BAB II

TINJAUAN PUSTAKA

2.1. Game

Game adalah sebuah permainan interaktif yang didalamnya dilakukan konteks berpura – pura namun terlihat realitas, *game* membutuhkan alat atau komputer untuk bermain. Tujuan pembuatan *game* adalah untuk menghibur, biasanya *game* banyak diminati oleh anak – anak hingga orang dewasa. Pada dasarnya *Game* mempunyai peran penting untuk perkembangan otak manusia, diantaranya untuk meningkatkan konsentrasi dan melatih memecahkan masalah dengan tepat dan cepat, karena dalam *game* terdapat berbagai konflik atau masalah yang menuntut kita untuk menyelesaikannya dengan cepat dan tepat. Jenis – jenis *game*, antara lain:

a. Action Games

Genre ini merupakan genre *game* yang paling popular. *Game* yang membutuhkan reflex, akurasi dan kecepatan berpikir pemain. Salah satu *subgenre action* yang populer adalah *First Person Shooter (FPS).*

b. Strategy Games

Genre strategi menitik beratkan kepada kemampuan pemain untuk berfikir cepat dalam mengambil tindakan dan diperlukannya strategi sebelum permainan dimulai. *Game* strategi dibedakan menjadi menjadi dua, yaitu *Turn Based Strategy* dan *Real Time Strategy*. *Turn Based Strategy* dilakukan dengan cara *turn based* yang berarti pemain akan bergantian menjalankan setiap taktiknya, selagi pemain melakukan

langkah, pihak lawan akan menunggu. Sedangkan *real time* mengharuskan pemain membuat keputusan secara bersamaan.

c. Role-playing Games (RPG)

Dalam genre RPG pemain dapat memilih satu karakter untuk dimainkan. Dalam *game* ini pemain dapat meningkatkan *level game*, seiring dengan bertambahnya *level*, karakter dapat berubah, bertambah nya senjata dan juga bertambah hewan peliharaannya.

d. Sport

Genre *sport* atau olahraga ini biasanya dibuat mirip dengan kondisi olahraga yang sebenarnya.

e. Puzzle

Genre *puzzle* merupakan game yang menyajikan teka-teki yang membutuhkan daya pikir, seperti menyamakan gambar, perhitungan angka, menyusun balok dan menyelesaikan suatu permasalahan yang diberikan.

1.2. Game Edukasi

Menurut Wandah Wibawanto, *Game* sebagai media pembelajaran bukanlah sebuah konsep baru, konsep ini muncul seiring dengan munculnya berbagai macam *game* (permainan), baik itu *game* tradisional maupun *game* digital. Akibatnya secara spesifik muncul istilah *game* edukasi, yaitu *game* yang secara khusus memiliki muatan pembelajaran dan ditujukan untuk meningkatkan kemampuan pemainnya dalam mempelajari suatu materi. *Game* mampu memancing minat belajar pemainnya, sehingga menghasilkan pengalaman baru

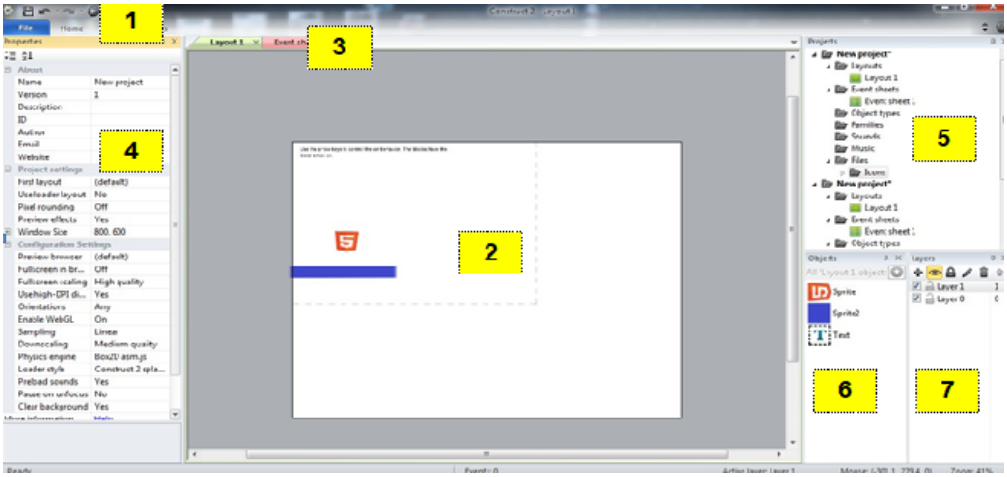
seperti perasaan senang yang pada akhirnya materi yang ingin disampaikan dapat diterima dengan mudah oleh pemain *game*.[2]

Game edukasi adalah *game* yang memiliki konten pendidikan. *Game* berjenis edukasi bertujuan untuk memancing minat terhadap pelajaran sambil bermain *game,* sehingga diharapkan dapat lebih memahami materi.

1.3. Construct 2

Construct 2 merupakan sebuah *game engine* berbasis HTML 5 buatan Scirra yang berasal dari London, Inggris. Dengan menggunakan *game engine* ini, programmer pemula maupun *expert* dapat dengan mudah membuat sebuah *game*. Construct 2 dikhususkan pada *game* berbasis 2(dua) dimensi yang menyediakan banyak fitur untuk mempercantik *game* yang ingin dirancang. Dalam Construct 2 tersedia 70 visual *effect* yang menggunakan *engine webGL,* serta dilengkapi dengan 20 *built-in plugin* dan *behavior*. Melalui Construct 2, programmer dapat mem-*publish game*nya melalui beberapa *platform* seperti (HTML 5, Google Chrome Webstore, Facebook, Phonegap, Windows Phone, Android, IOS, Tizen) dan sebagainya.[3]

Construct 2 adalah salah satu *tools* yang dapat digunakan untuk membuat *game* tanpa harus menulis kode pemrograman, karena sebagian besar logika untuk *game* dapat dibuat menggunakan menu. Kelebihan lain dari construct adalah fungsi-fungsi bawaan yang sudah disediakan dapat mempercepat proses pembuatan *game,* sehingga tidak perlu membuat ulang fungsi-fungsi tersebut untuk *game* yang akan di bangun.[3]



Gambar 2.1. Interface Construct 2

a. Menu Bar & Ribbon Tabs

Gambar 2.2. Quick Access Toolbar

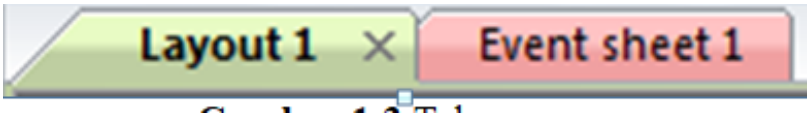
Tampilan menu dalam Construct 2 memakai bentuk *ribbon*. Tombol berbentuk roda gigi lambang Scirra akan menampilkan *drop-down ribbon*, yang di dalamnya berisi perintah-perintah standar seperti *New*, *Open*, dan *Save* (bentuknya berbeda-beda, tergantung pada versi *engine*). Di sebelahnya, terdapat *quick access tollbar* yang berisi empat buah perintah yang paling sering digunakan.

b. Layout

Laytou adalah tempat bekerja. Anda bisa menempatkan objek, mendesain *level*, dan sebagainya. Tampilan *layout* dibagi menjadi dua, yaitu *layout* dan *windows size*. Layout adalah seluruh lembar kerja berwarna putih, sedangkan *windows size* mempresentasikan ukuran layar yang dipakai. Batas antara *windows* dan *layout* ditandai dengan adanya garis putus-putus. Ukuran *layout* dan *windows*

size dapat diatur sesuai keinginan dengan melakukan konfigurasi pada ***properties.***

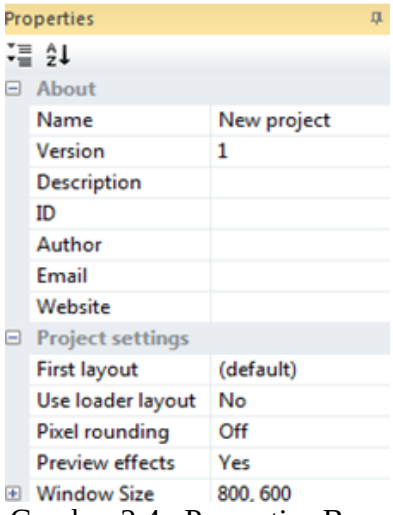
c. Tab



Gambar 2.3 . Tabs

Tab berfungsi untuk mengganti *layout* maupun *event sheet* yang ingin dikerjakan. Anda dapat melakukan *drag* untuk mengatur urutannya. *Layout* yang aktif ditandai dengan munculnya *icon close* di bagian pojok kanannya. Untuk menutup *tab*, tinggal klik tanda *close* tersebut, sedangkan untuk membukanya kembali dapat dilakukan dengan mengaksesnya di *folder* ***Layouts*** dan ***Event sheets*** dalam ***Project bar***.

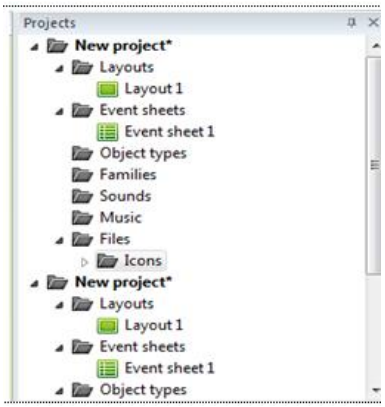
d. Properties Bar



Gambar 2.4. Properties Bar

Properties Bar berisi daftar pengaturan objek yang dapat anda ubah sesuai kebutuhan. Isi dari *properties bar* dapat berbeda-beda, tergantung pada objek apa yang dipilih. Opsi pengaturan ditujukkan dalam kolom kiri, sedangkan di kolom kanan dapat diisikan nilai sesuai yang diinginkan.

e. Project Bar



Gambar 2.5. Project Bar

Project Bar memberi gambaran umum tentang segala hal dalam proyek *game* yang dibuat, seperti jumlah *event*, *layout*, dan objek-objek yang dimiliki. Penambahan *event sheet* dapat dilakukan dengan cara **klik kanan pada *Event sheets* > *Add event sheet.*** Ketika project bar dipilih, maka *project properties* akan muncul pada *properties bar*. Berikut pengaturan yang sering digunakan :

1). First layout : *layout* yang pertama kali akan dibuka ketika *game* dimainkan.

2). Windows size : ukuran layar.

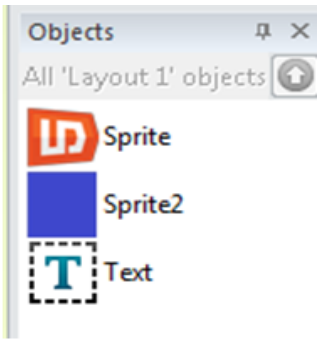
3). Preview browser : memilih *browser* mana yang akan dipakai untuk melakukan *playtest*.,

4). Fullscreen in browser : teknik-teknik penampilan *layout* dalam *game*.

5). Loader style : mengganti logo saat proses *loading.*

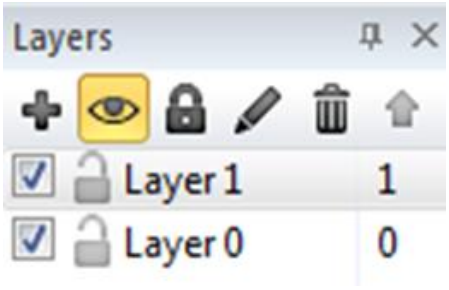
f. Object Bar

Object Bar berfungsi menunjukan objek secara spesifik, berdasarkan isi suatu *folder* dalam *project bar*. *Drag and drop* dapat dilakukan jika ingin memasukan objek ke dalam *layout.*



Gambar 2.6. Object Bar

g. Layers Bar



Gambar 2.7. Layers Bar

Layers bar digunakan untuk menambah, mengedit, maupun menghapus suatu *layer* dalam *layout. Layer* berperan besar untuk menciptakan kedalaman di *game* yang dibuat. Makin besar nilai *layer*, makin besar pula prioritas objek dalam *layout* tersebut. Misalnya jika kita menaruh objek dalam *layer* UI, objek tersebut akan menutupi objek lainnya

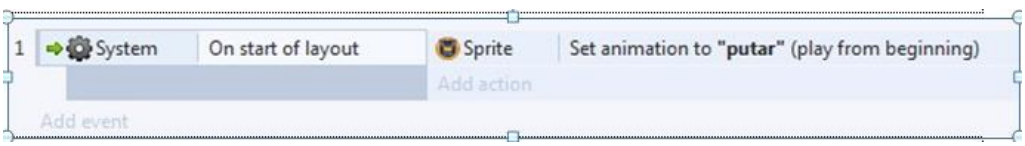
dalam *layout* yang lebih kecil nilainya.

Dalam menggunakan *layer bar*, perlu memperhatikan kegunaan tiap-tiap fungsi *icon*-nya. *Icon* berbentuk plus untuk menambah *layer* dan *trash* berfungsi untuk menghapus *layer*. Kemudian *icon* berbentuk mata berguna untuk men-*disable layout* hingga menjadi tak terlihat. Sedangkan *icon* bergambar gembok untuk mengunci objek- objek didalam *layer* untuk mencegah perubahan-perubahan yang tidak disengaja.

h. Events

Construct dapat mendefinisikan cara kerja *game* dengan memakai sistem blok logika, Sehingga tidak memerlukan pengetahuan *scripting* maupun *programming*. Hal inilah yang membuat *game engine* ini mudah dan cepat dikuasai banyak orang. Proses eksekusi suatu *event* menggunakan logika sebab akibat atau jika-maka. Jika suatu kondisi dipenuhi, maka suatu perintah akan dijalankan.

Sub event adalah anak dari sebuah *event* lain. Ketika kondisi dari *event* induk terpenuhi, maka *sub event* juga akan dijalankan. Akan tetapi, walaupun kondisi *sub event* terpenuhi, namun kondisi *event* induk belum terpenuhi, maka perintah *sub event* tidak akan dijalankan. Perintah yang dijalankan *sub event* tidak bersamaan dengan main *event*, namun yang main *event* dulu yang dijalankan. Setelah semua perintah main *event* dilakukan, baru perintah dalam sub *event* dieksekusi.



Gambar 2.8. Event

1). *Conditions* **:** syarat yang harus dipenuhi untuk melakukan suatu perintah.

2). *Actions* **:** kumpulan perintah yang dilakukan jika syaratnya sudah terpenuhi.

3). *Expressions* **:** berupa operasi logika maupun aritmetika. Bisa juga berisi nilai dari suatu objek atau variable.

4). *Sub events* **:** *event* yang berjalan ketika syarat dirinya dan *event* induk terpenuhi.

5). *Black sub event* **:** *sub event* yang tidak memiliki kondisi, sehingga apapun terjadi, aksi akan tetap diajalankan tanpa melihat kondisi *sub event.*

6). *Else* **:** kondisi yang bermakna "jika tidak" atau "selain itu". Misal pernyataan "jika hari minggu, maka sekolah libur", bentuk *else*-nya adalah "selain hari minggu, maka masuk sekolah".

7). *Groups* **:** berfungsi untuk mengelompokkan *event* berdasarkan perintah yang dijalankan.

8). *Comments* **:** untuk menulis catatan atau fungsi dari suatu *event* dengan menekan tombol "Q". Comment sangat penting jika

untuk mengerjakan proyek *game* yang besar.

9). *Includes* **:** suatu *event sheet* di-*include* (dimasukan) ke *event sheet* yang lain, sehingga anda dapat memakai *event sheet* yang sama di *layout* yang berbeda tanpa melakukan *copy-paste.*

10). *Toogle disabled* **:** untuk mematikan sementara suatu *event*, sehingga menjadi tidak berfungsi walaupun kondisinya terpenuhi.

11). *Invert* **:** berfungsi membalik suatu pernyataan. Misal pernyataan "jika minum makan haus" setelah *invert* maka menjadi "jika tidak minum maka haus".

12). *Make „Or" block* **:** beberapa kondisi yang berbeda dapat memiliki aksi yang sama. Untuk meletakan semua kondisi dalam satu *event*, maka digunakan *"make „Or" block"*. Contohnya, "jika player mati atau peluru habis, maka *game over".*

13). Variables **:** tempat menyimpan nilai suatu data. Sebuah variable dapat menyimpan

nilai yang berubah-ubah, atau bisa juga konstan. Artinya, ia hanya dapat diakses, namun tidak dapat diubah nilainya.

a). Global variables: global variable terletak di *event sheet* paling atas. Kata“global” berarti variable ini dapat diakses dari mana saja, termasuk dari *event sheet* berbeda.

b). Local variables: *local variable* hanya dimiliki oleh suatu *group* atau *event. Local variables* memiliki jangkauan tertentu, sehingga tidak semua *event* bisa mengaksesnya, walaupun dalam *event sheet* yang sama.

c). Text: *variable* yang menyimpan karakter huruf atau angka. Karakter berupa angka ditandai dengan adanya tanda petik (“), misalnya “123”.

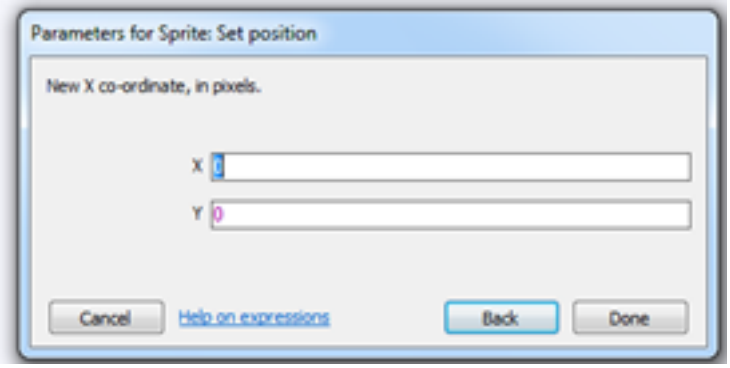
d). Number: *variable* yang menyimpan angka, misalnya 456.

e). Initial value: nilai awal dari suatu *variable*

i. Object Paramenters & Expressions

Kotak dialog parameter akan muncul ketika menambahkan suatu aksi ke dalam *event*, bisa juga saat mengedit kondisi atau aksi. Namun,

parameter hanya muncul saat menambahkan aksi tertentu saja, aksi seperti *Destroy* tidak membutuhkan parameter. Gambar dibawah ini merupakan salah satu contoh dialog penginputan parameter. Segala sesuatu yang di inputkan dalam parameter disebut **ekspresi *(expressions).***



Gambar 2.9. Pengisian parameter untuk perintah Set Position

Expressions dapat berupa angka. Teks, atau suatu rumus. Ketika memasukkan ekspresi, akan keluar ***Expresssions Panel*** yang berisikan kumpulan ekspresi yang dimiliki oleh objek-objek dalam *project*. Terdapat deskripsi singkat mengenai fungsi dari setiap ekspresi, sehingga tidak akan membingungkan untuk mengetahui fungsinya.

Contoh ekspresi:

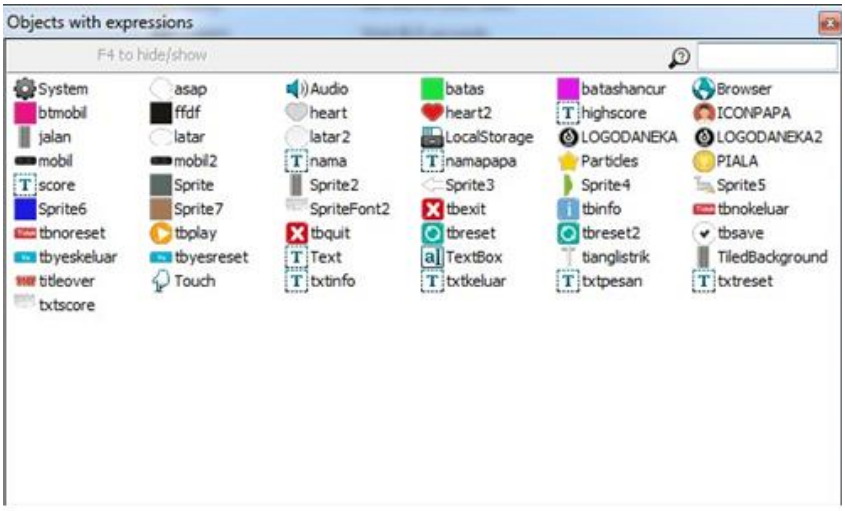
1) 15

2). *"Hello World"*

3). *Player.Health* + 50

4).*"Highscore"&highscore*

5).(*distance(monster.X,hero.X,monster.Y,hero.Y))*(monster.Speed/5)*



Gambar 2.10. Expressions panel dalam Tiang Listrik Hunter

Untuk mengambil nilai yang dimiliki oleh objek, digunakanlah *object expressions* untuk mendapatkannya. Caranya dengan mengetikkan nama_objek (dot) ekspresi, misal Mobil.*Speed.* Sebagai landasan untuk materi berikutnya, ada dua istilah penting yang perlu dipahami, yaitu objek dan *instance*. Sebuah objek dapat di ibaratkan adalah cetakan roti, sedangkan *instance* adalah roti yang dihasilkan. Satu buah cetakan dapat digunakan untuk membuat banyak roti. Begitu pula dengan objek, satu objek dapat digunakan untuk membuat banyak *instance*. Kesimpulannya, objek dan *instance* adalah dua hal yang berbeda.

Di dalam Construct 2, logika permainan diterapkan dengan blok-blok logika yang dinamakan *event*. Didalam *event*, ditaruhlah satu atau lebih kondisi, aksi hanya akan dilakukan pada *instance* yang memenuhi kondisi tersebut. Setiap *instance* berdiri sendiri, mereka tidak terhubung satu sama lain. Misalkan terdapat 5 buah *instance* monster. Ketika satu monster mati,

maka monster yang lain tidak akan ikut mati.

Instance dapat dibuat saat *runtime* melalui *event*, atau juga bisa dibuat sebelumnya dalam *layout* untuk *mendesain level*, menu, dan sebagainya. Pada saat mendesain *layout*, Construct 2 mengharuskan membuat minimal satu buah *instance* dari setiap objek. Jika tidak, maka akan memunculkan pesan *error.*

i. Behavior Reference

Behavior merupakan fitur Construct 2 yang berfungsi membuat objek-objek di dalamnya memiliki perilaku tertentu. Berikut penjelasan mengenai *behavior* dan fungsi- fungsinya:

1) 8 Direction

Membuat *object* dapat digerakan dengan *input* tertentu. Arah gerakan objek bisa diatur sedemikian rupa, mulai dari dua, empat, hingga delapan arah.

2) Anchor

Berfungsi untuk memposisikan objek secara otomatis agar sesuai dengan ukuran layar, hal ini berfungsi untuk mendukung berbagai ukuran layar.

3) Bound to Layout

Berfungsi agar obyek tidak keluar dari *layout game*, sehinga seakan-akan ada tembok yang membatasi saat objek hendak keluar *layout.*

4) Bullet

Berfungsi untuk membuat *object* maju lurus kedepan, ini biasa

digunakan untuk peluru, tetapi *bullet* juga mempunyai opsi tambahan seperti gravitasi dan memantul yang digunakan untuk membuat object seperti bola yang memantul, selain biasa digunakan untuk peluru, *bullet* juga dapat digunakan untuk *object* sebagai musuh yang selalu bergerak secara *otomatis*

5) Car

Berfungsi untuk membuat object dapat bergerak maju mundur belok kanan, kiri seperti memiliki kemudi, *car* biasanya digunakan untuk *game* yang bertema tentang kendaraan atau balapan

6) Custom movement

Membuat obyek dapat bergerak sesuai kebiasaan *(event based movement).*

7) Destroy Outside Layout

Menghancurkan obyek setelah keluar dari layar utama *game*. Jika anda melihat peluru yang menghilang setelah keluar dari layar pada *game*, itu sebenarnya tidak menghilang, peluru itu akan tetap maju secara terus menurus dan jika hal ini dibiarkan lama kelamaan akan membuat *loading game* jadi berat. Untuk menghindari hal tersebut maka gunakanlah *Destroy Outside Behavior* yang akan menghancurkan object secara otomatis setelah keluar dari layar.

8) Drag And Drop

Berfungsi untuk memberikan sifat pada object agar dapat ditarik dan diposisikan sesuai keinginan dengan mengklik atau menyentuh obyek

tersebut kemudian dapat dilepaskan jika posisi object sudah sesuai dengan yang anda inginkan dengan melepas klik atau sentuhan anda.

9) **Fade**

Memberikan sifat pada object agar dapat memudar dan menghilang secara otomatis. Contohnya : jika anda menembak musuh dan tembakan tersebut mengenai musuh, maka akan keluar api dan api tersebut akan memudar dan menghilang secara otomatis.

10) **Flash**

Membuat object dapat terlihat untuk beberapa saat lalu menghilang untuk beberapa saat kemudian muncul lagi sesuai waktu yang telah anda set dan akan terus berulang – ulang (seperti berkedip).

11) **Jump-Thru**

Untuk membuat suatu pijakan dapat dipijak dan dapat ditembus dari bawah.

12) **Solid**

Membuat suatu obyek dapat dipijak, sama seperti *jump-thru*. Namun, solid tidak dapat ditembus dari bawah.

13) **Line-of-Sight**

Berfungsi untuk membatasi jarak pandang object. Seperti pada *game* peperangan, biasanya ada *object* yang menghalangi jarak pandang *object* pemain untuk melihat musuh. Misal terhalang tembok, pohon dan lain sebagainya

14) **No Save**

Biasanya semua *object* dan tindakannnya akan disimpan dalam *game,* itu akan membuat *loading game* semakin lama semakin lambat. Dengan menggunakan *no save behavior* maka *object* yang telah dipasang *no save behavior* dan tindakan – tindakannya tidak akan disimpan dan tidak akan membuat *loading game* menjadi berat.

15) Path Finding

Berfungsi untuk membuat object sebagai pemain dapat menemukan jalan tercepat disekitar rintangan secara cepat.

16) Persist

Membuat object dapat mengingat tata letak yang berbeda pada saat ditinggalkan kemudian kembali lagi ke tempat tersebut. Object yang menggunakan persist *behaviour* disebut juga sebagai tata letak terus menerus. Ibaratnya, disaat anda telah menghancurkan dinding kemudian meninggalkannya, maka saat anda kembali lagi ke tempat tersebut kondisinya sama seperti saat anda tinggalkan (dindingnya tetap hancur)

17) Physics

Untuk contoh penggunaan *physics behavior*, anda lihat saja pada game *Angry Bird* dimana reruntuhan gedung berjatuhan kebawah dan jika salah satu object pada gedung yang roboh tersebut menyentuh object lain (gedung lain) maka object yang tersentuh akan ikut bergoyang atau bahkan ikut roboh.

18) Pin

Object yang diberi *Pin Behavior* akan memberikan kesan bahwa

object tersebut telah disematkan atau menempel pada obyek lain.

19) **Platform**

Obyek yang diberi *Platform Behavior* berfungsi sebagai Pemain dalam *game* tersebut yang dapat digerakkan sesuai keinginan anda.

20) **Rotate Behavior**

Berfungsi agar *game* seolah-olah berputar.

21) **Shadow Caster**

Memberikan efek *shadow* (bayangan) pada object yang diberi *Shadow Caster Behavior.*

22) **Sine**

Dapat menyesuaikan object (seperti posisi, ukuran atau sudut). Seperti mebuat rumput bergoyang secara teratur dan terus menerus. Ini akan mempercantik tampilan game anda.

23) **Timer**

Berfungsi untuk memberikan batas waktu untuk pemain menyelesaikan permainan. *Time Behavior* digunakan hampir disetiap *game.*

24) **Turret**

Apakah anda pernah memainkan *game contra*? Jika pernah pasti anda melihat didalam game contra ada Tank yang dapat dinaiki dan mengikuti arah gerakan si object pemain. Nah, itulah fungsi dari *Turret Behavior*

25) **Wrap**

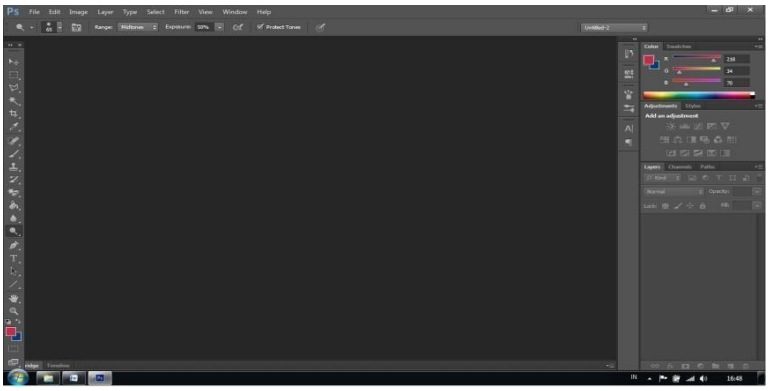
Ini berfungsi untuk me-*repositions object*. Misal pada permainan *Snake II* milik nokia, jika anda mengarahkan ularnya kebawah, maka setelah melewati batas ular tersebut akan muncul dari atas. Seperti itulah fungsi *wrap.*

26) Scroll To

Ini berfungsi untuk membuat kamera/*view,* seolah-olah mengikuti objek kemana pun berada.[3]

1.4. Adobe Photo shop

Menurut Asep Effendhy (2013:1) *Adobe Photoshop,* atau biasa disebut *photoshop* adalah perangkat lunak editor citra buatan *Adobe System* yang dikhususkan unutk pengeditan foto atau gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar *(market leader)* untuk perangkat lunak pengolah gambar atau foto, dan bersama *Adobe Acrobat* dianggap sebagai produk terbaik yang pernah diproduksi oleh *Adobe System.*[2]



Gambar 2.11. Tampilan Awal halaman *Adobe Photoshop CS6*

Bagian-bagian *Photoshop CS6* yaitu:

a. *TitleBar.*

Title bar merupakan sebuah tampilan jendela yang ada pada photoshop berfungsi untuk menampilkan judul nama *file* yang sedang dikerjakan. Selain itu *title bar* juga dapat digunakan untuk memindahkan posisi jendela melalui proses *drag* and *drop* serta mengatur ukuran jendela dari ukuran *maximize* ke ukuran *restore* atau sebaliknya.

Gambar 2.12. *Title Bar* pada Dokumen.

b. *MenuBar.*

Menu bar merupakan barisan menu yang disediakan oleh *adobe photoshop* berisi perintah-perintah yang dikelompokkan sedemikian rupa untuk memudahkan pekerjaan anda. Saat di posisi *default, menu bar* terdiri dari perintah *file, edit, image, select, filter, view, windows,* dan*help*.

Gambar 2.13. Tampilan *Menu Bar*.

c. *Option Bar.*

Merupakan barisan perintah yang dapat anda jalankan ketika sedang memakai salah satu *tool*. Tentunya setiap perintah pada *option bar* berbeda-beda tergantung tombol *tool bar* mana

yang anda pilih.

Gambar 2.14. Tampilan *Option bar* dengan *Move tool.*

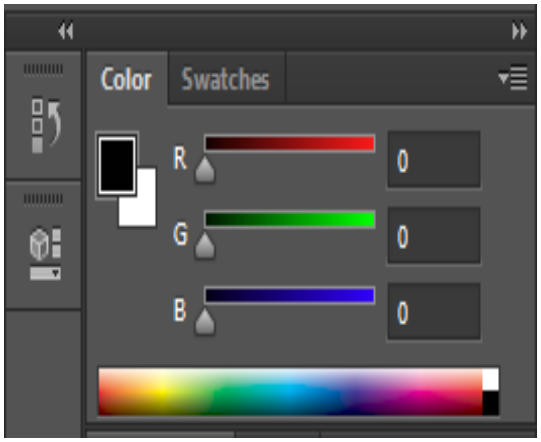
d. ToolBox.

Tool box berisikan seperangkat fasilitas untuk mengedit dan memanipulasi gambar. Tampilan *ToolBox* bersifat mengambang sehingga, *user* dapat memindahkan posisi sesuai keinginan. Tampilan *tool box* terlihat pada gambar berikut:

Gambar 2.15. Tampilan *ToolBox*.

e. Panel Color.

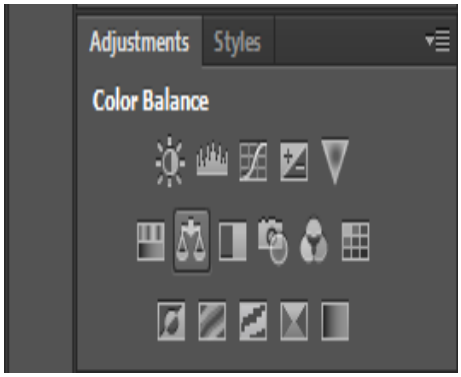
Panel Color berfungsi untuk memilih warna pada *Foreground* dan *Background* dengan cara menggeser warna hingga memperoleh kombinasi warna yang tepat.



Gambar 2.16. Tampilan *Panel Color*.

f. Panel Adjustments.

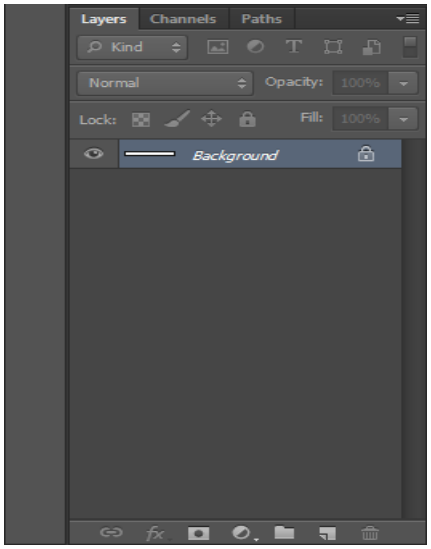
Panel Adjustments merupakan panel yang berisi perintah - perintah yang berfungsi untuk mengatur pencahayan objek, seperti mengatur gelap terang objek, mengatur kontras objek, mengatur warna gambar dan lain sebagainya.



Gambar 2.17. Tampilan *Panel Adjustments.*

g. DockPanel.

Dock Panel merupakan bagian yang menampung panel - panel yang tersedia pada *photo shop* sedangkan fungsi dari panel sendiri adalah untuk mengelola dan memanipulasi objek lanjut secara lebih *detail* dan *kompleks.*



Gambar 2.18. Tampilan *Dock Panel.*

1.5. Android

a. Pengertian *Android*

Menurut Hilmi Masruri, *Android* merupakan sistem operasi *open source* yang dimana semua orang bisa mengembangkan *android*, hal itulah yang membuat perkembangan aplikasi *android* semakin cepat dan bertumbuh kembang. *Android* awalnya dikembangkan oleh *Android*, Inc dengan dukungan finansial Google, yang kemudian membelinya pada tahun 2005.

b. Sejarah dan Perkembangan *Android*

Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk

menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc. membeli Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance, konsorsium* dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T Mobile, dan Nvidia.* Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode – kode *Android* di bawah lisensi *Apache,* sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Berikut ini perkembangan *Android* dari versi pertama kali dirilis sampai versi yang ada saat ini :

1) Astro 1.0 (Alpha)

   Versi pertama android dirilis pada 23 September 2008, awalnya versi ini dinamai Astro namun karena hak cipta penamaan, pihak android tidak menggunakan nama ini secara komersil. Versi android 1.0 sempat disematkan pada ponsel jenis HTC *dream.*

2) Bender 1.1 (Beta)

   Versi Bender 1.1 yang dirilis 09 Februari 2009 memiliki masalah sama seperti versi 1.0, yakni hak penamaan merk. Pada

versi 1.0 dan 1.1 ini *Google Play Store* yang sekarang kita kenal masih diluncurkan dengan nama *Android Market*.

3) Cupcake 1.5

Versi ketiga *android* dirilis pada 27 April 2009. Pada versi ii, barulah secara komersil *android* muncul dengan nama makanan penutup, nama *Cupcake* dipilih menjadi nama versi *android* ini. Fitur baru yang muncul pada versi ini salah satunya andalah *on-screen keyboard.*

4) Donut 1.6

Versi yang dirilis pada 15 September 2009 ini memiliki peningkatan pada fitur pencarian dan UI yang lebih *user friendly*. Donut 1.6 sudah mendukung teknologi CDMA/EVDO, 802.1x, dan VPNs. Pada *update* versi kali ini, android berfokus pda penambahan penggunaan jaringan dan layar.

5) Éclair 2.0 – 2.1

Éclair 2.0 – 2.1 dirilis pada 26 Oktober 2009. Dari versi inilah sampai sekarang kita mengenal fitur *navigasi* di *Google Maps*, yang pada akhirnya menggantikan fungsi peta konvensional dan sangat membantu mobilitas masyarakat.

6) Froyo 2.2

Froyo atau singkatan dari *Frozen Yoghurt* merupakan versi android yang dirilis pada 20 Mei 2010. Salah satu fitur yang muncul pada versi in adalah kunci pin pada ponsel dan pemolesan system- system yang sudah ada sebelumnya.

7) Gingerbread 2.3

Versi ini dirilis pada 06 desember 2010. Pada versi ini pembaharuan leih banyakk dari sisi hiburan, mulai dari dukungan format video dan yang paling fenomenal adalah dukungan kamera depan pada ponsel yang membawa tren photo *selfie*.

8) Honeycomb 3.0 / 3.1

Versi Honeycomb dirilis pada 22 Februari 2011. Pada versi ini diluncurkan untuk penggunaan OS android pada tablet. Versi ini mendukung *multi-processor* dan akselerasi *hardware* untuk gratis secara *virtual buttons*. Merk tablet pertama yang menggunakan ini adalah Motorola Xoom.

9) Ice Cream Sandwich 4.0

Ice Cream Sandwich 4.0 dirilis pada 19 Oktober 2011. Fitur yang ada pada versi tablet dimasukan kedalam ICS 4.0 ini, termasuk juga dengan penambahan fitur baru seperti *face unlock*, aplikasi email dan rekap penggunaan data internet.

10) Jelly Bean 4.1 / 4.2 / 4.3

Pada tahun 2012, *android* mengeluarkan versi Jellly Bean. Lewat versi Jelly Bean (4.0) Google Now mulai diperkenalkan yang berfungsi untuk *voice assistant* untuk berbagai keperluan secara cepat. Pada versi 4.2 terdapat fitur photo sphere untuk panorama, *daydream* sebagai *screensaver*, *power control*, dan sebagainya. Sedangkan pada versi 4.3 adalah pemutakhiran versi sebelumnya.

11) KitKat 4.4

Versi KitKat dirilis pada 31 Oktober 2013. Pada versi yang sebelumnya bernama Key Lime Pie ini membawa peningkatan signifikan dalam hal *user experience*. Versi KitKat optimal berjalan pada kapasitaas yang lebih besar dari versi *android* sebelumnya. Disarankan perangkat memiliki minimal RAM 512 MB.

12) Lollipop 5.0

Versi Lollipop dirilis pada 12 November 2014 dan tersedia resmi *over the air (*OTA). Perubahan yang paling terlihat dalam versi ini adalah *user interface* yang didesain ulang dan dibangun dengan *material design*.

13) Marshmallow 6.0

Versi Marsmallow dirilis pada 05 Oktober 2015 dengan memperkenalkan beberapa fitur canggih, diantaranya adalah *search bar*, perizinan aplikasi dan juga sensor sidik jari atau *finger screen*.

14) Nougat 7.0

Versi Nougat menampillkan perubahan besar untuk *android*, terlihat dari fitur – fitur terbaru diantaranya adalah *multi-window* yang memungkinkan 2 aplikasi secara bersamaan, selain itu pada versi ini dirilis juga 63 *emoticon* baru.

15) Oreo 8.0

Oreo 8.0 dirilis pada 21 Agustus 2017 dengan menambah lebih banyak fitur *multi-tasking* dan perubahan pada notifikasi. Pengguna bias mengatur mana saja notifikasi yang ingin ditampilkan. Tampilan UI-nya juga lebih rapi dan segar, serta difokuskan untuk memudahkan pengguna mengakses aplikasi dan mencari informasi.

16) Pie 9.0

Android versi 9.0 yang dinamai Pie dirilis pada 06 Agustus 2018. Pada versi ini penambahan fitur – fitur baru dari versi sebelumnya diantaranya adalah *smart reply* dari *notifikasi,* navigasi berbasis *gesture*, *adaptive battery*, *digital wellbeing* dan sebagainya.

17) Android 10

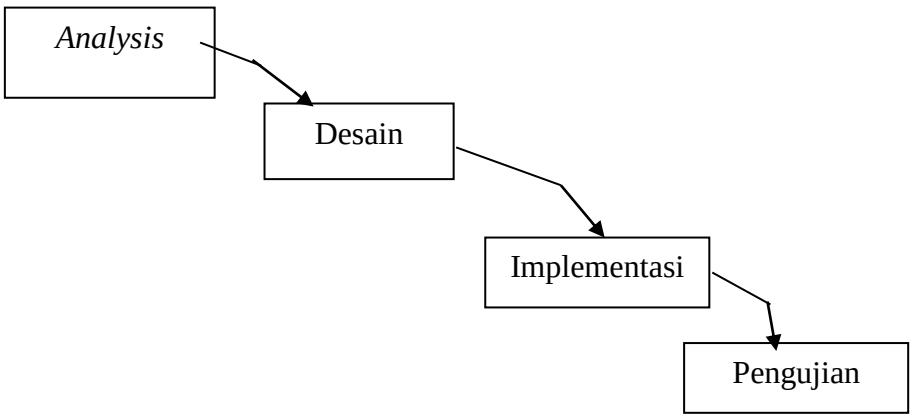
Versi *Android* 10 dirilis pada September 2019 ini tidak lagi menggunakan nama *dessert* atau makanan penutup seperti versi sebelumnya. Salah satu alasannya adalah karena pihak *android* tidak menemukan makanan yang berasal dari awalan huruf Q.

18) Android 11

Android 11 dirilis pada Juni 2020 dan diberi kode nama Red Velvet Cake. Pada versi ini terdapat fitur – fitur baru diantaranya bubbles, conversation notifications, voice acces, media control, device control, one time permission, background location, permission auto-reset, scoped storage dan sejumlah pemolesan pada sisi desain**.**

2.6 Metode Analisys

Waterfall adalah metodologi umum yang digunakan untuk mengembangkan sistem di berbagai organisasi. *Waterfall* terdiri dari beberapa fase yang dapat memantau proses dari sebuah sistem dalam analisis dan perancangannya.



Gambar 2.19. Systems waterfall

1. *Analysis*

Adalah fase *analysis* atau fase analisis. Pada fase analisis akan dilakukan analisis secara menyeluruh pada objek yang menjadi kajian berdasarkan fase sebelumnya. Fase analisis akan menentukan kebutuhan dari suatu sistem berdasarkan analisis yang dilakukan pada suatu objek. Dari hasil analisis tersebut akan menghasilkan alternatif dari beberapa sistem yang

akan dikembangkan serta kebutuhan dari *hardware* dan *software* yang dibutuhkan pada sistem.

2. Desain

Adalah *fase design* atau *fase* perancangan. Pada *fase* perancangan akan dilakukan perancangan dari berbagai komponen pada sistem seperti *input* dan *output,* database dan proses berdasarkan analisis pada *fase* sebelumnya. 22 *Fase* perancangan terbagi menjadi perancangan *logical* dan perancangan *physical.* Perancangan *logical* berfokus pada perancangan fungsi-fungsi pada sistem dan perancangan *physical* berfokus pada pengubahan fungsi dan spesifikasi menjadi suatu sistem yang berjalan.

3. Implementation

Adalah *fase implementation* atau fase penerapan. Pada fase penerapan akan membuat suatu sistem yang berjalan berdasarkan analisis dan perancangan yang telah dilakukan pada fase sebelumnya. Fase ini akan membuat sistem berdasarkan bahasa pemrograman yang digunakan dengan pengkodean serta sistem yang telah jadi akan diuji, dipasang dan diterapkan pada suatu organisasi.

4.pengujian

Tahap pengujian Untuk Memerikan dan memastikan bahwa sistem dapat digunakan secara handal dan akurat,serta meminimalkan resiko terjadinya kesalahan atau pada sistem. Jika ditemukan masalah,sistem akan diperbaiki dan diuji kembali.

2.7 Penelitian terdahulu

1. Penelitian terdahulu yang dilakukan oleh Dian Wahyu Putra, A. Prasita Nugroho, Erri Wahyu Puspitarini yang berjudul *"Game* Edukasi Berbasis *Android* Sebagai Media Pembelajaran Untuk Anak Usia Dini". *Game* edukasi ini merupakan aplikasi pembelajaran untuk anak usia dini dimulai dari usia 3 sampai 6 tahun yang berisi tentang materi pelajaran mengenal binatang, mewarnai, coret-coret, menyanyi serta alfabet. Metode penelitian dan pengembangan aplikasi edukasi ini adalah metode *waterfall* yang terdiri dari lima tahapan yaitu *(Requirement, Design, Implementation, Verivication, Maintenance.)* Dengan menerapakan hasil dari *game* edukasi ini, diharapkan dapat membantu anak-anak dalam belajar dan meningkatkan pola pikir kreatif serta menambah pengetahuan lebih maju.[4]
2. Penelitian terdahulu yang dilakukan oleh Sanriomi Sintaro, Rahmat Ramdani, Slamet Samsugi yang berjudul " Rancang Bangun *Game* Edukasi Tempat Bersejarah Di Indonesia". *Game* ini dikembangan dengan metode *Game Development Life Cycle*. Kemudian untuk mengetahui kelayakan *game* ini di uji menggunakan ISO9126 dengan empat aspek pengujian yaitu *Functionality, Usability, Efficiency* dan *Portability.* Berdasarkan Evaluasi hasil *observasi* yang dilakukan terhadap siswa kelas 5 dan 6 MIN 7 Bandar Lampung terjadi peningkatan kemampuan siswa dalam menjawab soal tentang tempat bersejarah di Indonesia khususnya yang terdapat di Pulau Sumatera dan Pulau Jawa yang terdapat di dalam *game* sebesar 50% dari kondisi

awal dimana siswa belum memainkan *game* edukasi tempat bersejarah di Indonesia.[5]

3. Penelitian terdahulu yang dilakukan oleh Diana Laily Fithri, dan Dave Andre Setiawan yang berjudul " Analisa Perancangan *Game* Edukasi Sebagai Motivasi Belajar Anak Usia Dini". Analisa dan perancangan *game* edukasi untuk anak usia dini yang terdiri dari pengenalan angka dan huruf merupakan pengembangan dan pengenalan huruf dan angka anak usia dini. Percancangan sistem dilakukan dengan metode waterfall dan bahasa permodelan UML *(United Modeling Language).*. Implementasi penelitian ini menghasilkan sebuah aplikasi *Game* Edukasi Berbasis *Android* yang sangat bermanfaat pada kalangan anak usia dini, disamping menempuh pembelajaran formal di sekolah, anak-anak dapat belajar sambil bermain dengan menggunakan aplikasi ini.[6]

Perbedaan penelitian ini dengan penelitian sebelumnya penelitian ini menggunakan construct 2 untuk mengajar *game* pintar menulis angka agar memudahkan anak usia dini untuk menulis dan mengenali angka.