

RANCANG BANGUN APLIKASI *BEAT LOOP* DENGAN MENGGUNAKAN MEDIA SUARA REKAM PADA *SMARTPHONE* BERBASIS ANDROID

Sultoni Latif¹, Aan Erlansari², Funny Farady Coastera³

^{1,2,3}Program Studi Infomatika, Fakultas Teknik, Universitas Bengkulu.
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA
(telp: 0736-341022; fax: 0736-341022)

¹sultoni_latif@yahoo.com

²aan_erlanshari@unib.ac.id

³ffaradyc@unib.ac.id

Abstrak : Aplikasi ini dibuat berdasarkan konsep dasar pada *loop station*. Di mana konsep dasar yang akan diterapkan adalah perulangan. Data yang akan diolah pada aplikasi ini berupa data-data *beat* musik yang berextensi .wav dan data suara yang direkam langsung pada aplikasi. Aplikasi ini bertujuan untuk membantu dan mempermudah dalam menciptakan *beat-beat* musik loop yang baru untuk menambah variasi musiknya. Pengujian yang dilakukan pada penelitian ini yaitu pengujian fungsi program dan pengujian *black box*. Pengujian fungsi dilakukan dengan menguji beberapa fungsi pada aplikasi, sedangkan pengujian *black box* dilakukan dengan cara membagi domain *input* kasus pengujian dengan mengungkapkan kelas-kelas kesalahan. Hasil dari penelitian ini adalah sebuah aplikasi Android yang menerapkan konsep dasar *loop station* yaitu perulangan (*looping*) sehingga dapat digunakan oleh pengguna *beatboxer* atau para *user* lain sebagai media hiburan menciptakan musik *loop* yang unik dan menarik.

Kata Kunci : Aplikasi *Beat Loop*, *Loop Station*, Android, Musik

Abstract: The application is made based on the basic concept of loop station. The basic concept of loop station that will be applied is looping. Data which will be processed in this application is music beat in .wav extension form and sound data which directly recorded on the application. Goals of this application expected do help and facilitate in creating some new and unique music beats in order to add variations of music beat. Testings conducted in this research are testing of function program and black box. Function testing is done by giving snippet code function continued by screen shot on the application. While black box testing is done by dividing the input domain of test cases by revealing the error classes. The result of this research is an android application that implements the basic concept of loop station is looping so that it can be used by beatboxer users or other users as entertainment media create unique and interesting loop music.

Keywords: *Beat Loop Application*, *Loop Station*, *Android*, *Music*

I. PENDAHULUAN

Perkembangan musik di Indonesia sangat dipengaruhi oleh perkembangan musik dunia. Indonesia memiliki kebudayaan dan kesenian yang

kuat sejak jaman dahulu. Ini terbukti dengan banyaknya kesenian-kesenian daerah, terutama musik daerah. Meskipun demikian, negara Indonesia tidak menutup diri dari dunia luar. Banyak aliran-aliran musik yang masuk ke Indonesia yang berasal dari Eropa maupun Asia dan berkembang seiring berjalannya waktu. Kemajuan teknologi dengan munculnya internet dan sarana-sarana yang dapat mengakses informasi dengan cepat mempengaruhi selera musik dan perkembangan musik di Indonesia, sehingga sekarang banyak bermunculan jenis-jenis musik di Indonesia contohnya musik *pop*, *reggae*, *rock*, *underground*, *punk*, *funk*, *jazz*, *rap*, *hiphop*, *k-pop*, dan masih banyak lagi aliran musik yang berkembang.

Beatbox merupakan salah satu bentuk seni musik yang memfokuskan diri dalam menghasilkan bunyi-bunyi ritmis dan ketukan drum, instrument musik, maupun tiruan dari bunyi-bunyian lainnya seperti suara tetesan air, motor gede, suara robot, langkah kaki, dan khususnya suara *turntable*, melalui alat-alat ucap manusia seperti mulut, lidah, bibir, dan rongga-rongga ucap lainnya. Pada prakteknya *beatbox* juga diterapkan untuk genre musik lainnya seperti *Rock*, *Pop*, *R'nB*, dan sebagainya. Imitasi suara sudah dikenal sejak dahulu kala, contohnya seni-seni vokal perkusi seperti musik Bol di India, Kouji di China, maupun teknik vokal perkusi dari Afrika turut menjadi landasan dalam *beatboxing*. Di Indonesia pun terdapat Tari Kecak yang musik latarnya merupakan paduan bunyi-bunyi dari kosakata tertentu. Pemain *beatbox* atau lebih dikenal sebagai *beatboxer*, mampu mendemonstrasikan segala bentuk bunyi-bunyian dengan handal [7].

Beatboxer biasanya menggunakan alat pendukung seperti *mic*, *sound* dan *loop station*.

Loop station merupakan alat untuk memberikan efek "perulangan" pada suara yang direkam. *Loop station* digunakan untuk memperkaya suara yang dihasilkan sehingga terdengar semakin kompleks. Fitur pada *loop station* yang akan dimasukkan ke dalam sistem adalah fitur *loop* dan fitur *pitch*. Fitur *loop* adalah sebuah fitur pada *loop station* yang memberikan efek perulangan pada *beat* sedangkan fitur *pitch* adalah fitur yang digunakan untuk menaikkan atau menurunkan nada. Salah satu contoh *loop station* adalah Boss RC 505 dan Boss RC 300. Namun dalam prakteknya para *beatboxer* di Indonesia masih banyak yang belum menggunakan peralatan tersebut. Mereka terkadang hanya menggunakan *mic* yang ada di dalam peralatan VCD dan karaoke. Karena harga yang cukup mahal untuk sebuah *loop station* Boss RC 505 atau Boss RC 300 yang di banderol dengan harga Rp.8.000.000,- sampai Rp.10.000.000,- [4].

Berdasarkan latar belakang di atas, dengan adanya keterbatasan pada alat rekam yang dialami pada *beatboxer* di Indonesia maka akan dibuat penelitian tentang aplikasi musik yang berbasis *smartphone* yaitu Aplikasi *Beat Loop* dengan Menggunakan Media Suara Rekam sebagai *beatnya*. Aplikasi ini menerapkan konsep dasar dari *loop station* yaitu perulangan. Aplikasi ini berbasis Android sehingga para *beatboxer* dapat melakukan rekaman pada *smartphone* milik sendiri. Aplikasi ini diharapkan dapat membantu para *beatboxer* untuk dapat menyalurkan hobinya tanpa harus mengeluarkan biaya yang besar untuk membeli sebuah *loop station*.

II. LANDASAN TEORI

A. Perkembangan *Beatbox* di Indonesia

Sekitar tahun 2007, *beatbox* mulai berkembang di Indonesia. Jakarta *Beatboxing Community* atau

yang sering disingkat dengan JBC, merupakan sebuah komunitas *beatbox* pertama di Indonesia yang didirikan oleh Billy 'Bdabx' dan Tito 'Titz' (yang merupakan salah satu personel band *Bondan and Fade 2 Black*). Namun pada pertengahan tahun 2008, JBC berubah nama menjadi '*Indonesia Beatboxing Community*' atau IBC, karena melihat respon yang cukup tinggi dari para *beatboxer* dari luar daerah Jakarta untuk ikut bergabung. Sejauh ini komunitas *beatbox* terbesar di Indonesia adalah *Indonesia Beatbox Community* (IBC). Meskipun pada umumnya kegiatan IBC berada di seputar Jakarta, namun komunitas ini berafiliasi dengan komunitas *beatbox* lainnya dari luar Jakarta yang tersebar di seluruh Indonesia [2].

Pada 31 Maret 2012 kejuaraan *beatbox* dunia ke-3 diselenggarakan di Berlin, Jerman. Untuk pertama kalinya Indonesia berpartisipasi kejuaraan *beatbox* dunia atau *World Beatbox Battle* yang diikuti oleh Billy 'BdaBX' yang merupakan pendiri dari *Indonesia Beatbox Community* (IBC). Lewat audisi *online wildcard*, selain Billy 'BdaBX' ada juga *beatboxer* dari Indonesia lainnya yang mencoba ikut audisi seperti Lazuandi dari Depok, Jawa Barat. Meskipun Billy 'BdaBX' belum memenangkan kejuaraan ini, tapi ini adalah langkah awal yang sangat bagus untuk Indonesia. Perkembangan *beatbox* di Indonesia sudah tersebar luas, hal ini bisa dilihat dari munculnya berbagai komunitas *beatbox* di berbagai daerah yang ada di Indonesia. Kini seni musik *beatbox* sudah banyak dikenal dan diminati di Indonesia, mulai dari kalangan pelajar, mahasiswa dan masyarakat umum. Di berbagai media seperti televisi pun sudah banyak program yang menampilkan jasa *beatboxer* untuk mengisi acara. Karena dianggap unik dan menarik, *beatbox* dinilai menjadi suatu daya tarik bagi sebuah acara atau pertunjukan [2].

B. Audio/Suara

Suara bergerak seperti gelombang dengan kecepatan 750 mph (pada tingkat laut). Gelombang suara bervariasi dalam tingkatan tekanan suara (*amplitude*) dan dalam frekuensi *pitch*. Akustika merupakan ilmu fisika yang mempelajari suara. Tekanan suara diukur dalam satuan *decibel* (dB). Terlalu banyak suara akan membuat multimedia menjadi berisik dan merugikan. Suara di alam ini merupakan gelombang analog.

Jumlah waktu yang diperlukan untuk terjadinya suatu getaran atau gelombang dinamakan periode (T). Sedangkan jumlah gelombang yang terjadi pada setiap detik dinamakan frekuensi (f) dengan satuan m/dt (Hz). Sebagai tambahan, suara yang dapat diterima telinga manusia berkisar antara 20 Hz sampai dengan 20.000 KHz (1 Hz = 0,001 KHz). Saat ini komputer sudah terintegrasi dengan perangkat yang dapat membangkitkan sinyal suara dengan baik. Ada dua macam suara yang sering digunakan pada komputer, yaitu *audio digital* dan MIDI [3].

C. Audio Digital

Audio digital merupakan *versi digital* dari suara analog. Perubahan suara analog menjadi suara *digital* membutuhkan suatu alat yang disebut *Analog to Digital Converter* (ADC). ADC akan mengubah *amplitude* sebuah gelombang analog ke dalam waktu interval (sampel) sehingga menghasilkan representasi *digital* dari suara.

Berlawanan dengan ADC, *Digital to Analog Converter* (DAC) akan mengubah suara *digital* ke suara analog (*speaker*). *Audio digital* merupakan representasi dari suara asli (*original sound*). Dengan kata lain, *audio digital* merupakan sampel suara. Kualitas perekaman *digital* tergantung pada seberapa sering sampel diambil (angka sampling atau frekuensi dihitung dalam kilohertz atau seribu

sampel per detik). Tiga frekuensi sampling yang paling sering digunakan dalam multimedia adalah kualitas CD 44.1 kHz, 22.05 kHz, dan 11.025 kHz dengan ukuran sampel 8 bit dan 16 bit. Ukuran sampel 8 bit menyediakan 256 unit deskripsi jarak dinamis atau amplitudo (level suara dalam satu waktu). Ukuran *file* dari *audio digital* bergantung pada angka sampling, resolusi dan *channel*-nya (*Stereo* atau *Mono*) [3].

Kualitas perekaman *digital* bergantung pada seberapa sering sampel diambil dan berapa banyak angka yang digunakan untuk menyajikan nilai dari tiap sampel (*bitdepth*, ukuran sampel, resolusi, jarak dinamis). Semakin sering sampel diambil, semakin banyak data yang disimpan mengenai sampel, semakin bagus resolusi dan kualitas suara yang ditangkap ketika diputar. Artinya, kualitas suara akan semakin tinggi. Semakin tinggi kualitas suara, semakin besar pula ukuran *file* yang dihasilkan. Resolusi *audio* (8 bit atau 16 bit) menentukan akurasi proses *digital* dari suara. Penggunaan bit yang lebih besar untuk ukuran sampel akan menghasilkan hasil rekaman yang menyerupai *versi* aslinya [3].

D. Pitch

Pitch Shifting (menggeser *fundamental frequency*) adalah salah satu fungsi yang digunakan untuk melakukan pergeseran terhadap frekuensi dasar suatu nada suara. Misalkan suara alat musik pada nada tertentu terdengar tidak pas (fals), maka nada tersebut bisa dikoreksi dengan menggunakan *pitch shifting* [5].

Pitch dapat dikatakan sebagai tinggi rendahnya nada. Sebuah not A yang dimainkan diatas C4 (*middle C*) memiliki frekuensi *pitch* sebesar 440 Hz. Perubahan yang sedikit tidak mempengaruhi secara signifikan dari *tone* yang dimainkan. Perubahan yang masih diijinkan sebesar lima

persen. Namun perubahan tersebut akan terasa signifikan bila dimainkan secara bersama-sama dengan nada yang lainnya. *Pitch* bergantung pada amplitudo dari suara. Hal ini sangat penting untuk suara-suara berfrekuensi rendah. Contohnya suara *bass* akan terdengar lebih rendah. Seperti yang sudah lama dialami oleh banyak orang, dimana persepsi dari *pitch* dapat ‘kabur’ contohnya ilusi pada audio dan *tritone paradox*.

Pitch seringkali diberi label dengan menggunakan cara *scientific* yang merupakan kombinasi dari huruf dan angka dimana merepresentasikan frekuensi fundamental, contohnya A4 atau A440. Namun hal ini menimbulkan 2 masalah yaitu dalam standarisasi Barat penulisan G4## memiliki penulisan *pitch* yang sama dengan A4. Kedua, persepsi manusia terhadap *pitch* adalah logaritmik, sehingga hal yang diterima ketika mendengar A220 dan A440 adalah sama ketika mendengar A440 dan A880.

Untuk menghindari hal tersebut, para ilmuwan berusaha merepresentasikan *pitch* dari nada-nada yang ada dengan menggunakan *numerical scale*. Cara ini berdasarkan pada logaritma dari frekuensi fundamental masing- masing nada. Contohnya, dalam mengadaptasi bunyi *pitch* ke dalam sistem MIDI dapat dilakukan konversi sebagai berikut : Hal ini membentuk penomoran yang linear untuk 12 nada dalam 1 oktaf. Contohnya untuk A440 memiliki nomor *pitch* 69. Sistem ini bersifat fleksibel dimana dapat mengikutsertakan “*microtones*” yang tidak ditemukan dalam keyboard yang standar [5].

E. Aransemen Musik

Aransemen berasal dari bahasa Belanda *Arrangement*, yang artinya penyesuaian komposisi musik dengan nomor suara penyanyi atau instrumen musik yang di dasarnya atas sebuah

komposisi yang telah ada sehingga esensi musiknya tidak berubah. Orang yang melakukan aransemen lagu dikenal dengan sebutan *Aranger* atau pengaransemen. Modal dasar yang harus dimiliki oleh seorang *arranger* adalah menguasai pengetahuan tentang harmoni. Ada tiga jenis aransemen sebagai berikut :

a. Aransemen Vokal

Setiap lagu dapat disusun aransemen khusus vokal, yaitu dalam dua suara, tiga suara, empat suara. Untuk menyusun aransemen vokal, yang paling mudah adalah menyusun atransemen lagu dalam dua suara, karena untuk menyusun aransemen lagu dalam tiga dan empat suara ada banyak persyaratan yang harus diperhatikan. Untuk memperoleh hasil yang lebih baik dan memuaskan, setelah selesai disusun aransemen lagunya kemudian dicoba untuk dinyanyikan secara bersama-sama, apabila dirasa kurang baik/memuaskan maka dapat dicoba lagi untuk menyusun aransemen lagu tersebut hingga pada akhirnya diperoleh hasil yang sangat memuaskan.

b. Aransemen Instrumen

Dalam menyusun aransemen instrumen sangat berbeda dengan aransemen vokal. Untuk menyusun aransemen instrumen kita harus menyesuaikan dengan alat-alat musik yang dipergunakan. Semakin lengkap alat musik yang kita pergunakan, semakin banyak pula kemungkinan variasi yang dapat diciptakan. Untuk menyusun aransemen instrumen, kita harus berpedoman pada pengetahuan ilmu harmoni dan akord. Bagian-bagian dari suatu aransemen musik dikenal dengan istilah *Partituur* (Belanda), *Partitura* (Italia), *Part* (Inggris), *Parte* (Perancis). Dan dalam aransemen instrumen, kebanyakan partitur dimainkan bergantian tugas, sedangkan dalam aransemen

vokal pada umumnya semua partitur umumnya berbunyi bersamaan.

c. Aransemen Campuran

Yang dimaksud aransemen campuran adalah campuran aransemen vocal dan instrumen. Teknik yang dilakukan adalah menggabungkan dua jenis arransmen yang telah ada. Dalam aransemen campuran pada umumnya yang ditonjolkan adalah vokalnya, sedangkan instrumennya berfungsi untuk pengiring dan memeriahkan, sehingga pertunjukan yang disajikan bertambah sempurna. Untuk mengendalikan keseimbangan dalam menampilkan aransemen yang telah disusun diperlukan adanya seorang pemimpin yaitu seorang dirigen atau konduktor [9].

F. Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*. Pada saat perilisan perdana *Android*, 5 November 2007, *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh

dari *Google* atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution* (OHD) [8].

G. UML (*Unified Modelling Language*)

UML terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang kita sebut sebuah model [1]. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C [1].

H. Perulangan

Loop secara berulang mengeksekusi sebarisan instruksi yang sama sampai kondisi akhir ditemui. Dengan kata lain, *looping* atau *loop* artinya mengulangi eksekusi blok program tertentu sampai tercapai kondisi untuk menghentikannya (terminasi) [5]. Setiap perulangan memiliki 4 bagian yaitu :

1. Inisialisasi (*initialization*)
2. Badan program (*body*)/(*statement*)
3. Iterasi (*iteration*), dan
4. *Termination*.

Struktur kontrol pengulangan adalah berupa pernyataan dari Java yang mengizinkan kita untuk mengeksekusi blok *code* berulang-ulang sesuai dengan jumlah tertentu yang diinginkan. Ada tiga macam jenis dari struktur kontrol pengulangan yaitu [5]:

a. Statement *While*

Pernyataan *while loop* adalah pernyataan atau blok pernyataan yang diulang-ulang sampai mencapai kondisi yang cocok. Bentuk pernyataan *while*,

```
while( boolean_expression ){  
statement1;  
statement2;  
...  
}
```

Pernyataan di dalam *while loop* akan dieksekusi berulang-ulang selama kondisi *boolean_expression* bernilai benar (*true*).

b. Statement *Do While*

Do-while loop mirip dengan *while-loop*. Pernyataan di dalam *do-while loop* akan dieksekusi beberapa kali selama kondisi bernilai benar (*true*). Perbedaan antara *while* dan *do-while loop* adalah dimana pernyataan di dalam *do-while loop* akan dieksekusi sedikitnya satu kali [6].

Bentuk pernyataan *do-while*,

```
do{  
statement1;  
statement2;  
...  
}while( boolean_expression );
```

Pernyataan di dalam *do-while loop* akan dieksekusi pertama kali, dan akan dievaluasi kondisi dari *boolean_expression*. Jika nilai pada *boolean_expression* tersebut bernilai *true*, pernyataan di dalam *do-while loop* akan dieksekusi lagi.

c. Statement Perulangan *For*

Perulangan *for* menyediakan sarana mengulang kode dalam jumlah yang tertentu. Pengulangan ini terstruktur untuk mengulangi kode sampai tercapai batas tertentu [6].

Berikut bentuk dasar perulangan *for* :

for(InitializationExpression; LoopCondition; StepExpression)
statement

1. *InitializationExpression*, digunakan untuk inialisasi variabel kendali perulangan.
2. *LoopCondition*, membandingkan variabel kendali perulangan dengan suatu nilai batas.
3. *StepExpression*, menspesifikasikan cara variabel kendali dimodifikasi sebelum iterasi berikutnya dari perulangan.

III. METODE PENELITIAN

A. Jenis Penelitian

Jika dilihat dari tujuannya, jenis penelitian yang digunakan adalah penelitian terapan. Penelitian terapan bertujuan untuk memperoleh penemuan-penemuan yang berkenaan dengan aplikasi/penerapan teori-teori tertentu. Penelitian yang akan dilakukan yaitu penelitian tentang Implementasi fungsi *looping* dan metode *pitch* menggunakan rekaman suara pada aplikasi *loop beat*. Fungsi *looping* digunakan pada perulangan suara yang telah di rekam atau pun yang telah ada pada aplikasi *loop beat* dan fungsi *pitch* digunakan untuk menaikkan atau menurunkan nada pada setiap *beat* hasil rekaman atau *beat* yang telah ada.

B. Teknik Pengumpulan Data

Berikut ini merupakan teknik pengumpulan data yang digunakan dalam penelitian ini :

1. Studi Pustaka

Penelitian ini dilakukan dengan mencari informasi melalui internet serta penulis mempelajari jurnal yang berhubungan dengan musik dan pemograman Android serta mempelajari perangkat lain yang diperlukan dalam perancangan tugas akhir ini. Dari bahan-bahan tersebut diambil teori-teori yang dapat dijadikan landasan untuk menganalisa masalah yang di temukan dalam penelitian.

2. Studi Literatur

Pengumpulan data dengan cara mengumpulkan literatur, jurnal, paper dan bacaan – bacaan yang ada kaitannya dengan judul penelitian.

C. Metode Pengembangan Sistem

Pengembangan aplikasi *beat loop* yang digunakan dalam penelitian ini adalah model *prototyping*. Berikut tahap-tahap yang dilakukan didalam membangun aplikasi *loop beat* menggunakan model *prototyping*:

1. Analisa Kebutuhan

- a. Perencanaan dimulai dengan pengumpulan data dan informasi dari aplikasi yang akan dibangun. Data masukan yang dibutuhkan dalam aplikasi ini adalah informasi dan data seputar *beat* lagu.

- b. Kebutuhan data keluaran

Adapun data keluaran berupa audio atau *beat* lagu sesuai data masukan dan yang dibangun dalam aplikasi *loop beat*.

- c. Kebutuhan antarmuka

Kebutuhan antarmuka pada aplikasi adalah kemudahan dan kenyamanan pengguna saat mengakses aplikasi *loop beat*.

2. Perancangan Sistem dan Aplikasi

Tahap ini bertujuan memperkirakan pengkodean sistem dan merancang tampilan dari sistem. Tahap ini juga membantu dalam menspesifikasikan kebutuhan perangkat keras dan sistem. Untuk membangun aplikasi *beat loop* ini, maka dibutuhkan metode mendesain tampilan yang menarik bagi si pengguna. Kemudian dirancang *flowchart*, beserta *interface* guna mempermudah pembuatan sistem yang akan dibangun. Desain ini meliputi desain proses, kategori, level, dan desain *interface*.

Dalam pembuatan sistem ini diperlukan beberapa perangkat lunak dan perangkat keras guna membantu pembuatan dan penyelesaian sistem. Berikut adalah perangkat lunak dan perangkat keras yang dibutuhkan untuk menyelesaikan sistem.

3. Penerapan dan Pengujian Sistem

Tahap penerapan dan pengujian program adalah hasil dari tahapan implementasi. Hasil pemrograman di tahap sebelumnya diterapkan pada tahap ini dan diuji kelayakannya. Proses pengujian yang dilakukan pada sistem yang dibuat menggunakan dua metode pengujian, yaitu pengujian fungsi dan *black box testing*.

a. Pengujian fungsi

Dalam pengujian ini, meneliti kode-kode program yang ada dan akan menganalisis apakah ada kesalahan atau tidak. Jika terdapat bagian dari kode yang menghasilkan *output* yang tidak sesuai, maka akan dilakukan pengecekan satu per satu dan dilakukan perbaikan. Pengujian dilakukan dengan cara:

- 1) Pengujian penerapan Fungsi *looping* dan metode *pitch*
- 2) Menjalankan fungsi-fungsi dan meneliti kode-kode program yang terdapat dalam aplikasi *loop beat*.

3) Menjalankan aplikasi *loop beat* di *smartphone* berbasis android

b. *Black Box Testing*

Sedangkan pada pengujian ini, dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari aplikasi yang telah dibuat. Pengujian sistem ditujukan dengan mencari tahu kekurangan yang terdapat pada sistem *game* lalu memperbaiki sistem yang belum optimal yang terdapat pada aplikasi *loop beat*.

c. Metode Uji Kelayakan Sistem

Uji kelayakan dilakukan untuk mendapatkan penilaian langsung terhadap sistem yang dihasilkan. Adapun uji kelayakan ini adalah sebagai berikut:

1) Kuesioner

Kuesioner merupakan daftar pertanyaan yang diajukan pada responden untuk mencari jawaban dari permasalahan yang diteliti [8]. Teknik pengambilan sampel yang digunakan adalah teknik *purposive sampling*. *Purposive sampling* adalah teknik pengambilan sampel yang berdasarkan penilaian peneliti bahwa sampel tersebut merupakan pihak yang dapat dijadikan sampel penelitian.

2) Tabulasi Data

Proses perhitungan data kuesioner menggunakan skala *Likert*. Skala *Likert* adalah perhitungan skor pada tiap-tiap interval dari pernyataan yang diberikan ke responden. Sebelum melakukan perhitungan dengan menggunakan skala *Likert*, maka terlebih dahulu dicari interval kelas dengan persamaan:

$$i = \frac{m - nk}{k}$$

Keterangan:

i = interval kelas

m = angka tertinggi skor

n = angka terendah skor

k = banyak kelas

Kemudian hasil dari proses perhitungan yang telah dilakukan akan disajikan dalam bentuk tabel.

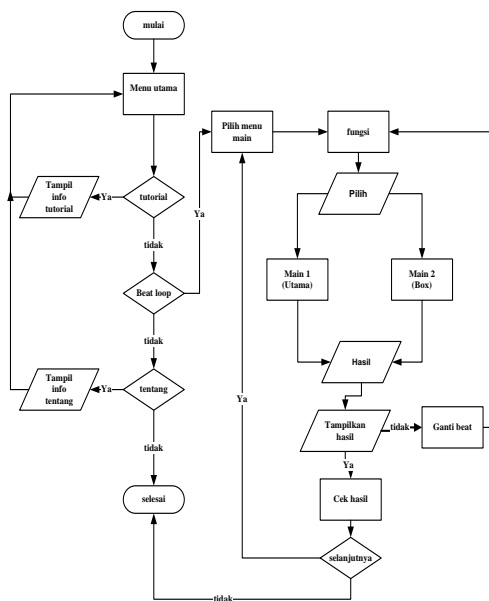
D. Penggunaan dan Pemeliharaan

Setelah aplikasi selesai maka pengguna akan menggunakan aplikasi. Jika terdapat pengembangan fungsional dari aplikasi yang diinginkan oleh pengguna, maka akan dilakukannya pemeliharaan.

IV. ANALISA DAN PERANCANGAN

A. Diagram Alur Aplikasi

Secara garis besar cara kerja sistem yang dibangun ditampilkan dalam Gambar 4.1. Alur aplikasi *beat loop* dengan menerapkan fungsi *looping* dan fungsi *pitch* ini menyajikan permainan *beat* musik dari *beat* yang tersedia atau *beat* yang merupakan hasil rekaman. Dimana hasilnya akan menciptakan musik yang baru dari hasil gabungan *beat-beat* yang unik. berikut ini adalah diagram alur aplikasi, seperti pada Gambar 4.1.



Gambar 4.1 Diagram Alur Aplikasi

V. PEMBAHASAN

A. Implementasi AntarMuka

1. Icon Beat Loop dan Splash Screen

Icon beat loop merupakan *Icon* yang akan tampil pada *smartphone* pengguna ketika pengguna telah menginstal aplikasi *beat loop*. Berikut adalah tampilan dari *Icon Vocalizing* ditunjukkan pada Gambar 5.1.



Gambar 5.1 Icon Beat Loop

Icon beat loop ini merupakan *icon* khusus yang dibuat untuk membedakan aplikasi *beat loop* dengan aplikasi lainnya. Selanjutnya saat aplikasi *beat loop* dijalankan maka akan tampil tampilan *splash screen* dari aplikasi aplikasi *beat loop* yang ditunjukkan pada Gambar 5.2.



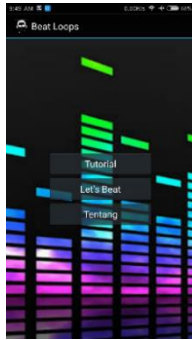
Gambar 5.2 Splash Screen

Pada gambar 5.2 merupakan tampilan *splash screen* merupakan tampilan awal ketika pengguna menjalankan aplikasi *beat loop*. Pada tampilan *splash screen* akan tampil lambang aplikasi dan nama aplikasi.

2. Halaman Menu Utama aplikasi

Setelah tampilan *splash screen* dijalankan, maka tampilan selanjutnya adalah halaman menu utama aplikasi. Menu utama aplikasi merupakan

menu yang akan tampil setelah splash screen dijalankan yang ditunjukkan pada Gambar 5.3.

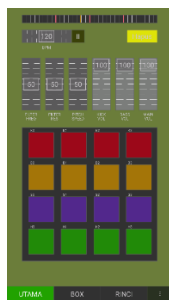


Gambar 5.3 Tampilan Menu Utama Aplikasi *Beat Loop*

Pada gambar 5.3 merupakan tampilan menu utama yang menampilkan tiga pilihan menu yang dapat pengguna pilih, yaitu menu tutorial merupakan menu yang akan memberikan informasi bagaimana cara menggunakan aplikasi. Menu *let's beat* pada aplikasi merupakan menu utama aplikasi yang digunakan untuk menjalankan aplikasi dan menu tentang merupakan menu yang menjelaskan tentang pembuat aplikasi *beat loop*.

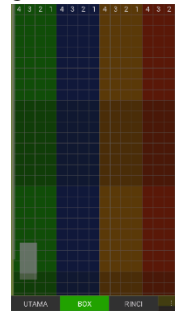
3. Halaman Menu *Let's beat*

Pada halaman menu *Let's beat* merupakan halaman aplikasi utama dimana aplikasi *beat loop* berjalan, pada menu ini pengguna diberikan 16 *beat* awalan yang dapat digunakan secara langsung. Dengan cara menekan *beat* yang akan di pilih. Pada menu utama kita dapat mengatur tempo dari 80 Bpm-160 Bpm. Pada menu ini kita dapat juga mengatur *pitch Seed* yang dapat mengubah suara *beat* sesuai dengan keinginan kita yang ditunjukkan pada gambar 5.4.



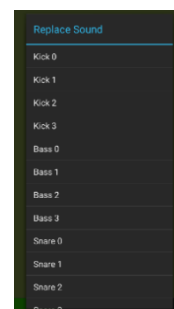
Gambar 5.4 Tampilan Menu *Let's Beat*

Pada gambar 5.4 pengguna dapat memilih jenis tampilan yang disediakan berupa tampilan menu utama, tampilan menu *box* dan tampilan rinci. Dimana setiap tampilan berbeda-beda sesuai dengan fungsi masing-masing pada menu ini terdapat menu hapus yang berguna untuk menghapus secara keseluruhan pilihan *beat* sesuai dengan warna yang di pilih. Tampilan menu *box* ditunjukkan pada gambar 5.5.



Gambar 5.5 Tampilan Menu *Box*

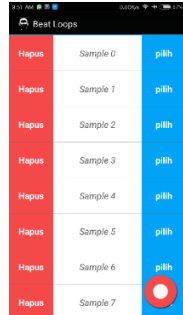
Pada gambar 5.5 pengguna akan di berikan tampilan berupa kotak yang terdiri dari 32 kotak tiap *beat*-nya sesuai dengan warna masing masing di antaranya kotak merah untuk *beat kick*, kotak kuning untuk *beat bass*, kotak ungu untuk *beat snare* dan kotak hijau untuk *beat hit hat*. Tampilan menu *replace sound* ditunjukkan pada gambar 5.6.



Gambar 5.6 Tampilan Menu *Replace Sound*

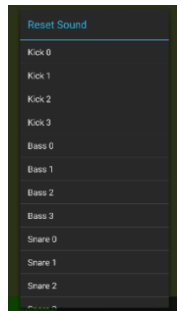
Pada gambar 5.6 pengguna dapat mengubah *beat* yang tersedia pada menu utama dengan *beat-beat* yang baru. Dimana *beat* baru tersebut di rekam melalui aplikasi secara langsung. *Beat* dapat berubah suara ketukan atau pun suara lainnya.

Sesuai dengan keinginan si pengguna dalam Mecampurkan *beat-beat* music. Tampilan menu rekam *beat* ditunjukkan pada gambar 5.7.



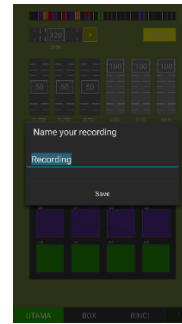
Gambar 5.7 Tampilan Menu Rekam *Beat*

Pada gambar 5.7 pengguna menggunakan tombol bulat merah untuk merekam suara yang ingin dijadikan *beat* terbaru. Dalam menu ini pengguna dapat memutar secara langsung hasil rekaman yang telah di buat dengan menekan tombol tulisan sampel 1-dst Dan pengguna dapat menghapus langsung hasil rekaman apabila tidak sesuai. Tampilan menu *reset sound* ditunjukkan pada gambar 5.8.



Gambar 5.8 Tampilan Menu *Reset Sound*

Pada gambar 5.8 pengguna dapat mereset ulang *beat* yang telah di ubah dengan menggunakan suara rekam. Pada menu ini *beat* akan kembali ke *beat* awal sesuai pada saat kita menjalankan program untuk pertama. Tampilan menu utama untuk *recording* ditunjukkan pada gambar 5.9.



Gambar 5.9 Tampilan Menu Utama Untuk *Recording*

Pada gambar 5.9 pengguna dapat memberikan nama untuk hasil rekaman yang di buat setelah kita memainkan *beat loop*. Hasil dari rekaman tersebut akan masuk ke dalam folder *beat loop* pada memori. Format yang dihasilkan adalah format dengan ekstensi *.wav*. berikut contoh hasil rekaman pada folder *beat loop* dapat di lihat pada gambar 5.10.

A. Pengujian *Black Box*

Pengujian *black box* telah dilakukan dengan menggunakan metode *equivalence partitioning* dan terdapat 7 kelas uji dan 18 skenario. Dari pengujian tersebut didapatkan hasil bahwa semua skenario uji berhasil 100%. Sehingga, dapat disimpulkan semua hasil yang diharapkan berhasil dijalankan dan berfungsi sebagaimana mestinya.

VI. PENUTUP

A. Kesimpulan

Berdasarkan analisa perancangan sistem, implementasi, dan pengujian sistem, maka dapat disimpulkan bahwa :

1. Penelitian ini telah berhasil menghasilkan aplikasi *beat loop* yang dapat digunakan oleh pengguna *beatboxer* atau para *user* lain sebagai media hiburan menciptakan musik *loop* yang unik dan menarik.
2. Pada perbandingan dengan aplikasi sejenis yang terdapat pada *playstore*. Didapatkan hasil bahwa aplikasi yang dirancang memiliki

- kelebihan yang tidak terdapat pada aplikasi sejenisnya yaitu dapat merekam suara secara langsung dan mengganti *beat* dengan file format .wav yang tersedia di *smartphone*.
3. Pada penelitian ini telah dilakukan pengujian *black box* yang dilakukan untuk menguji fungsi-fungsi khusus dari aplikasi di mana terdapat sebanyak 7 kelas uji dan 18 skenario dengan hasil 100% berjalan dengan baik.
 4. Berdasarkan hasil pengujian sistem dengan cara menggunakan angket, Rancang bangun aplikasi *beat loop* dengan menggunakan media suara rekam pada *smartphone* berbasis Android dapat disimpulkan bahwa variabel tampilan mendapatkan penilaian sangat baik (26%), baik (67%) dan cukup baik (7%). Variabel kinerja sistem mendapatkan penilaian sangat baik (23%), baik (67%), dan cukup baik (10%). Variabel kemudahan pengguna mendapatkan penilaian sangat baik (30%), baik (67%) dan cukup baik (3%).
 5. Aplikasi ini telah diunggah di *playstore* dengan link sebagai berikut:
<https://play.google.com/store/apps/details?id=unib.latif.drumloops.pemula>.

B. Saran

Berdasarkan analisa dan perancangan sistem, implementasi, dan pengujian sistem, maka saran yang dapat diberikan untuk pengembangan penelitian selanjutnya sebagai berikut :

1. Sistem ini dapat terus dikembangkan lebih lanjut dalam hal fungsi yang digunakan, kedepannya diharapkan untuk dapat menggunakan fungsi-fungsi selain *pitch Speed* seperti *trim*, *playback* dan *costum effect*.
2. Sistem ini dapat dilakukan pengembangan dengan pembaruan *beat* secara *online* sehingga

- dapat membuat aplikasi ini dapat digunakan lebih luas dan memiliki banyak *beat* baru.
3. Sistem ini dapat terus dikembangkan lebih lanjut dalam hal peningkatan kualitas fitur rekam karena pada aplikasi ini masih terdapat *noise-noise* ketika merekam suara yang masih belum jernih.

REFERENSI

- [1] A. S., Rosa dan Shalahuddin, M. (2013). *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Informatika. Bandung.
- [2] Banoë, Pono. (2003). *Kamus Musik* Yogyakarta: Penerbit Kanisius.
- [3] Binanto, I.(2010). *Multimedia Digital – Dasar Teori dan Pengembangannya*. Yogyakarta: Penerbit C.V Andi Offset.
- [4] Exxtrastore. (2017). *Boss RC-505 Loop Station*.www.ebay.com/itm/Boss-RC-505-Loop-Station. diakses pada 17/06/2017 21:09.
- [5] Muljono, S. Y. (2012). Sintesis Nada Saron Menggunakan Pitch Shifting Phase Vocoder untuk Standarisasi Suara Saron. Semarang: Universitas Dian Nuswantoro Semarang.
- [6] Naughton, P. (1996). *Java Handbook: Konsep Dasar Pemrograman Java*. Yogyakarta: Penerbit Andi.
- [7] Razzak, Rifiana Abdul. (2013). *Kreativitas Musik Kelompok Beatbox Community Of Semarang*. Skripsi Fakultas Bahasa Dan Seni, Universitas Negeri Semarang. Semaarang: Vol 1 No.1.
- [8] Safaat, Nazarudin Harahap. (2012). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika. Bandung.
- [9] Ubay.(2014).*Pengertian Aransemen*. www.perpussekolah.com/2014/12/pengertian-aransemen.html. diakses pada 05/010/2017 20:09.