**BAB II**

**LANDASAN TEORI**

**2.1 *Game***

Dalam Bahasa Indonesia *game* diartikan sebagai permainan. Pada Kamus Besar Bahasa Indonesia permainan adalah sesuatu yang digunakan untuk bermain. [5]

*Game* adalah sistem yang dibuat dimana pemain dapat berinteraksi dengan konflik buatan. Sistem buatan yang mengandung peraturan dan jenis permainan yang dapat diselesaikan oleh pemain sehingga mendapatkan hasil yang memuaskan.

**2.1.1 Permainan Video**

Permainan video didefinisikan sebagai alat yang menampilkan permainan menggunakan peralatanmedia elektronik, seperti: Pesawat televisi, monitor, dll. Pada umunya, permainan video memerlukan perangkat keras komputer untuk menjalakanya seperti PC dan *Playstation* yang dimana komponen tersebut disambungkan ke pesawat televisi atau monitor untuk menampilkannya.

**2.1.2** ***Genre***

*Genre* atau ragam permainan video digunakan untuk membedakan juga menggolongkan [permainan video](https://id.wikipedia.org/wiki/Permainan_video) berdasarkan jenis interaksi atau *gameplay* permainannya. Permainan video diklasifikasikan berdasarkan pengaturan dan *Gameplay* yang diterapkan dalam permainan.

Berikut ini adalah macam-macam dari genre permainan video:

* Aksi

Permainan aksi membutuhkan pemain menggunakan kemampuan refleks, akurasi, dan waktu yang tepat untuk menyelesaikan sebuah tantangan. *First Person Shooter*, *Third Person Shooter*, dan *Fighting* merupakan contoh permainan *genre* aksi.

* Balap

 Ragam permainan video yang menuntut keterampilan pemain untuk mengemudi dalam sebuah kompetisi balap, seperti balap mobil, motor, dan sepeda.

* Olahraga

Ragam permainan video yang menuntut keterampilan pemain untuk melakukan pertandingan olahraga secara virtual, seperti : sepak bola dan basket.

* Papan

Ragam permainan yang menyerupai permainan papan pada dunia nyata, seperti catur dan monopoli.

* Peran

*Role Playing Game* adalah ragam permainan video yang menerapkan pemain sebagai karakter utama atau tokoh untuk memecahkan suatu misi seperti *quest* dan mengalahkan *boss* terakhir untuk menyelesaikan permainan.

* Petualangan

Ragam permainan petualangan yang menyerupai sebuah cerita pada buku cerita. Tujuan dari permainan raga ini adalah menamatkan permainan tersebut.

* Simulasi

Ragam permainan yang menggambarkan kondisi dalam dunia nyata, seperti simulasi pilot pesawat terbang.

* Strategi

Ragam ini memiliki 2 *sub-genre*, diantaranya *Real-Time strategy* dan *Turn-based strategy*. *Real-Time strategy* adalah ragam permainan strategi dengan waktu yang bersamaan bermain dengan pemain atau musuh komputer. *Turn-based strategy* ragam permainan yang dimana pemainnya menunggu giliran untuk bergerak.

**2.2 Unreal Engine**

Unreal Engine adalah *game engine* yang membantu untuk membuat perminan. Unreal Engine dibuat dari beberapa komponen yang bekerja sama untuk menggerakkan permainan. Sistem peralatan dan editor yang besar memungkinkan pengguna untuk mengatur dan memanipulasi aset untuk membuat *gameplay* dari game yang dibuat.

Komponen Unreal Engine meliputi *sound engine, physics engine, input* dan *gameplay* *framework*, dan *online module*.[12]

**2.2.1 *Sound Engine***

*Sound Engine* bertanggung jawab dalam music dan suara di permainan. Unreal Engine memungkinkan berbagai *file* suara untuk mengatur suasana hati dan menambahkan *realism* untuk permainan. Terdapat banyak penggunaan suara dalam permainan. *Ambient sound* terus menerus dilator belakang. *Sound effects* dapat dimainkan berulang-ulang ketika dibutuhkan dan dipicu oleh *event* tertentu di*game*.[12]

**2.2.2 *Physics Engine***

Dalam dunia nyata, objek diatur oleh hukum fisika. Objek bertabrakan dan ditetapkan dalam gerakan menurut hukum Newton. Tarik menarik diantara objek juga mematuhi hukum gravitasi dan teori relativitas umum Einstein. Dalam dunia *game*, untuk objek bereaksi sama seperti di dunia nyata, diperlukan pembuatan system yang sama melalui pemrograman. *Unreal physics engine* menggunakan PhysX engine, dibuat oleh NVIDIA, untuk melakukan kalkukasi interaksi fisik seperti dikehidupan nyata, seperti tabrakan dan dinamika fluida.[12]

**2.2.3 *The Graphic Engine***

Untuk sebuah gambar dapat ditampilkan dilayar, gambar tersebut harus sudah di *render* ke tampilan layar yang dituju(seperti *PC/TV* atau *mobile devices*) *graphic engine* bertanggung jawab pada output di layar dengann mengambil informasi tentang keseluruhan *scene* seperti *color, texture, geometry, The Shadow* dari objek dan cahaya, dan titik lihat dari *scene*, dan mempertimbangkan interaksi silang dari faktor-faktor yang mempengaruhi keseluruhan warna, cahaya, bayangan, dan oklusi objek.

*Engine* ini dapat digunakan untuk membuat permainan segala *platforms*(PC, Xbox, PlayStation, dan *mobile devices*). Didukung DirectX 11/12, OpenGL, dan JavaScript/WebGL rendering.[12]

**2.2.4 *Input and The Gameplay Framework***

Unreal Engine terdiri dari sistem *input* yang mengubah *key* dan menekan tombol oleh pemain menjadi aksi yang dilakukan oleh karakter dalam *game*. System *input* ini dapat dikonfigurasi melalui *Gameplay framework*. *Gameplay framework* memiliki fungsi untuk melacak progress permainan dan mengatur aturan dari *game*. *Heads-up* *display(HUDs)/user interfaces(UIs)* adalah bagian dari *Gameplay framework* untuk menyediakan umpan balik untuk pemain selama jalannya permainan. Kelas *gameplay*  seperti *GameMode, GameState,* dan *PlayerState* mengatur aturan dan mengontrol keadaan permainan. Karakter yang berada dalam *game* dikontrol antara pemain(menggnakan kelas *PlayerController)* atau *AI*(menggunakan kelas *AIController*). Karakter dalam *game*, apakah dikendalikan oleh pemain atau *AI*, merupakan bagian dari kelas dasar yang diketahui dengan kelas *Pawn*. Kelas *Character* merupakan subset dari kelas *Pawn*, yang dimana dibuat spesifik untuk representasi pemain berorientasi vertikal, sebagai contoh, manusia.[12]

Dengan *Unreal Gameplay framework* dan *controllers* ditempat, itu memungkinkan untuk kustomisasi perilaku dan fleksibilitas pemain.[12]

**2.2.5 *Light and Shadow***

Cahaya adalah alat yang kuat dalam pembuatan *game*. Unreal Engine 4 menyediakan satu set dari dasar cahaya yang dapat dengan mudah diletakkan di *game level*. Mereka adalah *Directional Light, Point Light, Spot Light,* dan *Sky Light*.

*Directional* *Light* memancarkan sinar lampu parallel, *Point Light* memancarkan cahaya seperti bohlam(dari satu titik secara radial kearah luar di semua arah), *Spot Light* memancarkan cahaya dalam bentuk luar kerucut, dan *Sky Light* meniru cahaya dari langit ke bawah pada objek di *level*.[12]

**2.2.6 *Post-Process Effects***

*Post-process effects* adalah efek yang ditambahkan di akhir untuk meningkatkan kualitas dalam *scene*. UE4 *post-process effects* meliputi *Anti-Aliasing* menggunakan *Temporal Anti-Aliasing(TXAA),* *Bloom, Color Grading, Depth of Field, Eye Adaptation, Lens Flare, Post Process Materials, Scene Fringe, Screen Space Reflection,* dan *Vignette*.[12]

**2.2.7 *Artificial Intelligence***

*Artificial intelligence* atau kecerdasan buatan manusia yang dibuat untuk dapat meniru manusia di dunia nyata. Manusia menciptakan *AI* untuk diberikan benda otak, kemampuan berpikir, dan mengambil keputusan.[12]

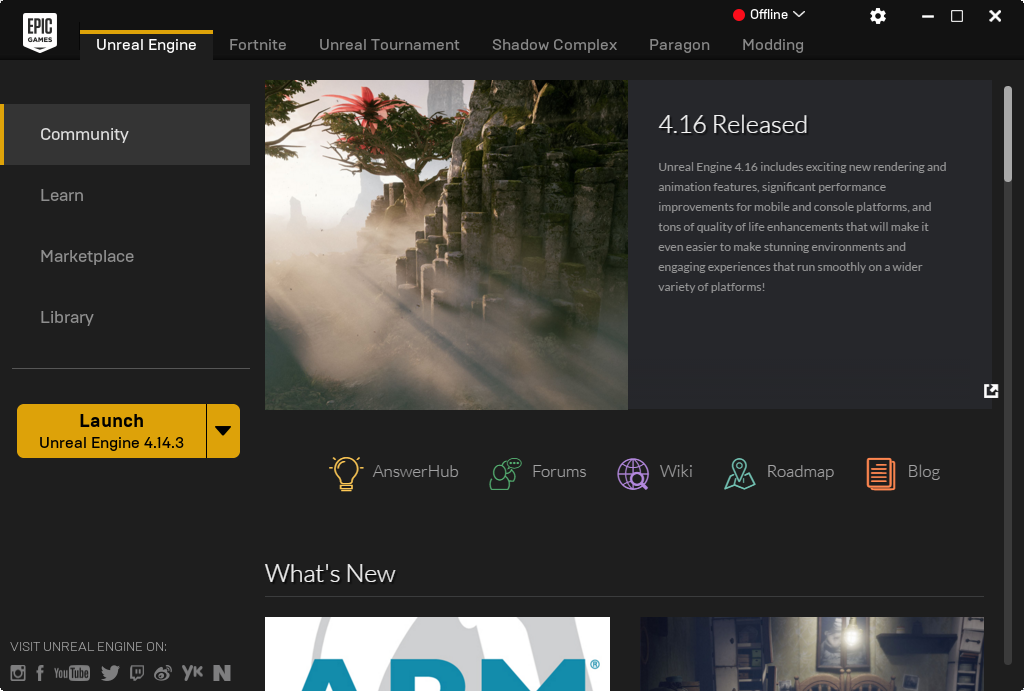
Dalam *game*, *NPC(Non Player Character)* diberikan beberapa bentuk AI yang memungkinkan pemain berinteraksi denganya. Seperti, memberikan *NPC* kemampuan untuk mencari lokasi untuk menyerang. Jika diserang, mereka akan lari, bersembunyi, dan mencari lokasi yang sesuai untuk melawan balik. Unreal Engine 4 menyediakan dasar yang bagus untuk *AI* dan fondasi dasar untuk di kustomisasi dan diperbaiki *AI* sebagai *NPC* dalam *game*.[12]

**2.2.8 *Online and Multiplatform Capabilities***

Unreal Engine 4 menyediakan kemampuan untuk membuat permainan untuk berbagai *platforms*. Ketika membuat *game* menggunakan Unreal Engine 4, itu portable ke berbagai macam *platform*, seperti *Web*, iOS, Linux, Windows, dan Android. Unreak Engine 4 memiliki *online subsystem* yang menyediakan kemampuan untuk integrasi fungsi yang tersedia di Xbox Live, Facebook, Steam, dan lainya.[12]

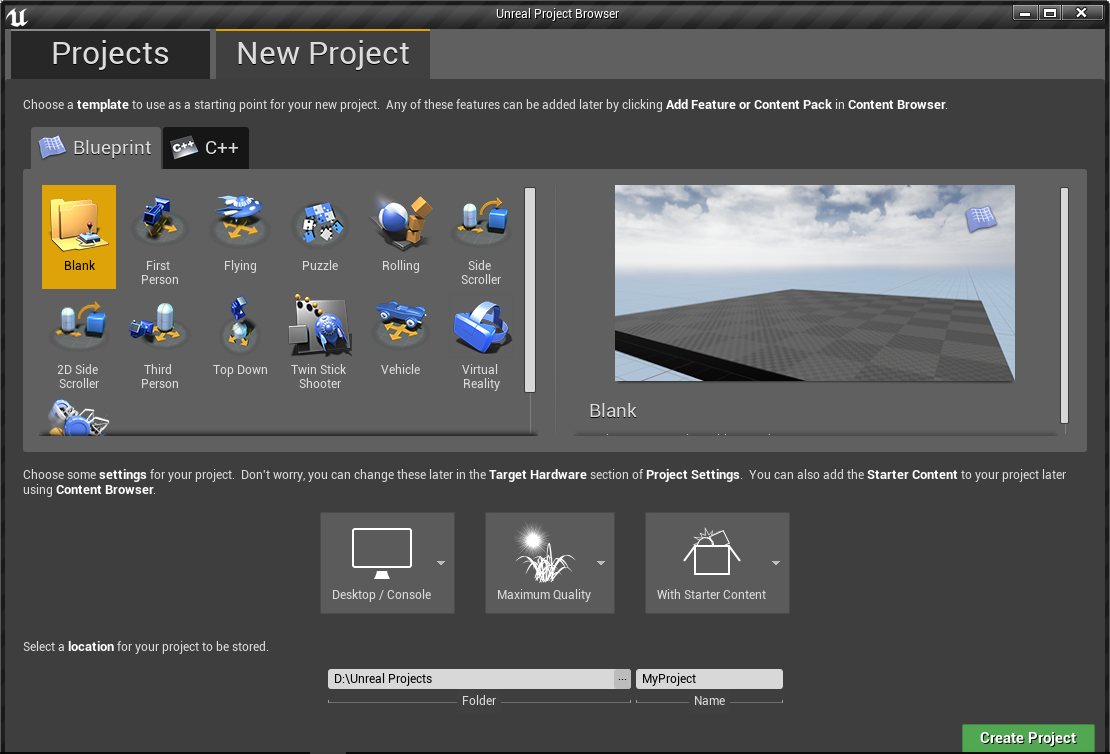
**2.3 *Unreal Editor***

Unreal Editor menydeiakan tampilan visual yang terbuat oleh *viewports* dan jendela yang memungkinakn untuk impor, mengatur, edit, dan memberikan perilaku/interaksi ke aset permainan. Berikut adalah tampilan dari *start menu*: (**Gambar 2.1)**



**Gambar 2.1** *Start Menu*

*Start menu* mempunyai fitur memungkinkan pengguna utuk memilih versi dari *game engine* yang diinginkan untuk dijalanan. Terdapat akses ke *marketplace*, yang dimana *library* tempat contoh permainan yang dibuat oleh pembuat lainnya, yang dimana dapat di unduh(beberapa gratis, beberapa berbayar). Menu ini menyediakan pembaharuan dan berita terakhir dari Epic untuk memastikan pembuat tetap sejalan dengan pembaharuan dan perubahan terbaru. Berikut adalah tampilan dari *project browser*: (**Gambar 2.2)**



**Gambar 2.2** *Project Browser*

*Project Browser* menyediakan opsi untuk membuat *game level* yang sudah dikustomisasi. Pada *project browser* dapat memilih tipe permainan seperti *first-person* *shooting* *level* dan pengaturan *third-person game*, atau 2D/3D *side-scrolling platform level* langsung baik dalam *Blueprint* atau C++ sebagai *template dasar*. Pada tab *New Project* dapat memilih target perangkat (PC/*mobile*), kualitas gambar, dengan atau tidak dengan konten Unreal yang termasuk di *startup project*. Berikut adalah tampilan dari *Content Browser*: (**Gambar 2.3**)



**Gambar 2.3** *Content Browser*

*Content Browser* adalah jendela dimana pengguna dapat menemukan semua konten (aset) yang dimiliki. Aset dikategorikan menjadi beberapa direktori seperi *Audio, Materials, Animations, Particle Effects,* dan lainya. Jendela ini memiliki tombol *import,* yang dimana dapat membawa aset yang dibuat pada aplikasi yang berbeda kedalam permainan. Berikut adalah tampilan *toolbar*: (**Gambar 2.4)**



**Gambar 2.4** *Toolbar*

*Toolbar* menyediakan akses cepat kealat dan editor. Tampilan dasar meliputi akses cepat ke *blueprint* dan *Matinee editors*. *Quick play* dan *launch game* termasuk ke dalam standar tampilan. Berikut adalah tampilan *viewport*: (**Gambar 2.5**)



**Gambar 2.5** *Viewport*

*Viewport* adalah jendela dalam dunia permainan apa yang dilihat sesuai dengan yang di permainan. Jendela ini untuk meletakkan dan memindahkan objek. Ketika tombol *play* ditekkan, *viewport* menjadi hidup dan dapat berinteraksi dengan *game level*. Berikut adalah tampilan *scene outliner*: (**Gambar 2.6)**



**Gambar 2.6** *Scene Outliner*

*Scene Outliner* memiliki daftar dari objek yang berada didalam *scene*. Hanya apa yang dimuat didalam *scene* yang terdaftar di *scene outliner*. Berikut adalah tampilan *Modes* : (**Gambar 2.7**)



**Gambar 2.7** *Modes*

Jendela *Modes* memiliki kemampuan untuk membuat dan meletakkan objek ke dunia *game*. Terdapat *Place, Paint, Landscape, Foliage* dan *Geometry Editing* yang dapat dipilih. *Place* adalah untuk meletakkan objek ke dunia *game*. *Paint* adalah untuk mewarnai simpul dan tekstur dari objek. *Landscape* dan *Foliage* adalah alat yang berguna untuk membuat dataran natural berskala besar dalam permainan. *Geometry Editing* menyediakan alat untuk modifikasi dan edit objek.

**2.4 Blueprint *Visual Scripting System***

Unreal *editor* menawarkan kemampuan untuk membuat *custom events* untuk *game level* melalui system *visual scripting*. Sebelum Unreal Engine 4, dikenal sebagai Kismet system. Dalam Unreal Engine 4, system ini dirubah dengan tambahan fitur dan kemampuan. System terbaru ini rilis dengan nama *Blueprint*.[12]

Terdapat beberapa tipe dari *blueprint*: *Class Blueprint, Data-only Blueprint,* dan *Level Blueprint*. Dengan blueprint, memiliki kemampuan untuk mengontrol jalannya permainan tanpa harus melalui *coding* yang sesungguhnya. Itu membuat mudah ­*non-programmers* untuk mendesain dan modifikasi jalannya permainan.[12]

*Blueprint* seperti *coding* dengan tampilan antarmuka. Fungsi node pada *Blueprint* dapat di *select, drag,* dan *drop* secara logika untuk membangkitkan respon yang diingikan secara spesifik dalam permainan. *Blueprint* memiliki kemampuan untuk prototipe, implementasi, atau modifikasi elemen dalam *gameplay* secara virtual.[12]

**2.5 Blender**

Blender adalah perangkat lunak 3D modeling dan animasi yang berstatus gratis dan *open source*. Blender adalah program komputer grafis yang dapat membuat gambar dan animasi dalam tiga dimensi dengan kualitas tinggi.[14]

Blender integrasi penuh pembuatan konten 3D, menawarkan berbagai alat penting, diantaranya, *modeling, rendering,* animasi, edit video, VFX, *compositing, texturing, rigging*, banyak tipe simulasi dan pembuatan permainan. *Cross platform*, dengan *GUI* OpenGL yang seragam disemua *platform* utama(Dapat disesuaikan dengan skrip Python). Kualitas tinggi arsitektur 3D yang mempercepat dan efisien dalam alur kerja pembuatan. Karena *Open Source* blender sangat didukung oleh komunitas dan forum.[1]

**2.6 Fuse**

Fuse adalah sebuah aplikasi yang digunakan untuk membuat karakter 3D. Fuse dibuat dengan kolaborasi antara Mixamo dan Vladlen Koltun grup peneliti di Universitas Stanford, dengan Siddhartha sebagai kepala arsitek dan programmer. Mixamo bertujuan untuk mendemokratisasikan pemodelan karakter dan perakitan, memberikan pengguna control penuh atas keunikan karakter sekaligus menjaga kualitas aset tetap tinggi. Mixamo memungkinkan integrasi langsung dan mulus dengan *Auto-Rigger* dan koleksi animasi.[4]

Fuse adalah sebuah aplikasi *standalone* pembuat karakter 3D yang memungkinkan untuk membuat karakter yang unik untuk digunakan didalam video permainan. Fuse hadir dengan 70 lebih bagian tubuh, 150 lebih mesh baju, 42 lebih zat teksturing dinamik, gratis tanpa batas *Auto-Rigs* dan animasi melalui servis Mixamo(membutuhkan Adobe ID).[3]

*Auto-Rigger* adalah cara mudah yang disediakan website Mixamo untuk membuat karakter 3D siap untuk dianimasi. Pada normalnya, *rigging* sebuah karakter membutuhkan professional, dengan *Auto-Rig* dari website Mixamo hanya membutuhkan beberapa menit saja.[3]

**2.7** **Pseudocode**

Pseudocode adalah cara penulisan algoritma yang menyerupai bahasa pemrograman tingkat tinggi. Pseudocode menggunakan bahasa yang hamper menyerupai bahasa pemrograman. Biasanya pseudocode menggunakan bahasa yang mudah dipahami secara universal dan juga lebih ringkas dari pada algoritma. Pseudocode berisi deskripsi dari algoritma pemrograman computer yang menggunakan struktur sederhana dari beberapa bahasa pemrograman tetapi bahasa tersebut hanya ditujukan agar dapat dibaca manusia. Sehingga pseudocode tidak dapat dipahami oleh computer. Supaya notasi pseudocode biasa dipahami oleh computer maka harus diterjemahkan terlebih dahulu menjadi sintaks bahasa pemrograman tertentu.

Dalam pseudocode, tidak ada sintaks standar resmi. Karena itu, pseudocode ini dapat diterapkan dalam berbagai bahasa pemrograman. Disarankan menggunakan *keyword* yang umum digunakan seperti: *if, then, else, while, do, repeat, for,* dan lainya. Keuntungan menggunakan notasi pseudocode adalah kemudahan mentranslasi ke notasi bahasa pemrograman, karena terdapat korespondensi antara setiap pseudocode dengan notasi bahasa pemrogaman. Berikut adalah **Tabel 2.1** yang menunjukan perbandingan beberapa kata yang biasa digunakan dalam penulisan algoritma dengan menggunakan kalimat deskriptif dan pseudocode.[11]

**Tabel 2.1** Tabel perbandingan kalimat deskriptif dan pseudocode

|  |  |
| --- | --- |
| Kalimat Deskriptif | Pseudocode |
| Masukkan panjang | Input panjang  Read panjang  Baca panjang |
| Hitung luas dengan rumus panjang x lebar | Luas ← panjang \* lebar  Output Luas  Print Luas  Write Luas |
| Jika sudah selesai, cetak luas | If kondisi\_selesai == true then print luas |
| Nilai B ditambah 5 | B ← B+5 |
| Jika nilai A lebih kecil dari 5 maka nilai B dibagi 3 | If A<5 then B←B/3 |
| Jika nilai A lebih besar dari nilai B maka tampilkan A, jika A lebih kecil dari B maka tampilkan nilai B | If A>B then print A else print B |

Struktur penulisan pseudocode secara umum sama dengan struktur penulisan algoritma dengan menggunakan kalimat deskriptif yaitu dimulai dari judul/header, deklarasi/kamus dan diakhiri dengan deskripsi. Meskipun tidak ada sintaks khusus dalam penulisan pseudocode, tetapi terkadang pseudocode dituliskan dengan menggunakan gaya penulisan dari beberapa bahasa pemrograman yang ada, seperti Fortran, Pascal, C, dan lain-lain.[11]

**2.8 Struktur Navigasi**

Struktur navigasi adalah struktur (alur) dari suatu program yang merupakan rancangan hubungan dan rantai kerja dari beberapa area yang berbeda dan dapat membantu mengorganisasikan seluruh elemen pembuatan website dengan memberikan perintah atau pesan. Ada empat macam bentuk dasar dari struktur navigasi, yaitu :

A. Struktur Navigasi Linier

Struktur navigasi linier yaitu struktur yang digunakan untuk menelusuri program secara berurutan:

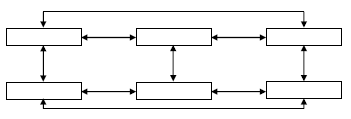


**Gambar 2.8** Struktur Navigasi Linier

Struktur navigasi linier menampilkan tampilan program secara berurutan. Struktur navigasi linier dapat menampilkan satu halaman setelah atau sebelumnya.

B. Struktur Navigasi Non-Linier

Struktur navigasi non-linier memungkinkan pemakai menelusuri program tanpa dibatasi oleh rute. Berikut adalah gambar struktur dasar navigasi non-linier :

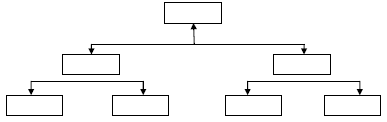


**Gambar 2.9** Struktur Navigasi Non-linier

Pada struktur navigasi non-linier diperkenankan menelusuri program tanpa harus berurutan. Percabangan pada struktur non-linier ini berbeda dengan percabangan pada struktur hirarki, karena percabangan pada non-linier ini walaupun terdapat mempunyai kedudukan yang sama.

C. Struktur Navigasi Hirarki

Struktur navigasi hirarki adalah struktur navigasi yang memungkinkan pemakai menelusuri program melalui titik-titik percabangan dari suatu struktur pohon. Berikut adalah gambar struktur dasar navigasi hirarki :

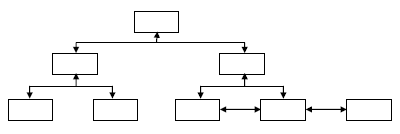


**Gambar 2.10** Struktur Navigasi Hirarki

Struktur navigasi hirarki memiliki 2 jenis kriteria halaman, yaitu halaman utama dan halaman pendukung. Halaman utama bisa digambarkan seperti *Main Menu* dalam sebuah permainan, sementara halaman pendukung adalah halaman yang dipilih pada piliha menu di halaman utama.

D. Struktur Navigasi Campuran

Struktur navigasi campuran adalah struktur navigasi yang memungkinkan pemakai dengan bebas menelusuri program, pada bagian tertentu gerakkan dibatasi secara hirarki ataupun linier. Berikut adalah gambar struktur dasar navigasi campuran :



**Gambar 2.11** Struktur Navigasi Campuran

Struktur navigasi campuran biasa digunakan dalam pembuatan permainan video, karena permainan video dapat bebas kembali ke menu awal atau menuju ke *stage* tertentu.