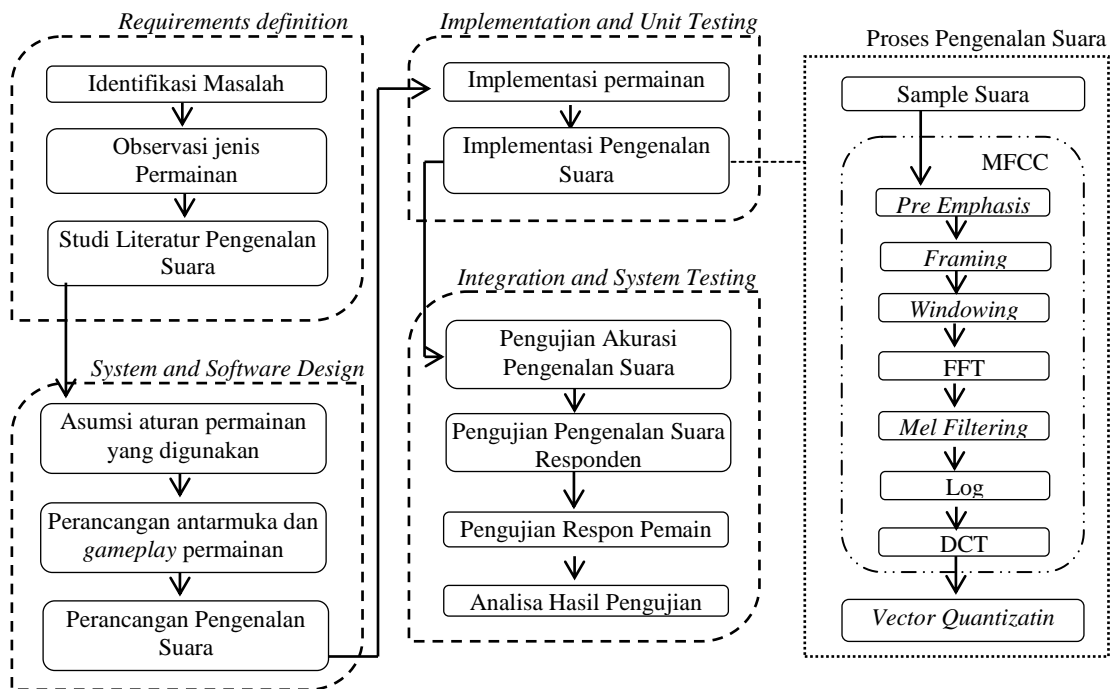


## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Desain Penelitian



Gambar 3.1. Desain Penelitian

Desain penelitian yang dibuat menggambarkan tahapan yang dilakukan oleh peneliti. Desain tersebut mengacu pada model pengembangan perangkat lunak *waterfall* Ian Sommerville. Namun peneliti membatasi tahapan hanya sampai *integration and system testing*, mengingat tahapan *operation and maintenance* tidak relevan dilakukan pada penelitian kali ini.

##### 3.1.1 Requirements Definition

Tahap ini merupakan tahap awal dimana peneliti mengidentifikasi masalah mengenai permainan yang akan dibuat dan pengenalan suara. Identifikasi masalah lebih jelas disajikan di BAB I. Selanjutnya dilakukan observasi mengenai permainan

dengan jenis serupa yang akan dibuat. Jenis-jenis permainan serupa yang diobservasi dapat dilihat di BAB II. Selanjutnya dicari landasan teori yang menunjang untuk mengimplementasikan pengenalan suara pada permainan yang akan dibuat.

### 3.1.2 *System and Software Design*

Pada tahap ini, peneliti menggambarkan kebutuhan permainan yang akan dibuat. Kebutuhan tersebut bisa berupa rancangan antar muka, rancangan *game play* ataupun rancangan kode program. Adapun secara umum permainan yang akan dibuat bergenre *arcade*, dengan *gameplay* terdapat sebuah pesawat luar angkasa yang sedang melakukan perjalanan. Dalam perjalanan tersebut terdapat meteor yang akan menghancurkan pesawat tersebut. Pemain harus menggerakkan pesawat ke kanan atau ke kiri dengan cara mengucapkan “kanan” atau “kiri” untuk menghindari meteor tersebut. Semakin banyak meteor yang dihindari, semakin tinggi nilai yang didapatkan. Terdapat asumsi-asumsi peneliti dalam mengembangkan permainan ini karena disesuaikan dengan kebutuhan penelitian.

### 3.1.3 *Implementation and Unit Testing*

Pada tahapan ini, awalnya peneliti membuat permainan sesuai dengan *gameplay* dan asumsi yang telah dibuat pada tahap sebelumnya. Kemudian dibuat proses pengenalan suara secara terpisah dari permainan yang dibuat. Merujuk pada BAB II gambar 2.4, proses pengenalan suara pada umumnya terdapat 4 tahapan yaitu pengambilan *sample* suara, *denoising*, ekstraksi ciri kemudian pencocokkan pola. Namun pada penelitian kali ini tahapan yang digunakan hanya 3 yaitu pengambilan *sample* suara, ekstraksi ciri dan pencocokkan pola. Proses *denoising* atau pengurangan gangguan (*noise*) tidak diterapkan pada penelitian kali ini karena dirasa perlu dilakukan penelitian lagi untuk penerapannya. Proses penelitian *denoising* tersebut mungkin bisa menganalisa seberapa baik proses penghilangan gangguan dalam *sample* suara yang diambil. Proses ekstraksi ciri dalam penelitian ini menggunakan metode *Mel Frequency Ceptral Coefficient* (MFCC). Adapun untuk

proses pencocokkan polanya, metode yang digunakan adalah *Vector Quantization*. Untuk lebih jelasnya metode MFCC dan *Vector Quantization* telah dipaparkan di BAB II. Proses pengenalan suara, diambil referensi dari program OCVolume – *Java Speech Recognition Engine*. Program tersebut merupakan program *open source* yang dapat dijadikan referensi dalam penelitian.

### 3.1.4 Integration and System Testing

Pada tahapan ini peneliti menggabungkan unit-unit program yang telah dibuat yaitu permainan dengan pengenalan suaranya. Kemudian dilakukan pengujian program dengan *sample* tertentu. Setelah itu dihitung persentase keberhasilan pengenalan suara yang diterapkan dalam permainan tersebut. Selain itu dilakukan observasi dengan kuisioner terhadap pemain untuk mendapatkan respon dari permainan tersebut. Persentase keberhasilan dan respon pemain tersebut menjadi kesimpulan dalam penelitian kali ini.

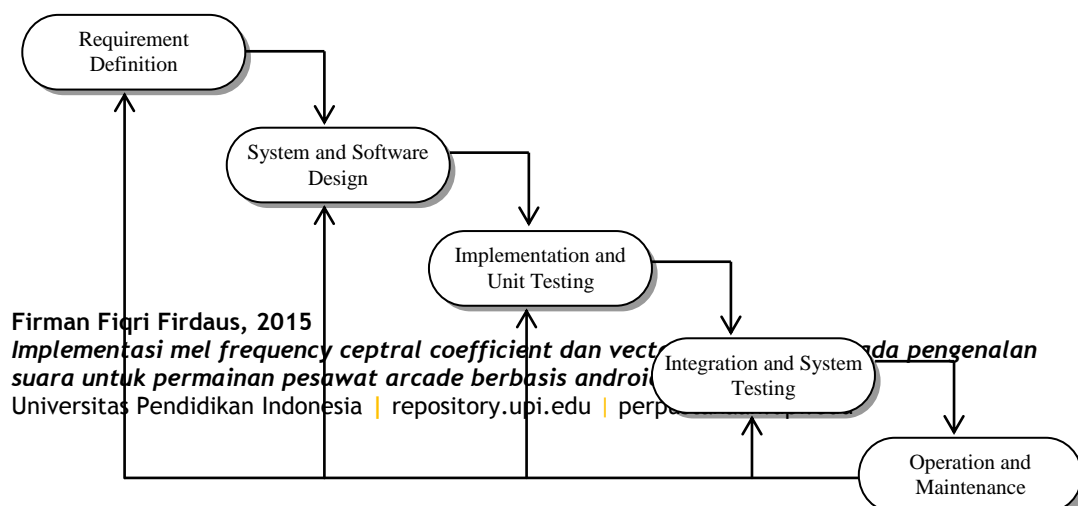
## 3.2. Metode Penelitian

### 3.2.1. Metode Pengumpulan Data

Dalam tahap ini, peneliti melakukan studi literatur mengenai ekstraksi ciri dengan MFCC, *Vector quantization*, dan teori lain yang dibutuhkan dalam penelitian ini. Adapun sumber yang diambil didapatkan dari jurnal, karya tulis, buku, *e-book*, artikel, atau tulisan lain yang dapat dijadikan sebagai acuan.

### 3.2.2. Metode Pembangunan Perangkat Lunak

Pembangunan perangkat lunak dalam penelitian ini menggunakan model *waterfall* (Sommerville, 2011), dimana alur prosesnya adalah sebagai berikut :



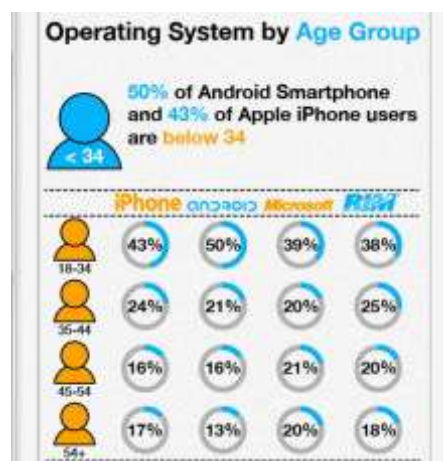
Gambar 3.2. Model *Waterfall*

Model *waterfall* seperti yang digambarkan pada gambar 3.1 terdapat beberapa proses yang dilalui. Dalam model *waterfall* Sommerville terdapat kemungkinan untuk kembali ke tahap sebelumnya apabila terjadi kesalahan atau perbaikan.

- a. *Requirement Definition*, Tahap awal dimana adanya analisis untuk menentukan kebutuhan, batasan, dan tujuan (*goal*) dari perangkat lunak sesuai yang diinginkan. Hal tersebut kemudian didefinisikan secara rinci dan terbentuk sebagai spesifikasi sistem.
- b. *System and Software Design*, *System design* merupakan proses perancangan perangkat keras maupun perangkat lunak yang dilibatkan untuk menunjang sistem yang akan dibangun. Sementara itu *software design* merupakan proses perancangan yang melibatkan identifikasi dan menggambarkan dasar sistem serta hubungan satu sama lain.
- c. *Implementation and Unit Testing*, Pada tahap ini, *software design* yang telah dilakukan sebelumnya kemudian diimplementasikan dalam bentuk unit program. Setelah unit program dibuat, kemudian dilakukan *testing* pada unit program tersebut untuk memastikan implementasi berjalan dengan baik.
- d. *Integration and System Testing*, Setelah semua unit program berhasil diimplementasikan dan lolos *testing* maka dilanjutkan dengan mengintegrasikan setiap unit untuk membentuk sistem yang diinginkan. Sistem yang sudah dibentuk kemudian di tes kembali untuk memastikan unit program dapat berjalan satu sama lain dalam sistem dan sistem yang dibuat sudah memenuhi kebutuhan.
- e. *Operation and Maintenance*, Tahap ini merupakan tahap dimana sistem sudah dipasang kemudian melakukan perbaikan ketika terdapat kesalahan atau *error* yang tidak ditemukan sebelumnya saat pembangunan sistem berlangsung. Perbaikan juga dilakukan jika terdapat kebutuhan baru yang perlu ada pada sistem.

### 3.2.3. Metode Analisa Data

Pengujian pertama kali dilakukan oleh peneliti sendiri. Pengujian ini ditujukan untuk menghitung akurasi dari pengenalan suara yang dibuat. Tahapan yang dilakukan dalam pengujian ini adalah dengan mengucapkan kata “kanan”, “kiri” dan “diam” masing-masing sebanyak 20 kali, kemudian dihitung akurasi pengenalan suaranya. Setelah itu dilakukan pengujian oleh responden. Proses ini bertujuan untuk mengetahui berapa tingkat akurasi pengenalan kata selain dengan suara peneliti dan juga mengetahui respon atau penilaiannya untuk permainan tersebut. Jumlah orang yang jadi responden (*sample*) apabila dalam penelitian sederhana dengan kontrol yang mendalam seperti pencocokan dapat digunakan antara 10 – 20 *sample* (Roscoe, 1975). Merujuk pada pernyataan sebelumnya, peneliti mengambil jumlah *sample* sebanyak 20 orang. Rentang umur yang dijadikan *sample* adalah 18-34 tahun. Rentang tersebut diambil berdasarkan data pengguna *smartphone android* dengan rentang umur paling banyak.



Gambar 3.3. Kelompok Usia Pengguna Sistem Operasi *Smartphone*  
(Sumber: <http://www.go-gulf.ae/blog/smartphone/>)

Mengingat karena keterbatasan waktu, tenaga dan biaya, peneliti membatasi cakupan *sample* hanya di lingkungan Ilmu Komputer Universitas Pendidikan Indonesia. Dalam pengujian ini responden akan diminta mengucapkan kata “kanan”, “kiri” dan “diam” masing-masing 10 kali. Kemudian dicatat jumlah suara yang berhasil dikenali dan yang gagal dikenali untuk dihitung akurasinya. Untuk menghitung persentase keberhasilan pengenalan suara, rumus yang digunakan adalah

Firman Fiqri Firdaus, 2015

*Implementasi mel frequency ceptral coefficient dan vector quantization pada pengenalan suara untuk permainan pesawat arcade berbasis android*

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

$$\%Keberhasilan = \frac{Jumlah\ ucapan\ berhasil\ dikenali}{Total\ jumlah\ ucapan} \times 100\%$$

Setelah itu responden diminta mengisi kuesioner yang bertujuan untuk menilai permainan yang dibuat. Kuesioner yang diisi terdiri dari 10 pertanyaan pilihan ganda. Setiap pertanyaan terdiri dari 4 pilihan jawaban yaitu sangat setuju, setuju, tidak setuju dan sangat tidak setuju. Hasil dari kuesioner tersebut dianalisa dan disajikan dengan metode distribusi frekuensi.

Pengujian terakhir dilakukan oleh ahli dalam bidang pengembangan permainan *mobile*. Proses ini dilakukan untuk memvalidasi apakah permainan yang dibuat telah sesuai standar permainan yang ada dipasaran. Selain itu proses ini juga dilakukan untuk menelaah menurut ahli apakah permainan dengan penerapan suara dapat membuat pemainnya merasa senang atau tidak.

### 3.3. Alat dan Bahan Penelitian

Dalam penelitian ini, peneliti menggunakan berbagai alat bantu penunjang baik berupa perangkat keras maupun perangkat lunak. Adapun perangkat keras dan spesifikasi yang digunakan adalah sebagai berikut :

1. Laptop / Perangkat Komputer, dengan spesifikasi :
  - *Processor* Intel Pentium P6200
  - RAM 3 GB DDR3
  - Hard disk 500 GB
  - VGA Intel HD Graphics
  - *Mouse* dan *Keyboard*
2. Smartphone dengan Sistem Operasi *Android 4.2 (Jelly Bean)*

Adapun perangkat lunak yang digunakan dalam penelitian ini adalah :

1. Sistem Operasi Windows 8.1
2. Sistem Operasi *Android 4.2 (Jelly Bean)*

Firman Fiqri Firdaus, 2015

*Implementasi mel frequency ceptral coefficient dan vector quantization pada pengenalan suara untuk permainan pesawat arcade berbasis android*

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

3. NetBeans IDE 8.0
4. Eclipse ADT (*Android Developer Tools*)
5. OC Volume – *Java Speech Recognition Engine*
6. *Game Framework Cocos2D*
7. CorelDRAW X5

Selain berbagai alat yang digunakan, peneliti juga menggunakan berbagai bahan sebagai penunjang dalam penelitian, yaitu :

1. Jurnal, buku, karya ilmiah, *e-book*, serta tulisan lain yang membantu peneliti dalam memahami permainan yang dibuat, MFCC, *Vector Quantization* dan teori-teori lain yang digunakan dalam penelitian.
2. Tutorial penggunaan *framework* Cocos2D dalam membuat permainan yang diakses melalui internet.