

Penerapan Algoritma Golomb Coding Pada Aplikasi Kompresi Short Message Service (SMS)

Linni Hairani Butar Butar

Program Studi Teknik Informatika, Universitas Budi Darma Medan, Indonesia

Email: linni1111@gmail.com

Abstrak-Penggunaan handphone berbasis android ini sangat lebih mudah dalam menyampaikan suatu pesan kepada orang yang ditujukan. Seperti bisa mengirim pesan melalui WA, telegram, instagram dan aplikasi lainnya. Sehingga dengan cara ini manusia lebih cepat dalam mengirim suatu pesan. Walaupun sudah tidak banyak digunakan, tetapi masih ada orang yang melakukan pengiriman pesan melalui sms, dan tidak semua orang juga memiliki handphone bersistem operasi android. Adapun permasalahan dalam penggunaan sms ini yaitu semakin banyak teks yang dikirim maka akan dibutuhkan biaya yang sangat besar, jika melebihi 160 karakter maka isi pesan itu akan dikirim menjadi dua sms. Masalah tersebut bisa diatasi dengan cara menyingkat pesan yang dikirim, tetapi muncul masalah lain yaitu terjadi kesalah pahaman dalam membaca isi pesan karena banyak yang tidak mengerti dengan pesan yang disingkat-singkat tersebut.

Kata Kunci: Short Message Service; Android; Handphone; Golomb Coding.

Abstract-The use of this android-based mobile phone is very easy to convey a message to the intended person. Like being able to send messages via WA, Telegram, Instagram and other applications. So in this way humans are faster in sending a message. Although it is not widely used, there are still people who send messages via SMS, and not everyone also has a mobile phone with an Android operating system. The problem in using this sms is that the more text that is sent, it will cost a lot, if it exceeds 160 characters, the content of the message will be sent into two sms. This problem can be solved by abbreviating the message sent, but another problem arises, namely a misunderstanding in reading the contents of the message because many do not understand the abbreviated message.

Keywords: Short Message Service; Android; Handphone; Golomb Coding.

1. PENDAHULUAN

Teknologi informasi saat ini sangat berkembang pesat, sehingga manusia sudah tidak bisa terhindar dari pemakaian akan aplikasi pengiriman pesan, aplikasi SMS sudah menjadi sarana komunikasi antara sesama manusia. Walaupun saat ini SMS sudah tidak banyak digunakan orang dikarenakan dengan munculnya handphone yang berbasis android. Penggunaan handphone berbasis android ini juga lebih mudah dalam menyampaikan suatu pesan kepada orang yang ditujukan. Contohnya bisa mengirim pesan melalui WA, telegram, Instagram dan aplikasi lainnya. Dengan cara ini manusia lebih cepat dalam mengirim suatu pesan. Walaupun sudah tidak banyak digunakan tetapi masih ada orang yang melakukan pengiriman pesan melalui sms, dan tidak semua orang juga memiliki handphone bersistem operasi android tersebut.

Permasalahan dalam penggunaan SMS yaitu semakin banyak teks yang dikirim maka akan dibutuhkan biaya yang sangat besar, jika melebihi 160 karakter maka isi pesan itu akan dikirim menjadi dua SMS. Masalah tersebut bisa diatasi dengan cara menyingkat pesan yang dikirim, tetapi muncul masalah yang lainnya yaitu terjadi kesalah pahaman dalam membaca isi pesan karena banyak yang tidak mengerti dengan pesan yang disingkat-singkat tersebut. Penggunaan teks SMS yang panjang sering dilakukan untuk mengucapkan selamat lebaran atau selamat natal dan lainnya yang bahkan bisa sampai dihitung sebagai 3 (tiga) SMS, sehingga membutuhkan biaya yang lebih besar untuk mengirim pesan tersebut. Untuk mengatasi hal tersebut, bisa dilakukan proses kompresi sehingga byte yang panjang menjadi lebih singkat sehingga menghemat biaya pengiriman SMS tersebut. Adapun hasil penelitian yang dibuat oleh Eka Kristina Gulo pada tahun 2017 berhasil dalam mengkompresi data audio dengan menggunakan algoritma *Golomb Coding* yang hasil kompresinya dapat dikembalikan menjadi data asli tanpa ada perubahan, sehingga rasio kompresinya juga tidak terlalu besar. Untuk memastikan data audionya bisa dilihat pada data semula[1]. Pada tahun 2017 Laurentinus meneliti *Short Message Service* (SMS) merupakan suatu layanan yang memungkinkan pengguna untuk saling berkomunikasi dengan mengirimkan pesansingkat berupa teks dengan cepat dan biaya yang kecil. SMS ditujukan untuk mengirim pesan yang bukan bersifat sensitif/rahasia dikarenakan tidak adanya keamanan yang menjaga isi pesan dari pihak tidak berwenang, maka dari itu diperlukan aplikasi yang dapat menjamin keamanan dan keutuhan data dari transaksi pengiriman pesan SMS[2].

Saat ini banyak algoritma kompresi yang digunakan salah satunya algoritma *golomb coding* yang merupakan hasil kompresinya dapat dikembalikan menjadi data asli tanpa ada perubahan. Dengan menerapkan algoritma *golomb coding* untuk mengkompresi pesan pada SMS sehingga isi pesan menjadi lebih sedikit atau lebih kecil. Lalu si penerima SMS akan membaca isi pesan yang sudah dikompresi tersebut dengan melakukan proses dekompresi terlebih dahulu. Ada beberapa penelitian terdahulu yang menjadi panduan bagi penulis untuk menyelesaikan hasil dari penelitian ini :

Tabel 1. Penelitian Terdahulu

Tahun	Penulis	Judul	Kelebihan	Kekurangan
2018	E. K. Gulo	Perancangan Aplikasi Kompresi Audio dengan Menerapkan Algoritma Golomb	Penulisan bagus dan terstruktur membuat pembaca lebih santai ketika membaca jurnal, dan isinya	Tidak terdapat kejelasan nilai kriteria, dan tidak terdapat hasil matriks keputusan menggunakan

Tahun	Penulis	Judul	Kelebihan	Kekurangan
			juga bermanfaat untuk menentukan susu gym terbaik	matriks jadi kurang efektif untuk melihat hasilnya
2018	Laurentinus	Implementasi Kriptografi Dan Kompresi Sms Menggunakan Algoritma Rc6 Dan Algoritma Huffman Berbasis Android	Dengan menggunakan metode tersebut didapatkan bahwa kriteria yang paling dominan adalah kriteria kejujuran dan metode ini dapat digunakan untuk memecahkan masalah	Didalam pembahasannya tidak diberitahukan dengan jelas apa saja kriteria yang dibuat terlebih dahulu
2018	F. Gram <i>et al</i>	Analisis Perbandingan Kompresi dan Dekompresi Menggunakan Algoritma Shannon	Dapat dipahami dan dimengerti dengan mudah sehingga berguna	Didalam pembentukan matriksnya masih berbentuk tabel belum berbentuk matriks

2. METODOLOGI PENELITIAN

2.1 Penerapan

Penerapan adalah suatu perbuatan mempraktekkan suatu teori, metode, dan hal lain untuk mencapai tujuan tertentu dan untuk suatu kepentingan yang diinginkan oleh suatu kelompok atau golongan yang telah terencana dan tersusun sebelumnya[3].

2.2 Kompresi

Kompresi merupakan proses untuk menghilangkan berbagai redundansi dengan menemukan sumber redundansi disetiap data. Kompresi audio dilakukan dengan cara memadatkan isi file sehingga ukurannya menjadi lebih kecil dengan tetap menjaga kualitas representasi dari file audio tersebut.

2.3 Golomb Coding

Golomb Coding adalah salah satu teknik data *lossless*. Hal ini mampu mengompresi data berukuran lebih besar ke dalam data berukuran lebih kecil sambil tetap membiarkan data asli direkonstruksi kembali setelah dekompresi. Pengkode *golomb* menggunakan parameter m yang untuk membagi nilai masukan menjadi dua bagian: q , hasil pembagian oleh m , dan r itu sisa. Itu hasil bagi. Dikirim *unary coding*, diikuti oleh sisa dalam *encoding* biner terpotong.

Langkah di dalam proses kompresi dengan algoritma *Golomb*: *Golomb Code* dari nilai integer positif n adalah gabungan representasi bit *codeword* q dan r , dimana: Nilai q , yaitu $q = n / m$ Nilai r , yaitu $r = n - qm$

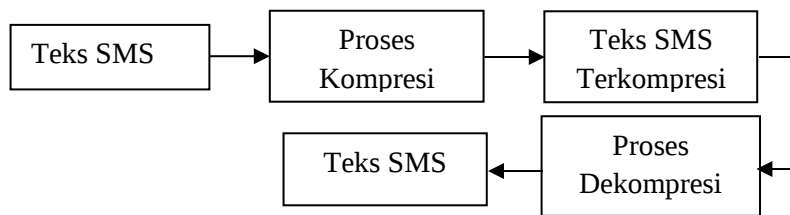
1. q bernilai 1, 2, ... dan dinyatakan dalam unary code q bernilai 1, 2, ... dan dinyatakan dalam *unary code*.
2. r bernilai 0, 1, 2, ..., $m-1$
3. Nilai parameter $m > 0$ ditentukan di awal
4. Jika m adalah kelipatan pangkat 2. Panjang bit *codeword* r , yaitu bit *Codeword* r = representasi biner dari nilai r
5. Jika m bukan kelipatan pangkat 2. *Codeword* r ditentukan oleh batasan representasi s , dengan perhitungan $s = 2 \log_2 m - m$ Jika nilai $r < s$, maka: Panjang bit *codeword* r , yaitu $\log_2 m$ bit *Codeword* r = representasi biner dari nilai r Jika nilai $r \geq s$, maka: Panjang bit *codeword* r , yaitu $\log_2 m$ bits *Codeword* r = representasi biner dari $r+2 \log_2 m$.

3. HASIL DAN PEMBAHASAN

Dalam hal ini pengirim mengetik pesan sms yang akan di kompresi dengan memasukkan nomor tujuan dan selanjutnya pengirim mengkompres pesan tersebut atau pengirim juga bisa mendekompresi pesan tersebut. Penerima pesan melihat isi pesan yang terkompresi dan mengetahui nomor tujuan sipengirim. Dengan teknik kompresi yang digunakan agar penyimpanan pesan teks tersebut semakin menghemat biaya dalam pengiriman pesan teks. Dengan menerapkan algoritma *Golomb Coding* merupakan kompresi data *lossless* yang memanipulasi *bit* data dalam sms untuk meminimalkan ukuran dengan cara membagi data menjadi dua keluaran, apabila persentase *byte* nol dari data masukan rendah, maka algoritma *Golomb Coding* akan menghasilkan pesan teks kompresi yang dapat mendekati nilai satu, bahkan bisa melebihi satu data terekspansi, sehingga dalam pengiriman pesan tidak banyak memakan kapasitas ruang penyimpanan yang lebih besar serta waktu dan biaya pengaksesan menjadi lebih banyak.

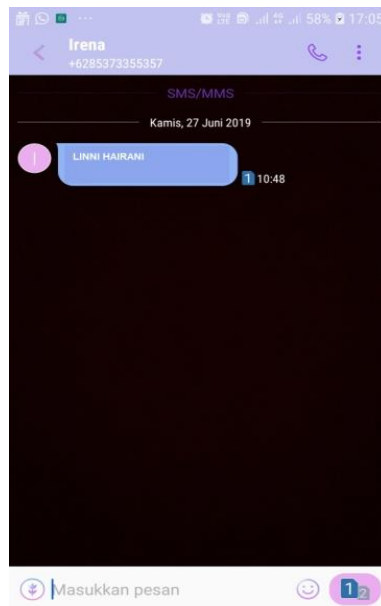
Penelitian ini merancang aplikasi kompresi pesan teks dengan menggunakan bahasa pemrograman eclipse. Proses yang dilakukan adalah Teks SMS (*encoding*) terlebih dahulu akan di kompresi, sehingga akan mengurangi ukuran teks SMS berdasarkan algoritma *Golomb Coding*, kemudian hasil *encoding* kompresi di dekompresi kembali berdasarkan *Golomb Coding* sehingga menghasilkan *decoding* hingga dihasilkan ukuran Teks SMS terdahulu.

Berikut ini merupakan prosedur kompresi dan dekompresi teks SMS.



Gambar 1. Prosedur Kompresi dan Dekompresi Teks SMS

Dalam masalah kapasitas ruang penyimpanan pesan sms bahkan bisa melebihi satu data terekspansi, sehingga dalam pengiriman pesan tidak banyak memakan kapasitas ruang penyimpanan yang lebih besar serta waktu dan biaya pengaksesan menjadi lebih banyak. Solusi dalam mengatasi masalah tersebut dapat digunakan teknik kompresi agar penyimpanan pesan teks tersebut semakin menghemat biaya dalam pengiriman pesan teks. Dalam teknik kompresi membutuhkan metode pengkompresian memiliki kelebihan dan kekurangan, yaitu data yang diukur dengan ukuran atau kecepatan kapasitas kecepatan kompresi. Dan pesan teks yang telah dikompresi dapat dikirim lebih mudah dan lebih cepat karna kapasitasnya yang lebih kecil. Berikut adalah langkah untuk mengkompresi dan mendekomposisi Teks SMS. Berikut adalah Teks SMS "LINNI HAIRANI" yang akan dikompresi dan dekomposisi.



Gambar 2. Teks SMS yang akan dikompresi

3.1 Proses Kompresi Algoritma Golomb Coding

Berikut ini adalah langkah-langkah kompresi dan dekomposisi algoritma *Golomb Coding* dengan contoh perhitungan kompresi Teks SMS.

- a. Buat daftar frekuensi kemunculan tiap-tiap simbol

Misalkan yang akan dikompresi "LINNI HAIRANI" seperti tabel dibawah ini:

Tabel 2. Ukuran *string* sebelum dikompresi

Karakter	Frekuensi	ASCII Desimal	ASCII Binary	Bit	Bit Frekuensi	x
L	1	76	01001100	8	8	
I	4	73	01001001	8	32	
N	3	78	01001110	8	24	
Space	1	32	00100000	8	8	
H	1	72	01001000	8	8	
A	2	65	01000001	8	16	
R	1	82	01010010	8	8	
Jumlah Bit x Frekuensi					104	

- b. Langkah selanjutnya menghitung dengan Codeword unary(q) dan binary(r) dengan menggunakan nilai m=5

Tabel 3. Codeword

n	q	r	Codeword unary(q)	binary(r)	(q) [®]
0	0	0	0	00	000
1	0	1	0	01	001
2	0	2	0	10	010
3	0	3	0	110	0110
4	0	4	0	111	0111
5	1	0	10	00	1000
6	1	1	10	01	1001
7	1	2	10	10	1010
8	1	3	10	110	10110
9	1	4	10	111	101 U11
10	2	0	110	00	11000
11	2	1	110	01	11001
12	2	2	110	10	11010
13	2	3	110	110	110110
14	2	4	110	111	110111
15	3	0	1110	00	111000

c. Kemudian melakukan perhitungan dengan tabel codeword unary(q) dan binary(r) dari kalimat "LINNI HAIRANI".

Berdasarkan perhitungan yang telah dilakukan, maka dapat disimpulkan dalam bentuk tabel dibawah ini :

Tabel 4. Hasil kompresi menggunakan codeword

m=5	N	Char	Frek	Codeword	Bit	BitxFreq
	0	I	4	000	3	12
	1	N	3	001	3	9
	2	A	2	010	3	6
	3	Sp	1	0110	4	4
	4	L	1	0111	4	4
	5	H	1	1000	4	4
	6	R	1	1001	4	4
					Total	43 bit

Tabel 5. Hasil karakter sesudah dilakukan kompresi

Karakter	Codeword	Bit
L	0111	4
I	000	3
N	001	3
N	001	3
I	000	3
Sp	0110	4
H	1000	4
A	010	3
I	000	3
R	1001	4
A	010	3
N	001	3
I	000	3
Bit		43

Maka hasil kompresi dari Teks SMS "LINNI HAIRANI" =0111000001001000011010000100001001010001000. Sehingga total bit yang diperoleh adalah sebanyak 43 bit. Kemudian sebelum di dapatkan hasil keseluruhan akhir kompresi dilakukan penambahan *string bit* itu sendiri yaitu *padding bit* dan *flag bit*. Jika sisa bagi panjang *string bit* terhadap 8 adalah 0 maka tambahkan 00000001. Nyatakan dengan bit akhir. Sedangkan jika sisa bagi panjang *string bit* terhadap 8 adalah $n(1,2,3,4,5,6,7)$ maka tambahkan 0 sebanyak $7 - n + "1"$ di akhir *string bit*. Nyatakan dengan L. Lalu tambahkan bilangan biner dari $9 - n$. nyatakan dengan bit akhir. karena jumlah string bit 43 tidak habis dibagi delapan dan sisanya 3 bit, nyatakan sisa bagi tersebut dengan nilai n . maka tambahkan 0 sebanyak $7 - n + "1"$ di akhir string bit. Nyatakan dengan L, lalu tambahkan bilangan biner dari $9 - n$. Nyatakan dengan bit akhir.

01110000	01001000	01101000	01000010
01010001	00000001	00000110	

Gambar 3. Perhitungan Penambahan Bit

String bit setelah penambahan *padding* dan *flagbit* adalah sebagai berikut : 01110000010010000110100001000010010100010000000100000110. Total bit keseluruhan setelah ada penambahan bit adalah $43+5+8=56$. Selanjutnya lakukan pemisahan bit menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit seperti gambar di bawah ini.

$$7-n + "1"$$

$$7-3 + "1" = 00001$$

Bit Akhir 9-n

$$\text{Bit Akhir} = 9-3 = 6 = 00000110$$

Gambar 4. Hasil pembagian menjadi 8 bit

Berdasarkan pada pembagian kelompok nilai biner, didapatkan 11 kelompok nilai biner baru yang sudah terkompresi beserta nilai biner penambahan bit. Setelah pembagian dilakukan, maka nilai yang sudah dibagi dirubah kedalam suatu karakter dengan terlebih dahulu mencari nilai desimal dari *string bit* tersebut menggunakan kode ASCII untuk mengetahui nilai yang sudah terkompresi. Adapun nilai simbol yang sudah terkompresi dapat dilihat pada tabel di bawah ini:

Tabel 5. Nilai Desimal dan Biner *Teks* Kompresi

No	Biner	Nilai Desimal Terkompresi	Simbol
1	01110000	112	p
2	01001000	72	H
3	01101000	104	h
4	01000010	66	B
5	01010001	81	Q
6	00000001	1	
7	00000110	6	-

Setelah nilai desimal diketahui, maka mengubah nilai desimal kedalam karakter yang dihasilkan itulah yang akan dikirim melalui SMS. Jadi SMS yang dikirim adalah pHhBQ -

Berdasarkan hasil kompresi dengan algoritma *golomb coding* dapat diketahui bahwa :

Jumlah Bit sebelum dikompresi = 104bit

Jumlah bit sesudah dikompresi = 56 bit

Kemudian menghitung rasio kompresi dalam bentuk persentase yaitu dapat dituliskan dalam rumus berikut :

$$\begin{aligned} \text{Rasio} &= (1 - (\text{Nsesudah} / \text{Nsebelum})) \times 100\% \\ &= (1 - (56/104)) \times 100\% \\ &= (1 - 0,53) \times 100\% \\ &= 0,47 \times 100\% \\ &= 47\% \end{aligned}$$

4. Proses Dekompresi Algoritma *Golomb Coding*

Selanjutnya proses dekompresi hal yang dilakukan adalah menganalisa keseluruhan bit hasil dari kompresi sebelumnya. Adapun bit keseluruhan hasil kompresi dapat dilihat pada tabel berikut :

1. Buat daftar tabel frekuensi untuk dekompresi

Tabel 6. Nilai Desimal dan Biner *teks* kompresi yang akan didekompresi

No	Simbol	Nilai Desimal Terkompresi	Biner
1	P	112	01110000
2	H	72	01001000
3	H	104	01101000
4	B	66	01000010
5	Q	81	01010001
6		1	00000001
7	-	6	00000110

Berdasarkan pada tabel di atas maka diambil seluruh nilai biner dan digabungkan menjadi "011100000100100001101000010000100101000100000001000001100000100000110".

Selanjutnya adalah dengan mengembalikan *binary* menjadi *string* bit semula dengan menghilangkan biner yang ditebalkan. Untuk mengembalikan *binary* menjadi *string bit* semula dapat dilakukan melalui langkah berikut ini. Lakukan pembacaan pada 8 bit terakhir, hasil pembacaan berupa bilangan desimal. Nyatakan hasil pembacaan dengan n . Hilangkan bit pada bagian akhir sebanyak $7+n$. Setelah dilakukan perhitungan pembacaan bit akhir. Nilai biner yang dihilangkan sebanyak 8 bit pada akhir. $n = 1$. Hilangkan $7 + n$ atau $7+5 = 12$. Penjelasan diatas menunjukan bahwa bit akhir harus dihilangkan. Hasil pengembalian *binary* menjadi *string bit* semula dapat dilihat sebagai berikut ini:
01110000010010000110100001000010010100010000000100000110.

Berdasarkan perhitungan dengan algoritma *golomb coding bit* pada diatas berjumlah 43 bit seperti diawal sehingga dilakukan pembacaan *string bit* awal. Adapun tabel hasil perhitungan diatas adalah sebagai berikut:

Tabel 7. Pencocokan tabel

No.	Nilai	Keterangan
1	0	Tidak Ada
2	01	Tidak Ada
3	011	Tidak Ada
4	0111	Ada Pada Tabel
5	0	Tidak Ada
6	00	Tidak Ada
7	000	Ada Pada Tabel
9	0	Tidak Ada
10	00	Tidak Ada
11	001	Ada Pada Tabel
13	0	Tidak Ada
14	00	Tidak Ada
15	001	Ada Pada Tabel
16	0	Tidak Ada
17	00	Tidak Ada
18	000	Ada Pada Tabel
19	0	Tidak Ada
20	01	Tidak Ada
21	011	Tidak Ada
22	0110	Ada Pada Tabel
23	1	Tidak Ada
24	10	Tidak Ada
25	100	Tidak Ada
26	1000	Ada Pada Tabel
27	0	Tidak Ada
28	01	Tidak Ada
29	010	Ada Pada Tabel
30	0	Tidak Ada
31	00	Tidak Ada
32	000	Ada Pada Tabel
33	1	Tidak Ada
34	10	Tidak Ada
35	100	Tidak Ada
36	1001	Ada Pada Tabel
37	0	Tidak Ada
38	01	Tidak Ada
39	010	Ada Pada Tabel
40	0	Tidak Ada
41	00	Tidak Ada
42	001	Ada Pada Tabel
43	0	Tidak Ada
44	00	Tidak Ada
45	000	Ada Pada Tabel

Maka dari penjabaran diatas dapat dibentuk tabel *Golomb Coding* dan nilai teks SMS awal.

Tabel 8. Hasil dekompresi algoritma *Golomb Coding*

Codeword	Karakter
0111	L
000	I
001	N
001	N
000	I
0110	SP
1000	H
010	A
000	I
1001	R
010	A
001	N
000	I

4. KESIMPULAN

Berdasarkan uraian dari pembahasan di atas, dapat diberikan kesimpulan bahwa proses *Golomb Coding* untuk mengompresi dilakukan dengan mengganti setiap *bit* dan memperkecil setiap ukuran karakter. Proses kompresi pada sms dapat dilakukan dengan *Golomb Coding* sehingga pesan yang dikirim tidak banyak menggunakan karakter sehingga menghemat biaya.

REFERENCES

- [1] E. K. Gulo, "Perancangan Aplikasi Kompresi Audio dengan Menerapkan Algoritma Golomb," *Pelita Inform. Budi Dharma*, vol. 16, pp. 436–439, 2017.
- [2] Laurentinus, "Implementasi Kriptografi Dan Kompresi Sms Menggunakan Algoritma Rc6 Dan Algoritma Huffman Berbasis Android," *J. Ilm. Inform. Glob.*, vol. 8, no. 1, pp. 36–42, 2017.
- [3] "Kamus Besar Bahasa Indonesia (KBBI)." .
- [4] F. Gram *et al.*, "Analisis Perbandingan Kompresi dan Dekompresi Menggunakan Algoritma Shannon-," vol. 3, no. 3, pp. 5197–5204, 2016.
- [5] E. M. Harahap, D. Rachmawati, S. Si, M. Kom, and M. Kom, "Impelementasi Kompresi Teks Menggunakan Metode Huffman untuk Menghemat Karakter pada Short Message Service," p. 7, 2012.
- [6] B. Ibrahim, Rohmat Nur (STMIK Mardira Indonesia, "Perbandingan Kompresi File Menggunakan Algoritma Run Length Dengan Two Level Hoshing," *J. Comput. Bisnis*, vol. 1, no. 2, pp. 90–104, 2007.
- [7] P. . Dr.Suarga, M.Sc., M.Math., *Algoritma dan pemrograman*. Yogyakarta: ANDI, 2012.
- [8] Y. Sugiyani *et al.*, "Perancangan Aplikasi Edukatif Berbasis Multimedia," vol. 1, no. September, pp. 55–59, 2014.
- [9] M. S. Rosa A.S, *Rekayasa Perangkat Lunak*. Bandung: INFORMATIKA Bandung, 2016.
- [10] S. K. Alfa Satyaputra, M.Sc,Eva Maulina Aritonang, *JAVA for Beginners with eclipse 4.2 JUNO*. Jakarta, 2012.