



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Metode Penelitian

Pada penelitian ini, dibuat sebuah aplikasi untuk pengenalan bentuk *chord* dan melodi pada alat musik gitar dengan teknologi berbasis *Augmented Reality*. Aplikasi ini bertujuan untuk memudahkan dan menambah daya tarik pengguna dalam mempelajari berbagai macam bentuk dari *chord* baik mayor maupun minor dan melodi pada alat musik gitar. Penelitian ini menggunakan metode *marker based system*, di mana pengguna dapat menggunakan kamera *smartphone* untuk mendeteksi *marker* yang telah diregistrasi, kemudian kamera mengidentifikasi koordinat *marker* dan melakukan *rendering* objek *virtual 3D chord* atau melodi yang dipilih untuk ditampilkan di atas *marker*. Pengguna juga bisa mendengarkan suara dari objek yang ditampilkan dengan menekan sebuah tombol yang muncul di atas *marker* dan tersedia pada layar *smartphone*.

Berikut merupakan tahapan-tahapan yang dilakukan dalam penelitian:

- a. Melakukan perancangan objek *virtual 3D* dari bentuk-bentuk *chord* dan melodi yang digunakan sebagai tampilan *Augmented Reality* aplikasi. Setiap bentuk objek *chord* dan melodi ini mencakup tujuh *chord* mayor yaitu bentuk A, B, C, D, E, F, G dan tujuh *chord* minor yaitu bentuk Am, Bm, Cm, Dm, Em, Fm Gm, dan tiga melodi yaitu bentuk A, C dan F. Setiap objek 3D ini dirancang menggunakan *Autodesk 3Ds Max* dan diimplementasikan menggunakan *Prefab ARCamera* dan *ImageTarget*.

- b. Melakukan perancangan suara untuk objek *chord* dan melodi. Perancangan ini diawali dengan me-*record* suara untuk setiap bentuk *chord* dan melodi menggunakan gitar, kemudian mengimplementasikan hasil *recording* pada setiap objek *chord* dan melodi.
- c. Melakukan perancangan *marker* yang digunakan untuk menampilkan bentuk-bentuk objek *chord* dan melodi. Proses perancangan ini dilakukan menggunakan *Adobe Photoshop CS6* untuk membuat gambar *marker* dengan pola yang unik atau abstrak.
- d. Melakukan perancangan tampilan antarmuka. Pada tampilan awal terdapat tombol-tombol yang digunakan untuk berinteraksi dengan aplikasi, serta halaman untuk melihat cara penggunaan aplikasi dan disediakan tombol apabila pengguna menutup aplikasi. Kemudian merancang beberapa antarmuka lainnya sesuai dengan kebutuhan aplikasi.
- e. Melakukan pengujian untuk setiap *GameObject*, modul-modul serta antar muka yang digunakan dalam menginterpretasikan setiap kebutuhan sistem dalam penelitian ini.
- f. Melakukan *building* apk pada *Unity 3D* dengan menggunakan *Android Studio* untuk membuat aplikasi menjadi *installer* atau apk, sehingga dapat dipasang pada *smartphone*.
- g. Melakukan pengujian instalasi untuk memastikan aplikasi bisa terpasang pada *smartphone*
- h. Melakukan pengujian akhir untuk memastikan fungsi-fungsi dari aplikasi dapat berjalan dengan baik dengan menggunakan *smartphone*.

3.2 Perancangan dan Pembuatan Aplikasi

3.2.1 Perangkat Penelitian

Berikut ini merupakan perangkat – perangkat yang digunakan dalam perancangan aplikasi yang meliputi perangkat keras dan perangkat lunak.

- a. Laptop dengan sistem operasi *Windows 7* yang digunakan untuk pengembangan serta pengujian aplikasi dengan spesifikasi seperti yang ada pada tabel 3.1.
- b. Sebuah *smartphone* dengan sistem operasi *Android Jelly Bean 4.1* yang digunakan dalam pemasangan aplikasi untuk menampilkan objek *chord* dan melodi di atas *marker* dengan memanfaatkan kamera dari *smartphone* tersebut.
- c. *Unity 3D* dengan *MonoDevelop* sebagai *script editor* yang berfungsi sebagai *compiler* untuk melakukan pembuatan aplikasi, mulai dari objek, suara dan antar muka. Pengembangan aplikasi menggunakan bahasa pemrograman *C#*.
- d. *Vuforia SDK* digunakan sebagai *library Augmented Reality*.
- e. *Marker* merupakan gambar (*image*) yang terintegrasi dengan *Vuforia SDK* untuk menampilkan objek 3D *chord* maupun melodi pada saat adanya proses *tracking* dari kamera *smartphone*.
- f. *Autodesk 3Ds Max* dipakai untuk pemodelan objek 3d *chord* dan melodi gitar.
- g. *Android Studio* dipakai dalam pengembangan *system* aplikasi sebagai SDK untuk mem-*build* *Apk Android* dari *Unity 3D*.

Processor	Intel i3-2330U
Memory	2 GB
Hard disk	500 GB
Operating System	Windows 7 64-bit

Tabel 3.1 Tabel Spesifikasi Laptop

3.2.2 Perancangan Aplikasi

Perancangan dan pengembangan aplikasi dimulai dengan merancang *marker* yang digunakan untuk menampilkan objek *virtual* 3D dari setiap bentuk *chord* dan melodi, kemudian merancang model objek dari setiap bentuk *chord* dan melodi serta merancang antar muka yang digunakan. Semua kebutuhan aplikasi itu dirancang, dikodekan dan diuji secara terpisah kemudian disatukan dalam satu kesatuan sampai mendapatkan hasil yang tepat. Berikut komponen-komponen yang dirancang pada penelitian ini:

a Perancangan Objek 3D dan Suara

Proses awal perancangan bentuk-bentuk *chord* dan melodi adalah mendesain objek 3D menggunakan *Autodesk 3DS Max*. Kemudian, pada *Unity 3D*, dibuat sebanyak tujuh belas *scene* terdiri dari tujuh *scene chord* mayor, tujuh *scene chord* minor dan tiga *scene* melodi dengan bentuk A, C dan F. *Scene* sendiri merupakan *level-level game* atau tempat di dalam *3D space* yang digunakan untuk menampung *game object* dalam membuat aplikasi pada *Unity 3D*[21]. Jadi, pada penelitian ini *scene* digunakan untuk menampung setiap *GameObject* yang dibutuhkan untuk perancangan aplikasi.

1. Prefab ARCamera

Prefab adalah bentuk instansiasi dari *GameObject* yang disimpan dengan pengaturan-pengaturan tertentu untuk digunakan pada setiap *scene* yang berbeda tanpa melakukan pengaturan dari awal lagi dengan siklus perancangannya yaitu *add component*, *coding* dan *testing* [21][22]. Prefab *ARCamera* merupakan *GameObject default* yang disediakan oleh *Vuforia SDK* yang digunakan untuk melakukan semua tugas yang berkaitan dengan mengakses data kamera, *me-render*

gambar di latar belakang dan melakukan proses *tracking* objek. *Prefab ARCamera* dapat dijalankan dengan menambahkan *license key Vuforia*.

Berikut modul-modul yang digunakan beserta fungsionalitasnya pada *Prefab ARCamera*:

- Modul *VuforiaBehaviour* merupakan sebuah *class* yang disediakan oleh *Vuforia* SDK yang mempunyai tugas untuk mengambil data kamera, *handle tracking* dan melakukan *rendering* untuk menampilkannya di latar belakang.
- Modul *DefaultInitializationErrorHandler* bertugas menangani kesalahan-kesalahan dan menampilkan dialog pada layar untuk memberitahu pengembang bahwa telah terjadi kesalahan inisialisasi. Salah satu identifikasi yang dilakukan adalah mengecek valid atau tidak *license key* yang digunakan.
- Modul *DatabaseLoadBehaviour* mengaktifkan akses *DataSet* secara otomatis. Di mana *DataSet* bertugas untuk mengenali objek *tracking* yang sudah diregistrasi dan di simpan dengan *file XML*. Cara kerjanya adalah mencocokkan parameter objek *tracking* dengan data yang ada pada kamera seperti nama, posisi dan orientasi objek tersebut.

2. *Prefab ImageTarget*

Sama seperti *Prefab ARCamera*, *ImageTarget* juga merupakan *default prefab* yang disediakan *Vuforia* SDK. Pada penelitian ini, *Prefab ImageTarget* digunakan untuk menentukan koordinat objek, *marker* dan *sound*. Setiap bentuk objek *chord* dan melodi beserta fungsi masing-masing dirancang pada *ImageTarget*. *GameObject* ini mengakses modul *ImageTargetBehaviour* untuk melakukan *tracking* hasil *image target* secara *runtime*.

Berikut merupakan *GameObject* beserta fungsi masing-masing yang dirancang pada *Prefab ImageTarget*:

- *GameObject Gitar*

GameObject gitar dirancang untuk merepresentasikan objek *chord* dan melodi gitar. Perancangan dilakukan dengan melakukan *import GameObject chord* dan melodi, kemudian mengatur posisi, rotasi dan skala di atas *marker*.

- *GameObject Sound dan Click*

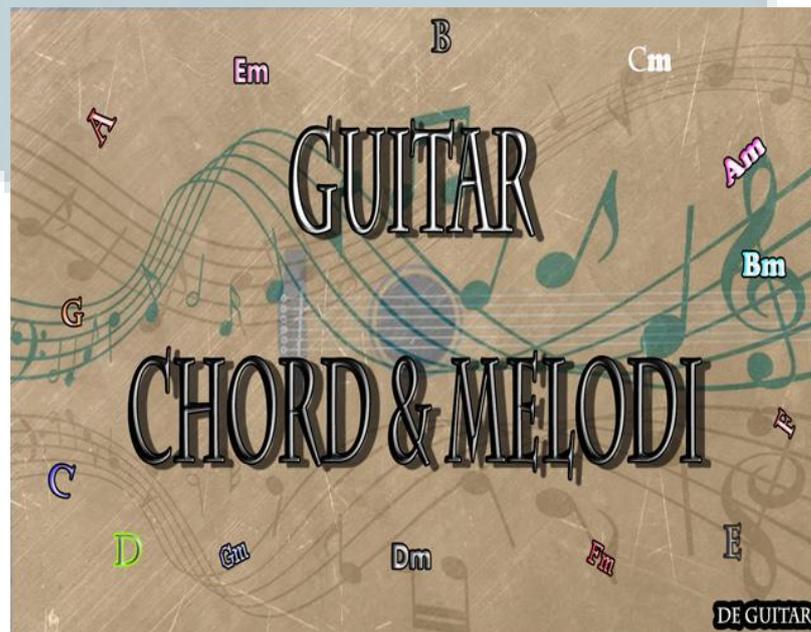
GameObject Sound dan *Click* merupakan dua *GameObject* yang dirancang untuk merepresentasikan suara *chord* dan melodi. *Click* digunakan untuk menampilkan *texture* 3D tombol “*Play Sound*” yang bisa disentuh oleh pengguna, sedangkan *Sound* mempunyai tugas untuk mengakses *source file* suara. Tombol ini akan ditampilkan pada saat *class DataSet* melakukan *tracking marker*, sehingga akan dimunculkan pada saat yang sama dengan objek *chord* dan melodi. Ketika *marker* disentuh pada koordinat bidang tombol “*Play Sound*”, maka modul *VirtualButtonEventHandler* akan menangani *event* sentuhan tangan tersebut untuk mengidentifikasi koordinat sentuhan dengan koordinat yang teregistrasi. Selanjutnya, modul *VirtualButtonBehaviour* akan mengakses *public function* dan menangkap *action* yang diberikan oleh modul kemudian melakukan *update action* untuk memunculkan suara *chord* atau melodi yang dipilih oleh pengguna.

b Perancangan *Marker*

Pada perancangan *marker* ini, diawali dengan menentukan pola gambar yang akan dipakai untuk dijadikan sebuah *marker* yang nantinya berfungsi untuk menampilkan objek 3D yang teridentifikasi koordinatnya melalui kamera *smartphone* yang terintegrasi dengan *Vuforia SDK*, di mana *marker* yang terdeteksi

oleh kamera kemudian diubah menjadi *binary* dan kamera melakukan identifikasi posisi koordinat dan mengalkulasikan apakah pola *marker* tersebut sesuai dengan yang sudah diregistrasi dengan *template*. Penentuan pola *marker* dilakukan dengan cara mendesain sebuah gambar yang unik atau abstrak menggunakan *Adobe Photoshop CS6*. Kemudian *marker* yang sudah selesai didesain, di-*upload* ke *database Vuforia SDK* untuk didaftarkan ke *library*, sehingga *marker* tersebut bisa dipakai untuk menampilkan objek 3D dari semua bentuk *chord* dan melodi pada alat musik gitar yang telah ditentukan.

Pada gambar 3.2 menampilkan desain dari gambar yang dijadikan *marker*.



Gambar 3.1 *Marker Chord dan Melodi Gitar*

c Perancangan Antarmuka

1. Antarmuka Halaman Awal

Perancangan antarmuka diawali dengan membuat halaman utama yang akan menjadi tampilan awal saat aplikasi dijalankan. Pada halaman awal ini dirancang sebuah *scene MainMenu_1* yang digunakan untuk menampung tiga buah tombol yaitu tombol “PLAY”, “TUTORIAL” dan “EXIT” di mana tombol-tombol tersebut diakses melalui modul *MainMenu_1*. Tombol “PLAY” diletakan pada bagian *top center* untuk mempermudah pengguna mengakses *menu* berikutnya apabila tidak perlu lagi melihat cara pemakaian aplikasi. Selanjutnya, tombol “TUTORIAL” diletakan tepat di bawah tombol “PLAY” untuk mempermudah pengguna dalam mengakses *menu* pemakaian aplikasi, dan yang terakhir tombol “EXIT” diletakan pada *bottom center* yang berfungsi untuk menutup aplikasi.

Selanjutnya dirancang sebuah *background* dengan warna dasar hijau. Hijau adalah warna yang tenang dan santai, di mana warna ini dapat membantu orang yang sering merasa tegang dalam belajar meningkatkan kemampuan membaca siswa. Para siswa yang membaca materi tulisan di atas lembaran hijau transparan akan meningkatkan kecepatan membaca dan pemahaman. Di dalam desain aplikasi, warna hijau dapat digunakan untuk memberikan kesan segar, sehingga dapat membuat proses penggunaan aplikasi dapat berjalan dengan menarik dan cepat dimengerti [23].

Selanjutnya merancang warna teks pada setiap tombol, di mana warna yang digunakan adalah putih. Warna putih sendiri memberikan kesan yang simpel

dan minimalis sehingga akan memberikan efek *relax* pada pengguna dalam menggunakan aplikasi[23].

Warna *background* dan tombol yang dirancang pada *menu* halaman awal, digunakan juga pada semua menu aplikasi ini. Pada gambar 3.3 menampilkan perancangan halaman awal aplikasi.

2. Antarmuka *Tutorial*

Perancangan halaman tutorial berfungsi untuk memberikan tampilan cara penggunaan aplikasi. Sebuah *scene Tutorial* dirancang pada halaman ini dengan menampilkan sebuah *image* yang telah didesain menggunakan *Adobe Photoshop CS6*, di mana *image* tersebut memuat langkah-langkah penggunaan aplikasi.

Halaman ini dapat diakses melalui tombol "*Tutorial*" yang ada pada halaman awal dengan menggunakan modul *Tutorial*. Terdapat juga tombol "*BACK*" yang diletakkan pada *bottom center* dari antarmuka untuk mengakses kembali halaman awal. Tombol "*BACK*" ini diakses dengan modul "*BackTutorial*".

3. Antarmuka *Menu Play*

Proses perancangan antarmuka *menu play* dilakukan dengan membuat empat buah tombol yaitu tombol "*CHORD-MAYOR*" yang diletakkan pada *top center* yang berfungsi untuk mengakses pilihan *chord* mayor yang akan dimainkan, tombol "*CHORD-MINOR*" yang diletakkan tepat di bawah tombol sebelumnya dan mempunyai fungsi untuk mengakses *menu* pilihan *chord*

minor, tombol “*MELODY*” yang diletakan juga tepat di bawah tombol sebelumnya yang mempunyai fungsi untuk mengakses *menu* pilihan melodi dan tombol “*BACK*” yang diletakkan pada posisi *bottom center*, mempunyai fungsi untuk memberikan akses kembali ke *menu* sebelumnya. Terdapat sebuah *scene MainMenu_2* yang digunakan untuk menampung *background* dan tombol-tombol tersebut. Kemudian setiap fungsi dari tombol-tombol tersebut dirancang pada modul *MainMenu_2*.

4. Antarmuka *Menu* Pilihan *Chord* Mayor

Menu ini berfungsi untuk menampilkan pilihan tujuh bentuk *chord* mayor yang akan di-*play*, di mana *scene MenuMayor* digunakan untuk menampung *image background* dan *image* dari setiap *chord* mayor. Tombol “*Click Here To Play*” dan tombol “*scroll*” diletakan tepat di bawah *image chord* mayor, untuk mempermudah pengguna dalam menentukan pilihan *chord* mayor yang akan di-*play*. Fungsionalitas dari tombol dan *image* tersebut dirancang pada modul *MenuMayor*.

5. Antarmuka *Menu* Pilihan *Chord* Minor

Menu ini berfungsi untuk menampilkan pilihan tujuh bentuk *chord* minor yang akan di-*play*, di mana *scene MenuMinor* digunakan untuk menampung *image background* dan *image* dari setiap *chord* minor. Tombol “*Click Here To Play*” dan tombol “*scroll*” diletakan tepat di bawah *image chord* minor, untuk mempermudah pengguna dalam menentukan pilihan *chord* minor yang akan

di-play. Fungsionalitas dari tombol dan *image* tersebut dirancang pada modul *MenuMinor*.

6. Antarmuka *Menu* Pilihan Melodi

Menu ini berfungsi untuk menampilkan pilihan tiga bentuk melodi yang akan di-play, di mana *scene MenuMelodi* digunakan untuk menampilkan *image background* dan *image* dari setiap *chord* mayor. Tombol “*Click Here To Play*” dan tombol “*scroll*” diletakkan tepat di bawah *image* melodi untuk mempermudah pengguna dalam menentukan pilihan bentuk melodi yang akan di-play. Fungsionalitas dari tombol dan *image* tersebut dirancang pada modul *MenuMelodi*.

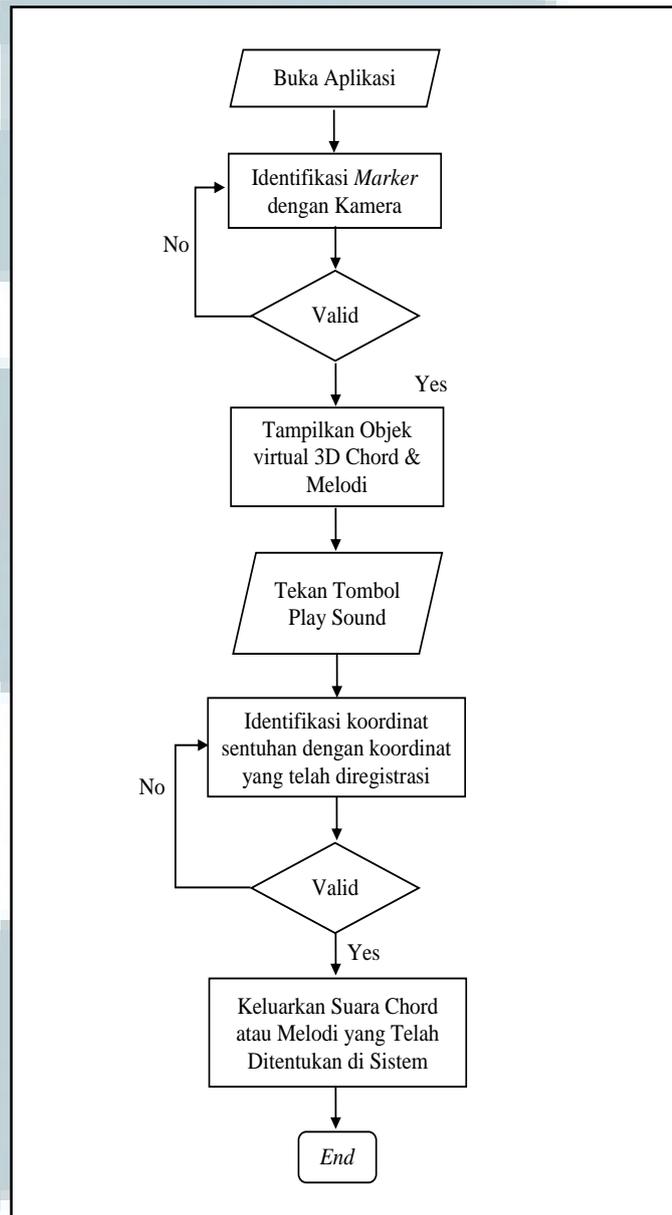
7. Antarmuka *Augmented Reality Chord* dan Melodi Gitar

Terdapat sebuah objek *virtual 3D* gitar *full body* lengkap dengan bentuk *chord* dan melodi yang terletak pada bagian *neck* gitar ini, ditampilkan pada saat pengguna menekan tombol “*Click Here To Play*” untuk mengaktifkan kamera dan mengarahkan ke sebuah *marker* yang telah diregistrasi. Pada *Menu* ini juga terdapat sebuah tombol *virtual* “*PLAY SOUND*” yang dapat digunakan untuk memunculkan suara dari bentuk *chord* atau melodi yang ditampilkan.

Terdapat juga tombol “*BACK*” yang terletak pada posisi *bottom center* tepat di bawah tombol “*PLAY SOUND*” yang berfungsi mengakses *menu* pilihan *chord* dan melodi sebelumnya. Objek *Virtual 3D chord* dan melodi ditampilkan pada setiap *scene* mayor, minor dan melodi. fungsionalitas dari tombol “*BACK*” dirancang pada modul *BackMajor*, *BackMinor* dan *BackMelodi*.

3.2.3 Alur Kerja Aplikasi

Aplikasi *chord* dan melodi gitar berbasis *Augmented Reality* dikendalikan oleh pengguna dengan cara menyiapkan *marker* yang telah diregistrasi untuk menjalankan aplikasi. Pada gambar 3.2 menampilkan *flowchart* dari aplikasi.



Gambar 3.2 *Flowchart* Aplikasi

Berdasarkan *flowchart* aplikasi yang ditampilkan pada gambar 3.4, proses kerja sistem diawali dengan pengguna membuka aplikasi *Augmented Reality chord* dan melodi gitar yang telah terpasang pada *smartphone Android*. Selanjutnya, mengaktifkan kamera *smartphone* dan mengarahkannya pada sebuah *marker* yang telah diregistrasi dengan *Vuforia SDK*, sehingga sistem akan mengidentifikasi koordinat *marker* untuk menyesuaikan posisi orientasi obyek yang ditangkap kamera. Apabila koordinat telah sesuai dengan yang telah ditentukan, maka sistem akan menampilkan bentuk obyek virtual 3D *chord* dan melodi gitar, sebaliknya jika tidak sesuai, maka tidak akan ada objek *virtual 3D* yang akan ditampilkan oleh sistem.

Sebuah tombol *virtual 3D* “*Play Sound*” akan ditampilkan dalam waktu yang sama dengan obyek *virtual 3D chord* dan melodi. Tombol ini dapat disentuh oleh pengguna untuk memunculkan bentuk suara dari *chord* dan melodi gitar. Setelah tombol tersebut disentuh, sistem akan mengidentifikasi dan koordinat sentuhan dengan koordinat yang telah diregistrasi. Apabila hasil yang didapat sesuai, maka sistem akan memunculkan suara, tetapi sebaliknya apabila tidak sesuai dengan koordinat yang telah diregistrasi, maka sistem tidak akan memunculkan suara.