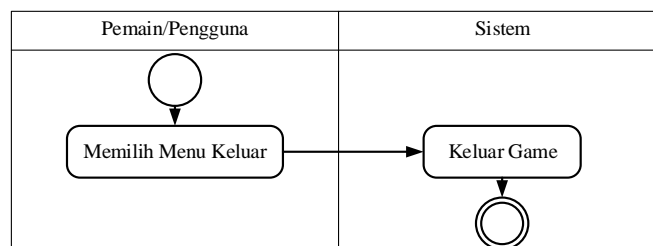
5. Diagram *Activity* Tutup



Gambar 4.8 Diagram *Activity* Tutup Petunjuk

Aktifitas ini pemain menekan tombol tutup yang ada pada tampilan petunjuk, yang akan dilanjutkan ke tampilan main menu.

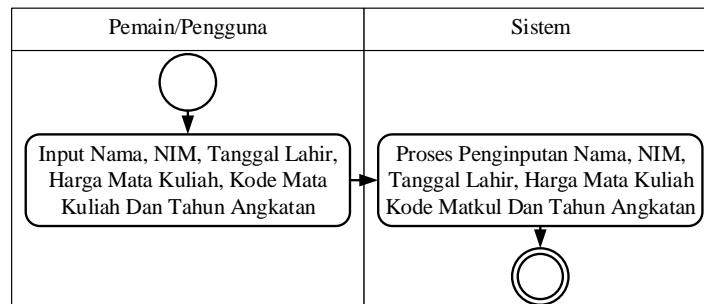
6. Diagram *Activity* Keluar



Gambar 4.9 Diagram *Activity* Keluar Game

Aktifitas ini pemain menekan tombol keluar yang dimana akan melakukan proses keluar dari dalam game.

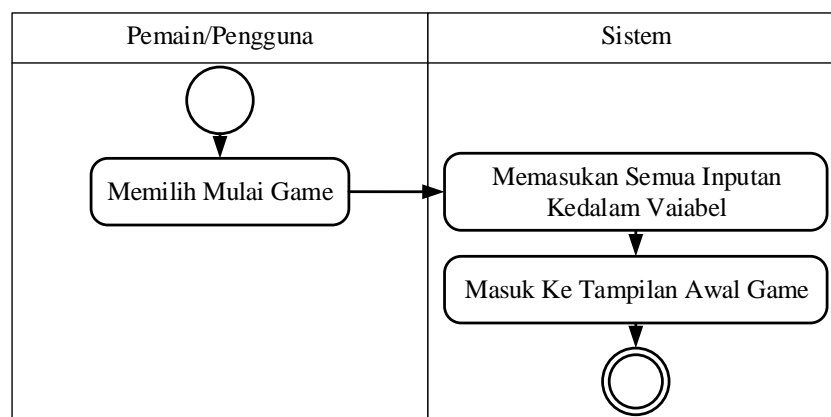
7. Diagram *Activity* Pengaturan Player



Gambar 4.10 Diagram *Activity* Pengaturan Player

Aktifitas ini pemain diharuskan melakukan proses penginputan nama, nim, tanggal lahir, harga mata kuliah, kode mata kuliah dan tahun angkatan untuk bisa melanjutkan ke awal permainan.

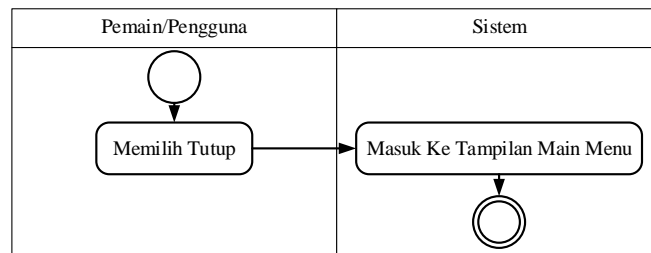
8. Diagram *Activity* Mulai Game



Gambar 4.11 Diagram *Activity* Mulai Game

Aktifitas ini pemain menekan tombol mulai game. Dimana semua masukan *user* akan dimasukan kedalam variabel dan permainan akan dilanjutkan ke awal permainan.

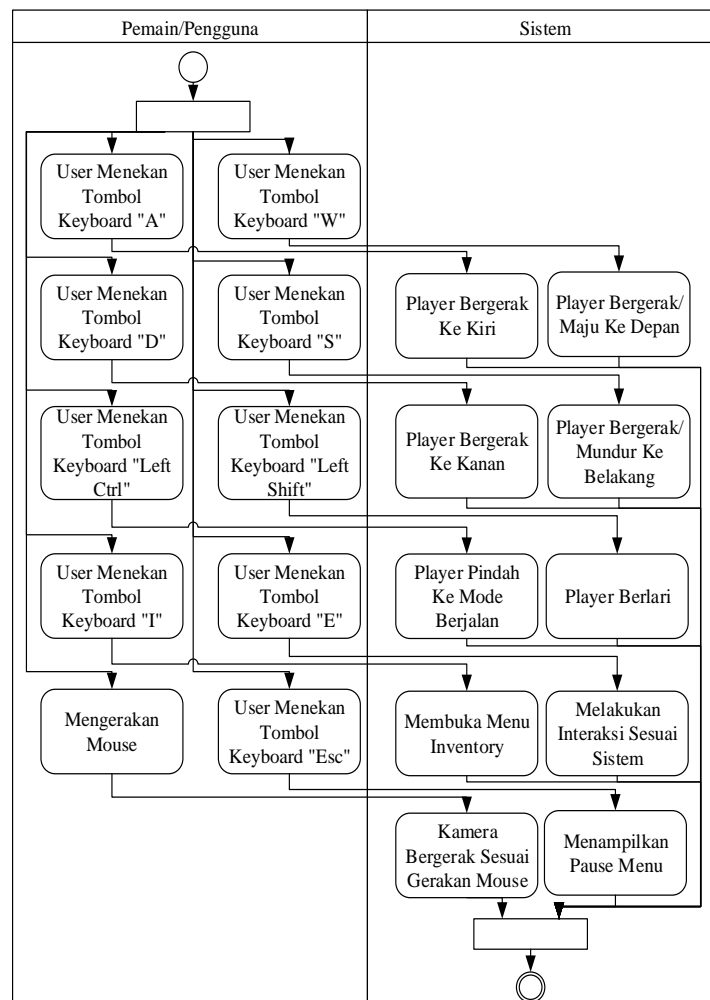
9. Diagram *Activity* Tutup



Gambar 4.12 Diagram *Activity* Tutup Pengaturan Player

Aktifitas ini pemain menekan tombol tutup yang dimana akan dilanjutkan ke tampilan main menu.

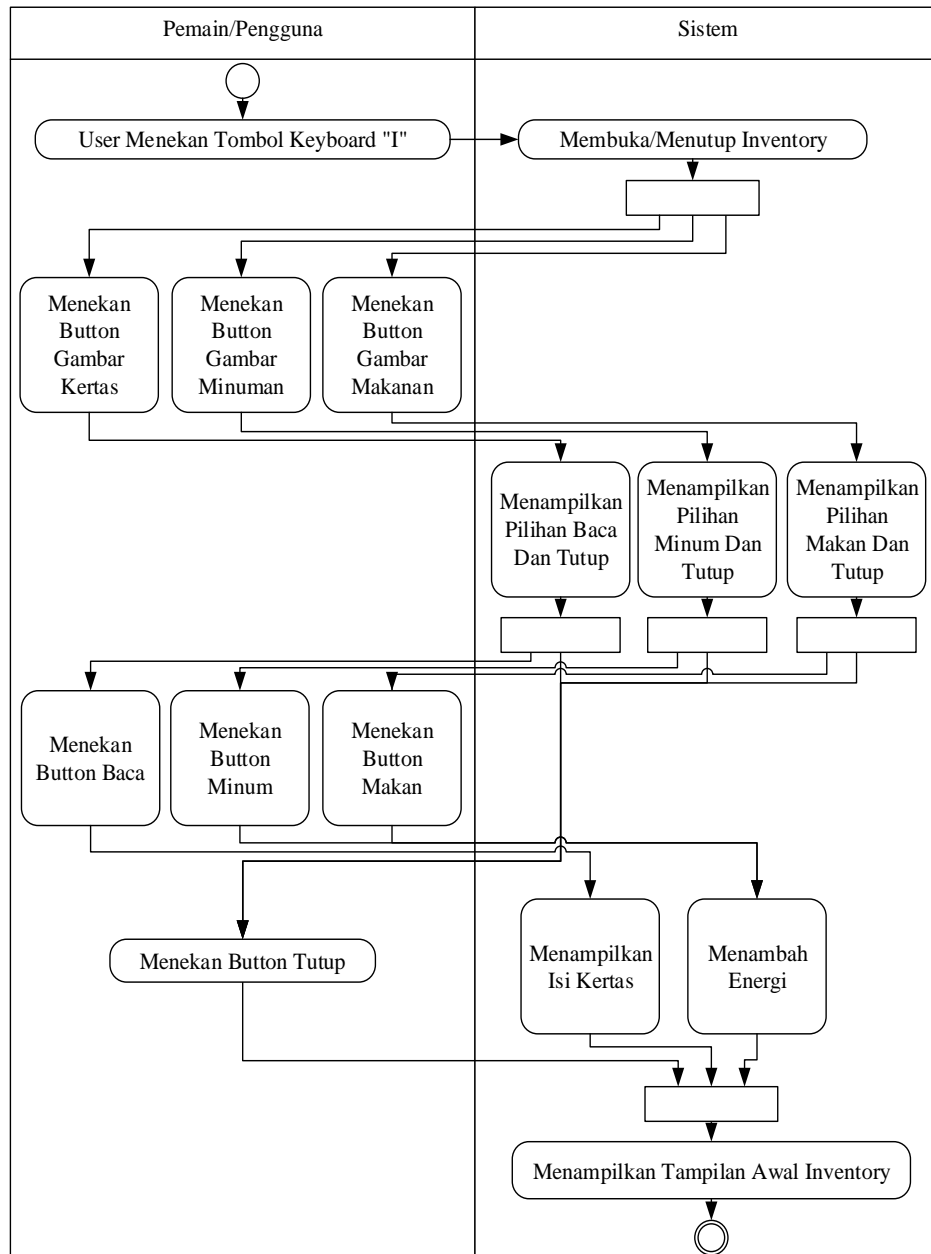
10. Diagram *Activity* Awal Permainan



Gambar 4.13 Diagram *Activity* Awal Permainan

Aktifitas ini pemain bisa memulai menjalankan misi/simulasi dengan menggerakkan player dan berinteraksi dengan objek yang ada.

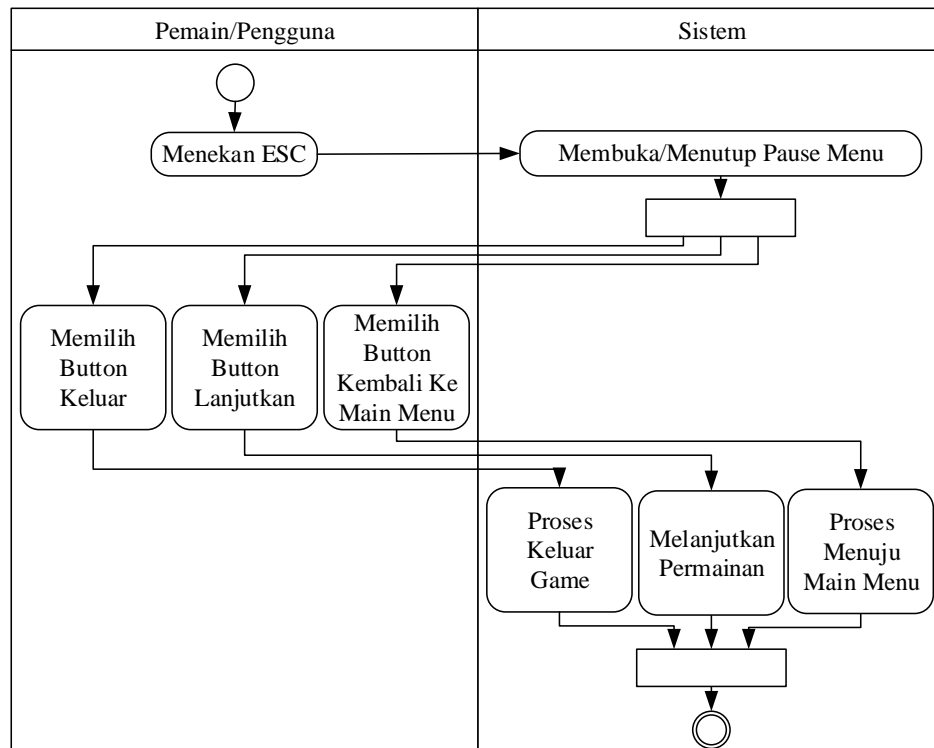
11. Diagram *Activity Inventory*



Gambar 4.14 Diagram *Activity Inventory*

Aktifitas ini pemain bisa berinteraksi dengan sesuatu yang ada pada inventory, misal makan, minum dan membaca.

12. Diagram *Activity* Pause Menu



Gambar 4.15 Diagram *Activity* Pause Menu

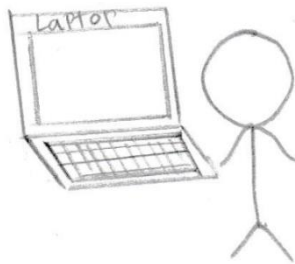
Aktifitas ini pemain menekan button yang ada pada pause menu, setiap button yang ada mempunyai proses interaksinya sendiri sendiri.

4.1.2. Storyboard

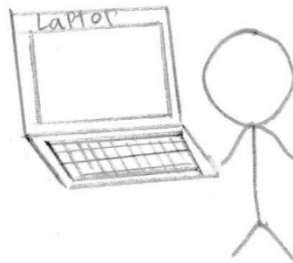
Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengannaskah, dengan storyboard kita dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji, sehingga menghasilkan persepsi yang sama pada ide cerita.

Berikut merupakan storyboard dari Game Simulasi Kerja Praktek Mahasiswa Fakultas Teknologi Informasi Unveritas Bale Bandung yang dikembangkan :

Tabel 4.23 *Storyboard* Misi Pertama

Scene : 1	Shoot :		
Action : Pertama player mendaftar dengan cara membuka laptop di rumah kemudian menekan tombol daftar.			
		Audio : Audio Back Sound	
Item Yang Ada			
No	Jenis Objek	Nama Objek	Keterangan
1	Makanan	Roti Bulat	Untuk Memulihkan Enegi
2		Roti Sandwich	Untuk Memulihkan Enegi
3	Minuman	Monster Enegi	Untuk Memulihkan Enegi
4		Pepsi	Untuk Memulihkan Enegi
5	Bacaan	Proposal	Untuk Daftar
6	Barang	Lemari Baju	Untuk Mengganti Baju
7		Laptop	Untuk Berinteraksi

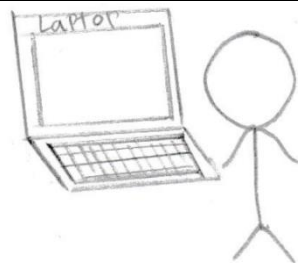
Tabel 4.24 *Storyboard* Misi Kedua

Scene : 2	Shoot :	
Action : Kedua player mencetak proposal dengan cara membuka laptop di rumah kemudian menekan tombol cetak.		
		Audio : Audio Back Sound

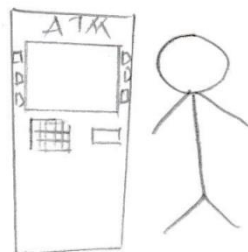
Item Yang Ada			
No	Jenis Objek	Nama Objek	Keterangan
1	Makanan	Roti Bulat	Untuk Memulihkan Enegi
2		Roti Sandwich	Untuk Memulihkan Enegi
3	Minuman	Monster Enegi	Untuk Memulihkan Enegi
4		Pepsi	Untuk Memulihkan Enegi
5	Bacaan	Proposal	Untuk Daftar

6	Barang	Lemari Baju	Untuk Mengganti Baju
7		Laptop	Untuk Berinteraksi

Tabel 4.25 *Storyboard* Misi Ketiga

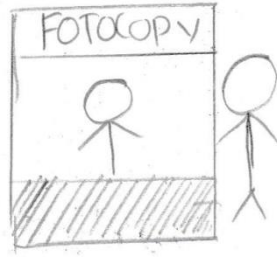
Scene : 3		Shoot :		
Action : Ketiga player membuat kode va (virtual account) dengan cara membuka laptop di rumah kemudian menekan tombol generate kode va.				
				Audio : Audio Back Sound
Item Yang Ada				
No	Jenis Objek	Nama Objek	Keterangan	
1	Makanan	Roti Bulat	Untuk Memulihkan Enegi	
2		Roti Sandwich	Untuk Memulihkan Enegi	
3	Minuman	Monster Enegy	Untuk Memulihkan Enegi	
4		Pepsi	Untuk Memulihkan Enegi	
5	Bacaan	Proposal	Untuk Daftar	
6	Barang	Lemari Baju	Untuk Mengganti Baju	
7		Laptop	Untuk Berinteraksi	

Tabel 4.26 *Storyboard* Misi Keempat

Scene : 4		Shoot :		
Action : Keempat player melakukan pembayaran dengan cara membuka atm di jalan raya kemudian melihat notifikasi untuk melaksanakan pembayaran.				
				Audio : Audio Back Sound
Item Yang Ada				
No	Jenis Objek	Nama Objek	Keterangan	
1	Makanan	Roti Bulat	Untuk Memulihkan Enegi	

2		Roti Sandwich	Untuk Memulihkan Energi
3		Roti Panjang	Untuk Memulihkan Energi
4		Kue Citron	Untuk Memulihkan Energi
5		Susu	Untuk Memulihkan Energi
6	Minuman	Susu Coklat	Untuk Memulihkan Energi
7		Monster Energy	Untuk Memulihkan Energi
8		Pepsi	Untuk Memulihkan Energi
9		Proposal	Untuk Mendaftar
10	Bacaan	Kwitansi	Hasil Pembayaran
11		FC Kwitansi	Untuk Mendaftar
12	NPC	Pedagan Fotocopy	Untuk Fotocopy
13		Pedagan Minuman	Untuk Membeli Minuman
14		Pedagang Makanan	Untuk Membeli Makanan
15	Barang	ATM	Untuk Pembayaran

Tabel 4.27 Storyboard Misi Kelima

Scene : 5		Shoot :			
Action : Kelima player melakukan fotocopy pembayaran dengan cara ke tukang fotocopy di jalan raya kemudian berdialog dengan tukang fotocopy.					Audio : Audio Back Sound
Item Yang Ada					
No	Jenis Objek	Nama Objek	Keterangan		
1	Makanan	Roti Bulat	Untuk Memulihkan Enegi		
2		Roti Sandwich	Untuk Memulihkan Enegi		
3		Roti Panjang	Untuk Memulihkan Enegi		
4		Kue Citron	Untuk Memulihkan Enegi		
5	Minuman	Susu	Untuk Memulihkan Enegi		
6		Susu Coklat	Untuk Memulihkan Enegi		
7		Monster Energy	Untuk Memulihkan Enegi		

8		Pepsi	Untuk Memulihkan Energi
9	Bacaan	Proposal	Untuk Mendaftar
10		Kwitansi	Hasil Pembayaran
11		FC Kwitansi	Untuk Mendaftar
12	NPC	Pedagan Fotocopy	Untuk Fotocopy
13		Pedagan Minuman	Untuk Membeli Minuman
14		Pedagang Makanan	Untuk Membeli Makanan
15	Barang	ATM	Untuk Pembayaran

Tabel 4.28 *Storyboard* Misi Keenam

Scene : 6	Shoot :	
Action : Keenam player melakukan penyerahan proposal dan fotocopy kwitansi di gedung fti, kemudian berdialog dengan petugas tata usaha.		
		Audio : Audio Back Sound

Item Yang Ada			
No	Jenis Objek	Nama Objek	Keterangan
1	Makanan	Roti Bulat	Untuk Memulihkan Energi
2		Roti Sandwich	Untuk Memulihkan Energi
3		Roti Panjang	Untuk Memulihkan Energi
4		Kue Citron	Untuk Memulihkan Energi
5	Minuman	Susu	Untuk Memulihkan Energi
6		Susu Coklat	Untuk Memulihkan Energi
7		Monster Energy	Untuk Memulihkan Energi
8		Pepsi	Untuk Memulihkan Energi
9	Bacaan	Proposal	Untuk Mendaftar
10		Kwitansi	Hasil Pembayaran
11		FC Kwitansi	Untuk Mendaftar

Tabel 4.29 Storyboard Misi Ketujuh

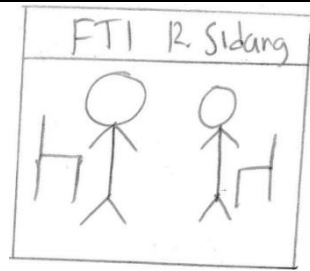
Scene : 7		Shoot :		<div>KERJA PRAKTEK</div>
Action : Ketujuh player melakukan kerja praktek. Dengan cara kembali ke jalan raya. Dalam game hanya kata kata yang muncul.				
Audio : Audio Back Sound				
Item Yang Ada				
No	Jenis Objek	Nama Objek	Keterangan	
1	Makanan	Roti Bulat	Untuk Memulihkan Enegi	
2		Roti Sandwich	Untuk Memulihkan Enegi	
3		Roti Panjang	Untuk Memulihkan Enegi	
4		Kue Citron	Untuk Memulihkan Enegi	
5	Minuman	Susu	Untuk Memulihkan Enegi	
6		Susu Coklat	Untuk Memulihkan Enegi	
7		Monster Energy	Untuk Memulihkan Enegi	
8		Pepsi	Untuk Memulihkan Enegi	
9	Bacaan	Proposal	Untuk Mendaftar	
10		Kwitansi	Hasil Pembayaran	
11		FC Kwitansi	Untuk Mendaftar	

Tabel 4.30 Storyboard Misi Kedelapan

Scene : 8		Shoot :		
Action : Kedelapan player mencetak laporan. dengan cara membuka laptop di rumah kemudian menekan tombol cetak.				
				Audio : Audio Back Sound
Item Yang Ada				
No	Jenis Objek	Nama Objek		Keterangan

1	Makanan	Roti Bulat	Untuk Memulihkan Enegi
2		Roti Sandwich	Untuk Memulihkan Enegi
3		Roti Panjang	Untuk Memulihkan Enegi
4		Kue Citron	Untuk Memulihkan Enegi
5	Minuman	Susu	Untuk Memulihkan Enegi
6		Susu Coklat	Untuk Memulihkan Enegi
7		Monster Energy	Untuk Memulihkan Enegi
8		Pepsi	Untuk Memulihkan Enegi
9	Bacaan	Proposal	Untuk Mendaftar
10		Kwitansi	Hasil Pembayaran
11		FC Kwitansi	Untuk Mendaftar
12		Laporan	Untuk Bimbingan Dan Sidang
13	Barang	Laptop	Untuk Berinteraksi

Tabel 4.31 *Storyboard* Misi Kesembilan

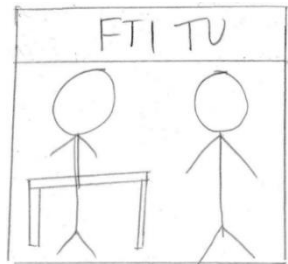
Scene : 9		Shoot :		
Action : Kesembilan player melakukan bimbingan. dengan cara berdialog dengan pembimbing di gedung fti. Dalam game bimbingan di haruskan 5 kali. Ada bimbingan Bab 1, 2, 3, 4 dan 5.				
Item Yang Ada				
No	Jenis Objek	Nama Objek	Keterangan	
1	Makanan	Roti Bulat	Untuk Memulihkan Enegi	
2		Roti Sandwich	Untuk Memulihkan Enegi	
3		Roti Panjang	Untuk Memulihkan Enegi	
4		Kue Citron	Untuk Memulihkan Enegi	
5	Minuman	Susu	Untuk Memulihkan Enegi	
6		Susu Coklat	Untuk Memulihkan Enegi	
7		Monster Energy	Untuk Memulihkan Enegi	
8		Pepsi	Untuk Memulihkan Enegi	

9	Bacaan	Proposal	Untuk Mendaftar
10		Kwitansi	Hasil Pembayaran
11		FC Kwitansi	Untuk Mendaftar
12		Laporan	Untuk Bimbingan Dan Sidang

Tabel 4.32 *Storyboard* Misi Kesepuluh

Scene : 10		Shoot :	
Action : Kesepuluh player melakukan fotocopy laporan dengan cara ke tukang fotocopy di jalan raya kemudian berdialog dengan tukang fotocopy.			
		Audio : Audio Back Sound	
Item Yang Ada			
No	Jenis Objek	Nama Objek	Keterangan
1	Makanan	Roti Bulat	Untuk Memulihkan Enegi
2		Roti Sandwich	Untuk Memulihkan Enegi
3		Roti Panjang	Untuk Memulihkan Enegi
4		Kue Citron	Untuk Memulihkan Enegi
5	Minuman	Susu	Untuk Memulihkan Enegi
6		Susu Coklat	Untuk Memulihkan Enegi
7		Monster Energy	Untuk Memulihkan Enegi
8		Pepsi	Untuk Memulihkan Enegi
9	Bacaan	Proposal	Untuk Mendaftar
10		Kwitansi	Hasil Pembayaran
11		FC Kwitansi	Untuk Mendaftar
12	NPC	Pedagan Fotocopy	Untuk Fotocopy
13		Pedagan Minuman	Untuk Membeli Minuman
14		Pedagang Makanan	Untuk Membeli Makanan
15	Barang	ATM	Untuk Pembayaran

Tabel 4.33 *Storyboard* Misi Kesebelas

Scene : 11		Shoot :		
Action : Kesebelas player melakukan penyerahan laporan untuk sidang. dengan cara berdialog dengan petugas tata usaha di gedung fti.				
Audio : Audio Back Sound				
Item Yang Ada				
No	Jenis Objek	Nama Objek	Keterangan	
1	Makanan	Roti Bulat	Untuk Memulihkan Enegi	
2		Roti Sandwich	Untuk Memulihkan Enegi	
3		Roti Panjang	Untuk Memulihkan Enegi	
4		Kue Citron	Untuk Memulihkan Enegi	
5	Minuman	Susu	Untuk Memulihkan Enegi	
6		Susu Coklat	Untuk Memulihkan Enegi	
7		Monster Energy	Untuk Memulihkan Enegi	
8		Pepsi	Untuk Memulihkan Enegi	
9	Bacaan	Proposal	Untuk Mendaftar	
10		Kwitansi	Hasil Pembayaran	
11		FC Kwitansi	Untuk Mendaftar	

Tabel 4.34 *Storyboard* Misi Keduabelas

Scene : 12		Shoot :		
Action : Keduabelas player melakukan sidang. dengan cara berdialog dengan pembimbing di gedung fti.				
				Audio : Audio Back Sound
Item Yang Ada				
No	Jenis Objek	Nama Objek	Keterangan	
1	Makanan	Roti Bulat	Untuk Memulihkan Enegi	

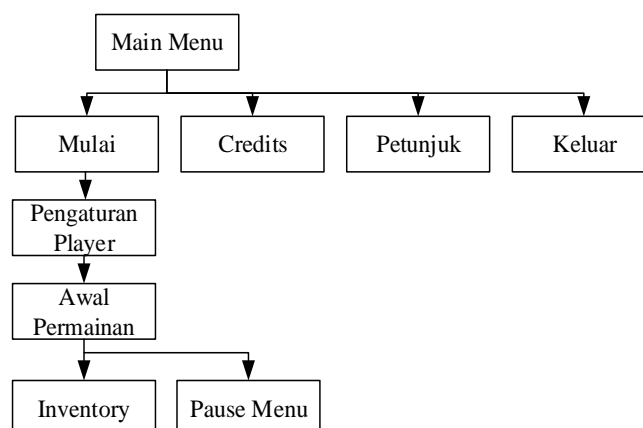
2		Roti Sandwich	Untuk Memulihkan Enegi
3		Roti Panjang	Untuk Memulihkan Enegi
4		Kue Citron	Untuk Memulihkan Enegi
5		Susu	Untuk Memulihkan Enegi
6	Minuman	Susu Coklat	Untuk Memulihkan Enegi
7		Monster Energy	Untuk Memulihkan Enegi
8		Pepsi	Untuk Memulihkan Enegi
9		Proposal	Untuk Mendaftar
10	Bacaan	Kwitansi	Hasil Pembayaran
11		FC Kwitansi	Untuk Mendaftar

4.1.3. Perancangan Arsitektur

Perancangan arsitektur yang dibuat meliputi beberapa perancangan diantaranya struktur menu, perancangan antarmuka dan jaringan semantik. Berikut deskripsinya.

1. Struktur Menu

Perancangan struktur menu dibuat agar pengguna/*player* dapat dengan mudah memilih menu-menu dari aplikasi *game* simulasi yang sedang berjalan. Struktur menu pada *game* simulasi ini dapat dilihat pada gambar berikut :



Gambar 4.16 Struktur Menu

2. Perancangan Antarmuka

Perancangan antarmuka dibutuhkan agar pada proses pembuatan aplikasi *game* ini menjadi lebih mudah dan teratur dan juga dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun, berikut ini adalah perancangan antarmuka dari aplikasi *game* yang akan dibangun.

a. Antarmuka Main Menu

Berikut adalah rancangan dari antarmuka menu utama :

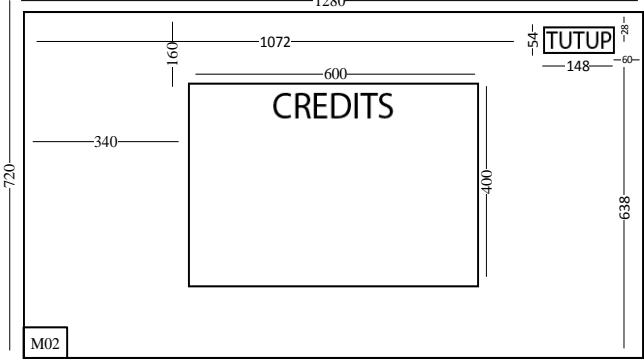
Tabel 4.35 Antarmuka Menu Utama

Satuan : Pixel			
Keterangan <ul style="list-style-type: none"> • Klik Mulai Untuk Masuk Ke M04 • Klik Credits Untuk Masuk Ke M02 • Klik Petunjuk Untuk Masuk Ke M03 • Klik Keluar Untuk Keluar Dari Permainan 			
No	Jenis Objek	Nama Objek	Keterangan
1	Tombol	Mulai	Masuk Ke Menu Pengaturan Character
2		Credits	Melihat Credits
3		Petunjuk	Melihat Petunjuk Permainan
4		Keluar	Untuk Menutup Aplikasi

b. Antarmuka Credits

Berikut adalah rancangan dari antarmuka credit :

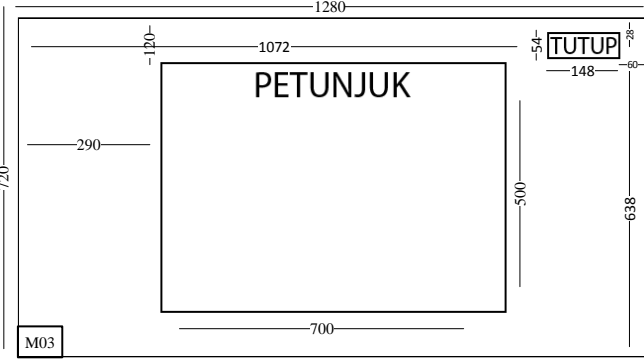
Tabel 4.36 Antarmuka Credits

			
Satuan : Pixel			
Keterangan			
<ul style="list-style-type: none"> Klik Tutup Untuk Masuk Ke M01 			
No	Jenis Objek	Nama Objek	Keterangan
1	Tombol	Tutup	Untuk Masuk Ke Main Menu

c. Antarmuka Petunjuk

Berikut adalah rancangan dari antarmuka petunjuk :

Tabel 4.37 Antarmuka Petunjuk

			
Satuan : Pixel			
Keterangan			
<ul style="list-style-type: none"> Klik Tutup Untuk Masuk Ke M01 			

No	Jenis Objek	Nama Objek	Keterangan
1	Tombol	Tutup	Untuk Masuk Ke Main Menu

d. Antarmuka Pengaturan Character

Berikut adalah rancangan dari antarmuka pengaturan character :

Tabel 4.38 Antarmuka Pengaturan Character

Satuan : Pixel

Keterangan

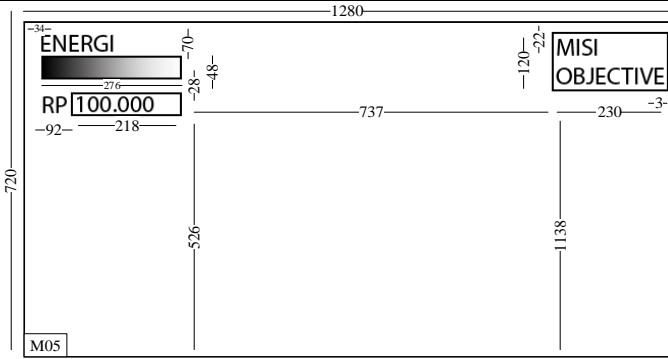
- Klik Mulai Game Untuk Masuk Ke M05
- Klik Tutup Untuk Masuk Ke M01

No	Jenis Objek	Nama Objek	Keterangan
1	Tombol	Mulai Game	Untuk Masuk Ke Awal Permainan
2		Tutup	Untuk Masuk Ke Main Menu
3	Editable	Nama	Masukan Berupa Huruf
4	Text	NIM	Masukan Berupa Huruf
5		Tanggal Lahir	Masukan Berupa Huruf
6		Harga Matkul	Masukan Berupa Number
7		Kode Matkul	Masukan Berupa Huruf
8		Tahun Ajaran	Masukan Berupa Number

e. Antarmuka Permainan

Berikut adalah rancangan dari antarmuka awal permainan :

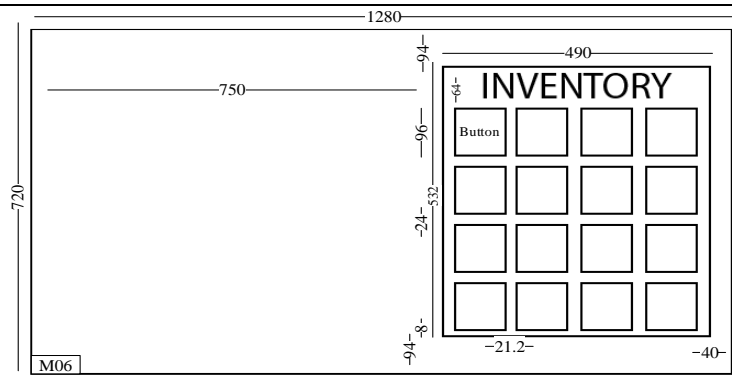
Tabel 4.39 Antarmuka Awal Permainan

	
Satuan : Pixel	
Keterangan <ul style="list-style-type: none"> • Tekan Tombol Keyboard I Untuk Ke M06 • Tekan Tombol Keyboard Esc Untuk Ke M07 	

f. Antarmuka Inventory

Berikut adalah rancangan dari antarmuka inventory:

Tabel 4.40 Antarmuka Inventory

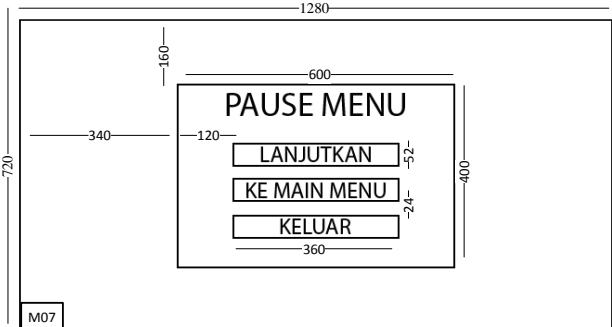
	
Satuan : Pixel	
Keterangan <ul style="list-style-type: none"> • Klik Button Untuk Melakukan Interaksi • Tekan I Di Keyboard Untuk Masuk Ke M05 	

No	Jenis Objek	Nama Objek	Keterangan
1	Button	Button Item	Untuk Melakukan Interaksi Terhadap Inventory

g. Antarmuka Pause Menu

Berikut adalah rancangan dari antarmuka pause menu :

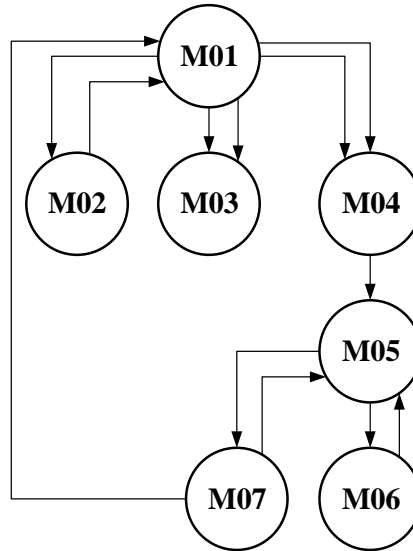
Tabel 4.41 Antarmuka Pause Menu

			
Satuan : Pixel			
Keterangan <ul style="list-style-type: none"> • Klik Lanjutkan Untuk Masuk Ke M05 • Klik Ke Main Menu Untuk Masuk Ke M01 • Klik Tutup Untuk Menutup Aplikasi 			
No	Jenis Objek	Nama Objek	Keterangan
1	Button	Lanjutkan	Untuk Melanjutkan Permainan
2		Ke Main Menu	Untuk Masuk Ke Main Menu
3		Keluar	Untuk Menutup Aplikasi <i>Video Game</i>

3. Jaringan Semantik

Struktur aplikasi ini dibuat secara modular, yaitu program dipecah menjadi modul-modul kecil yang mudah dibuat, mudah dites, dan mudah dimodifikasi. Dalam pembuatan *video game* ini menggunakan alat bantu

bagian *structural* yang disebut juga jaringan semantik dapat dilihat pada gambar 4.17 yang ada pada halaman selanjutnya.



Gambar 4.17 Jaringan Semantik

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1. Implementasi

Tahap implementasi merupakan tahap penciptaan perangkat lunak kedalam bahasa pemrograman yang dimengerti oleh computer, serta merupakan tahap kelanjutan dari kegiatan perancangan sistem.

5.1.1. Implementasi Perangkat Keras

Perangkat keras yang akan digunakan untuk menjalankan aplikasi *game* Simulasi Kerja Praktek Mahasiswa Fakultas Teknologi Informasi Universitas Bale Bandung ini tidak harus yang berspesifikasi tinggi. Adapun spesifikasi perangkat keras minimum yang digunakan saat implementasi dan pengujian aplikasi *game* ini dapat dilihat pada table 5.1

Tabel 5.1 Spesifikasi Perangkat Keras

No	Nama Perangkat Keras	Spesifikasi
1	Prosesor	Pentium (R) Dual-Core E5700 @ 3.00GHz
2	RAM	DDR3 2x2GB
3	Motherboard	Ampttron G41/G41D3LM LGA775
4	VGA	VGA Card NVIDIA GeForce GTX 640 GDDR5 1GB 256-bit
5	Harddisk	Western Digital 300GB
6	Monitor	Monitor LG 1600 15.6" Resolusi 1366x768
7	Keyboard	Votre KB-2308 USB Standard
8	Mouse	Genius NetScroll 120
9	Speaker	Genius SP-S115
10	Power Suply	PSU GameMax Pure 650W

5.1.2. Implementasi Perangkat Lunak

Spesifikasi perangkat lunak minimum yang dibutuhkan untuk menjalankan aplikasi *game* ini yaitu sistem operasi windows 7 SP2 dan DirectX12 untuk mendukung performa grafik untuk menjalankan aplikasi.

5.1.3. Implementasi Aplikasi

Aplikasi *game* ini merupakan merupakan sebuah *game* ber-genre simulasi yang dimana pemain harus menyelesaikan misi untuk dapat mengetahui simulasi berikutnya.

5.1.4. Implementasi Antarmuka

Adapun implementasi antarmuka pada aplikasi *game* Survive From Death adalah sebagai berikut :

1. Antarmuka Main Menu

Antarmuka menu utama merupakan tampilan menu utama dari aplikasi *game* ini. Terdapat 4 menu dapat dilihat pada gambar berikut:



Gambar 5.1 Tampilan Antarmuka Main Menu

Disini pengguna dapat mengklik mulai untuk masuk kedalam pengaturan player, mengklik credits untuk melihat credits, mengklik petunjuk untuk melihat petunjuk dan mengklik keluar untuk keluar dari *game*.

2. Antarmuka Credits

Antarmuka credits merupakan tampilan dari siapa pembuat game ini.

Dapat dilihat pada gambar berikut:



Gambar 5.2 Tampilan Antarmuka Credits

Disini pengguna dapat mengklik tutup agar dapat kembali ke main menu.

3. Antarmuka Petunjuk

Antarmuka petunjuk berisi tampilan tentang petunjuk permainan dalam game. Dapat dilihat pada gambar berikut:



Gambar 5.3 Tampilan Antarmuka Petunjuk

Disini pengguna dapat mengklik tutup agar dapat kembali ke main menu.

4. Antarmuka Pengaturan Player

Antarmuka pengaturan player merupakan tampilan dimana pengguna harus meng-*input*-kan profil pengguna. Dapat dilihat pada gambar berikut:



Gambar 5.4 Tampilan Antarmuka Pengaturan Player

Disini pengguna dapat mengklik mulai game jika sudah mengisi semua inputan dan mengklik tutup agar dapat kembali ke main menu.

5. Antarmuka Awal Permainan

Antarmuka awal permainan merupakan tampilan awal simulasi dimulai, dimana pengguna dapat menggerakkan player. Dapat dilihat pada gambar berikut:

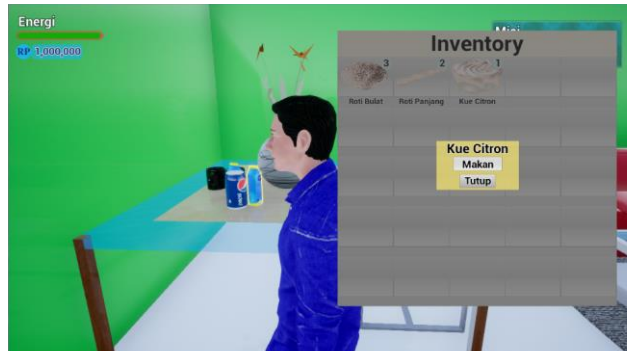


Gambar 5.5 Tampilan Antarmuka Awal Permainan

Disini pengguna bisa menekan tombol yang ada di dalam petunjuk, misalnya : menekan tombol I untuk ke inventory atau Esc untuk masuk ke pause menu.

6. Antarmuka Inventory

Antarmuka inventory merupakan tampilan penyimpanan barang-barang. Dapat dilihat pada gambar tersebut:



Gambar 5.6 Tampilan Antarmuka Inventory

Disini tampilannya sedang melakukan interaksi terhadap kue citron dengan mengklik button kue citron yang ada di inventory.

7. Antarmuka Pause Menu

Antarmuka pause menu merupakan tampilan pause dimana pengguna ingin beristirahat dalam memainkan *game*. Dapat dilihat pada gambar berikut:



Gambar 5.7 Tampilan Antarmuka Pause Menu

Disini pengguna dapat mengklik lanjutkan untuk melanjutkan permainan, mengklik ke main menu untuk menuju ke main menu dan keluar game untuk keluar dari aplikasi game ini.

5.2. Pengujian

Pengujian dilakukan bertujuan untuk untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak. Pengujian dibagi menjadi dua ada pengujian *alpha* dan *beta*. Berikut adalah tahapan-tahapan dari pengujian.

5.2.1. Pengujian Alpha

Pada tahap alpha, pengujian dilakukan untuk memastikan aplikasi game ini dapat berjalan sesuai dengan kebutuhan dan tujuan yang diharapkan.

1. Pengujian Menu Main Menu

Pengujian menu main menu merupakan proses dimana pengguna melakukan interaksi terhadap menu-menu yang ada pada tampilan main menu. Adapun hasil dari proses pengujian dapat dilihat pada table berikut:

Tabel 5.2 Pengujian Menu Main Menu

Kasus dan Hasil Uji			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Klik Menu Mulai Game	Sistem Melakukan Proses Untuk Menampilkan Pengaturan Player	Sitem Menampilkan Pengaturan Player	[√] Sukses
			[] Gagal
Klik Menu Credits	Sistem Melakukan Proses Untuk Menampilkan Credits	Sitem Menampilkan Credits	[√] Sukses
			[] Gagal
Klik Menu Petunjuk	Sistem Melakukan Proses Untuk Menampilkan Petunjuk	Sitem Menampilkan Petunjuk	[√] Sukses
			[] Gagal

Klik Menu Keluar	Sistem Melakukan Proses Untuk Menutup Aplikasi	Sistem Menutup Aplikasi Game	[√] Sukses
			[] Gagal

2. Pengujian Menu Credits

Pengujian menu credits merupakan proses dimana pengguna melakukan interaksi terhadap menu-menu yang ada pada tampilan credits. Adapun hasil dari proses pengujian dapat dilihat pada table berikut:

Tabel 5.3 Pengujian Menu Credits

Kasus dan Hasil Uji			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Klik Menu Credits	Sistem Melakukan Proses Untuk Menampilkan Halaman Credits	Sitem Menampilkan Halaman Credits	[√] Sukses
			[] Gagal
Klik Menu Tutup	Sistem Melakukan Proses Untuk Menampilkan Halaman Main Menu	Sitem Menampilkan Halaman Main Menu	[√] Sukses
			[] Gagal

3. Pengujian Menu Petunjuk

Pengujian petunjuk merupakan proses dimana pengguna melakukan interaksi terhadap menu-menu yang ada pada tampilan petunjuk. Adapun hasil dari proses pengujian dapat dilihat pada tabel 5.4 yang ada pada halaman selanjutnya:

Tabel 5.4 Pengujian Menu Petunjuk

Kasus dan Hasil Uji			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Klik Menu Petunjuk	Sistem Melakukan Proses Untuk Menampilkan Petunjuk	Sitem Menampilkan Petunjuk	[√] Sukses
			[] Gagal
Klik Menu Tutup	Sistem Melakukan Proses Untuk Menampilkan Main Menu	Sitem Menampilkan n Main Menu	[√] Sukses
			[] Gagal

4. Pengujian Menu Pengaturan Player

Pengujian menu pengaturan player merupakan proses dimana pengguna melakukan interaksi terhadap menu-menu yang ada pada tampilan pengaturan dan proses penginputan dari pengguna. Adapun hasil dari proses pengujian dapat dilihat pada table berikut:

Tabel 5.5 Pengujian Menu Pengaturan Player

Kasus dan Hasil Uji			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Input Nama	Sistem Menerima Proses Masukan	Sistem Menerima Proses Masukan	[√] Sukses
			[] Gagal
Input NIM	Sistem Menerima Proses Masukan	Sistem Menerima Proses Masukan	[√] Sukses
			[] Gagal
Input Tanggal Lahir	Sistem Menerima Proses Masukan	Sistem Menerima Proses Masukan	[√] Sukses
			[] Gagal

Input Harga Matakuliah	Sistem Menerima Proses Masukan	Sistem Menerima Proses Masukan	[√] Sukses
			[] Gagal
Input Kode Matakuliah	Sistem Menerima Proses Masukan	Sistem Menerima Proses Masukan	[√] Sukses
			[] Gagal
Input Nama	Sistem Menerima Proses Masukan	Sistem Menerima Proses Masukan	[√] Sukses
			[] Gagal
Klik Menu Mulai Game	Sistem Melakukan Proses Untuk Menuju Ke Awal Permainan	Sitem Menuju Awal Permainan	[√] Sukses
			[] Gagal
Klik Menu Tutup	Sistem Melakukan Proses Untuk Menampilkan Main Menu	Sitem Menampilkan Main Menu	[√] Sukses
			[] Gagal

5. Pengujian Menu Inventory

Pengujian menu inventory merupakan proses dimana pengguna melakukakn interaksi pada menu inventory. Adapun hasil dari proses pengujian dapat dilihat pada table berikut:

Tabel 5.6 Pengujian Menu Inventory

Kasus dan Hasil Uji			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Menekan Tombol I	Sistem Menutup Tampilan Inventory	Sistem Menutup Tampilan Inventory	[√] Sukses
			[] Gagal

Klik Button Bergambar Makanan	Sistem Menampilkan Menu Makan dan Tutup	Sistem Menampilkan Menu Makan dan Tutup	[√] Sukses
			[] Gagal
Klik Button Bergambar Minuman	Sistem Menampilkan Menu Minum dan Tutup	Sistem Menampilkan Menu Minum dan Tutup	[√] Sukses
			[] Gagal
Klik Button Bergambar Kertas	Sistem Menampilkan Menu Baca dan Tutup	Sistem Menampilkan Menu Baca dan Tutup	[√] Sukses
			[] Gagal

Tabel 5.7 Pengujian Menu Inventory – Klik Gambar Makanan

Klik Button Bergambar Makanan			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Klik Menu Makan	Sistem Menambah Energi Player	Sistem Menambah Energi Player	[√] Sukses
			[] Gagal
Klik Menu Tutup	Sistem Kembali Ke Halaman Inventory	Sistem Kembali Ke Halaman Inventory	[√] Sukses
			[] Gagal

Tabel 5.8 Pengujian Menu Inventory – Klik Gambar Minuman

Klik Button Bergambar Minuman			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Klik Menu Minum	Sistem Menambah Energi Player	Sistem Menambah Energi Player	[√] Sukses
			[] Gagal

Klik Menu Tutup	Sistem Kembali Ke Halaman Inventory	Sistem Kembali Ke Halaman Inventory	[√] Sukses
			[] Gagal

Tabel 5.9 Pengujian Menu Inventory – Klik Gambar Kertas

Klik Button Bergambar Kertas			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Klik Menu Baca	Sistem Menampilkan Halaman Kertas.	Sistem Menampilkan Halaman Kertas.	[√] Sukses
			[] Gagal
Klik Menu Tutup	Sistem Kembali Ke Halaman Inventory	Sistem Kembali Ke Halaman Inventory	[√] Sukses
			[] Gagal

6. Pengujian Menu Pause Menu

Pengujian menu pause menu merupakan proses dimana pengguna melakukan interaksi pada menu-menu yang ada pada tampilan pause menu. Adapun hasil dari proses pengujian dapat dilihat pada table berikut:

Tabel 5.10 Pengujian Menu Pause Menu

Kasus dan Hasil Uji			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Klik Menu Lanjutkan	Sistem Melanjutkan Permainan	Sistem Melanjutkan Permainan	[√] Sukses
			[] Gagal

Klik Menu Ke Main Menu	Sistem Menampilkan Main Menu	Sistem Menampilkan Main Menu	[√] Sukses
			[] Gagal
Klik Menu Keluar	Sistem Melakukan Proses Menutup Aplikasi	Sistem Menutup Aplikasi Game	[√] Sukses
			[] Gagal

7. Pengujian Kontrol Pemain

Pengujian kontrol pemain dimaksudkan untuk menguji semua input kontrol dari mouse maupun keyboard yang telah disediakan. Adapun hasil dari proses pengujian dapat dilihat pada table berikut:

Tabel 5.11 Pengujian Kontrol Pemain

Kasus dan Hasil Uji			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Tekan Tombol Left Ctrl	Animasi Player Berubah Dari Lari Ke Berjalan Dan Sebaliknya	Animasi Player Berubah Dari Lari Ke Berjalan Dan Sebaliknya	[√] Sukses
			[] Gagal
Tekan Tombol Left	Animasi Player Berubah Dari Lari Ke Sprint Atau Dari Berjalan Ke Sprint.	Animasi Player Berubah Dari Lari Ke Sprint Atau Dari Berjalan Ke Sprint.	[√] Sukses
			[] Gagal
Tekan Tombol W	Player Bergerak Ke Depan	Player Bergerak Ke Depan	[√] Sukses
			[] Gagal
Tekan Tombol S	Player Bergerak Ke Belakang	Player Bergerak Ke Belakang	[√] Sukses
			[] Gagal

Tekan Tombol A	Player Bergerak Ke Kanan	Player Bergerak Ke Kanan	[√] Sukses
			[] Gagal
Tekan Tombol D	Player Bergerak Ke Kiri	Player Bergerak Ke Kiri	[√] Sukses
			[] Gagal
Tekan Tombol E	Sistem Melakukan Interaksi Sesuai Proses	Sistem Melakukan Interaksi Sesuai Proses	[√] Sukses
			[] Gagal
Tekan Tombol Esc	Sistem Menampilkan Halaman Pause Menu	Sistem Menampilkan Halaman Pause Menu	[√] Sukses
			[] Gagal
Tekan Tombol I	Sistem Menampilkan Halaman Inventory	Sistem Menampilkan Halaman Inventory	[√] Sukses
			[] Gagal
Menggerakkan Mouse	Camera Player Bergerak	Camera Player Bergerak	[√] Sukses
			[] Gagal

Tabel 5.12 Pengujian Kontrol Pemain – Interaksi Player

Interaksi Player			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Dialog	Berdialog Sesuai Dialog Yang Sudah Ditentukan	Berdialog Sesuai Dialog Yang Sudah Ditentukan	[√] Sukses
			[] Gagal
Mengambil Barang	Sistem Memasukan Barang Tersebut Kedalam Inventory	Sistem Memasukan Barang Tersebut Kedalam Inventory	[√] Sukses
			[] Gagal

Membuka Pintu	Sistem Membuka Pintu	Sistem Membuka Pintu	[√] Sukses
			[] Gagal
Menutup Pintu	Sistem Menutup Pintu	Sistem Menutup Pintu	[√] Sukses
			[] Gagal
Membuka Laptop	Sistem Menampilkan Halaman Laptop	Sistem Menampilkan Halaman Laptop	[√] Sukses
			[] Gagal
Membuka Lemari	Sistem Menampilkan Halaman Lemari	Sistem Menampilkan Halaman Lemari	[√] Sukses
			[] Gagal
Membuka ATM	Sistem Menampilkan Halaman ATM	Sistem Menampilkan Halaman ATM	[√] Sukses
			[] Gagal
Transfer Player	Sistem Mentrasnfer Player Ke Level Lain	Sistem Mentrasnfer Player Ke Level Lain	[√] Sukses
			[] Gagal

8. Pengujian Lainnya

Pengujian lainnya adalah pengujian yang tidak terkait pada kontrol pemain. Adapun hasil dari proses pengujian dapat dilihat pada table berikut:

Tabel 5.13 Pengujian Lainnya

Kasus dan Hasil Uji			
Masukan	Keluaran Yang Diharapkan	Pengamatan	Kesimpulan
Player Mendekati Objek Yang Bisa Melakukan Interaksi	Sisi Objek Tersebut Bercahaya Muncul Huruf E Untuk Interaksi	Sisi Objek Tersebut Bercahaya Muncul Huruf E Untuk Interaksi	[√] Sukses
			[] Gagal

Energi Berkurang	Setiap Detik Energi Berkurang	Setiap Detik Energi Berkurang	[√] Sukses
			[] Gagal
Pengurangan Uang	Uang Berkurang Jika Membeli Makanan / Minuman	Uang Berkurang Jika Membeli Makanan / Minuman	[√] Sukses
			[] Gagal

9. Kesimpulan Pengujian *Alpha*

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah berjalan sesuai dengan yang diharapkan dan secara fungsional sudah dapat menghasilkan keluaran yang diharapkan.

5.2.2. Pengujian *Beta*

Pengujian *beta* hampir sama dengan *alpha* tapi yang melakukan pengujian dalam *beta test* ini adalah orang yang tidak terkait dalam pembuatan *game* ini.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dengan adanya aplikasi *game* ini maka mahasiswa dapat mengetahui dengan jelas kemana dia akan pergi ketika melaksanakan mata kuliah kerja praktek Fakultas Teknologi Informasi Universitas Bale Bandung.

6.2. Saran

Pada aplikasi *game* ini tidak ada *non playable character* (NPC) yang interaktif, maka dari itu bila ingin melanjutkan aplikasi *game* ini maka saya sarankan menambahkan npc yang interaktif, perbanyak environment dan buatlah *game* ini lebih interaktif.

DAFTAR PUSTAKA

- Dianty, M. A. (2016, Desember 14). *Flowmap*. Dipetik Agustus 15, 2019, dari Sistem Informasi Manajemen: <https://sisteminformasimanajemen15089.wordpress.com/2016/12/14/flowmap/>
- Adobe. (2017, Mei 24). Dipetik Agustus 15, 2019, dari Workspace Basics: <https://helpx.adobe.com/beta/fuse/help/workspace-basics.html>
- Adobe. (2018, July 3). Dipetik Agustus 15, 2019, dari Common Question: <https://helpx.adobe.com/beta/fuse/faq.html?promoid=XKMMHJ7Y&mv=other>
- Adobe Audition. (2010, Februari 3). Dipetik Agustus 15, 2019, dari Dinuantz': <https://dinuantz.wordpress.com/tag/belajar-adobe-audition/>
- Aisha, M. (2018, April 14). *Pengertian Multimedia : Jurusan, Jenis, Manfaat, dan Contoh*. Dipetik Mei 28, 2019, dari Jagad.Id: <https://jagad.id/pengertian-definisi-multimedia-jurusan-jenis-manfaat-dan-contoh/>
- Alvaeyza. (2010, Februari 3). *Adobe Audition*. Dipetik Agustus 15, 2019, dari Dinuantz': <https://dinuantz.wordpress.com/category/about-me/>
- Chandra. (2013, Agustus 19). *Wave Editor Adalah Aplikasi Editor Untuk Pemula*. Dipetik Agustus 15, 2019, dari UTekno: <https://utekno.com/wave-editor-adalah-aplikasi-audio-editor-untuk-pemula-1744>
- Epic Game.inc. (2004). *Getting Started With UE4*. Dipetik Agustus 15, 2019, dari Unreal Engine: <https://docs.unrealengine.com/en-US/GettingStarted/index.html>
- Epic Games.Inc. (2004). *Blueprint Visual Scripting*. Dipetik Mei 28, 2019, dari Unreal Engine: <https://docs.unrealengine.com/en-US/Engine/Blueprints/index.html>
- Hairi, R. R. (2014, Juli 12). *Mengenal Nama Dan Fungsi Tool Google Sketchup Untuk Pemula*. Dipetik Agustus 15, 2019, dari Mari Belajar: <http://arsitektur-me.blogspot.com/2014/07/mengenal-nama-dan-fungsi-tool-google.html>
- Harsatput. (2013, Juni 5). *Game Engine Technology (Unreal Engine)*. Dipetik Mei 28, 2019, dari Pengetahuan: <http://harsatput24.blogspot.com/2013/06/game-engine-technology-unreal-engine.html>

- Hirnandin, A. (2018). Video Game Edukatif Terhadap Kemampuan Mengenal Anggota Tubuh Manusia Pada Anak Tunarungu Kelas I. *Jurnal Pendidikan Khusus*, 1-17.
- Husen, M. K. (2017, September 17). *Photohop Dasar*. Dipetik Agustus 15, 2019, dari Irul Design 08: <http://iruldesign08.blogspot.com/2017/09/materi-belajar-photoshop-pemula-lengkap.html>
- Iqram, A. (2013, Februari 11). *FlowMap-Definisi, Simbol dan Penggunaannya*. Dipetik Agustus 15, 2019, dari JS (Iqram Blogger): <http://andiiqram.blogspot.com/2013/02/flow-map-definisi-simbol-dan.html>
- Novaliendry, D. (2013). Aplikasi Game Geografi berbasis Multimedia Interaktif. *Jurnal Teknologi Infromasi & Pendidikan*, 106-118.
- Nugroho, A. (2010). *Rekayasa Perangkat Lunak Berorientasi Objek Dengan Metode USDP*. Yogyakarta.
- P, A. A. (2013, Juni 22). *Penjelasan Tentang Activity Diagram*. Dipetik Juli 26, 2019, dari Sarana Sharing dan Berbagai Ilmu: <https://asrielpangloli.blogspot.com/2013/06/penjelasan-tentang-activity-diagram.html>
- Pengertian, T. (2013, Januari 2). *Pengertian Fungsi dan Contoh Browser*. Dipetik Agustus 15, 2019, dari Temukan Pengertian: <https://www.temukanpengertian.com/2013/01/pengertian-browser.html>
- Pratama, E. (2014, Oktober 22). *Apa Itu 3D Modelling?* Dipetik Mei 28, 2019, dari Edho Pratama: <http://mynameisedho.blogspot.com/2014/10/apa-itu-3d-modeling.html>
- Santoso, M. B. (2012, Juni 6). *Definisi, Peran dan Fungsi Mahasiswa*. Dipetik 07 17, 2019, dari Pamuncar: <http://pamuncar.blogspot.com/2012/06/definisi-peran-dan-fungsi-mahasiswa.html>
- Saputra, C. S., & Samopa, F. (2012). Pengembangan Aplikasi Peta Tiga Dimensi Menggunakan Unreal Engine. *Jurnal Teknik Pomits*, 1-6.
- Saputra, S. E. (2011, Oktober 3). *Pemrograman Berorientasi Obyek*. Dipetik Mei 28, 2019, dari Supriyadi Eka Saputra: <https://cuepricornz.wordpress.com/tutorial/pemrograman-berorientasi-obyek/>
- Sari, N. M. (2019, Maret 30). *Fungsi DirectX, Cara Kerja, Komponen, hingga Versinya yang Perlu Diketahui*. Dipetik Agustus 15, 2019, dari Liputan 6: <https://hot.liputan6.com/read/3929836/fungsi-directx-cara-kerja-komponen-hingga-versinya-yang-perlu-diketahui>
- Simanjuntak, W. (2016, April 18). *Finite State Machine (FSM)*. Dipetik Mei 28, 2019, dari Welly Simanjuntak:

<http://blog.ub.ac.id/wellysimanjuntakblog/2016/04/18/finite-state-machine-fsm/>

Sitompul, D. (2014, Maret 18). *Apakah yang dimaksud dengan Simulasi?* Dipetik Mei 28, 2019, dari Teknik Permodelan Dan Simulasi:

<https://komekstensi.wordpress.com/2014/03/18/apakah-yang-dimaksud-dengan-simulasi-2/>

Susilawati. (2014). Perancangan Game Space Ship Dengan Metode Quad Tree. *Pelita Informatika Budi Darma*, 80.

Wahyuni, S. (2011, Mei 3). *Apa itu Mozilla Firefox?* Dipetik Agustus 15, 2019, dari Internet dan Komputer:

<https://sriwahyuni46.wordpress.com/2011/05/03/apa-itu-mozilla-firefox/>

Wahyuningrum, A. (2013, Juni 27). *Pengenalan Ms. Visio*. Dipetik Agustus 15, 2019, dari Anisa:

<http://blog.ub.ac.id/anisawahyuningrum/2013/06/27/pengenalan-ms-visio/>

Widiartanto, Y. H. (2018, Maret 21). *Potensi Industri Game Indonesia Rp 11 Triliun, Tapi Belum Tergarap*. Dipetik Agustus 15, 2019, dari Kompas.com:

<https://ekonomi.kompas.com/read/2018/03/21/061500826/potensi-industri-game-indonesia-rp-11-triliun-tapi-belum-tergarap>

Wikipedia. (2019, Juli 2). *Adobe Fuse CC*. Dipetik Agustus 15, 2019, dari Wikipedia: https://en.wikipedia.org/wiki/Adobe_Fuse_CC

Wikipedia. (2019, Mei 4). *Model*. Dipetik Mei 28, 2019, dari Wikipedia: <https://id.wikipedia.org/wiki/Model>

LAMPIRAN

Lampiran 1 : Hasil Wawancara – 10 Mahasiswa

No	Pertanyaan	Jawaban	Persentase
1	Apakah Anda Tau Tentang Kerja Praktek?	Ya	60%
		Tidak	40%
2	Apakah Anda Ada Masalah Pada Saat Kerja Praktek?	Ya	50%
		Tidak	50%
3	Apakah Anda Menginginkan Sistem Informasi Baru Mengenai Kerja Praktek?	Ya	70%
		Tidak	30%
4	Apakah Anda Tertarik Tentang Game?	Ya	80%
		Tidak	20%
5	Bila Dibuatkan Apakah Anda Ingin Mencobanya?	Ya	100%
		Tidak	0%

Lampiran 2 : Hasil Wawancara – Mahasiswa Angga Triatna

No	Pertanyaan	Jawab
1	Bagaimana Prosedur Kerja Praktek Dari Mulai Pendaftaran Sampai Dengan Dinyatakan Lulus?	Pertama-Tama Melakukan Pendaftaran Ke Pihak TU FTI UNIBBA Untuk Mengajukan Kerja Praktek. Sambil Menyerahkan Proposal Kerja Praktek Jika Disetujui Maka Mulai Kerja Praktek Di Tempat Yang Sudah Ditentukan Melakukan Beberapa Kegiatan Di Perusahaan Tsb Waktu Pelaksanaan Kerja Praktek Hanya 1 Semester Di Semester 5. Atau Bisa Di Semester 7 Yang Paling Penting Yaitu Pelaporan Karena Itu Merupakan Salah Satu Syarat Lulus Kerja Praktek Jika Pelaporan Sudah Disusun Maka Dilakukan Sidang Kerja Praktek Oleh Penguji Setelah Lulus Sidang, Maka Pelaporan Tadi Di Revisi, Di Cetak Dan Di Cover Soft Maka Kerja Praktek Dinyatakan Lulus

Bandung, 22 Maret 2019

Narasumber


ANGGA TRIATNA

Lampiran 3 : Hasil Wawancara – Petugas TU Yusuf Muharam, S.Kom

No	Pertanyaan	Jawab
1	Bagaimana Tatacara Kerja Praktek Dimulai Dari Pendaftaran Sampai Dengan Dinyatakan Lulus?	<p>Pertama Mahasiswa Mengambil Mata Kuliah Kerja Praktek, Mahasiswa Mendaftarkan Diri Sebagai Peserta Kerja Praktek, Mahasiswa Menyerahkan Proposal Kerja Praktek Tersebut, Mahasiswa, Mahasiswa Melakukan Survei Lokasi, Mahasiswa Menentukan Rencana Kerja Praktek Dengan Pembimbing Lapangan, Mahasiswa Melakukan Kerja Praktek Di Lokasi Tersebut,</p> <p>Mahasiswa Membuat Aplikasi Sederhana, Mahasiswa Membuat Laporan Mengenai Topik Dari Kerja Praktek, Mahasiswa Melakukan Bimbingan Untuk Pembuatan Laporan, Mahasiswa Menyerahkan Laporan, Mahasiswa Melakukan Sidang, Mahasiswa Dinyatakan Lulus</p>

Bandung, 23 Mei 2019

Narasumber


YUSUF MUHARAM, S.Kom

Lampiran 4 : Hasil Wawancara – Mahasiswa Megantari Suhendar

No	Pertanyaan	Jawab
1	Bagaimana Tatacara Kerja Praktek Dari Mulai Pendaftaran Sampai Dengan Dinyatakan Lulus?	Melakukan Persiapan (Pembutan Proposal), Survei Lokasi Kerja Praktek, Mempelajari Struktur Organisasi, Pengumpulan Data, Analisa Data, Desain, Pemecahan Masalah (Solusi), Coding/ Pembuatan Aplikasi, Prototype, Trial And Error, Penyusunan Laporan Kerja Praktek, Presentasi, Sidang Penentuan Lulus Atau Tidak Lulus

Bandung, 13 Juni 2019

Narasumber


MEGANTARI SUHENDAR

Lampiran 5 : Script Program

Ini adalah salah satu *scripting blueprint* yang di convert ke c++, dikarenakan terlalu banyak blueprint maka penyusun melampirkan salah satu saja.

Interaction.cpp

```
#include "Skripsi.h"
#include "Interaction__pf4034160924.h"
#include "GeneratedCodeHelpers.h"
#include
"Runtime/Engine/Classes/Engine/ComponentDelegateBinding.h"
#include
"Runtime/Engine/Classes/Engine/SimpleConstructionScript.h"
#include "Runtime/Engine/Classes/Engine/SCS_Node.h"
#include
"Runtime/Engine/Classes/Components/StaticMeshComponent.h"
#include "Runtime/Engine/Classes/Components/MeshComponent.h"
#include "Runtime/Engine/Classes/Components/PrimitiveComponent.h"
#include "Runtime/Engine/Classes/Components/SceneComponent.h"
#include "Runtime/Engine/Classes/Components/ActorComponent.h"
#include "Runtime/Engine/Classes/Engine/EngineBaseTypes.h"
#include "Runtime/Engine/Classes/Engine/AssetUserData.h"
#include "Runtime/Engine/Public/ComponentInstanceDataCache.h"
#include "Runtime/Engine/Classes/EdGraph/EdGraphPin.h"
#include "Runtime/Engine/Classes/Engine/NetSerialization.h"
#include "Runtime/Engine/Classes/Components/InputComponent.h"
#include "Runtime/Engine/Classes/GameFramework/PlayerInput.h"
#include "Runtime/InputCore/Classes/InputCoreTypes.h"
#include
"Runtime/Engine/Classes/GameFramework/PlayerController.h"
#include "Runtime/Engine/Classes/GameFramework/Controller.h"
#include "Runtime/Engine/Classes/GameFramework/PlayerState.h"
#include "Runtime/Engine/Classes/GameFramework/Info.h"
#include "Runtime/Engine/Classes/Components/BillboardComponent.h"
#include "Runtime/Engine/Classes/Engine/Texture2D.h"
#include "Runtime/Engine/Classes/Engine/Texture.h"
#include "Runtime/Engine/Classes/Engine/TextureDefines.h"
#include
"Runtime/Engine/Classes/Interfaces/Interface_AssetUserData.h"
#include "Runtime/Engine/Classes/GameFramework/LocalMessage.h"
#include
"Runtime/Engine/Classes/GameFramework/OnlineReplStructs.h"
#include "Runtime/CoreUObject/Public/UObject/CoreOnline.h"
#include "Runtime/Engine/Classes/GameFramework/Pawn.h"
#include
"Runtime/Engine/Classes/GameFramework/PawnMovementComponent.h"
```

```

#include
"Runtime/Engine/Classes/GameFramework/NavMovementComponent.h"
#include
"Runtime/Engine/Classes/GameFramework/MovementComponent.h"
#include "Runtime/Engine/Classes/GameFramework/PhysicsVolume.h"
#include "Runtime/Engine/Classes/GameFramework/Volume.h"
#include "Runtime/Engine/Classes/Engine/Brush.h"
#include "Runtime/Engine/Classes/Components/BrushComponent.h"
#include "Runtime/Engine/Classes/PhysicsEngine/BodySetup.h"
#include "Runtime/Engine/Classes/PhysicsEngine/AggregateGeom.h"
#include "Runtime/Engine/Classes/PhysicsEngine/SphereElem.h"
#include "Runtime/Engine/Classes/PhysicsEngine/ShapeElem.h"
#include "Runtime/Engine/Classes/PhysicsEngine/BoxElem.h"
#include "Runtime/Engine/Classes/PhysicsEngine/SphylElem.h"
#include "Runtime/Engine/Classes/PhysicsEngine/ConvexElem.h"
#include
"Runtime/Engine/Classes/PhysicsEngine/TaperedCapsuleElem.h"
#include "Runtime/Engine/Classes/PhysicsEngine/BodySetupEnums.h"
#include
"Runtime/Engine/Classes/PhysicalMaterials/PhysicalMaterial.h"
#include
"Runtime/Engine/Classes/PhysicsEngine/PhysicsSettingsEnums.h"
#include
"Runtime/Engine/Classes/PhysicalMaterials/PhysicalMaterialPropertyBase.h"
#include "Runtime/Engine/Classes/Vehicles/TireType.h"
#include "Runtime/Engine/Classes/Engine/DataAsset.h"
#include "Runtime/Engine/Classes/PhysicsEngine/BodyInstance.h"
#include "Runtime/Engine/Classes/AI/Navigation/NavigationTypes.h"
#include
"Runtime/Engine/Classes/AI/Navigation/NavAgentInterface.h"
#include "Runtime/AIModule/Classes/AIController.h"
#include
"Runtime/AIModule/Classes/Navigation/PathFollowingComponent.h"
#include "Runtime/NavigationSystem/Public/NavigationData.h"
#include
"Runtime/Engine/Classes/AI/Navigation/NavigationDataInterface.h"
#include "Runtime/AIModule/Classes/AIResourceInterface.h"
#include
"Runtime/Engine/Classes/AI/Navigation/PathFollowingAgentInterface.h"
#include "Runtime/AIModule/Classes/BrainComponent.h"
#include
"Runtime/AIModule/Classes/BehaviorTree/BlackboardComponent.h"
#include "Runtime/AIModule/Classes/BehaviorTree/BlackboardData.h"
#include
"Runtime/AIModule/Classes/BehaviorTree/Blackboard/BlackboardKeyTy
pe.h"

```

```

#include
"Runtime/AIModule/Classes/Perception/AIPerceptionComponent.h"
#include "Runtime/AIModule/Classes/Perception/AISenseConfig.h"
#include "Runtime/AIModule/Classes/Perception/AISense.h"
#include
"Runtime/AIModule/Classes/Perception/AIPerceptionTypes.h"
#include
"Runtime/AIModule/Classes/Perception/AIPerceptionSystem.h"
#include "Runtime/AIModule/Classes/AISubsystem.h"
#include "Runtime/AIModule/Classes/AISystem.h"
#include "Runtime/Engine/Classes/AI/AISystemBase.h"
#include
"Runtime/AIModule/Classes/BehaviorTree/BehaviorTreeManager.h"
#include "Runtime/AIModule/Classes/BehaviorTree/BehaviorTree.h"
#include
"Runtime/AIModule/Classes/BehaviorTree/BTCompositeNode.h"
#include "Runtime/AIModule/Classes/BehaviorTree/BTNode.h"
#include
"Runtime/GameplayTasks/Classes/GameplayTaskOwnerInterface.h"
#include "Runtime/AIModule/Classes/BehaviorTree/BTTaskNode.h"
#include "Runtime/AIModule/Classes/BehaviorTree/BTService.h"
#include
"Runtime/AIModule/Classes/BehaviorTree/BTAuxiliaryNode.h"
#include "Runtime/AIModule/Classes/BehaviorTree/BTDecorator.h"
#include
"Runtime/AIModule/Classes/BehaviorTree/BehaviorTreeTypes.h"
#include
"Runtime/AIModule/Classes/BehaviorTree/BehaviorTreeComponent.h"
#include "Runtime/GameplayTags/Classes/GameplayTagContainer.h"
#include
"Runtime/AIModule/Classes/EnvironmentQuery/EnvQueryManager.h"
#include "Runtime/AIModule/Classes/EnvironmentQuery/EnvQuery.h"
#include
"Runtime/AIModule/Classes/EnvironmentQuery/EnvQueryOption.h"
#include
"Runtime/AIModule/Classes/EnvironmentQuery/EnvQueryGenerator.h"
#include
"Runtime/AIModule/Classes/EnvironmentQuery/EnvQueryNode.h"
#include
"Runtime/AIModule/Classes/EnvironmentQuery/Items/EnvQueryItemType
.h"
#include
"Runtime/AIModule/Classes/EnvironmentQuery/EnvQueryTest.h"
#include
"Runtime/AIModule/Classes/EnvironmentQuery/EnvQueryTypes.h"
#include
"Runtime/AIModule/Classes/DataProviders/AIDataProvider.h"

```

```

#include
"Runtime/AIModule/Classes/EnvironmentQuery/EnvQueryContext.h"
#include
"Runtime/AIModule/Classes/EnvironmentQuery/EnvQueryInstanceBlueprintWrapper.h"
#include
"Runtime/AIModule/Classes/EnvironmentQuery/EQSQueryResultSourceInterface.h"
#include
"Runtime/AIModule/Classes/Blueprint/AIAsyncTaskBlueprintProxy.h"
#include "Runtime/AIModule/Classes/AITypes.h"
#include "Runtime/AIModule/Classes/HotSpots/AIHotSpotManager.h"
#include
"Runtime/AIModule/Classes/Navigation/NavLocalGridManager.h"
#include "Runtime/AIModule/Classes/Perception/AISenseEvent.h"
#include
"Runtime/AIModule/Classes/Actions/PawnActionsComponent.h"
#include "Runtime/AIModule/Classes/Actions/PawnAction.h"
#include "Runtime/GameplayTasks/Classes/GameplayTasksComponent.h"
#include "Runtime/GameplayTasks/Classes/GameplayTask.h"
#include "Runtime/GameplayTasks/Classes/GameplayTaskResource.h"
#include
"Runtime/NavigationSystem/Public/NavFilters/NavigationQueryFilter.h"
#include "Runtime/NavigationSystem/Public/NavAreas/NavArea.h"
#include "Runtime/Engine/Classes/AI/Navigation/NavAreaBase.h"
#include
"Runtime/AIModule/Classes/Perception/AIPerceptionListenerInterface.h"
#include "Runtime/AIModule/Classes/GenericTeamAgentInterface.h"
#include
"Runtime/Engine/Public/VisualLogger/Vj2khmccLDDH1GxGcJxFPBN4bzMLPqVsp8.h"
#include "Runtime/Engine/Classes/GameFramework/EngineMessage.h"
#include "Runtime/Engine/Classes/GameFramework/DamageType.h"
#include "Runtime/Engine/Classes/GameFramework/Character.h"
#include
"Runtime/Engine/Classes/Components/SkeletalMeshComponent.h"
#include
"Runtime/Engine/Classes/Components/SkinnedMeshComponent.h"
#include "Runtime/Engine/Classes/Engine/SkeletalMesh.h"
#include "Runtime/Engine/Classes/Animation/Skeleton.h"
#include "Runtime/Engine/Classes/Engine/SkeletalMeshSocket.h"
#include "Runtime/Engine/Classes/Animation/SmartName.h"
#include "Runtime/Engine/Classes/Animation/BlendProfile.h"
#include "Runtime/Engine/Public/BoneContainer.h"

```

```

#include
"Runtime/Engine/Classes/Interfaces/Interface_PreviewMeshProvider.
h"
#include "Runtime/Engine/Classes/Materials/MaterialInterface.h"
#include "Runtime/Engine/Classes/Engine/SubsurfaceProfile.h"
#include "Runtime/Engine/Classes/Materials/Material.h"
#include "Runtime/Engine/Public/MaterialShared.h"
#include "Runtime/Engine/Classes/Materials/MaterialExpression.h"
#include "Runtime/Engine/Classes/Materials/MaterialFunction.h"
#include
"Runtime/Engine/Classes/Materials/MaterialFunctionInterface.h"
#include
"Runtime/Engine/Classes/Materials/MaterialParameterCollection.h"
#include "Runtime/Engine/Classes/Engine/BlendableInterface.h"
#include "Runtime/Engine/Public/Components.h"
#include "Runtime/Engine/Public/PerPlatformProperties.h"
#include "Runtime/Engine/Public/SkeletalMeshReductionSettings.h"
#include "Runtime/Engine/Classes/Animation/AnimSequence.h"
#include "Runtime/Engine/Classes/Animation/AnimSequenceBase.h"
#include "Runtime/Engine/Classes/Animation/AnimationAsset.h"
#include "Runtime/Engine/Classes/Animation/AnimMetaData.h"
#include "Runtime/Engine/Public/Animation/AnimTypes.h"
#include "Runtime/Engine/Classes/Animation/AnimLinkableElement.h"
#include "Runtime/Engine/Classes/Animation/AnimMontage.h"
#include "Runtime/Engine/Classes/Animation/AnimCompositeBase.h"
#include "Runtime/Engine/Public/AlphaBlend.h"
#include "Runtime/Engine/Classes/Curves/CurveFloat.h"
#include "Runtime/Engine/Classes/Curves/CurveBase.h"
#include "Runtime/Engine/Classes/Curves/RichCurve.h"
#include "Runtime/Engine/Classes/Curves/RealCurve.h"
#include "Runtime/Engine/Classes/Curves/IndexedCurve.h"
#include "Runtime/Engine/Classes/Curves/KeyHandle.h"
#include "Runtime/Engine/Classes/Animation/AnimEnums.h"
#include "Runtime/Engine/Classes/Animation/TimeStretchCurve.h"
#include
"Runtime/Engine/Classes/Animation/AnimNotifies/AnimNotify.h"
#include
"Runtime/Engine/Classes/Animation/AnimNotifies/AnimNotifyState.h"
#include "Runtime/Engine/Public/Animation/AnimCurveTypes.h"
#include
"Runtime/Engine/Classes/Animation/AnimCurveCompressionSettings.h"
#include
"Runtime/Engine/Classes/Animation/AnimCurveCompressionCodec.h"
#include
"Runtime/Engine/Classes/Animation/GLT1Kqob5UDEML61gCyjnAcfMXgkdP3
wGcg45ichCurve.h"
#include "Runtime/Engine/Classes/PhysicsEngine/PhysicsAsset.h"

```

```

#include
"Runtime/Engine/Classes/PhysicsEngine/PhysicalAnimationComponent.
h"
#include
"Runtime/Engine/Classes/PhysicsEngine/PhysicsConstraintTemplate.h
"
#include
"Runtime/Engine/Classes/PhysicsEngine/ConstraintInstance.h"
#include "Runtime/Engine/Classes/PhysicsEngine/ConstraintTypes.h"
#include
"Runtime/Engine/Classes/PhysicsEngine/ConstraintDrives.h"
#include "Runtime/Engine/Classes/EditorFramework/ThumbnailInfo.h"
#include "Runtime/Engine/Public/Animation/NodeMappingContainer.h"
#include
"Runtime/Engine/Public/Animation/NodeMappingProviderInterface.h"
#include "Runtime/Engine/Classes/Animation/MorphTarget.h"
#include "Runtime/Engine/Classes/Animation/AnimInstance.h"
#include "Runtime/Engine/Public/Animation/AnimNotifyQueue.h"
#include "Runtime/Engine/Public/Animation/PoseSnapshot.h"
#include
"Runtime/ClothingSystemRuntimeInterface/Public/ClothingAssetInter
face.h"
#include "Runtime/Engine/Classes/Engine/SkeletalMeshSampling.h"
#include
"Runtime/Engine/Classes/Engine/SkeletalMeshLODSettings.h"
#include "Runtime/Engine/Classes/Engine/Blueprint.h"
#include "Runtime/Engine/Classes/Engine/BlueprintCore.h"
#include
"Runtime/Engine/Classes/Engine/BlueprintGeneratedClass.h"
#include "Runtime/Engine/Classes/Engine/TimelineTemplate.h"
#include "Runtime/Engine/Classes/Components/TimelineComponent.h"
#include "Runtime/Engine/Classes/Curves/CurveVector.h"
#include "Runtime/Engine/Classes/Curves/CurveLinearColor.h"
#include
"Runtime/Engine/Classes/Engine/InheritableComponentHandler.h"
#include
"Runtime/Engine/Classes/Interfaces/Interface_CollisionDataProvide
r.h"
#include
"Runtime/Engine/Classes/Animation/AnimBlueprintGeneratedClass.h"
#include
"Runtime/Engine/Classes/Engine/DynamicBlueprintBinding.h"
#include
"Runtime/Engine/Classes/Animation/AnimStateMachineTypes.h"
#include "Runtime/Engine/Classes/Animation/AnimNodeBase.h"
#include "Runtime/Engine/Classes/Animation/AnimClassInterface.h"
#include "Runtime/Engine/Public/SingleAnimationPlayData.h"

```



```

#include
"Runtime/ClothingSystemRuntimeInterface/Public/ClothingSimulation
FactoryInterface.h"
#include
"Runtime/ClothingSystemRuntimeInterface/Public/ClothingSimulation
Interactor.h"
#include
"Runtime/Engine/Classes/GameFramework/CharacterMovementComponent.
h"
#include
"Runtime/Engine/Classes/AI/Navigation/NavigationAvoidanceTypes.h"
#include
"Runtime/Engine/Classes/GameFramework/RootMotionSource.h"
#include "Runtime/Engine/Public/AI/RVOAvoidanceInterface.h"
#include
"Runtime/Engine/Classes/Interfaces/NetworkPredictionInterface.h"
#include "Runtime/Engine/Classes/Components/CapsuleComponent.h"
#include "Runtime/Engine/Classes/Components/ShapeComponent.h"
#include "Runtime/Engine/Classes/Engine/Player.h"
#include
"Runtime/Engine/Classes/Matinee/InterpTrackInstDirector.h"
#include "Runtime/Engine/Classes/Matinee/InterpTrackInst.h"
#include "Runtime/Engine/Classes/GameFramework/HUD.h"
#include "Runtime/Engine/Classes/Engine/Canvas.h"
#include "Runtime/Engine/Classes/Debug/ReporterGraph.h"
#include "Runtime/Engine/Classes/Debug/ReporterBase.h"
#include "Runtime/Engine/Classes/Engine/Font.h"
#include "Runtime/Engine/Classes/Engine/FontImportOptions.h"
#include "Runtime/SlateCore/Public/Fonts/CompositeFont.h"
#include "Runtime/SlateCore/Public/Fonts/FontProviderInterface.h"
#include "Runtime/Engine/Classes/GameFramework/DebugTextInfo.h"
#include "Runtime/Engine/Classes/Camera/PlayerCameraManager.h"
#include "Runtime/Engine/Classes/Camera/CameraTypes.h"
#include "Runtime/Engine/Classes/Engine/Scene.h"
#include "Runtime/Engine/Classes/Engine/TextureCube.h"
#include "Runtime/Engine/Classes/Camera/CameraModifier.h"
#include
"Runtime/Engine/Classes/Particles/EmitterCameraLensEffectBase.h"
#include "Runtime/Engine/Classes/Particles/Emitter.h"
#include
"Runtime/Engine/Classes/Particles/ParticleSystemComponent.h"
#include "Runtime/Engine/Classes/Particles/ParticleSystem.h"
#include "Runtime/Engine/Classes/Particles/ParticleEmitter.h"
#include "Runtime/Engine/Public/ParticleHelper.h"
#include "Runtime/Engine/Classes/Particles/ParticleLODLevel.h"
#include
"Runtime/Engine/Classes/Particles/ParticleModuleRequired.h"
#include "Runtime/Engine/Classes/Particles/ParticleModule.h"

```

```

#include
"Runtime/Engine/Classes/Particles/ParticleSpriteEmitter.h"
#include
"Runtime/Engine/Classes/Distributions/DistributionFloat.h"
#include "Runtime/Engine/Classes/Distributions/Distribution.h"
#include "Runtime/Engine/Classes/Particles/SubUVAnimation.h"
#include
"Runtime/Engine/Classes/Particles/TypeData/ParticleModuleTypeData
Base.h"
#include
"Runtime/Engine/Classes/Particles/Spawn/ParticleModuleSpawn.h"
#include
"Runtime/Engine/Classes/Particles/Spawn/ParticleModuleSpawnBase.h
"
#include
"Runtime/Engine/Classes/Particles/Event/ParticleModuleEventGenera
tor.h"
#include
"Runtime/Engine/Classes/Particles/Event/ParticleModuleEventBase.h
"
#include
"Runtime/Engine/Classes/Particles/Event/ParticleModuleEventSendTo
Game.h"
#include
"Runtime/Engine/Classes/Particles/Orbit/ParticleModuleOrbit.h"
#include
"Runtime/Engine/Classes/Particles/Orbit/ParticleModuleOrbitBase.h
"
#include
"Runtime/Engine/Classes/Distributions/DistributionVector.h"
#include
"Runtime/Engine/Classes/Particles/Event/ParticleModuleEventReceiv
erBase.h"
#include "Runtime/Engine/Classes/Engine/InterpCurveEdSetup.h"
#include
"Runtime/Engine/Classes/Particles/ParticleSystemReplay.h"
#include
"Runtime/Engine/Classes/Materials/MaterialInstanceDynamic.h"
#include "Runtime/Engine/Classes/Materials/MaterialInstance.h"
#include
"Runtime/Engine/Classes/Materials/MaterialLayersFunctions.h"
#include
"Runtime/Engine/Classes/Materials/LT1Kqob5UDEML61gCyjnAcfMXgkdP3w
Gcg45s.h"
#include "Runtime/Engine/Public/StaticParameterSet.h"
#include
"Runtime/Engine/Classes/Camera/CameraModifier_CameraShake.h"
#include "Runtime/Engine/Classes/Camera/CameraShake.h"

```

```

#include "Runtime/Engine/Classes/Camera/CameraAnim.h"
#include "Runtime/Engine/Classes/Matinee/InterpGroup.h"
#include "Runtime/Engine/Classes/Matinee/InterpTrack.h"
#include "Runtime/Engine/Classes/Camera/CameraAnimInst.h"
#include "Runtime/Engine/Classes/Matinee/InterpGroupInst.h"
#include "Runtime/Engine/Classes/Matinee/InterpTrackMove.h"
#include "Runtime/Engine/Classes/Matinee/InterpTrackInstMove.h"
#include "Runtime/Engine/Classes/Camera/CameraActor.h"
#include "Runtime/Engine/Classes/Camera/CameraComponent.h"
#include "Runtime/Engine/Classes/GameFramework/CheatManager.h"
#include "Runtime/Engine/Classes/Engine/DebugCameraController.h"
#include
"Runtime/Engine/Classes/Components/DrawFrustumComponent.h"
#include
"Runtime/Engine/Classes/GameFramework/ForceFeedbackEffect.h"
#include "Runtime/Engine/Classes/Engine/NetConnection.h"
#include "Runtime/Engine/Classes/Engine/ChildConnection.h"
#include "Runtime/Engine/Classes/Engine/PackageMapClient.h"
#include "Runtime/Engine/Classes/Engine/NetDriver.h"
#include "Runtime/Engine/Classes/Engine/World.h"
#include "Runtime/Engine/Classes/Engine/Level.h"
#include "Runtime/Engine/Classes/Components/ModelComponent.h"
#include "Runtime/Engine/Classes/Engine/LevelActorContainer.h"
#include "Runtime/Engine/Classes/Engine/LevelScriptActor.h"
#include "Runtime/Engine/Classes/Engine/NavigationObjectBase.h"
#include
"Runtime/Engine/Classes/AI/Navigation/NavigationDataChunk.h"
#include "Runtime/Engine/Classes/Engine/MapBuildDataRegistry.h"
#include "Runtime/Engine/Classes/GameFramework/WorldSettings.h"
#include "Runtime/Engine/Classes/AI/NavigationSystemConfig.h"
#include
"Runtime/Engine/Classes/GameFramework/DefaultPhysicsVolume.h"
#include
"Runtime/Engine/Classes/PhysicsEngine/PhysicsCollisionHandler.h"
#include "Runtime/Engine/Classes/Sound/SoundBase.h"
#include "Runtime/Engine/Classes/Sound/SoundClass.h"
#include "Runtime/Engine/Classes/Sound/SoundMix.h"
#include "Runtime/Engine/Classes/Sound/SoundConcurrency.h"
#include "Runtime/Engine/Classes/Sound/SoundAttenuation.h"
#include "Runtime/Engine/Classes/Engine/Attenuation.h"
#include "Runtime/Engine/Public/IAudioExtensionPlugin.h"
#include "Runtime/Engine/Classes/Sound/SoundSubmix.h"
#include "Runtime/Engine/Classes/Sound/SoundEffectSubmix.h"
#include "Runtime/Engine/Classes/Sound/SoundEffectPreset.h"
#include "Runtime/Engine/Public/IAmbisonicsMixer.h"
#include "Runtime/Engine/Classes/Sound/SoundWave.h"

```

```

#include
"Runtime/AudioPlatformConfiguration/Public/AudioCompressionSettings.h"
#include "Runtime/Engine/Classes/Sound/SoundGroups.h"
#include "Runtime/Engine/Classes/Engine/CurveTable.h"
#include "Runtime/Engine/Classes/Sound/SoundEffectSource.h"
#include "Runtime/Engine/Classes/Sound/SoundSourceBusSend.h"
#include "Runtime/Engine/Classes/Sound/SoundSourceBus.h"
#include "Runtime/Engine/Classes/GameFramework/GameModeBase.h"
#include "Runtime/Engine/Classes/GameFramework/GameSession.h"
#include "Runtime/Engine/Classes/GameFramework/GameStateBase.h"
#include "Runtime/Engine/Classes/GameFramework/SpectatorPawn.h"
#include "Runtime/Engine/Classes/GameFramework/DefaultPawn.h"
#include "Runtime/Engine/Classes/Components/SphereComponent.h"
#include
"Runtime/Engine/Classes/GameFramework/FloatingPawnMovement.h"
#include
"Runtime/Engine/Classes/GameFramework/SpectatorPawnMovement.h"
#include "Runtime/Engine/Classes/Engine/ServerStatReplicator.h"
#include
"Runtime/Engine/Classes/GameFramework/GameNetworkManager.h"
#include "Runtime/Engine/Classes/Sound/AudioVolume.h"
#include "Runtime/Engine/Classes/Sound/ReverbEffect.h"
#include "Runtime/Engine/Classes/Engine/BookmarkBase.h"
#include "DmgTypeBP_Environmental__pf1001828891.h"
#include "Runtime/Engine/Classes/Engine/BookMark.h"
#include "Runtime/Engine/Classes/Components/LineBatchComponent.h"
#include "Runtime/Engine/Classes/Engine/LevelStreaming.h"
#include "Runtime/Engine/Classes/Engine/LevelStreamingVolume.h"
#include "Runtime/Engine/Classes/Engine/DemoNetDriver.h"
#include
"Runtime/Engine/Classes/Particles/ParticleEventManager.h"
#include "Runtime/Engine/Classes/AI/NavigationSystemBase.h"
#include
"Runtime/Engine/Classes/AI/Navigation/AvoidanceManager.h"
#include "Runtime/Engine/Classes/Engine/GameInstance.h"
#include "Runtime/Engine/Classes/Engine/LocalPlayer.h"
#include "Runtime/Engine/Classes/Engine/GameViewportClient.h"
#include "Runtime/Engine/Classes/Engine/ScriptViewportClient.h"
#include "Runtime/Engine/Classes/Engine/Console.h"
#include "Runtime/Engine/Classes/Engine/DebugDisplayProperty.h"
#include "Runtime/Engine/Classes/Engine/Engine.h"
#include
"Runtime/Engine/Classes/GameFramework/GameUserSettings.h"
#include "Runtime/Engine/Classes/Engine/AssetManager.h"
#include "Runtime/Engine/Classes/Engine/EngineCustomTimeStep.h"
#include "Runtime/Engine/Classes/Engine/TimecodeProvider.h"
#include "Runtime/Engine/Classes/GameFramework/OnlineSession.h"

```

```

#include
"Runtime/Engine/Classes/Materials/LT1Kqob5UDEML61gCyjnAcfMXgkdP3w
Gcge.h"
#include "Runtime/Engine/Classes/Engine/WorldComposition.h"
#include "Runtime/Engine/Classes/Particles/WorldPSCPool.h"
#include "Runtime/Engine/Classes/Engine/Channel.h"
#include "Runtime/Engine/Classes/Engine/ReplicationDriver.h"
#include "Runtime/Engine/Classes/GameFramework/TouchInterface.h"
#include "Runtime/UMG/Public/Blueprint/UserWidget.h"
#include "Runtime/UMG/Public/Components/Widget.h"
#include "Runtime/UMG/Public/Components/Visual.h"
#include "Runtime/UMG/Public/Components/PanelSlot.h"
#include "Runtime/UMG/Public/Components/PanelWidget.h"
#include "Runtime/UMG/Public/Components/SlateWrapperTypes.h"
#include "Runtime/UMG/Public/Slate/WidgetTransform.h"
#include "Runtime/SlateCore/Public/Layout/Clipping.h"
#include "Runtime/UMG/Public/Blueprint/WidgetNavigation.h"
#include "Runtime/SlateCore/Public/Input/NavigationReply.h"
#include "Runtime/SlateCore/Public/Types/SlateEnums.h"
#include "Runtime/SlateCore/Public/Layout/FlowDirection.h"
#include "Runtime/UMG/Public/Binding/PropertyBinding.h"
#include "Runtime/UMG/Public/Binding/DynamicPropertyPath.h"
#include "Runtime/PropertyPath/Public/PropertyPathHelpers.h"
#include "Runtime/SlateCore/Public/Layout/Geometry.h"
#include "Runtime/SlateCore/Public/Input/Events.h"
#include "Runtime/SlateCore/Public/Styling/SlateColor.h"
#include "Runtime/SlateCore/Public/Styling/SlateBrush.h"
#include "Runtime/SlateCore/Public/Layout/Margin.h"
#include "Runtime/SlateCore/Public/Styling/SlateTypes.h"
#include "Runtime/UMG/Public/Animation/UMGSequencePlayer.h"
#include "Runtime/UMG/Public/Animation/WidgetAnimation.h"
#include "Runtime/MovieScene/Public/MovieSceneSequence.h"
#include "Runtime/MovieScene/Public/MovieSceneSignedObject.h"
#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneEvaluationTemplat
e.h"
#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneTrackIdentifier.h
"
#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneEvaluationTrack.h
"
#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneSegment.h"
#include "Runtime/MovieScene/Public/MovieSceneTrack.h"
#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneEvalTemplate.h"

```

```

#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneTrackImplementati
on.h"
#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneEvaluationField.h
"
#include "Runtime/MovieScene/Public/MovieSceneFrameMigration.h"
#include "Runtime/MovieScene/Public/MovieSceneSequenceID.h"
#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneEvaluationKey.h"
#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneSequenceHierarchy
.h"
#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneSequenceTransform
.h"
#include
"Runtime/MovieScene/Public/Evaluation/MovieSceneSequenceInstanced
ata.h"
#include "Runtime/MovieScene/Public/MovieSceneSection.h"
#include "Runtime/MovieScene/Public/MovieScene.h"
#include "Runtime/MovieScene/Public/MovieSceneSpawnable.h"
#include "Runtime/MovieScene/Public/MovieScenePossessable.h"
#include "Runtime/MovieScene/Public/MovieSceneBinding.h"
#include "Runtime/MovieScene/Public/MovieSceneFwd.h"
#include "Runtime/UMG/Public/Animation/WidgetAnimationBinding.h"
#include "Runtime/UMG/Public/Blueprint/WidgetTree.h"
#include "Runtime/Slate/Public/Widgets/Layout/Anchors.h"
#include "Runtime/UMG/Public/Blueprint/DragDropOperation.h"
#include "Runtime/UMG/Public/Components/NamedSlotInterface.h"
#include
"Runtime/Engine/Classes/Haptics/HapticFeedbackEffect_Base.h"
#include "Runtime/Engine/Classes/Engine/LatentActionManager.h"
#include "Runtime/Engine/Classes/Matinee/MatineeActor.h"
#include "Runtime/Engine/Classes/Matinee/InterpData.h"
#include "Runtime/Engine/Classes/Matinee/InterpGroupDirector.h"
#include
"Runtime/Engine/Classes/Components/ChildActorComponent.h"
#include
"Runtime/Engine/Classes/AI/Navigation/NavRelevantInterface.h"
#include "Runtime/Engine/Classes/Engine/StaticMesh.h"
#include "Runtime/Engine/Classes/Engine/StaticMeshSocket.h"
#include
"Runtime/Engine/Classes/AI/Navigation/NavCollisionBase.h"
#include "Runtime/Engine/Classes/Engine/TextureStreamingTypes.h"
#include
"Runtime/NavigationSystem/Public/NavAreas/NavArea_Obstacle.h"
#include "Runtime/UMG/Public/Components/WidgetComponent.h"

```

```

#include "Runtime/Engine/Classes/Engine/TextureRenderTarget2D.h"
#include "Runtime/Engine/Classes/Engine/TextureRenderTarget.h"
#include
"Runtime/Engine/Classes/Materials/MaterialInstanceConstant.h"
#include "InteractUI__pf4034160924.h"
#include "PickUp_BP__pf4034160924.h"
#include "ALS_BaseCharacter__pf1533377440.h"
#include "ParentAI__pf379577716.h"
#include
"Runtime/Engine/Classes/Engine/InputActionDelegateBinding.h"
#include "Runtime/Engine/Classes/Engine/InputDelegateBinding.h"
#include
"Runtime/Engine/Classes/Engine/InputKeyDelegateBinding.h"
#include "Runtime/Slate/Public/Framework/Commands/InputChord.h"
#include
"Runtime/Engine/Classes/Engine/InputAxisDelegateBinding.h"
#include "ALS_Interface__pf1533377440.h"
#include "PlayerWidget__pf570508212.h"
#include "InventoryBP__pf4285404021.h"
#include "ALS_Gait__pf712164115.h"
#include "ALS_Stance__pf712164115.h"
#include "ALS_RotationMode__pf712164115.h"
#include "ALS_MovementMode__pf712164115.h"
#include "ALS_PivotParams__pf2302773760.h"
#include "IdleEntryState__pf712164115.h"
#include "ALS_ViewMode__pf712164115.h"
#include "ExcMenu__pf864017494.h"
#include "Fade__pf4034160924.h"
#include "MisiSelesai__pf4034160924.h"
#include "MyDialogueWidget__pf567404935.h"
#include "ALS_CameraSettings__pf2302773760.h"
#include "MyGameIsntance__pf4285404021.h"
#include "Runtime/Engine/Classes/Kismet/KismetMathLibrary.h"
#include
"Runtime/Engine/Classes/Kismet/BlueprintFunctionLibrary.h"
#include "Runtime/UMG/Public/Blueprint/WidgetBlueprintLibrary.h"
#include "Runtime/Engine/Public/Slate/SGameLayerManager.h"
#include "Runtime/SlateCore/Public/Rendering/RenderingCommon.h"
#include "Runtime/Engine/Classes/Slate/SlateBrushAsset.h"
#include "Runtime/Engine/Classes/Kismet/GameplayStatics.h"
#include "Runtime/Engine/Classes/Components/AudioComponent.h"
#include
"Runtime/Engine/Classes/GameFramework/ForceFeedbackAttenuation.h"
#include
"Runtime/Engine/Classes/Components/ForceFeedbackComponent.h"
#include "Runtime/Engine/Classes/Sound/DialogueWave.h"
#include "Runtime/Engine/Classes/Sound/DialogueTypes.h"
#include "Runtime/Engine/Classes/Sound/DialogueVoice.h"

```

```

#include "Runtime/Engine/Classes/Sound/DialogueSoundWaveProxy.h"
#include "Runtime/Engine/Classes/Components/DecalComponent.h"
#include "Runtime/Engine/Classes/GameFramework/SaveGame.h"
#include "Runtime/Engine/Classes/Kismet/GameplayStaticsTypes.h"
#include "Runtime/Engine/Classes/Kismet/KismetSystemLibrary.h"
#include "Runtime/Engine/Classes/Engine/CollisionProfile.h"
#include "Runtime/Engine/Classes/Kismet/KismetArrayLibrary.h"
#include "ItemEnum__pf4285404021.h"
#include "Harga__pf4034160924.h"
#include "StaminaPickup__pf4034160924.h"
#include "HealPickup__pf4034160924.h"
#include "Runtime/Engine/Classes/Kismet/KismetTextLibrary.h"
#include "Proposal__pf4034160924.h"
#include "Laporan__pf4034160924.h"
#include "Pemabayaran__pf4034160924.h"
#include "FotocopyPembayaran__pf4034160924.h"
#include "mISION1__pf4034160924.h"
#include "NameAI__pf567404935.h"
#include
"../Plugins/Marketplace/DialoguePlugin/Source/DialoguePlugin/Public/Dialogue.h"
#include "Runtime/Engine/Classes/Kismet/KismetStringLibrary.h"
#include
"Runtime/Engine/Classes/GameFramework/SpringArmComponent.h"
#include "InteractableVj2khmccLDDH1GxGcJxFPBN4bzMLPqVsp8.h"
#include "InteractableVariableforInteract__pf117433938.h"
#include "InteractableVariable__pf117433938.h"
#include "PlayerData__pf1533377440.h"
#include
"Runtime/Engine/Classes/Kismet/KismetNodeHelperLibrary.h"
#include "ALS_Camera_Targets__pf2302773760.h"
#include "ALS_CameraSettings_Stance__pf2302773760.h"
#include "ALS_CameraSettings_Gait__pf2302773760.h"
#include "CardinalDirection__pf712164115.h"
#include "Runtime/Engine/Classes/Components/ArrowComponent.h"

#ifdef _MSC_VER
#pragma warning (push)
#pragma warning (disable : 4883)
#endif
PRAGMA_DISABLE_DEPRECATION_WARNINGS
PRAGMA_DISABLE_OPTIMIZATION
AInteraction_C__pf4034160924::AInteraction_C__pf4034160924(const
FObjectInitializer& ObjectInitializer) : Super(ObjectInitializer)
{
    if(HasAnyFlags(RF_ClassDefaultObject) &&
(AInteraction_C__pf4034160924::StaticClass() == GetClass())){

```



```

        AInteraction_C__pf4034160924::__CustomDynamicClassInitializ
ation(CastChecked<UDynamicClass>(GetClass()));}
        bpv__Mesh__pf =
CreateDefaultSubobject<UStaticMeshComponent>(TEXT("Mesh"));
        bpv__Radius__pf =
CreateDefaultSubobject<USphereComponent>(TEXT("Radius"));
        bpv__InteractUI__pf =
CreateDefaultSubobject<UWidgetComponent>(TEXT("InteractUI"));
        RootComponent = bpv__Mesh__pf;
        bpv__Mesh__pf->CreationMethod =
EComponentCreationMethod::Native;
        auto& __Local__0 = (*(AccessPrivateProperty<UStaticMesh*
>((bpv__Mesh__pf), UStaticMeshComponent::__PPO__StaticMesh() )));
        __Local__0 =
CastChecked<UStaticMesh>(CastChecked<UDynamicClass>(AInteraction_
C__pf4034160924::StaticClass())->UsedAssets[0],
ECastCheckedType::NullAllowed);
        static TWeakObjectPtr<UPROPERTY> __Local__2{};
        const UPROPERTY* __Local__1 = __Local__2.Get();
        if (nullptr == __Local__1){
            __Local__1 = (UActorComponent::StaticClass())-
>FindPropertyByName(FName(TEXT("bCanEverAffectNavigation"))));
            check(__Local__1);
            __Local__2 = __Local__1;}
        (((UBoolProperty*)__Local__1)-
>SetPropertyValue_InContainer((bpv__Mesh__pf), true, 0));
        bpv__Radius__pf->CreationMethod =
EComponentCreationMethod::Native;
        bpv__Radius__pf->AttachToComponent(bpv__Mesh__pf,
FAttachmentTransformRules::KeepRelativeTransform );
        auto& __Local__3 = (*(AccessPrivateProperty<float
>((bpv__Radius__pf), USphereComponent::__PPO__SphereRadius() )));
        __Local__3 = 50.000000f;
        bpv__Radius__pf->AreaClass =
UNavArea_Obstacle::StaticClass();
        bpv__Radius__pf->bVisible = false;
        bpv__Radius__pf->bHiddenInGame = false;
        (((UBoolProperty*)__Local__1)-
>SetPropertyValue_InContainer((bpv__Radius__pf), true, 0));
        bpv__InteractUI__pf->CreationMethod =
EComponentCreationMethod::Native;
        bpv__InteractUI__pf->AttachToComponent(bpv__Mesh__pf,
FAttachmentTransformRules::KeepRelativeTransform );
        auto& __Local__4 = (*(AccessPrivateProperty<EWidgetSpace
>((bpv__InteractUI__pf), UWidgetComponent::__PPO__Space() )));
        __Local__4 = EWidgetSpace::Screen;

```

```

        auto& __Local__5 =
        (*(AccessPrivateProperty<TSubclassOf<UUserWidget>
        >((bpv__InteractUI__pf), UWidgetComponent::__PPO__WidgetClass()
        ))));
        __Local__5 = UInteractUI_C__pf4034160924::StaticClass();
        auto& __Local__6 = (*(AccessPrivateProperty<FIntPoint
        >((bpv__InteractUI__pf), UWidgetComponent::__PPO__DrawSize() )));
        __Local__6.X = 100;
        __Local__6.Y = 100;
        bpv__InteractUI__pf->RelativeLocation = FVector(0.000000,
        0.000000, 695.000000);
        bpv__InteractUI__pf->bVisible = false;
        (((UBoolProperty*)__Local__1)-
        >SetPropertyValue_InContainer((bpv__InteractUI__pf), true, 0));
        bpv__MeshNew__pf =
        CastChecked<UStaticMesh>(CastChecked<UDynamicClass>(AInteraction_
        C__pf4034160924::StaticClass())->UsedAssets[1],
        ECastCheckedType::NullAllowed);
        bpv__TrigerRadius__pf = 0.000000f;
        bpv__triggerLocation__pf = FVector(0.000000, 0.000000,
        0.000000);
        bpv__UiLocation__pf = FVector(0.000000, 0.000000,
        100.000000);
        bpv__TextWidget__pf = FTextStringHelper::CreateFromBuffer(
        TEXT("NSLOCTEXT(\"[D4ACF3FC4790ACBCD66DAB826003D666]\",
        \"FDF9D78C44DCB69EC87B69BCAC4A08A0\", \"Interact\")"));
        bpv__Player__pf = nullptr;
        bpv__InrteractUI__pf = nullptr;
        bpv__SetMat__pf = nullptr;
        bpv__SetScale__pf = FVector(1.000000, 1.000000, 1.000000);}
PRAGMA_ENABLE_OPTIMIZATION
void
AInteraction_C__pf4034160924::PostLoadSubobjects(FObjectInstancin
gGraph* OuterInstanceGraph){
    Super::PostLoadSubobjects(OuterInstanceGraph);
    if(bpv__Mesh__pf){
        bpv__Mesh__pf->CreationMethod =
        EComponentCreationMethod::Native;}
    if(bpv__Radius__pf){
        bpv__Radius__pf->CreationMethod =
        EComponentCreationMethod::Native;}
    if(bpv__InteractUI__pf){
        bpv__InteractUI__pf->CreationMethod =
        EComponentCreationMethod::Native;}}
PRAGMA_DISABLE_OPTIMIZATION
void
AInteraction_C__pf4034160924::__CustomDynamicClassInitialization(
UDynamicClass* InDynamicClass){

```

```

        ensure(0 == InDynamicClass-
>ReferencedConvertedFields.Num());
        ensure(0 == InDynamicClass->MiscConvertedSubobjects.Num());
        ensure(0 == InDynamicClass->DynamicBindingObjects.Num());
        ensure(0 == InDynamicClass->ComponentTemplates.Num());
        ensure(0 == InDynamicClass->Timelines.Num());
        ensure(nullptr == InDynamicClass->AnimClassImplementation);
        InDynamicClass->AssembleReferenceTokenStream();
        // List of all referenced converted classes
        InDynamicClass-
>ReferencedConvertedFields.Add(UInteractUI_C__pf4034160924::Static
Class());
        InDynamicClass-
>ReferencedConvertedFields.Add(APickUp_BP_C__pf4034160924::Static
Class());
        InDynamicClass-
>ReferencedConvertedFields.Add(AALS_BaseCharacter_C__pf1533377440
::StaticClass());
        InDynamicClass-
>ReferencedConvertedFields.Add(AParentAI_C__pf379577716::StaticCl
ass());
        extern UClass* Z_Construct_UClass_UInterfaceBP_C();
        InDynamicClass-
>ReferencedConvertedFields.Add(Z_Construct_UClass_UInterfaceBP_C(
));
        extern UClass* Z_Construct_UClass_UInventInterface_C();
        InDynamicClass-
>ReferencedConvertedFields.Add(Z_Construct_UClass_UInventInterfac
e_C());
        // List of all referenced converted structures
        extern UScriptStruct*
Z_Construct_USNcEH1C3UXSaTcNMHNLGNuLZqjGKbSrEMZ5404021();
        InDynamicClass-
>ReferencedConvertedFields.Add(Z_Construct_USNcEH1C3UXSaTcNMHNLGN
uLZqjGKbSrEMZ5404021());
        FConvertedBlueprintsDependencies::FillUsedAssetsInDynamicCl
ass(InDynamicClass, &__SNcEH1C3UXSaTcNMHNLGNuLZqjGKbSrEMZets);
        auto __Local__7 =
NewObject<USceneComponent>(InDynamicClass,
USceneComponent::StaticClass(),
TEXT("DefaultSceneRoot_GEN_VARIABLE"), (EObjectFlags)0x00280029);
        InDynamicClass->ComponentTemplates.Add(__Local__7);
        auto __Local__8 =
NewObject<UComponentDelegateBinding>(InDynamicClass,
UComponentDelegateBinding::StaticClass(),
TEXT("ComponentDelegateBinding_1"), (EObjectFlags)0x00000000);
        InDynamicClass->DynamicBindingObjects.Add(__Local__8);
        static TWeakObjectPtr<UPROPERTY> __Local__10{};

```

```

    const UProperty* __Local__9 = __Local__10.Get();
    if (nullptr == __Local__9){
        __Local__9 = (UActorComponent::StaticClass())-
>FindPropertyByName(FName(TEXT("bCanEverAffectNavigation"))));
        check(__Local__9);
        __Local__10 = __Local__9;}
    (((UBoolProperty*)__Local__9)-
>SetPropertyValue_InContainer((__Local__7), false, 0));
    __Local__8->ComponentDelegateBindings =
TArray<FBlueprintComponentDelegateBinding> ();
    __Local__8->ComponentDelegateBindings.AddUninitialized(2);
    FBlueprintComponentDelegateBinding::StaticStruct()-
>InitializeStruct(__Local__8-
>ComponentDelegateBindings.GetData(), 2);
    auto& __Local__11 = __Local__8-
>ComponentDelegateBindings[0];
    __Local__11.ComponentPropertyName = FName(TEXT("Radius"));
    __Local__11.DelegatePropertyName =
FName(TEXT("OnComponentEndOverlap"));
    __Local__11.FunctionNameToBind =
FName(TEXT("BndEvt__Radius_K2Node_ComponentBoundEvent_1_Component
EndOverlapSignature__DelegateSignature"));
    auto& __Local__12 = __Local__8-
>ComponentDelegateBindings[1];
    __Local__12.ComponentPropertyName = FName(TEXT("Radius"));
    __Local__12.DelegatePropertyName =
FName(TEXT("OnComponentBeginOverlap"));
    __Local__12.FunctionNameToBind =
FName(TEXT("BndEvt__Radius_K2Node_ComponentBoundEvent_0_Component
BeginOverlapSignature__DelegateSignature"));}
PRAGMA_ENABLE_OPTIMIZATION
void
AInteraction_C_pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_0(int32 bpp__EntryPoint__pf){
    UUserWidget*
bpfv__CallFunc_GLT1Kqob5UDEML61gCyjnAcfMXgkdP3wGcg{};
    check(bpp__EntryPoint__pf == 23);
    if(::IsValid(bpv__InteractUI__pf)){
        bpfv__CallFunc_GLT1Kqob5UDEML61gCyjnAcfMXgkdP3wGcg =
bpv__InteractUI__pf->UWidgetComponent::GetUserWidgetObject();}
    b0l__K2Node_DynamicCast_AsInteract_UI__pf =
Cast<UInteractUI_C_pf4034160924>(bpfv__CallFunc_GLT1Kqob5UDEML61
gCyjnAcfMXgkdP3wGcg);
    b0l__K2Node_DynamicCast_bSuccess4__pf =
(b0l__K2Node_DynamicCast_AsInteract_UI__pf != nullptr);
    if (!b0l__K2Node_DynamicCast_bSuccess4__pf){
        return; //KCST_GotoReturnIfNot}

```

```

        bpv__InrteractUI__pf =
b01__K2Node_DynamicCast_AsInteract_UI__pf;
        if(::IsValid(bpv__InrteractUI__pf)){
            bpv__InrteractUI__pf-
>bpf__SetText__pf(bpv__TextWidget__pf);}
        return; // KCST_GotoReturn}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_1(int32 bpp__EntryPoint__pf){
    int32 __CurrentState = bpp__EntryPoint__pf;
    do{
        switch( __CurrentState ){
            case 13: {}
            case 14: {

                b01__K2Node_DynamicCast_AsALS_Base_Character__pf =
Cast<AALS_BaseCharacter_C__pf1533377440>(b01__K2Node_ComponentBou
ndEvent_OtherActor1__pf);
                b01__K2Node_DynamicCast_bSuccess__pf =
(b01__K2Node_DynamicCast_AsALS_Base_Character__pf != nullptr);
                if
(!b01__K2Node_DynamicCast_bSuccess__pf){
                    __CurrentState = -1;
                    break;}}

                case 15: {
                    if (
b01__K2Node_DynamicCast_AsALS_Base_Character__pf &&
b01__K2Node_DynamicCast_AsALS_Base_Character__pf->GetClass()-
>ImplementsInterface(UInterfaceBP_C::StaticClass()) ){

                        b01__K2Node_DynamicCast_AsInterface_BP__pf.SetObject(b01__K
2Node_DynamicCast_AsALS_Base_Character__pf);
                        void* IAddress =
b01__K2Node_DynamicCast_AsALS_Base_Character__pf-
>GetInterfaceAddress(UInterfaceBP_C::StaticClass());

                        b01__K2Node_DynamicCast_AsInterface_BP__pf.SetInterface(IAd
dress);}

                        else{

                            b01__K2Node_DynamicCast_AsInterface_BP__pf.SetObject(nullpt
r);}

                            b01__K2Node_DynamicCast_bSuccess2__pf =
(b01__K2Node_DynamicCast_AsInterface_BP__pf != nullptr);
                            if
(!b01__K2Node_DynamicCast_bSuccess2__pf){
                                __CurrentState = 17;
                                break;}}

```

```

        case 16:{
            if(::IsValid(b01__K2Node_DynamicCast_AsInterface_BP__pf)){
                IInterfaceBP_C::Execute_bpf__RegInteract__pf(b01__K2Node_Dy
namicCast_AsInterface_BP__pf.GetObject() , this);}}
            case 17:{
                bpf__SetVisibility__pf(true);}
            case 18:{
                if(::IsValid(b01__K2Node_DynamicCast_AsALS_Base_Character__
pf)){
                    b01__K2Node_DynamicCast_AsALS_Base_Character__pf->bpv__ChekIfThisDoor__pf = true;}
                    __CurrentState = -1;
                    break;}
            default:
                break;}} while( __CurrentState != -1 );}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_2(int32 bpp__EntryPoint__pf){
    int32 __CurrentState = bpp__EntryPoint__pf;
    do{
        switch( __CurrentState ){
            case 1:{
                bpf__SetVisibility__pf(false);}

            case 2:{

                if(::IsValid(b01__K2Node_DynamicCast_AsALS_Base_Character1_
__pf)){
                    b01__K2Node_DynamicCast_AsALS_Base_Character1__pf->bpv__ChekIfThisDoor__pf = false;}
                    __CurrentState = -1;
                    break;}

                case 19: {}
                case 20: {
                    b01__K2Node_DynamicCast_AsALS_Base_Character1__pf =
Cast<AALS_BaseCharacter_C__pf1533377440>(b01__K2Node_ComponentBou
ndEvent_OtherActor__pf);
                    b01__K2Node_DynamicCast_bSuccess1__pf =
(b01__K2Node_DynamicCast_AsALS_Base_Character1__pf != nullptr);;
                    if
(!b01__K2Node_DynamicCast_bSuccess1__pf){
                        __CurrentState = -1;
                        break;}}

                case 21: {
                    if (
b01__K2Node_DynamicCast_AsALS_Base_Character1__pf &&
b01__K2Node_DynamicCast_AsALS_Base_Character1__pf->GetClass()-
>ImplementsInterface(UInterfaceBP_C::StaticClass()) ){

```

```

        b01__K2Node_DynamicCast_AsInterface_BP1__pf.SetObject(b01__
K2Node_DynamicCast_AsALS_Base_Character1__pf);
        void* IAddress =
b01__K2Node_DynamicCast_AsALS_Base_Character1__pf-
>GetInterfaceAddress(UInterfaceBP_C::StaticClass());
        b01__K2Node_DynamicCast_AsInterface_BP1__pf.SetInterface(IA
ddress);}

        else{
            b01__K2Node_DynamicCast_AsInterface_BP1__pf.SetObject(nullp
tr);}

            b01__K2Node_DynamicCast_bSuccess3__pf =
(b01__K2Node_DynamicCast_AsInterface_BP1__pf != nullptr);
            if
(!b01__K2Node_DynamicCast_bSuccess3__pf){
                __CurrentState = 1;
                break;}}

        case 22:{

            if(::IsValid(b01__K2Node_DynamicCast_AsInterface_BP1__pf)){
                IInterfaceBP_C::Execute_bpf__UnregInteract__pf(b01__K2Node_
DynamicCast_AsInterface_BP1__pf.GetObject() , this);}
                __CurrentState = 1;
                break;}

            default:
                break;}} while( __CurrentState != -1 );}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_3(int32 bpp__EntryPoint__pf){
    check(bpp__EntryPoint__pf == 12);
    return; // KCST_GotoReturn}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_4(int32 bpp__EntryPoint__pf){
    check(bpp__EntryPoint__pf == 11);
    return; // KCST_GotoReturn}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_5(int32 bpp__EntryPoint__pf){
    check(bpp__EntryPoint__pf == 10);
    return; // KCST_GotoReturn}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_6(int32 bpp__EntryPoint__pf){
    check(bpp__EntryPoint__pf == 9);
    return; // KCST_GotoReturn}

```

```

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_7(int32 bpp__EntryPoint__pf){
    check(bpp__EntryPoint__pf == 8);
    return; // KCST_GotoReturn}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_8(int32 bpp__EntryPoint__pf){
    check(bpp__EntryPoint__pf == 7);
    return; // KCST_GotoReturn}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_9(int32 bpp__EntryPoint__pf){
    check(bpp__EntryPoint__pf == 6);
    return; // KCST_GotoReturn}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_10(int32 bpp__EntryPoint__pf){
    check(bpp__EntryPoint__pf == 5);
    return; // KCST_GotoReturn}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_11(int32 bpp__EntryPoint__pf){
    check(bpp__EntryPoint__pf == 4);
    return; // KCST_GotoReturn}

void
AInteraction_C__pf4034160924::bpf__ExecuteUbergraph_Interaction__
pf_12(int32 bpp__EntryPoint__pf){
    check(bpp__EntryPoint__pf == 3);
    return; // KCST_GotoReturn}

void AInteraction_C__pf4034160924::bpf__ReceiveBeginPlay__pf(){
    bpf__ExecuteUbergraph_Interaction__pf_0(23);}

void
AInteraction_C__pf4034160924::bpf__BndEvt__Radius_K2Node_Componen
tBoundEvent_1_ComponentEndOverlapSignature__DelegateSignature__pf
(UPrimitiveComponent* bpp__OverlappedComponent__pf, AActor*
bpp__OtherActor__pf, UPrimitiveComponent* bpp__OtherComp__pf,
int32 bpp__OtherBodyIndex__pf){
    b0l__K2Node_ComponentBoundEvent_OverlappedComponent__pf =
bpp__OverlappedComponent__pf;
    b0l__K2Node_ComponentBoundEvent_OtherActor__pf =
bpp__OtherActor__pf;
    b0l__K2Node_ComponentBoundEvent_OtherComp__pf =
bpp__OtherComp__pf;
    b0l__K2Node_ComponentBoundEvent_OtherBodyIndex__pf =
bpp__OtherBodyIndex__pf;
    bpf__ExecuteUbergraph_Interaction__pf_2(19);}

```



```

void
AInteraction_C__pf4034160924::bpf__BndEvt__Radius_K2Node_Componen
tBoundEvent_0_ComponentBeginOverlapSignature__DelegateSignature__
pf(UPrimitiveComponent* bpp__OverlappedComponent__pf, AActor*
bpp__OtherActor__pf, UPrimitiveComponent* bpp__OtherComp__pf,
int32 bpp__OtherBodyIndex__pf, bool bpp__bFromSweep__pf,
FHitResult const& bpp__SweepResult__pf__const){
    typedef FHitResult T__Local__13;
    T__Local__13& bpp__SweepResult__pf =
*const_cast<T__Local__13 *>(&bpp__SweepResult__pf__const);
    b0l__K2Node_ComponentBoundEvent_OverlappedComponent1__pf =
bpp__OverlappedComponent__pf;
    b0l__K2Node_ComponentBoundEvent_OtherActor1__pf =
bpp__OtherActor__pf;
    b0l__K2Node_ComponentBoundEvent_OtherComp1__pf =
bpp__OtherComp__pf;
    b0l__K2Node_ComponentBoundEvent_OtherBodyIndex1__pf =
bpp__OtherBodyIndex__pf;
    b0l__K2Node_ComponentBoundEvent_bFromSweep__pf =
bpp__bFromSweep__pf;
    b0l__K2Node_ComponentBoundEvent_SweepResult__pf =
bpp__SweepResult__pf;
    bpf__ExecuteUbergraph_Interaction__pf_1(13);}
void
AInteraction_C__pf4034160924::bpf__UnregAI__pf(AParentAI_C__pf379
577716* bpp__AI__pf){
    b0l__K2Node_Event_AI__pf = bpp__AI__pf;
    bpf__ExecuteUbergraph_Interaction__pf_3(12);}
void
AInteraction_C__pf4034160924::bpf__RegAI__pf(AParentAI_C__pf37957
7716* bpp__AI__pf){
    b0l__K2Node_Event_AI1__pf = bpp__AI__pf;
    bpf__ExecuteUbergraph_Interaction__pf_4(11);}
void
AInteraction_C__pf4034160924::bpf__UnregInteract__pf(AInteraction
_C__pf4034160924* bpp__Interaction__pf){
    b0l__K2Node_Event_Interaction__pf = bpp__Interaction__pf;
    bpf__ExecuteUbergraph_Interaction__pf_5(10);}
void
AInteraction_C__pf4034160924::bpf__RegInteract__pf(AInteraction_C
__pf4034160924* bpp__Interaction__pf){
    b0l__K2Node_Event_Interaction1__pf = bpp__Interaction__pf;
    bpf__ExecuteUbergraph_Interaction__pf_6(9);}
void AInteraction_C__pf4034160924::bpf__pick__pf(){
    bpf__ExecuteUbergraph_Interaction__pf_7(8);}
void
AInteraction_C__pf4034160924::bpf__Pickup__pf(FItemStructure__pf4
285404021 bpp__Iteminfo__pf){

```

```

        b01__K2Node_Event_Iteminfo__pf = bpp__Iteminfo__pf;
        bpf__ExecuteUbergraph_Interaction__pf_8(7);}
void
AInteraction_C__pf4034160924::bpf__Interaction__pf(AALS_BaseCharacter_C__pf1533377440* bpp__Infestigator__pf){
    b01__K2Node_Event_Infestigator__pf = bpp__Infestigator__pf;
    bpf__ExecuteUbergraph_Interaction__pf_9(6);}
void
AInteraction_C__pf4034160924::bpf__UnRegisterInteract__pf(APickUp_BP_C__pf4034160924* bpp__Interactable__pf){
    b01__K2Node_Event_Interactable__pf = bpp__Interactable__pf;
    bpf__ExecuteUbergraph_Interaction__pf_10(5);}
void
AInteraction_C__pf4034160924::bpf__RegisterInteract__pf(APickUp_BP_C__pf4034160924* bpp__Interactable__pf){
    b01__K2Node_Event_Interactable1__pf =
bpp__Interactable__pf;
    bpf__ExecuteUbergraph_Interaction__pf_11(4);}
void AInteraction_C__pf4034160924::bpf__UseItem__pf(int32
bpp__AmountToUse__pf, int32 bpp__Healpoint__pf, int32
bpp__StaminaPoint__pf){
    b01__K2Node_Event_AmountToUse__pf = bpp__AmountToUse__pf;
    b01__K2Node_Event_Healpoint__pf = bpp__Healpoint__pf;
    b01__K2Node_Event_StaminaPoint__pf = bpp__StaminaPoint__pf;
    bpf__ExecuteUbergraph_Interaction__pf_12(3);}
void
AInteraction_C__pf4034160924::bpf__UserConstructionScript__pf(){
    bpf__SetupMesh__pf();
    bpf__SetTrigger__pf();
    bpf__SetupUI__pf();}
void AInteraction_C__pf4034160924::bpf__setHighlight__pf(bool
bpp__Hightlight__pf){
    if(::IsValid(bpv__Mesh__pf)){
        bpv__Mesh__pf-
>UPrimitiveComponent::SetRenderCustomDepth(bpp__Hightlight__pf);}
}
void AInteraction_C__pf4034160924::bpf__SetupMesh__pf(){
    bool bpfv__CallFunc_SetStaticMesh_ReturnValue__pf{};
    if(::IsValid(bpv__Mesh__pf)){
        bpfv__CallFunc_SetStaticMesh_ReturnValue__pf =
bpv__Mesh__pf->SetStaticMesh(bpv__MeshNew__pf);}
    if(::IsValid(bpv__Mesh__pf)){
        bpv__Mesh__pf->SetMaterial(0, bpv__SetMat__pf);}
    if(::IsValid(bpv__Mesh__pf)){
        bpv__Mesh__pf->SetMaterial(0, bpv__SetMat__pf);}}
void AInteraction_C__pf4034160924::bpf__SetTrigger__pf(){
    FHitResult
bpfv__CallFunc_K2_SnCEH1C3UXSaTcNMHNLGNuLZqjGKbSrEMZ__pf{};

```

```

        if(::IsValid(bpv__Radius__pf)){
            bpv__Radius__pf-
>USphereComponent::SetSphereRadius(bpv__TrigerRadius__pf, true);}
        if(::IsValid(bpv__Radius__pf)){
            bpv__Radius__pf-
>USceneComponent::K2_SetRelativeLocation(bpv__triggerLocation__pf
, false, /*out*/
bpfv__CallFunc_K2_SnCEH1C3UXSaTcNMHNLGNuLZqjGKbSrEMZ__pf,
false);}
void AInteraction_C__pf4034160924::bpf__SetupUI__pf(){
    FHitResult
bpfv__CallFunc_K2_SnCEH1C3UXSaTcNMHNLGNuLZqjGKbSrEMZ__pf{};
    if(::IsValid(bpv__InteractUI__pf)){
        bpv__InteractUI__pf-
>USceneComponent::K2_SetRelativeLocation(bpv__UiLocation__pf,
false, /*out*/
bpfv__CallFunc_K2_SnCEH1C3UXSaTcNMHNLGNuLZqjGKbSrEMZ__pf,
false);}
void AInteraction_C__pf4034160924::bpf__SetVisibility__pf(bool
bpp__Visibility__pf){
    if(::IsValid(bpv__InteractUI__pf)){
        bpv__InteractUI__pf-
>USceneComponent::SetVisibility(bpp__Visibility__pf, false);}
void
AInteraction_C__pf4034160924::bpf__Interact__pf(AALS_BaseCharacte
r_C__pf1533377440* bpp__player__pf){
    bpv__Player__pf = bpp__player__pf;}
PRAGMA_DISABLE_OPTIMIZATION
void
AInteraction_C__pf4034160924::__SnCEH1C3UXSaTcNMHNLGNuLZqjGKbSrEM
Zets(TArray<FBlueprintDependencyData>& AssetsToLoad){
    const FCompactBlueprintDependencyData
LT1Kqob5UDEML61gCyjnAcfMXgkdP3wGcg] ={
        {0, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh /Engine/EngineMeshes/Cube.Cube
        {1, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh /Engine/BasicShapes/Cube1.Cube1 };
    for(const FCompactBlueprintDependencyData& CompactData :
LocCompactBlueprintDependencyData){

        AssetsToLoad.Add(FBlueprintDependencyData(F__NativeDependen
cies::Get(CompactData.ObjectRefIndex), CompactData));}
PRAGMA_ENABLE_OPTIMIZATION
PRAGMA_DISABLE_OPTIMIZATION

```

```

void
AInteraction_C__pf4034160924::__StaticDependenciesAssets(TArray<F
BlueprintDependencyData>& AssetsToLoad){
    __SNcEH1C3UXSaTcNMHNLGNuLZqjGKbSrEMZets(AssetsToLoad);
    const FCompactBlueprintDependencyData
LT1Kqob5UDEML61gCyjnAcfMXgkdP3wGcg] ={
        {2, FBlueprintDependencyType(true, false, false,
false), FBlueprintDependencyType(false, false, false, false)},
// BlueprintGeneratedClass
/Game/Inventory/Item/Interface/InterfaceBP.InterfaceBP_C
        {3, FBlueprintDependencyType(true, false, false,
false), FBlueprintDependencyType(false, false, false, false)},
// BlueprintGeneratedClass
/Game/Inventory/Item/Interface/InventInterface.InventInterface_C
        {4, FBlueprintDependencyType(true, false, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.StaticMeshComponent
        {5, FBlueprintDependencyType(true, false, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.SphereComponent
        {6, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/NavigationSystem.NavArea_Obstacle
        {7, FBlueprintDependencyType(true, false, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.WidgetComponent
        {8, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// MaterialInstanceConstant
/Engine/EngineMaterials/Widget3DPassThrough_Translucent.Widget3DP
assThrough_Translucent
        {9, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// MaterialInstanceConstant
/Engine/EngineMaterials/Widget1JZ7MuMn6D1B8ujUbHEkpC76sK6tG5ic81.
Widget1JZ7MuMn6D1B8ujUbHEkpC76sK6tG5ic81
        {10, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// MaterialInstanceConstant
/Engine/EngineMaterials/Widget3DPassThrough_Opaque.Widget3DPassTh
rough_Opaque
        {11, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// MaterialInstanceConstant
/Engine/EngineMaterials/Widget3DPassThrough_Opaque_OneSided.Widge
t3DPassThrough_Opaque_OneSided
        {12, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},

```

```

// MaterialInstanceConstant
//Engine/EngineMaterials/Widget3DPassThrough_Masked.Widget3DPassTh
rough_Masked
    {13, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// MaterialInstanceConstant
//Engine/EngineMaterials/Widget3DPassThrough_Masked_OneSided.Widge
t3DPassThrough_Masked_OneSided
    {14, FBlueprintDependencyType(true, false, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.SceneComponent
    {15, FBlueprintDependencyType(true, false, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/Engine.PointerToUberGraphFrame
    {16, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.PrimitiveComponent
    {17, FBlueprintDependencyType(true, false, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.Actor
    {18, FBlueprintDependencyType(true, false, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/Engine.HitResult
    {19, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.UserWidget
    {20, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.StaticMesh
    {21, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.MaterialInterface
    {22, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/Milk.Milk
    {23, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.Character
    {24, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.KismetMathLibrary
    {25, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.GameplayStatics
    {26, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.KismetSystemLibrary

```

```

        {27, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/Engine.LatentActionInfo
        {28, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.KismetTextLibrary
        {29, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.Texture2D
        {30, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh /Game/StarterContent/Food/Bread/Bread.Bread
        {31, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh
/Game/StarterContent/Food/Chocolate/chocolate.chocolate
        {32, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh /Game/StarterContent/Food/Citron/Citron.Citron
        {33, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh /Game/StarterContent/Food/Milk/Milk.Milk
        {34, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D
/Game/Inventory/Item/Texture/MonsterEnergy.MonsterEnergy
        {35, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh
/Game/StarterContent/Food/MonsterDrink/DrinkEnergy.DrinkEnergy
        {36, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D
/Game/Inventory/Item/Texture/rotiPanjang.rotiPanjang
        {37, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh
/Game/StarterContent/Food/newBread/Newbread.Newbread
        {38, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/RotiLapis.RotiLapis
        {39, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh
/Game/StarterContent/Food/SandWitch/sandwich.sandwich
        {40, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/Pepsi.Pepsi

```

```

        {41, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh
/Game/StarterContent/Food/Pepsi/Pepsi_Can.Pepsi_Can
        {42, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/Proposal.Proposal
        {43, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh
/Game/StarterContent/Architecture/House/Home/Kertas.Kertas
        {44, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/Laporan.Laporan
        {45, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D
/Game/AdvancedLocomotionV3/Characters/HUDAsset/Kwitansi.Kwitansi
        {46, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// StaticMesh
/Game/StarterContent/Architecture/House/Home/paper.paper
        {47, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/Misi.Misi
        {48, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/BreadBulat.BreadBulat
        {49, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D
/Game/Inventory/Item/Texture/ChokolateShake.ChokolateShake
        {50, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/Citron.Citron
        {51, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.WidgetBlueprintLibrary
        {52, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Font /Engine/EngineFonts/Roboto.Roboto
        {53, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Betul2.Betul2
        {54, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.PlayerController

```

```

        {55, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.Image
        {56, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Betul.Betul
        {57, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.Button
        {58, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.ArrowComponent
        {59, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.SpringArmComponent
        {60, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.CameraComponent
        {61, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.AnimationInstance
        {62, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.AnimationMontage
        {63, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Enum /Script/Engine.EMovementMode
        {64, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/InputCore.Key
        {65, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.CurveFloat
        {66, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.Controller
        {67, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.CameraShake
        {68, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.GameInstance
        {69, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// SoundWave /Game/SFX/Suces.Suces
        {70, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.CharacterMovementComponent

```



```

        {71, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.SkeletalMeshComponent
        {72, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.KismetArrayLibrary
        {73, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.MovementComponent
        {74, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// SkeletalMesh /Game/MainChara/Cloth.Cloth
        {75, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// SkeletalMesh /Game/MainChara/MianChara.MianChara
        {76, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// SoundWave /Game/SFX/ATMPrint.ATMPrint
        {77, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// SoundWave /Game/SFX/Printer.Printer
        {78, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// SoundWave /Game/SFX/Pickup.Pickup
        {79, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// SoundWave /Game/SFX/MainMenu.MainMenu
        {80, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.KismetStringLibrary
        {81, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.CurveBase
        {82, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.TimelineComponent
        {83, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.Pawn
        {84, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.PhysicalMaterial
        {85, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.SaveGame
        {86, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Enum /Script/Engine.EDrawDebugTrace

```

```

        {87, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.CapsuleComponent
        {88, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.KismetNodeHelperLibrary
        {89, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// CurveFloat
/Game/AdvancedLocomotionV3/Blueprints/Curves/CameraLerpCurves/Cam
eraLerp_3.CameraLerp_3
        {90, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// CurveFloat
/Game/AdvancedLocomotionV3/Blueprints/Curves/CameraLerpCurves/Cam
eraLerp_1.CameraLerp_1
        {91, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// CurveFloat
/Game/AdvancedLocomotionV3/Blueprints/Curves/CameraLerpCurves/Cam
eraLerp_2.CameraLerp_2
        {92, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.PawnMovementComponent
        {93, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Enum /Script/Engine.ETimelineDirection
        {94, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class
/Script/ClothingSystemRuntime.ClothingSimulationFactoryNv
        {95, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/AIModule.AIController
        {96, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/SlateCore.PointerEvent
        {97, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Enum /Script/UMG.ESlateVisibility
        {98, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.UMGSequencePlayer
        {99, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.WidgetAnimation
        {100, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},

```

```

// Texture2D
/Game/AdvancedLocomotionV3/Blueprints/AI/Texture/DialogTexture.DialogTexture
    {101, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// DelegateFunction
/Script/Engine.TimerLQMLrMmvrva5skwyM4QzDUoGQJGhaovqyg
    {102, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/DialoguePlugin.DialogueNode
    {103, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.Widget
    {104, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/SlateCore.SlateColor
    {105, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/Engine.TimerHandle
    {106, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.SoundBase
    {107, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.AudioComponent
    {108, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.VerticalBoxSlot
    {109, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// SoundWave /Game/SFX/Laptop.Laptop
    {110, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.VerticalBox
    {111, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.PanelWidget
    {112, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.Overlay
    {113, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.TextBlock
    {114, FBlueprintDependencyType(false, true, false, false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/DialoguePlugin.Dialogue

```

```

        {115, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/DialoguePlugin.DialogueUserWidget
        {116, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/SlateCore.Geometry
        {117, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/SlateCore.KeyEvent
        {118, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/UMG.EventReply
        {119, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/Engine.KismetInputLibrary
        {120, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/ImageStack.ImageStack
        {121, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// SkeletalMesh /Game/PaYusuf/Pembimbing.Pembimbing
        {122, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Skeleton /Game/PaYusuf/PaYusuf_Skeleton.PaYusuf_Skeleton
        {123, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/AnimGraphRuntime.AnimNode_Root
        {124, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct
/Script/AnimGraphRuntime.AnimNode_BlendSpacePlayer
        {125, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// BlendSpace1D /Game/PaYusuf/BSPYUSUF.BSPYUSUF
        {126, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/LSFTI.LSFTI
        {127, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/LSLFTI.LSLFTI
        {128, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.ProgressBar
        {129, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D
/Game/Inventory/Item/Texture/ImageInteract.ImageInteract

```

```

        {130, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Class /Script/UMG.CanvasPanel
        {131, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/LSUnibba.LSUnibba
        {132, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/LS.LS
        {133, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// ScriptStruct /Script/SlateCore.SlateBrush
        {134, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D
/Game/AdvancedLocomotionV3/Characters/HUDAsset/BarRP.BarRP
        {135, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D
/Game/AdvancedLocomotionV3/Characters/HUDAsset/Uang.Uang
        {136, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D
/Game/AdvancedLocomotionV3/Characters/HUDAsset/BarBG.BarBG
        {137, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D
/Game/AdvancedLocomotionV3/Characters/HUDAsset/HungerBar.HungerBar
        {138, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// Texture2D /Game/Inventory/Item/Texture/Unibba150.Unibba150
        {139, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// WidgetBlueprintGeneratedClass
/Game/Inventory/Item/InteractUI.InteractUI_C
        {140, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// BlueprintGeneratedClass
/Game/Inventory/Item/PickUp_BP.PickUp_BP_C
        {141, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},
// BlueprintGeneratedClass
/Game/AdvancedLocomotionV3/Blueprints/ALS_BaseCharacter.ALS_BaseCharacter_C
        {142, FBlueprintDependencyType(false, true, false,
false), FBlueprintDependencyType(false, false, false, false)},

```

```

// BlueprintGeneratedClass
/Game/AdvancedLocomotionV3/Blueprints/AI/ParentAI.ParentAI_C
    {143, FBlueprintDependencyType(true, false, false,
false), FBlueprintDependencyType(false, false, false, false)},
// UserDefinedStruct /Game/Inventory/ItemStructure.ItemStructure
};

    for(const FCompactBlueprintDependencyData& CompactData :
LocCompactBlueprintDependencyData){
        AssetsToLoad.Add(FBlueprintDependencyData(F__NativeDependen
cies::Get(CompactData.ObjectRefIndex), CompactData));}}
PRAGMA_ENABLE_OPTIMIZATION
struct FRegisterHelper__AInteraction_C__pf4034160924{
    FRegisterHelper__AInteraction_C__pf4034160924(){
        FConvertedBlueprintsDependencies::Get().RegisterConvertedCl
ass(TEXT("/Game/Inventory/Item/Interaction"),
&AInteraction_C__pf4034160924::__StaticDependenciesAssets);}
    static FRegisterHelper__AInteraction_C__pf4034160924
Instance;};
FRegisterHelper__AInteraction_C__pf4034160924
FRegisterHelper__AInteraction_C__pf4034160924::Instance;
PRAGMA_ENABLE_DEPRECATED_WARNINGS
#ifdef _MSC_VER
#pragma warning (pop)
#endif

```

RIWAYAT HIDUP PENULIS



Aggia Bintang Ramadhan Hermanto lahir di Bandung, pada 28 Januari 1997. Anak pertama dari dua bersaudara, yang terlahir dari pasangan Jojon Hermanto dan Ira Iyos Susilawati. Yang bertempat tinggal di KP.Cijagra RT03 RW04 KEC.Paseh KAB. Bandung. Memulai pendidikan di SD Negeri Lokasari I (2004-2009), kemudian melanjutkan pendidikan ke SMP Negeri 1 Paseh (2009-2012), serta melanjutkan masa SMK di SMK Negeri 1 Majalaya (2012-2015) dan mengambil jurusan elektronika industri. Untuk mendapatkan gelar sarjana penulis melanjutkan ke jenjang S1, penulis melanjutkan pendidikan di Universitas Bale Bandung Fakultas Teknologi Informasi Jurusan Teknik Informatika. Penulis pernah aktif sebagai anggota HIMA-IF sampai tahun 2018. Penulis juga pernah mengikuti kompetisi game dan mendapatkan juara harapan.