

## BAB II

### LANDASAN TEORI

#### 2.1 Teori Intelegensi Ganda ( *Multiple Intelleginces Theory* )

Howard gardner, adalah seorang professor pendidikan dan psikologi dari Harvard University, Amerika Serikat, mendefinisikan *intelegensi* sebagai kemampuan untuk memecahkan persoalan dan menghasilkan produk dalam suatu *setting* yang bermacam-macam dan dalam situasi yang nyata [ SUP04].

Menurut Howard Gardner, intelegensi tidak hanya satu macam saja, tetapi dibagi menjadi beberapa bagian, dan tiap *intelegensi* merepresentasikan kemampuan seseorang dalam menyelesaikan suatu masalah yang dihadapi, adapun *intelegensi* dibagi atas 9 macam, yaitu [SUP04]:

1. Intelligensi Intrapersonal (*Intrapersonal Intelligence*)

Kemampuan yang berkaitan dengan pengetahuan akan diri sendiri dan kemampuan untuk bertindak secara adaptif berdasarkan pengenalan diri itu. Yang termasuk didalam jenis ini adalah kemampuan berefleksi dan keseimbangan diri seseorang.

2. Intelligensi Naturalis (*Naturalist Intelligence*)

Kemampuan seseorang untuk dapat mengerti tentang tumbuhan dan hewan dengan baik, dapat membuat distingsi konsekuensi lain dalam alam natural, kemampuan untuk memahami dan menikmati

alam, dan menggunakan kemampuan itu secara produktif dalam berburu, bertani, dan mengembangkan pengetahuan tentang alam.

3. *Inteligensi Eksistensial (Existential Intelligence)*

Kemampuan seseorang untuk menjawab persoalan-persoalan terdalam tentang eksistensi atau keberadaan manusia. Inteligensi ini dimiliki oleh para filsuf.

4. *Inteligensi Ruang-Visual (Spatial Intelligence)*

Kemampuan menangkap dunia ruang-visual secara tepat, seperti dimiliki *pemburu, arsitektur, navigator, dan decorator*. Yang termasuk dalam inteligensi ini adalah kemampuan mengenal bentuk dan benda secara tepat, melakukan perubahan suatu benda dalam pikirannya dan mengenali perubahan itu, serta mengungkapkan data dalam suatu grafik termasuk keseimbangan relasi, warna, garis dan bentuk.

5. *Inteligensi Kinestetik-Badani (Bodily-Kinesthetic Intelligence)*

Kemampuan seseorang menggunakan tubuh atau gerak tubuh untuk mengekspresikan gagasan dan perasaan seperti pada *aktor, atlet, penari, pemahat, dan ahli bedah*. Yang termasuk didalam inteligensi ini adalah ketrampilan koordinasi dan fleksibilitas tubuh.

6. *Inteligensi Musikal (Musical Intelligence)*

Kemampuan seseorang untuk mengembangkan, mengekspresikan, dan menikmati bentuk-bentuk musik dan suara. Yang termasuk

didalam inteligensi ini adalah kepekaan terhadap ritme, melodi, dan intonasi untuk memainkan alat musik, menyanyi, dan menciptakan lagu.

7. *Inteligensi Interpersonal (Interpersonal Intellegence)*

Kemampuan seseorang untuk mengerti dan menjadi peka terhadap perasaan, intensi, motivasi, watak, dan temperamen orang lain. Secara umum inteligensi ini berkaitan dengan kemampuan seseorang untuk menjalin relasi dan komunikasi dengan berbagai orang. Inteligensi ini banyak dimiliki oleh penggerak massa.

8. *Inteligensi Linguistik (Linguistic Intellegence)*

Kemampuan seseorang untuk menggunakan dan mengolah kata secara edektif, baik secara oral maupun tertulis seperti dimiliki para *pencipta puisi, editor, jurnalis, sastrawan, pemain sandiwara* maupun *orator*. Kemampuan ini berkaitan dengan penggunaan dan pengembangan bahasa secara umum.

9. *Inteligensi Matematis-logis (Logical-Mathematical Intellegence)*

Kemampuan seseorang yang berkaitan dengan penggunaan bilangan dan logika secara efektif, seperti dimiliki seorang *matematikus, ilmuwan, programmer, dan logikus*.

Teori inteligensi yang sesuai dengan program game yang akan dibuat adalah *Inteligensi Matematis-logis*, yaitu yang berkaitan dengan penggunaan logika dan sistematis untuk memecahkan permasalahan yang

dihadapi. Teori yang berikutnya adalah *Inteligensi Kinestetik Badani* yang berkaitan dengan daya reflek tubuh.

## 2.2 Teori Dalam Pembuatan Program Permainan

Urutan pembuatan game dan pengembangan program game adalah sebagai berikut [ HEN89] :

1. Tentukan tipe permainan apa yang ingin dibuat.

Penentuan ini sebagai dasar sebelum mulai bekerja sampai didapat ide yang kira-kira bagus untuk dibuat program permainan.

2. Definisikan model permainan dan tujuannya

Ide yang didapat itu dituangkan dalam bentuk model permainan yang dibuat. Pada tahap ini sebaiknya model permainannya ditulis secara jelas sehingga jika hendak menambah cerita, karakter lain dalam permainannya atau suatu aksi baru, maka bisa dilihat atau dicek dari yang sudah ditulis sehingga permainannya tetap konsisten dan tidak membingungkan.

3. Definisikan secara jelas *GAME WORLDS*-nya

*Games worlds* adalah elemen-elemen utama yang terdapat dalam suatu program permainan yang terdiri dari :

- a. *Game board*

Bentuk tampilan permainan, latar belakang, dan lainnya.

b. Instruksi untuk permainan

Instruksi untuk pemain harus jelas supaya tidak membingungkan dan pemain dapat menentukan strategi dari permainannya.

c. Informasi untuk pemain

Informasi ini penting ditampilkan dalam program permainan ketika sedang berjalan, misalnya skor, waktu dan lainnya.

d. Penghargaan

Penghargaan memegang peranan penting dalam program permainan karena dengan adanya penghargaan (*ucapan selamat dan lainnya*) setelah bermain maka penghargaan akan merangsang pemain untuk memainkan *level* yang lebih tinggi lagi.

e. Variasi

Program permainan tanpa *variasi* yang memadai akan membuat orang cepat bosan tetapi *variasi* juga tidak boleh berlebihan sehingga akhirnya tidak jelas karena selalu ada kemungkinan *variasi* yang berlebihan akan membuat alur permainan tidak konsisten.

f. Tingkat Kesulitan

Tingkat kesulitan dari program permainan akan membuat pemain menjadi bergairah dan merasa senang jika melewati tingkat kesulitan yang diberikan.

4. Pastikan bahwa bisa dimainkan

Permainan haruslah dapat dimainkan, karena tampilan grafik yang luar biasa cepat atau permainan yang tidak mungkin diselesaikan akan membuat program permainan tidak dapat dimainkan.

5. Rancang program sebaik mungkin

Gunakan teknik pemrograman yang sesuai dalam membuatnya dan selalu buatlah supaya program mudah dimodifikasi.

6. Pegujian program

Program yang selesai dibuat harus diuji. Pertama oleh perancangnya sendiri (*alpha tes*) untuk menentukan kesalahan *logis* yang mungkin terjadi. Kedua oleh orang lain (*Beta tes*) untuk mengetahui kesalahan *logis* yang tidak terlihat oleh perancangnya sehingga dapat memberi masukan, saran atau ide dalam pengembangan program yang sedang dibuat.

### 2.3 Pemrograman *Actionscript*

Pemrograman dengan *actionscript* dapat digunakan untuk pembuatan visualisasi di segala bidang terutama aplikasi interaktif berbasis multimedia dan *web*. *Actionscript* adalah pemrograman visual berontasi object. Seperti *JavaScript*. *Actionscript* mempunyai sintaks, tata bahasa, dan struktur yang mirip dengan bahasa pemrograman *C++*.

*Actionscript* mempunyai beberapa komponen, seperti *predifined object* dan *fungsi*. *Actionscript* mengikuti sintaks (*Syntax*), *keywords*, *operator* dan *variable* untuk menyimpan dan mengambil informasi. European Computer Manufacture Association (*ECMA*) membuat dokumen *ECMA-262* yang diambil dari *JavaScript*, dan *ECMA 262* menjadi spesifikasi *ActionScript*.

*Actionscript* merupakan bahasa pemrograman yang digunakan dalam Macromedia Flash. Penggunaan *Actionscript* mempunyai beberapa tujuan diantaranya adalah [SUT02] :

1. Memberikan kebebasan berkreasi bagi desainer.
2. Membuat animasi lebih interaktif.
3. Dapat menampilkan animasi tertentu (*non-linear*).

### 2.3.1 Sintaks Actionscript

*Actionscript* mempunyai aturan tata bahasa dan tanda baca sendiri sehingga mengenali awal dan akhir sebuah pernyataan. Suatu pernyataan diakhiri dengan tanda *semicolon*(;). *sintaks (syntax)* dalam *actionscript* ada dua macam, yaitu *dot syntax* dan *slash syntax*.

#### 2.3.1.1 Dot syntax

Dalam *Actionscript*, suatu dot (.) digunakan untuk [SUT02]:

1. menunjukkan *property method* dari *object* atau *movie clip*
2. mengenali *target path* ke suatu *movie clip* atau *variable*

*Dot syntax* dimulai dengan nama dari *object* atau *movie clip*, diikuti dengan *dot*, dan diakhiri dengan *property*, *method*, atau *variable*.

Contohnya, *movie clip* mempunyai *property*:

- a. *\_x* menunjukkan posisi *ordinat x* pada *stage*
- b. *setProperty( root.D4, x, 465)*: berarti menempatkan *movie clip* D4 dengan *ordinat x* bernilai 465

selain *property*, *movie clip* juga mempunyai *method* *play()*.

Expresi suatu *method* dari *object* atau *movie clip* mengikuti pola yang sama. *Dot syntax* juga digunakan untuk dua alias khusus yaitu *\_parent* dan *\_root*. Contohnya, *method* *play* dari *instance* lingkaranMc menggerakkan *playhead* pada *timeline* dari lingkaranMc, seperti pernyataan berikut :

### 1. *\_root*

- a. *Alias \_root* berhubungan dengan *timeline* utama.
- b. *Alias root* dapat digunakan untuk membuat *target part* yang *absolut*.

### 2. *\_parent*

- a. *alias parent* mengacu pada *movie clip* bila mempunyai hirarki.
- b. *Alias parent* dapat digunakan untuk membuat *target path* yang relatif.



### 2.3.1.2 Slash Syntax

*Slash syntax* digunakan dalam Flash 3 dan 4 untuk menunjukkan *target path* dan *movie clip* atau *variable*. Sintaks ini masih digunakan pada Flash 5 dan Flash MX, tetapi tidak disarankan. Dalam *Slash syntax*, *Slash* digunakan untuk menunjukkan *target path* atau *variable* di samping *dot syntax*. Untuk menunjukkan suatu *variable*, nama *variable* dituliskan sesudah *colon* (:).

*Slash syntax* biasanya digunakan untuk *action tellTarget*, yang juga tidak disarankan.

### 2.3.1.3 Kurung Kurawal

Pernyataan pada *Actionscript* dikelompokkan bersama ke dalam satu blok dengan kurung kurawal ( {} ).

## 2.3.2 Tipe Data

Tipe data menggambarkan macam elemen informasi yang dapat ditangani oleh *variable* dalam *Actionscript*. Terdapat dua macam tipe data yaitu [SUT02]:

1. Tipe data primitif (*string*, *number*, dan *Boolean*)

Tipe data yang mempunyai nilai konstanta, sehingga dapat menangani nilai aktual dan elemen yang ada.

2. Tipe data referensi (*movie clip* dan *object*)

Tipe data yang mempunyai nilai yang dapat berubah, sehingga berisi referensi dan nilai aktual yang dimiliki oleh elemen.

### a. *Movie clip*

*Movie clip* adalah simbol yang dapat menjalankan animasi pada Flash *movie*, berisi tipe data yang berhubungan dengan elemen grafik [SUT02]. Tipe data *movie clip* dapat mengontrol simbol *movie clip* dengan menggunakan *method* dari *object movie clip*. *Method* dapat dipanggil dengan *dot(.) operator*.

*method* dari *movie clip* *ikanMc* adalah *play()*. Pernyataan tersebut digunakan untuk menjalankan *movie clip* *ikanMc*. Bila dilakukan perubahan (*editing*) pada *symbol movie clip instance* yang terdapat pada *stage* dengan sendirinya akan berubah juga.

### b. Object

Suatu *object* adalah koleksi dari *property*, dimana setiap *property* mempunyai nama dan nilai [SUT02]. Nilai dan *property* dapat berupa tipe data primitif atau tipe data referensi. Hal ini dapat digunakan untuk menempatkan objek di dalam satu sama lain, atau membuat "*nest*". Untuk membuat spesifikasi *object* dengan *property* yang dimilikinya dapat digunakan *dot (.)*.

## 2.3.3 operator

*operator* merupakan simbol yang biasa dilibatkan dalam program untuk melakukan operasi atau manipulasi [SUT02]. *Operator* dapat menggabungkan,

membandingkan atau mengubah nilai suatu ekspresi. Elemen dimana *operator* melakukan *performansi* disebut *operan*. *Operator* terdiri dari beberapa macam dan mempunyai karakteristik serta penggunaan yang berbeda-beda. Contohnya, *operator +* menambahkan nilai *literal numerik* pada nilai *variable bilangan*, sedangkan *bilangan dan 5* adalah *operan*.

#### a. Dot Array Access Operator

*Dot operator (.)* dan *array operator ([])* dapat digunakan untuk mengakses *property* dari *prefined* atau *custom object*, termasuk *movie clip*, pada *dot operator*, nama *object* di sebelah kiri dan nama *property* atau *variable* di sebelah kanan. Nama *property* atau *variable* merupakan nama *identifer*, dan tidak boleh menggunakan *string* atau *variable* yang dievaluasi ke dalam *string*.

*Dot operator* dan *array operator* mempunyai peranan yang sama, tetapi mempunyai perbedaan sebagai berikut :

1. *Dot Operator* menggunakan *identifier* sebagai *property*.
2. *Array access operator* untuk mengisi dan mengakses isi dari *array*.

*Array Access operator* dapat digunakan:

1. Untuk mengambil nama *instance* dan *variable* secara dinamis. Ekspresi di dalam operator *[]* dievaluasi dan hasil

evaluasi digunakan sebagai nama variable yang diambil dari nama *movie clip*.

2. Pada sebelah kiri pernyataan *assignment*. Hal ini mengikuti inisialisasi nama *instance Variable*, dan *object* secara dinamis.
3. Untuk menyatakan *array* multidimensi

#### 2.3.4 Penggunaan Action

*Action* adalah pernyataan atau perintah *Actionscript* [SUT02}. Beberapa *action* ditempatkan pada satu *frame* atau *object* untuk menciptakan *Actionscript*. *Action* dapat independen satu dengan lainnya, seperti pernyataan berikut ini :

```
gotoAndplay(2);
stop();
loadMovie("logo.swf", 1);
```

*Action* bersarang (*nested actions*) dapat dibuat dengan menggunakan satu *action* dalam *action* lain, sehingga *action* akan saling mempengaruhi. Dalam contoh berikut, *action if* memerintah *gotoAndplay* untuk dieksekusi.

```
If (nilai < 67.0) {
    gotoAndplay(1);
    for (i=1; i<7; i++) {
```

```

    total = total + nilai;
  }
}

```

Action dapat menggerakkan *playhead* pada *timeline* (*gotoAndplay*), kontrol aliran *script* dengan perulangan (*do while*) atau kondisi logika (*if*), atau membuat fungsi dan variable baru (*function*, *setVariable*). Tabel dibawah ini berisi daftar semua *action* [DIK03].

Tabel 2.1 Action

Action				
<i>Break</i>	<i>Evaluate</i>	<i>Include</i>	<i>Print</i>	<i>stopDrag</i>
<i>Call</i>	<i>For..in</i>	<i>loadVariable</i> <i>es</i>	<i>removeMovieClip</i>	<i>tellTarget</i>
<i>Continue</i>	<i>fsCommand</i>	<i>nextFrame</i> <i>nextScene</i>	<i>Return</i>	<i>toggleHighQuality</i>
<i>Delete</i>	<i>Function</i>	<i>On</i>	<i>setVariable</i>	<i>stopDrag</i>
<i>Do...while</i>	<i>getURL</i>	<i>onClipEvent</i>	<i>setProperty</i>	<i>Trace</i>
<i>Duplicate</i>	<i>gotoAndplay</i> <i>gotoAndstop</i>	<i>Play</i>	<i>startDrag</i>	<i>unloadMovie</i>
<i>Else</i>	<i>If</i>	<i>prevFrame</i>	<i>Stop</i>	<i>Var</i>
<i>Else if</i>	<i>ifFrameLoaded</i>	<i>prevScene</i>	<i>stopAllSound</i>	<i>while</i>

#### 2.4 Adobe After Effect 5.5

Animasi pada adobe after effect 5.5 sangat bervariasi dan mempunyai keunggulan dibandingkan dengan seri sebelumnya [VIS04]. *Plugin-plugin* yang tersedia langsung dalam *software* juga sangat variatif.

Dalam pembuatan program game perang pesawat terbang nantinya akan menggunakan effect pergerakan langit, laut dan perkotaan.

