

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Sebelumnya

Berikut ini adalah penelitian-penelitian yang pernah dilakukan sebelum nya :

1. Penelitian yang dilakukan oleh Helert Sinaga, dkk (2018) mengangkat judul “Perbandingan Algoritma Huffman dan Run Length Encoding Untuk Kompresi File Audio”, pada penelitian disimpulkan bahwa pada file audio berformat *.WAV dan *.mp3, algoritma Huffman dapat menghasilkan rasio kompresi dengan persentase yang lebih kecil dibandingkan dengan algoritma *Run Length Encoding*. Penelitian yang dilakukan oleh Helert Sinaga ini tidak jelaskan menggunakan aplikasi atau membangun aplikasi tertentu sehingga dapat disimpulkan bahwa penelitian ini hanya melakukan analisa proses kompresi dengan melakukan perhitungan-perhitungan secara manual terhadap berapa sampel file yang memiliki ukuran yang berbeda-beda, sedangkan pada penelitian yang akan dilakukan akan menghasilkan aplikasi kompresi file WAV berbasis android.
2. Penelitian selanjutnya yang dilakukan oleh Nur Aisyah dan Soeb Aripin (2020), yang mengangkat judul “Penerapan Algoritma Elias Omega Code Pada Kompresi File Audio Aplikasi Murottal Muzzamil Hasbalah”. Penelitian ini menghasilkan aplikasi berbasis android yang berfungsi untuk melakukan kompresi terhadap file audio (Mp3) Murottal Muzzamil

Hasbalah, file-file tersebut telah dimasukan kedalam aplikasi sehingga pengguna tinggal memilih file yang akan di kompresi dan tidak tersedia form untuk memasukan file lain untuk dilakukan kompresi, tentunya hal ini akan memungkinkan pengguna dari aplikasi melakukan proses kompresi secara berulang terhadap file-file yang sama yang pada akhirnya akan menghasilkan tumpukan file yang akan menyebabkan ruang penyimpanan pada perangkat menjadi cepat penuh, berbeda dengan aplikasi yang akan dirancang didalam penelitian ini, pada penelitian yang akan dilakukan pengguna aplikasi dapat memilih file WAV apa saja yang akan dikompresi, sehingga dapat menghasilkan file-file kompresi baru.

3. Penelitian yang dilakukan oleh Nadia Fariza Rizky, dkk (2020) dengan judul penelitian “Penerapan Algoritma Elias Delta Codes Dalam Kompresi File Teks”. Penelitian yang dilakukan Nadia Fariza Rizky, dkk ini menghasilkan sebuah aplikasi berbasis *desktop* yang dibangun dengan menggunakan aplikasi Visual Studio 2008 yang hanya dapat melakukan proses kompresi terhadap file berekstensi txt dan tidak dapat melakukan kompresi terhadap file lain seperti Ms.Word sehingga dapat dikatakan aplikasi dari hasil penelitian tersebut hanya memiliki kemampuan yang sangat terbatas, berbeda dengan aplikasi yang akan dihasilkan pada penelitian yang akan dilakukan. Pada aplikasi yang akan dilakukan proses kompresi dapat dilakukan pada perangkat *Smartphone* android serta dapat melakukan kompresi terhadap file audio WAV.

II.2 Pengertian Kompresi

Kompresi merupakan proses pengubahan sekumpulan data menjadi suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data. Kompresi data adalah proses mengkodekan informasi menggunakan bit atau information-bearing unit yang lain yang lebih rendah dari pada representasi data yang tidak terkodekan dengan suatu sistem *encoding* tertentu. (Nadia F.R Dkk, 2020)

Ukuran file atau berkas yang besar bukan saja menjadi masalah dalam hal penyimpanan, namun juga menjadi masalah lain saat komunikasi antar komputer. Pemakaian data dengan ukuran lebih besar akan memakan waktu transfer yang lebih lama di bandingkan dengan data yang memiliki ukuran yang lebih kecil. Oleh karena itu untuk mengatasi masalah ini dapat menggunakan kompresi data.

Secara umum, metode kompresi data dapat dibagi ke dalam dua kelompok yaitu *lossy* dan *losseless*. *Lossy* teknik kompresi data dimana hasil kompresi dari data yang terkompresi tidak sama dengan data aslinya karena ada informasi yang hilang, namun sudah cukup untuk digunakan. *Lossless* Teknik kompresi dimana data hasil kompresi dapat dikompres lagi dengan hasil yang sama dari data sebelumnya.(Ledi Varia Simanjuntak, 2020).

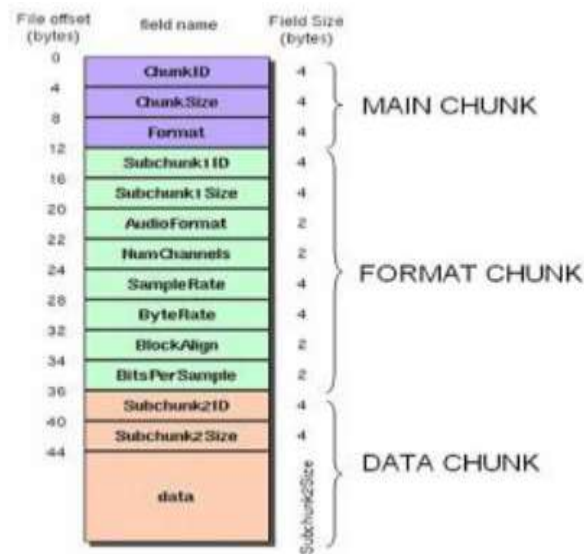
II.3. Audio

Audio merupakan salah satu elemen dalam multimedia, di mana elemen ini dapat dirasakan dengan indera pendengaran. *Audio* terdiri dari kata yang diucap, suara-suara, musik dan bahkan gangguan (*noise*). *Audio* sendiri adalah gelombang

yang merambat akibat dari tekanan yang melalui beberapa media (benda padat, cair, atau gas). Suara terdiri dari frekuensi dan dapat didengar dalam jangkauan pendengaran tingkatan yang cukup kuat untuk didengar. (Nur Aisyah dan Soeb Aripin, 2020).

II.3.1. WAV

WAV adalah singkatan dari istilah dalam bahasa Inggris *wave form audio format* merupakan standar format berkas audio yang dikembangkan oleh Microsoft dan IBM. WAV merupakan varian dari format *bit stream* RIFF dan mirip dengan format IFF dan AIFF yang digunakan komputer Amiga dan Macintosh. Baik WAV maupun AIFF kompatibel dengan sistem operasi Windows dan Macintosh. Walaupun WAV dapat menampung audio dalam bentuk terkompresi, walaupun pada umumnya format WAV merupakan audio yang tidak terkompresi. Format file WAV merupakan bagian dari spesifikasi RIFF milik Microsoft yang digunakan untuk penyimpanan file-file multimedia. File WAV dimulai dengan bagian *header* dan diikuti oleh rentetan data *chunk*. File WAV terdiri dari 3 bagian yaitu *main chunk*, *format chunk*, dan *data chunk*. Sinyal suara yang direpresentasikan file WAV dalam bentuk *discrete*, berupa deret bilangan yang merepresentasikan *amplitude* dalam domain waktu pada bagian file *header* terdapat informasi tentang file WAV tersebut, diantaranya menyatakan nilai *sample rate*, jumlah *channel*, dan *bit per sample*. Dari keterangan pada file *header* tersebut dapat diketahui berapa sampel yang dicuplik dari sinyal analog tiap detik. Struktur file WAV dapat dilihat pada gambar berikut:



Gambar III.1 Struktur File WAV
(Sumber : Muhammad Abdul Rivai Hi Wahab; 2017)

II.4. Algoritma *Elias Delta Code*

Elias delta code adalah sebuah algoritma kompresi yang dibuat oleh Peter Elias. Kode elias gamma menambah panjang kode dalam *unary* (α). Kode berikutnya, δ (*delta*), kemudian ditambahkan panjang pada kode biner (β). Dengan demikian, *elias delta code*, yang juga untuk bilangan bulat positif, sedikit lebih kompleks untuk dibangun.

Adapun aturan untuk mengkodekan sebuah bilangan dengan menggunakan *elias delta code* adalah:

1. Tuliskan n dalam bilangan biner (*binary*). Bit yang paling kiri (paling signifikan) akan menjadi 1.
2. Hitung jumlah bit-nya, hapus bit paling kiri dari n dan tambahkan perhitungan dalam bilangan biner (*binary*) pada bagian kiri dari n setelah bit paling kiri dari n dihapus.

3. Hitung jumlah bit-nya, hapus bit paling kiri dari n dan tambahkan perhitungan dalam bilangan biner (*binary*) pada bagian kiri dari n setelah bit paling kiri dari n dihapus.
4. Kurangi 1 dari perhitungan pada langkah ke-2 dan tambahkan jumlah nol ke kode.

Ketika langkah-langkah ini diterapkan pada integer ke-17, hasilnya adalah : $17 = 10001$ (lima bit). Hapus angka 1 yang paling kiri dan tambahkan $5 = 101$ sehingga hasilnya $101|0001$. Tiga bit sudah ditambahkan, kemudian tambahkan 2 nol untuk mendapatkan kode delta $00|101|0001$.

Sehingga dihasilkan table perhitungan sebagai berikut :

Tabel II.1 Tabel Perhitungan Delta Code

$1=2^0+0 \rightarrow L =0 \rightarrow 1$	$10=2^3+2 \rightarrow L =3 \rightarrow 00100010$
$2=2^1+0 \rightarrow L =1 \rightarrow 0100$	$11=2^3+3 \rightarrow L =3 \rightarrow 00100011$
$3=2^1+1 \rightarrow L =1 \rightarrow 0101$	$12=2^3+4 \rightarrow L =3 \rightarrow 00100100$
$5=2^2+1 \rightarrow L =2 \rightarrow 01101$	$14=2^3+6 \rightarrow L =3 \rightarrow 00100110$
$6=2^2+2 \rightarrow L =2 \rightarrow 0110$	$15=2^3+7 \rightarrow L =3 \rightarrow 00100111$
$7=2^2+3 \rightarrow L =2 \rightarrow 01111$	$16=2^4+0 \rightarrow L =4 \rightarrow 001010000$
$8=2^3+0 \rightarrow L =3 \rightarrow 00100000$	$17=2^4+1 \rightarrow L =4 \rightarrow 001010001$
$9=2^3+1 \rightarrow L =3 \rightarrow 00100001$	$18=2^4+2 \rightarrow L =4 \rightarrow 001010010$

Decode dengan *elias delta code* dilakukan dalam langkah-langkah berikut:

1. Baca bit dari kode sampai proses *decode* dengan *elias gamma code* dapat dilakukan. Proses ini dapat dilakukan dengan beberapa langkah berikut ini:
2. Hitung jumlah nol terdepan dari kode tersebut lalu gantikan perhitungan tersebut dengan C .
3. Periksa bit bagian kiri $2C + 1$ (C nol, diikuti dengan 1, lalu diikuti dengan bit C selebihnya). Ini merupakan *decode elias gamma code* $M + 1$.
4. Baca bit M berikutnya. Sebut ini sebagai L .

5. Bilangan bulat yang di *decode* adalah $2M + L[8]$.

(Nadia F.R , dkk, 2020).

II.5. Android

Android merupakan sistem operasi yang dikembangkan untuk perangkat mobile berbasis Linux. Pada awalnya sistem operasi ini dikembangkan oleh Android Inc. yang kemudian dibeli oleh Google pada tahun 2005.

Dalam usaha mengembangkan Android, pada tahun 2007 dibentuklah Open Handset Alliance (OHA), sebuah konsorsium dari beberapa perusahaan, yaitu Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, dan T-Mobile dengan tujuan untuk mengembangkan standar terbuka untuk perangkat mobile. Pada tanggal 9 Desember 2008, diumumkan bahwa 14 orang anggota baru akan bergabung dengan proyek Android, termasuk PacketVideo, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc (Efmi Maiyana, 2018).

II.5.1. Android Studio

Android Studio merupakan sebuah IDE (*Integrated Development Environment*) untuk pengembangan aplikasi android, aplikasi ini dipublikasikan oleh Google pada tanggal 16 mei 2013 dan tersedia secara gratis dibawah lisensi

Apache 2.0, Android studio ini menggantikan *software* pengembangan android sebelumnya yaitu Eclipse. (Efmi Maiyana, 2018).

II.5.2. IDE (*Integrated Development Environment*)

Integrated Development Environment adalah aplikasi pengembang perangkat lunak dengan fungsi-fungsi terintegrasi yang dibutuhkan untuk membangun sebuah perangkat lunak seperti *code editor*, *debugger*, *compiler*, dan sebagainya. (Efmi Maiyana, 2018).

II.5.3. *Android Software Development Kit* (Android SDK)

Android SDK adalah *tools API* (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Pada Android SDK ini terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, kode contoh dan tutorial. SDK memungkinkan pengembang membuat aplikasi untuk *platform* Android SDK, Android mencakup proyek sampel dengan kode sumber, perangkat pengembangan, *emulator* dan perpustakaan yang diperlukan untuk membangun aplikasi Android. Aplikasi yang ditulis dengan bahasa pemrograman Java dan berjalan di Dalvik, mesin *virtual* yang dirancang khusus untuk penggunaan *embedded* yang berjalan diatas *kernel Linux*. (Efmi Maiyana, 2018).

II.5.4. *Android Virtual Device* (AVD)

Android Virtual Device merupakan emulator untuk menjalankan program aplikasi Android yang kita buat. AVD ini selanjutnya digunakan sebagai tempat

untuk test dan menjalankan aplikasi Android tanpa harus menggunakan perangkat Android yang sebenarnya. Sebelum menggunakan AVD harus menentukan karakteristiknya, misalkan dalam menentukan versi Android, jenis dan ukuran layar dan besarnya memori. AVD bisa dibuat sebanyak yang kita inginkan. (Efmi Maiyana, 2018).

II.6. *Unified Modeling Language (UML)*

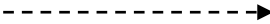
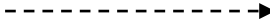
Unified Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. (Julianto Simatupang, 2019).

A. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Tabel II.2 Simbol *Use Case Diagram*

Simbol	Diskripsi
<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 diawal di awal frase nama <i>use case</i>.

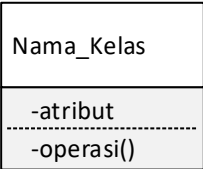
Aktor/actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Asosiasi /association 	Komunikasi antara aktor dan use case yang berpartisipasi pada <i>use case</i> atau use case memiliki interaksi dengan actor.
Ekstensi / <i>extend</i> <<extends>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan misal Arah panah mengarah pada <i>use case</i> yang ditambahkan.
Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum lainnya.
Menggunakan / <i>include</i> 	<i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan

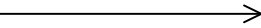
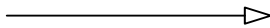
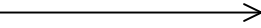
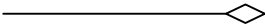
(Sumber : Julianto Simatupang, 2019)

B. Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Tabel II.3 Simbol *Class Diagram*

Simbol	Diskripsi
Kelas 	Kelas pada struktur sistem.

Antar muka/ <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi- generalisasi- spesialisasi (umum khusus).
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole- part</i>).

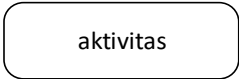
(Sumber : Julianto Simatupang, 2019)

C. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan *actor*.

Tabel II.4. Simbol Activity Diagram

Simbol	Diskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

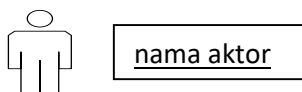
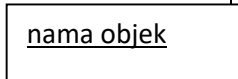
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

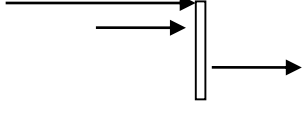
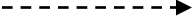
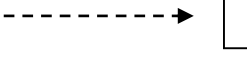
(Sumber : Julianto Simatupang, 2019)

D. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Tabel II.5. Simbol Sequence Diagram

Simbol	Diskripsi
Aktor  atau	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Garis hidup / <i>lifeline</i> : 	Menyatakan kehidupan suatu objek.
Objek 	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya

Pesan tipe <i>create</i> << <i>create</i> >> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
Pesan tipe <i>send</i> 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah yang mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

(Sumber : Julianto Simatupang, 2019)