

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Profil Platform**

Bandung Music Council (BMC) didirikan pada Tahun 2016, yang diketuai oleh Erlan Effendy. Bandung Music Council (BMC) yang berlokasi di Gedung IKA UNPAD, Jl. Singa Perbangsa No.1 Lebak Gede, Kecamatan Coblong, Kota Bandung, Jawa Barat. Bandung Music Council (BMC) adalah sebuah platform sebagai wadah bagi seluruh pelaku industri kreatif di Kota Bandung, yang tidak memandang usia atau genre musik. Erlan Effendi sebagai ketua berharap dengan hadirnya wadah bagi para musisi di Kota Bandung dimaksudkan dapat bertukar pikiran dan diharapkan bisa mengakomodir cita-cita yang diinginkan para seniman musik.



*Gambar 2.1 Logo Bandung Music Council (BMC)*

Musisi asal Bandung tersebut resmi mendeklarasikan diri bersatu dalam wadah Bandung Music Council. Deklarasi yang digelar di Prama Grand Preanger (27/10/2016) tersebut juga dihadiri langsung oleh Kepala Badan Ekonomi Kreatif, Triawan Munaf. Bandung Music Council di bawah naungan Yayasan Bandung Musik Indonesia ini menjadi sebuah harapan baru untuk menjaga eksistensi, advokasi dan memperkuat jaringan kreatif industri musik di Kota Bandung.

## **2.2 Landasan Teori**

Landasan teori bertujuan memberikan gambaran dari teori yang terkait dalam pembangunan aplikasi.

### **2.2.1 Musik Tradisional**

Musik dapat didefinisikan sebagai sebuah cetusan ekspresi atau pikiran yang dikeluarkan secara teratur dalam bentuk bunyi. Asal kata musik berasal dari bahasa Yunani yaitu mousike yang diambil dari nama dewa dalam mitologi Yunani kuno yaitu Mousa yakni yang memimpin seni dan ilmu (Ensiklopedi National Indonesia, 1990 : 413). Tradisional berasal dari kata Traditio (Latin) yang berarti kebiasaan yang sifatnya turun temurun. Kata tradisional itu sendiri adalah sifat yang berarti berpegang teguh terhadap kebiasaan yang turun temurun (Salim dan Salim, 1991 : 1636).

Tradisi berasal dari kata tradisi yang berarti sesuatu yang turun temurun (adat, kepercayaan, kebiasaan, ajaran) dari nenek moyang. Dengan kata lain, tradisi adalah kebiasaan yang diwariskan dari satu generasi ke generasi berikutnya secara turun temurun. Dipertegas lagi oleh Esten (1993 : 11) bahwa tradisi adalah kebiasaan turun – temurun sekelompok masyarakat berdasarkan nilai – nilai budaya masyarakat yang bersangkutan. (Ensiklopedi Nasional Indonesia, 1990 : 4141) mendefinisikan tradisi sebagai kebiasaan yang diwariskan dari suatu generasi ke generasi berikutnya secara turun temurun, Kebiasaan yang diwariskan mencakup berbagai nilai budaya, meliputi adat istiadat, sistem kemasyarakatan, sistem pengetahuan, bahasa, kesenian dan sistem kepercayaan.[4]

### **2.2.2 Alat Musik Tradisional Indonesia**

Alat musik tradisional Indonesia adalah instrumen musik yang berasal dari Indonesia dan telah digunakan selama berabad-abad. Musik tradisional berasal dari dua suku kata yaitu musik dan tradisi yang memiliki arti ilmu atau seni menyusun nada atau suara dalam urutan, kombinasi, dan hubungan temporal untuk menghasilkan komposisi (suara) yang mempunyai kesatuan dan kesinambungan. Musik tradisional Nusantara adalah musik yang berkembang di

seluruh wilayah kepulauan dan merupakan kebiasaan turun-temurun yang masih dijalankan dalam masyarakat. Musik ini tersebar hampir di seluruh pelosok negeri dan setiap daerah mempunyai ciri khas yang berbeda. Musik Nusantara lahir, tumbuh, dan berkembang di seluruh wilayah Nusantara. Beberapa contoh alat musik tradisional Indonesia adalah gamelan, angklung, kendang, sasando, dan rebab.[5]

### **2.2.3 Arasemen Dengan Alat Musik Tradisional**

untuk mengaransemen lagu dengan alat musik tradisional Indonesia melibatkan pemahaman tentang unsur-unsur musik tradisional, prinsip-prinsip komposisi, dan karakteristik alat musik tradisional yang digunakan. Berikut adalah beberapa landasan teori yang perlu dipertimbangkan dalam aransemen lagu dengan alat musik tradisional Indonesia:

#### **1. Teori Skala dan Mode Tradisional:**

Skala dan mode tradisional Indonesia, seperti pelog dan slendro, memiliki peran penting dalam menciptakan nuansa musik. Pemahaman yang baik tentang skala-skala ini akan membantu dalam memilih nada-nada yang sesuai untuk alat musik tradisional.

#### **2. Prinsip Melodi dan Harmoni:**

Pemahaman tentang prinsip-prinsip melodi (alur melodi) dan harmoni (paduan nada) dalam musik tradisional adalah kunci dalam menciptakan aransemen yang seimbang. Beberapa alat musik tradisional berperan sebagai alat melodi, sementara yang lain berperan sebagai alat harmoni.

#### **3. Teknik Bermain Alat Musik Tradisional:**

Setiap alat musik tradisional memiliki teknik bermain yang khas. Misalnya, teknik memetik, memukul, atau meniup alat musik tradisional harus dikuasai dengan baik untuk menciptakan permainan yang berkualitas.

#### **4. Pola Ritme dan Tala Tradisional:**

Ritme dan tala (pola irama) adalah bagian penting dari musik tradisional Indonesia. Pemahaman tentang pola-pola ritme yang digunakan dalam alat

musik tradisional seperti kendang, gamelan, atau angklung akan memengaruhi ritme keseluruhan komposisi.

**5. Dinamika dan Ekspresi:**

Dinamika (volume) dan ekspresi (perasaan) dalam permainan alat musik tradisional harus dipahami dan diterapkan dengan tepat untuk menciptakan nuansa dan emosi yang diinginkan dalam lagu.

**6. Kombinasi Alat Musik Tradisional:**

Mempelajari bagaimana alat musik tradisional berinteraksi satu sama lain dalam sebuah ansambel atau ensemble adalah kunci dalam menciptakan aransemen yang harmonis dan seimbang.

**7. Konteks Budaya dan Tradisional:**

Memahami konteks budaya dan tradisional dari alat musik dan lagu yang digunakan penting. Ini melibatkan pengetahuan tentang makna, penggunaan, dan konteks sejarah alat musik dan komposisi tersebut.

**8. Kreativitas dan Inovasi:**

Meskipun memahami tradisi sangat penting, terdapat ruang untuk kreativitas dan inovasi dalam aransemen lagu dengan alat musik tradisional. Beberapa musisi mencoba menggabungkan unsur-unsur modern dengan alat musik tradisional untuk menciptakan suara yang unik.

**9. Penyusunan Notasi:**

Notasi musik tradisional, seperti not balok atau notasi gamelan, mungkin digunakan untuk mendokumentasikan aransemen. Pemahaman notasi ini diperlukan jika Anda ingin mencatat atau berbagi aransemen Anda.

**10. Pemahaman Genre Musik:**

Setiap genre musik tradisional, seperti kroncong, dangdut, gamelan, dan lainnya, memiliki karakteristik khas. Pemahaman genre ini akan membantu Anda mengarahkan aransemen Anda ke dalam konteks yang sesuai.

#### **2.2.4 Loopers**

*Loopers* atau pada musik adalah media yang digunakan oleh para musisi untuk merekam dan memutar secara berulang (*loop*) dari suara yang dimainkan oleh instrumen atau suara vokal. Aplikasi *Loopers* umumnya digunakan untuk

membuat musik secara langsung atau untuk memperluas suara yang dimainkan oleh seorang musisi tunggal. Dan *Loop Stations* merupakan perangkat keras alat musik elektronik yang digunakan seorang musisi untuk merekam dan memutar kembali suara secara terus-menerus atau dalam bentuk *loop*. Alat musik ini biasanya digunakan oleh gitaris, *bassist*, penyanyi, dan pemain *keyboard* untuk membuat variasi musik yang lebih kompleks. Beberapa contoh *Loopers* atau *Loop Stations* yang populer di antaranya adalah Boss RC-300, TC Electronic Ditto X2 Looper, dan Electro-Harmonix 22500 Dual Stereo Looper.[3]

### 2.2.5 Metode Looping

Pada dunia pemrograman Metode looping adalah urutan perintah yang terus-menerus diulang sampai dengan kondisi yang diinginkan tercapai. Kondisi ini memiliki bentuk yang berbeda-beda Seperti contohnya untuk mendapatkan data, mengubah data, dan juga apakah suatu nilai sudah bisa mencapai jumlah yang sudah ditentukan. Singkatnya looping ini merupakan perulangan untuk menghemat waktu dan usaha dalam mengoptimasikan tugas yang berulang.

*Logika Looping* Salah satu jenis dari logika yaitu ada logika looping. Logika ini digunakan untuk bisa menjalankan kode ataupun instruksi secara berulang-ulang sampai dengan kondisi terpenuhi atau apabila kondisi berhenti tercapai. Logika looping adalah salah satu yang paling penting dan dibagi menjadi dua kategori yaitu entry controlled loop dan exit controlled loop.

### 2.2.6 Unified Modeling Language (UML)

*Unified Modeling Language* (UML) merupakan bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML tidak berdasarkan pada bahasa pemrograman tertentu. Standar spesifikasi UML dijadikan standar *defacto* oleh OMG (*Object Management Group*) pada tahun 1997. UML yang berorientasikan obyek mempunyai beberapa notasi standar. Spesifikasi ini menjadi populer dan standar karena sebelum adanya UML, telah ada berbagai macam spesifikasi yang berbeda. Hal ini menyulitkan komunikasi antar pengembang perangkat lunak. Untuk itu beberapa pengembang

spesifikasi yang sangat berpengaruh berkumpul untuk membuat standar baru. UML dirintis oleh Grady Booch, James Rumbaugh pada tahun 1994 dan kemudian Ivar Jacobson.

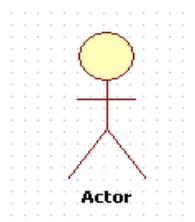
Seperti yang telah dipaparkan di atas, UML yang merupakan turunan dan beberapa metode mempunyai kumpulan diagram *grafts* sebagai kombinasi dari konsep pemodelan data (*Entity Relationship Diagram*), pemodelan bisnis (*work flow*), pemodelan obyek, dan pemodelan komponen. Diagram *grafts* tersebut merupakan tampilan dari beberapa level abstraksi yang dapat digunakan secara bersama oleh semua proses pada seluruh *lifecycle* pembangunan perangkat lunak serta pada implementasi kebeberapa teknologi yang berbeda. UML memiliki berbagai jenis diagram, diantaranya adalah Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram:

### 2.2.7 Use Case Diagram

*Use Case* diagram berisi mengenai interaksi antara sekelompok proses dengan sekelompok *actor*, menggambarkan fungsionalitas dari sebuah sistem yang dibangun dan bagaimana sistem berinteraksi dengan dunia luar. *Use Case diagram* dapat digunakan selama proses analisis untuk menangkap kebutuhan sistem dan untuk memahami bagaimana sistem seharusnya bekerja. *Use Case* diagram terdiri dari beberapa elemen pemodelan utama, yaitu *Actor*, *Use Case*, *Association*, *Dependency*, dan *Generalization*.

#### 1. Actor

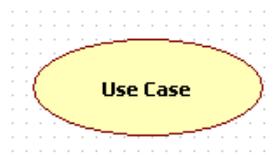
Pada dasarnya *actor* bukanlah bagian dari *use case* diagram, namun untuk dapat terciptanya suatu *use case* diagram diperlukan beberapa *actor*. *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan stick man. Pada gambar berikut menunjukkan Simbol *Actor* pada *Use Case* Diagram:



**Gambar 2.2 Aktor Use case**

## 2. UseCase

*Use case* adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. Pada gambar berikut merupakan Simbol *Use Case* pada *Use Case Diagram*:



**Gambar 2.3 Use Case**

## 3. Association

*Association* menghubungkan *link* antar *element*, dan bukan menggambarkan aliran data / informasi pada sistem. *Association* digunakan untuk menggambarkan bagaimana aktor terlibat dalam *usecase*. Ada 4 jenis relasi yang bisa timbul pada *usecase* diagram, yaitu *Association* antara aktor dan *usecase*, *Association* antara *usecase*, *Generalization/Inheritance* antara *usecase*, *Generalization/Inheritance* antara aktors.

## 4. Dependency

*Dependency* adalah sebuah *element* bergantung dalam beberapa cara ke *element* lainnya.

## 5. Generalization

*Generalization* disebut juga *inheritance* (pewarisan), sebuah *element* dapat merupakan spesialisasi dari *element* lainnya

### 2.2.8 Activity Diagram

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram

aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

### **2.2.9 Sequence Diagram**

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang disusun dalam suatu urutan waktu.

Secara khusus, diagram ini berasosiasi dengan *use case*. *Sequence* diagram menggambarkan *behavior* internal sebuah sistem. Dan lebih menekankan pada penyampaian *message* dengan parameter waktu

### **2.2.10 Class Diagram**

*Class* diagram adalah diagram yang menunjukkan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika dan menggambarkan struktur statis dari sebuah sistem. *Class* diagram digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang digunakan. *Class* diagram memberikan gambaran (diagram statis) tentang sistem/perangkat lunak dan relas-relasi yang ada didalamnya. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan yaitu bagian atas adalah bagian nama dari *class*, bagian tengah mendefinisikan *property/atribut class*, bagian akhir mendefinisikan *method-method* dari sebuah *class*. *Class* sebaiknya diberi nama menggunakan kata benda sesuai dengan domain atau bagian atau kelompoknya.

### **2.2.11 Adobe Photoshop dan Adobe Illustrator**

Adobe Photoshop dan Illustrator adalah salah satu perangkat lunak grafis. Photoshop umumnya digunakan untuk mengedit dan memanipulasi gambar, foto, ataupun grafik, sementara Illustrator umumnya digunakan untuk membuat grafik vektor, seperti logo, ikon, dan ilustrasi. Keduanya merupakan produk yang dikembangkan oleh Adobe Inc.

Adobe Photoshop memiliki fitur seperti layers, filter, dan tools seleksi yang memungkinkan pengguna untuk melakukan editing dan manipulasi pada setiap

elemen dalam gambar. Sementara itu, Adobe Illustrator menggunakan path dan bentuk matematika sebagai dasar pembuatan grafik, sehingga grafik yang dibuat dengan Illustrator bisa diubah ukuran tanpa kehilangan kualitas.

Kedua perangkat lunak ini sangat populer di kalangan desainer grafis, fotografer, ilustrator, dan seniman digital karena memiliki fitur-fitur yang lengkap dan dapat membantu untuk menciptakan hasil yang profesional dalam proses pengolahan grafis.

### **2.2.12 Unity 3D**

Aplikasi unity 3D adalah game engine merupakan sebuah software pengolah gambar, grafik, suara, input, dan lain-lain yang ditujukan untuk membuat suatu game, meskipun tidak selamanya harus untuk game. Contohnya adalah seperti materi pembelajaran untuk simulasi membuat SIM. Kelebihan dari game engine ini adalah bisa membuat game berbasis 3D maupun 2D, dan sangat mudah digunakan.

Unity merupakan game engine yang ber-multiplatform. Unity mampu di publish menjadi Standalone (.exe), berbasis web, berbasis web, Android, iOS Iphone, XBOX, dan PS3. Walau bisa dipublish ke berbagai platform, Unity perlu lisensi untuk dapat dipublish ke platform tertentu. Tetapi Unity menyediakan untuk free user dan bisa di publish dalam bentuk Standalone (.exe) dan web. Untuk saat ini Unity sedang di kembangkan berbasis AR (Augment Reality). Untuk mengaktifkan lisensi, Unity perlu adanya lisensi. Sebagai contoh ketika ingin mengaktifkan free user, langkah pertama adalah mendownload softwarena secara gratis pada web [www.unity3d.com](http://www.unity3d.com).

Setelah selesai instalasi, maka Unity meminta untuk 5 terhubung dengan internet untuk aktifasinya. Lalu selesai unity akan otomatis run ke program. Untuk langkah selanjutnya Unity tidak perlu lagi memerlukan koneksi internet saat menjalankan aplikasinya. Unity bukan software game engine baru, sehingga banyak tutorial yang tersebar luas, bahkan banyak game dan tutorial untuk Unity yang telah beredar. Unity cepat berkembang dikarenakan bisa free user dan banyak di implementasikan ke berbagai platform disamping banyaknya tutorial yang bisa dengan mudah dicari. Dengan Unity3D kita dapat membuat game 3D,

FPS dan 2D game bahkan Game Online, fitur lain tentang Unity berikut selengkapnya : 1. Membuat Game 2D / 3D. 2. Membuat Game FPS, Simulasi dan Aplikasi Augmented Reality. 3. Membuat Game Online. 4. Dukungan Konversi : Mobile Android, Iphone, Blackberry, Windows, Linux, Flash, Webplayer. 5. Online Publish Google Play, Android market. 6. Dukungan kode : C#, Javascript dan Boo. 7. Dukungan Extensi file, 3ds, obj, fbx.

### **2.2.13 Bahasa Pemrograman C#**

Bahasa pemrograman C# adalah bahasa pemrograman modern yang banyak digunakan untuk membuat aplikasi dan game. Bahasa pemrograman C# sering digunakan dengan platform .NET, dan merupakan bahasa pemrograman yang kuat dan mudah dipelajari. Bahasa pemrograman C# awalnya dikembangkan oleh Microsoft, dan sekarang telah menjadi bahasa pemrograman yang populer di seluruh dunia.