BAB II
LANDASAN TEORI

2.1 Musik

Dalam bagian ini akan dibahas tentang pengertian musik, interval, jenis-jenis interval, identitas interval dan *ear training*.

2.1.1 Pengertian Musik

Dalam Kamus Besar Bahasa Indonesia, musik berarti nada atau suara yg disusun demikian rupa sehingga mengandung irama, lagu, dan keharmonisan. Musik juga diartikan sebagai bahasa emosi di mana perasaan komposer dikomunikasikan kepada pendengar [**7**]. Sedangkan menurut buku *Seni dan Budaya* karya Harry Susantio, dkk., Musik adalah bunyi yang diatur menjadi pola yang dapat menyenangkan telinga atau dapat mengkomunikasikan telinga, perasaan, dan suasana hati. Musik memiliki ritme, melodi, dan harmoni yang memberikan kedalaman dan memungkinkan penggunaan beberapa instrumen atau bunyi-bunyian. Apabila instrumen atau suara dimainkan atau dinyanyikan bersama pada nada yang berbeda, instrumen dan suara itu tampil dalam harmoni [**8**].

2.1.2 Interval

Interval merupakan dasar dari teori musik. Sebelum mempelajari teori lain seperti *scales*, *chords*, dll., seseorang harus terlebih dahulu memahami interval. Interval adalah jarak antar dua nada musik. Baik itu jarak ke atas maupun jarak ke bawah. Ketika seseorang mendengar musik, maka ia mendengar interval.

Sistem pelabelan interval adalah bagian dari sejarah musik yang telah ada selama lebih dari 300 tahun. Satuan terkecil interval adalah 1/2 atau *semitone*.1 oktaf intervalnya adalah 6 atau 12 *semitone* [**9**].

Interval memiliki dua tipe, yaitu [**10**]:

1. *Harmonic* : Interval yang didapat ketika dua nada dimainkan secara bersamaan.
2. *Melodic* : Interval yang didapat ketika memainkan dua nada secara terpisah, satu setelah yang lain.

Interval juga memiliki dua jenis identitas yang berlaku untuk kedua tipe di atas. Identitas interval ditentukan oleh *Quantity*, dan *Quality*.

Interval Quantity didasarkan pada jumlah garis dan spasi yang terkandung oleh interval pada staf musik. *Interval Quantity* terdiri atas :

1. Unison
2. Second
3. Third
4. Fourth
5. Fifth
6. Sixth
7. Seventh
8. Octave

Interval Quality adalah apa yang memberikan interval suara yang berbeda. *Interval Quality* didasarkan pada jumlah setengah *step* dari satu not ke not yang lain. *Interval Quality* terdiri atas :

1. Major
2. Minor
3. Perfect
4. Diminished
5. Augmented

2.1.3 Ear Training

Ear Training merupakan latihan yang sangat penting bagi musisi, lebih penting daripada *performing*, *conducting*, bahkan lebih penting daripada berlatih alat musik [**11**]. Karena musik adalah seni yang sangat erat hubungannya dengan suara, maka sangat penting bagi musisi untuk melatih telinganya agar bisa membedakan nada dalam musik yang ia dengarkan.

Kemampuan seseorang dalam membedakan nada dibagi dalam dua jenis, yaitu *Perfect Pitch* dan *Relative Pitch*. *Perfect pitch* merupakan kemampuan untuk menamai atau menyanyikan suatu not secara tiba-tiba, tanpa ada not lain sebagai referensi. Sedangkan *Relative Pitch* merupakan kemampuan untuk menamai atau menyanyikan suatu not dengan menggunakan not yang telah dimainkan atau diingat sebelumnya sebagai referensi. *Relative pitch* lebih berkaitan dengan mengetahui jarak antar not [**12**].

2.2 Edukasi

Edukasi atau pendidikan ialah suatu usaha yang sadar yang teratur dan sitematis, yang dilakukan oleh orang – orang yang diserahi tanggung jawab untuk mempengaruhi anak agar mempunyai sifat dan tabiat yang sesuai dengan cita – cita pendidikan. Edukasi juga bisa diartikan sebagai bantuan yang diberikan secara sengaja kepada anak dalam pertumbuhan jasmani maupun rohani.

Secara teoritis pengertian mendidik dan mengajar tidaklah sama. Mengajar berarti menyerahkan atau manyampaikan ilmu pengetahuan atau keterampilan dan lain sebagainya kepada orang lain, dengan menggunakan cara – cara tertentu sehingga ilmu – ilmu tersebut bisa menjadi milik orang lain.

Lain halnya mendidik, bahwa mendidik tidak hanya cukup dengan hanya memberikan ilmu pengetahuan ataupun keterampilan, melainkan juga harus ditanamkan pada anak didik nilai – nilai dan norma – norma susila yang tinggi dan luhur. Dari pengertian di atas diketahui bahwa mendidik lebih luas dari pada mengajar. Mengajar hanyalah alat atau sarana dalam mendidik .dan mendidik harus mempunyai tujuan dan nilai – nilai yang tinggi [**13**].

2.2.1 Unsur – unsur Edukasi

Unsur – unsur edukasi adalah [**14**]:

1. Anak didik, yaitu pihak yang menjadi objek utama edukasi.
2. Pendidik, yaitu pihak yang menjadi subjek dari pelaksanaan edukasi.
3. Materi, yaitu bahan atau pengalaman belajar yang disusun menjadi kurikulum.
4. Alat edukasi, yaitu tindakan yang menjadi kelangsungan mendidik.

5. Lingkungan, yaitu keadaan yang berpengaruh terhadap hasil edukasi.
6. Dasar dan landasan edukasi, yaitu landasan yang menjadi fundamental dari segala kegiatan edukasi.

2.2.2 Faktor – faktor yang membatasi kemampuan edukasi

Adapun faktor-faktor yang membatasi kemampuan edukasi diantaranya [**14**]:

1. Faktor anak

   Di dalam anak didik terdapat potensi-potensi yang butuh edukasi dari luar
2. Faktor pendidik

   Guru mempunyai metode penyampaian yang berbeda dan beragam
3. Faktor lingkungan

   Lingkungan sangat berpengaruh baik positif maupun negatif

2.3 Game

Dalam bagian ini akan dibahas pengertian game, klasifikasi game berdasarkan platform maupun berdasarkan genre, serta game edukasi.

2.3.1 Pengertian Game

Istilah "*video game*" awalnya mengacu pada jenis spesifik perangkat, yaitu sebuah sistem komputer yang menciptakan sinyal tampilan video untuk televisi, tapi kini istilah tersebut telah menjadi istilah yang mencakup segala jenis perangkat [**15**].

Video game dikendalikan oleh komputer melalui interaksi pengguna dan imersi audio-visual, *video game* memungkinkan pemain untuk menanggapi situasi yang menantang dalam dunia fantasi. Sangat penting untuk dicatat bahwa "fantasi" merujuk pada fakta bahwa dunia yang dimaksud adalah maya, dan tidak nyata, misalnya pemain dapat menjelajahi fantasi bermain di liga utama bisbol.

2.3.2 Klasifikasi Game

Game bisa diklasifikasikan kedalam beberapa hal, diantaranya klasifikasi berdasarkan *platform*, dan berdasarkan *genre*.

2.3.2.1 Berdasarkan Platform

Tiga kategori utama platform pada game adalah *personal computer (PC)*, *console*, dan *mobile* [**15**].

1. Personal Computer (PC)

PC game adalah game yang dibuat untuk computer baik berbasis Windows, Mac, ataupun Linux. *PC* menyediakan kekuatan grafis dan pemrosesan yang kuat yang memungkinkan pengembang untuk membuat game yang mutakhir. Tetapi kelemahan game *PC* adalah mahal karena pengguna harus menggunakan hardware yang *up-to-date* untuk memainkan game *PC* dengan baik. Selain itu, kelemahan game *PC* bagi pengembang adalah banyaknya varian dari konfigurasi *PC* membuat sulit bagi pengembang untuk memastikan game tersebut berjalan dengan benar pada semua *setup PC*.

2. Console

Console adalah *hardware* yang dibuat oleh pihak ketiga seperti Sony, Microsoft, dan Nintendo. *Console* terhubung ke televisi dan tujuan utamanya adalah untuk bermain game. *Game Console* sangat menarik bagi game pengembang karena mereka hanya perlu memikirkan satu konfigurasi *hardware* ketika membuat *software* untuk konsol. Sangat kontras dengan *PC* yang memiliki opsi konfigurasi yang tak terbatas.

3. Mobile

Mobile platform terdiri atas sesuatu yang *portable* dan bisa digenggam, termasuk ponsel, PDA, iPods, dan *handheld game* seperti Nintendo DSi atau Sony PSP. *Game mobile* memiliki kontrol yang sederhana (terutama jika dibandingkan dengan *PC*).

2.3.2.2 Berdasarkan Genre

Menurut buku “Fundamentals of Game Design” [**16**], genre game bisa dibagi menjadi 9, yaitu:

1. Action

Action game adalah game dimana kebanyakan dari tantangan yang disajikan merupakan dari tes *physical skill* dan koordinasi pemain. Salah satu sub-genre

action game adalah *shooters game*, baik yang 2D maupun 3D seperti *First Person Shooters* (FPS).

2. Strategy

Strategy game menantang pemain untuk mencapai kemenangan dengan perencanaan, khususnya melalui perencanaan serangkaian tindakan yang dilakukan melawan satu lawan atau lebih. Kemenangan diraih dengan perencanaan matang dan pengambilan keputusan yang optimal.

3. Role Playing Game (RPG)

RPG adalah game dimana pemain mengontrol satu atau lebih karakter yang biasanya di desain oleh pemain itu sendiri, dan memandu mereka melewati berbagai rintangan yang diatur oleh komputer. Perkembangan karakter dalam hal kekuatan dan kemampuannya adalah kunci dari game jenis ini.

4. Sports

Sports game mensimulasikan berbagai aspek dari olahraga atletik nyata maupun imajiner, apakah itu memainkan pertandingan, me-*manage* tim dan karir, atau keduanya. Salah satu contoh game jenis ini adalah *Pro Evolution Soccer 2012* (PES 2012), dimana pemain bisa memainkan pertandingan, menjadi manajer tim, maupun menjadi pemain dan mengembangkan karirnya sendiri.

5. Vechicle Simulation

Vechicle simulation membuat feeling mengendarai kendaraan, baik kendaraan nyata maupun kendaraan imajiner. Performa dan karakteristik kasar mesin harus menyerupai kenyataan, kecuali jika yang didesain adalah kendaraan imajiner.

6. Construction and Management Simulation

CMS game adalah game tentang proses. Tujuan pemain bukan untuk mengalahkan musuh, tetapi membangun sesuatu dengan konteks proses yang sedang berjalan. Semakin pemain mengerti dan mengontrol proses, semakin sukses sesuatu yang ia bangun. Game seperti ini biasanya menyediakan dua jenis permainan, yaitu mode bebas dimana pemain bebas membangun sesuatu, dan mode misi dimana terdapat skenario hal apa yang harus dibangun oleh pemain.

7. Adventure

Adventure game adalah cerita interaktif tentang karakter protagonis yang dimainkan oleh pemain. Penyampaian cerita dan eksplorasi adalah elemen inti dari game ini. Penyelesaian teka-teki dan tantangan konseptual adalah bagian besar dari permainan.

8. Artificial Life and Puzzle Game

Artificial Life game adalah game yang membuat tiruan dari kehidupan sebenarnya. Biasanya ada dua jenis game ini, tiruan kehidupan manusia, contohnya The SIMS, dan tiruan kehidupan binatang, contohnya Tamagochi.

9. Online Game

Istilah *online game* disini mengacu kepada *multiplayer* game dimana mesin dari para pemain terhubung dengan jaringan.

2.3.3 Game Edukasi

Game edukasi adalah game digital yang dirancang untuk pengayaan pendidikan (mendukung pengajaran dan pembelajaran), menggunakan teknologi multimedia interaktif [**17**].

Kriteria perancangan game edukasi yang ideal dibagi menjadi enam *point*, yaitu :

1. Rasa ingin tahu, fantasi dan kontrol
    a. Motivasi intrinsik, menyenangkan untuk digunakan. Pengguna memiliki kontrol terhadap permainan.
    b. Integrasi antara materi edukasi dan aspek fantasi serta permainan
    c. Pengguna memegang kendali permainan dan dapat mengatur kecepatan bermain.
    d. Mendorong keingintahuan, pengguna dapat melakukan eksplorasi bebas. Permainan mengandung rahasia tersembunyi.
    e. Pengguna dapat menyimpan kemajuan permainan.
    f. Menyediakan lebih dari jalur. Ada beberapa cara untuk memenangkan permainan.

g. Pengguna dapat melakukan *trial-error* untuk menyelesaikan permainan.
h. Keberhasilan ditentukan oleh pengetahuan, bukan kebetulan.
i. Task disajikan secara *incremental* dan bertahap.
j. Simulasi realistik dunia.
k. Materi edukasi disesuaikan dengan materi dunia nyata.
l. Personalisasi, yaitu pengguna dapat mengubah karakter dan objek-objek.
m. Menyediakan banyak pilihan dan tema.
n. Hindari pengulangan, hindari *drill-practice*.
o. Sediakan elemen tidak terduga dan kejutan.
p. Poin tidak hilang saat menjawab pertanyaan.

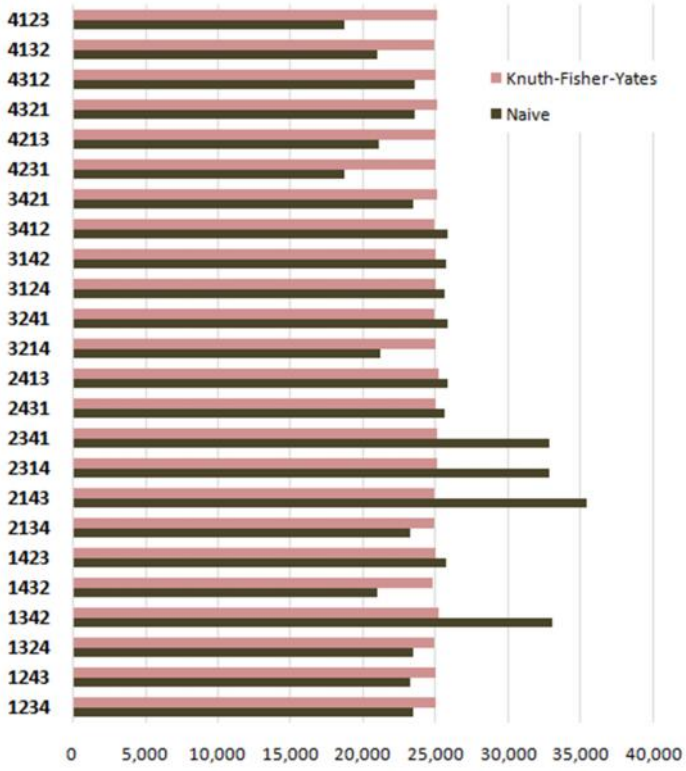
2. Tantangan
   a. Kinerja pemain terukur dan didefinisikan dengan baik.
   b. Tantangan diberikan secara terus menerus tapi disesuaikan dengan tingkat pemahaman pemain.
   c. Kompleksitas semakin berkembang sejalan dengan kemampuan pengguna. Tersedia banyak level.
   d. Memonitor kinerja pengguna, berdasarkan hasil *monitoring* ini kesulitan kemudian disesuaikan.
   e. Pengguna dapat melihat *progress*nya setiap saat.
   f. Menyediakan *hint* dan instruksi untuk membantu pengguna.
   g. Bantuk pengguna menemukan ‘*flow state’* yaitu keseimbangan antara tantangan dengan kemampuan untuk menghadapi tantangan.
3. Sosialisasi
   a. Pemain dapat berkolaborasi bersama.
   b. Pemain dapat saling berkompetisi.
   c. Pemenang bisa lebih dari satu (*multiple winner*)

4. Pedegogi
    a. Menyebutkan secara eksplisit target umur dan menyesuaikan rancangan permainan sesuai umur tersebut.
    b. Aktivitas pembelajaran dilaksanaan bersamaan dengan permainan.
    c. Objektif pembelajaran jelas.
    d. Memberi kesempatan kepada guru dan orangtua untuk ikut berperan.
    e. Memberikan petunjuk yang jelas sehingga pemain dapat berkonsentrasi kepada isi permainan, bukan bagaimana cara menggunakan permainan.
    f. Menyediakan tutorial sehingga pemain tidak perlu membaca manual.
5. Teknologi
    a. Menggunakan teknologi yang tersedia di sekolah dan masyarakat umum (tidak membutuhkan *hardware* yang terlalu tinggi)
    b. Memberikan lisensi yang terjangkau sekolah.
    c. Menggunakan antarmuka yang intuitif.
6. Pengguna anak-anak dan yang berkebutuhan khusus
    a. Memberikan petunjuk yang dibacakan selain petunjuk tertulis.
    b. Gambar, objek, layar tertata rapi.
    c. Permainannya sendiri merupakan aktivitas yang penting (*play for the sake of play*)
    d. Transformasi terlihat. Saat anak melakukan interaksi, ada *instant feedback*.
    e. *Input* dan *output* yang mudah terlihat.
    f. Tantangan diberikan secara *gradual*.
    g. Permainan menyenangkan untuk diulang, beberapa karakter, musik mudah teringat oleh anak.
    h. Objek dan suara diambil dari kehidupan sehari-hari yang mudah dikenali.
    i. Menginspirasi anak, bahkan setelah komputer dimatikan.

2.4 Fisher-Yates Shuffle

Fisher-Yates Shuffle (dinamai berdasarkan penemunya, Ronald Fisher dan Frank Yates) digunakan untuk mengubah urutan masukan yang diberikan secara acak. Permutasi yang dihasilkan oleh algoritma ini muncul dengan probabilitas yang sama [**18**].

Algoritma ini dinyatakan bias karena permutasi yang dihasilkan oleh algoritma ini muncul dengan probabilitas yang sama, hal ini dibuktikan dengan percobaan mengacak suatu set kartu yang dilakukan berulang – ulang seperti yang bisa dilihat pada gambar II.1.



Gambar II.1 Hasil perbandingan algoritma shuffle [**19**]

Pada gambar II.1 dilakukan tes terhadap 4 kartu, yang kemudian diacak sebanyak 600.000 kali. Keterangan pada sumbu Y menunjukkan permutasi / kemungkinan kombinasi kartu yang muncul, sedangkan keterangan pada sumbu X menunjukkan jumlah kemunculan kombinasi itu muncul. Warna pink menunjukkan hasil dari algoritma Fisher-Yates shuffle sedangkan warna abu-abu kehijauan menunjukkan algoritma Naïve shuffle. Hasilnya algoritma Fisher-Yates

menghasilkan nilai yang hampir sama untuk setiap kemungkinan kombinasi kartu, sedangkan pada algoritma Naïve kombinasi kartu tertentu muncul jauh lebih sering dan sebagian lagi muncul lebih sedikit dibanding kombinasi kartu lain.

Metode Fisher-Yates secara umum adalah :

1. Ketika masih ada elemen tersisa untuk diacak
2. Ambil elemen secara acak dari elemen yang tersisa
3. Kemudian tukar dengan elemen saat ini.

Gambar II.2 menunjukkan algoritma fisher – yates shuffle yang ditulis dalam bahasa javascript.

```
function shuffle(array) {
  var m = array.length, t, i;

  // While there remain elements to shuffle…
  while (m) {

    // Pick a remaining element…
    i = Math.floor(Math.random() * m--);

    // And swap it with the current element.
    t = array[m];
    array[m] = array[i];
    array[i] = t;
  }

  return array;
}
```

Gambar II.2 Algoritma Fisher-Yates Shuffle [**20**]

2.5 Tools

Pada sub bab ini akan dijelaskan *tools* yang digunakan dalam pembangunan game edukasi *adventure* teori musik ini.

2.5.1 Object Oriented Paradigm

Konsep dasar dari paradigma berorientasi objek (cara melakukan sesuatu) relatif mudah untuk dipahami dan diterapkan. Alan Kay, penemu Smalltalk, telah mengerjakan *'Personal Computer for Children of all Ages*' sejak 1968, sebagai sasarannya adalah anak-anak. Tidak mengherankan bahwa konsep dasarnya sederhana [**21**].

Pendekatan berorientasi objek diciptakan (atau lebih tepatnya, berkembang) karena kesulitan orang – orang yang telah berusaha untuk mendapatkan sistem berkualitas baik, dihasilkan tepat waktu, dan sesuai anggaran, terutama untuk sistem yang besar dengan banyak orang yang terlibat.

Beberapa justifikasi yang biasanya diberikan kepada orientasi objek :

1. Objek lebih mudah dipahami masyarakat
2. Spesialis dapat berkomunikasi lebih baik.
3. Data dan proses tidak dipisahkan secara artifisial.
4. Kode dapat digunakan kembali lebih mudah.
5. Orientasi Obyek matang dan terbukti dengan baik.

2.5.2 Unified Modelling Language (UML)

Pada pertengahan 1990-an, metodologi yang paling terkenal adalah yang ditemukan oleh Ivar Jacobson, James dan Grady Booch. Masing-masing memiliki perusahaan konsultan sendiri, menggunakan metodologi sendiri dan notasi sendiri. Pada tahun 1996, Jacobson dan Rumbaugh bergabung Rasional Corporation (didirikan oleh Booch), dan mereka mengembangkan satu set notasi yang dikenal sebagai Unified Modeling Language (UML) [**21**].

UML memiliki 13 jenis diagram. Spesifikasi UML tidak mengatakan di mana diagram harus digunakan dalam metodologi tertentu.

1. *Use Case Diagram* mengkategorikan cara di mana sistem yang digunakan.
2. *Class Diagram* menunjukkan kelas dan bagaimana mereka dapat dipasang bersama-sama (mereka juga dapat menunjukkan *object*).
3. *Object Diagram* memperlihatkan obyek saja dan bagaimana mereka dapat dipasang bersama-sama.
4. *Activity Diagram* menunjukkan aktivitas oleh manusia atau objek dalam cara yang mirip dengan *Flow Chart*.
5. *State Machine Diagram* menunjukkan berbagai state setiap obyek dengan siklus hidup yang menarik atau rumit.

6. *Communication Diagram* menunjukkan pesan yang dikirim antara objek dalam skenario tertentu.
7. *Sequence Diagram* menunjukkan informasi yang sama dengan *communication diagram*, tetapi menekankan urutan daripada koneksi.
8. *Package Diagram* menunjukkan bagaimana kelas terkait dikelompokkan bersama-sama, untuk kepentingan pengembang.
9. *Deployment Diagram* menunjukkan mesin, proses dan artefak dikerahkan untuk jadi seorang sistem.
10. *Component Diagram* menunjukkan komponen dapat digunakan kembali (benda atau subsistem) dan *interface* mereka.
11. *Interaction Overview Diagram* menunjukkan langkah-langkah individu dari suatu kegiatan menggunakan *sequence diagram*.
12. *Timing Diagram* menunjukkan kendala waktu yang tepat untuk pesan dan menyatakan objek.
13. *Composite Structure Diagram* menunjukkan bagaimana benda cocok bersama dalam suatu agregasi atau komposisi, menampilkan antarmuka dan berkolaborasi benda.

2.5.3 ActionScript 3.0

ActionScript 3.0 adalah adalah bahasa berorientasi objek untuk membuat aplikasi dan konten multimedia ber-*script* untuk pemutaran di *runtime* Flash *client* (seperti Flash Player dan Adobe AIR). Dengan sintaks yang mirip dengan Java dan C#, bahasa ActionScript seharusnya akrab bagi programmer berpengalaman [**22**].

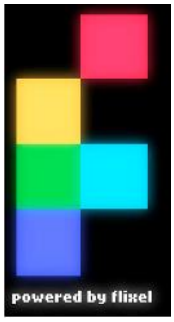
Beberapa fitur kunci dari bahasa ActionScript 3.0 adalah :

1. *First-class support* untuk *constructs* berorientasi objek umum, seperti kelas, objek, dan interface
2. Single-threaded execution model
3. Runtime type-checking
4. Compile-time type-checking opsional

5. Fitur dinamis seperti *runtime* pembuatan fungsi konstruktor baru dan variable
6. Runtime exceptions
7. *Direct support* untuk CML sebagai tipe data *built-in*
8. *Package* untuk pengorganisasian *code libraries*
9. Namespaces untuk kualifikasi identifier
10. Regular expressions

2.5.4 Flixel

Flixel merupakan *library game-making open source* yang benar – benar gratis untuk penggunaan pribadi atau komersial. Flixel ditulis seluruhnya dalam ActionScript 3 oleh Adam Saltsman "Atom", dan dirancang untuk bisa digunakan dengan *development tools* gratis. Flixel mudah dipelajari, diperluas dan disesuaikan. Flixel telah digunakan dalam ratusan game, termasuk nominasi IGF, game “Adult Swim”, dan “avant-garde experiment”. Banyak pengguna Flixel membuat game pertamanya di Flixel [**23**]. Logo flixel dapat dilihat pada gambar II.3.



Gambar II.3 Logo Flixel

Fitur dasar flixel antara lain :

1. Menampilkan ribuan benda bergerak
2. Basic collision antar objek
3. Pengelompokan objek bersama untuk menyederhanakan.
4. Mudah menghasilkan dan memancarkan partikel
5. Membuat tingkat permainan menggunakan *tilemaps*
6. Menampilkan teks, menyimpan game, *scrolling*

7. Input mouse & keyboard
8. Utilitas matematika & warna

Sedangkan fitur *advance*-nya antara lain :

1. Merekam dan memutar kembali replay
2. Powerful interactive debugger
3. Sistem kamera untuk layar *split*
4. *Pathfinding* dan mengikuti
5. Mudah untuk mendaur-ulang objek.

2.5.5 Extensible Markup Language (XML)

XML merupakan singkatan dari *Extensible Markup Language*. Pengembangan dan penggunaannya telah mengikuti jalur umum dalam perangkat lunak dan dunia IT. Ini dimulai lebih dari sepuluh tahun yang lalu dan pada awalnya sangat sedikit yang menggunakannya, tetapi kemudian menarik perhatian publik dan mulai merasuki dunia pertukaran data. Setelah itu, utilitas yang tersedia untuk memproses dan mengelola XML menjadi lebih canggih, sedemikian rupa sehingga banyak orang mulai menggunakannya tanpa benar-benar menyadari keberadaannya. Akhir-akhir ini telah terjadi sedikit reaksi di tempat tertentu atas kegagalan yang dirasakan dan titik lemah, yang telah menyebabkan alternatif yang diusulkan berbagai perbaikan.

Namun demikian, XML sekarang memiliki tempat permanen di sistem TI dan sulit untuk membayangkan aplikasi non-sepele yang tidak menggunakan XML baik untuk konfigurasi atau data sampai tingkat tertentu. Untuk alasan ini sangat penting bahwa pengembang perangkat lunak modern memiliki pemahaman menyeluruh tentang prinsip-prinsipnya, apa yang ia mampu, dan bagaimana menggunakannya untuk keuntungan mereka.

Ada dua penggunaan utama untuk XML, yang pertama adalah cara untuk mewakili *low-level data*, contohnya adalah file konfigurasi. Yang kedua adalah cara untuk menambahkan *metadata* ke dokumen, misalnya seseorang mungkin ingin menekankan kalimat tertentu dalam sebuah laporan dengan menempatkan dalam huruf miring atau tebal. Penggunaan pertama untuk XML dimaksudkan

sebagai pengganti cara yang lebih tradisional ini telah dilakukan sebelumnya, biasanya dengan cara daftar pasangan nama / nilai seperti yang terlihat di file INI di Windows atau file Properti Java. Penerapan kedua XML mirip dengan bagaimana HTML file bekerja. Teks dokumen yang terkandung dalam wadah keseluruhan, unsur <body>, dengan frase individu dikelilingi oleh tag <i> atau <b> [**24**].