

BAB II

LANDASAN TEORI

2.1. Pengertian *Game*

Game berasal dari bahasa Inggris yang memiliki arti dasar permainan. Permainan yang dimaksud merujuk pada pengertian “kelincahan intelektual”. Kelincahan intelektual pada tingkat tertentu merupakan ukuran sejauh mana *Game* itu menarik untuk dimainkan secara maksimal (Prayono,2012).

Game merupakan sebuah bentuk seni dimana penggunanya disebut dengan pemain (player), diharuskan membuat keputusan – keputusan dengan tujuan untuk mengelola sumber daya yang diperoleh dari kesempatan – kesempatan bermain miliknya, untuk mencapai suatu tujuan tertentu. Video *Game* adalah bentuk *Game* yang interaksi utamanya melibatkan media video dan juga audio.

Berdasarkan representasi visualnya, *Game* dapat dibedakan menjadi 2 jenis yaitu *Game* 2 Dimensi (2D) atau 3 Dimensi (3D). *Game* 2D adalah *Game* yang secara matematis hanya melibatkan 2 elemen koordinat kartesius, yaitu X dan Y, sehingga konsep kamera pada *Game* 2D hanya menentukan “gambar” mana pada *Game* yang dapat dilihat oleh pemain. Sedangkan *Game* 3D selain melibatkan koordinat X dan Y, juga melibatkan elemen Z pada perhitungannya, sehingga konsep kamera pada *Game* 3D benar – benar menyerupai konsep kamera pada kehidupan nyata, yaitu selain digeser, juga dapat diputar dengan sumbu – sumbu tertentu.

Berikut ini adalah macam – macam jenis *Game* berdasarkan cara pembuatannya, cara pemasarannya dan mesin yang menjalankannya. Jenis *Game* diantaranya yaitu :

1. *Game* PC



Gambar 2. 1 Contoh *Game* PC (One Piece Pirate Warrior 3)

Game yang dimainkan pada PC (Personal Computer) yang memiliki kelebihan yaitu memiliki tampilan antarmuka yang baik untuk input maupun output, output visual kualitas tinggi karena layar komputer biasanya memiliki resolusi yang jauh lebih tinggi dibandingkan dengan layar televisi biasa. Kekurangannya adalah spesifikasi komputer yang sangat bervariasi antara satu komputer dengan komputer lainnya yang menyebabkan beberapa *Game* dapat ditampilkan dengan baik pada satu komputer, tetapi tidak berjalan dengan baik pada komputer yang lainnya.

2. *Game* Console



Gambar 2. 2 Contoh *Game* Console (Naruto Series)

Game yang dijalankan pada suatu mesin spesifik yang biasanya tersedia dirumah pribadi, seperti Microsoft Xbox 360, Nintendo Wii, Playstation, dan lain – lain.

3. *Game* Arcade



Gambar 2.3 Contoh Game Arcade

Game yang dijalankan pada mesin dengan input dan output audio visual yang telah terintegrasi dan tersedia ditempat – tempat umum, seperti mall (Time Zone).

2.1.1. Cerita Rakyat

Menurut halaman dari situs web (dongengceritarakyat.com) dijelaskan bahwa, Cerita rakyat adalah cerita atau kisah yang asal muasalnya bersumber dari masyarakat serta tumbuh berkembang di masa lampau. Cerita ini menjadi ciri khas (tradisi/budaya) pada bangsa (kota/tempat) tersebut, yang mempunyai kultur budaya dengan keanekaragaman termasuk didalamnya khasanah kekayaan budaya serta sejarah pada setiap bangsa (kota/tempat) yang diceritakan. Biasanya, cerita rakyat menceritakan suatu peristiwa pada suatu tempat sehingga menjadi legenda asal muasal tempat tersebut. Tokoh yang berperan pada cerita rakyat biasanya berwujud manusia, binatang, hingga para dewa. Cerita rakyat biasanya lahir dan hidup di tengah masyarakat dari suatu daerah yang ceritanya kemudian diwariskan dari generasi pertama ke generasi yang akan datang. Cerita rakyat merupakan suatu sastra daerah setempat yaitu sastra yang dituturkan dengan bahasa daerah setempat seperti cerita rakyat Jawa Tengah yang dituturkan dengan bahasa Jawa, Cerita Jawa Barat yang dituturkan dengan bahasa Sunda, begitu juga dengan Padang, Aceh, dan Papua menceritakan cerita rakyatnya dengan bahasa daerah mereka sendiri.

Ciri – ciri cerita rakyat adalah sebagai berikut :

1. Disampaikan secara lisan. Salah satu sifat utama cerita rakyat adalah cara penyampaiannya dari mulut ke mulut. Pada umumnya, cerita rakyat disampaikan secara individu kepada individu lain atau sekelompok individu.
2. Memiliki banyak variasi. Cerita rakyat merupakan suatu yang dinamis, dimana seiring waktu mengalami perubahan seperti penambahan ataupun pengurangan.
3. Bersifat anonim. Artinya cerita rakyat tidak memiliki penulis yang jelas dan kapan ditulisnya tidak ada yang tahu.
4. Memiliki unsur pendidikan. Cerita rakyat memiliki nilai budaya dan pesan-pesan moral sehingga dapat dijadikan pelajaran bagi orang yang mendengarnya.

2.1.2. Jenis- Jenis Cerita Rakyat

Cerita rakyat memiliki berbagai macam jenis antara lain :

1. Dongeng

Dongeng menurut James Danandjaja dalam dosenpendidikan.com (2016) merupakan cerita rakyat lisan yang tidak dianggap benar-benar terjadi oleh empunya cerita. Dongeng juga tidak terikat oleh tempat maupun waktu, karena dongeng diceritakan terutama untuk menghibur. Dongeng merupakan karya sastra fiktif yang diceritakan untuk menghibur dan memiliki pesan moral yang baik untuk anak-anak.

2. Fabel

Cerita yang menceritakan kehidupan hewan yang berperilaku menyerupai manusia. Fabel adalah cerita fiksi atau khayalan belaka (fantasi). Kadang kala fabel memasukkan karakter minoritas berupa manusia.(Wikipedia.org, 2020).

3. Mite

Secara etimologi bahasa, Mite atau mitos berasal dari bahasa Yunani “Mythos” yang berarti cerita dewa. Makaryk dalam Burhan (2004) menjelaskan bahwa Mitos merupakan cerita masa lampau yang

dipahami sebagai sebuah cerita yang berkaitan dengan dewa-dewa atau tentang kehidupan supernatural yang lain, juga sering mengangund sifat pendewaan manusia. Mitos kebanyakan menampilkan cerita tentang kepahlawanan, asal- usul alam, manusia Contoh cerita rakyat lokal yang termasuk mite adalah Legenda Dewi Sri, Mite Nyi Roro Kidul. Sedangkan cerita rakyat internasional yang termasuk mite adalah kisah dewa Zeus dalam Mitologi Yunani.

4. Legenda

Menurut Firdaus dan Faizah dalam Jurnal Cerita Rakyat Masyarakat Rambah Kabupaten Rokan Hulu Provinsi Riau (2013) Legenda merupakan cerita yang mencerminkan kehidupan masyarakat setempat yang bersifat sekuler, terjadi pada masa lampau, dan bertempat di dunia seperti yang dikenal. Legenda terbagi menjadi empat jenis yaitu Legenda keagamaan yang menceritakan kisah-kisah para wali penyebar agama Islam (Sunan Kalijaga, Syekh Siti Jenar, Ratu Calon Arang), Legenda kegaiban yang mengisahkan kepercayaan rakyat pada alam gaib (contohnya kerajaan gaib orang Bunian di Sumatra, kerajaan gaib di Pajajaran di Jawa Barat), Legenda perseorangan yang menceritakan tokoh tertentu yang pernah ada dan terjadi (Contohnya Si Pitung dari Jakarta, Lutung Kasarung dari Jawa Barat, Rara Mendut dan Jaka Tingkir dari Jawa Tengah, dan Suramenggolo dari Jawa Timur, serta Legenda lokal yang berhubungan dengan nama tempat terjadinya gunung, bukit, danau, dan sebagainya.

5. Sage

Sage merupakan cerita rakyat yang berhubungan dengan sejarah dimana sage menceritakan tentang kepahlawanan, keajaiaban, kesaktian, dan juga keberanian suatu tokoh. Contoh yang termasuk sage adalah Damar Wulan, Caadara, Calon Arang, dan lain-lain.

2.1.3. Timun Mas

Di kutip dari Jurnal “Cerita Rakyat Timun Emas : Gambaran

Runtuhnya Ideologi Patriarkal” (Reno Wulan Sari, 2016). Sebagai Landasan teori dari tema jurnal ini.

Cerita rakyat Timun Emas adalah kisah seorang perempuan yang sangat ingin memiliki seorang anak. Perempuan tersebut bernama Nyi Niken. Telah bertahun-tahun lamanya Nyi Niken dan suaminya berusaha untuk memperoleh keturunan. Namun, segala usaha yang dilakukan menjadi sia-sia ketika menyadari bahwa Nyi Niken tak kunjung hamil. Harapan itupun segara pupus ketika suami Nyi Niken meninggal dunia karena terlalu keras berusaha untuk menghadirkan keturunan dengan berbagai pengobatan. Dalam keadaan yang merasa sepi setelah sepeninggal suaminya, Nyi Niken akhirnya mengambil keputusan yang cukup berat. Ia meminta bantuan raksasa Ki Butakala untuk memberinya seorang anak sebagai teman hidupnya. Ki Butakala memberikan syarat jika nanti anak yang dilahirkan adalah laki-laki maka anak tersebut akan menjadi milik Nyi Niken seumur hidup, namun jika anak itu terlahir perempuan maka harus segera diberikan kepada Ki Butakala sebagai bahan santapannya. Nyi Niken menyetujui segala persyaratan dengan harapan bahwa anak yang kelak lahir adalah laki-laki. Ki Butakala memberikan petunjuk buah kelapa yang harus diminum oleh Nyi Niken. Setelah meminum buah kelapa yang jatuh tepat di depannya saat perjalanan pulang, Nyi Niken pun hamil. Setelah beberapa bulan maka lahirlah anak perempuan yang cantik dan diberi nama Timun Emas. Sesuai dengan perjanjiannya dengan Ki Butakala, maka pada saat Timun Emas tumbuh remaja Ki Butakala datang untuk mengambilnya. Pada saat itu, Nyi Niken berusaha untuk mengulur waktu dengan berbagai janji yang diberikannya. Ki Butakala setuju dengan perjanjian tersebut dan akan kembali datang setelah Timun Emas benar-benar menjadi dewasa. Pada saat itu, munculah pangeran yang jatuh cinta kepada Timun Emas.

Menyadari nasibnya akan kehilangan putri tunggalnya, maka Nyi Niken meminta bantuan Ki Pertapa untuk menyelamatkan Timun Emas. Ki Pertapa memberikan beberapa barang yang harus dipergunakan Timun Emas ketika ia dikejar oleh Ki Butakala. Dan benar, barang-barang tersebut mampu

menyelamatkan nasib Timun Emas dari cengkraman Ki Butakala. Maka, Nyi Niken merasa bahagia, terlebih Timun Emas menyetujui keinginan Ibunya agar ia segera menikah dengan salah seorang pangeran yang selama ini mengharapkannya.

2.1.4. Unity 5.5

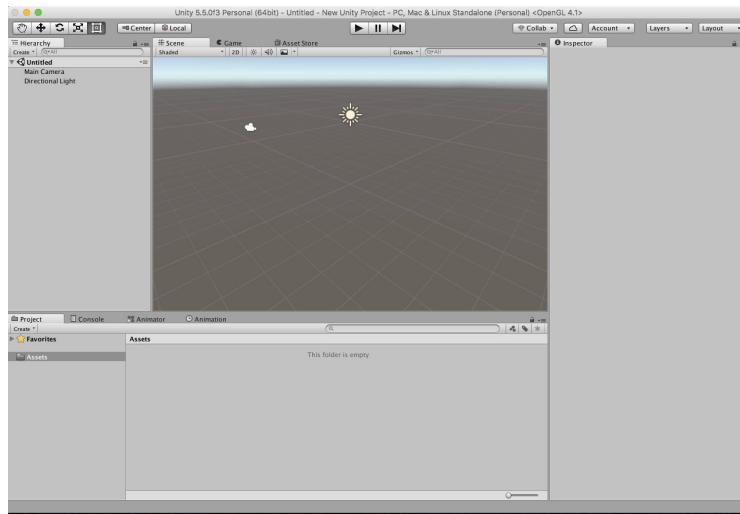
Unity 3D adalah sebuah *Game engine* yang berbasis crossplatform. Unity dapat digunakan untuk membuat sebuah *Game* yang bisa digunakan pada perangkat komputer, Android atau IOS, PS3 dan bahkan X-Box.(Habibie, 2012)

Unity adalah sebuah tool yang terintegrasi untuk membuat sebuah *Game*, arsitektur bangunan dan simulasi. Unity bisa untuk *Games PC* dan *Games Online* untuk *Game* online diperlukannya sebuah plugin, yaitu Unity Web Player.

Unity tidak dirancang untuk proses desain atau modelling, dikarenakan unity bukan tool untuk mendesain. Jika ingin mendesain, dipergunakan editor lain seperti Blender untuk 3D atau Corel Draw untuk 2D.

Fitur Scripting yang disediakan, mendukung 3 bahasa pemrograman, JavaScript, C#, dan Boo. Flexible dan EasyMoving, rotating dan scaling *objects* hanya perlu sebaris kode. Begitu juga dengan duplicating, removing, dan changing properties. Visual Properties Variables yang di definisikan dengan scripts ditampilkan pada Editor. Bisa digeser, di drag and drop, bisa memilih warna dengan color picker. Berbasis .NET. Artinya penjalanan program dilakukan dengan Open Source .NET Platform, Mono atau bisa juga dengan Visual Studio.

Dikutip dari web www.dicoding.com unity memiliki banyak window dan tab pada tampilan awal. Tak hanya hanya itu, bahkan bagi yang sudah menggunakan Unity dalam waktu yang cukup lama pun belum tentu sudah mengenal semua komponen UI yang ada pada Unity, berikut untuk tampilan awalnya.



Gambar 2. 4 Tampilan awal Unity

Ketika pertama kali membuka Unity dan setup project, akan muncul halaman yang berisikan beberapa window terbuka. Ada lima window utama yang akan paling sering digunakan ketika mengembangkan *Game* dan menggunakan Unity. Kelima window tersebut adalah window *Scene*, *Project*, *Hierarchy*, *Inspector*, dan *Game*. Secara umum, berikut adalah penjelasan dari masing - masing window :

- a. *Scene*: Window yang digunakan untuk membangun *Game*. Di dalamnya bisa melihat dan mengatur *object* di dalam sebuah *scene*
- b. *Project*: window yang berisikan seluruh aset yang digunakan untuk membuat proyek *Game* yang bisa terdiri dari file, script, audio, dan lain – lain yang juga disimpan di dalam hardrive komputer.
- c. *Hierarchy*: Window yang berisi *Gameobject* atau kumpulan *Gameobject* yang kita gunakan di dalam *scene*.
- d. *Inspector*: Window yang menampilkan konteks atau keterangan dari *object* atau aset yang sedang kita pilih.
- e. *Game*: Window ini digunakan untuk melihat tampilan ketika permainan di jalankan.
- f. *Toolbar* : Toolbar berisikan tombol yang membantu dalam mengatur berbagai komponen di dalam permainan.

2.1.4.1. *Scene*

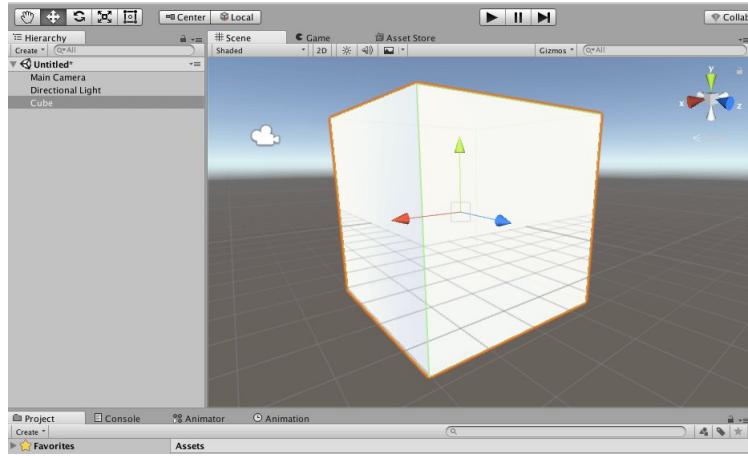
Window *scene* digunakan untuk melihat secara visual *Game* yang akan di bangun. Dengan transform tools yang berada pada toolbar Unity, bisa melakukan modifikasi kepada *Gameobject* yang akan dimasukkan ke dalam *scene*.



Gambar 2. 5 Icon Transform Tools pada Unity

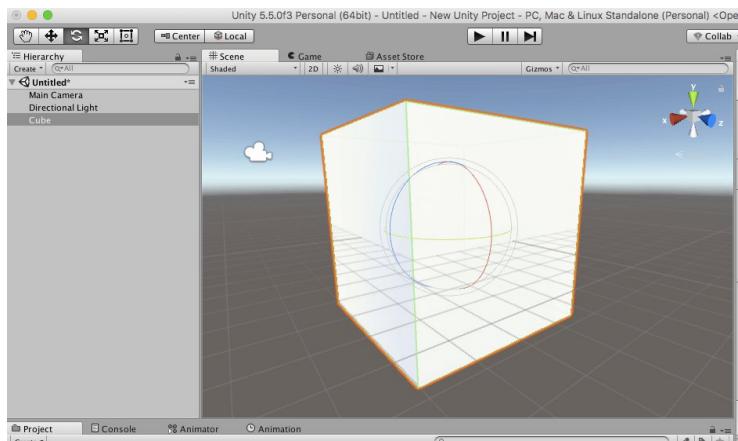
Ada lima tools yang bisa di gunakan untuk memodifikasi *GameObject* atau biasa disebut dengan transform tools. Diberi nama transform tools karena komponen *GameObject* yang akan di modifikasi adalah komponen transform yang di dalamnya terdapat informasi posisi, rotasi dan skala/ukuran berikut adalah fungsi untuk masing – masing transform tools:

- a. *Hand tools* : digunakan untuk menggerakan posisi sudut pandang di dalam *scene*.
- b. *Position transform tools* : digunakan untuk mengubah komponen posisi sebuah *Gameobject* yang terhadap sumbu x, y, dan z. Terdapat handle yang bisa membantu kita untuk mengubah posisi terhadap satu sumbu saja misalnya sumbu merah bisa di drag untuk menggerakan terhadap sumbu x saja atau bisa drag icon kotak di antara tiga panah (merah, hijau, dan biru) untuk menggerakan *object* terhadap komponen x,y, dan z.



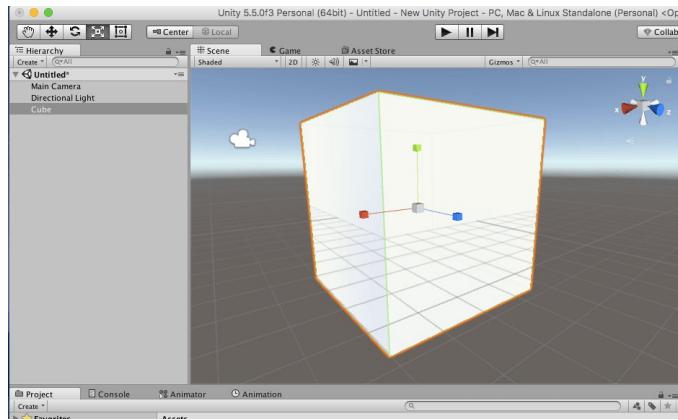
Gambar 2. 6 Tampilan dari Position Transform tools

- c. *Rotation transform tools* : Digunakan untuk mengubah komponen rotasi pada sebuah *Gameobject* terhadap sumbu x, y, dan z. Sama seperti tools lainnya, terdapat juga ikon sumbu yang akan membantu dalam merotasi objek terhadap sumbu saja.



Gambar 2. 7 Tampilan dari Rotation transform tools

- d. *Scale transform tools* : Digunakan untuk mengubah komponen scale/ukuran dari sebuah *Gameobject* terhadap sumbu x,y, dan z. Terdapat juga sumbu bantuan untuk mengubah *scale* terhadap satu sumbu saja.



Gambar 2. 8 Tampilan Dari Scale Transform Tools

- e. *Rectangle transform tools* : Digunakan untuk mengubah komponen dari sudut pandang 2D. Biasanya tools ini akan sangat membantu jika ingin memodifikasi komponen UI dalam *Game*.

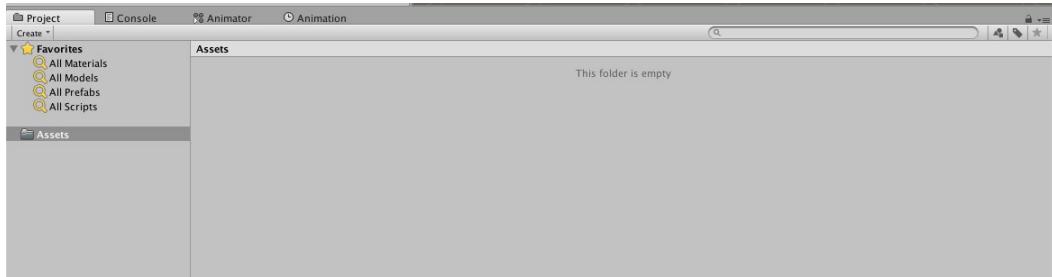
Kemudian di dalam window *Scene* terdapat toolbars yang cukup banyak, antara lain :



Gambar 2. 9 Tampilan Toolbars Windows *Scene*

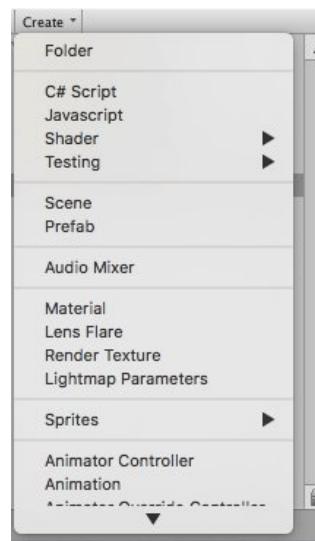
- a. *Shaded* : Digunakan untuk mengganti mode tampilan objek – objek di dalam *Gamescene*.
- b. 2D : Digunakan untuk mengubah perspektif dari 3D menjadi 2D atau sebaliknya.
- c. *Scene Lighting* : Mengatur tampilan pencahayaan pada *scene*.
- d. *Audio* : Mematikan atau menyalakan audio di dalam *scene*.
- e. *Efek* : Menampilkan atau menghilangkan efek dari *Skybox*, *Flares*, *Fog*, dan lain –lain di dalam *scene*.
- f. *Gizmos* : Digunakan untuk memilih objek – objek yang ditampilkan di dalam *scene*.
- g. *Search Bar* : Digunakan untuk mencari sebuah *Gameobject* di dalam *scene*.

2.1.4.2. Project Window



Gambar 2. 10 Tampilan Project Windows Pada Unity

Project window digunakan untuk mengorganisir aset yang digunakan. Struktur file yang dibuat di dalam project window akan disimpan dalam struktur yang sama. Di dalam project window, kita bisa memasukan aset yang akan kita gunakan. Untuk membuat aset baru di dalam sebuah folder pada project window, bisa klik tombol Create di bagian atas project window atau bisa klik kanan lalu pilih tulisan Create dan bisa memilih akan membuat aset baru di dalam bentuk apa. Bisa membuat sebuah folder kosong untuk merapikan struktur penyimpanan.



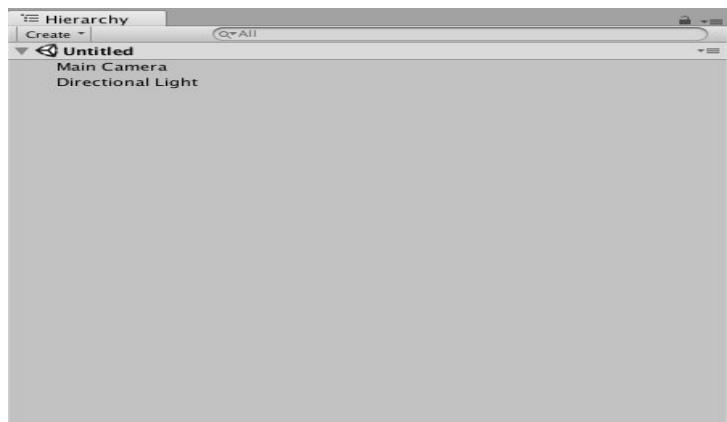
Gambar 2. 11 Contoh Pembuatan Aset Baru

Selain membuat secara manuak, bisa juga memasukan aset dari sumber eksternal. Memasukan aset dengan cara import aset ataupun drag and drop aset dari sumber file nya ke dalam window project secara langsung. Lalu ketika ingin menggunakan aset tersebut

ke dalam *Game*, hanya memerlukan drag and drop dari project window ke *scene* window

2.1.4.3. Hierarchy

Hierarchy berisikan seluruh *GameObject* yang ada di dalam *Gamescene*. Jika di dalam project windw adalah aset yang di dalam harddisk, hierarchy berisikan aset yang di guanakan di dalam *scene*. Bisa memasukan aset dari project window ke hierarchy dengan cara drag and drop atau juga bisa create asset menggunakan drop down create yang ada di toolbar hierarchy



Gambar 2. 12 Tampilan Hierarchy pada Unity

Susunan aset yang ditampilkan di dalam hierarchy berdasarkan waktu kemunculannya. Namun kita juga bisa mengubah susunan tersebut seara manual. Lalu jika *Game* objek dihapus dari *Gamescene*, maka di dalam hierarchy juga akan hilang. Kemudan di dalam hierarchy ada konsep parent and child. Kita bisa memasukan sebuah objek menjadi child dari objek lain. Caranya adalah dengan drag and drop *GameObject* di hierarchy ke dalam *GameObject* lain yang ingin dijadikan parent.



Gambar 2. 13 Konsep Parent And Child Hierarchy

Ketika kita melakukan transformasi elemen dari sebuah *GameObject*, kalau *GameObject* tersebut tidak memiliki parent, maka perubahan akan terjadi relatif terhadap world space. Sedangkan kalau memiliki parent *object*, transformasi relatif terhadap parent. Lalu apabila kita melakukan transformasi kepada parent *object*, maka child *object* juga akan terkena efeknya.

2.1.4.4. Inspector

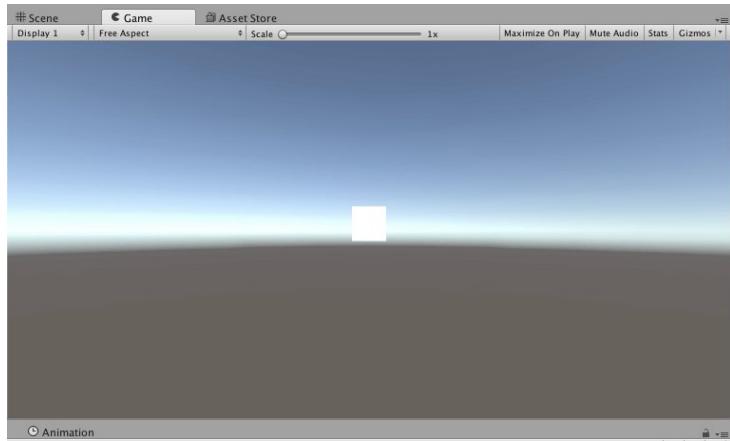


Gambar 2. 14 Tampilan Inspector pada Unity

Inspector adalah tempat untuk mengubah nilai dari properti dan setting. Window inspector bisa berbeda tampilan dan fungsinya untuk tiap komponen yang berbeda. Inspector bisa menampilkan informasi dari player setting, asset, *GameObject*, *scene*, *prefab*, dan lain-lain. Jika kita sedang melihat sebuah *GameObject*, maka yang ditampilkan oleh 36 inspector adalah komponen yang ada di dalamnya. Pada umumnya bagian atas akan berisikan nama *GameObject* yang sedang dilihat beserta beberapa informasi seperti

tag dan layer, lalu di bagian bawahnya adalah komponen yang terdapat dalam *Gameobject* tersebut. Satu hal yang pasti, setiap *Gameobject* pasti memiliki header dan komponen transform.

2.1.4.5. Gameview



Gambar 2. 15 Tampilan Gameview pada Unity

Windows *Game* view merupakan halaman untuk menampilkan current state dari *Game* ketika dijalankan. Ada tiga opsi yang bisa kita lakukan ketika ingin masuk ke *Game* view yakni play, pause, dan play frame by frame. Pada Window *Game* view terdapat beberapa tools di toolbar yang bisa membantu kita mengubah bagaimana *Game* kita ditampilkan di layar. Display dropdown menu digunakan untuk memilih kamera yang ada di *scene* yang akan menampilkan *Gameobject* di layar. Lalu terdapat dropdown free aspect yang bisa kita gunakan untuk mengubah resolusi dan rasio dari layar *Game*. Menu ini cocok untuk testing *Game* pada tampilan layar yang berbeda semisal untuk android, iphone, atau PC. Kita juga bisa membuat custom resolusi menggunakan menu ini. Lalu ada scale yang berguna untuk zoom in dan zoom out. Maximize on Play digunakan untuk menampilkan window *Game* view secara full screen ketika diplay. Mute audio digunakan untuk mematikan audio di dalam *Game*. Stats berguna untuk menampilkan status dari *Game*, terutama dari sisi resource yang digunakan. Gizmos digunakan untuk memilih dan

menampilkan *Gameobject* tertentu di dalam *scene*. Ketika kita masuk ke menu *Game* mode dengan menekan tombol play, kita bisa mengubah value dari property di inspector secara run time. Namun perubahan itu hanya terjadi di play mode saja, ketika kita kembali ke editing mode, value yang kita ubah selama di play mode tidak akan tersimpan. Untuk membantu mengingatkan kita jika sedang masuk ke play mode, tampilan layar Unity akan menjadi lebih gelap apabila sedang masuk ke play mode.

2.1.5. Microsoft Visual Studio

Merupakan sebuah perangkat lunak yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi windows, ataupun aplikasi web. Visual studio mencakup kompiler, SDK, Integratared Development Environtment (IDE), dan dokumnetasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe. Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di windows) ataupun managed code (bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (sumber : wikipedia.org).

2.1.6. Bahasa Pemrograman C#

Microsoft membuat C# seiring dengan pembuatan Framework .NET. Chief Architect dalam pembuatan C# adalah Anders Hejlsberg yang sebelumnya berperan dalam pembuatan Borland Delphi dan Turbo Pascal. C# menjajikan produktifitas dan kemudahan yang ada di Visual basic dengan kemampuan dan fleksibiitas yang ada di C/C++. C# adalah salah satu dari banyak bahasa

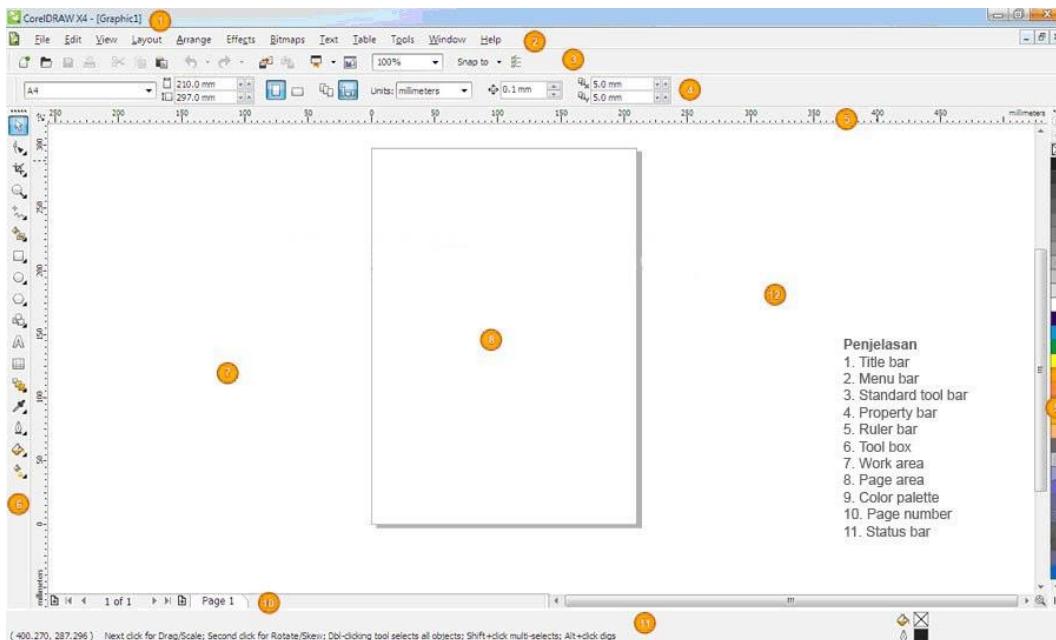
yang bisa dipakai untuk pemrograman .NET. Kelebihan utama bahasa ini sintaksnya yang mirip C, namun lebih mudah dan lebih bersih. Untuk perbandingan penulis cantumkan informasi Managed C++ dan Visual Basic .NET.

- a. Managed C++: Managed C++ adalah ekstensi terhadap C++ untuk membuat program .NET. salah satu keunikan Managed C++ adalah kita bisa mencampur kode – kode managed dengan unmanaged dalam pemrograman kita. Ini akan sangat berguna bagi pihak – pihak yang sudah memiliki banyak kode C++ namun ingin bermigrasi ke platform .NET. Dalam pemrograman Managed C++ kita masih akan terikat dengan konsep – konsep sulit C++ sehingga produktifitas akan lebih rendah dibanding jika kita menggunakan C#
- b. Visual Basic .NET: Perbedaan antara C# dengan Visual Basic .NET yang akan langsung terlihat adalah sintaksnya. C# memiliki beberapa fitur yang tidak ada di Visual Basic .NET sehingga C# sedikit lebih fleksibel. Perlu diketahui bahwa Visual basic .NET adalah bahasa yang sepenuhnya berorientasi objek dan dibuat untuk pemrograman .NET
(sumber : <https://id.wikipedia.org/>)

2.1.7. Corel Draw

Menurut Rahmat Widiyanto dalam bukunya “Teknik Profesional CorelDraw”, definisi dari corel draw adalah editor grafik vector yang dibuat oleh corel, Corel sendiri adalah sebuah perusahaan perangkat luak yang bermarkas di Ottawa, Kanada. Corel Draw pada awalnya dikembangkan untuk dijalankan ada Sistem Operasi Windows 2000 dan yang lebih baru. Corel Draw sendiri adalah sebuah program komputer untuk melakukan editing pada garis vector(Rahmat Widiyanto, 2008). Karena kegunaan Corel draw adalah sebagai alat untuk pengolahan gambar, maka program ini sering digunakan untuk menyelesaikan pekerjaan di bidang publikasi atau percetakan maupun bidang yang lain yang butuh proses visualisasi Kegunaan Corel Draw itu sebenarnya sangat banyak, biasanya di gunakan untuk

membuat Logo, desain, menggambar ilustrasi, membuat brosur dan lain – lain. Disini penulis untuk membuat sebuah Asset *Game* antara lain, Karakter, *object GUI* dan masih banyak lagi. Corel draw juga memiliki tampilan yang cukup simpel dan mudah di ingat untuk tata letak dari menu – menu yang ada dicorel draw.



Gambar 2. 16 Tampilan lembar kerja dari Corel Draw

- Menu Bar : Area yang berada dibawah title bar yang berisi kumpulan perintah – perintah yang muncul ketika mengklik teks – teks yang ada didalamnya.
- Property bar : Kumpulan property, opsi dan perintah yang muncul sesuai dengan tool atau objek yang sedang aktif saat itu. Contohnya jika sebuah teks tool aktif pada saat itu, berarti perintah yang muncul pada property tool berhubungan dengan teks.
- Toolbar : Kumpulan perintah langsung (Shortcut) untuk menu bar dan perintah lain.
- Title bar : Area yang menampilkan nama file CorelDraw yang sedang aktif
- Rulers : Garis horizontal dan vertikal yang mengelilingi drawing window yang berfungsi menentukan ukuran dan posisi suatu objek.
- Toolbox : Kumpulan tool yang digunakan untuk membuat, mengedit

dan memodifikasi objek yang dibuat.

- f. Drawing window : Area diluar drawing page yang dibatasi oleh scroll bar dikanan dan dibawah.
- g. Drawing Page : Area segi empat ditengah drawing window. Area ini sekaligus merupakan area yang muncul ketika mencetak hasil karya kita nanti.
- h. Color palette : Area yang disebelah kanan yang berisikan kumpulan kotak warna sesuai yang diinginkan, Docker, sebuah jendela yang berisikan perintah – perintah yang berhubungan dengan tool/perintah tertentu yang diinginkan.
- i. Status bar : Area yang paling bawah jendela utama coreldraw yang berisi informasi objek yang terseleksi serta property yang dimilikinya, seperti teks, ukuran, warna, resolusi dan lain –lain.
- j. Document navigator : Area dikiri bawah untuk mengontrol atau menambah halaman (page).
- k. Navigator : Tombol disudut kanan bawah yang menampilkan jendela kecil untuk membantu kita menentukan posisi ketika menggambar.

2.1.8. Usecase

Usecase diagram digunakan untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya. Diagram Usecase tidak menjelaskan secara detail tentang penggunaan usecase, namun hanya memberi gambaran singkat hubungan antara aktor, dan sistem, Melalui diagram usecase dapat diketahui fungsi – fungsi apa saja yang ada pada sistem. Nama suatu usecase harus didefinisikan sesimple mungkin dan dapat dipahami. (Fatimah, 2014). Berikut ini adalah simbol – simbol dari usecase.

Tabel 2. 1 Simbol-simbol Usecase

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan bahwa peran yang penting dalam kaitan berinteraksi dengan use case.
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya dan yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya (objek induk (<i>ancestor</i>)).
4		<i>Include</i>	Menspesifikasi bahwa use case sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasi bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.

6		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		System	Menspesifikasiakan paket yang menampilkan sistem secara terbatas.
8		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemennya (sinergi).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.1.9. Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing – masing alur berawal, kejadian yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, dimana sebagian besar state adalah action dan sebagian besar transisi ditrigger oleh selesainya state sebelumnya (Internal Processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses – proses dan jalur – jalur aktivitas dari level atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses.

Dipakai pada businnes modeling untuk memperlihatkan urutan aktifitas proses bisnis. (Sari,2012).

Tabel 2. 2 Simbol – simbol Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actifity</i>	Memperlihatkan bagaimana masing-masing kelas antar mukasaling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan ekspektasi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk pada awalnya.
4		<i>Actifity Final</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

2.1.10. Class Diagram

Class diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Kelas memiliki 3 bagian utama yaitu attribute, operation, dan name. Kelas – kelas pada struktur sistem harus melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. (Fatimah,2014).

Tabel 2. 3 Simbol- simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
2		<i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan multiplicity
3		Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi
4		Kebergantungan / Dependency	Relasi antar kelas dengan makna kebergantungan antar kelas
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran