# SAMPUL

PERANCANGAN FRAMEWORK GAME VR MENGGUNAKAN GODOT ENGINE

**SKRIPSI**

**Oleh**

**Joel Robert Justiawan 2101629672**

**Anggara Aryamufti 2001578853**

**Kelas/Kelompok: LZ51/002**



**Computer Science Program**

**Game Application and Technology Study Program**

**School of Computer Science**

**Universitas Bina Nusantara**

**Jakarta**

**2019**

# HALAMAN JUDUL

**PERANCANGAN FRAMEWORK GAME VR MENGGUNAKAN GODOT ENGINE**

**SKRIPSI**

**diajukan sebagai salah satu syarat**

**untuk gelar kesarjanaan pada**

**Program Studi Teknik Informatika**

**Jenjang Pendidikan Strata-1**

**Oleh**

**Joel Robert Justiawan 2101629672**

**Anggara Aryamufti 2001578853**

**Kelas/Kelompok: LZ51/002**



**Computer Science Program**

**Computer Science Study Program**

**School of Computer Science**

**Universitas Bina Nusantara**

**Jakarta**

**2019**

# BAB 1 PENDAHULUAN

## LATAR BELAKANG MASALAH

Perancangan sebuah *game* dilakukan melalui sebuah pemrograman yang diawali dengan membuat *framework* sebagai mesin tempat beroperasinya *game* tersebut. Setelah *framework* nya ada, dilengkapi dengan *engine*, dan perancangan *game* di lanjutkan dengan melengkapi fitur-fitur dalam *framework* nya sesuai jenis *game* yang dirancang. Untuk membuat *framework game* memerlukan waktu yang lama dan cukup rumit. Dari permasalahan ini, penulis bermaksud membuat aplikasi *framework* yang dapat digunakan oleh para *designer game* untuk merancang *game* nya sehingga akan mempersingkat waktu bagi para *designer* dan *developer* *game* saat membangun dan memodifikasi *game* nya.

Bagan 1 Beberapa Game yang dibuat menggunakan framework bernama Corgi Engine dari More Mountains

Pada gambar 1.1, adalah contoh *game* yang dibuat dengan *framework* bernama *Corgi Engine* dari perusahaan *More Mountains*. Pengembang *game* hanya perlu memasang *framework* tersebut, kemudian mengembangkan dan menambahkan gambar, suara, dan mekanisme dari *game* tersebut. Dari sisi nilai ekonomis dan efisiensi, penggunaan *framework* yang sudah siap digunakan dapat mempersingkat waktu perancangan. Data statistik Pengguna *framework* di kalangan *designer game* menunjukkan

Tidak banyak perusahaan yang membuat *framework game* yang siap digunakan baik hobi maupun profesional untuk para pengembang *game*. Beberapa perusahaan yang membuat *framework* diantaranya:

* More Mountains: Corgi Engine, Top Down Engines
* MonoGames
* libGDX
* Ogre
* SFML

Dari data *framework* diatas tersebut, belum ada yang menyediakan fitur *Virtual Reality* di dalam *framework* tersebut. Hal ini dikarenakan sebagai berikut:

* Komputasi yang dibutuhkan cenderung tinggi
* Teknologi relatif masih baru dan belum menjadi satu bidang yang di minati
* Kurangnya sosialisasi & pemasaran di bidang *virtual reality*.
* Minimnya niat dan minimnya jumlah pasaran terkait *virtual reality*
* Sulitnya proses implementasi, pembuatan, dan alat-alat yang tersedia untuk mengembangkan game *virtual reality*
* Rumitnya proses & *framework plugin* yang ada yang dibutuhkan untuk membuat *virtual reality*
* Kurangnya pengetahuan di bidang *virtual reality*

*Virtual Reality* merupakan teknologi baru yang membuat permainan *game* lebih menarik dan imersif karena para pemain akan merasakan permainan seperti nyata. Sehingga *VR* lebih di cari oleh para penggemar *game*. Oleh karena itu sudah seharusnya para *designer game* mengembangkan *game* nya dengan aplikasi *VR*. Berdasarkan kenyataan inilah penulis bermaksud membuat *framework* yang dilengkapi dengan fitur *VR*. Sehingga diharapkan *framework* yang didesain menjadi sebuah keunggulan dari desain framework penulis, karena memiliki fitur *VR*.:



[Foto Ini](https://en.wikipedia.org/wiki/Virtual_reality) oleh Penulis Tidak Diketahui dilisensikan atas nama[CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

Bagan 2 Seseorang sedang bermain sebuah permainan virtual reality

Aplikasi *framework* yang dibuat penulis akan memudahkan *designer game* karena para desainer *game* hanya tinggal memodifikasi dari *framework* tersebut dan menambahkan fitur-fitur yang diinginkan.*Framework game* sebenarnya saat ini ada disediakan di pasaran, contohnya Corgi (More Mountains, t.thn.), Topdown (More Mountains, t.thn.) tetapi *framework-framework* ini berbayar dan memiliki aturan-aturan yang harus dipatuhi, sehingga ketika kita akan merancang sebuah *game*, kita harus memikirkan masalah *budget* dan *cost* pengeluaran untuk pembelian *framework* nya, aturan-aturan yang diberikan oleh vendor aplikasidanberlangganan *engine* yang akan di gunakan pada pembuatan *game* nya.serta *designer game* harus menghindari membuat *game* resmi yang akan dipublikasikan ke dalam repositori perangkat lunak *(app store)* yang berisi asset yang berhak cipta tanpa meminta ijin ke pemegang hak cipta tersebut karena akan terkena pelanggaran DMCA. Dari permasalahan inilah *framework game* yang dibuat oleh penulis adalah free sehingga dapat digunakan oleh para *designer* dan *developer game* dengan gratis, tanpa takut dikenai sanksi hak cipta dan tuntutan dari aturan aturan. *Framework* yang dibuat penulis akan dipublikasikan ke *app store* sehingga para *designer game* dapat dengan mudah memperolehnya dan mengaplikasikan nya tanpa harus mengeluarkan biaya sedikit pun serta dapat memasang, dan memodifikasi aset-aset nya misal musiknya tanpa takut melanggar hak cipta.

*Framework* yang akan dibuat diberi nama Hexagon Engine. *Framework* ini memanfaatkan *engine* dari Godot (Juan Linietsky, 2014), dimana dalam Godot *engine* hanya memiliki keterbatasan pada fitur dan aset, di dalam Godot *engine* adanya fitur standar dan asset standar. Oleh karena itu, di kembangkan dengan pembuatan *framework* ini sehingga memiliki fitur-fitur yang lebih lengkap lagi, seperti tersedianya template-template permainan, *Virtual Reality, platformer*, simulasi. Fitur-fitur yang ada dalam *framework* yang akan dibuat ini merupakan *Virtual Reality* dimana diantara nya adalah interaksi terhadap *item* (mengambil, mengutak-atik, mengubah), Gerakan badan kita (Pindah / teleportasi, jalan kaki / naik kendaraan), music, layar sentuh. *Framework* yang akan dibuat ini akan di publikasikan ke vendor aplikasi sehingga mempermudah *designer* dan *developer* *game* membuat rancangan *game* nya, yaitu hanya dengan mengunduh aplikasinya dan menggunakan fitur-fitur dalam rancangan *game* nya secara bebas. Diharapkan dengan adanya *framework* yang tidak berbayar ini akan membantu *designer* dan *developer* menciptakan game yang membantu banyak orang misalnya game pembelajaran, game simulasi untuk dunia Pendidikan dengan gratis.

Dari uraian diatas dapat di simpulkan bahwa pokok permasalahan dibuatnya rancangan framework game adalah :

1. Mempermudah *designer* *game* dalam membuat rancangan dan konstruksi *game* yang akan mereka buat
2. Mengurangi *cost* (biaya) pengeluaran *designer game*
3. Mempermudah kinerja *designer game* untuk mengembangkan *game* yang mereka buat
4. Mempercepat proses pembuatan *game* bagi para *developer* dan *designer game*
5. Menyediakan dukungan berbagai tipe permainan, cara bermain, jenis layar (misalnya *VR*)
6. Membantu dunia Pendidikan memperoleh *game* simulasi untuk proses beleajar mengajar secara gratis

## PERUMUSAN MASALAH

Berdasarkan latar belakang yang telah dikemukakan di atas, maka perumusan masalah dalam perancangan ini adalah;

1. Perlu adanya framework untuk pembuatan *game* yang tidak berbayar, agar para *designer game* lebih mudah dan efisien dalam mendesain *game* yang akan mereka buat
2. Kebebasan dalam menciptakan dan memodifikasi *game* ke dalam sebuah *framework*
3. Tidak ada nya tuntutan pelanggaran hak cipta dan tuntutan aturan-aturan pembuat *framework*

## BATASAN MASALAH

Agar skripsi ini lebih terarah pada tujuan perancangan, maka diberikan batasan-batasan sebagai berikut :

1. Output dari perancangan berupa sebuah *framework game*
2. Perancangan *framework* di sertai *sample game*
3. Hasil perancangan terbatas pada sebuah *framework game*

## TUJUAN PENELITIAN

Berdasarkan rumusan masalah yang ada, maka tujuan dari perancangan ini adalah:

1. Merancang *Framework game* bernama Hexagon Engine
2. Membantu para *designer game* untuk dapat menciptakan dan mendesain *game* nya sendiri dengan bebas karena Hexagon Engine dapat di unduh gratis.
3. Memanfaatkan loop hole supaya tidak terkena pelanggaran *copyright*

## MANFAAT PENELITIAN

Hasil perancangan ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Memberikan kebebasan bagi para *designer game* untuk mengekpresikan rancangan gamenya.
2. Meringankan *budget* dan *cost* pengeluaran pembuatan *game* bagi para *designer game* yang akan membuat *game*
3. Memperkenalkan dan memberikan informasi kepada masyarakat tentang *framework* pembuatan sebuah *game*.

***1.6 SISTEMATIKA PENULISAN***

Sistematika penulisan yang digunakan dalam penulisan skripsi ini adalah sebagai berikut :

BAB 1: PENDAHULUAN

Bab ini berisi latar belakang pembuatan framework game VR , ruang lingkup, rumusan masalah, tujuan dan manfaat, dan sistematika penulisan.

BAB 2: TINJAUAN PUSTAKA

Bab ini berisi kumpulan teori-teori yang berkaitan dengan framework, VR, engine game dan pemograman , menggunakan teori dari buku, artikel, dan jurnal yang mendukung.

BAB 3: METODE PENELITIAN

Bab ini berisi analisis aplikasi serupa yang sudah beredar, dan membahas tahap-tahap perancangan dan pengembangan aplikasi.

BAB 4: HASIL DAN PEMBAHASAN

Bab ini berisi proses pembutan framework, pemrograman framework, VR, pembuatan contoh-contoh game dalam framework dan hasil uji coba framework

BAB 5: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil penelitian.

# BAB 2 TINJAUAN PUSTAKA

Bagi para pembuat *game*, menciptakan sebuah *game* merupakan kegiatan utama yang dilakukan. Untuk menciptakan sebuah *game*, selain *budget* dan gagasan yang perlu disiapkan, sarana dan prasarana pendukung juga perlu diperhatikan. Salah satu faktor utama dalam menciptakan sebuah gama adalah *engine* penggereak atau media gamenya. Saat ini *engine – engine* penggerak, seperti *Unity* (Unity Technologies, 2005), *Unreal* (Epic Games, t.thn.), *Unigine* (Unigine, 2005), *Cryengine* (CryTek, 2002), *ClickTeam* (Clickteam, 1993), *Cocos* (Chukong Technologies, 2010), , *Godot* (Juan Linietsky, 2014), *Greenfoot* (Kölling, 2003), *GDevelop* (Rival, 2008), *Stride* (Silicon Studio, 2014), dan lain-lain. Seorang pembuat *game* ketika akan menciptakan *game* dari awal akan menggunakan salah satu dari *engine* tersebut dan melakukan pemrograman yang akan memerlukan waktu yang lama dan panjang. Selain hal tersebut, ada beberapa konsekuensi yang di hadapi yaitu untuk properti-properti yang akan diletakan di dalamnya seperti musik, gambar, kodingan, dan lain-lain tidak dapat sembarangan di aplikasikan karena terdapat ketentuan dan aturan yang ditetapkan dari vendor seperti *Google Play*, *Steam*, *App Store*, dan lain-lain.

Jika seorang pembuat *game* ingin mempercepat waktu pembuatan *game*, pembuat *game* dapat menggunakan *framework* yang disediakan yang dibangun diatas *engine-engine* tersebut. Beberapa *framework* itu sendiri ada yang berbayar, gratis, berlisensi sumber terbuka dan sumber tertutup. Dan umumnya, *framework-framework* di desain hanya untuk hal tertentu saja serta memiliki keterbatasan fitur.Berangkat dari kenyataan diatas tersebut, penulis bermaksud untuk merancang sebuah *framework* untuk pembuatan *game*, dimana di dalam framework tersebut akan menyediakan berbagain rancangan game sehingga akan mempersingkat waktu pembuat game dalam mencetuskan gagasan-gagasannya dalam membuat game. Framework yang dibuat ini dapat digunakan secara gratis oleh para pembuat game dan dalam pengoperasiannya, para pembuat game dapat menggunakan properti-properti apa saja secara bebas (dianggap modifikasi).

Salah satu keunggulan framework yang dapat dibuat oleh penulis adalah didalam framework ini terdapat modus *Virtual Reality*.

Perancangan framework game dilaksanakan dengan mempelajari framework game yang sudah ada dan membaca beberapa literatur yang telah di pelajari selama ini di perkuliahan serta mengembangkan framework untuk dapat dipakai dalam pembuatan game.

Dalam pembuatan framework game ini, engine yang digunakan oleh penulis adalah Godot. Framework yang dirancang oleh penulis diberi nama **Hexagon Engine**. Fitur – fitur dalam Hexagon Engine meliputi diantaranya: *Platformer*, Permainan serius (*Serious Game*), *Real time strategy*, Simulasi, Pendidikan & Ujian, dan sebagainya. Fitur – fitur yang ada inilah yang menjadi pembeda dari framework – framework lain yang sudah ada saat ini. Hexagon Engine memberikan wadah bagi para pembuat game untuk menciptakan game, mengeksplor game buatannya, serta dapat menambahkan properti – properti apapun itu secara bebas (Modifikasi).

## 2.1 FRAMEWORK

Framework didefinisikan sebagai kerangka kerja untuk mengembangkan aplikasi berbasis web, desktop, mobile dan sebagainya (Adani, 2020)

Framework game merupakan kerangka kerja untuk mengembangkan pembuatan game dan melengkapi game tersebut dengan fitur-fitur sesuai keinginan designer.

Fungsi dari Framework game adalah :

* + 1. Lebih terstruktur

Fungsi framework yang utama adalah membuat source code menjadi lebih terstruktur. Terstruktur disini, berarti program yang dibuat akan dimasukkan ke dalam setiap komponen sesuai dengan fungsi nya masing-masing.

Salah satu contoh dari kode program terstruktur dapat dilihat dari framework PHP, yaitu Laravel dalam pembuatan website yang menggunakan konsep paradigma MVC (Model, View, Controller). Terdapat tiga komponen utama untuk mengembangkan website menggunakan model framework tersebut.

Demikian juga fungsi framework game akan membuat kode program game tersusun rapi dengan tiga komponen utama untuk mengembangkan game menggunakan framework tersebut.

Model berfungsi untuk tempat atau wadah menampung kode program berupa algoritma pemrograman dan penghubung input output audio visual dan data base aplikasi.

View berfungsi sebagai wadah menampung kode program untuk membuat tampilan yang nantinya ditampilkan kepada customer / client.

Controller berfungsi untuk menghubungkan model dan view agar menjadi sebuah aplikasi game secara keseluruhan.

Dapat dikatakan juga, model disini untuk menangani tugas back end. Kemudian, view untuk menangani tugas front end. Dengan adanya konsep MVC tersebut, kode program akan tersusun rapi serta mempersingkat kerja dari developer.

* + 1. Membantu kinerja

Fungsi yang kedua adalah membantu kinerja dari developer sendiri. sebenarnya dalam membuat sebuah design game dapat dilakukan tanpa menggunakan bantuan framework. jika proyek yang ditangani dalam lingkup kecil. Tetapi jika proyek pembuatan game dalam skala untuk organisasi besar, maka diperlukan framework untuk memudahkan kinerja dari developer dalam segi efisiensi waktu serta resource yang dibutuhkan.

Selain itu, apabila dikerjakan oleh beberapa team maka framework akan sangat membantu efektivitas kerja dan sinkronisasi. Setiap team dapat langsung memahami dan mengerjakan tugas masing – masing tanpa perlu mengecek satu persatu komponen baik dari segi back end maupun front end.

* + 1. Meningkatkan keamanan

Selain membantu kinerja dari developer, dari segi keamanan aplikasi game juga akan meningkat. Karena dalam framework telah diidentifikasi oleh beberapa developer ahli dari berbagai negara.

Kemudian, sebuah kerangka kerja juga memiliki beberapa versi yang setiap saat akan selalu ada pembaruan dari segi fitur serta untuk mengurangi adanya bugs. Keamanan akan selalu diperbarui dan dimonitoring agar tidak terjadi permasalahan yang lebih serius pada aplikasi game misalnya terjadinya crash, glitch, dan sebagainya.

* + 1. Mempermudah pemeliharaan dan dokumentasi

Jika ingin menambahkan atau mengurangi beberapa fitur dari aplikasi game yang telah dibuat, maka dengan menggunakan framework dapat memudahkan untuk melakukan hal tersebut. dan dengan adanya framework memiliki fitur untuk melakukan maintenance pada program.

Dengan melakukan maintenance, dapat mengubah versi aplikasi game tersebut dan menambahkan beberapa fitur dengan lebih mudah dan aman. Selanjutnya, dari segi dokumentasi juga lebih terstruktur. Dalam kerangka kerja, setiap dokumen aplikasi yang dibangun dapat diidentifikasi dengan mudah dan cepat.

* + 1. Mempercepat proses pembuatan

Dengan adanya framework dapat mempercepat dalam proses pembuatan design game. Developer dapat mengembangkan aplikasi dengan menggunakan komponen – komponen yang telah tersedia dalam framework. Sehingga, tidak perlu untuk menyusun ulang dari awal kode program.

* + 1. Standarisasi fitur

Dalam framework juga terdapat berbagai fitur yang siap digunakan seperti modus game, template, modul, dan sebagainya. Jadi tidak perlu membuat fitur lagi dari awal. Setiap fitur standar dapat juga dimodifikasi sesuai kebutuhan.

Untuk framework ini, modus game terdiri dari beberapa jenis, yang setiap jenis memiliki sub-jenisnya masing2 sesuai tema dari game yang dibuat atas framework ini. Misalkan sebuah permainan platformer bertemakan sulap, maka jenis game tersebut adalah Platformer Sulap, atau secara resmi disebut *XTQN-Platformer-Mage*.

### CODING

Coding adalah proses pengolahan kode yang ditulis menggunakan bahasa tertentu (Dicoding, 2020).

Di era digital dan perkembangan teknologi yang pesat seperti sekarang ini. Profesi programmer atau developer perangkat lunak banyak menarik minat atau menjadi primadona sebagian orang. Perkembangan perangkat lunak atau aplikasi tidak terlepas dari para programmer yang mengembangkannya, dengan kata lain programmer lah yang meng coding aplikasi tersebut.

**2.2.1 Pengertian Coding**



*Coding* adalah salah satu tindakan dari langkah-langkah pemrograman dengan menuliskan kode atau skrip dalam bahasa pemrograman. Supaya skrip tersebut dapat dipahami oleh komputer, maka saat proses *coding* harus mengikuti aturan sintaks yang berlaku. Aturan sintaks sangat tergantung dari bahasa pemrograman yang digunakan saat menuliskan skrip.

Dengan kata lain *coding* merupakan kegiatan untuk memberitahukan komputer apa yang harus dia kerjakan. Sebuah kode yang ada pada skrip bisa diibaratkan layaknya bahasa sehari-hari. Setiap kode di tulis akan membantu komputer untuk mengetahui dan memahami apa yang ingin dilakukan pada komputer. Komputer akan menerima perintah ini dan komputer akan melakukan operasi berdasarkan perintah yang di tuliskan.

Aturan sintaks sangat penting untuk dipatuhi. Karena komputer merupakan mesin yang hanya mampu menerima kode atau perintah yang di masukkan. Apabila tidak menuliskan kode sesuai dengan aturan sintaks dan ada kode yang salah di tulis, maka perintah yang di tulis tidak dapat dijalankan oleh komputer.

* + 1. **Bahasa Pemrograman**

langkah pertama untuk pengcodingan adalah menentukan bahasa pemrograman yang akan dipakai. Tentukan bahasa pemrograman sesuai dengan tujuan dan apa yang akan di buat. Terdapat berbagai macam bahasa pemrograman, setiap bahasa pemrograman memiliki tujuan dan karakteristiknya masing-masing.

Misalnya akan memulai *coding* untuk membuat aplikasi android, maka bisa digunakan bahasa pemrograman Java ataupun Kotlin. Sedangkan jika akan memulai *coding* untuk membuat aplikasi web, maka bisa menggunakan bahasa pemrograman PHP, Javascript, maupun bahasa pemrograman lainnya. Untuk pembuatan framework game, bahasa pemrograman yang dipakai adalah GDScript

Berikut ini beberapa bahasa pemrograman populer yang di gunakan untuk memulai *coding*.

1. PHP
2. Python
3. Javascript
4. Java
5. C++
6. C#
7. C
8. GDScript
9. Ruby
10. Swift
11. R
12. Go
    * 1. **Contoh Coding**

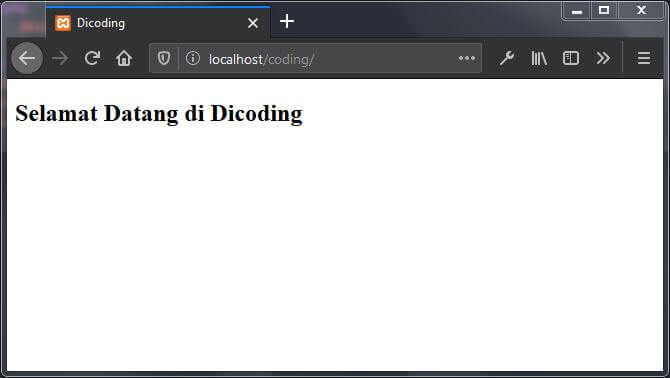
Dari penjelasan sebelumnya dijelaskan bahwa *coding* merupakan kegiatan saat menuliskan kode program menggunakan bahasa pemrograman tertentu. Sehingga kode program tersebut dapat dipahami dan dijalankan oleh komputer.

* + - 1. **Contoh Coding dengan PHP**

contoh kode program dalam bahasa pemrograman PHP, untuk menampilkan teks “Selamat Datang di Dicoding” seperti berikut.

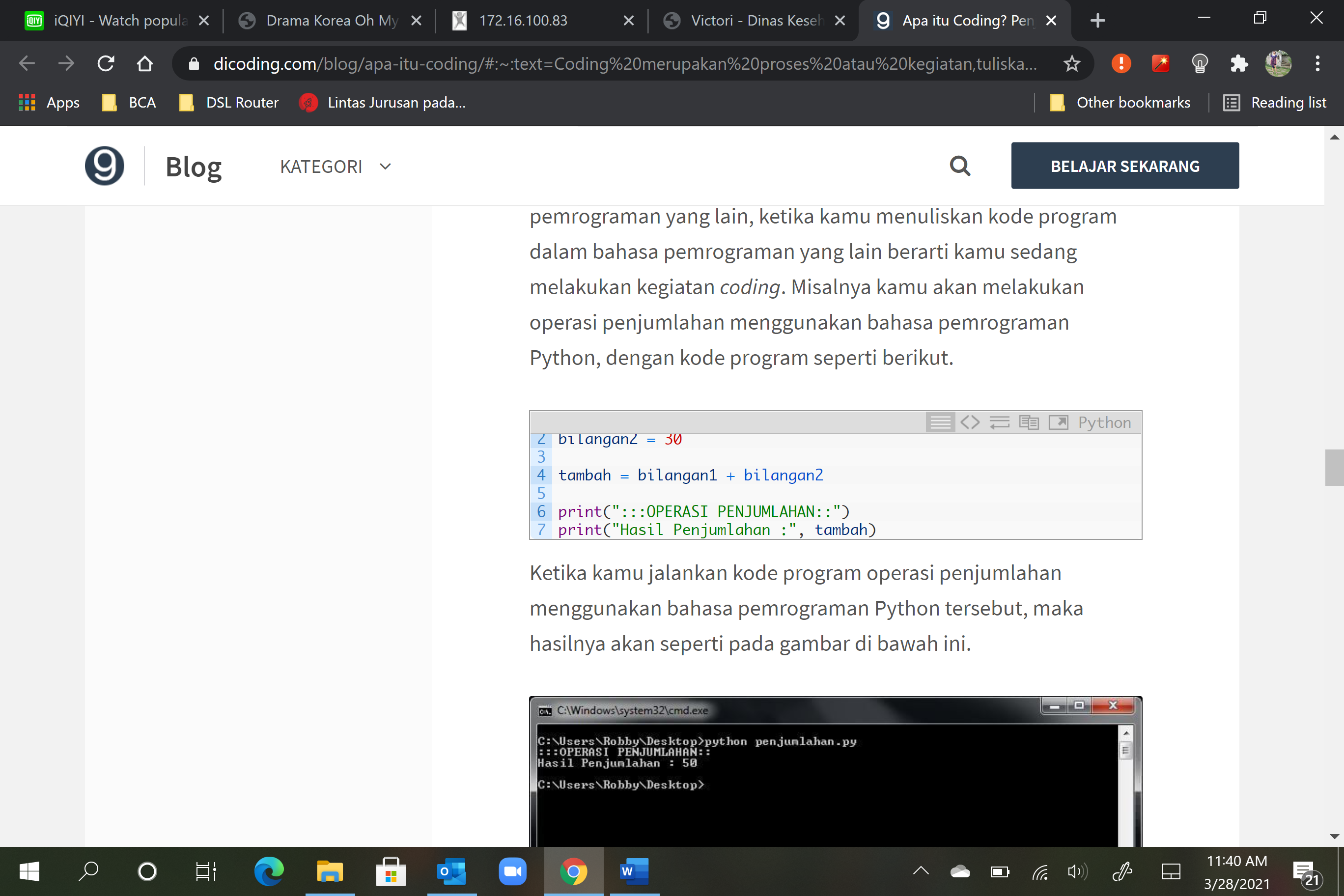
|  |  |
| --- | --- |
|  |  |

Saat menuliskan kode program tersebut berarti sedang melakukan coding. Kode program tersebut saat dijalankan melalui web browser, maka hasilnya akan seperti pada gambar di bawah.



* + - 1. **Contoh Coding dengan Python**

Contoh penggunaan bahasa program python, misalnya akan melakukan operasi penjumlahan menggunakan bahasa pemrograman Python, dengan kode program seperti berikut.



Ketika kode program operasi penjumlahan menggunakan bahasa pemrograman Python tersebut dijalankan, maka hasilnya akan seperti pada gambar di bawah ini.



* + - 1. **Contoh Coding dengan GDscript**

GDScript adalah bahasa pemrograman khusus *Godot Engine* yang merupakan bahasa pemrograman tingkat tinggi, dinamis. Bahasa pemrograman ini mirip Python, sehingga memiliki kata kunci dan cara pengetikan yang mirip demikian. Tujuan dari GDScript adalah optimalisasi, fleksibilitas, dan integrasi yang baik dengan *Godot Engine*.

Beberapa orang dapat mempelajari lebih baik dengan melihat syntax. Berikut adalah contoh bagaimana cara GDScript bekerja:

*# A file is a class!*

*# Inheritance*

**extends** BaseClass

*# (optional) class definition with a custom icon*

**class\_name** MyClass, "res://path/to/optional/icon.svg"

*# Member variables*

**var** a = 5

**var** s = "Hello"

**var** arr = [1, 2, 3]

**var** dict = {"key": "value", 2: 3}

**var** typed\_var: int

**var** inferred\_type := "String"

*# Constants*

**const** ANSWER = 42

**const** THE\_NAME = "Charly"

*# Enums*

**enum** {UNIT\_NEUTRAL, UNIT\_ENEMY, UNIT\_ALLY}

**enum** Named {THING\_1, THING\_2, ANOTHER\_THING = -1}

*# Built-in vector types*

**var** v2 = Vector2(1, 2)

**var** v3 = Vector3(1, 2, 3)

*# Function*

**func** some\_function(param1, param2):

**var** local\_var = 5

**if** param1 < local\_var:

print(param1)

**elif** param2 > 5:

print(param2)

**else**:

print("Fail!")

**for** i **in** range(20):

print(i)

**while** param2 != 0:

param2 -= 1

**var** local\_var2 = param1 + 3

**return** local\_var2

*# Functions override functions with the same name on the base/parent class.*

*# If you still want to call them, use '.' (like 'super' in other languages).*

**func** something(p1, p2):

.something(p1, p2)

*# Inner class*

**class** Something:

**var** a = 10

*# Constructor*

**func** \_init():

print("Constructed!")

**var** lv = Something.new()

print(lv.a)

* + 1. **Kesimpulan**

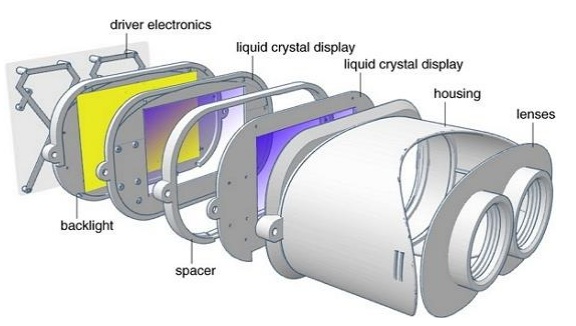
*Coding* merupakan proses atau kegiatan pengolahan kode yang dituliskan menggunakan bahasa pemrograman tertentu. Sehingga komputer dapat memahami apa yang dituliskan dan melakukan operasi sesuai dengan apa yang ditulis.

### Virtual Reality (VR)

Virtual reality adalah sebuah teknologi yang membuat pengguna atau user dapat berinteraksi dengan lingkungan yang ada dalam dunia maya yang disimulasikan oleh komputer, sehingga pengguna merasa berada di dalam lingkungan tersebut. Di dalam bahasa Indonesia virtual reality dikenal dengan istilah realitas maya. Teknologi virtual reality yang lebih awal adalah Peta Bioskop Aspen, yang diciptakan oleh MIT pada tahun 1977. Programnya adalah suatu simulasi kasar tentang kota Aspen di Colorado, dimana para pemakai bisa mengembara dalam salah satu dari tiga gaya yaitu musim panas, musim dingin, dan poligon. Dua hal pertama tersebut telah didasarkan pada foto, karena para peneliti benar-benar memotret tiap-tiap pergerakan yang mungkin melalui pandangan jalan kota besar pada kedua musim tersebut, dan yang ketiga adalah suatu model dasar 3 dimensi kota besar. Akhir tahun 1980 istilah “Virtual Reality” telah dipopulerkan oleh Jaron Lanier, salah satu pelopor modern dari bidang tersebut. Lanier yang telah mendirikan perusahaan VPL Riset pada tahun 1985, telah mengembangkan dan membangun sistem “kacamata hitam dan sarung tangan” yang terkenal pada masa itu. Secara umum, VR sudah dikembangkan cukup lama, tepatnya mulai dari tahun 1800-an. Berikut ditampilkan sejarah VR dari awal hingga saat ini.

* 1800-an – Mulai muncul ide untuk membuat sebuah alternatif realitas seiring dengan mulai munculnya praktek fotografi.
* 1838 – Ditemukan stereoskop pertama yang menggunakan dua cermin kembar untuk memproyeksikan sebuah gambar.
* 1839 – Stereoskop tersebut dikembangkan menjadi View-Master dan kemudian dipatenkan satu abad kemudian pada tahun 1939.
* 1956 – Morton Heilig yang memiliki latar belakang di industri Motion Picture Hollywoodmulai menginginkan orang-orang mampu merasakan suasana bagaikan masuk ke dalam sebuah film. Dibuatlah simulasi Sensorama yang dapat membuat penggunanya merasakan suasana lingkungan perkotaan bagaikan dengan menaiki sepeda motor. Sudah dilengkapi dengan multisensor stimulasi, sehingga penggunanya mampu melihat jalan, mendengar mesin motor berbunyi, merasakan getaran motor, dan mencium bau mesin motor di sebuah dunia yang didesain teknologi.
* 1960 – Morton Heilig kemudian mematenkan peralatan yang dinamakan dengan Telesphere Mask. Banyak investor yang kemudian tertarik bekerjasama.
* Pertengahan 1980 – Mulai digunakan istilah “virtual reality”. Adalah Jaron Lanier, founder dari VPL Research yang mulai mengembangkan peralatan virtual reality, termasuk goggle (kaca mata) dan sarung tangan yang dibutuhkan seseorang untuk merasakan pengalaman VR.
* Saat ini – Setelah enam dekade dikembangkan dengan bantuan dana investor, kini VR bisa dinikmati secara luas dengan harga yang ekonomis, menggunakan peralatan berkualitas tinggi yang mudah diakses.

Virtual reality bekerja dengan memanipulasi otak manusia sehingga seolah-olah merasakan berbagai hal yang virtual terasa seperti hal yang nyata. Bisa dibilang, virtual reality merupakan proses penghapusan dunia nyata di sekeliling manusia, kemudian membuat si pengguna merasa tergiring masuk ke dunia virtual yang sama sekali tak bersentuhan dengan dunia nyata. Untuk dapat melakukan hal ini, tentu dibutuhkan berbagai perangkat tambahan. Paling minimalnya, jika Anda ingin merasakan masuk ke dalam dunia virtual reality, maka dibutuhkan sebuah headset VR, seperti misalnya yang kini banyak ditemukan di pasaran adalah Oculus Rift atau Samsung Gear VR.



**Gambar 2.1. Susunan komponen pada perangkat VR**

Susunan komponen pada perangkat VR Secara kasat mata, headset VR ini berbentuk seperti kacamata selam, namun dengan lensa tertutup. Bagian yang seperti kacamata selam ini dinamakan sebagai VR box, yang merupakan tempat untuk meletakkan smartphone yang berfungsi memproyeksikan gambar virtual. VR yang menggunakan smartphone ini merupakan perangkat VR versi standar. Berbeda misalnya dengan VR headset yang sudah menggunakan teknologi canggih seperti Oculus Rift, di dalamnya bukan lagi menggunakan smartphone, melainkan sudah terdapat sebuah layar yang menampilkan video dan gambar virtual reality yang juga bisa terhubung dengan komputer menggunakan bluetooth. Ada 4 elemen penting dalam virtual reality.

Adapun 4 elemen itu adalah sebagai berikut :

1. **Virtual world**, sebuah konten yang menciptakan dunia virtual dalam bentuk screenplay maupun script.
2. **Immersion**, sebuah sensasi yang membawa pengguna teknologi virtual reality merasakan ada di sebuah lingkungan nyata yang padahal fiktif. Immersion dibagi dalam 3 jenis, yakni:

* Mental immersion, membuat mental penggunanya merasa seperti berada di dalam lingkungan nyata.
* Physical immersion,membuat fisik penggunanya merasakan suasana di sekitar lingkungan yang diciptakan oleh virtual reality tersebut.
* Mentally immersed, memberikan sensasi kepada penggunanya untuk larut dalam lingkungan yang dihasilkan virtual reality.

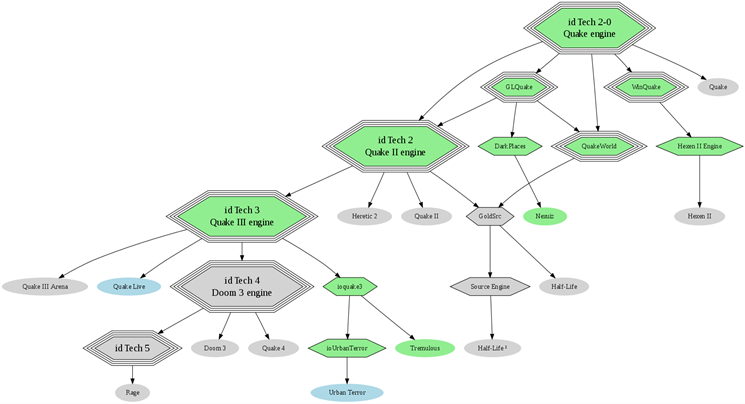
1. **Sensory feedback** berfungsi untuk menyampaikan informasi dari virtual world ke indera penggunanya. Elemen ini mencakup visual (penglihatan), audio (pendengaran) dan sentuhan.
2. **Interactivity**yang bertugas untuk merespon aksi dari pengguna, sehingga pengguna dapat berinteraksi langsung dalam medan fiktif atau virtual world. Pemakai melihat suatu dunia semu yang sebenarnya adalah gambar-gambar bersifat dinamis. Melalui Headphone atau speaker, pendengar akan mendengar suara yang realistis. Melalui headset, glove, dan walker, semua gerakan pemakai dipantau oleh sistem yang akan memberikan reaksi yang sesuai sehingga pemakai seolah-olah merasakan pada situasi yang nyata, baik secara fisik maupun psikologis.

Virtual reality sendiri memiliki efek negatif yaitu Cybersickness. Cybersickness adalah gangguan yang sering dialami dalam penggunaan virtual reality. Penderita akan merasa ketegangan mata dan bahkan disertai rasa pusing. Tekadang penderita secara psikologis masih terbawa pada suasana semu walaupun sebenernya penderita sudah kembali ke dunia nyata. Jadi, gangguan ini tidak boleh dianggap remeh karena penderita biasanya susah kembali ke dunia nyata dan selalu terangan-angan dalam dunia semu.

### Game Engine

***Game engine*** (arti harfiah: "mesin permainan") adalah sistem [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak" \o "Perangkat lunak) yang dirancang untuk menjadi dasar pembuatan [permainan video](https://id.wikipedia.org/wiki/Permainan_video" \o "Permainan video), seperti permainan di komputer, [konsol](https://id.wikipedia.org/wiki/Konsol_permainan" \o "Konsol permainan), atau ponsel.[[1]](https://id.wikipedia.org/wiki/Game_engine" \l "cite_note-1) *Game engine* memberikan kemudahan bagi pengembang permainan karena menyediakan fungsi-fungsi inti dari sebuah permainan, misalnya [grafika](https://id.wikipedia.org/wiki/Grafika" \o "Grafika) (menghasilkan grafika 2-dimensi atau 3-dimensi), [fisika](https://id.wikipedia.org/wiki/Fisika" \o "Fisika) (menghitung dan menyimulasikan hukum-hukum gerak dan hukum fisika lainya), [audio](https://id.wikipedia.org/wiki/Audio" \o "Audio), atau [kecerdasan buatan](https://id.wikipedia.org/wiki/Kecerdasan_buatan" \o "Kecerdasan buatan). Sebuah *game engine* dapat digunakan untuk membuat lebih dari satu permainan, dan pengembang permainan dapat mengoptimisasi proses pengembangan dengan cara menggunakan atau mengadaptasi *game engine* yang telah ada sebelumnya.[[2]](https://id.wikipedia.org/wiki/Game_engine" \l "cite_note-2)

**2.4.1 Sejarah**



Sebelum game engine, permainan biasanya ditulis sebagai entitas tunggal: permaianan untuk [Atari 2600](https://id.wikipedia.org/wiki/Atari_2600" \o "Atari 2600), misalnya, harus dirancang dari bawah ke atas untuk mengoptimalkan penggunaan perangkat keras layar. Rutinitas tampilan inti ini sekarang disebut [kernel](https://id.wikipedia.org/wiki/Kernel_(ilmu_komputer)" \o "Kernel (ilmu komputer)) oleh pengembang retro. Platform lain memiliki lebih banyak kelonggaran, tetapi ketika tampilan tidak menjadi perhatian, kendala memori menyabotase upaya untuk membuat desain padat data yang dibutuhkan mesin. Bahkan pada platform yang lebih akomodatif, sangat sedikit yang dapat digunakan kembali di antara permainan. Kemajuan pesat perangkat keras arkade — yang merupakan ujung tombak pasar pada saat itu — berarti bahwa sebagian besar kode harus dibuang setelahnya, karena permainan generasi selanjutnya akan menggunakan desain permainan yang sama sekali berbeda yang memanfaatkan kelebihan sumber daya. Jadi sebagian besar desain permainan selama 1980-an dirancang melalui seperangkat aturan hard-code dengan sejumlah kecil level dan data grafik. Sejak [zaman keemasan permainan video arkade](https://id.wikipedia.org/wiki/Zaman_keemasan_permainan_video_arkade" \o "Zaman keemasan permainan video arkade), sudah umum bagi [perusahaan permainan video](https://id.wikipedia.org/w/index.php?title=Daftar_perusahaan_permainan_video&action=edit&redlink=1" \o "Daftar perusahaan permainan video (halaman belum tersedia)) untuk mengembangkan mesin game in-house untuk digunakan dengan perangkat lunak pihak pertama.

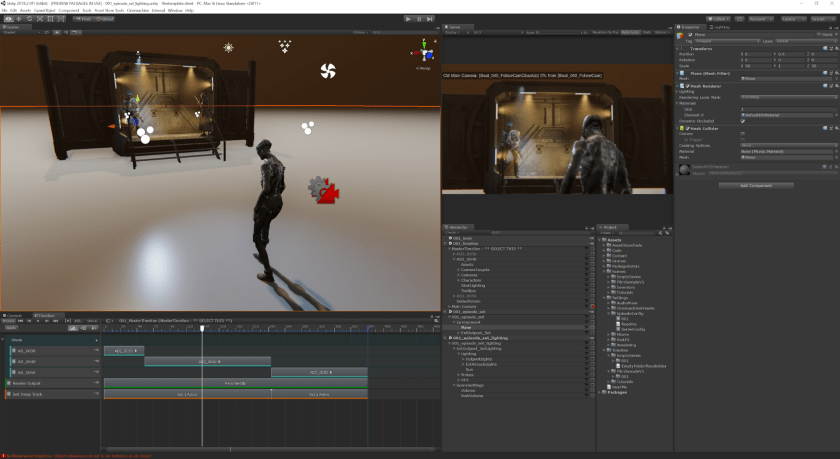
Pada perkembangan selanjutnya, produsen mesin game memutuskan mengizinkan pengguna untuk memanfaatkan produk mereka. Sama seperti [permainan adalah sebuah industri](https://id.wikipedia.org/wiki/Industri_permainan_video" \o "Industri permainan video), begitu pula mesin yang mereka buat. Mesin permainan utama datang dengan harga yang bervariasi, baik dalam bentuk biaya berlangganan atau pembayaran lisensi.[[3]](https://id.wikipedia.org/wiki/Game_engine#cite_note-3)

[Unreal Engine 4](https://id.wikipedia.org/wiki/Unreal_Engine), salah satu mesin game utama dan digunakan untuk membuat beberapa permainan terkenal seperti *Fortnite*,   [*PlayerUnknown's Battlegrounds*](https://id.wikipedia.org/wiki/PlayerUnknown%27s_Battlegrounds), dan [*Life Is Strange 2*](https://id.wikipedia.org/w/index.php?title=Life_Is_Strange_2&action=edit&redlink=1), mengadopsi struktur yang dapat digunakan secara gratis dengan royalti pada semua penjualan permainan yang menggunakan mesin ini.[[4]](https://id.wikipedia.org/wiki/Game_engine#cite_note-4) Meskipun terdapat perbedaan di antara mesin game karena dibuat selama proses pembuatan permainan, pengembang permainan yang berbeda mungkin terbiasa dengan sistem untuk diubah, atau tertarik oleh manfaat besar dari mesin tersebut terlepas dari pembayarannya.

Mesin game lain yang saat ini mendatangkan penghasilan penting adalah [mesin Unity](https://id.wikipedia.org/w/index.php?title=Unity_(mesin_game)&action=edit&redlink=1" \o "Unity (mesin game) (halaman belum tersedia)), yang menggunakan modul pembayaran yang mirip dengan Unreal Engine. Mesin ini berada di belakang game seperti [*Rust*](https://id.wikipedia.org/w/index.php?title=Rust_(permainan_video)&action=edit&redlink=1), *[Subnautica](https://id.wikipedia.org/w/index.php?title=Subnautica&action=edit&redlink=1" \o "Subnautica (halaman belum tersedia))*, dan [*Life Is Strange: Before the Storm*](https://id.wikipedia.org/w/index.php?title=Life_Is_Strange:_Before_the_Storm&action=edit&redlink=1).

Layaknya memasak, pengembangan game juga memerlukan alat-alat sebelum menghidangkan karyanya. Dengan bantuan *tools*, proses pembuatan lebih mudah dibuat sehingga bisa disajikan ke pemain lebih cepat. Berikut daftar *tools* yang umum digunakan oleh *game developer*dalam membangun kreasinya

* + 1. **Unity**

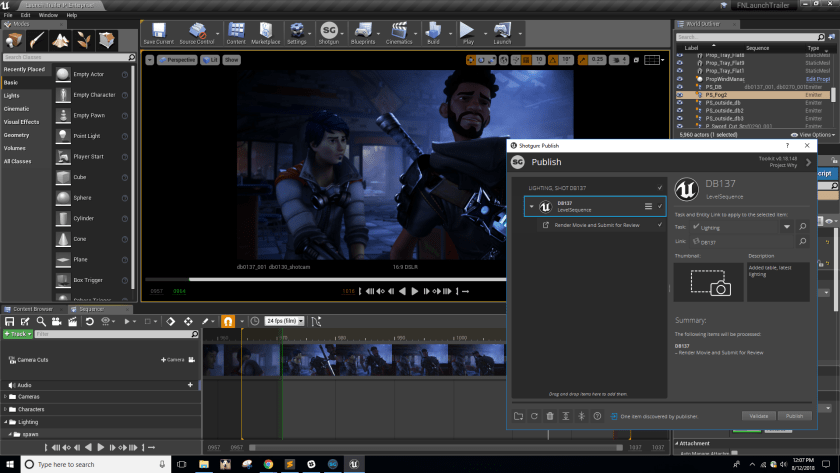


Unity adalah sebuah game engine cross platform. Unity mengklaim project yang dibuat bisa dicompile ke lebih dari 25 platformmulai dari PC, mobile, console, hingga *smart TV*.

Unity cukup populer digunakan oleh pembuat game mobile. Unity mengatakan [di websitenya](https://unity3d.com/public-relations) 50% dari game mobile menggunakan Unity sebagai toolsnya.

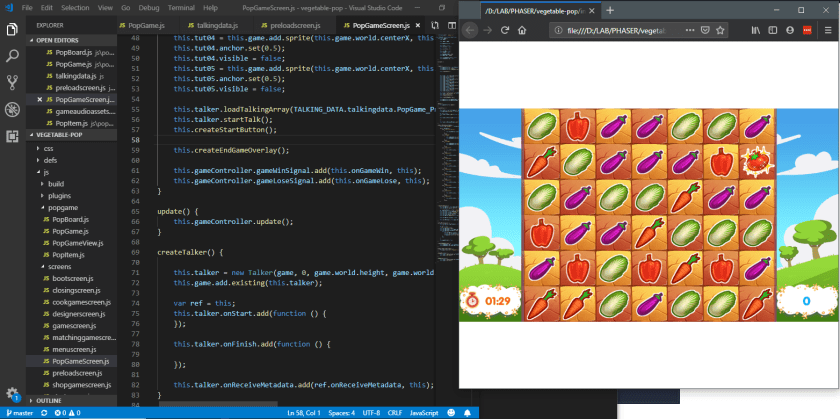
Unity bisa digunakan secara gratis tanpa batasan fitur apapun**,**hanya saja kamu diharuskan menampilkan *splash screen* Unity di awal game. Untuk menghilangkannya cukup berlangganan sekitar Rp 360.000 sebulan. Dengan berlangganan bisa mendapatkan diskon 20% dari Asset Store yang tersedia.

* Developer: Unity Technologies
* Contoh game: Pokemon GO, Ori and the Blind Forest, Cuphead
* Kelebihan: Support 3D & 2D, multiplatform, konten Asset Store beragam
* Harga: Gratis. Berlangganan, mulai dari $25 per bulan
* Website: <https://unity.com/>
  + 1. **Unreal Engine**



Unreal Engine adalah game engine dari Epic Games. Walaupun pada awalnya dikembangkan untuk game First Person Shooter, kini game engine ini populer digunakan dalam membuat game dari beragam genre. Unreal memiliki serangkaian tools yang sangat lengkap mulai *coding*, desain environment, *cinematic camera*, hingga multiplayer. Game engine ini dapat digunakan dengan mengetik langsung bahasa pemrograman C++ atau menggunakan fitur fitur *visual coding*.

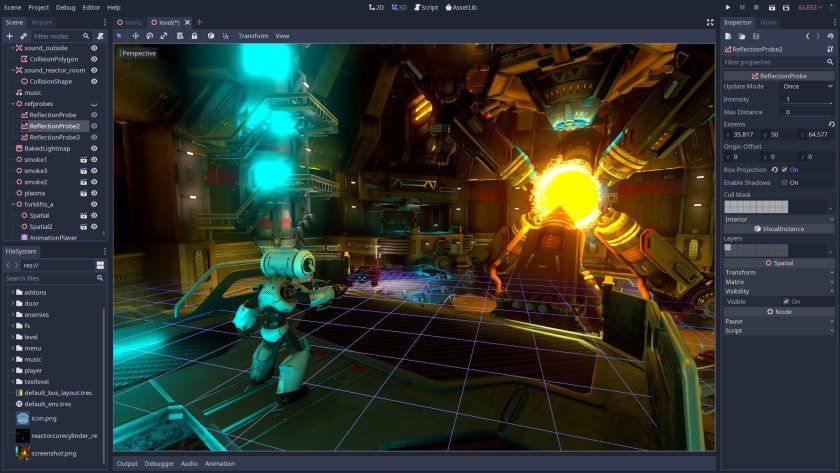
* Developer: Epic Games
* Contoh Game: Fortnite, Hellblade: Senua’s Sacrifice, Devil May Cry, Batman : Arkham Asylum
* Harga: Gratis. Jika game yang dirilis menggunakan Unreal mendapat keuntungan mulai dari $3000, Epic mendapat royalti sebesar 5%
* Website: <https://www.unrealengine.com/>
  + 1. **Phaser**



Phaser adalah game framework untuk HTML5. Karena dikembangkan sangat spesifik untuk browser, Phaser sangat mumpuni untuk membuat game ringan atau komponen interaktif di web. Sebagai contoh, membuat game instant Facebook dengan menggunakan Phaser. Framework ini sudah menyediakan beragam *tools* untuk membuka gambar, animasi, suara, hingga interaksi physics.

Secara native, Phaser dapat digunakan dengan bahasa pemrograman JavaScript**.**Karena hanya berupa framework, web developer bisa menggunakan Phaser sebagai bagian dari aplikasi yang dikembangkan dan menjalankannya melalui pipeline yang sudah dimiliki sebelumnya.

* Developer: open source dipimpin oleh [@photonstorm](https://twitter.com/photonstorm)
* Contoh game: [C.A.T.S Instant Fight](https://phaser.io/news/2019/04/cats-instant-fights), [Junior Chess](https://phaser.io/news/2019/06/junior-chess), [Magikmon](https://phaser.io/news/2019/06/magikmon)
* Kelebihan: Output kecil dan ringan
* Harga: Gratis, [plugin dan premium support dijual terpisah](https://phaser.io/shop)
* Website: <http://phaser.io/>
  + 1. **Godot**



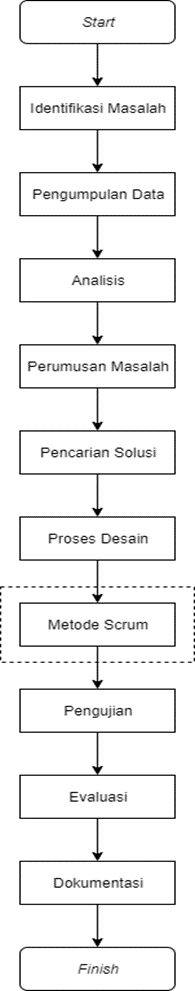
Godot adalah game engine 2D dan 3D gratis dan open source. Pengembangannya bisa diikuti di <https://github.com/godotengine>. Game engine ini sangat ringan dan hanya berukuran sekitar 25 MB.

Godot menawarkan konsep *node*. Dengan konsep ini setiap objek yang ada di dalam game bisa ditambahkan interaksi dan fungsi yang berbeda dengan menambahkan objek *node*spesifik. Godot juga menawarkan visual editor sehingga mendesain *environment* game dapat dilakukan dengan mudah.

Scripting di Godot bisa dilakukan dengan GDScript, C#, C++, dan VisualScript. Karena *logic* yang dibuat bisa menggunakan visual scripting, Godot juga bisa digunakan oleh pengguna non-programmer.

# BAB 3 METODE PENELITIAN

## Diagram Alir kerangka berpikir



Tahapan-tahapan yang disebutkan dalam Gambar 3.1 dapat dijabarkan sebagai berikut:

1. Identifikasi Masalah

Pada tahapan ini dilakukan pencarian dan analisa masalah. Dari masalah yang telah teridentifikasi digunakan untuk pertimbangan terhadap solusi dan desain perancangan.

2. Pengumpulan Data

Pengumpulan data bertujuan untuk membuat fondasi yang kuat bagi pelaksanaan penelitian. Pengumpulan data dilakukan dengan metode kuesioner dan pengumpulan informasi hasil upload perancangan di website

- Kuesioner

Pembagian kuesioner dilakukan sebelum pembuatan rancangan framework dimulai pada tanggal 02 Pebruari 2021 hingga 25 Pebruari 2021 dengan total responden 40 orang menggunakan aplikasi.

- Data upload hasil rancangan framework di website itch.io, Sourceforge, dan Odysee

3. Analisis

Analisis dilakukan melalui 2 cara, yaitu analisis pengguna dan analisis aplikasi sejenis. Analisis pengguna dilakukan untuk memahami kebutuhan pengguna secara spesifik yang dirujuk dari hasil pembagian kuesioner. Analisis aplikasi sejenis digunakan untuk mencari referensi dari aplikasi sejenis yang sudah beredar di pasar sekaligus membandingkan fitur-fitur yang sudah ada.

4. Perumusan Masalah

Masalah yang telah berhasil diidentifikasi dan dianalisis dibuat menjadi pertanyaan yang melatarbelakangi penelitian.

5. Pencarian Solusi

Pertanyaan-pertanyaan dari perumusan masalah kemudian dijawab dengan solusi yang telah dicari dan dijadikan pertimbangan dalam perancangan framework.

6. Proses Desain

Proses desain diawali dengan diskusi terhadap perancangan framework, diantaranya desain sistem framework, desain tampilan framework, fitur-fitur framework dan pembuatan framework melalui pemrograman, *drag & drop*, dan pengisian parameter.

7. Pengujian

Tahap ini dilakukan untuk menguji hasil framework kepada pengguna sekaligus mengumpulkan data untuk diteliti. Hasil pengujian akan disertai dengan kuesioner untuk dilanjutkan ke tahap evaluasi.

9. Evaluasi

Evaluasi dilakukan untuk memaparkan hasil dari penelitian yang telah dianalisis juga disertai dengan kesimpulan.

## Perancangan Framework game VR

Framework game adalah kerangka kerja untuk mengembangkan pembuatan game. Didalam framework game yang dibuat dilengkapi dengan fitur-fitur yang diperlukan designer untuk mengembangkan gamenya. Engine yang digunakan dalam perancangan framework ini adalah godot engine. Dalam pemprograman framework digunakan bahasa program GDScript

* + 1. Proses coding framework

Proses coding diawali dengan menyiapkan game engine dan membuat proyek. Proses selanjutnya adalah menyusun elemen-elemen framework. Elemen-elemen framework adalah sebagai berikut :

* UI seperti teks, tombol, pilihan, dan lain lain
* Audio
* Visual

Setelah semua elemen disusun, mulai dilakukan pengisian kodingan setiap elemen.

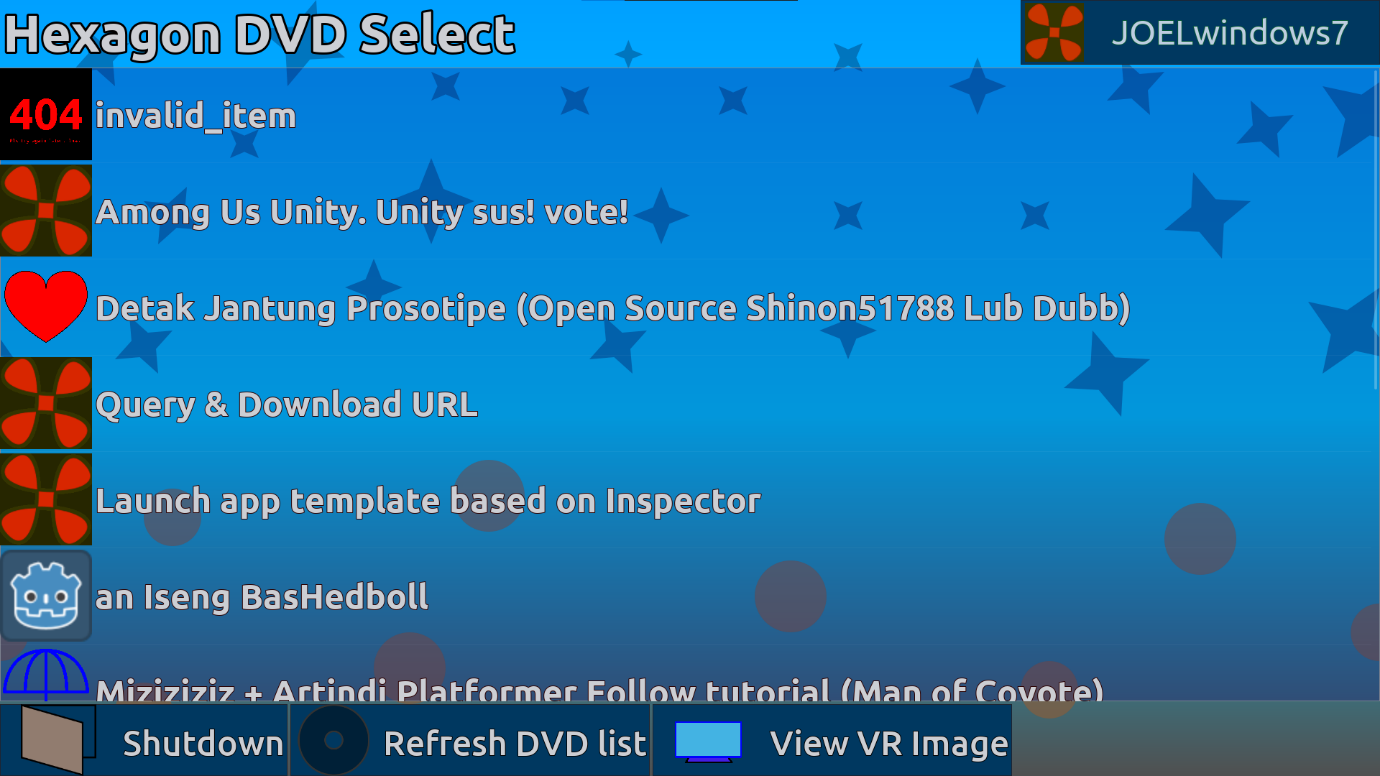
Berikut hasil dari coding framework

* + 1. Pembuatan fitur-fitur
    2. Pemasangan engine
    3. Pembuatan program game VR

# BAB 4 HASIL DAN PEMBAHASAN

* 1. Gambaran hasil rancangan framework hexagone engine adalah sebagai berikut :  
     Sebuah gambar berisi teks, cuplikan layar, hijau

     Deskripsi dibuat secara otomatisSebuah gambar berisi teks, meja biliar, bola biliar, meja

     Deskripsi dibuat secara otomatis
  2. Sample programe hexagone engine sebagai gambaran :   
     Sebuah gambar berisi teks

     Deskripsi dibuat secara otomatisSebuah gambar berisi teks, monitor, cuplikan layar, hitam

     Deskripsi dibuat secara otomatis

# DAFTAR PUSTAKA

(t.thn.). Diambil kembali dari https://stackoverflow.com/questions/5068992/whats-the-difference-between-an-engine-and-a-framework

(t.thn.). Diambil kembali dari https://www.gamedesigning.org/career/video-game-engines/

(t.thn.). Diambil kembali dari https://www.quora.com/Whats-the-difference-between-a-game-engine-and-a-game-development-framework

(t.thn.). Diambil kembali dari https://ourcodeworld.com/articles/read/308/top-15-best-open-source-javascript-game-engines

(t.thn.). Diambil kembali dari https://www.g2.com/categories/vr-game-engine

(t.thn.). Diambil kembali dari https://gamedev.stackexchange.com/questions/31772/what-is-the-difference-between-a-game-framework-and-a-game-engine

(t.thn.). Diambil kembali dari http://markbro.com/5-framework-backend-terbaik-di-tahun-2020/

Adams, E. (2013). *Fundamentals of Game Design.* New Riders.

Asmiatun, S., & Putri, A. N. (2017). *Belajar Membuat Game 2D dan 3D Menggunakan Unity.* Deepublish.

*BUAT GAME 3D DENGAN UNITY 3D*. (2019, Juli 11). Diambil kembali dari BUAT GAME 3D DENGAN UNITY 3D: http://ariusman.id/buat-game-3d-dengan-unity-3d/

Byl, P. D. (2014). *Holistic Mobile Game Development with Unity.* Focal Press.

Chukong Technologies. (2010, Desember). Diambil kembali dari https://www.cocos.com/

Clickteam. (1993). *Clickteam*. Diambil kembali dari https://www.clickteam.com/

CryTek. (2002, Mei). *CryEngine*. Diambil kembali dari https://www.cryengine.com/

Deitel, P. J., & Deitel, H. M. (2016). *C: How to Program.* Pearson.

DRAGANOV, D. D. (2-14). *Freemium Mobile Games: Design and Monetization.* CreateSpace Independent Publishing Platform.

Dzone. (t.thn.). Diambil kembali dari https://dzone.com/articles/top-5-free-and-open-source-game-engines-you-should

Epic Games. (t.thn.). *Unreal Engine*. Diambil kembali dari https://www.unrealengine.com/

Gahan, A. (2012). *3ds Max Modeling for Games: Insider's Guide to Game Character, Vehicle, and Environment Modeling.* Taylor & Francis.

Gameplay Analytics. (t.thn.). Diambil kembali dari https://gameanalytics.com/blog/top-5-game-engines-for-beginners/?utm\_term=&utm\_campaign=Dynamic+Keyword+Campaign&utm\_source=adwords&utm\_medium=ppc&hsa\_acc=9473236018&hsa\_cam=978897011&hsa\_grp=50214658724&hsa\_ad=231772194854&hsa\_src=g&hsa\_tgt=dsa-295317350

Games From Scratch. (2015, Juni 13). *GameDev Glossary: Library Vs Framework Vs Engine*. Diambil kembali dari https://gamefromscratch.com/gamedev-glossary-library-vs-framework-vs-engine/

Hanly, J. R., & Koffman, E. B. (2009). *Problem Solving and Program Design in C.* Addison-Wesley.

Hidayatno, A., Destyanto, A. R., & Iman, M. R. (t.thn.). *Bermain untuk Belajar: Merancang Permainan sebagai Media Pembelajaran yang Efektif.* Penerbit LeutikaPrio.

Hocking, J. (2015). *Unity in Action: Multiplatform Game Development in C#.* Manning Publications Company.

Juan Linietsky, d. (2014, Januari). *Godot Engine*. Diambil kembali dari https://godotengine.org

Kölling, M. (2003). *Greenfoot*. Diambil kembali dari https://www.greenfoot.org/

Kurniawan, D. (2020, Mei 13). *Yuk! Mengenal Apa Itu Framework & Fungsinya*. Diambil kembali dari Yuk! Mengenal Apa Itu Framework & Fungsinya: https://www.niagahoster.co.id/blog/apa-itu-framework/

LibGDX. (t.thn.). Diambil kembali dari https://libgdx.com/

Monogames. (t.thn.). Diambil kembali dari https://www.monogame.net/

More Mountains. (t.thn.). *Corgi Engine*. Diambil kembali dari https://corgi-engine.moremountains.com/

More Mountains. (t.thn.). *Topdown Engine*. Diambil kembali dari https://topdown-engine.moremountains.com/

Ogre. (t.thn.). Diambil kembali dari https://www.ogre3d.org/

Pranata, B. A., & Pamoedji, A. K. (2015). *Mudah Membuat Game dan Potensi Finansialnya dengan Unity 3D.* Elex Media Komputindo.

Rival, F. (2008, Agustus). *GDevelop*. Diambil kembali dari https://gdevelop-app.com/

Sapio, F. (2017). *Getting Started with Unity 5.x 2D Game Development.* Packt Publishing.

Sapio, F., & Saher, A. (2016). *Unity 5.x 2D Game Development Blueprints.* Packt Publishing.

SFML. (t.thn.). Diambil kembali dari https://www.sfml-dev.org/

Silicon Studio. (2014). *Stride*. Diambil kembali dari https://stride3d.net/

Thorn, A. (2018). *Unity 2018 By Example: Learn about game and virtual reality development by creating five engaging projects, 2nd Edition.* Packt Publishing Ltd.

Unigine. (2005, Mei 4). *Unigine*. Diambil kembali dari https://unigine.com/

Unity Technologies. (2005). *Unity*. Diambil kembali dari https://unity.com