BAB II

LANDASAN TEORI

2.1 Game Adventure

2.1.1 Pengertian Game

Menurut Sallen dan Zimmerman (2003), "Game adalah sebuah sistem dimana pemainberinteraksi dalam sebuah konflik artifisial yang digerakkan oleh peraturan-peraturan yang menghasilkan hasil yang kuantitatif."

Menurut Callois (Ignasius, 2014), game adalah aktivitas yang mencakup karakteristik berikut : *fun* (bebas bermain adalah sebuah pilihan, bukan kewajiban), *separate* (terpisah), *uncertain, non-productive, governed by rules* (ada aturan) dan *fictitious* (pura-pura)

Sebagai kata benda, game berarti aktivitas mental atau fisik yang memiliki peraturan tertentu dan dilakukan oleh orang-orang untuk mencari kesenangan.

2.1.2 Pengertian Game Adventure

Menurut Rollings dan Adams (2006), *Game adventure* adalahpermainandimana pemain mengendalikan pemeran protagonis dalam sebuah cerita interaktif dalam permainan yang melibatkan penjelajahan dan *puzzle solving*.

Menurut Pedersen (2003), Tujuan utama dari game adventure adalah penyelesaian dari cerita yang diangkat. Sedangkan menurut kondisi yang paling umum yang mengindikasikan kegagalan player dalam bermain adalah kematian dari karakter utama, meskipun masih banyak alternatif kegagalan lain tergantung dari misi yang dijalankan.

2.2 Game Pembelajaran

Menurut Ismail (2006), *Education games* (permainan edukatif) adalah suatu kegiatan yang sangat menyenangkandan dapat merupakan cara atau alat pendidikan yang bersifat mendidik.Dari uraian diatas dapat disimpulkan bahwa *educationgames*(permainan edukatif) adalah sebuah permainan yang digunakan dalamprosespembelajaran dan dalam permainan tersebut mengandung unsur mendidikatau nilai-nilai pendidikan.Selain itu, untuk pemilihan permainan, diusahakan agarseluruh aspekyang dimiliki anak dapat berkembang dengan baik, baik dari segi kognitif,afektif dan juga psikomotorik. Oleh karena itu perlu ditunjang alat bantu yangtepat saat bermain. Adapun kriteria-kriteria pemilihan alat bantu tersebut agarpermainan dapat membantu belajar secara optimal dan tidak terjadi kekeliruandalam menyelesaikan dan menentukan alat dan bahan yang diperlukan secaratepat guna.

2.3Elemen Game

Menurut Schell (2008), terdapat 4 elemen *Game* yang saling berkaitan, yaitu:

1. Mekanisme

Mekanisme merupakan suatu prosedur dan aturan dalam sebuah *Game*. Mekanisme tersebut menentukan seorang pemain dapat atau tidak mencapai tujuannya, dan apa yang terjadi ketika pemain berusaha untuk mencapainya.

Mekanisme sendiri terdiri dari 6 kategori yaitu:

a. *Space*

Setiap *Game* pastinya akan terjadi pada suatu tempat tertentu yang disebut *Space* (ruang), merupakan sebuah *Magic circle* pada suatu *Game*. Ruang ini menunjuk pada bermacam-macam tempat yang ada di *Game* dan ruang-ruang ini saling berkaitan antara satu dengan yang lainnya.

b. *Objects, Attributes and States*

Objek bisa berupa karakter, *scoreboards*, maupun properti tambahan yang terdapat didalam game. Selain itu didalam suatu objek juga terdapat yang disebut atribut, sebagai contoh kecepatanKecepatan merupakan atribut suatu objek, lalu kecepatan tersebut juga mempunyai *State* (status) yang dapat berupa angka sebagai contoh 40 *Kmph.*

c. *Actions*

Action adalah segala sesuatu yang dikerjakan oleh karakter maupun objek yang terdapat didalam *Game*.

d. *Rules*

Rules merupakan suatu keadaan yang menentukan suatu ruang yang ada, suatu objek, suatu aksi yang dapat dilakukan serta konsekuensi yang dihasilkan, dan membatasi aksi yang dapat dilakukan untuk mencapai tujuan akhir dari *Game* tersebut.

e. *Skills*

Suatu *Game* yang dimainkan menuntut pemain untuk memiliki suatu *skill* tertentu untuk menyelesaikan tantangan yang ada. Apabila *skill* pemain sesuai dengan tingkat kesulitan *Game* yang dimainkan maka pemain akan merasa tertantang dan terus memainkan *Game* tersebut.

f. *Chance*

Chance merupakan gabungan dari semua mekanisme yang ada, *Chance* merupakan sebuah elemen yang penting didalam sebuah *Game*, karena *Chance* merupakan sebuah ketidakpastian, dan ketidakpastian berarti kejutan. Kejutan adalah sebuah sumber kepuasan seseorang dan sumber dari kesenangan.

2. Cerita

Cerita merupakan sebuah rangkaian event yang terjadi dalam sebuah game, bisa jadi sebuah cerita terasa datar atau bisa juga bercabang. Ketika akan menyampaikan cerita dalam sebuah *Game*, maka harus dipilih suatu mekanika yang sesuai agar memperkuat cerita tersebut.

3. Estetika

Estetika menentukan bagaimana tampilan, dan keseluruhan sebuah *Game*. Estetika merupakan hal yang penting, karena estetika berhubungan langsung dengan pemain.

4. Teknologi

Teknologi merupakan gabungan dari semua elemen-elemen pembuatan *Game* yang ada, tidak boleh ada yang lebih banyak satu dengan yang lain, karena akan menghancurkan estetika *game* tersebut. Selain itu teknologi juga merupakan suatu fasilitas untuk membuat sebuah game menjadi nyata.

2.4 Genre Game

Genre Game merupakan salah satu hal yang harus diperhitungkan dalam pembuatan *game*, karena berhubungan dengan *gameplay* yang dihasilkan. Didalam buku karya Fullerton (2008), terdapat 11 *Genre Game*, yaitu:

a. Action Games

Action Games menggabungkan antara kecepatan reaksi pemain, pengendalian mata dan tangan pemain.

b. Strategy Games

Strategy Games terfokus pada taktik, perencanaan serta manajemen unit dan sumber daya yang ada.

c. Role-playing Games

Role-playing Games berkutat pada pembuatan dan pengembangan karakter. Biasanya game pada genre ini memiliki alur cerita yang kaya yang terikat pada quest tertentu.

d. Sports Games

Sports Games merupakan simulasi dari permainan yang ada di dunia nyata seperti tennis, sepakbola, baseball dan yang lainnya.

e. Racing Games

Terdapat 2 macam *Racing Games*, yaitu *arcade styles* dan *racing simulators.* Keduanya mengusung tema balapan namun memiliki gaya permainan yang berbeda.

f. Simulation/Building Games

Simulation/Building Game berfokus pada manajemen sumber daya yang digabungkan dengan pembangunan, bisa membangun sebuah perusahaan maupun sebuah kota. Tidak seperti strategy games yang berfokus pada menguasai. Kebanyakan *simulation/building game* meniru sistem yang ada di dunia nyata dan memberikan pemain kesempatan untuk mengatur perkembangan kota/perusahaannya sendiri.

g. Flight and Other Simulations

Simulations Game dibuat berdasarkan aktivitas yang ada di dunia nyata. Seperti mengendarai pesawat terbang atau mengendarai *tank. Simulations Game* mengharuskan pemain untuk menguasai kemampuan tertentu untuk memainkan *game* tersebut.

h. Adventure Games

Adventure games mengkhususkan pada eksplorasi, mengkoleksi barang-barang dan menyelesaikan *puzzle.* Pada awalnya *adventure games* didesain hanya berisi *text*, dan deksripsi dari tempat-tempat yang ada.

i. Edutainment

Edutainment menggabungkan antara pembelajaran dan kesenangan bermain. Tujuan akhirnya adalah menyenangkan pemain sekaligus memberikan pelajaran kepada *user*.

j. Children's Games

Children's Games didesain khusus untuk anak yang berusia diantara 2 sampai 12 tahun. Pada *genre* ini diselipkan nilai edukasinya, namun lebih ditekankan pada nilai kesenangan yang dihasilkan.

k. Casual Games

Casual games didesain agar bisa dinikmati oleh semua orang, baik laki-laki maupun wanita, tua dan muda. *Casual games* dapat menggunakan *genre* apapun dan *gameplay* yang mudah untuk menarik *audience* untuk memainkan game ini.

2.5 Pengetikan sepuluh jari

Menurut Yenni (2007), upaya untuk meningkatkan kemampuan siswa dalam mengetik 10 jari belum dapat terlaksana sesuai dengan tujuan yang hendak dicapai. Hal ini terlihat dari cara kerja siswa yang belum mengefektifkan ke 10 jarinya dengan benar waktu proses belajar mengajarberlangsung, ditambah dengan disiplin diri siswa yang sangat kurang untuk dapat menguasai mengetik10 jari dengan benar.

Menurut Eka (2015), dalam dunia kerja yang berbasis teknologi internet dan komputer, kemampuan mengetik cepat dengan menggunakan 10 jari menjadi sangat penting. Ketika mengetik dengan 10 jari, pekerjaan menjadi lebih cepat selesai.,dan secara tidak langsung dapat menaikkan nilai diri dalam dunia kerja.

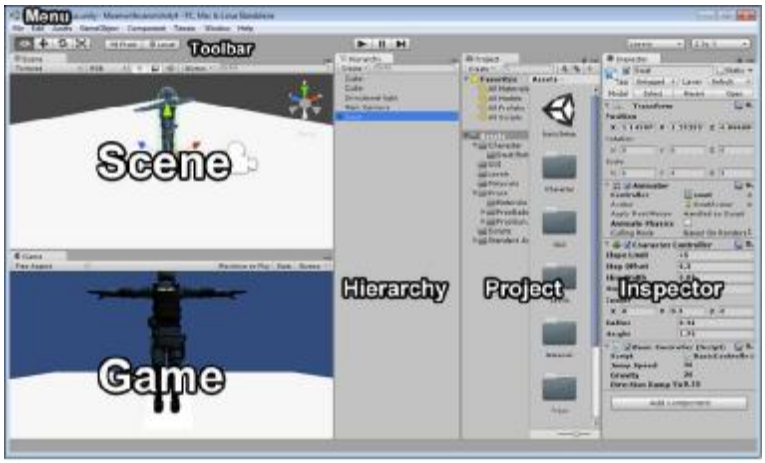
2.6Unity3D Game Engine

Menurut Roedawan (2014), *Unity* adalah sebuah 3D *game engine* multiplatformyang dapat di gunakan untuk membuat dan mengembangkan *game* dengan berbagai macam pilihan platform, seperti Windows, Mac, Linux, iOS, Android, XBox 360, Wii U, dan PS3 serta web-based menggunakan Unity plugins. Unity berbasis object-oriented, yang artinya setiap object dalam unity mempunyai class dan setiap class dapat berasal dari "base" atau "parent" class. Sebagai game engine dengan basis object-oriented, unity memiliki hierarchy class.

Setiap *scene* terdiri dari banyak *game object*, di dalam game engine lain sering di kenal *entities.* Masing-masing memiliki nama, posisi, dan orientasi. Selain itu, game object juga bisa di hubungkan dengan parent-child yang posisi orientasi nya bergantung pada game object induknya.Sistem *scripting*pada *unity* di dasarkan pada *mono*, dimana *mono* merupakan versi *open-source* dari *.net*. Layaknya *.net*, *mono* mendukung banyak bahasa pemrograman dalam pengembangannya, namun pada *unity* hanya mendukung bahasa pemrograman *c#*,*boo*, dan *javascript.* Dalam *unity*, *scripts* merupakan sebuah komponen dan dapat di bubuhkan pada *game object* seperti seperti komponen lainnya, hal ini bisa di lihat di *Unity Manual* pada bagian *using scipts* yang menjelaskan bagaimana cara membuat *scripts* dan membubuhkannya ke *game object.*

2.6.1 Memahami dan Mengoptimalisasi Antarmuka Pengguna

Dalam antarmuka unity terdapat beberapa mode tata letak yang dapat di pilih. Tata letak *Unity* selalu dimulai dengan mode *Wide*, hal ini berbeda apabila pengguna telah merubahnya. Cara merubah nya pada akses ke *Window-Layouts* dan memilih opsi lain, seperti 4 *Split* atau 2 *by 3*.



Gambar 1.1 Interface pada unity

1. **Scene:** Tampilan ini digunakan untuk *position*, *rotate*, *scale*, dan *select* objek pada *game* dan juga mengarahkan level pengguna.
2. **Game:** Tempat untuk memainkan dan menguji *game*. Ini akan menghasilan pengalaman bermain dari *player* seakurat mungkin.
3. **Hierarchy:** Berisi list objek-objek yang ada di *game* (seperti karakter, kamera, level geometri, pencahayaan, dan tekstur GUI)
4. **Project:** Tempat untuk membuat, mengatur, dan mengakses asset-aset *game* dari model 3D dan tekstur 2D sampai *script* C#, seluruh elemen akan dilist disini.
5. **Inspector:** Melakukan konfigurasi berbagai objek *game* (dari tampilan *Hierarchy*) atau asset (dari tampilan *Project*). Termasuk merubah peraturan ***Transform,*** mengkonfigurasi komponen yang telah ada dan membuat komponen baru. Pada *inspector* juga dapat dilakukan penyesuaian preferensi lain untuk *game*.
6. **Toolbar:** Termasuk merubah alat yang digunakan untuk memanipulasi objek permainan dan navigasi adegan, alat control yang digunakan untuk

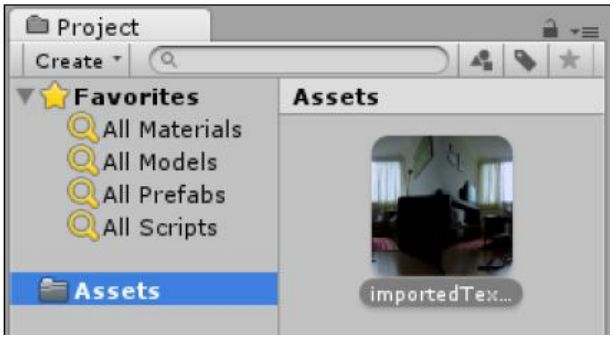
bermain/berhenti dan menghentikan level, dan alat *drop-down* yang digunakan untuk mengelola *layers* dan *layouts*.

7. **Menu:** Memberikan akses ke beragam daftar perintah yang meliputi impor/ekspor asset, pengaturan preferensi, pembuatan objek permainan, medan, tata letak, dan dokumentasi.

2.6.2 Import Konten

Dalam pengembangan sebuah game, para pengembang memerlukan aset tambahan di luar aset unity. Maka dari itu, para pengembang mengimport aset dan konten. Setelah membuat sebuah model 3D, audio clip, movie clip, atau texture, bisa diimpor ke dalam project yang diingin. Berikut adalah langkah-langkah untuk mengimpor sebuah asset ke dalam menu:

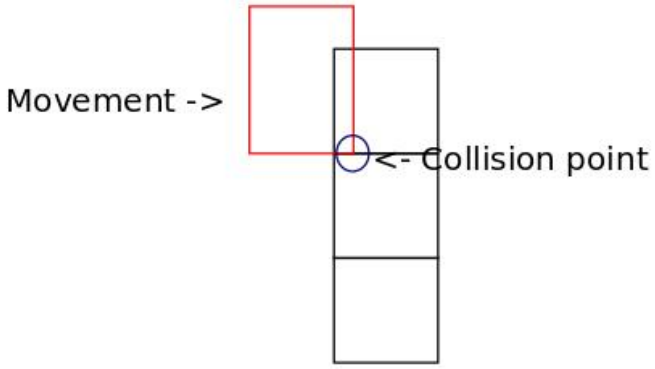
1. Dalam editor *Unity*, akses menu *Assets*
2. Pilih opsi *Import New Asset.*
3. Cari file yang diinginkan, lalu pilih impor
4. File tersebut sudah masuk ke dalam list *Project* seperti gambar dibawah ini:



Gambar 2.2 Tampilan List Project

2.7 Collision Events

Menurut Roedawan (2014), *collision event* adalah keadaan dimana suatu obyek bertumbukan dengan obyek lain.*Collision event* yang terjadi dapat ditindaklanjuti dengan reaksi yang dapat diatur dalam *action* yang merupakan hasil dari tumbukan tersebut. Normalnya, bila tidak ada *action* yang ditentukan saat terjadi *collision*, obyek yang bersifat *solid* tidak akan bisa menembus obyek *solid* lainnya. Namun bila salah satu obyek ada yang *non-solid*, maka kedua obyek tersebut akan menembus obyek yang lain.



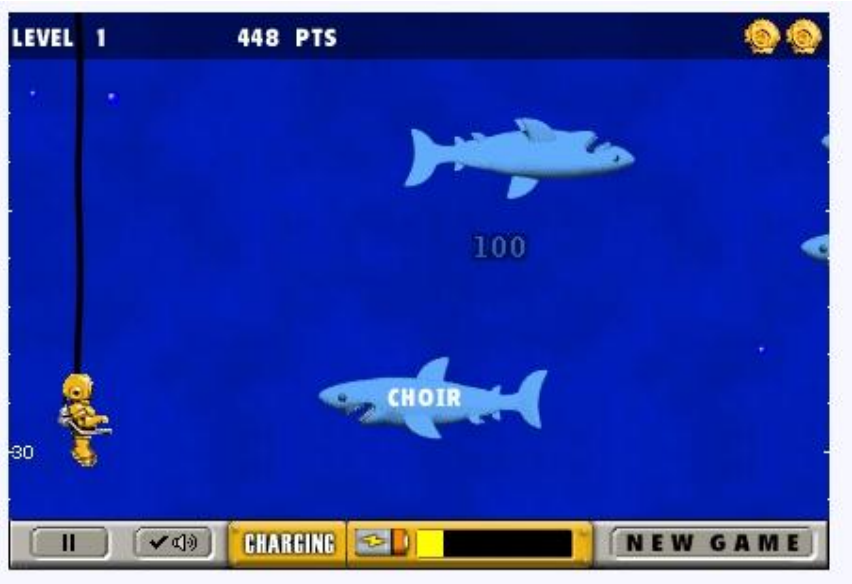
Gambar 2.3 Ilustrasi teori Collision

Ada banyak penggunaan *collision event* dalam pembuatan game. *Collision event* dapat digunakan untuk membuat obyek tidak bisa menembus tembok, atau menghancurkan obyek lain seperti misalnya bila obyek tersebut tertembak peluru.

Gambar 2.4 Collision yang terjadi antara Mario dan bata pada game Super Mario Brothers

2.8 Keyboard Events

Menurut Roedawan (2014),*keyboard events* terjadi ketika pemain menekan tombol pada*keyboard*. Ada berbagai event yang dapat diwakili oleh tiap tombol yang ditekan. Salah satunya adalah <*No key*>. *Event* ini terjadi jika tidak ada sama sekali tombol yang ditekan. Dan *keyboard events*khusus yang lain adalah<*Any key*>yang terjadi pada saat pemain menekan tombol apapun. Ketika pemain menekan beberapa tombol, maka akan terjadi beberapa *event* sesuai dengan tombol yang ditekan



Gambar 2.5 Contoh Keyboard Event pada game Typer Shark

2.9 Action / shooter

Menurut Roedawan (2014), *Action* mengindikasikan kejadian yang akan terjadi sebagai akibat dari sebuah event. Ketika sebuah action ditempatkan pada suatu event, pembuat game dapat memberika parameter action sesuai nilai yang diinginkan.

Beberapa contoh *action* tersebut antara lain :

1. Hspeed : Action ini berfungsi untuk menggerakkan obyek secara horizontal ke arah kanan jika Hspeed bernilai positif dan kekiri jika Hspeed bernilai negatif. Opsi relative digunakan untuk menambah kecepatan horizontal dari kecepatan sebelumnya atau dengan kata lain Hspeed yang bersifat incremental.
2. Vspeed : Action ini berfungsi untuk menggerakkan obyek secara vertikal ke arah atas jika Vspeed bernilai positif dan ke bawah jika Vspeed bernilai negatif. Opsi relative digunakan untuk menambah kecepatan

vertikal dari kecepatan sebelumnya atau dengan kata lain Vspeed yang bersifat incremental.

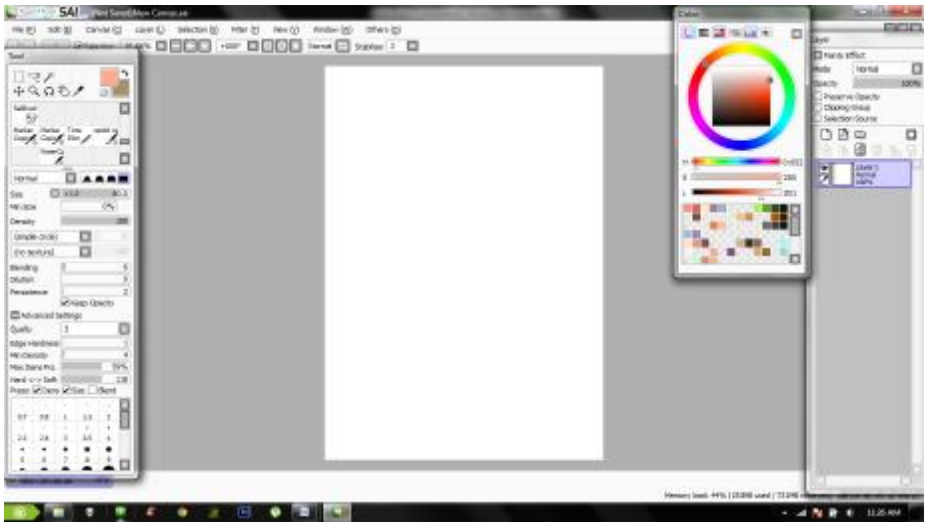
3. Change Sprite :Digunakan untuk mengganti sprite yang telah dipasangkan untuk suatu obyek. Mengganti sprite adalah salah satu fitur penting. Change sprite digunakan pada saat kita ingin mengganti sprite karakter sesuai dengan arah pergerakannya. Dan diletakkan pada event keyboard sesuai dengan tombol arah yang ditekan.

2.10Digital Painting

Digital Painting berasal dari kata digital dan paint, yang berarti sebagai proses membuat ilustrasi atau lukisan menggunakan program dan tools yang sesuai. Digital Painting seringkali menggunakan pen tablet untuk memudahkan menggambar digital.

2.11 Paint Tool Sai

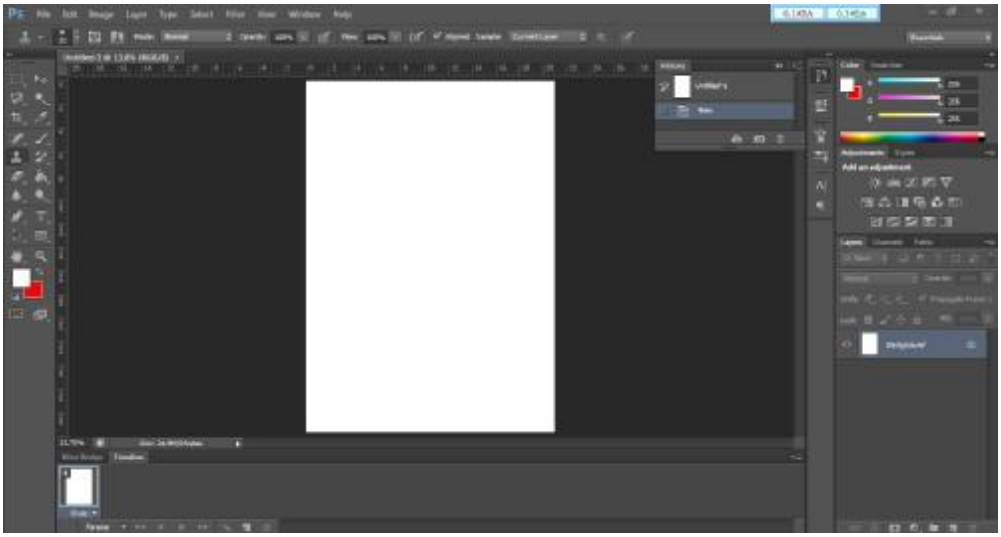
Paint Tool Sai adalah aplikasi untuk membuat ilustrasi grafis, sangat cocok untuk membuat ilustrasi / digital painting. Paint Tool Sai tidak hanya mampu membaca atau menulis file bereksitensi SAI, namun juga PSD, JPG, serta PNG.



Gambar 2.6 Layout Paint Tool Sai

2.12Adobe Photoshop

Adobe Photoshop adalah aplikasi pengedit grafis yang dikembangkan oleh Adobe Systems. File Photoshop atau dikenal sebagai file berekstensi .PSD, menyimpan hasil gambar lengkap dengan masks, transparansi, teks dan lain-lain dan dapat diekspor menjadi banyak file gambar seperti JPG, PNG maupun gambar animasi GIF.



Gambar 2.7 Layout Adobe Photoshop CS6

2.13 Flowchart

Flowchart merupakan kumpulan dari gambar yang biasanya berupa simbol atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya.Simbol-simbol dalam flowchart dibagi menjadi 2 kelompok:

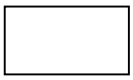
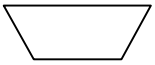
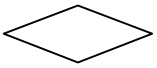
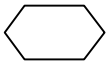
1. *Flow Direction Symbols*; dipakai untuk menggabungkan antara simbol yang satu dengan simbol yang lainnya.

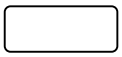
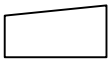
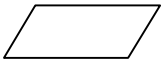
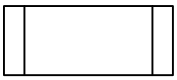
Tabel 2.1 Simbol *Flow Direction*

	***Symbol Off-line Connector*** (Simbol untuk keluar/masuk prosedure atau proses dalam lembar/halaman yang lain)
	***Symbol Connector*** (Simbol untuk keluar/masuk prosedur atau proses dalam lembar/halaman yang sama)

2. *Processing symbols*; menunjukkan jenis operasi pengolahan dalam suatu prosedur.

Tabel 2.2 Simbol *Processing*

	***Symbol Process*** (Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer)
	***Symbol Manual Operation*** (Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer)
	***Symbol Decision*** (Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi)
	***Symbol Predefined Process*** (Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di

	dalam *storage*)
	***Symbol Off-line Storage*** (Simbol yang menunjukkan bahwa data di dalam simbol ini akan disimpan)
	***Symbol Manual Input*** (Simbol untuk pemasukan data secara manual *on-line keyboard*)
	***Symbol Keying Operation*** (Simbol operasi dengan menggunakan mesin yang mempunyai *keyboard*)
	***Symbol Input Output*** (Merepresentasikan fungsi I/O yang membuat sebuah data dapat diproses (input) atau ditampilkan (output) setelah mengalami eksekusi informasi)
	***Symbol Prosedur*** (Simbol ini berperan sebagai blok pembangun dari suatu program. Prosedur memiliki suatu flowchart yang berdiri sendiri diluar flowchart utama.)