

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Animasi

Menurut Munir (2013:317) "Animasi adalah rangkaian gambar yang disusun berurutan atau dikenal dengan istilah *frame*, satu *frame* terdiri dari satu gambar. Jika susunan gambar tersebut ditampilkan bergantian dengan waktu tertentu maka akan terlihat bergerak".

Dari pengertian diatas menyimpulkan bahwa animasi merupakan gambar yang terdiri dari beberapa didalam nya jika ditampilkan maka akan terlihat bergerak.

2.1.1. Pengertian *Game*

Menurut Novaliendry (2013: 111) "*Game* berasal dari kata bahasa Inggris yang memiliki arti dasar permainan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual (*intellectualplayability*)". *Game* juga bisa diartikan sebagai arena keputusan dan aksi pemainnya, ada target-target yang ingin dicapai pemainnya.

Menurut Jasson (2009:2) "*Game* adalah suatu *system* atau program dimana satu atau lebih pemain mengambil keputusan melalui kendali pada objek didalam *game* untuk suatu tujuan tertentu".

Pengertian *game* di atas menerangkan bahwa *game* atau permainan adalah suatu program mengambil keputusan untuk suatu tujuan tertentu melalui objek di

dalamnya dengan tujuan sebagai kegiatan yang mendidik maupun bersenang-senang untuk mengisi waktu luang.

2.1.2. Jenis-jenis *Game*

Jenis *game* yang lebih dikenal dengan *genre*. *Genre* juga berarti format atau gaya dari sebuah *game*. Format sebuah *game* bisa murni sebuah *genre* atau bisa merupakan campuran (*hybrid*) dari beberapa *genre* lain. (Jasson, 2009:6-14).

Dalam penjelasan tersebut dapat dipahami bahwa jenis *game* merupakan format atau gaya dari permainan itu sendiri.

Berikut ini akan dijelaskan macam-macam jenis *game* tersebut (Jasson, 2009:6-14).

1. *Quiz Game*

Jenis permainan ini adalah dimana pemain berusaha untuk menjawab pertanyaan dengan benar.

2. *Puzzle Game*

Game jenis ini memberikan tantangan pada pemainnya dengan cara menjatuhkan sesuatu dari sisi sebelah atas kebawah. Semakin lama akan semakin cepat dan semakin banyak objek yang jatuh.

3. *Shooting Game*

Jenis *game* ini adalah tembak-menembak merupakan tema utamanya, karena tujuan dalam *game* jenis ini hanya untuk membunuh lawan ataupun musuh dengan senjata yang telah disediakan.

4. *Adventure Game* (Petualangan)

Game jenis ini banyak menghadirkan aksi dalam petualangan seperti menjelajah tempat-tempat baru berupa pulau, lautan atau hutan.

5. *Fighting Game*

Fighting Game adalah *game* pertarungan dan perkelahian antara dua karakter dan biasanya bisa dimainkan multiplayer. Biasanya terdapat kombinasi gerakan dan jurus dari tombol-tombol controller yang digunakan untuk mengalahkan musuh.

6. *Sport Game*

Sports Game adalah permainan yang bertema olahraga. Antara lain adalah sepakbola, bola basket, bola voli, tennis, tinju, gulat dan lain-lain.

7. *Racing Game*

Racing Game adalah *game* yang bersifat balapan untuk mencapai garis *finish*.

2.1.3. Elemen-Elemen Dasar *Game*

Menurut Jasson (2009:20-29) “Fungsi tampilan dari berbagai *game* ada yang serupa atau mirip satu sama lain, baik fungsi maupun bentuknya. Walaupun *genre* satu sama lain berbeda, namun elemen itu tetap memiliki fungsi dan maksud yang sama”.

Berikut adalah keterangan berbagai elemen-elemen yang ada dalam sebuah *game* yang terdiri atas:

1. *Title* (Judul)

Judul adalah elemen penting dalam pengembangan sebuah *game*. Membuat judul sebuah *game* tidaklah mudah. Judul harus singkat namun menyiratkan isi dari maksud *game* tersebut.

2. *Title Screen* (Layar Judul)

Judul memang merupakan hal yang penting, namun untuk membuat judul lebih menarik adalah tampilan grafis yang sesuai dengan judul *game* tersebut. Pada *game* masa kini, sebelum *title screen* muncul terkadang sudah ada beberapa animasi pembuka yang muncul.

3. *Credits* (Daftar Nama Individu yang Terlibat)

Maksud dari *credits* adalah menunjukkan bahwa yang ikut serta dalam membuat *game* tersebut dengan penghargaan kepada pihak yang membantu dalam pengembangan *game* tersebut. *Game* masa kini berisikan banyak *credits*, individu yang terlibat dimulai dari *designer*, *programmer*, *sound engineer*, *two dimention and three dimention artist* dan sebagainya. Terdapat juga kata ucapan terima kasih yang disampaikan pada pihak-pihak tertentu yang dianggap telah membantu dan berjasa kepada pengembang *game* tersebut.

4. *Cutscene/Intro* (Pengenalan Cerita)

Tujuan dari *cutscene/intro* adalah untuk membuka awal pengenalan sebelum pemain memulai *game*, suatu penting di dalam alur cerita *game* maupun suatu transisi antara tingkat satu dengan tingkat selanjutnya. Adanya *cutscene* atau *intro* pada *game* ini, membuat pemain secara perlahan akan masuk ke alur permainan atau kepada bagian selanjutnya dari *game*.

5. *Control Panel* (Panel Kendali)

Tidak kalah pentingnya adalah *control* panel di mana pemain dapat mengatur permainan. Contoh memulai *game* baru, menghentikan sementara *game*, mengulangi *game*, menyimpan *game*, dan sebagainya.

6. *User Interface* (Antarmuka)

Berbeda dengan *control panel* yang hanya aktif ketika dibutuhkan. Sementara *user interface* tetap aktif selama pemain memainkan *game* dan tidak hanya tampilan pada layar, *keyboard* dan *mouse* yang digunakan adalah *user interface*. Untuk layar maka *user interface* adalah simbol atau tampilan khusus yang membantu pemain memainkan *game*. Contoh pada *game* jenis *First Person Shooter* (FPS) *user interface*-nya adalah jumlah peluru dan nyawa yang ada pada tampilan layar.

7. *Help* (Bantuan)

Tidak semua *game* dapat dimengerti dengan cepat oleh pemain *game* tersebut. Untuk itulah elemen *help* ini menjadi salah elemen yang cukup penting. Di dalam *help* ini semua instruksi yang bisa membantu pemakai ditampilkan dengan cepat dan sederhana sehingga mudah dipahami dan bisa langsung digunakan.

8. *Mouse Pointer* (Ikon Penunjuk *Mouse*)

Bentuk *pointer mouse* bisa bervariasi sesuai dengan tema dari *game* yang dibuat. Terkadang bentuk *pointer* bisa berubah jika menyentuh area tertentu di dalam *game* (disebut *hotspot*).

9. *Music and Sound* (Musik dan Efek Suara)

Semua jenis *game* saat ini sewajarnya dilengkapi musik dan suara yang cocok dengan tema dari *game* yang akan dikembangkan. Namun jika *game* yang dikembangkan tidak memiliki suara dan musik akan terkesan tidak *profesional* dan kurang bermutu.

10. *Art* (Seni)

Pada dasarnya *game* yang baik dihasilkan dari gambar yang memang memiliki seni yang baik pula. Gambar juga memegang peranan penting dalam elemen *game* yang akan dikembangkan. Itu sebabnya kenapa *game* masa kini khususnya di bidang *art* (seni) dipegang oleh beberapa orang yang memiliki keahlian tersendiri.

11. *Storyline* (Cerita)

Alur cerita tidak digunakan pada semua *game*, hanya jenis *game* tertentu yang menggunakan alur cerita misalnya *adventure game*, *RPG game*, *Real Time Strategi (RTS) game* dan *Action game*.

12. *Playability* (Kemampuan untuk Dimainkan)

Game yang dikembangkan harus mampu dimainkan oleh seorang pemain pemula, dalam arti cukup mudah dan cepat dimengerti bagaimana cerita dan *game* itu berjalan. Terkadang sebagian *game* yang sukses dipasaran lebih berdasarkan faktor ini dibandingkan faktor lain seperti grafis yang hebat.

13. Level (Tingkatan)

Hampir setiap *game* memiliki beberapa level untuk dimainkan. Maksudnya supaya pada tahap awal pemain bisa menguasai permainan dengan baik. Pada tingkat selanjutnya pemain tidak akan merasa bosan karena tantangan akan meningkat.

14. Demo

Pada dasarnya *game* muncul dengan mode demo terlebih dahulu sehingga bisa dimainkan sebelum *software* aslinya muncul di pasaran. Memang tidak

semua *game* menyediakan demo *game*, tetapi keuntungan promosi bisa didapatkan dengan menyediakan demo *game*.

15. *Invincible Mode* (Tak Terkalahkan)

Mode tak terkalahkan ini bertujuan untuk pemeriksaan dari *game* itu sendiri dimana beberapa pihak diizinkan untuk menjalankan *game* dalam bentuk *invincible mode* dimana karakter *game* pemain tidak bisa kalah atau mati.

16. *Exit Screen* (Layar Keluar)

Layar keluar ini merupakan elemen penting dalam pengembangan *game*. *Game* yang baik biasanya memiliki *exit screen* pada *game* yang dikembangkan.

17. *Documentations* (Dokumentasi)

Game yang baik tidak dianggap selesai tanpa adanya dokumentasi yang cukup. Dokumentasi ini ditujukan kepada para pemain. Pada dokumentasi biasanya berisikan cara memainkan *game*, menangani masalah yang timbul dari segi *software* maupun *hardware*.

18. *Copyright* (Hak Cipta)

Hal paling penting dalam pengembangan sebuah perangkat lunak *game* ini adalah hak cipta yang diberikan oleh pembuat *game*. Hal ini untuk mencegah masalah yang berkaitan dengan hak cipta dimasa depan.

19. *Setup* (Pemasangan atau Instal)

Sudah menjadi kewajiban bagi pembuat perangkat lunak untuk membuat *game* dalam bentuk *setup*. Program *setup* pada *game* akan dikompilasi terlebih dahulu menjadi paket yang padat.

2.1.4. Manfaat Bermain *Game*

Menurut Harsono (2014:76) “Seseorang yang bermain *game* akan lebih fokus terhadap apa yang terjadi di sekitarnya dan kemampuan visualnya lebih meningkat bila dibandingkan dengan orang yang tidak bermain *game*”.

Selain itu menurut Harsono (2014:78) “*Game* dapat digunakan sebagai pengalih perhatian yang ampuh bagi yang sedang menjalani perawatan yang menimbulkan rasa sakit. Dengan main *game*, rasa sakit dan pening mereka berkurang, tensi darahnya pun menurun, dibandingkan dengan mereka yang hanya istirahat setelah diterapi”.

Penulis mengambil kesimpulan dari dua pengertian diatas tentang manfaat bermain *game* yang berupa meningkatkan seseorang menjadi fokus apa yang dilakukan dan mengalihkan kan perhatian menjadi sesuatu yang menyenangkan.

Adapun manfaat lainnya daari bermain *game* (Harsono, 2014:80-82) adalah sebagai berikut:

1. Melatih Konsentrasi

Dengan bermain *game* menjadi lebih mudah berkonsentrasi karena harus fokus untuk menyelesaikan *game* yang membutuhkan konsentrasi tinggi, sehingga lama kelamaan konsentrasi lebih mudah.

2. Melatih Kesabaran

Game juga melatih kesabaran karena ketika kita ingin memenangkan sebuah permainan kita harus mengatur strategi yang kuat dan disilin. Untuk membangun itu tidak bisa secara instan yakni harus bertahan satu demi satu sehingga bisa melatih kita untuk lebih sabar.

3. Pandai Bahasa Inggris

Memang tidak aneh lagi bagi para *gamer* bahwa bermain *game* membuatnya malah bisa belajar bahasa Inggris. Hal ini karena pada *game* kebanyakan menggunakan bahasa Inggris sehingga *gamer* mau tidak mau harus mengerti karena kelamaan memakai bahasa Inggris akhirnya jadi sedikit mengerti.

4. Melatih Kecepatan Tangan

Dengan bermain *game* membuat tangan menjadi reflek seolah-olah memiliki mata sendiri untuk menekan tombol *keyboard*. Akibatnya ketika mengetik akan terasa lebih cepat karena sudah terlatih.

5. Mudah Memecahkan Masalah

Dalam *game* sebenarnya isinya masalah yang harus kita selesaikan. Seorang *Gamer* akan lebih mudah memecahkan masalah setiap *game* jika sebelumnya memang sudah sering bermain *game* jadi semakin lama orang bermain *game* yang lain pun akan terasa mudah.

6. Mengetahui Ilmu Komputer

Dengan bermain *game* membuat kita harus mengetahui *game* apa saja yang bisa dimainkan di laptop, sehingga lama-kelamaan tahu *game-game* dan syarat spesifikasi laptop yang harus dipakai.

7. Pintar Mengatur Strategi

Game menuntut untuk membuat strategi karena *game* memiliki *level* yang meningkat yakni dari yang mudah sampai tingkat yang lebih susah. Apalagi *game* yang jenisnya strategi ini akan sangat melatih kecerdasan strategi seseorang.

8. Tidak Gampang Menyerah

Sudah tidak aneh lagi dengan kata-kata “*Game Over*” bagi para *gamer*. Dengan bermain *game* diri kita tidak gampang menyerah karena sudah berlatih dengan *game* yang tidak jarang mengalami kekalahan namun justru merasa penasaran untuk menantanginya lagi.

2.1.5 Media Pembelajaran

Menurut Zamroni (2013:489) “Pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. Pembelajaran merupakan bantuan yang diberikan pendidik agar dapat terjadi proses perolehan ilmu dan pengetahuan, penguasaan kemahiran dan tabiat, serta pembentukan sikap dan kepercayaan pada peserta didik. Dengan kata lain, pembelajaran adalah proses untuk membantu peserta didik agar dapat belajar dengan baik”.

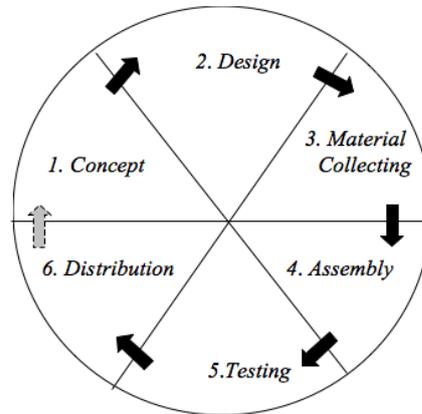
Berdasarkan dari kutipan di atas pembelajaran adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar.

2.2. Teori Pendukung

2.2.1. Metodologi Pengembangan Multimedia

Menurut Munir (2013:97) “Metodologi dalam pengembangan *software* selalu dikaitkan dengan kerangka kerja atau *framework* karena menggunakan pendekatan sistem informasi”. Tujuan mewujudkan kerangka kerja adalah untuk membimbing peneliti dalam mengembangkan *software* tersebut.

Menurut Munir (2013:104) “Metodologi pengembangan multimedia terdiri dari 6 tahapan, yaitu *concept*, *design*, *material collecting*, *assembly*, *testing* dan *distribution*”. Seperti gambar berikut ini:



Sumber: Sutopo dalam Munir (2013:104)

Gambar II.1.

Metodologi Pengembangan Multimedia

1. *Concept*

Tahapan *concept* (pengonsepan) adalah tahapan untuk menentukan tujuan dan siapa pengguna program (identifikasi audiens).

2. *Design*

Design (perancangan) adalah tahap pembuatan spesifikasi mengenai arsitektur program, gaya, tampilan dan kebutuhan materil/bahan untuk program.

3. *Material Collecting*

Material collecting adalah tahap pengumpulan bahan yang sesuai dengan kebutuhan yang dikerjakan.

4. *Assembly*

Tahap *assembly* adalah tahap pembuatan semua objek atau bahan multimedia.

5. *Testing*

Tahap *testing* (pengujian) dilakukan setelah menyelesaikan tahap pembuatan (*assembly*) dengan menjalankan aplikasi/program dan melihatnya apakah ada kesalahan atau tidak.

6. *Distribution*

Pada tahap ini, aplikasi akan disimpan dalam suatu media penyimpanan.

2.2.2. *HTML5*

Menurut Prasetya (2013:13-14) “HTML (*HyperText Markup Language*) telah mencapai versi 5 (atau dikenal *HTML5*) yang merupakan standar *web* masa depan”. *HTML5* merupakan revisi baru yang merujuk pada spesifikasi HTML 4.01. Proyek ini digagas oleh *Web Hypertext Application Technology Working Group* (WHATWG) karena melihat HTML 4.01 sudah cukup lama tidak diperbaharui oleh Konsorsium *word wide web* (*World Wide Web Consortium/W3C*). Saat itu W3C dan WHATWG berinisiatif untuk berkerja sama mengembangkan *HTML5*.

HTML5 sudah menyediakan fitur *built-in* (tanpa tambahan *plugin* atau API) untuk mengimplementasikan animasi, *canvas*, *event-handler*, validasi *form*, *drag & drop*, audio & video, *geolocation*, dan komunikasi *socket*.

Sebagai simpulan, *HTML5* dibangun dengan merujuk pada spesifikasi *HTML* 4.01. ini juga menegaskan bahwa *HTML5* menjamin adanya kompatibilitas yang baik terhadap versi di bawahnya.

Dari pengertian di atas *HTML5* (*Hypertext Markup Language*) sebagai bahasa pemrograman yang digunakan pada halaman *web* dalam pembuatan

dokumen dan aplikasi, sebagai bahasa untuk pertukaran dokumen *web*, dan menampilkan informasi lebih menarik pada halaman *web* yang terdiri dari berbagai macam format tampilan seperti teks, animasi *link*, *audio-video*.

2.2.3. Android

Menurut Safaat (2014:1) “*Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *Linux* yang mencakup sistem operasi, *middleware* dan aplikasi”. Pada masa saat ini kebanyakan vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis *Android*, vendor-vendor itu antara lain *HTC*, *Motorola*, *Samsung*, *LG*, *HKC*, *Huawei*, *Archos*, *Webstation Camangi*, *Dell*, *Nexus*, *SciPhone*, *WayteQ*, *Sony Ericsson*, *Acer*, *Philips*, *T-Mobile*, *Nexian*, *IMO*, *Asus* dan masih banyak lagi vendor *smartphone* didunia yang memproduksi *Android*.

Menyimpulkan dari pengertian di atas bahwa *android* adalah sistem operasi untuk perangkat *mobile* berbagai jenis *smartphone*.

Menurut Safaat (2014:3) “*Android* dipuji sebagai *platform mobile* pertama yang lengkap, Terbuka, dan Bebas”. Berikut ini beberapa kelebihan dari sistem operasi *Android* adalah sebagai berikut:

1. Lengkap (*Complete Platform*)

Sistem operasi *Android* menyediakan banyak *tools* dalamnya sehingga *developer* mudah dalam mengembangkan aplikasi yang diinginkannya.

2. Terbuka (*Open Source Platform*)

Platform Android yang bersifat *open source* (terbuka), hal ini yang membuat sistem operasi *Android* mudah dikembangkan oleh *developer*.

3. Bebas (*Free Platform*)

Developer tidak perlu membayar royalti untuk memperoleh *lisence* sehingga dapat dengan bebas mengembangkan, mendistribusikan dan memperdagangkan sistem operasi *Android*.



Sumber: Safaat (2014:1)

Gambar II.2.

Logo Android

2.2.4. PhoneGap

Menurut Prasetya (2013:117) “*PhoneGap* adalah sebuah kerangka kerja (*framework*) *open source* yang digunakan untuk membangun aplikasi-aplikasi *mobile* lintas *platform* dengan memanfaatkan teknologi-teknologi *web* standart, mencakup HTML, CSS, dan *JavaScript*. Secara khusus, *PhoneGap* didasarkan pada spesifikasi HTML5 yang mampu menghasilkan aplikasi-aplikasi *mobilemodern*”.

Selain itu, *PhoneGap* adalah sebuah *platform* HTML5 yang memungkinkan pengembang melakukan teknologi berbasis *web* (HTML, CSS, dan *JavaScript*) untuk membuat aplikasi *Mobile* asli (*native*). Dengan menggunakan *PhoneGap*, pengembang dapat menulis aplikasi mereka dan menyebarkannya ke enam *platform Mobile* utamadan toko aplikasi, termasuk *Apple iOS*, *Android*, *Blackberry*, *Windows Phone*, *WebOS*, Samsung Bada, dan *Symbian* (Triansah, dkk, 2015:2).

Berdasarkan dari kutipan di atas bahwa *Phone Gap* adalah sebuah kerangka kerja yang dipakai untuk membuat aplikasi *cross-platform mobile* dengan HTML, CSS, dan *JavaScript*. *Phone Gap* menjadi suatu solusi yang ideal untuk seorang *web developer* yang tertarik dalam pembuatan aplikasi di *smartphone*.

2.2.5. Construct 2

Menurut Permana (2015:114) “*Construct 2* adalah sebuah *tool* berbasis *HTML 5* untuk menciptakan sebuah permainan”. Dengan *tool Construct 2* memungkinkan siapa saja membuat *game* tanpa harus memiliki pengalaman pemrograman. Dikembangkan oleh *Scirra Ltd*, hal ini ditujukan terutama untuk para *non-programmer* yang ingin menciptakan sebuah *game* secara *drag and drop* menggunakan editor visual dan berbasis sistem logika perilaku. *Construct 2* ini didesain khusus untuk pengembang *game* yang *non-programmer* dimana hanya dibutuhkan logika dan teori *game* yang diciptakan. Untuk menciptakan sebuah *game*, pengembang hanya melakukan *drag and drop* menggunakan editor visual.



Sumber: Permana (2015:114)

Gambar II.3.

Tampilan *Construct 2*

Mengambil kesimpulan tentang penjelasan dari *Construct 2* yang merupakan salah satu tools yang digunakan untuk membuat *game* tanpa harus menulis kode pemrograman, karena sebagian besar logika untuk *game* dapat dibuat menggunakan menu, dengan begitu semua orang dapat membuat *game* tanpa harus belajar ribet dengan bahasa pemrograman.

2.2.6. Storyboard

Storyboard dalam pembuatan aplikasi pembelajaran interaktif digunakan sebagai sarana untuk merancang uraian yang berisi penjelasan, *layout* visual dan *audio* penjelasan dari masing-masing ditempatkan pada kolom. Satu kolom dalam *storyboard* mewakili satu tampilan di layar *monitor*. Sedangkan menurut Binanto (2010:255) “*Storyboard* merupakan pengorganisasi grafik, contohnya adalah sederetan ilustrasi atau gambar yang ditampilkan berurutan untuk keperluan visualisasi awal dari suatu *file*, animasi, atau urutan media interaktif, termasuk interaktivitas di *web*”.

2.2.7. Pengujian Black Box

Menurut Rizky (2011:264) “*Black-Box Testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya”.

Sedangkan menurut Rosa dan Shalahuddin (2013:275) “*Black-box Testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”.

Pengujian *Black Box*:

1. Digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang.
2. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut.
3. Dari keluaran yang dihasilkan, kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya.

Berdasarkan konsep pengujian *Black Box (functionality)* testing mengidentifikasi kesalahan yang berhubungan dengan kesalahan fungsionalitas PL yang tampak dalam kesalahan *output*.

Langkah-langkah dalam *Black Box*:

1. Analisa kebutuhan dan spesifikasi
2. Pemilihan *input*
3. Pemilihan *output*
4. Peleksi *input*
5. Pengujian
6. *Review* hasil
7. Evaluasi