

BAB 2

LANDASAN TEORI

2.1 Musik

2.1.1 Interval

Bagaikan atom yang merupakan dasar dari segala *matter*, *interval* merupakan dasar dari melodi dan harmoni. *Interval* merupakan jarak antar dua nada. Terdapat dua macam jenis *interval*, yaitu *vertical interval* dan *horizontal interval* [12].

1. *Vertical Interval* yaitu *interval* dengan nada yang satu ditumpuk diatas nada yang lain dalam satu ketukan.
2. *Horizontal Interval* yaitu *interval* dengan nada yang di letakan setelah nada yang lain.

Tiap-tiap jarak *interval* memiliki namanya sendiri, berikut definisi dari tiap-tiap interval [1]:

1. *Perfect Unison*, *interval* ini memiliki jarak sebesar nol *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi P1 untuk mempersingkat istilah.
2. *Minor Second*, *interval* ini memiliki jarak sebesar satu *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi m2 untuk mempersingkat istilah.
3. *Major Second*, *interval* ini memiliki jarak sebesar dua *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi M2 untuk mempersingkat istilah.
4. *Minor Third*, *interval* ini memiliki jarak sebesar tiga *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi m3 untuk mempersingkat istilah.

5. *Major Third, interval* ini memiliki jarak sebesar empat *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi M3 untuk mempersingkat istilah.
6. *Perfect Fourth, interval* ini memiliki jarak sebesar lima *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi P4 untuk mempersingkat istilah.
7. *Augmented Forth / Diminished Fifth, interval* ini memiliki jarak sebesar enam *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi d5 atau A4 untuk mempersingkat istilah.
8. *Perfect Fifth, interval* ini memiliki jarak sebesar tujuh *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi P5 untuk mempersingkat istilah.
9. *Minor Sixth, interval* ini memiliki jarak sebesar delapan *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi m6 untuk mempersingkat istilah.
10. *Major Sixth, interval* ini memiliki jarak sebesar sembilan *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi M6 untuk mempersingkat istilah.
11. *Minor Seventh, interval* ini memiliki jarak sebesar sepuluh *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi m7 untuk mempersingkat istilah.
12. *Major Seventh, interval* ini memiliki jarak sebesar sebelas *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi M7 untuk mempersingkat istilah.
13. *Perfect Octave, interval* ini memiliki jarak sebesar dua belas *semitones* atau *half-step* dari nada *root*. *Interval* ini juga biasa disingkat menjadi P8 untuk mempersingkat istilah.

Istilah lain dari tiap-tiap *interval* dapat dibagi menjadi lima yaitu: *m (minor)*, *M (Major)*, *P (Perfect)*, *D (Diminished)*, *A (Augmented)*.

2.1.2 Scale

Scale atau tangga nada merupakan deretan nada yang dimulai dari nada *tonic* dan berakhir pada nada *tonic*. *Scale* dalam pada dasarnya dapat dianggap sebagai simbol, atau abstraksi dari aliran musik. Dalam sebuah *scale* jarak antara satu nada dengan nada berikutnya disebut dengan *step*. Untuk membangun sebuah *scale* dapat menggunakan rumus *step* dari *scale* tersebut [12].

2.1.3 Major Scale

Major Scale atau tangga nada mayor secara tradisional merupakan tangga nada yang terdiri dari *tonic*, *supertonic*, *mediant*, *subdominant*, *dominant*, *submediant* dan *leading tone* atau biasa disimbolkan dengan angka romawi. Lengkapnya dapat dilihat pada tabel 2.1.

Tabel 2.1 Nama Tradisional Nada-Nada Major Scale

Jenis	Simbol
<i>Tonic</i>	I
<i>Supertonic</i>	II
<i>Mediant</i>	III
<i>Subdominant</i>	IV
<i>Dominant</i>	V
<i>Submediant</i>	VI
<i>Leading Tone</i>	VII

Major Scale dapat dibangun dengan menggunakan rumus *step* W-W-H-W-W-W-H. W melambangkan dua *semitones* dan H melambangkan satu *semitones*. Sebagai contoh apabila kita memiliki *tonic* dengan nada C setiap iterasi dapat dilihat pada tabel 2.2.

Tabel 2.2 Interasi Rumus Major Scale

Nada Sebelum	Step	Nada Sesudah	Scale
C	W	D	CD
D	W	E	CDE
E	S	F	CDEF
F	W	G	CDEFG
G	W	A	CDEFGA
A	W	B	CDEFGAB
B	S	C	CDEFGABC

2.2 Game

2.2.1 Pengertian Game

Game atau permainan umumnya berfungsi sebagai hiburan. *Game* biasanya berisi aturan-aturan, tantangan, skor dan memiliki level dengan kesulitan yang beragam. *Game* merupakan aktivitas yang dilakukan untuk bersenang-senang, dalam sebuah permainan tentunya ada peraturan yang menentukan menang atau kalahnya pemain. *Game* merupakan alat yang efektif untuk belajar dikarenakan sifat game yang mampu menawarkan lingkungan hipotesis untuk siswa, di mana mereka dapat mengeksplorasi berbagai keputusan alternatif tanpa takut mengalami kegagalan. *Game* dapat dimanfaatkan untuk melatih kemampuan siswa dalam menyelesaikan permasalahan, mencari solusi, berfikir cepat serta berkompetisi [13]. *Game Single Player* atau pemain tunggal, pemain harus bermain dengan keahlian, berpacu dengan waktu, menyelesaikan permasalahan atau *puzzle*, mencari skor setinggi mungkin [14].

2.2.2 Platform Game

Beberapa klasifikasi *platform* game yang berjalan dapat dilihat pada bagian berikut [15]:

1. *Arcade Game* yaitu *game* yang biasa ditempatkan di mall, restaurant, taman hiburan atau tempat-tempat tertentu yang berupa box dengan layar dan perangkat kontrol *built-in* yang dapat dimainkan dengan memasukkan koin.
2. *Console Game* yaitu *video game* yang dapat dimainkan menggunakan perangkat *console* tertentu seperti Playstation 1, Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii.
3. *Personal Computer Game* yaitu *video game* dapat yang dimainkan menggunakan perangkat PC atau *Personal Computer*. *PC game* umumnya dapat memiliki kapasitas masukan, pemrosesan, keluaran baik audio maupun video yang lebih baik jika dibandingkan dengan *arcade game* dan *console game*.
4. *Mobile Game* yaitu *video game* dapat yang dimainkan menggunakan perangkat *mobile* atau perangkat *portable*. *Game* ini biasanya dapat dimainkan pada perangkat *smartphone*, *tablet*, *graphic calculator* atau *handheld-console*.

2.2.3 Genre Game

Berikut ini beberapa pengelompokan jenis *game* berdasarkan *genre*-nya diantaranya adalah [16]:

1. *Action*, Mekanik utama *game action* berkisar pada akurasi, pergerakan, kecepatan mengambil keputusan, reflek dan *timing*.
2. *Adventure, Game* petualangan menekankan pengalaman cerita melalui dialog dan pemecahan teka-teki. Mekanisme *gameplay* menekankan pada pengambilan keputusan atas tindakan. Pemecahan teka-teki biasanya berkisar pada menggabungkan atau memanipulasi *item* untuk menjalankan cerita.

3. *Educational, Game* edukasi mengajar pemain melalui permainan. Biasanya ditujukan untuk anak kecil, game edukasi menawarkan cara yang menyenangkan dan tidak langsung melatih mata pelajaran yang "tidak menyenangkan" seperti mengeja, matematika, sejarah, dll. Game edukasi sering disebut "Edutainment".
4. *Racing / Driving, Game* Balap/Mengemudi memungkinkan pemain untuk balapan, atau mengendarai kendaraan. Balapan dapat dilakukan di kendaraan, di atas tunggangan, berjalan kaki atau dalam grafik yang sepenuhnya abstrak. Game dengan genre ini harus memiliki aspek *racing/driving* untuk sebagian besar *gamplay game*-nya, bukan hanya sebagai *short sequence*.
5. *Role-Playing (RPG), Game* Role-Playing (RPG) termasuk dalam genre game yang bervariasi yang berfokus pada pengembangan karakter.
6. *Simulation*, Terdapat banyak jenis *game* simulasi. Kesamaan dari semua simulasi adalah *game* simulasi dimodelkan secara realistis dengan situasi dan/atau variabel kehidupan nyata daripada genre *game* yang lainnya.
7. *Sports, Game* olahraga adalah *game* di mana pemain mengontrol baik pemain atau manajer olahraga nyata atau fiksi.
8. *Strategy, Game* strategi berkisar pada penggunaan sumber daya strategis dan/atau taktis sering kali dalam skenario pertempuran atau manajerial.

2.2.4 Perspective dan Viewpoints Game

Berikut ini beberapa pengelompokan jenis *game* berdasarkan *perspective* dan *viewpoints*-nya diantaranya adalah [16]:

1. *First Person*
2. *Third Person*
3. *Isometric*
4. *Platform*
5. *Side-Scrolling*
6. *Top-Down*

2.2.5 Game Edukasi

Berikut ini beberapa pengelompokan jenis *game* berdasarkan konten dari *game* edukasi yang diangkat diantaranya adalah [16]:

1. *Ecology / Nature*
2. *Foreign Language*
3. *Geography*
4. *Graphics / Art*
5. *Health / Nutrition*
6. *History*
7. *Math / Logic*
8. *Music*
9. *Pre-school / Toddler*
10. *Reading / Writing*
11. *Religion*
12. *Science*
13. *Sociology*
14. *Typing*

2.3 Game Development

2.3.1 Unity

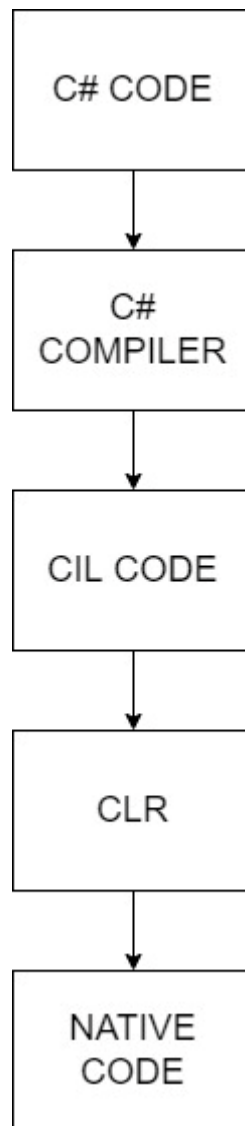
Unity (umumnya dikenal sebagai Unity3D) adalah *game engine* dan *integrated development environment* (IDE) untuk membuat media interaktif, biasanya video game. Unity dapat digunakan untuk membantun *game* bergrafis 2D, 3D, *Augmented Reality* serta *Virtual Reality*. Unity terkenal dengan kemampuannya melakukan prototyping yang cepat dan dengan target penerbitan yang berjumlah besar. Unity menggunakan bahasa C# untuk bagian *scripting*. Unity memiliki beberapa *build target* yang dapat melakukan *compilation* kode menjadi bahasa *native* pada *platform* yang dituju. Berikut beberapa *target build* yang disediakan:

1. StandaloneOSX
2. StandaloneWindows
3. iOS

4. Android
5. StandaloneWindows64
6. WebGL
7. WSAPlayer
8. StandaloneLinux64
9. PS4
10. XboxOne
11. tvOS
12. Switch
13. Stadia
14. CloudRendering
15. PS5

2.3.2 C#

Common Intermediate Language (CIL) adalah semacam kode biner yang membutuhkan untuk ditafsirkan. CIL adalah platform dan bahasa yang independen, seperti *C# open standard*. Keuntungan menggunakan interpretasi *Common Language Runtime* (CLR) alih-alih kode mesin adalah bahwa ada kerangka kerja yang jelas dan tidak ada masalah file .dll yang tidak kompatibel dan dengan demikian menghindari masalah seperti yang biasa disebut "DL Hell" pada buku "Avancerad Pocket Visual Basic NET". CLR mengkompilasi kode CIL ke kode mesin ketika bagian dari kode tersebut digunakan, disebut *Just In Time Compilation*. Ini memungkinkan pengoptimalan platform dan keamanan jenis runtime. Deretan proses dapat dilihat pada gambar 2.1.



Gambar 2.1 Urutan Proses Kompilasi