

Penerapan Algoritma *Elias Delta Codes* Dalam Kompresi File Teks

Nadia Fariza Rizky*, Surya Darma Nasution, Fadlina

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: 'nadyarizky97@gmail.com

Abstrak

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. Melihat kompresi akan pemanipulasian data, diperlukan kompresi data yang mampu memperkecil ukuran data, sehingga memperoleh keuntungan mengurangi pemakaian ruang media penyimpanan eksternal, mempercepat proses transfer data antar media penyimpanan, pemanipulasian data yang digunakan dalam penelitian ini adalah algoritma *elias delta codes*. Kompresi merupakan suatu cara yang dilakukan untuk memperkecil bit atau data dari hasil semula ke hasil yang baru. Dimana kompresi ini akan diterapkan kedalam algoritma, yaitu algoritma *elias delta codes*. Sesudah mendapatkan hasil kompresinya maka akan dirancang ke dalam *microsoft visual studio 2008*.

Kata Kunci: Kompresi; File; Elias Delta Codes

Abstract

File size or large files are not only a problem in terms of storage, but also another problem when communication between computers. Data usage with a larger size will take longer transfer times compared to data that has a smaller size. Therefore to overcome this problem you can use data compression. Seeing the compression of data compression, data compression is needed that can reduce the size of the data, so that it gains the advantage of reducing the use of external storage media space, accelerating the process of transferring data between storage media, data utilization used in this study is the *elias delta codes* algorithm. Compression is a method used to reduce the bit or data from the original result to a new result. Where this compression will be applied into the algorithm, the *elias delta codes* algorithm. After getting the results of the compression it will be designed into *Microsoft Visual Studio 2008*.

Keywords: Compression; File; Elias Delta Codes

1. PENDAHULUAN

Kompresi *file* merupakan proses pengubahan sekumpulan data menjadi suatu bentuk kode untuk mengurangi ukuran *file* sebelum menyimpan atau memindahkan data tersebut kedalam media penyimpanan (*storage device*). Apabila data yang akan disimpan pada media penyimpanan semakin bertambah atau berukuran besar, maka media penyimpanan tidak dapat menyimpan data tersebut melebihi dari kapasitas. Data dengan ukuran besar akan memakan waktu yang transfer yang lebih lama dibandingkan dengan data yang ukurannya lebih kecil.

Kapasitas penyimpanan atau memori merupakan komponen yang sangat mahal dan sulit untuk diperoleh. adapun inovasi-inovasi dalam bidang komputer ini banyak terhambat dikarenakan alasan kapasitas memori yang masih langka dan mahal. inovasi perangkat lunak dan aplikasi melahirkan dilema baru, yaitu semakin besarnya ukuran file atau berkas yang dihasilkan.

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.

Kompresi data terdiri dari dua proses utama yaitu kompresi dan dekompresi atau pemulihan data kembali seperti semula. Kompresi data dalam konteks ilmu komputer merupakan ilmu dan seni yang menampilkan informasi dalam bentuk yg pendek. Kompresi data bertujuan untuk mengurangi jumlah bit yang digunakan untuk menyimpan ataupun mengirim suatu informasi. Dengan adanya kompresi maka dapat diharapkan bisa menghemat biaya serta waktu yang dikeluarkan serta mempercepat proses transfer data.

File teks yang di kompresi memiliki ukuran yang lebih besar. Dengan menerapkan algoritma *elias delta codes* file teks yang berukuran lebih besar akan dikompresi menjadi ukuran yang lebih kecil. Algoritma tersebut akan memberikan hasil rancangan yang baik dan memampatkan ukuran yang efektif.

Dari penelitian sebelumnya Mhd. Ranjani pane telah berhasil membuat aplikasi kompresi teks *Shannon Fano* dan *Unary Coding* sehingga mendapatkan hasil sesuai dengan proses kompresi yang dikerjakan. Dan pada saat proses kompresi, aplikasi ini membaca karakter yang ada dalam data teks, kompresi *file* teks yang efektif untuk digunakan dengan melakukan perbandingan kompresi *file* teks dengan menggunakan metode *Shannon Fano* dengan *Unary Coding*.

Sekarang ini banyak sekali terdapat algoritma kompresi salah satunya adalah algoritma *elias delta code* yang digunakan untuk mengkompresi data dan memberikan manfaat dalam penyimpanan serta membutuhkan ruang penyimpanan yang lebih sedikit dibandingkan dengan data yang tidak dikompresi.

2. METODE PENELITIAN



2.1 Kompresi

Kompresi merupakan proses pengubahan sekumpulan data menjadi suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data[1]. 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[2].

2.2 File Text

File-text merupakan *file* yang berisi informasi dalam bentuk teks. Data yang berasal dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca. Masukan dan keluaran data teks direpresentasikan sebagai set karakter atau sistem kode yang dikenal oleh sistem komputer. Ada tiga macam set karakter yang umum digunakan untuk masukan dan keluaran pada komputer, yaitu ASCII, EBCDIC, dan Unicode. ASCII (*American Code for Information Interchange*) merupakan suatu standar internasional dalam kode huruf dan simbol seperti Hexdan Unicode, tetapi ASCII lebih bersifat universal. ASCII digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII memiliki komposisi bilangan biner sebanyak 8 *bit*, dimulai dari 00000000 hingga 11111111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam sistem bilangan desimal[6].

2.3 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[8].

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 $00|101|0001$.

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:
 - a. Hitung jumlah nol terdepan dari kode tersebut lalu gantikan perhitungan tersebut dengan C.
 - b. Periksa bit bagian kiri $2C + 1$ (C nol, diikuti dengan 1, lalu diikuti dengan bit C selebihnya). Ini merupakan *decode elias gamma code* $M + 1$.
2. Baca bit M berikutnya. Sebut ini sebagai L.
3. Bilangan bulat yang di *decode* adalah $2M + L$ [8].

3. ANALISA DAN PEMBAHASAN

Ukuran data yang semakin besar mengharuskan pemakai komputer untuk berbagai macam cara agar dapat menyimpan sejumlah data, informasi data dalam media penyimpanan yang terbatas. Penyimpanan data merupakan suatu hal yang penting, terlebih lagi bagin suatu instansi atau perusahaan yang memiliki banyak data. Sebelum melakukan tahap perancangan sistem, perlu dilakukan analisa sistem yang akan di bangun. Analisa sistem merupakan suatu teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari supaya bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka. Penulis akan menganalisa aplikasi kompresi file teks yang berekstensi *.txt dengan menggunakan algoritma Elias Delta Code. Permasalahan yang akan di bahas ialah cara merancang sebuah aplikasi kompresi file teks sehingga aplikasi yang telah dirancang dapat mengkompresi file teks yang berukuran besar, sehingga ukurannya menjadi lebih kecil, serta baik dan aman dalam proses penyimpanan maupun transfer data tanpa mengurangi isi dari data tersebut.

3.1 Penerapan Algoritma Elias Delta Codes Pada File Teks

Pada masalah ini ialah sebuah file teks yang berekstensi .txt dengan menggunakan algoritma Elias Delta Codes. Dalam proses kompresi file teks yang dilakukan terbagi menjadi 2 proses, yaitu:

1. Kompresi

Adapun langkah-langkah untuk kompresi file teks dokumen.txt dengan algoritma elias delta yaitu:



- a. Langkah pertama dalam melakukan proses kompresi file adalah dengan memasukkan file teks dokumen.txt isi file teks dokumen.txt adalah **NADIA DAN DIANA ADA DIMANA**.
- b. Setelah file teks diterima maka timer akan melakukan pencatatan waktu yang diperlukan untuk melakukan proses kompresi mulai awal hingga selesai
- c. Selanjutnya adalah dengan melakukan pembacaan isi file. Pada tahap ini akan dilakukan pembentukan karakter set yaitu karakter-karakter yang terdapat pada isi file.
- d. Setelah karakter set diperoleh maka frekuensi kemunculan untuk tiap-tiap karakter akan dihitung. Berikut adalah hasil scanning terhadap file teks.
 Karakter Set = {N,A,D,I,SPASI,M}
 Frekuensi karakter =
 N = 4
 A = 9
 D = 5
 I = 3
 SPASI = 4
 M = 1
- e. Setelah karakter set dan frekuensi kemunculan tiap karakter diketahui maka selanjutnya karakter set akan diurutkan berdasarkan frekuensi kemunculannya mulai dari karakter dengan frekuensi kemunculan terbesar ke karakter dengan kemunculan frekuensi terkecil. Jika terdapat lebih dari satu karakter dengan frekuensi kemunculan yang sama maka diurutkan berdasarkan abjad. Untuk hasil pengurutan karakter set dan frekuensi karakter dapat dilihat pada tabel 1. dibawah ini :

Tabel 1. Hasil Pengurutan Karakter Set Kompresi

No	Karakter	Frekuensi
1	A	9
2	D	5
3	N	4
4	SPASI	4
5	I	3
6	M	1

- f. Selanjutnya adalah dengan membentuk tabel kode *Elias Delta Codes*. Untuk kode-kode yang terbentuk pada tabel *Elias Delta Codes* dapat dilihat pada tabel berikut:
- g. Berikut ini adalah tabel frekuensi data dari kode binary atau data yang belum di kompresi tabel tersebut dapat dilihat pada dibawah ini:

Tabel 2. frekuensi data dari kode binary

No	Karakter	Frekuensi	ASCII binary coding	Jumlah bit	Total bit
1	A	9	01000001	8	72
2	D	5	01000100	8	40
3	N	4	01001110	8	32
4	SPASI	4	00100000	8	32
5	I	3	01001001	8	24
6	M	1	01001101	8	8
Total		26	Total		208

- h. berikut ini adalah langkah-langkah dalam kompresi Elias Delta Codes:
 1. Tulis n dalam biner. Bit yang paling kiri (paling signifikan) akan menjadi 1.
 2. Hitung bit, hapus bit paling kiri dari n , dan tambahkan hitungan, di dalam biner, apa yang tersisa dari bit n yang paling kiri setelah telah dihapus.
 3. Kurangi 1 dari hitungan di langkah 2, dan tambahkan jumlah nol sebanyak hasil hitungan tersebut ke awal kode.

Tabel 3. Tabel Kode Elias Delta

$1 = 2^0 + 0 \rightarrow L = 0 \rightarrow 1$	$10 = 2^3 + 2 \rightarrow L = 3 \rightarrow 00100010$
$2 = 2^1 + 0 \rightarrow L = 1 \rightarrow 0100$	$11 = 2^3 + 3 \rightarrow L = 3 \rightarrow 00100011$
$2 = 2^1 + 0 \rightarrow L = 1 \rightarrow 0100$	$12 = 2^3 + 4 \rightarrow L = 3 \rightarrow 00100010$
$4 = 2^2 + 0 \rightarrow L = 2 \rightarrow 01100$	$13 = 2^3 + 5 \rightarrow L = 3 \rightarrow 00100101$
$5 = 2^2 + 1 \rightarrow L = 2 \rightarrow 01101$	$14 = 2^3 + 6 \rightarrow L = 3 \rightarrow 00100010$
$6 = 2^2 + 2 \rightarrow L = 2 \rightarrow 01110$	$15 = 2^3 + 7 \rightarrow L = 3 \rightarrow 00100111$
$7 = 2^2 + 3 \rightarrow L = 2 \rightarrow 01111$	$16 = 2^4 + 0 \rightarrow L = 4 \rightarrow 001010000$
$8 = 2^3 + 0 \rightarrow L = 3 \rightarrow 00100000$	$17 = 2^4 + 1 \rightarrow L = 4 \rightarrow 001010001$
$9 = 2^3 + 1 \rightarrow L = 3 \rightarrow 00100001$	$18 = 2^4 + 2 \rightarrow L = 4 \rightarrow 001010010$



Tabel 4. penggantian karakter dengan kode Elias Delta

No	Karakter	Frekuensi Karakter	Kode elias delta	Bit	Frekuensi X Bit
1	A	9	1	1	9
2	D	5	0100	4	20
3	N	4	0101	4	16
4	SPASI	4	01100	5	20
5	I	3	01101	5	15
6	M	1	01110	5	5
Total karakter		26	Total bit		85

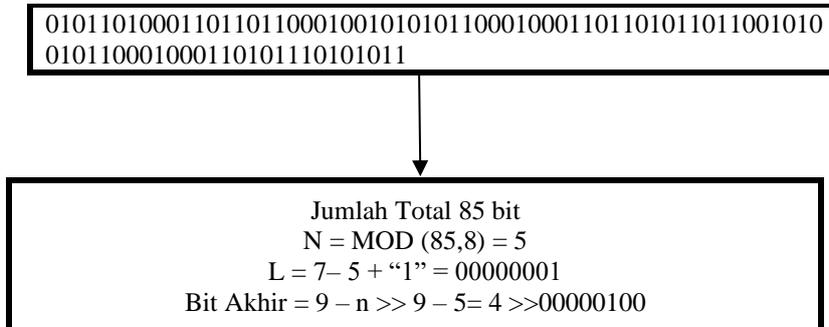
- i. Setelah diketahui kode untuk masing-masing karakter maka akan diperoleh sebuah string bit untuk nadia.txt dengan isi dokumen adalah NADIA DAN DIANA ADA DIMANA seperti yang terlihat pada gambar dibawah ini.

```
0101,1,0100,01101,1,01100,0100,1,0101,01100,0100,01101,1,0101,1,01100,1,0100,1,01100,0100,01101,01110,1,0101,1
```

Gambar 1. String Bit Kode Elias Delta Pada dokumen.txt

- j. Selanjutnya adalah melakukan *generate string* bit kode Elias Delta Code pada file dokumen.txt menjadi kode ASCII namun sebelum melakukan generate dilakukan pemeriksaan terhadap panjang string bit akhir.
1. Jika sisa bagi panjang *string bit* terhadap 8 adalah 0 maka tambahkan 00000001. Nyatakan dengan Bit Akhir.
 2. 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.

Langkah-langkah pemeriksaan string bit dapat dilihat pada gambar 2. dan gambar 3. dibawah ini:

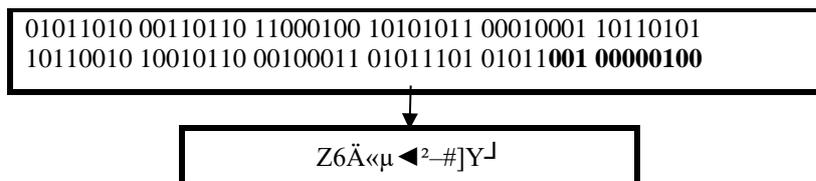


Gambar 2. Pemeriksaan Panjang String Bit

```
0101101000110110110001001010101100010001101101011011001010101100010001101011101010110010000100
```

Gambar 3. Hasil Penambahan Pada String Bit

Berikut ini adalah hasil generate kode elias delta kode menjadi kode ascii dapat dilihat dibawah ini :



Gambar 4. Hasil String Bit Ke ASCII

Rasio kompresi proses kompresi yang ditandai dengan dimulainya perhitungan waktu hingga diberhentikannya waktu adalah sebagai berikut:

Input Stream = $26 \times 8 = 208$

Output Stream = 85

Rasio Kompresi = $\frac{85}{208} \times 100\% = 40.86\%$

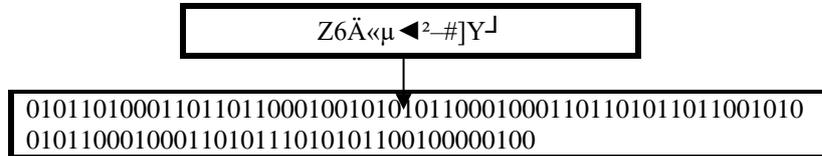


Sisa file yg dikompresi akan tersimpan di file header jadi pada saat di dekompresi file tersebut bisa dikembalikan seperti semula.

2. Dekompresi

Adapun langkah-langkah untuk dekompresi dengan algoritma elias delta yaitu sebagai berikut :

- a. Langkah pertama untuk melakukan dekompresi file adalah dengan memasukkan file hasil kompresi. Saat file dimasukkan timer akan melakukan pencatatan waktu yang dilanjutkan dengan melakukan generate terhadap isi fie ke binary. Hasil generate dapat dilihat dibawah ini:

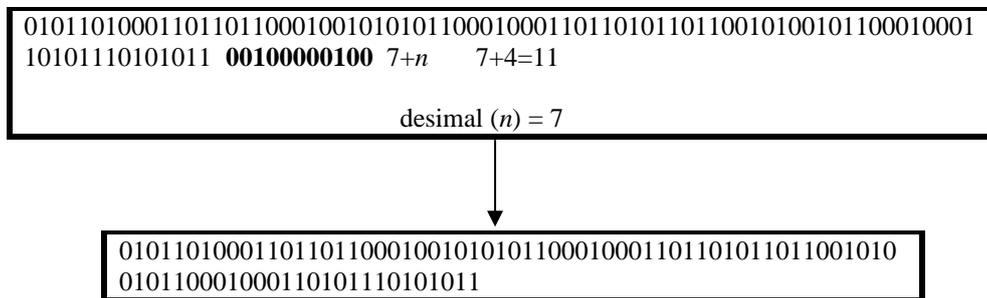


Gambar 5. Hasil ASCII Ke Binary

- b. Selanjutnya adalah dengan mengembalikan binary menjadi string bit semula. Untuk mengembalikan binary menjadi string bit semula dapat dilakukan melalui langkah-langkah sebagai berikut :

1. Lakukan pembacaan pada 8 bit terakhir, hasil pembacaan berupa bilangan decimal. Nyatakan hasil pembacaan dengan n .
2. Hilangkan bit pada bagian akhir sebanyak $7+n$.

Hasil pengembalian binary menjadi string bit semula dapat dilihat pada gambar dibawah ini:



Gambar 6. Pengembalian Binary Ke String Bit Semula

String bit pada gambar diatas berjumlah 85 bit seperti string bit diawal. Selanjutnya dilakukan pembacaan string bit dari indeks ke-1 sampai indeks ke-n. Indeks ke-1 =0 tidak ada pada tabel maka ditambahkan dengan indeks ke-2 menjadi "01", kode tersebut tidak terdapat pada tabel maka dan dilanjutkan sampai indeks ke-4 menjadi "0101", terdapat pada tabel dan menunjukkan karakter berupa "N". lakukan hal yang sama pada sisa string bit, pada indeks ke-5="1" terdapat pada tabel dan menjadi karakter berupa "A". dan begitu seterusnya sampai indeks ke-n sehingga dihasilkan string semula seperti gambar dibawah ini:

NADIA DAN DIANA ADA DIMANA

Gambar 7. Hasil Dekompresi

4. KESIMPULAN

Beberapa kesimpulan yang dapat diambil dari penelitian yang telah dilakukan diantaranya cara kerja atau prosedur Algoritma Elias Delta Codes untuk kompresi file teks telah bekerja sesuai dengan prosedur yang telah diterapkan. Sehingga file teks yang memiliki ukuran yang lebih besar dapat di kompresi menjadi ukuran yang lebih kecil. Proses penerapan algoritma Elias Delta Codes terhadap kompresi file teks tersusun berdasarkan langkah-langkah yang sudah diterapkan. Sehingga proses pengompresian file teks telah terbukti bahwa proses penerapan tersebut file yang ukurannya besar dapat di perkecil dengan menerapkan Algoritma Elias Delta Codes. Dalam perancangan algoritma elias delta codes dapat menggunakan program Microsoft visual studio 2008, dan diharapkan dapat membantu dalam pengkompresian file teks.

REFERENCES

[1] Shannon, F. Menggunakan, J. S. Informasi, F. I. Komputer, and U. M. Buana, "Perancangan Aplikasi Kompresi SMS dengan," 2010.
 [2] P. Informatika, B. Darma, M. Program, and S. Teknik, "PERANCANGAN APLIKASI KOMPRESI FILE AUDIO DENGAN," no. 12110669, pp. 104–109, 2013.



- [3] Y. Laia, "Optimasi Rasio Kompresi Dan Kompleksitas Waktu Kompresi File Teks Menggunakan Algoritma Lempel-ZIV-Welch Dengan Fibonacci Search," vol. 1, pp. 61–66, 2016.
- [4] I. Pendahuluan et al., "Perancangan aplikasi kompresi menggunakan metode shannon fano dan unary coding pada file teks," vol. 12, no. September, pp. 306–311, 2017.
- [5] dkk T.Sutoyo, S.SI., M.Kom., TEORI PENGOLAHAN CITRA DIGITAL. yogyakarta: ANDI OFFSET, 2009.
- [6] Sutardi, "Implementasi dan analisis kinerja algoritma shannon- fano untuk kompresi file text," vol. 6, no. 1, pp. 53–60, 2014.
- [7] fathul wahid, Dasar-Dasar Algoritma & Pemrograman. yogyakarta: ANDI OFFSET, 2004.
- [8] D.Salomon, Handbook of Data Compression. London: Springer-Verlag London Limited 2010, 2010.
- [9] R. A. S. M. Shalahuddin, Rekayasa Perangkat Lunak. Bandung: INFORMATIKA Bandung, 2016.
- [10] Hendrayudi, Dasar-Dasar Pemrograman Microsoft Visual Basic 2008. Bandung: PT. SARANA TUTORIAL NURANI SEJAHTERA, 2011.
- [11] R. Priyanto, Langsung Bisa VISUAL BASIC.NET 2008. yogyakarta: ANDI OFFSET, 2009.