

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini akan digunakan lima tinjauan pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan pustaka yang diambil yaitu:

1. Menurut Manfred R Schroder, *The Journal of the Acoustical Society of america* 43 (4), 829-834, 1968 : Frekuensi fundamental dari sinyal periodik tidak memiliki komponen untuk pengukuran. Pengukuran hanya dapat ditentukan dengan mengukur frekuensi komponen harmonik yang lebih tinggi dan menghitung frekuensi yang lebih besar. berdasarkan konsep-konsep ini hasil dari analog komputer dan instrumentasi menunjukan hal ini adalah metode baru.
2. Menurut Ming Zhao, Jing Lin, Yonghao Miao, Xiaoqiang Xu, *Measurement* 91, 421-439, 2016 : (HPS) *Harmonic Product Spectrum* di usulkan untuk mendeteksi dan mengidentifikasi berbagai sumber modulasi getaran yang terkubur dalam sinyal getaran. Dengan metode ini, fitur impulsif terkait kesalahan dapat dikenali, sementara pengaruh yang disebabkan oleh modulasi non-kesalahan dihilangkan. Atas dasar ini, indeks signifikan harmonik lebih lanjut ditetapkan untuk mengandung informasi yang sempit. Dengan menggunakan indeks ini, pita resonansi optimal dimana impuls gangguan yang paling signifikan dapat ditentukan secara akurat. Kempulan dari (HPS) adalah produk *spectrum sideband* yang diintergrasikan untuk mengurangi *noise in-band* dan lebih jauh memulihkan impuls gangguan.
3. Menurut Rehan Ahmed, Roberto Gill-Pita, David Ayllon, Lorena Alvarez, *Audio Engineering Society Convention* 130, 2011 :

Algoritma deteksi *pitch* didasarkan pada *Harmonic Product Spectrum*. Karena sinyal pada nada bicara berfluaksi dengan mudah dan algoritma berbasis bingkai pun digunakan untuk mengekstrasi beberapa nada dalam setiap bingkai. Kemudian frekuensi dasar (*pitch*) untuk masing-masing sumber diestimasi dan dilacak setelah membandingkan semua frame. Frekuensi fundamental yang diperkirakan dari sumber kemudian digunakan untuk menghasilkan satu set

topeng biner yang memungkinkan pemisahan sinyal dalam domain *Short Time Fourier Transform* menunjukkan pemisahan yang cukup besar dari sinyal bicara.

4. Menurut Robert S. Boyer dan J. Strother Moore pada tahun 1997 : bagaimana mencocokkan sebuah karakter dari sebelah kanan *pattern* sehingga proses pencariannya lebih cepat. Dibalik algoritma ini dengan melakukan pencocokan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang dibutuhkan. Dimana kita bisa melompati atau tidak pada pola merupakan perbandingan-perbandingan karakter yang diprediksikan akan gagal. Karakter paling kanan pada pola merupakan karakter pertama yang akan dicocokkan dengan teks.

5. Menurut Rahul B Kodag, Mukesh D Patil, Chandrakant J Gaikwad, *International Conference on Computing, Analytics and Security Trends (CAST) 622-626* pada tahun 2016 : Dimana Tonik adalah salah satu bagian integral dari musik yang dianalisis menggunakan metode ekstraksi tonik musik India. Algoritma yang kompleks secara komputasi dan sedikit lebih lambat untuk ekstraksi tonik. Penggunaan Harmonic Product Spectrum mengusulkan metode berbeda yaitu puncak tertinggi dan pencocokan template. Keadaan seni dan algoritme yang diusulkan juga telah dibandingkan menggunakan korpus musik Hindustan dan Karnatik. Analisis objektif menghasilkan peningkatan akurasi dibandingkan dengan algoritma yang sudah ada untuk ekstraksi tonik nada mono.

Berdasarkan penelitian terdahulu seperti di atas menunjukkan bahwa metode yang digunakan mampu meningkatkan kinerja sistem yang dapat dilihat perbedaannya yaitu:

1. Metode pengembangan yang digunakan mengacu pada kemudahan dalam menganalisis gelombang suara yang dihasilkan.
2. Sistem yang dibuat menghasilkan aplikasi *android* yang dapat di instal pada perangkat *smartphone* dengan sistem operasi *android*.
3. Penambahan grafik untuk menggambarkan hasil proses dengan skor penilaian.

2.2 Aplikasi

Menurut Yasin (2012), Aplikasi yaitu:

“Aplikasi merupakan sekumpulan elemen yang saling berinteraksi dan saling berketerkaitan antara satu dengan yang lain dalam melakukan suatu kegiatan secara bersama untuk mencapai tujuan tertentu”.

Menurut Sutabri (2012), Aplikasi adalah:

“Alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya”.

Dari pengertian diatas penulis menyimpulkan aplikasi adalah *software* atau alat terapan yang dibuat untuk mengerjakan tugas-tugas khusus yang saling berinteraksi untuk tujuan tertentu.

2.3 Teknik Vokal Bernyanyi

Menurut Dimas Aditya Wicaksono (2019):

Teknik vokal adalah cara memproduksi suara yang baik dan benar, sehingga suara yang keluar begitu indah, merdu dan nyaring. Menyanyikan sebuah lagu dibutuhkan latihan vokal agar dapat mengetahui tinggi rendahnya suatu nada dan terjadi suatu keselarasan antara nada dan suara. Beberapa dalam melatih unsur vokal yang harus dikuasai adalah latihan breath control, artikulasi dan notasi angka.

2.4 Metode Harmonic Product Spectrum

1. Menurut Garreth (2003) : *Harmonic Product Spectrum* merupakan metode *pitch detection algorithm* yang paling simpel dan bekerja dengan baik dalam berbagai kondisi. Harmonic Product Spectrum (HPS) memerlukan beberapa proses pengolahan dari audio file yaitu windowing dan Fast Fourier Transform (FFT). Ketika menggunakan FFT untuk mengukur frekuensi dari data, maka harus menentukan dasar analisa pada sebuah data yang pasti. FFT mengasumsikan bahwa data yang tetap adalah sebuah periode dari period signal. Bagi FFT *time-domain* dan *frequency-domain* adalah perputaran yang sama, sehingga titik terakhir dari time waveform dianggap berhubungan. Sehingga keterbatasan dalam sampling record menghasilkan *waveform* yang terpotong dengan perbedaan karakter dari sinyal aslinya, dan keterbatasan ini membuat perubahan yang tajam dalam pengukuran data. Untuk mengurangi efek perubahan ini, kita bisa menggunakan fungsi *Window* untuk sinyal dalam *time-domain* sebelum diproses menuju FFT. A-D *converter* untuk mengubah sinyal analog menjadi digital sebelum disimpan dalam komputer. Proses komputasi spektrum dari sinyal yang telah direkam adalah menggunakan *Fast Fourier Transform*. Pada gambar berikut ini merupakan hasil dari *Hamming Window* atau yang disebut juga proses pengubahan sinyal. *Harmonic Product Spectrum* digunakan untuk melakukan analisis dan penentuan akurasi. Tahapan pertama dalam algoritma *Harmonic Product Spectrum* ini adalah proses FFT (*Fast Fourier Transform*) yaitu sinyal dalam domain waktu kemudian diubah ke domain frekuensi untuk melihat range frekuensinya. Proses ini bertujuan untuk menentukan range frekuensi yang memiliki energi paling maksimum.

2.5 Metode Boyer Moore

Ide utama dari algoritma ini adalah dengan melakukan pencocokan dari paling kanan string yang dicari (Aulia, 2008). Dengan menggunakan algoritma ini, secara rata-rata proses pencarian akan lebih cepat dibandingkan dengan proses pencarian lainnya. Ide dibalik algoritma ini adalah bahwa dengan memulai pencocokan karakter dari kanan, dan bukan dari kiri, maka akan lebih

banyak informasi yang didapat. Dalam pencarian *Boyer-Moore*, proses awal yang dilakukan adalah penempatan *window* dari pattern di *text* yang tersedia. Proses pencarian dimulai dari karakter paling kanan pattern. Setiap karakter akan dibandingkan satu per satu. Jika terjadi ketidakcocokan, maka akan dicek nilai pergeseran yang mungkin dilakukan. Nilai terbesar yang didapat akan diambil dan pergeseran *window* akan dilakukan sesuai dengan nilai tersebut.

2.5.1 Kelebihan Metode Boyer Moore

Menurut Utomo, 2008 : Semakin panjang pola yang dicari, maka waktu pencarian semakin singkat.

2.5.2 Kekurangan Metode Boyer Moore

Menurut Aulia, 2008 : Jika pattern yang pendek dan tidak bagus maka lebih lambat untuk menganalisisnya serta tidak cocok untuk pencarian *binary string*.

2.6 Android

Menurut Nazrudin Safaat H (2011: 1), “Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi”. Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri.

2.6.1 Java

Menurut definisi *Sun Microsystem*, M. Shalahuddin dan Rosa A.S. (2010 : 1) : Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*standalone*) ataupun pada lingkungan jaringan.

2.6.2 Android Studio

Merupakan sebuah *tools Intergrated Development Environment (IDE)* sebagai alur jalan aplikasi pada platform android. Bahasa pemrograman yang digunakan adalah Java, sedangkan untuk penggunaan tampilan atau layout digunakan bahasa XML. Android Studio merupakan pengembangan yang lebih

eclipse, dikembangkan menjadi lebih kompleks dan profesional dengan tersedianya IDE serta Android SDK Tools.

2.7 *Waterfall*

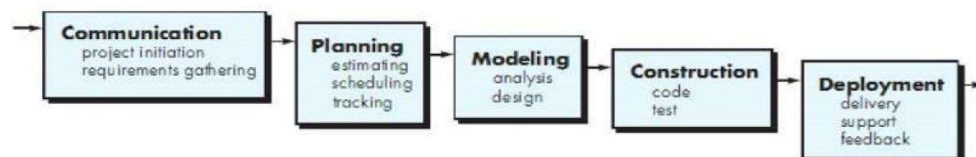
Menurut Pressman (2015), *Waterfall* yaitu:

“Model yang klasik dalam sistematis, berurutan dalam membangun *software*. Dari nama metode yang sebenarnya adalah “*Linear Sequential Model*” dan sering disebut juga dengan “*Classic Life Cycle*” atau metode *waterfall*”.

Waterfall bukanlah merupakan sesuatu yang lengkap untuk mengembangkan sebuah perangkat, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *waterfall* dibuat untuk memenuhi kebutuhan pengguna dan pada saat yang sama memungkinkan pengembang untuk lebih memahami kebutuhan pengguna secara lebih baik. Berikut adalah tahapan dalam metode *waterfall* :

1. Komunikasi dan pengumpulan data awal, yaitu analisis terhadap kebutuhan pengguna.
2. Perencanaan yaitu rencana kegiatan-kegiatan sebelum dilakukannya pemodelan dan pembentukan sistem.
3. Pemodelan digunakan untuk menggambarkan suatu sistem yang akan dikembangkan dengan menggunakan *software* dan *Interface*.
4. Pembentukan yaitu proses pengkodean untuk membentuk dari gambaran pemodelan yang dirancang.
5. Penyerahan dan umpan balik yaitu hasil pengujian yang telah dilakukan jika layak lalu dilakukan penyerahan ke *client* dan dilakukan tahap evaluasi dari hasil umpan balik saat penerapan.

Pressman (2015) menggambarkan metode pengembangan sistem yaitu *Waterfall* pada gambar 2.1 :



Gambar 2.1 Metode Pengembang Sistem Presman (2015)

Terdapat penjelasan dan langkah-langkah yang dilakukan dalam pengembangan sistem secara garis besar adalah :

2.8 Unified Modelling Language (UML)

Menurut Rosa & Salahuddin (2013), *Unified Modelling Language (UML)* yaitu:

“Bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *Unified Modelling Language (UML)* :

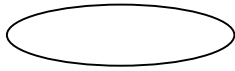
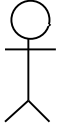
2.8.1 Use Case Diagram


Menurut Rosa & Salahuddin (2013), *Use Case* yaitu:

“*Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut”.

Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.2 berikut ini:

Tabel 2.2 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<i>Use case</i> : Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor: seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.	Asosiasi (<i>association</i>): merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada

		<i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		Generalisasi (<i>generalization</i>): merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum
5.	<< Include >> ▼	<i>Include</i> berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.
6.	<<Extend>> ▼	Ekstensi (<i>extend</i>) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

Sumber: Rosa & Salahuddin (2013)



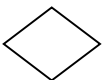

2.8.2 Activity Diagram


Menurut Rosa & Salahuddin (2013), *activity diagram* adalah:

”*Activity Diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem”.

Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.3 berikut ini:

Tabel 2.3 Simbol *Activity Diagram*

No	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

5.	Nama swimlane	Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber: (Rosa & Salahuddin, 2013)



2.8.3 Class Diagram




Menurut Rosa & Salahuddin (2013), *Class Diagram* adalah:

“*Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.4 berikut ini:

Tabel 2.4 Simbol *Class Diagram*

No	Simbol	Deskripsi
1.	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p>Nama_kelas</p> <hr/> <p>+Atribut</p> <hr/> <p>+Operasi</p> </div>	Kelas pada struktur sistem.
2.	<p style="text-align: center;"><i>Antar Muka/Interface</i></p> <div style="text-align: center;">  </div> <p style="text-align: center;">Nama_Interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	<p style="text-align: center;"><i>Asosiasi / Asociation</i></p>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan simbol
4.	<p style="text-align: center;"><i>Asosiasi Berarah / Directed Association</i></p> <div style="text-align: center;">  </div>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan simbol.

5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Ketergantungan / <i>dependency</i></p> 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	<p>Agregasi / <i>aggregation</i></p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Sumber: Rosa & Salahuddin (2013)

2.8.4 *Balsamiq Mockups*

Balsamiq Mockups menurut Faranello (2012) adalah salah satu software yang digunakan dalam pembuatan desain atau *prototyping* dalam pembuatan tampilan *user interface* sebuah aplikasi. Dengan menggunakan *Balsamiq Mockups* kita dimudahkan dalam pembuatan *user interface* karena *Balsamiq Mockup* sudah menyediakan *tools* yang dapat memudahkan dalam membuat desain *prototyping* aplikasi yang akan kita buat. *Software* ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna.

Balsamiq merupakan aplikasi yang disediakan untuk para *designer* guna mendesign *mockups*, dimana *mockups* adalah sebagai sebuah model dari suatu struktur atau alat baik *full size* ataupun berupa miniatur yang digunakan untuk pembelajaran, demo, *test* desain, promosi, dsb.

2.9 Metode Pengujian Sistem

Metode pengujian sistem merupakan metode yang digunakan untuk melakukan testing pada sistem yang dibangun sehingga di peroleh hasil berupa sistem yang sesuai fungsinya.

2.10 ISO 9126

Salah satu tolak ukur kualitas perangkat lunak adalah ISO 9126, yang dibuat oleh *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. ISO 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu, dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk software. Standar ISO 9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kunci kualitas untuk perangkat lunak komputer. Menurut (Abran et al. 2008), ISO 9126 adalah standar internasional yang diterbitkan oleh ISO untuk evaluasi kualitas perangkat lunak dan merupakan pengembangan dari ISO 9001. Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut (Al-Qutaish 2010,171) :

- 1) *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.
- 2) *Reliability* (Kehandalan). Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.
- 3) *Usability* (Kebergunaan). Kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.
- 4) *Efficiency* (Efisiensi). Kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut.
- 5) *Maintainability* (Pemeliharaan). Kemampuan perangkat lunak untuk dimodifikasi. Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional.
- 6) *Portability* (Portabilitas). Kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.