

Penerapan Algoritma Fibonacci Codes Pada Kompresi Aplikasi Audio Mp3 Berbasis Dekstop

Jesica Martina Br. Panjaitan

¹Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: jesticapanjaitan11@gmail.com

Abstrak– Kebutuhan terhadap kapasitas penyimpanan yang semakin besar merupakan penyebab munculnya berbagai teknik kompresi. Ukuran data yang besar mengakibatkan pemborosan memori dan lambatnya proses pemindahan data. Kompresi bertujuan untuk mengurangi ukuran data tersebut menjadi sekecil mungkin. Dengan melakukan kompresi, data yang besar akan berkurang ukurannya sehingga dapat menghemat kapasitas penyimpanan. Penelitian ini dilakukan dengan maksud dan tujuan untuk menganalisis algoritma *Fibonacci Codes* dalam kompresi file audio yang berekstensi .mp3 dengan parameter perbandingan *Ratio of Compression*, *Compression Ratio* dan juga *Space Savings*. Kompresi dilakukan untuk memperkecil ukuran bit file audio tersebut. Karena jika ukuran file audio yang besar tidak di kompresi akan terjadi kelambatan pada saat pengiriman data. Dan juga dapat memenuhi kapasitas dari suatu memori penyimpanan. Maka dengan adanya sistem kompresi ini, diharapkan dapat menanggulangi masalah yang ada di masyarakat. Pada algoritma kompresi yang dapat didekompresi.

Kata Kunci: Kompresi, Dekompresi, File Audio, mp3, Fibonacci Code, Lossy

Abstract– The need for greater storage capacity is the cause of the emergence of various compression techniques. Large data sizes result in wasteful memory and slow data transfer processes. Compression aims to reduce the size of the data to the smallest possible size. By doing compression, large data will be reduced in size so as to save storage capacity. This research was conducted with the aim and aim of analyzing the Fibonacci Codes algorithm in compressing audio files with the extension. Mp3 with comparison parameters of Ratio of Compression, Compression Ratio and Space Savings. Compression is done to reduce the bit size of the audio file. Because if the size of a large audio file is not compressed, there will be a delay when sending data. And also can meet the capacity of a storage memory. So with this compression system, it is hoped that it can overcome the problems that exist in society. On the compression algorithm that can be decompressed.

Keywords: Compression, Decompression, Audio Files, mp3, Fibonacci Code, Lossy

1. PENDAHULUAN

Perkembangan teknologi yang sangat pesat di era globalisasi saat ini telah memberikan banyak manfaat dalam kemajuan di berbagai aspek. Penggunaan teknologi oleh manusia dalam membantu menyelesaikan pekerjaan merupakan hal yang menjadi keharusan dalam hidup. Namun, manusia sebagai pengguna teknologi harus bijak dalam penggunaan teknologi. Kebutuhan manusia terhadap teknologi sudah hampir tidak bisa terhindarkan lagi, begitu juga kebutuhan akan kapasitas penyimpanan.

File audio merupakan salah satu file multimedia yang sangat berkembang pesat terutama pada industri music yang semakin maju akibat perkembangan teknologi yang membuat banyak sekali ekstensi file audio yang berukuran besar yang mana dalam penelitian ini menggunakan file audio berekstensi *.mp3. Untuk itu diperlukan adanya aplikasi kompresi audio untuk memperkecil atau mengurangi ukuran file audio. Semakin kecil ukuran file audio maka akan mengurangi *delay time* pada pengiriman data, aplikasi ini juga dapat mengefisienkan salah satu ruang penyimpanan. Aplikasi ini dikembangkan pada sistem operasi yang berbasis desktop dengan menggunakan aplikasi *Microsoft Visual Basic 2008*. Masalah *size* data merupakan salah satu aspek penting dalam perkembangan teknologi informasi. Kebutuhan ini disebabkan oleh data yang harus disimpan semakin banyak, khususnya bagi penggemar music dan audio lainnya. Dan solusi untuk permasalahan tersebut adalah kompresi atau pemampatan *size* pada file audio tersebut [1].

Kompresi adalah perubahan data ke dalam bentuk yang memerlukan bit yang lebih sedikit, biasanya dilakukan agar data yang disimpan atau dikirimkan dengan lebih efisien. Jadi, sudah terlihat jelas bahwa kegunaan dari kompresi yaitu mengecilkan presentase penyimpanan pada media penyimpanan. Pengompresian sangat berguna karena dapat membantu mengurangi konsumsi sumber daya mahal seperti *hard disk space* atau transmisi *bandwidth*.

Proses kompresi data didasarkan pada kenyataan bahwa semua jenis dan selalu terdapat pengulangan pada komponen data yang dimilikinya, misalnya di dalam suatu teks kalimat akan terdapat pengulangan penggunaan huruf alphabet dari huruf a sampai dengan huruf z. Maka dari itu untuk meminimalisir masalah *size* tersebut maka dibutuhkan proses kompresi.

Ada banyak algoritma yang digunakan untuk pengompresian suatu data, yaitu salah satunya adalah Algoritma *Fibonacci Code* (Kode Fibonacci). Kode Fibonacci adalah *variable length code*, dimana bilangan bulat yang lebih kecil mendapatkan kode pendek. Kode berakhir dengan dua bit satu dan nilai yang didapatkan adalah jumlah dari nilai-nilai Fibonacci yang sesuai untuk bit yang ditetapkan (kecuali bit terakhir, yang merupakan akhir dari kode). Dan dengan menggunakan algoritma ini, maka kita akan dimudahkan dalam pengompresian data berupa audio.

2. METODOLOGI PENELITIAN

2.1 Kompresi

Salah satu kegunaan kompresi adalah untuk memperkecil ukuran file dalam memori atau media penyimpanan, agar media tersebut dapat dihemat. Kompresi data berarti suatu teknik untuk memampatkan data agar diperoleh data dengan ukuran yang lebih kecil daripada ukuran aslinya sehingga lebih efisien dalam penyimpanan serta mempersingkat waktu pertukaran data tersebut.

2.2 Algoritma Fibonacci Code

Bilangan Fibonacci pertama kali dikemukakan oleh seorang ilmuwan yang berasal dari Italia yang bernama Leonardo da Pisa atau dikenal sebagai Leonardo Pisano yang lahir kisaran pada tahun 1175 sampai 1250.

Beliau juga dikenal sebagai Fibonacci yang termasuk juga memiliki sebuah peran dalam mengenalkan sistem penulisan dan perhitungan bilangan Arab ke dunia Eropa. Leonardo ialah seseorang yang memperkenalkan bilangan deret. Kemudian setelah meninggal dunia, dia sering disebut sebagai Fibonacci (berasal dari kata *filius Bonacci* artinya anak dari Bonacci). Ayahnya yang bernama William atau dikenal sebagai Bonacci. Dari situlah kemudian Leonardo memiliki julukan Fibonacci yang merupakan berasal dari kata *Filius Bonacci* yang artinya anak dari Bonacci.

Sebelum barisan atau bilangan ini ditemukan oleh Leonardo da Pisa di dunia Barat, berdasarkan pada buku *The Art of Computer Programming* karya Donald E. Knuth, barisan ini pertama kali telah dijelaskan oleh seorang matematikawan yang dari negara India yang bernama Gopala dan Hemachandra pada tahun 1150, ketika menyelidiki berbagai kemungkinan untuk memasukkan barang-barang ke dalam suatu kantong.

Fibonacci merupakan ilmuwan yang telah menulis beberapa judul buku yang mana salah satu judul buku yang terkenal dan menjadi tonggak awal penggunaan angka Arab di Eropa ialah "Liber Abaci" yang ditulis sekitar pada tahun 1202. Sekarang deret Fibonacci adalah salah satu objek matematika terkenal. Bilangan Fibonacci didefinisikan sebagai barisan bilangan yang suku-sukunya merupakan penjumlahan 2suku sebelumnya. Bilangan Fibonacci dapat ditunjukkan sebagai barisan bilangan 0,1,1,2,3,5,8,13,21,34,55,89,144,233,...

Fibonacci Code, berkaitan erat dengan representasi Fibonacci dari bilangan bulat, kode Fibonacci dari bilangan bulat positif n adalah Fibonacci yang mempresentasikan n dengan tambahan 1 ditambahkan ke ujung kanan. Dengan demikian, Fibonacci kode 5 adalah 0001|1 dan bahwa dari 33 adalah 101010|1. Hal ini jelas bahwa kode tersebut berakhir dengan sepasang 11.

Tabel 1. Dua Belas *Fibonacci Codes*

1	11
2	011
3	0011
4	1011
5	00011
6	10011
7	01011
8	01011
9	000011
10	010011
11	001011
12	101011

Sumber : Smith, David.2011 [7]

Decoding, loncatan bit dari kode 11, kemudian ganti 11 dengan 1. Perkalian bit diabaikan oleh angka.....,13, 8, 5, 3,2,1 (angka Fibonacci) dan penambahan produk. Jelas tidak perlu melakukan perkalian, hanya menggunakan 1 bit untuk memilih angka Fibonacci yang tepat untuk menambahkannya.

$$F(n) = F(n-1) + F(n-2), \dots$$

Kode Fibonacci sangat panjang, tetapi memiliki keuntungan sehingga menjadi lebih kuat dari kebanyakan *variable length code*. Perubahan sebuah bit dalam *Fibonacci Code* dapat diganti oleh sepasang bit berurutan dari 01 atau 10 hingga 11 ke 10 atau 10. Dalam kasus yang pertama, kode dapat dibaca sebagai dua kode, sedangkan dalam kasus terakhir, dua kode akan diterjemahkan sebagai kode tunggal.

2.3 Parameter Analisis Kompresi

Pada suatu teknik yang digunakan dalam kompresi data terdapat beberapa factor atau variable yang biasa digunakan untuk menganalisa kualitas dari suatu teknik kompresi data tersebut, yaitu :

1. Ratio of Compression (RC) [8]

Ratio of Compression (RC) adalah perbandingan antara ukuran data sebelum dikompresi dengan ukuran dan setelah dikompresi.

$$RC = \frac{\text{ukuran data sebelum di kompresi}}{\text{ukuran data setelah di kompresi}} \times 100 \%$$

2. Compression Ratio (CR)

Compression Ratio (CR) adalah presentase besar data yang telah dikompresi yang didapat dari hasil perbandingan antara ukuran data setelah dikomopresi dengan ukuran data sebelum dikompresi.

$$CR = \frac{\text{ukuran data setelah di kompresi}}{\text{ukuran data sebelum di kompresi}} \times 100 \%$$

3. Space Savings (SS)

Space Savings (SS) merupakan presentase isi antara data yang belum di kompresi dengan besar data yang telah dikompresi

$$CR = 1 - \frac{\text{ukuran data setelah di kompresi}}{\text{ukuran data sebelum di kompresi}}$$

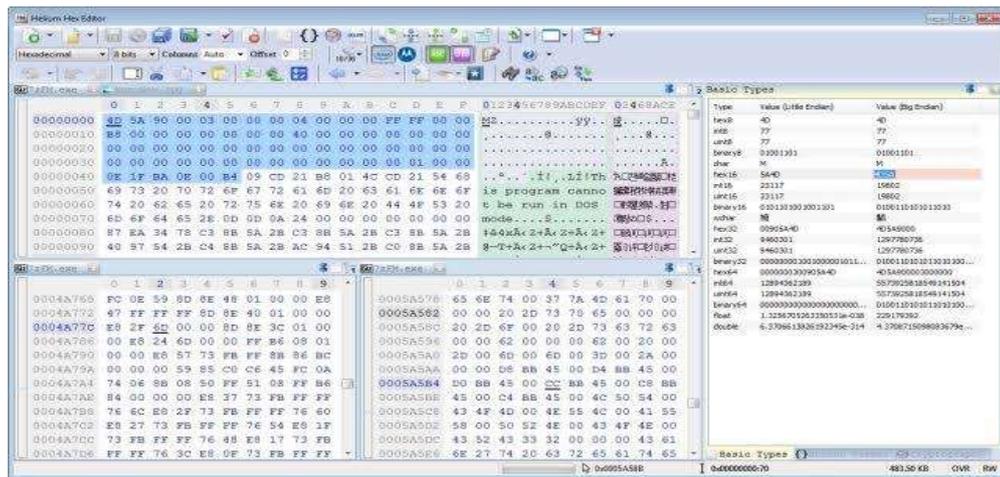
3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Analisa adalah usaha yang dilakukan untuk mengamati suatu hal dengan menyusun komponen pembentuknya atau menguraikan komponen tersebut agar dikaji dengan rinci. Semua hal dalam kehidupan manusia dapat dianalisa oleh manusia. Yang membedakan hanyalah metode dan cara menganalisanya. Dan dalam penelitian mengenai kompresi pada file. Analisa masalah merupakan tahapan pembangunan sistem untuk menentukan proses mengidentifikasi sebab dan akibat dibangunnya sebuah sistem agar sistem yang akan dibangun tersebut dapat berjalan sebagaimana mestinya sesuai dengan tujuan dari sistem tersebut. Permasalahan yang akan di angkat dari penelitian ini yang bagaimana memperkecil ukuran file audio dengan mengurangi adanya redudansi pada file yang memiliki ukuran yang lebih kecil guna untuk menghemat pemakain media penyimpanan data dan mempercepat transmisi atau pemindahan data. Mengompresi data dengan menggunakan algoritma Fibonacci Codes Pada penelitian file audio yang akan digunakan adalah *.mp3.

3.1.1. Penerapan Algoritma Fibonacci Codes

File audio yang akan dikompresi berukuran 6.807 Kb. Dari sampel mp3 di dapat nilai hexadecimal menggunakan bantuan software hex editor neo seperti pada gambar dibawah ini. Dari sample mp3 di dapat nilai hexadecimal menggunakan bantuansoftware hex editor seperti pada gambar di bawah ini :



Gambar 1. Nilai Hexadesimal File Mp3

Berdasarkan pada gambar di atas didapat nilai hexadecimal file mp3. Untuk keperluan hitungan manual hanya diambil sampel nilai sebanyak 16 karakter nilai hexadecimal file mp3. Nilai hexadecimal diambil dari sisi kiri atas sampai dengan bilangan ke 16.

Selanjutnya pembacaan isi file, adapun bilangan hexadecimal file mp3 sampel tersebut adalah 4D, 5A, 90, 00, 03, 00, 00, 00, 04, 00, 00, 00, FF, 00, 00.

Kemudian pemetaan ke tabel frekuensi, berdasarkan input nilai hexadecimal file audio mp3 tersebut maka setiap nilai akan dipetakan ke dalam tabel frekuensi. Kemunculan data yang digambarkan dalam tabel 1.

Tabel 1. Frekuensi Data

Nilai		Bit	Frekuensi	Bit * Frekuensi
Hex	Biner			
00	00000000	8	7	56
58	01011000	8	3	24
54	01010100	8	1	8

Nilai		Bit	Frekuensi	Bit * Frekuensi
Hex	Biner			
49	01000101	8	1	8
44	01000100	8	1	8
33	00110011	8	1	8
04	00000100	8	1	8
01	00000001	8	1	8
Total				128

Berdasarkan tabel di atas, satu nilai hexadecimal (karakter) bernilai delapan bit bilangan biner. Sehingga 16 bilangan hexadecimal mempunyai nilai biner sebanyak 128 bit. Untuk mengubah satuan menjadi byte maka jumlah keseluruhan bit dibagikan 8. Maka dapat dihasilkan $128/8 = 16$ byte

Masukan rumus untuk nilai hexadecimal. Langkah-langkah pembentukan sebuah kode Fibonacci untuk nilai sampel 49, 44, 33, 04, 00, 00, 00, 00, 01, 00, 54, 58, 58, 58, 00, 00 adalah sebagai berikut :

Nilai $n = 49$: Bilangan bulat positif

Deret Fibonacci yang mendekati nilai n adalah : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34. Proses pengurangan nilai n dengan setiap bilangan Fibonacci sampai sisa pengurangan nilai n adalah 0 :

1. $49-34 = 15$: Nilai n dikurangkan dengan delapan bilangan Fibonacci terbesar yang lebih kecil atau sama dengan nilai n dan catat sisa pengurangan.
2. $15-13 = 2$: Tukarkan nilai n dari sisa pengurangans sebelumnya lalu kurangkan nilai n dengan bilangan Fibonacci yang lebih kecil atau sama dengan nilai n dan catat sisa pengurangan.
3. $2-2 = 0$: Tukarkan nilai n dari sisa pengurangan sebelumnya lalu kurangkan nilai n dengan bilangan Fibonacci yang lebih kecil atau sama dengan nilai n .

Tabel 2. Proses Pembentukan Kode Fibonacci Nilai 49

Urutan Bilangan Fibonacci	F (0)	F (1)	F (2)	F (3)	F (4)	F (5)	F (6)	F (7)	F (8)	F (9)
Bilangan Fibonacci	0	1	1	2	3	5	8	13	21	34
Kode Fibonacci Sementara	-	-	0	1	0	0	0	1	0	1

Keterangan :

Jika bilangan yang dikurangkan adalah bilangan yang terdapat dalam deret Fibonacci $F(i)$, tambahkan angka “1” dan bilangan deret Fibonacci $F(i)$ yang tidak dikurangkan tambahkan angka “0”.

Untuk nilai n berikutnya dilakukan dengan langkah-langkah di atas.

Nilai $n = 44$

Deret Fibonacci yang mendekati nilai n adalah : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Proses pengurangan nilai n dengan setiap bilangan Fibonacci :

1. $44 - 34 = 10$
2. $10 - 8 = 2$
3. $2 - 2 = 0$

Tabel 3. Proses Pembentukan Kode Fibonacci Nilai 44

Urutan Bilangan Fibonacci	F (0)	F (1)	F (2)	F (3)	F (4)	F (5)	F (6)	F (7)	F (8)	F (9)
Bilangan Fibonacci	0	1	1	2	3	5	8	13	21	34
Kode Fibonacci Sementara	-	-	0	1	0	0	1	0	0	1

Nilai $n = 33$

Deret Fibonacci yang mendekati nilai n adalah :

0, 1, 1, 2, 3, 5, 8, 13, 21

Proses pengurangan nilai n dengan setiap bilangan Fibonacci :

1. $33 - 21 = 12$
2. $12 - 8 = 4$
3. $4 - 3 = 1$
4. $1 - 1 = 0$

Tabel 4. Proses pembentukan kode Fibonacci nilai 33

Urutan Bilangan Fibonacci	F (0)	F (1)	F (2)	F (3)	F (4)	F (5)	F (6)	F (7)	F (8)	F (9)
Bilangan Fibonacci	0	1	1	2	3	5	8	13	21	-
Kode Fibonacci Sementara	-	-	0	1	0	0	1	0	1	-

Nilai n = 04

Deret Fibonacci yang mendekati nilai n adalah : 0, 1, 1, 2, 3

Proses pengurangan nilai n dengan setiap bilangan Fibonacci :

1. $4 - 3 = 1$
2. $1 - 1 = 0$

Tabel 5. Proses Pembentukan Kode Fibonacci Nilai 4

Urutan Bilangan Fibonacci	F (0)	F (1)	F (2)	F (3)	F (4)	F (5)	F (6)	F (7)	F (8)	F (9)
Bilangan Fibonacci	0	1	1	2	3	-	-	-	-	-
Kode Fibonacci Sementara	-	-	1	0	1	-	-	-	-	-

Nilai n = 01

Deret Fibonacci yang mendekati nilai n adalah : 0,1

Proses pengurangan nilai n dengan setiap bilangan Fibonacci :

1. $1 - 1 = 0$

Tabel 6. Proses Pembentukan Kode Fibonacci Nilai 1

Urutan Bilangan Fibonacci	F (0)	F (1)	F (2)	F (3)	F (4)	F (5)	F (6)	F (7)	F (8)	F (9)
Bilangan Fibonacci	0	1	1	-	-	-	-	-	-	-
Kode Fibonacci Sementara	-	-	1	-	-	-	-	-	-	-

Nilai n = 54

Deret Fibonacci yang mendekati nilai n adalah : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Proses pengurangan nilai n dengan setiap bilangan Fibonacci :

1. $54 - 34 = 20$
2. $20 - 13 = 7$
3. $7 - 5 = 2$
4. $2 - 2 = 0$

Tabel 7. Proses pembentukan kode Fibonacci nilai 54

Urutan Bilangan Fibonacci	F (0)	F (1)	F (2)	F (3)	F (4)	F (5)	F (6)	F (7)	F (8)	F (9)
Bilangan Fibonacci	0	1	1	2	3	5	8	13	21	34
Kode Fibonacci Sementara	-	-	1	0	0	1	0	1	0	1

Nilai n = 58

Deret bilangan Fibonacci yang mendekati nilai n adalah : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Proses pengurangan nilai n dengan setiap bilangan Fibonacci :

1. $58 - 34 = 24$
2. $24 - 21 = 3$
3. $3 - 2 = 1$
4. $1 - 1 = 0$

Tabel 8. Proses pembentukan kode Fibonacci nilai 58

Urutan Bilangan Fibonacci	F (0)	F (1)	F (2)	F (3)	F (4)	F (5)	F (6)	F (7)	F (8)	F (9)
Bilangan Fibonacci	0	1	1	2	3	5	8	13	21	34
Kode Fibonacci Sementara	-	-	1	1	0	0	0	0	1	1

Maka dari hasil perhitungan tersebut didapatkan hasil kompresi seperti yang tertera di tabel 3.9 :

Tabel 9. Hasil akhir kode Fibonacci

Hex	Nilai		Bit	Frekuensi	Bit * Frekuensi
	Hex	Biner			
00	0	0	0	7	0
58	11000011	8	8	3	24
54	10010101	8	8	1	8
49	01000101	8	8	1	8
44	01001001	8	8	1	8
33	1010101	7	7	1	7
04	101	3	3	1	3
01	1	1	1	1	8
Total					66

Dari perhitungan tabel diatas setelah dikompresi dengan menggunakan Fibonacci code adalah 66 bit. Untuk diubah menjadi satuan byte maka dibagi 8. $66 / 8 = 8,25$ byte.

1. Ratio of Compression (RC)

$$RC \text{ Fibonacci Codes} = \frac{128}{66} = 1,93$$

2. Compression Ratio (CR)

$$CR \text{ Fibonacci Codes} = \frac{66}{128} \times 100 \% = 51,56 \%$$

3. Space Saving (SS)

$$SS \text{ Fibonacci Codes} = 100 \% - 51,26 \% = 48,44 \%$$

4. KESIMPULAN

Proses kompresi yang dilakukan dengan menggunakan algoritma *Fibonacci codes* dapat mengurangi penggunaan media penyimpanan data lebih dari 30 % untuk 1 file mp3 dan dapat dengan mudah dipahami dan untuk sumber refrensi juga

gampang untuk ditemukan. Penerapan algoritma *Fibonacci codes* pada kompresi file audio mp3 menggunakan aplikasi *Microsoft Visual Basic 2008* sukses dilakukan. Cukup menyesuaikan sesuai dengan rumus yang berlaku di algoritma *Fibonacci codes* untuk kompresi file audio mp3. Perancangan aplikasi kompresi audio mp3 menggunakan algoritma *Fibonacci codes* sangat mudah dilakukan karena sudah disertai dengan panduan yang lengkap, baik dari buku, internet, jurnal dan lain-lain.

REFERENCES

- [1] Munir, Rinaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung : Informatika.
- [2] Salomon, David. 2004. *Data Compression – The Complete Reference 3rd Edition*. JUSA : Elsevier – Morgan Kaufman
- [3] Binanto, I. 2010. *Multimedia Digital, Dasar Teori + Pengembangan*, Yogyakarta: ANDI
- [4] Ruckert, Martin, 2005. *Understanding MP3 syntax, semantics, mathematics and algorithms*. Birkhauser.
- [5] Elizabeth Irene Yuwono, 2015. *Analisis Dan Implementasi Penggunaan Algoritma Huffman Pada Kompresi File Audio*, Bandung: Informatika
- [6] Suarga, 2012. *Algoritma dan Pemrograman*, Jakarta Selatan. Republika
- [7] Smith, David Eugene dan Karpinski, Louis Charles. *The Hindu-Arabic Numerals*. 2011 :h.128
- [8] Salomon, Motta, 2010. *Handbook of Data Compression 5th*. Springer
- [9] Architecture and Design : Unified Modeling Language {UML}, [http://www.cetus-links.org/oo_uml.html]
- [10] Hendrayudi. 2009. *VB 2008 untuk Berbagi Keperluan Pemrograman*. Jakarta Pusat : PT. Elex Media Komputindo
- [11] Teti Purnama, Surya, Rivalri, "Penerapan Algoritma Levenstein pada Aplikasi Kompresi File Mp3" vol. II, Nomor 1
- [12] Ihsan and D. P. Utomo, "Analisis Perbandingan Algoritma Even-Rodeh Code Dan Algoritma Subexponential Code Untuk Kompresi File Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
- [13] S. R. Saragih and D. P. Utomo, "Penerapan Algoritma Prefix Code Dalam Kompresi Data Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
- [14] Lamsah and D. P. Utomo, "Penerapan Algoritma Stout Codes Untuk Kompresi Record Pada Databade Di Aplikasi Kumpulan Novel," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.