

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Berdasarkan penelitian yang dilakukan oleh Dine Tiara Kusum (2021) dengan judul “Fast Fourier Transform (FFT) Dalam Transformasi Sinyal Frekuensi Suara Sebagai Upaya Perolehan Average Energy (AE) Musik” Dalam sistem temu kembali informasi dari sebuah musik diperlukan proses ekstraksi musik menjadi sinyal frekuensi suara. Tujuan dari penelitian ini adalah untuk mencari nilai Average Energy (AE) yang merupakan salah satu parameter yang digunakan dalam information music retrieval. Fast Fourier Transform (FFT) adalah metode yang digunakan untuk mentransformasi sinyal suara menjadi spectrum suara berbasis sinyal frekuensi yang dapat digunakan dalam proses analisa data suara lebih lanjut. Berdasarkan ujicoba dari musik yang digunakan pada penelitian ini diperoleh nilai Average Energy (AE) sebesar 0.00044998

Berdasarkan penelitian yang dilakukan Umam Hidayaturrohman (2020) dengan judul “Implementasi Transformasi Fourier Untuk Transformasi Domain Waktu Ke Domain Frekuensi Pada Luaran Purwarupa Alat Pendeteksian Gula Darah Secara Non-Invasif*” Transformasi Fourier pertama kali diperkenalkan oleh matematikawan dan fisikawan asal Prancis yaitu Josep Fourier (1768-1830). Metode ini dianggap yang terbaik dalam hal analisis spektrum (Kim, 2003). Secara

garis besar, metode Transformasi Fourier dibagi menjadi 2 bagian yaitu Fourier Kontinu dan Fourier Diskret. Transformasi Fourier Kontinu banyak dilakukan pada sistem yang sederhana dan data yang sedikit karena memberikan hasil yang sangat baik. Jika sudah kompleks dan data yang melibatkan banyak, penerapannya akan sulit dilakukan bahkan tidak dapat untuk diselesaikan

Berdasarkan penelitian yang dilakukan Prativi Nugraheni Hanggarsari (2017) dengan judul “Simulasi Sistem Pengacakan Sinyal Suara Secara Realtime Berbasis Fast Fourier Transform (FFT)”, Dalam penelitian ini, telah dilakukan simulasi data pengacakan sinyal suara berbasis FFT. Sistem ini melakukan perekaman sinyal suara secara realtime sebagai sinyal input melalui microphone yang tersedia pada perangkat komputer. Dalam proses enkripsi, sinyal informasi yang berupa sinyal analog (domain waktu) dikonversikan ke dalam domain frekuensi menggunakan algoritma Fast Fourier Transform (FFT) 16 titik dan 32 titik. Sistem pembicaraan aman dengan menggunakan FFT 32 titik memiliki waktu proses yang lebih lama dibandingkan FFT 16 titik. Untuk sistem yang menggunakan FFT 16 titik membutuhkan waktu rata-rata 0,18564 detik sedangkan pada FFT 32 titik membutuhkan waktu rata-rata 2,70816 detik dengan selisih waktu 2.55252 detik.

Berdasarkan penelitian dari Yuri Ariyanto (2018) dengan judul “Steganografi Menggunakan Metode Discrete Fourier Transform (DFT)” Hal ini dilakukan selain memberikan jaminan keamanan terhadap gambar, tapi juga tidak mengurangi estetika pada gambar tersebut. Analisa yang dilakukan adalah tingkat

keberhasilan proses insertion dan extraction, serangan pada citra, uji kemiripan dengan pengujian NPCR (Number of Pixel of Change Rate), UACI (Unified Averaged Changed Intensity), dan PSNR (Peak Signal-to-Noise Ratio) pada proses insertion dan extraction. DFT disimpulkan aman terhadap serangan berupa cropping, resize, dan editing. Selain itu, dihasilkan nilai presentase perubahan yang rendah pada pengujian NPCR & UACI dan nilai yang tinggi pada pengujian PSNR.

II.2. Landasan Teori

II.2.1. Perancangan Sistem

Perancangan sistem dapat didefinisikan yaitu merumuskan bagaimana cara membentuk suatu struktur. Perancangan sistem juga dapat diartikan sebagai berikut :

Tahap setelah analisis dari siklus pengembangan sistem.

1. Pendefinisian dari kebutuhan-kebutuhan fungsional.
2. Persiapan untuk rancang bangun implementasi.
3. Menggambarkan bagaimana suatu sistem dibentuk.
4. Yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesalahan yang utuh dan berfungsi.

Termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem. (Andy Victor, 2018).

II.2.2. Discrete Fourier Transform

DFT merupakan prosedur matematika yang digunakan untuk menentukan harmonik atau frekuensi yang merupakan isi dari urutan sinyal diskrit. Urutan sinyal diskrit adalah urutan nilai yang diperoleh dari sampling periodik sinyal kontinu dalam domain waktu . DFT berasal dari fungsi Transformasi Fourier $X(f)$ yang didefinisikan: (Yuri Ariyanto : 2018)

Dimana:

N = jumlah sampel input

$X(m)$ = urutan ke- m komponen output DFT($X(0), X(1), \dots, X(N-1)$)

m = indeks output DFT dalam domain frekwensi ($0, 1, \dots, N-1$)

$x(n)$ = urutan ke- n sampel input ($x(0), x(1), \dots, x(N-1)$)

n = indeks sampel input dalam domain waktu ($0, 1, \dots, N-1$)

II.2.3 Website

Website adalah halaman informasi yang disediakan melalui jalur internet dan memiliki fasilitas hiperteks untuk menampilkan data berupa teks, gambar, suara, animasi, dan data multimedia lain nya sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. *Website* dapat diartikan sekumpulan halaman situs yang saling berhubungan menampilkan berbagai macam informasi maupun data berupa teks, gambar diam ataupun bergerak, data

animasi, suara, video maupun gabungan dari semuanya yang bisa diakses dimana saja kapan saja diseluruh dunia. Dimana *website* dikategorikan menjadi dua, yaitu:

1. *Web Statis*

Adalah web yang berisi/menampilkan informasi-informasi yang sifatnya *statis* (tetap). Disebut statis karena pengguna tidak dapat berinteraksi dengan *web* tersebut. Pengguna hanya dapat melihat isi dokumen pada halaman web dan apabila diklik akan berpindah kehalaman web yang lain. Interaksi pengguna terbatas hanya melihat informasi yang ditampilkan, tetapi tidak bisa mengolah informasi yang dihasilkan. Biasanya merupakan *HTML* yang ditulis pada editor teks dan disimpan dalam bentuk *.html* atau *.htm*.

2. *Web Dinamis*

Adalah *web* yang menampilkan informasi serta dapat berinteraksi dengan pengguna. *Web* yang dinamis memungkinkan pengguna untuk berinteraksi dengan *form* sehingga dapat mengolah informasi yang ditampilkan. Bersifat teraktif, tidak kaku, dan terlihat lebih indah. (Andi Ridho Rachman, 2018).

II.2.4 Basis Data (*Database*)

Basis data atau *database* merupakan koleksi dari data-data yang terorganisir dengan rapi sehingga data dapat dengan mudah disimpan dan dimanipulasi. Salah satu tujuan dari *database* adalah memberikan pengguna suatu

pandangan abstrak dari data, yaitu sistem menyembunyikan rincian bagaimana data disimpan dan dipelihara. Sistem *database* harus dibuat semudah mungkin untuk dimengerti karena kebanyakan pengguna sistem *database* adalah orang-orang yang kurang terlatih di bidang teknologi. (Ni Kadek Ceryna : 2018 : 102)

II.2.5. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat men-*download* langsung dari *web* resminya. (Randi Palit, 2018).

II.2.6. MySQL

MySQL (My Structure Query Language) merupakan sebuah program pembuat *database* yang bersifat *Open Source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *Windows* maupun *linux*. *MySQL* juga merupakan sebuah perangkat lunak sistem manajemen basis

data *SQL* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi user. *MySQL* juga sering dikenal dengan nama sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam table yang terpisah. Tabel -table tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa table dalam suatu permintaan. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan sebagainya dapat digunakan aplikasi berbasis web seperti *PHP MyAdmin* dengan aplikasi *XAMPP*. (Muhammad Faisal Widad : 2018)

II.2.7. Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana. Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi (Aditya : 2020)

II.2.8. *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, TMobile, 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 (Harni Kusniyati Dan Dicky : 2018)

Android merupakan sistem operasi *mobile*. *Android* tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke *hardware*, maupun data-data ponsel sekalipun, atau data sistem sendiri. *Android* merupakan sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Beberapa pengertian lain dari *Android*, yaitu :

1. Merupakan *platform* terbuka (*Open Source*) bagi para pengembang (*Programmer*) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli *Google Inc.* dari *Android Inc.*
3. Bukan bahasa pemrograman, tetapi hanya menyediakan lingkungan hidup atau *run time enviroment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk alat/*device* dengan sistem memori yang kecil. (Ni Kadek Ceryna, dkk: 2018 : 101)

Android merupakan sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh (*mobile*) seperti *smartphone* dan *tablet*. *Android* adalah sistem operasi yang bersifat *open source* dan *google* merilis kodenya di bawah lisensi *apache*. *Android* awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan finansial dari *google* dan kemudian *google* membelinya pada tahun 2005. *Android* dibuat menggunakan bahasa pemrograman C(core), C++, Java(UI) dan dukungan berbagai bahasa pemrograman lain. Berdasarkan *dashboard developer android*, *Android 8.0/1 Oreo* adalah versi *Android* yang paling banyak digunakan, yakni sekitar 28.3% dari keseluruhan perangkat *Android* di seluruh dunia. Hingga saat ini, *Android* sistem operasi berbasis *mobile* yang paling banyak digunakan di seluruh dunia dan android sudah di rilis versi 10 pada 3 september 2019. (Roberto Kaban, dkk: 2019)

II.2.9.PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, *Java*, dan *Perl* serta mudah untuk dipelajari. PHP merupakan bahasa *scripting server side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* yang akan menterjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain PHP adalah *akronim* dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML”. PHP (singkatan rekursif: *PHP Hypertext Preprocessor*) merupakan script yang dapat disisipkan ke dalam HTML. PHP dipakai untuk membuat sistem berbasis web yang dinamis. PHP bahasa pemrograman yang *server-side* sehingga program tersebut akan dijalankan atau diproses oleh server . Hingga saat ini PHP sudah rilis versi 7.4.1 pada 18 desember 2019. (Roberto kaban : 2019)

II.2.10. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan *visual* yang memungkinkan pengembang

sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi.

Unified Modeling Language (UML) biasa digunakan untuk :

- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *functionality* dengan *stereotypes*. (Omni Alfina dan Fitriana Harahap : 2019)

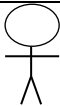
UML merupakan singkatan dari *Unified Modeling Language* yang merupakan sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan mempelajari kode program.

UML juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu developer ke *developer* lainnya. (Janiver W. Janis : 2020)

1. Use case Diagram

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini:

Tabel II.1. Simbol Use Case

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan

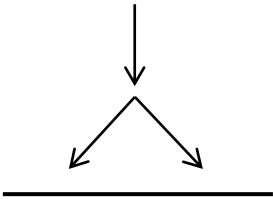
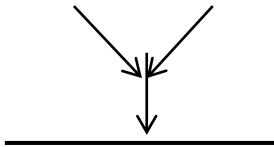
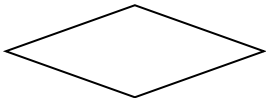
	dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Janiver W. Janis : 2020)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

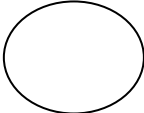
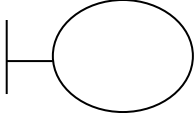
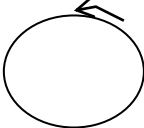
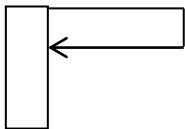
(Sumber : Janiver W. Janis : 2020)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar

objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding

	lurus dengan durasi aktivitas sebuah operasi.
---	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Janiver W. Janis : 2020)

4. Class Diagram (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/ Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih

0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Janiver W. Janis : 2020)