ANALISA PENERAPAN ALGORTIMA *GOLDBACH CODES* DAN METODE *SHANNON-FANO* PADA KOMPRESI FILE TEKS

Muhammad Apriyanto[1], Hutrianto[2]
Fakultas Teknik Ilmu Komputer, Universitas Bina Darma
Email: muhammadapriyanto1604@gmail.com

ABSTRAK

Perkembangan teknologi berperan penting dalam pertukaran informasi yang cepat. Peningkatan penggunaan data telah menyebabkan masalah pada penyimpanan data dan secara tidak langsung meningkatkan permintaan akan penyimpanan data. Semakin besar ukuran file, semakin banyak ruang penyimpanan yang dibutuhkan. Saat mengirimkan informasi dalam bentuk teks, masih terdapat permasalahan, salah satunya adalah teks yang berukuran besar. Alternatif lain untuk mengatasi masalah ini adalah dengan mengompres file agar lebih kecil untuk menghemat ruang penyimpanan. Kompresi data biasanya diterapkan pada komputer, karena setiap simbol yang muncul pada komputer memiliki nilai bit yang berbeda. Algoritma yang digunakan dalam penelitian ini adalah algoritma *Goldbach Code* dan dengan menggunakan metode *Shannon-fano*. Teknik kompresi ini mengganti karakter berulang dengan pola tertentu sehingga ukuran file dapat diperkecil. Algoritma tersebut terlebih dahulu menyediakan rangkaian *string* sebagai input, kemudian bagaimana cara menghasilkan output dari algoritma tersebut berupa *string biner* atau kode yang mengubah setiap string input, sehingga string tersebut memiliki bit yang lebih sedikit daripada string yang tidak dikompresi. Algoritma tersebut akan dihitung performanya berdasarkan *Compression Ratio, Ratio of Compression, Redundancy*, dan Waktu Kompresi.

Kata kunci: *Kompresi , Algoritma Goldbach Code , shannon-fano, File Teks*

ABSTRACT

Technological developments play an important role in the rapid exchange of information. The increase in data usage has caused problems with data storage and indirectly increased the demand for data storage. The larger the file size, the more storage space it will take up. When sending information in text form, there are still problems, one of which is large text. Another alternative to solve this problem is to compress the files to make them smaller to save storage space. Data compression is usually applied to computers, because each symbol that appears on the computer has a different bit value. The algorithm used in this study is the Goldbach Code algorithm and using the Shannon-fano method. This compression technique replaces repeating characters with a specific pattern so that the file size can be reduced. The algorithm first provides a series of strings as input, then how to produce the output of the algorithm in the form of a binary string or code that changes each input string, so that the string has fewer bits than the uncompressed string. The performance of the algorithm will be calculated based on the Compression Ratio, Ratio of Compression, Redundancy, and Compression Time.

Keywords: *Compression, Goldbach Code Algorithm, Shannon-Fano, Text File*

1. PENDAHULUAN

Peningkatan penggunaan data telah menyebabkan masalah pada penyimpanan data dan secara tidak langsung meningkatkan permintaan akan penyimpanan data. Semakin besar ukuran file, semakin banyak ruang penyimpanan yang dibutuhkan. "Saat ini, file memiliki berbagai jenis format. Database data numerik tradisional terstruktur. Informasi yang dihasilkan dari aplikasi

bisnis. Dokumen teks tidak terstruktur, email, video, audio, data ticker saham dan transaksi keuangan" [12]. File teks saat ini banyak digunakan untuk menyimpan informasi penting dan sederhana. "Data dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat pada database merupakan contoh input data teks yang terdiri dari karakter, angka, dan tanda baca" [7].

Dalam ilmu komputer, kompresi data merupakan salah satu cara untuk mengompresi data, sehingga hanya membutuhkan sedikit ruang penyimpanan untuk menyimpan data dengan lebih efisien atau mempersingkat waktu untuk pertukaran data. Permasalahan mendasar dalam proses kompresi data adalah bagaimana cara mengkompres data (khususnya data teks) yaitumendapatkan file teks yang ukurannya lebih kecil dari ukuran aslinya, kemudian penerima merekonstruksinya kembali ke data aslinya (dekompresi). Algoritma kode Goldbach adalah algoritma yang mengasumsikan penggunaan teori konjektur Goldbach, di mana semua bilangan genap positif yang lebih besar dari 2 merupakan penjumlahan dari dua bilangan prima.Goldbach Codes memiliki tiga kode, Goldbach Codes yang pertama dinamakan "G0". G0 mengkodekan bilangan bulat positif n dengan mengubahnya menjadi bilangan bulat positif genap dengan 2(n+3) dan kemudian menuliskan pasangan penjumlahan bilangan prima dalam keadaan terbalik.

Shannon dan Fano (1948) mengembangkan algoritma yang menghasilkan kode dengan angka lebih sedikit untuk setiap karakter yang berisi data dengan membangun pohon kode biner daripada menggunakan kode dengan panjang tertentu (seperti kode ASCII). Shannon menjelaskan kode yang lebih pendek ini, yang disebut kode panjang variabel. Jika frekuensi kemunculannya lebih tinggi, kodenya lebih pendek, dan sebaliknya. Yang harus diperhatikan dalam kompresi file adalah *Completeness* (integritas data setelah file dikompresi), *Compression ratio* (ukuran data setelah kompresi), dan *Optimally* (apakah ukuran file sama sebelum kompresi). Untuk mengetahui berapa besar nilai dari *Completeness, Compression ratio,* dan *Optimally* [9].

2. METODOLOGI PENELITIAN

2.1 Metode Penelitian

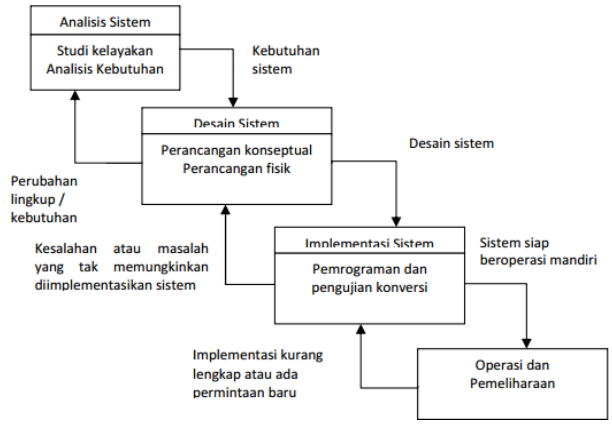
Metode yang digunakan dalam penelitian ini adalah metode penelitian deskriptif. Metode penelitian deskriptif merupakan metode penelitian yang banyak digunakan dalam penelitian, yang bertujuan untuk memberikan gambaran, penjelasan dan verifikasi terhadap peristiwa atau objek yang diteliti. Seperti yang dikemukakan oleh Sugiyono [11] "penelitian desktiptif adalah sebuah penelitian yang bertujuan untuk memberikan atau menjabarkan suatu keadaan atau fenomena yang terjadi saat ini dengan menggunakan prosedur ilmiah untuk menjawab masalah secara aktual".

2.2 Data Penelitian

Data penelitian yang menjadi objek penelitian dalam studi kasus Analisa Penerapan Algortima *GoldBach Codes* dan Metode *Shannon-Fano* pada Kompresi File Teks yaitu data-data jenis file teks dengan format ekstensi .txt atau .doc yang akan di gunakan dalam proses kompresi dan dekompresi.

2.3 Metode Pengembangan Sistem

Penelitian merupakan rangkaian kegiatan ilmiah yang bersumber dari pertanyaan untuk mencari jawaban. Dalam penelitian langkah kerja sangat penting, agar hasil yang dilaporkan dapat berjalan sesuai dengan tujuan penelitian. Seperti kebanyakan proses, pengembangan sistem informasi juga memiliki siklus/daur hidup. Daur hidup tersebut dinamakan SDLC (*System Development Life Cycle*) atau daur hidup pengembangan system. Dalam penelitian ini digunakan metode pengembangan *System Development Life Cycle* model *waterfall* [1].



Gambar 1. *SDLC model Waterfall* [1]

2.4 Kompresi File

Menurut Pane [7] "Kompresi adalah salah satu cara yang dapat merubah suatu citra atau gambar atau file dimana proses yang sering terjadi dalam kompresi sering terjadi tingkat penduplikatan dalam sebuah proses citra atau file". Sedangkan menurut Manullang [5] "Kompresi adalah pengubahan data yang berupa kumpulan karakter menjadi bentuk kode dengan tujuan untuk menghemat kebutuhan tempat penyimpanan dan waktu transmisi data".

Berdasarkan pendapat dari beberapa sumber di atas penulis berpendapat bahwa kompresi data atau file adalah proses konversi suatu fakta data menjadi data terbarui dengan ukuran data lebih kecil dari sebelumnya tanpa menghilangkan maksud dan tujuan dari data semula. Kompresi adalah untuk mengurangi ukuran data ke ukuran yang lebih kecil dari ukuran aslinya. Tujuan dari kompresi data adalah untuk memperkecil ukuran data (hasil kompresi) dari data asli untuk memperkecil ruang penyimpanan, memudahkan proses transmisi dan mengurangi kebutuhan *bandwidth*.

Ada beberapa parameter yang umum digunakan untuk menyatakan kinerja metode Kompresi, diantaranya sebagai berikut :

1) Compression Ratio (CR)

Compression Ratio (CR) adalah persentase besar dari data terkompresi yang diperoleh dari perbandingan antara ukuran data terkompresi dan data sebelum kompresi. Secara matematis bisa ditulis seperti ini:

$$\mathbf{CR} = \frac{\boldsymbol{Ukuran\ data\ setelah\ dikompresi}}{\boldsymbol{Ukuran\ data\ sebelum\ dikomprrsi}}\ \boldsymbol{x}\ \mathbf{100\%}$$

2) Ratio of Compression (RC)

Ratio of Compression (RC) adalah hasil perbandingan antara ukuran data yang tidak dikompresi dan data yang dikompresi.

$$\mathbf{RC} = \frac{\boldsymbol{Ukuran\ data\ setelah\ dikompresi}}{\boldsymbol{Ukuran\ data\ sebelum\ dikomprrsi}}$$

3) Redundancy (Rd)

Redundancy (*Rd*) bagian ekstra yang berisi data sebelum kompresi. Oleh karena itu, setelah melakukan kompresi data, dapat dihitung redundansi data, yaitu persentase selisih antara ukuran data sebelum kompresi dan data setelah kompresi.

Rd = 100% - CR

4) Space Savings (SS)

Space Savings (SS) adalah persentase selisih antara data yang belum dikompresi dengan besar data yang dikompresi.

SS = 1 – CR

5) Waktu Kompresi

Waktu kompresi adalah waktu yang diperlukan sistem untuk memasukkan file teks yang akan dikompresi sebelum proses kompresi selesai. Semakin sedikit waktu yang diperlukan sistem untuk melakukan kompresi, semakin efektif metode kompresi yang digunakan.

2.5 File Teks

File teks merupakan adalah salah satu jenis berkas pada teknologi komputasi yang terstruktur sebagai urutan baris dari berkas elektronik, file yang elemennya berupa baris. File teks dibuat menggunakan editor teks, seperti SideKick, Edit (DOS), Notepad dan editor turbo pascal itu sendiri. Menurut Latif & Nasution [2] "*File text* merupakan *file* yang berisi informasi- informasi yang di sajikan dalam bentuk *text* yang merupakan kumpulan dari karaktek-karakter atau*string* yang menjadi satu kesatuan sedangkan *string* merupakan tipe data yang digunakan untuk menggambarkan atau mempresentasikan kumpulan karakter (huruf, angka, symbol)".

File teks adalah file yang berisi informasi dalam bentuk teks. Data dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat pada database merupakan contoh input data teks yang terdiri dari karakter, angka, dan tanda baca. Ketiga bentuk himpunan karakter teks tersebut biasanya digunakan untuk input dan output pada sistem kerja komputer yaituASCII, *Unicode* dan *EBCDIC*. ASCII (*American Standard Code for Information Interchange)* adalah standar internasional untuk huruf dan simbol kode (seperti heksadesimal dan *Unicode*), tetapi ASCII lebih bersifat global. ASCII pada sistem komputer atau perangkat komputer lainnya terdiri dari delapan digit biner, mulai dari "00000000" hingga "11111111".

File teks pada sistem komputer terdiri dari tipe karakter (char) dan tipe string. Jenis karakter (char) terdiri dari huruf, angka, tanda baca atau karakter khusus, seperti "aA", "01", "% ^", dll. Jenis string ini berisi nol atau beberapa jenis jenis karakter, seperti "terkompresi" dan "Goldbach G0". Format ekstensi file teks yang umum digunakan meliputi: format data teks (.txt), format data dokumen (.doc), dan format teks kaya (.rtf).

Format data teks (.txt) adalah format teks yang digunakan untuk menyimpan huruf, angka, karakter kontrol (tab, baris baru, dll.) Atau simbol lain yang biasa digunakan dalam penulisan, seperti titik, koma, tanda kutip, dll. Huruf, angka, karakter kontrol atau simbol dalam file teks menempati satu byte. Tidak seperti tipe teks yang diformat, hanya satu huruf yang dapat memakan waktu beberapa byte untuk memformat surat tersebut, seperti font, ukuran, baik dicetak tebal maupun tidak. Keunggulan format data teks ini adalah ukuran datanya kecil, karena tidak ada fungsi tampilan teks yang diformat. Saat ini, perangkat lunak yang paling banyak digunakan untuk memanipulasi format data ini adalah Notepad.

2.6 Algoritma *Goldbach Codes*

Menurut Salomon & Motta [10] "Algoritma *Goldbach Code* adalah algoritma yang dibuat oleh Peter Fenwick yang dibuat menggunakan *conjecture Goldbach*. *Conjecture Goldbach* ini diciptakan oleh Christian Goldbach yaitu salah satu matematikawan terkenal pada abad 17. *Conjecture Goldbach* berisi setiap bilangan genap lebih besar dari 2 merupakan hasil penjumlahan dari 2 buah bilangan prima".

Algoritma *Goldbach Codes* adalah algoritma yang mengasumsikan penggunaan teori konjektur Goldbach, yaitu semua bilangan genap positif yang lebih besar dari 2 merupakan penjumlahan dari dua bilangan prima. Dugaan ini pertama kali disebutkan dalam sebuah surat kepada Euler pada tahun 1742. Dia mengatakan dalam surat itu bahwa bilangan genap yang lebih besar dari atau sama dengan 4 juga dapat ditulis sebagai penjumlahan dari dua bilangan bulat, tetapi setelah 260 tahun, Euler tidak dapat memberikan bukti.Penelitian telah menunjukkan bahwa dugaan Goldbach hampir benar, tetapi buktinya tidak mencukupi.

Tabel 1. Tabel *Goldbach Codes* G0

n	2(n+3)	Primes	*Codeword*
1	8	3+5	11
2	10	3+7	101
3	12	5+7	011
4	14	3+11	1001
5	16	5+11	0101
6	18	7+11	0011
7	20	7+13	00101
8	22	5+17	010001
9	24	11+13	00011
10	26	7+19	0010001
11	28	11+17	000101
12	30	13+17	000011
13	32	13+19	0000101
14	34	11+23	00010001

Kode G0 yang tercantum dalam tabel 1 didasarkan pada bilangan prima (dari kanan kekiri) 23, 19, 17, 13, 11, 7, 5 dan 3. Jelas bahwa kode bertambah panjang, namun tidak monoton dan tidak ada ekspresi yang diketahui untuk panjang G0 (n) sebagai fungsi dari n, kode G0 efisien untuk nilai n kecil [4].

Proses Kompresi .Berikut urutan proses kompresi menggunakan algoritma *Goldbach codes* pada file teks :

1) Langkah pertama mencari frekuensi kemunculan setiap karakter.
2) Kemudian urutkan karakter berdasarkan frekuensi kemunculan dari yang terbesar hingga terkecil.
3) Selanjutnya cari nilai 2(n+3).
4) Setelah nilai dari n2 ditemukan, langkah berikutnya menentukan bilangan prima dan *codeword* G0 algoritma *goldbach codes* dari nilai n2.
5) Langkah selanjutnya adalah mencari nilai bit tiap karakter, dimana jumlah digit *codeword* mewakili nilai bit karakter.
6) Hasil dari algoritma Goldbach Codes adalah teks sebelum dikompresi kemudiandikompresi menggunakan *codeword* yang mewakili setiap karakter.

Proses Dekompresi menggunakan algoritma *Goldbach codes*, file teks yang telah dikompresi dapat dikembalikan menjadi file teks awal dengan cara:

1) mengambil nilai biner 1 dengan urutan ke – 2 diikuti dengan nilai biner dibelakangnya sebagai identitas setiap karakter, begitu selanjutnya. Misalkan biner 1001101, ambil nilai biner 1 diurutan ke 2 diikuti dengan nilai biner sebelumnya maka di dapat 1001. Nilai biner "1001" tersebut mewakili identitas sebuah karakter.
2) Kemudian setelah identitas karakter didapat, pembacaan karakter tersebut melalui proses header atau pembacaan ulang karakter yang menjadi kunci dalam proses dekompresi(Nasution, Ginting, Syahrizal, & Rahim, 2017)

2.7 Metode Shannon-Fano

Pengkodean Shannon Fano merupakan algoritma kompresi sinyal digital pertama yang diperkenalkan dalam makalahnya yang berjudul " *A Mathematical Theory ofCommunication*" pada tahun 1948. Shannon dan Fano terus mengembangkan algoritme, yang menghasilkan kata sandi biner untuk setiap karakter yang terdapat dalam data. Redundansi minimal. Pada saat itu, metode ini adalah metode terbaik, tetapi hampir tidak pernah digunakan, dan dikembangkan kembali setelah algoritma Huffman muncul. "Pada dasarnya metode ini menggantikan setiap simbol dengan sebuah alternatif kode biner yang panjangnya ditentukan berdasarkan probabilitas dari simbol tersebut" [8].

Algoritma *Shannon Fano* didasarkan pada *variable–length code* yang berarti beberapa karakter pada data yang akan dikodekan direpresentasikan dengankode (*codeword*) yang lebih pendek dari karakter yang ada pada data. Jika frekuensi kemunculan karakter semakin tinggi, maka kode semakin pendek. Dengan demikian kode yang dihasilkan tidaksama panjang, sehingga kode tersebut bersifat unik.

Pembuatan pohon Shannon Fano didasarkan pada proses dari atas ke bawah, sedangkan Hoffman membuat pohon dari nol. Panjang codeword *Shannon Fano* dapat dihitung dengan rumus berikut: ( ) ( ) (5) dimana merupakan probabiltas kemunculan simbol dan merupakan panjang *codeword*. Hal ini juga penting untuk dicatat bahwa kode *Shannon Fano* memenuhi kondisi kode awalan yang berarti bahwa tidak ada *codeword* yang sama untuk *codeword* setelahnya. Bentuk dasar *Shannon Fano* pada tabel berikut :

Tabel 2. Bentuk dasar *Shannon Fano*

A	P0	X1	X2	X3	X4	CODE
A1	0.36	0	0			00
A2	0.18	0	1			01
A3	0.18	1	0			10
A4	0.12	1	1	0		110
A5	0.09	1	1	1	0	1110
A6	0.07	1	1	1	1	1111

Cara Kerja Shannon Fano

a) Sebutkan probabilitas setiap simbol.
b) Urutkan daftar yang paling sering keluar.
c) Bagilah daftarnya menjadi dua, asalkan jumlah probabilitas di bagian atas mendekati setengah bagian bawah.
d) Bagian atas diberi nilai 0, dan bagian bawah diberi nilai 1.
e) Lakukan langkah terakhir secara rekursif di paruh atas dan bawah.
f) Setiap kode memiliki bit informasi yang berbeda.
g) Kode simbol dengan probabilitas rendah memiliki digit lebih tinggi, dan simbol dengan probabilitas tinggi memiliki digit lebih sedikit.
h) Sekalipun panjang kode setiap simbol berbeda, proses decoding masih dapatdiselesaikan.
i) Proses kompresi terjadi karena simbol yang sering muncul dikodekan dengansejumlah kecil bit.

3. HASIL DAN PEMBAHASAN

3.1 Hasil

Hasil penelitian ini adalah Implementasi hasil analisis pelaksanaan aplikasi dan hasil perancangan sistem ke dalam bahasa pemrograman. Realisasi sistem dalam penelitian ini dibangun dengan menggunakan software Visual Studio 2010 dan bahasa pemograman *Visual Basic.NET* Proses realisasi perancangan sistem terbagi menjadi 4 bentuk, yaitu: form *homepage/*Beranda, form menu utama (menu kompresi dan dekompresi), form help/bantuan dan form Tentang sistem.

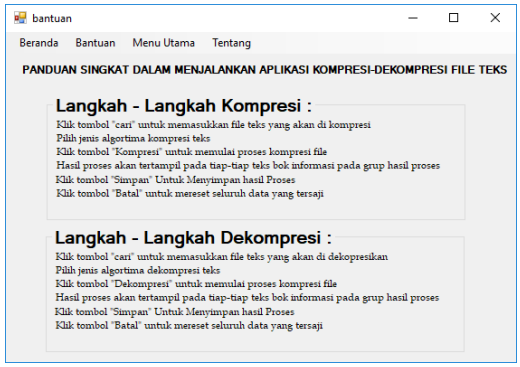
a. **Form *Beranda***

Form menu home page merupakan form pertama yang muncul saat menjalankan aplikasi. Formulir tersebut memiliki menu bar yang berisi empat menu kontrol tab, yaitu menu "Home", menu utama (menu kompresi dan dekompresi), menu "Help", dan menu "About". Tabel ini berisi opsi menu pada sistem, deskripsi judul, nama pembuat sistem, logo fakultas dan staf, dan departemen pembuat sistem. Gambar berikut menunjukkan tampilan menu utama.

Gambar 2. Tampilan Form Beranda

b. Form Bantuan

Apabila pengguna bingung dalam menggunakan sistem, formulir ini berisi tentang cara mengompres dan mendekompresi file agar pengguna dapat dengan mudah memahami penggunaan sistem. Gambar berikut menunjukkan tampilan formulir bantuan.



Gambar 3. Tampilan Form Bantuan

c. Form Menu Tentang

Jendela Formulir ini berisi informasi dari sistem dan pembuat sistem. Bentuk tampilan dapat dilihat pada gambar di bawah ini.

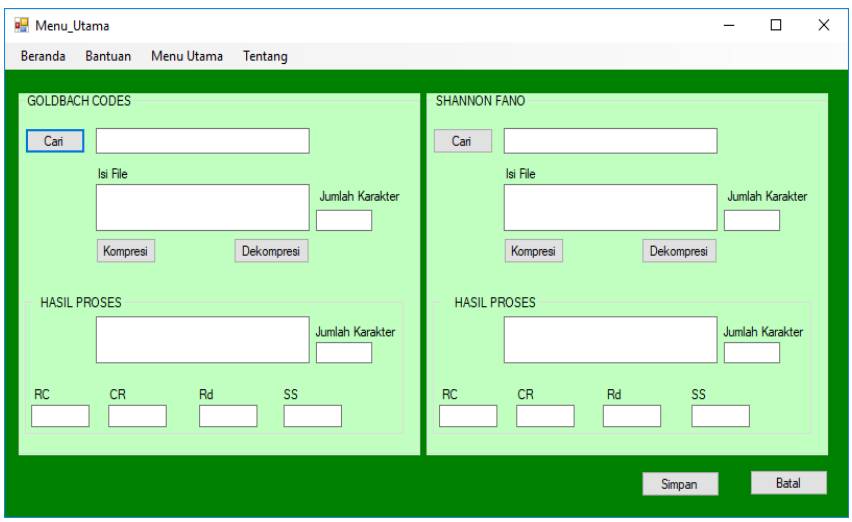
Gambar 4. Tampilan Form Menu Tentang

d. Form Menu Utama

Ketika pengguna memilih tab menu, formulir ini akan ditampilkan. Pengguna dapat memampatkan dan mendekompresi file, dan dapat memilih algoritma mana yang akan digunakan. Pada form ini terdapat beberapa *tombol*, *text box* dan label. Terdapat beberapa tombol yaitu tombol pencarian file, tombol kompres, tombol dekompres, tombolsimpan dan tombol batal.

Bentuk ini terbagi menjadi dua fungsi yaitu kompresi dan dekompresi. Fungsi tombol kompresi untuk mencari file adalah untuk mencari file yang akan dikompres. Setelah mendapatkan file, nama file akan muncul, dan konten file serta ukuran file akan muncul di kotak teks. Tombol kompres digunakan untuk mengompres file yang dipilih. Ketika tombol kompresi ditekan, proses kompresi akan terjadi, setelah analisis parameter kompresi dan perbandingan akan dihasilkan keluaran bit string dan ukuran file. Untuk mengompres ulang, klik tombol "Batal".

Fungsi tombol pencarian pada form yang telah dibuka adalah untuk menemukan file yang akan didekompresi. Setelah mendapatkan file, nama file akan muncul, dan konten file serta ukuran file akan muncul di kotak teks. Untuk mengekstrak file, tekan tombol dekompresi dan lakukan proses dekompresi, yang akan menghasilkan stringkeluaran dan ukuran file asli. Gambar berikut menunjukkan tampilan dalam bentuk menu.



Gambar 5. Tampilan Form Menu Utama

3.2 Pembahasan

Pada bagian ini akan dilakukan pengujian sistem untuk membuktikan bahwa sistem yang dibangun berfungsi dengan baik dan sesuai dengan analisis dan perancangan sistem sebelumnya. Dalam pengujian sistem dilakukan dengan proses kompresi dan dekompresi terhadap *file* berekstensi .txt atau .doc. Pada *file* tersebut akan disisipkan string dengan jumlah karakter dan frekuensi yang bervariasi atau berbeda-beda. Pada pengujian ini file yang digunakan adalah file teks berisikan sebuah karakter yang menggalami perkelipatan 10 dalam jumlah frekunsinya. Hasil pengujiannya dapat dilihat pada table berikut ini.

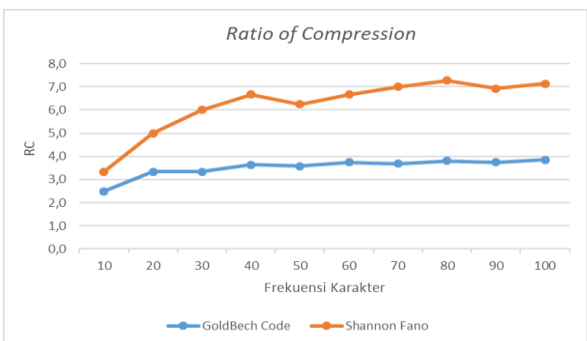
Tabel 3. Hasil Pengujian untuk Algortima *Goldbach Codes*

No	Karakter	Frekuensi	n(bit) Sebelum Kompresi	n(bit) Sesudah Kompresi	R C	CR (%)	Rd (%)	SS	Waktu Kompresi (ms)	Waktu Dekompresi (ms)
1	1	10	80	32	2,5	40	60	0,6	117	66
2	1	20	160	48	3,3	30	70	0,7	130	87
3	1	30	240	72	3,3	30	70	0,7	128	52
4	1	40	320	88	3,6	28	73	0,7	110	75
5	1	50	400	112	3,6	28	72	0,7	98	41
6	1	60	480	128	3,8	27	73	0,7	145	83
7	1	70	560	152	3,7	27	73	0,7	102	39
8	1	80	640	168	3,8	26	74	0,7	87	42
9	1	90	720	192	3,8	27	73	0,7	93	50
10	1	100	800	208	3,8	26	74	0,7	132	79

Tabel 4. Hasil Pengujian untuk Algortima *Shannon Fano*

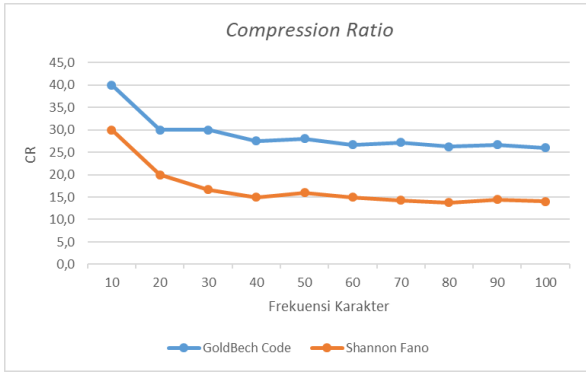
No	Karakter	Frekuensi	n(bit) Sebelum Kompresi	n(bit) Sesudah Kompresi	RC	CR (%)	Rd (%)	SS	Waktu Kompresi (ms)	Waktu Dekompresi (ms)
1	1	10	80	24	3,3	30	70	0,7	64	79
2	1	20	160	32	5	20	80	0,8	87	77
3	1	30	240	40	6	17	83	0,8	63	85
4	1	40	320	48	6,7	15	85	0,9	102	44
5	1	50	400	64	6,3	16	84	0,8	68	56
6	1	60	480	72	6,7	15	85	0,9	80	35
7	1	70	560	80	7	14	86	0,9	127	68
8	1	80	640	88	7,3	14	86	0,9	161	43
9	1	90	720	104	6,9	14	86	0,9	63	43
10	1	100	800	112	7,1	14	86	0,9	120	65

Berdasarkan tabel 3 dan tabel 4 di atas memperlihatkan hasil kompresi untuk file teks yang memilik karakter tungal dengan variasi frekunsi berbeda-beda menghasilkan nilai perhitungan *Ratio of Compression*, *Compression Ratio*, *Redudancy*, dan *Space Saving* rata-rata bernilai sama. Namun agar lebih memudahkan untuk melihat perbandingan efektifitas kode *Goldbach* dan algoritma *Shannon Fanno* untuk hasil uji kompresi tiap parameter dalam bentuk grafik. Berikut gambar grafik dari hasil perbadingan tiap parameter proses kompresi.



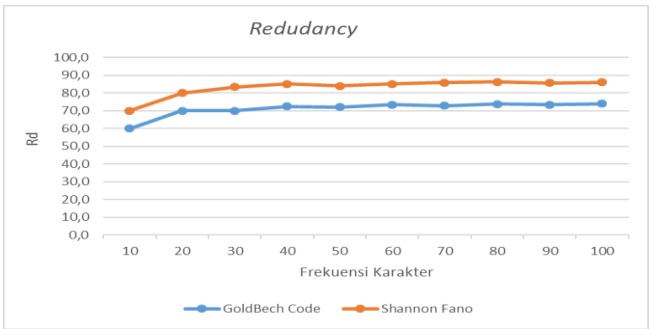
Gambar 6. Grafik Perbandingan *Ratio of Compression*

Dari gambar diatas dilihat *Ration of Comparation* dari penerapan algortima *Shannon fano* untuk kasus karakter tunggal dan frekunsi yang bervariasi trus mengalami peningkatan RC siring meningkatnya frekuensi karakter. Dibandingkan untuk RC pada algoritma *Goldbech Codes*, hanya mengalami sedikit perubahan penambahan nilai RC ketika frekuensi karakter trus meningkat.

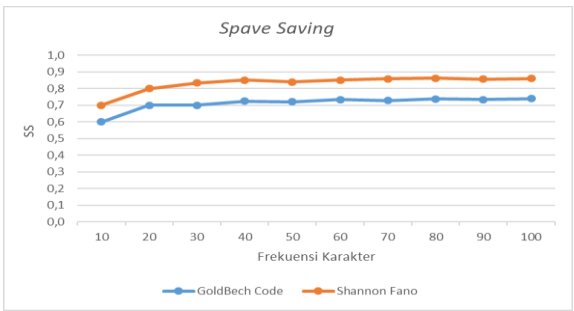


Gambar 7. Grafik Perbandingan *Compression Ratio*

Dari gambar diatas Comparation Ratio untuk algoritma *GoldBach codes* lebih sedikit besar dengan selisih nilai lebih kurang 10% dari CR algortima *Shannon Fano* untuk steiap uji coba file. CR kedua algoritma terus mengalami penurunan nilai secara konstan seiring dengan perubahan ferkunsi karakter yang digunakan.

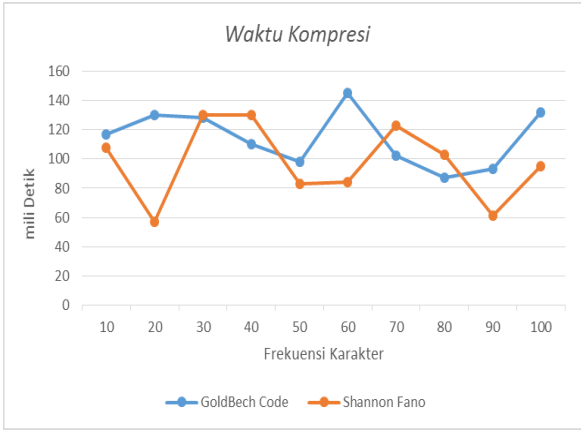


Gambar 8. Grafik Perbandingan *Redudancy*



Gambar 9. Grafik Perbandingan *Space Saving*

Gambar di atas menunjukkan hasil grafik penghematan ruang pengulangan karakter *(space savung)* dalam kode *Goldbach* dan algoritma *Shannon Fano*. Dapat disimpulkan dari gambar ini bahwa walaupun perbedaan nilai tidak berjauhan, ruang yang dihemat oleh hasil kompresi dari algoritma *Shannon Fano* lebih besar daripada yang ada pada algoritma Kode *Goldbach*. Artinya algoritma *Shannon Fano* lebih baik daripada algoritma *Goldbach* Code karena memiliki nilai rata-rata penghematan ruang yang lebih besar. Semakin tinggi frekuensi karakter, semakin kecil nilai penghematan ruang dari kedua algoritma ini. Artinya, ruang yang dihemat ketika mengulang karakter berbanding terbalik dengan frekuensi karakter.



Gambar 10. Grafik Perbandingan Waktu Kompresi

Gambar 10 menunjukkan diagram waktu kompresi dari karakter berulang pada algoritma *Goldbach Codes* dan *Shannon Fano*. Perubahan waktu dari masing-masing algoritma hampir sama, dan periode waktu perubahan dinamis lebih sedikit.Namun, jika menilai dari hasil gambar diatas, penulis bisa mendapatkan kompresi Akhir waktu. Algoritma *Shannon Fano* memiliki waktu kompresi yang sedikit lebih baik dari pada algoritma *Goldbach Codes*.

4. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan mengenai analisis penerapan algoritma *Goldbach Codes* dan metode *Shannon Fano* dalam kompresi file teks, dapat ditarik kesimpulan sebagai berikut:

1) Aplikasi yang dirancang dalam penelitian ini telah dapat menggunakan algoritma *Goldbach Codes* dan metode *Shannon Fano* untuk melakukan proses kompresi dan dekompresi.
2) Pengujian menggunakan file teks yang berisi jumlah karakter yang sama dengan frekuensi yang berbeda menunjukkan bahwa nilai yang mempengaruhi kecepatan kompresi, kecepatan kompresi, penghematan ruang dan redundansi adalah jumlah karakter, bukan frekuensi karakter.
3) Pengujian menggunakan file teks yang berisi beberapa karakter berbeda dengan frekuensiyang sama menunjukkan bahwa jika jumlah karakter kurang dari 15 karakter, maka performa kode Algortima *Goldbach* akan efektif; dan bila jumlah karakter lebih dari 15 karakter, gunakan Performa algoritma Shannon Fano masih sangat efektif.
4) Berdasarkan rasio kompresi dan nilai penghematan ruang yang diperoleh dari pengujian file teks, diketahui bahwa algoritma *Shannon Fano* lebih efektif untuk mengompresi file teks jika diukur dengan performansi algoritma *Goldbach Codes*.

DAFTAR PUSTAKA

[1] Kadir, Abdul. (2014). *Pengenalan Sistem Informasi Edisi Revisi*. Andi.Yogyakarta.

[2] Latif, M. A., & Nasution, S. D. (2018). *Analisa Perb andingan Algoritma Rice Codes Dengan Algoritma Goldbach Codes Pada Kompresi File Text Menggunakan Metode Exponential*. *13*, 5.

[3] Lubis, A. H., Nasution, S. D., & Ulfa, K. (2017). *Penerapan Algoritma Goldbach Codes Dalam Pemampatan Short Message Service Berbasis Android*. 5.

[4] Lubis, I. R. (2017). *Analisa Perbandingan Algoritma Huffman Dengan Algoritma Transformasi Burrows Wheeler Pada Kompresi Citra Menggunakan Metode Eksponensial*. *16*, 3.

[5] Manullang, D. I. (2018). *Perancangan Aplikasi Penyandian File Teks Dengan Algoritma Bifid Cipher*. *17*, 6.

[6] Nasution, S. D., Ginting, G. L., Syahrizal, M., & Rahim, R. (2017). Data Security Using Vigenere Cipher And Goldbach Codes Algorithm. *International Journal Of Engineering Research*, *6*(01), 5.

[7] Pane, M. R. (2017). *Perancangan Aplikasi Kompresi Menggunakan Metode Shannon Fano Dan Unary Coding Pada File Teks*. *12*, 6.

[8] Prasetio, M. R. (2019). Analisis Perbandingan Kinerja Algoritma Shannon Fano Dan Levenstein Code Pada Kompresi File Video. 99.

[9] Putra, Darma. 2010. Pengolahan Citra Digital: "Kompresi Data Citra" Yogyakarta :Andi Offset

[10] Salomon, D., & Motta, G. (2010). *Handbook Of Data Compression*. Springer Science & Business Media.

[11] Sugiyono. (2011). *Memahami Penelitian Kualitatif.* Bandung : Alfabeta

[12] Verma, Priyanka & Singh, Anupam (2016). "Fostering Stakeholder Trust through CSR Reporting: An Analytical Focus". IIM Kozhikode Society & Management Review. 186– 199. (2016).