IMPLEMENTASI ALGORITMA HUFFMAN DALAM KOMPRESI DATA PADA DATABASE

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika

Disusun oleh:

Nama : Ridwan Wulida Siam
NIM : 14650024

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA

2018

HALAMAN PENGESAHAN

Universitas Islam Negeri Sunan Kalijaga **FM-UINSK-BM-05-07/R0**

PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : B-30/UIN.02/D.ST/PP.01.1/05/2018

Skripsi/Tugas Akhir dengan judul : "Implementasi Algoritma Huffman dalam Kompresi Data pada Database"

Yang dipersiapkan dan disusun oleh :
Nama : Ridwan Wulida Siam
NIM : 14650024
Telah dimunaqasyahkan pada : 4 Mei 2018
Nilai Munaqasyah : A-
Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

M. Didik R Wahyudi, M.T
NIP. 19760812 200901 1 015

Penguji I

Dr.Bambang Sugiantoro
NIP.19751024 200912 1 002

Penguji II

Rahmat Hidayat.S.Kom.M.CS
NIP.19850514 201503 1 002

Yogyakarta, 14 Mei 2018
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan

Dr. Murtono, M.Si
NIP. 19691212 200003 1 001

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi
Lamp : -

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Ridwan Wulida Siam
NIM : 14650024
Judul Skripsi : " IMPLEMENTASI ALGORITMA HUFFMAN DALAM KOMPRESI DATA PADA DATABASE"

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Teknik Informatika.

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 24 April 2018
Pembimbing

M. Didik Rohmad Wahyudi, S.T., MT.
NIP. 19760812 200901 1 015

PERNYATAAN KEASLIAN SKRIPSI

SURAT KETERANGAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini:

Nama	: Ridwan Wulida Siam
NIM	: 14650024
Program Studi	: Teknik Informatika
Fakultas	: Sains dan Teknologi

Menyatakan bahwa skripsi saya yang berjudul **"Implementasi Algoritma Huffman Dalam Kompresi Data Pada Database"** merupakan hasil penelitian saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi, dan bukan plagiasi karya orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 23 April 2018

Yang menyatakan,

Ridwan Wulida Siam

NIM. 14650024

KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat Allah SWT, yang telah melimpahkan rahmat, hidayah serta karunia-Nya, sehingga penyusun dapat menyelesaikan Penelitian dengan judul "**Implementasi Algoritma Huffman Dalam Kompresi Data Pada Database**". Shalawat serta salam semoga selalu tercurahkan kepada uswatun hasanah kita, Nabi Muhammad SAW beserta keluarga, para sahabat dan pengikutnya termasuk kita semua yang senantiasa menantikan syafa'atnya kelak di Hari Akhir.

Penyusun menyadari bahwa penyusunan laporan penelitian ini tidak akan berjalan lancar tanpa dukungan dari berbagai pihak. Oleh karena itu, penyusun mengucapkan terima kasih banyak kepada:

1. Prof. Drs. Yudian Wahyudi, MA, Ph.D, selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Dr. Murtono, M.Si selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Dr. Bambang Sugiantoro, MT. selaku Ketua Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta sekaligus pembimbing akademik.
4. Bapak Muhammad Didik Rohmad Wahyudi, S.T., MT. selaku pembimbing tugas akhir, yang telah memberikan banyak sekali bantuan saran, nasehat, yang sangat bermanfaat dan juga telah mengarahkan saya selama ini.

5. Bapak Sumarsono, S.T., M.KOM. selaku pembimbing akademik, yang telah memberikan banyak sekali bantuan saran, nasehat, yang sangat bermanfaat dan juga telah mengarahkan saya selama ini.
6. Para dosen Teknik Informatika yang saya hormati yang telah memberikan banyak ilmunya, tidak hanya ilmu pengetahuan tapi juga pengalaman dan nasehat yang sangat berguna, kepada saya selama perkuliahan.
7. Kepada teman-teman seperjuangan saya, Mahasiswa Teknik Informatika 2014 yang telah menemani saya berproses selama ini, dan telah memberikan banyak sekali pelajaran berharga pada saya.
8. Kepada semua teman-teman di ITTC UIN Sunan Kalijaga Yogyakarta, yang sudah banyak membantu dalam proses penelitian ini.
9. Serta kepada semua pihak yang tidak bisa saya sebutkan satu persatu, yang sudah membantu keberlangsungan penelitian ini secara langsung maupuntidak langsung.

Penulis menyadari masih banyak sekali kekurangan dan kelemahan dalam penelitian ini. Oleh karena itu, segala kritik dan saran senantiasa penulis harapkan dari pembaca. Akhir kata, semoga penelitian ini dapat bermanfaat bagi dunia pendidikan dan segala pihak, terutama bagi penulis sendiri.

Yogyakarta, 24 April 2018

Penulis

HALAMAN PERSEMBAHAN

Dengan rasa syukur yang teramat besar, penulis persembahkan tugas akhir ini kepada:

1. Yang pertama saya persembahkan kepada kedua orang tua saya Bapak Drs. Sujarwanto dan Alm Ibu Purwanti yang telah mendidik dan membesarkan saya hingga saya dapat menyelesaikan tugas akhirnya dan menjadi pribadi yang lebih baik lagi.
2. Kepada kedua saudara perempuan saya mbak Adani dan Ginaris yang telah memberikan dukungan secara tidak langsung termasuk dukungan dalam bentuk semangat.
3. Kepada Om Koni dan Buatik sekeluarga yang telah memberikan fasilitas yang nyaman untuk tinggal dijogja selama berkuliah.
4. Kepada Sahabat saya Ireicca telah memberikan dukungan semangat dan lainnya.
5. Kepada Iqbal dan Hafiez teman main bareng mobile legend yang membantu merefresh pikiran jika menang
6. Kepada teman seperjuangan teknik informatika angkatan 2014 dan teman-teman ITTC

HALAMAN MOTTO

"Karena sesungguhnya sesudah kesulitan itu ada kemudahan."

(QS. Alam Nasyroh: 5)

"*Do Cool Thing That Matter*"

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

IMPLEMENTASI ALGORITMA HUFFMAN DALAM KOMPRESI DATA PADA DATABASE

RIDWAN WULIDA SIAM

14650024

INTISARI

Perkembangan teknologi semakin hari semakin cepat dan banyak, mengakibatkan banyaknya data digital yang tersimpan dalam sebuah media penyimpanan. Terlebih jika data tersebut trus disimpan dalam waktu yang tidak ditentukan. Sehingga besarnya media penyimpanan yang dibutuhkan untuk menampung data tersebut. Salah satu cara untuk menyelesaikannya adalah dengan memampatkan data tersebut hingga ukurannya menjadi lebih kecil. Algoritma huffman adalah salah satu algoritma kompresi data teks terbaik.

Namun dalam algoritma huffman ini terdapat beberapa kelemahan antara lain dalam pembentukan pohon. Dalam penelitian ini menawarkan konsep algoritma huffman dengan dua buah pohon. Yang artinya dapat memangkas setengah dari pembentukan satu pohon huffman. Dengan menggunakan dua buah pohon huffman dalam pembentukan pohon terutama kode prefik akan menjadi lebih pendek.

Tentunya dalam penerapan kompresi data huffman ini dalam sebuah program akan memakan memori dan waktu yang lebih besar karena adanya proses tambahan sebelum data tersebut disimpan ataupun ditampilkan.

Kata kunci : Algoritma Huffman, Kompresi, Dekompresi, Pohon Huffman

IMPLEMENTATION OF HUFFMAN ALGORITHM FOR DATA COMPRESSION ON DATABASE

RIDWAN WULIDA SIAM

14650024

ABSTRACT

Technological developments are getting faster and faster, resulting in a lot of digital data stored in a storage medium. Especially if the data is then stored in an unspecified time. So the amount of storage media needed to accommodate the data. One way to solve it is to compress the data until its size becomes smaller. The huffman algorithm is one of the best text data compression algorithms.

But in this huffman algorithm there are several weaknesses, among others, in the formation of trees. In this study offers the concept of huffman algorithm with two pieces of trees. Which means it can cut half of the formation of a huffman tree. By using two huffman trees in the formation of trees especially the prefix code will become shorter.

Of course, in the application of huffman data compression in a program will take up more memory and time due to additional processes before the data is stored or displayed.

Keywords: Huffman Algorithm, Compression, Decompress, Huffman Tree

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, perkembangan teknologi informasi dan komunikasi sangat cepat dan pesat. Teknologi informasi dan komunikasi sudah menjadi bagian penting dan mempengaruhi kehidupan di beberapa bidang. Banyak kemudahan yang diberikan, salah satunya kemudahan dalam penulisan dan penyimpanan data dokumen dalam bentuk digital. Dengan kemudahan yang ada, tidak berbanding lurus terhadap media penyimpanan dokumen digital tersebut karena pesatnya pertumbuhan terhadap data dokumen yang semakin hari semakin besar. Sedangkan dalam media penyimpanan yang ada saat ini masih sangat terbatas. Sebagai contoh kasus saja pada portal berita online yang tiap harinya rata-rata bertambah lebih dari 40 berita untuk satu kategori. Bayangkan saja jika sebuah portal berita memiliki lima kategori berita, maka setiap harinya mereka harus menyimpan setidaknya 200 berita belum ditambah jika pada saat itu sedang terjadi kasus atau kejadian besar misalkan saja terjadi bencana alam disuatu daerah atau pada moment tertentu contohnya saat ada pemilihan kepala daerah pastinya update berita pada portal berita tersebut makin besar lebih dari 40 berita.

Kompresi data adalah sebuah cara untuk memadatkan data sehingga hanya memerlukan ruang penyimpanan lebih kecil sehingga lebih efisien dalam penyimpanan atau dapat mempersingkat waktu pertukaran data tersebut. Salah satu algoritma untuk kompresi data yang paling terkenal

adalah algoritma huffman. Algoritma huffman adalah yang konsepnya mengurangi redudansi pada data yang ada dengan membentuk binari. Dengan melakukan kompresi data diharapkan data dokumen akan memiliki ukuran yang lebih kecil sehingga dapat menghemat media penyimpanan. Namun algoritma kompresi data huffman dengan membentuk langsung sebuah pohon huffman akan menimbulkan besarnya lebar dan dalamnya level dalam pohon tersebut sehingga perlu sebuah solusi untuk mengurangi kedalaman dan lebar pohon huffman tersebut untuk penentuan kode prefik di setiap karakternya.

1.2 Rumusan Masalah

Berdasarkan penjelasan latar belakang diatas, maka rumusan masalah yang akan di bahas adalah

1. Bagaimana melakukan implementasi algoritma Huffman untuk kompresi dan dekompresi data dengan menggunakan dua buah pohon huffman?
2. Mengetahui sejauh mana hasil dari algoritma huffman dengan menggunakan dua buah pohon huffman dari segi penyimpanan dan performa yang dihasilkan?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah :

1. Mengetahui penerapan algoritma Huffman untuk kompresi dan dekompresi data dengan menggunakan dua buah pohon huffman.

2. Mengetahui performa algoritma kompresi data huffman dengan menggunakan dua buah pohon huffman dibandingkan dengan satu pohon huffman.
3. Mengetahui dampak apa saja yang dihasilkan dalam penerapan algoritma kompresi tersebut dalam portal berita online dari segi penyimpanan data dan performa.

1.4 Batasan Masalah

Agar penelitian lebih terarah dan tidak menyimpang dari rumusan masalah yang ada, maka batasan masalah dari penelitian ini adalah :

1. Algoritma yang digunakan untuk proses kompresi dan dekompresi adalah algoritma Huffman.
2. Data yang akan diproses adalah bertipe karakter abjad (a-z) dan spasi.
3. Menghitung rasio hasil kompresi dan performa waktu dan memori.
4. Database yang digunakan adalah MYSQL.

1.5 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah dapat mengetahui seberapa efisien kompresi data menggunakan algoritma Huffman dengan menggunakan dua buah pohon huffman dalam kasus portal berita online. Diharapkan juga dapat menyediakan aplikasi yang dapat diterapkan pada website sehingga dapat menghemat penyimpanan terhadap data dari website tersebut. Terlebih pada website yang memiliki penyimpanan terbatas atau kecil.

BAB V
PENUTUP

5.1 Kesimpulan

Dari proses implementasi dan pengujian terhadap algoritma huffman pada penelitian ini dapat ditarik kesimpulan sebagai berrikut

1. Pembentukan 2 pohon huffman pada proses awal adalah proses yang sangat penting dan sangat mempegaruhi hasil dari kompresi data. Semakin data latih merepresentasikan data yang akan dikompresi maka hasil dari proses kompresi akan memiliki ukuran file yang lebih kecil.
2. Rasio kompresi algoritma huffman dengan 2 pohon huffman dan dengan 1 pohon dapat mencapai ≥39,75% dilihat dari hasil pengujian yang telah dilakukan.
3. Algoritma huffman dengan menggunakan 2 pohon menghasilkan rasio kompresi lebih besar dari pada algoritma huffman dengan 1 pohon saja jika data yang digunakan banyak atau besar.
4. Algoritma huffman dengan 2 pohon juga memiliki waktu eksekusi lebih cepat dalam menyimpan maupun membaca 1 data, tetapi memiliki waktu lebih lama dalam membaca banyak data.
5. Algoritma huffman dengan 2 pohon dan 1 pohon ini membutuhkan waktu proses yang tidak terlalu jauh berbeda untuk proses kompresi lalu menyimpan data, namun dalam segi memori yang dibutuhkan lebih besar daripada tidak menggunakan algoritma huffman.

6. Dalam proses membaca data berkebalikan dengan proses menyimpan data, waktu proses yang dibutuhkan lebih lama tetapi memakan memori yang lebih sedikit.
7. Dengan menggunakan algoritma huffman ini keamanan data pada penyimpanan lebih terjaga karena hasil kompresi tidak bisa secara langsung dibaca oleh manusia atau komputer tanpa melalui proses dekompresi terlebih dahulu.

5.2 Saran

Dalam proses penilitian ini masih terdapat banyak kekurangan, oleh karena itu penulis memberikan saran untuk penelitian selanjutnya, sebagai berikut:

1. Dapat diimplementasikan pada data lain selain teks, seperti gambar, audio, dan video sehingga lebih teruji konsep algoritma huffman dengan menggunakan 2 pohon huffman.
2. Dalam pembentukan pohon huffman karakter lain dapat menjadi pertimbangan selanjutnya.
3. Memadukan dengan algoritma lain seperti pembelajaran mesin.

DAFTAR PUSTAKA

Abdullah, M. M. (2008). *Kompresi String Menggunakan Algoritma LZW dan Huffman.* Bandung: Teknik Informatika Institut Teknologi Bandung.

Adrisatria, Y. (2013). *Penerapan A;goritma Huffman Dalam Dunia Kriptografi.* Bandung: Program Studi Teknik Informatika, Institute Teknologi Bandung.

Gorn, S. (1963). American Standard Code for Information Interchange. Communication of the ACM.

Hanif, I. (n.d.). *Kompresi Teks Menggunakan Algoritma Dan Pohon Huffman.* Bandung: Program Studi Teknik Informatika Institut Teknologi Bandung.

Huffman, D. A. (1952). A Method For The Construction Of Minimum-Redudancy Codes. *Massachusetts Institute Of Technology, Cambridge, Mass*, 1098 - 1101.

Nuryasin. (2014). *Perbandingan Kompresi File Data Dengan Algoritma Huffman, Half Byte, dan Run Length.* Jakarta: Universitas Islam Negeri Syarif Hidayatullah Jurnal Sistem Informasi, 7(2),2014, 1-6.

Prasetyo, Y. B. (2016). *Perbandingan Kompresi Teks Menggunakan Algoritma Huffman Statis, Huffman Dinamis, dan Modifikasi Algoritma Huffman.* Yogyakarta: Universitas Sanata Dharma Yogyakarta.

Shanmugasundaram, S., & Lourdusamy, R. (2011). A Comparative Study Of Text Compression Algorithms. *International Journal Of Wisdom Based Computing Vol 1 (3)*, 68-76.

Siahaan, A. P. (2016). *Impelementasi Teknik Kompresi Teks Huffman.* Medan: Fakultas Ilmu Komputer, Universitas Pembangunan Panca Budi.

Wahyudi, M. D. (2016). Pengambilan Isi Berita Online Dengan Document Object Model Berbasis PHP Untuk Sumber Data Mining. 7.

LAMPIRAN

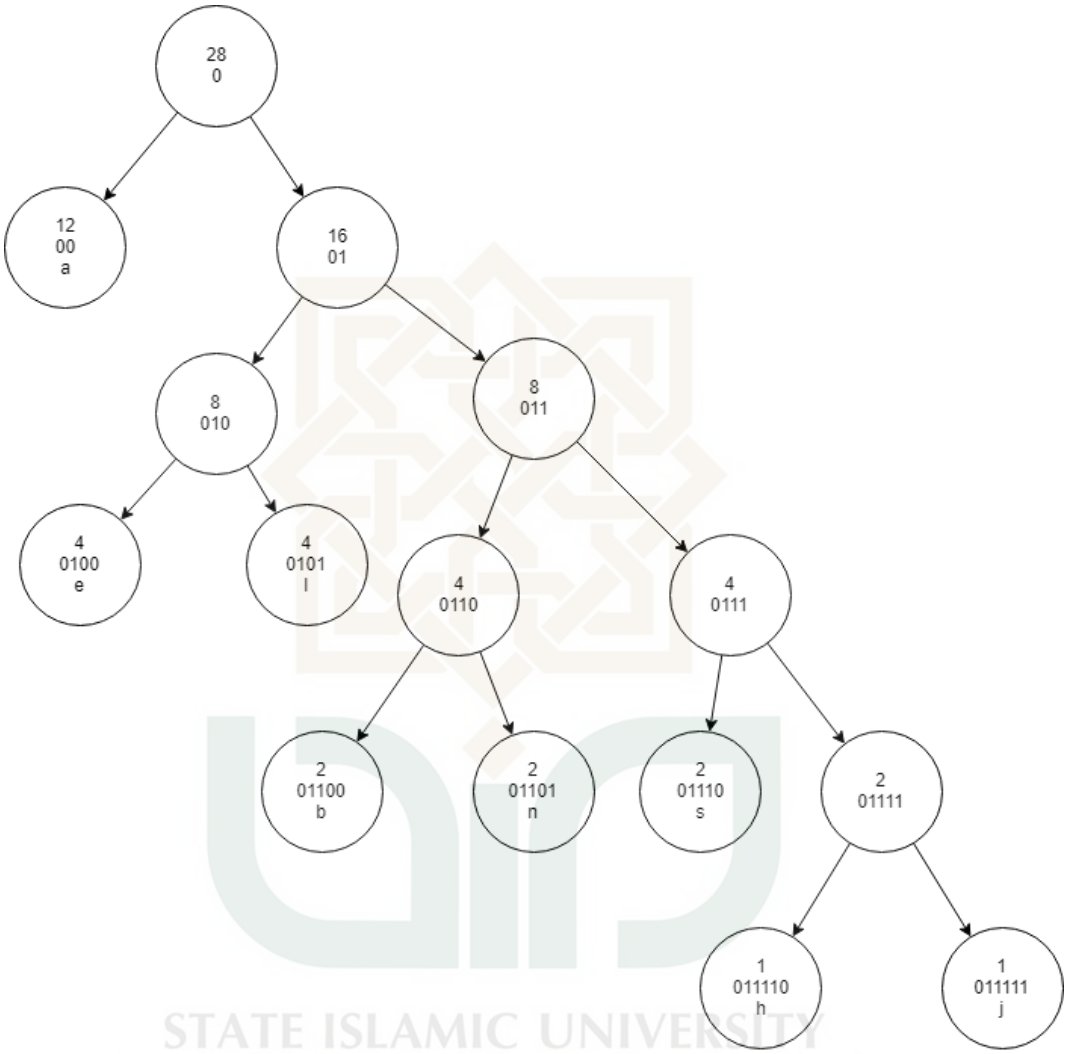
LAMPIRAN A : Hasil pada Data Percobaan

1. Pohon Huffman pada Percobaan Data Latih

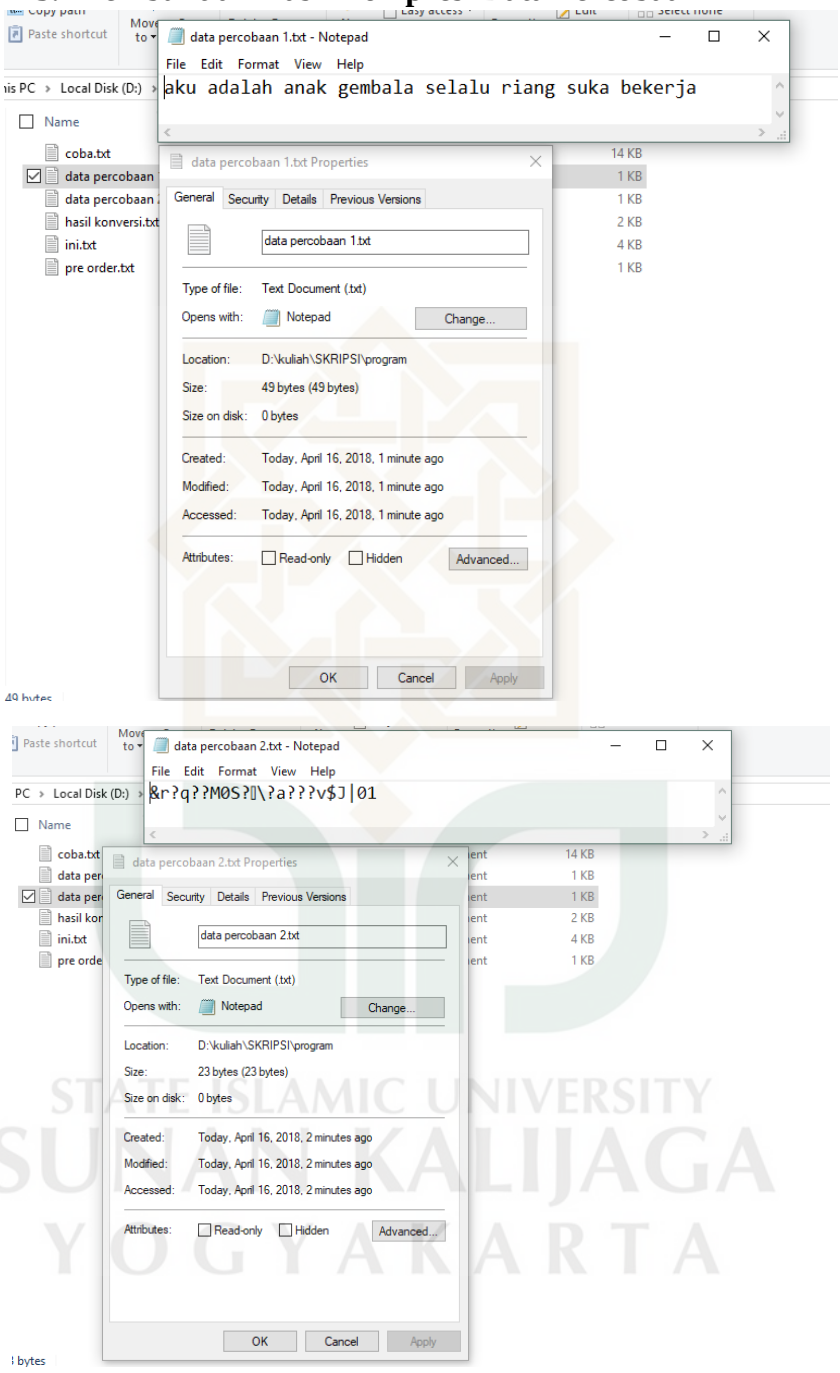
```
Node Object (
    [value] =>
    [huruf] =>
    [count] => 28
    [prefixCode] => 0
    [left] => Node Object (
            [value] => 97
            [huruf] => a
            [count] => 12
            [prefixCode] => 00
            [left] =>
            [right] =>
        )
    [right] => Node Object (
            [value] =>
            [huruf] =>
            [count] => 16
            [prefixCode] => 01
            [left] => Node Object (
                    [value] =>
                    [huruf] =>
                    [count] => 8
                    [prefixCode] => 010
                    [left] => Node Object (
                            [value] => 101
                            [huruf] => e
                            [count] => 4
                            [prefixCode] => 0100
                            [left] =>
                            [right] =>
                        )
                    [right] => Node Object (
                            [value] => 108
                            [huruf] => l
                            [count] => 4
                            [prefixCode] => 0101
                            [left] =>
                            [right] =>
                        )
                )
            [right] => Node Object (
                    [value] =>
                    [huruf] =>
                    [count] => 8
                    [prefixCode] => 011
                    [left] => Node Object (
                            [value] =>
                            [huruf] =>
                            [count] => 4
                            [prefixCode] => 0110
                            [left] => Node Object (
                                    [value] => 98
                                    [huruf] => b
                                    [count] => 2
                                    [prefixCode] => 01100
                                    [left] =>
                                    [right] =>
```

```
                                )
                        [right] => Node Object (
                                [value] => 110
                                [huruf] => n
                                [count] => 2
                                [prefixCode] => 01101
                                [left] =>
                                [right] =>
                                )
                        )
                [right] => Node Object (
                        [value] =>
                        [huruf] =>
                        [count] => 4
                        [prefixCode] => 0111
                        [left] => Node Object (
                                [value] => 115
                                [huruf] => s
                                [count] => 2
                                [prefixCode] => 01110
                                [left] =>
                                [right] =>
                                )
                        [right] => Node Object (
                                [value] =>
                                [huruf] =>
                                [count] => 2
                                [prefixCode] => 01111
                                [left] => Node Object (
                                        [value] => 104
                                        [huruf] => h
                                        [count] => 1
                                        [prefixCode] => 011110
                                        [left] =>
                                        [right] =>
                                        )
                                [right] => Node Object (
                                        [value] => 106
                                        [huruf] => j
                                        [count] => 1
                                        [prefixCode] => 011111
                                        [left] =>
                                        [right] =>
                                        )
                                )
                        )
                )
        )
)
```

2. Gambaran Visual Pohon Huffman Pada Percobaan



3. Pembuktian Hasil Kompresi Data Percobaan



LAMPIRAN B : Hasil pada Pengujian

1. Pohon Huffman pada Pengujian

```
Array (
    [0] => stdClass Object (
            [value] =>
            [huruf] =>
            [count] => 943380
```

```
            [prefixCode] => 0
            [left] => stdClass Object (
                    [value] =>
                    [huruf] =>
                    [count] => 382001
                    [prefixCode] => 00
                    [left] => stdClass Object (
                            [value] =>
                            [huruf] =>
                            [count] => 159023
                            [prefixCode] => 000
                            [left] => stdClass Object (
                                    [value] => 114
                                    [huruf] => r
                                    [count] => 77628
                                    [prefixCode] => 0000
                                    [left] =>
                                    [right] =>
                                )
                            [right] => stdClass Object (
                                    [value] =>
                                    [huruf] =>
                                    [count] => 81395
                                    [prefixCode] => 0001
                                    [left] => stdClass Object (
                                            [value] =>
                                            [huruf] =>
                                            [count] => 34558
                                            [prefixCode] => 00010
                                            [left] => stdClass Object (
                                                    [value] =>
                                                    [huruf] =>
                                                    [count] => 15375
                                                    [prefixCode] =>
000100
                                                    [left] => stdClass
Object (
                                                            [value] =>
                                                            [huruf] =>
                                                            [count] =>
7043
                                                            [prefixCode]
=> 0001000
                                                            [left] =>
stdClass Object (

[value] =>

[huruf] =>

[count] => 2676

[prefixCode] => 00010000

[left] => stdClass Object (

[value] => 113

[huruf] => q

[count] => 420

[prefixCode] => 000100000
```

```
[left] =>
[right] =>
                                                                    )
[right] => stdClass Object (
[value] => 118
[huruf] => v
[count] => 2256
[prefixCode] => 000100001
[left] =>
[right] =>
                                                                    )
                                                                )
                                                            [right] =>
stdClass Object (
[value] => 102
[huruf] => f
[count] => 4367
[prefixCode] => 00010001
[left] =>
[right] =>
                                                                )
                                                        )
                                                    [right] => stdClass
Object (
                                                            [value] =>
119
                                                            [huruf] => w
                                                            [count] =>
8332
                                                            [prefixCode]
=> 0001001
                                                            [left] =>
                                                            [right] =>
                                                        )
                                                )
                                            [right] => stdClass Object (
                                                    [value] => 106
                                                    [huruf] => j
                                                    [count] => 19183
                                                    [prefixCode] =>
000101
                                                    [left] =>
                                                    [right] =>
                                                )
                                        )
                                    [right] => stdClass Object (
                                            [value] => 108
                                            [huruf] => l
```

```
                                        [count] => 46837
                                        [prefixCode] => 00011
                                        [left] =>
                                        [right] =>
                                    )

                            )
                        )
                    [right] => stdClass Object (
                            [value] =>
                            [huruf] =>
                            [count] => 222978
                            [prefixCode] => 001
                            [left] => stdClass Object  (
                                    [value] => 101
                                    [huruf] => e
                                    [count] => 110050
                                    [prefixCode] => 0010
                                    [left] =>
                                    [right] =>
                            )
                            [right] => stdClass Object (
                                    [value] =>
                                    [huruf] =>
                                    [count] => 112928
                                    [prefixCode] => 0011
                                    [left] => stdClass Object (
                                            [value] => 103
                                            [huruf] => g
                                            [count] => 52011
                                            [prefixCode] => 00110
                                            [left] =>
                                            [right] =>
                                    )
                                    [right] => stdClass Object (
                                            [value] =>
                                            [huruf] =>
                                            [count] => 60917
                                            [prefixCode] => 00111
                                            [left] => stdClass Object (
                                                    [value] => 9
                                                    [huruf] =>
                                                    [count] => 26633
                                                    [prefixCode] =>
001110
                                                    [left] =>
                                                    [right] =>
                                            )
                                            [right] => stdClass Object (
                                                    [value] => 111
                                                    [huruf] => o
                                                    [count] => 34284
                                                    [prefixCode] =>
001111
                                                    [left] =>
                                                    [right] =>
                                            )
                                    )
                            )
                    )
            )
        [right] => stdClass Object (
                [value] =>
```

```
                [huruf] =>
                [count] => 561379
                [prefixCode] => 01
                [left] => stdClass Object (
                        [value] =>
                        [huruf] =>
                        [count] => 278631
                        [prefixCode] => 010
                        [left] => stdClass Object (
                                [value] =>
                                [huruf] =>
                                [count] => 138703
                                [prefixCode] => 0100
                                [left] => stdClass Object (
                                        [value] => 109
                                        [huruf] => m
                                        [count] => 64268
                                        [prefixCode] => 01000
                                        [left] =>
                                        [right] =>
                                    )
                                [right] => stdClass Object (
                                        [value] => 117
                                        [huruf] => u
                                        [count] => 74435
                                        [prefixCode] => 01001
                                        [left] =>
                                        [right] =>
                                    )
                            )
                        [right] => stdClass Object (
                                [value] => 110
                                [huruf] => n
                                [count] => 139928
                                [prefixCode] => 0101
                                [left] =>
                                [right] =>
                            )

                    )
                [right] => stdClass Object (
                        [value] => 97
                        [huruf] => a
                        [count] => 282748
                        [prefixCode] => 011
                        [left] =>
                        [right] =>
                    )
            )
    )
[1] => stdClass Object (
        [value] =>
        [huruf] =>
        [count] => 805340
        [prefixCode] => 1
        [left] => stdClass Object (
                [value] =>
                [huruf] =>
                [count] => 314272
                [prefixCode] => 10
                [left] => stdClass Object  (
                        [value] =>
                        [huruf] =>
```

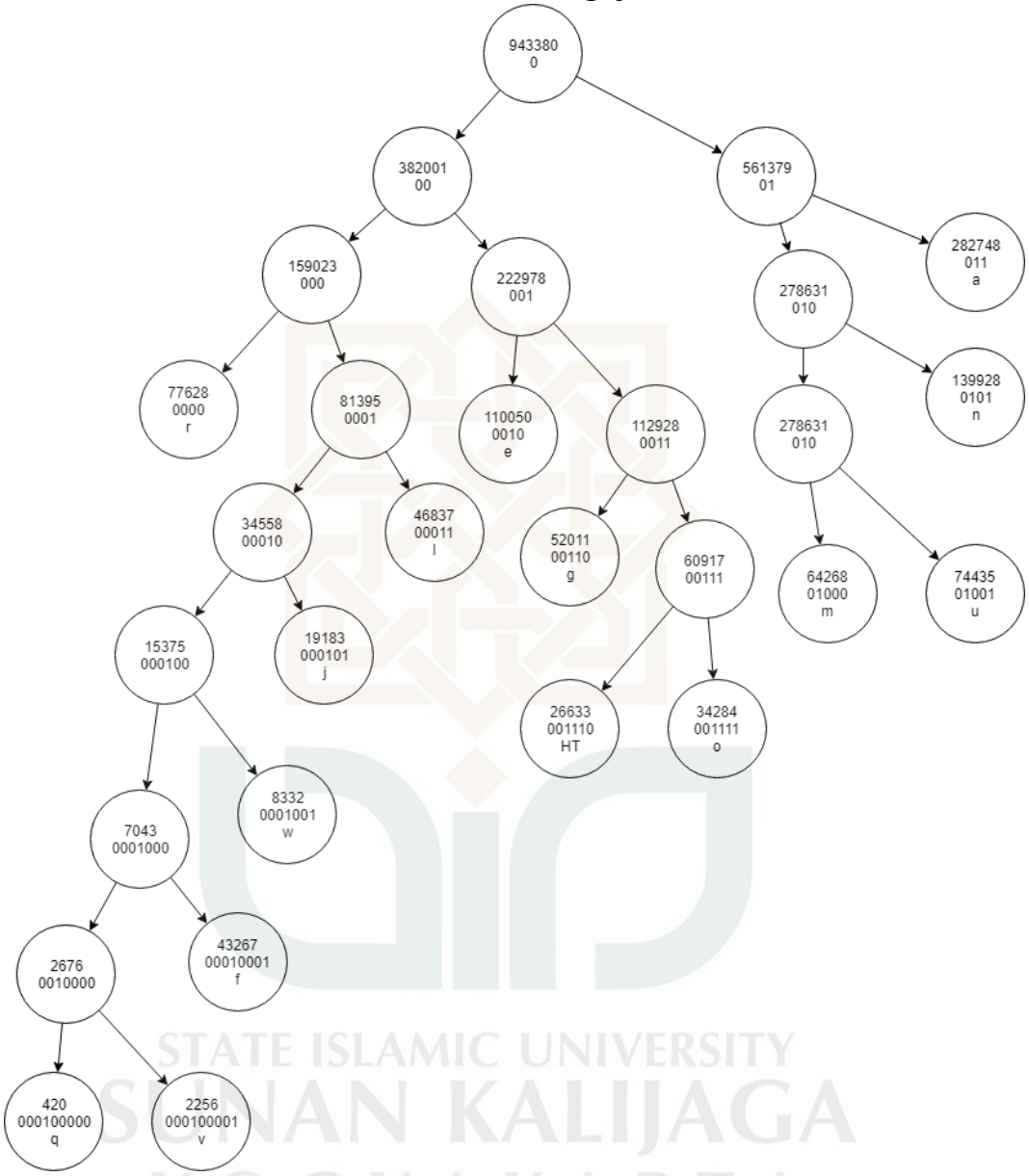
```
                                [count] => 146807
                                [prefixCode] => 100
                                [left] => stdClass Object (
                                        [value] => 115
                                        [huruf] => s
                                        [count] => 69409
                                        [prefixCode] => 1000
                                        [left] =>
                                        [right] =>
                                    )
                                [right] => stdClass Object  (
                                        [value] => 116
                                        [huruf] => t
                                        [count] => 77398
                                        [prefixCode] => 1001
                                        [left] =>
                                        [right] =>
                                    )
                            )
                        [right] => stdClass Object (
                                [value] =>
                                [huruf] =>
                                [count] => 167465
                                [prefixCode] => 101
                                [left] => stdClass Object (
                                        [value] => 107
                                        [huruf] => k
                                        [count] => 77863
                                        [prefixCode] => 1010
                                        [left] =>
                                        [right] =>
                                    )
                                [right] => stdClass Object (
                                        [value] =>
                                        [huruf] =>
                                        [count] => 89602
                                        [prefixCode] => 1011
                                        [left] => stdClass Object (
                                                [value] =>
                                                [huruf] =>
                                                [count] => 42603
                                                [prefixCode] => 10110
                                                [left] => stdClass Object (
                                                        [value] =>
                                                        [huruf] =>
                                                        [count] => 19136
                                                        [prefixCode] =>
101100
                                                        [left] => stdClass
Object (
                                                                [value] => 10
                                                                [huruf] =>
                                                                [count] =>
8745
                                                                [prefixCode]
=> 1011000
                                                                [left] =>
                                                                [right] =>
                                                            )
                                                        [right] => stdClass
Object (
                                                                [value] =>
                                                                [huruf] =>
```

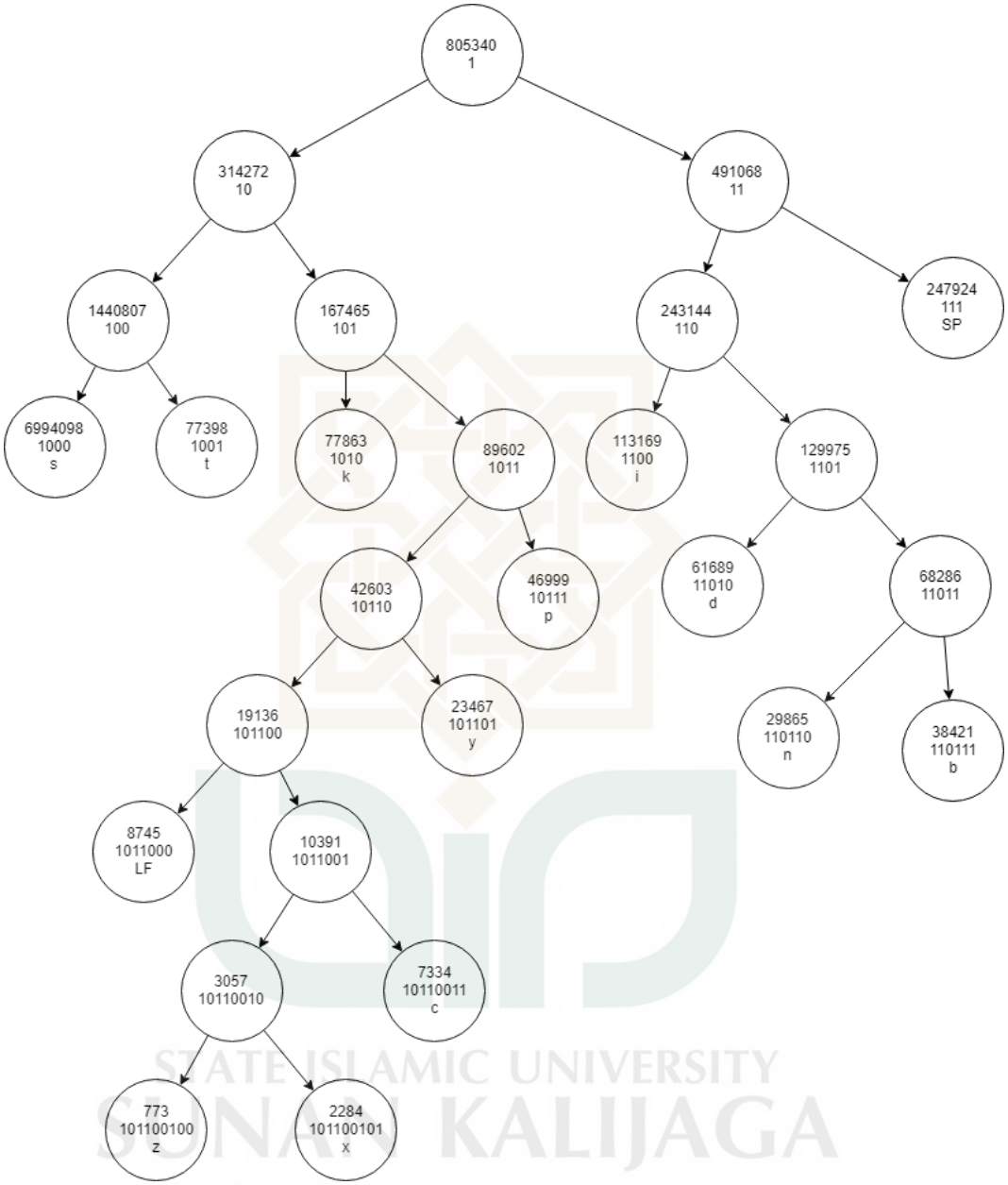
```
                                                                [count] =>
10391
                                                                [prefixCode]
=> 1011001
                                                                [left] =>
stdClass Object (
[value] =>
[huruf] =>
[count] => 3057
[prefixCode] => 10110010
[left] => stdClass Object (
[value] => 122
[huruf] => z
[count] => 773
[prefixCode] => 101100100
[left] =>
[right] =>
                                                                        )
[right] => stdClass Object (
[value] => 120
[huruf] => x
[count] => 2284
[prefixCode] => 101100101
[left] =>
[right] =>
                                                                    )
                                                                )
                                                                [right] =>
stdClass Object (
[value] => 99
[huruf] => c
[count] => 7334
[prefixCode] => 10110011
[left] =>
[right] =>
                                                                    )
                                                            )
                                                        )
                                                    [right] => stdClass Object (
```

```
                                                    [value] => 121
                                                    [huruf] => y
                                                    [count] => 23467
                                                    [prefixCode] =>
101101
                                                    [left] =>
                                                    [right] =>
                                                )
                                        )
                                    [right] => stdClass Object (
                                            [value] => 112
                                            [huruf] => p
                                            [count] => 46999
                                            [prefixCode] => 10111
                                            [left] =>
                                            [right] =>
                                        )
                                )
                            )
                        )
                    [right] => stdClass Object (
                            [value] =>
                            [huruf] =>
                            [count] => 491068
                            [prefixCode] => 11
                            [left] => stdClass Object (
                                    [value] =>
                                    [huruf] =>
                                    [count] => 243144
                                    [prefixCode] => 110
                                    [left] => stdClass Object (
                                            [value] => 105
                                            [huruf] => i
                                            [count] => 113169
                                            [prefixCode] => 1100
                                            [left] =>
                                            [right] =>
                                        )
                                    [right] => stdClass Object (
                                            [value] =>
                                            [huruf] =>
                                            [count] => 129975
                                            [prefixCode] => 1101
                                            [left] => stdClass Object (
                                                    [value] => 100
                                                    [huruf] => d
                                                    [count] => 61689
                                                    [prefixCode] => 11010
                                                    [left] =>
                                                    [right] =>
                                                )
                                            [right] => stdClass Object (
                                                    [value] =>
                                                    [huruf] =>
                                                    [count] => 68286
                                                    [prefixCode] => 11011
                                                    [left] => stdClass Object  (
                                                            [value] => 104
                                                            [huruf] => h
                                                            [count] => 29865
                                                            [prefixCode] =>
110110
                                                            [left] =>
```

```
                                                    [right] =>
                                                )
                                            [right] => stdClass Object (
                                                    [value] => 98
                                                    [huruf] => b
                                                    [count] => 38421
                                                    [prefixCode] =>
110111
                                                    [left] =>
                                                    [right] =>
                                                )
                                        )
                                    )
                                )
                            [right] => stdClass Object (
                                    [value] => 32
                                    [huruf] =>
                                    [count] => 247924
                                    [prefixCode] => 111
                                    [left] =>
                                    [right] =>
                                )
                        )
                )
        )
)
```

2. Visualisasi Huffman Tree Pengujian





3. Screenshoot Pengujian Rasio

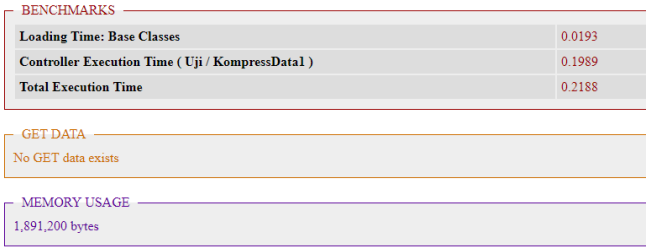
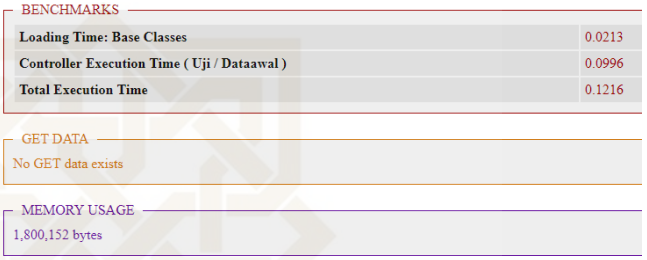
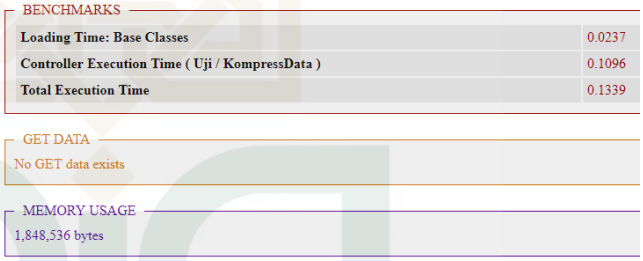
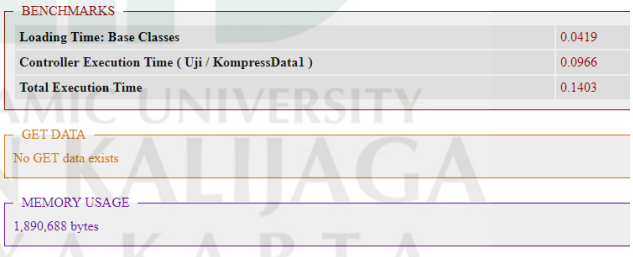
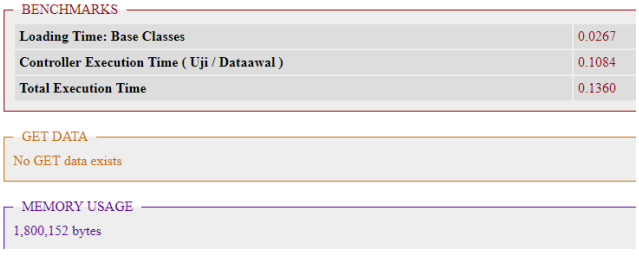
Data Awal	Data Hasil kompresi

id_baru	data_baru
1	[BLOB - 9 B]
2	[BLOB - 2.3 KiB]
3	[BLOB - 1.3 KiB]
4	[BLOB - 997 B]
5	[BLOB - 2.2 KiB]
6	[BLOB - 1.8 KiB]
7	[BLOB - 1.4 KiB]
8	[BLOB - 1.5 KiB]
9	[BLOB - 1.6 KiB]
10	[BLOB - 2.1 KiB]

id_huffman1	data_huffman1
1	[BLOB - 10 B]
2	[BLOB - 1.2 KiB]
3	[BLOB - 672 B]
4	[BLOB - 511 B]
5	[BLOB - 1.1 KiB]
6	[BLOB - 951 B]
7	[BLOB - 739 B]
8	[BLOB - 797 B]
9	[BLOB - 850 B]
10	[BLOB - 1.1 KiB]

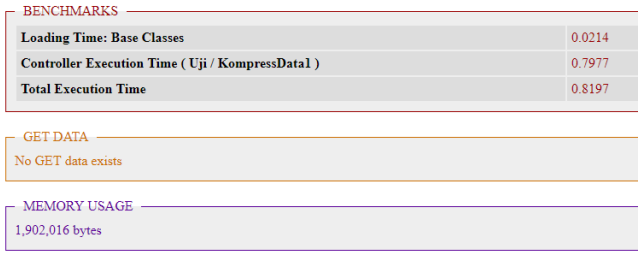
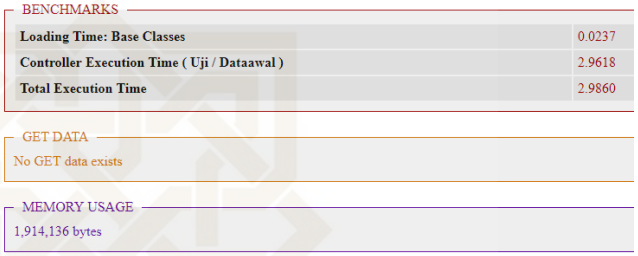
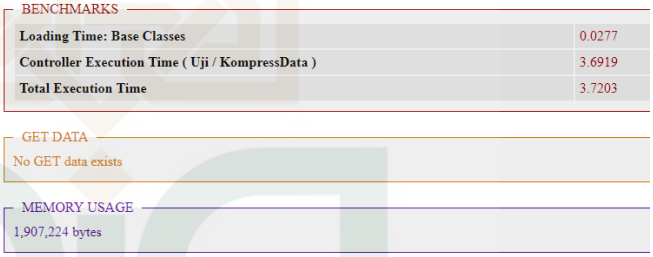
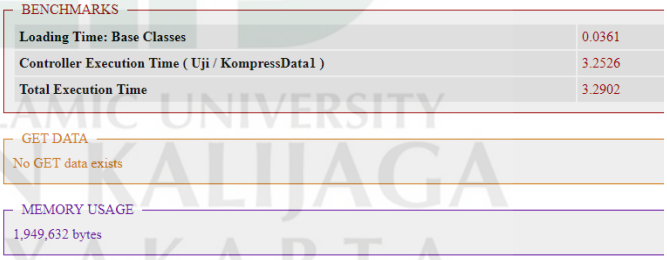
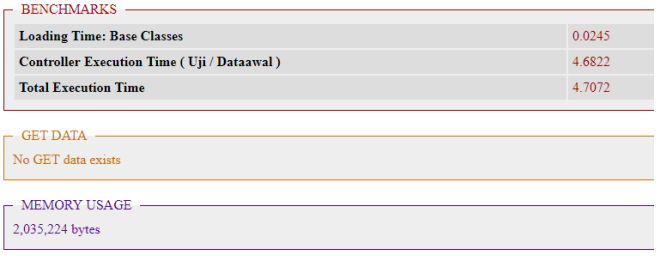
4. Screenshoot Pengujian Performa Save

Percobaan ke-		Hasil
1 **Indeks 1**	Tanpa Algoritma Huffman	BENCHMARKS Loading Time: Base Classes 0.0230 Controller Execution Time (Uji / Dataawal) 0.1534 Total Execution Time 0.1771 GET DATA No GET data exists MEMORY USAGE 1,800,408 bytes
	Dengan Algoritma Huffman 2 Pohon	BENCHMARKS Loading Time: Base Classes 0.0214 Controller Execution Time (Uji / KompressData) 0.1902 Total Execution Time 0.2126 GET DATA No GET data exists MEMORY USAGE 1,848,920 bytes

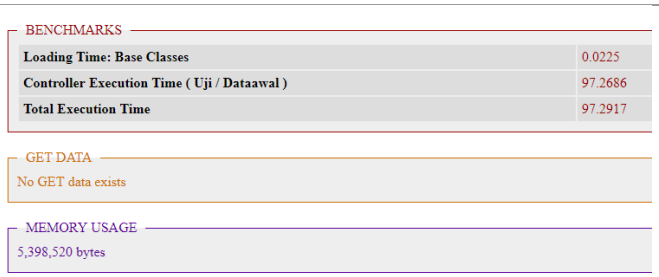
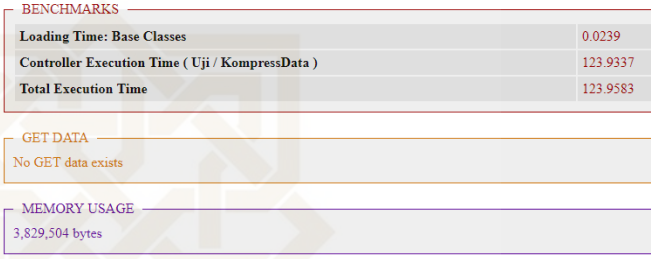
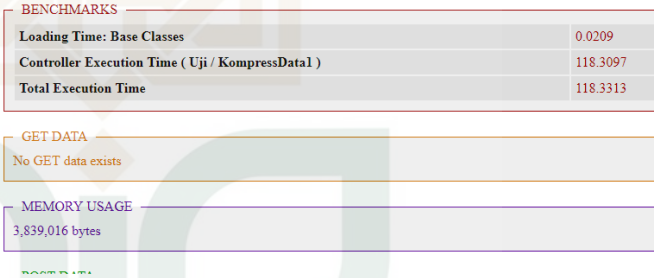
	Dengan Algoritma Huffman 1 Pohon	
2 Indeks 10	Tanpa Algoritma Huffman	
	Dengan Algoritma Huffman 2 Pohon	
	Dengan Algoritma Huffman 1 Pohon	
3 Indeks 100	Tanpa Algoritma Huffman	

	Dengan Algoritma Huffman 2 Pohon	BENCHMARKS Loading Time: Base Classes 0.0250 Controller Execution Time (Uji / KompressData) 0.1128 Total Execution Time 0.1384 GET DATA No GET data exists MEMORY USAGE 1,848,536 bytes
	Dengan Algoritma Huffman 1 Pohon	BENCHMARKS Loading Time: Base Classes 0.0213 Controller Execution Time (Uji / KompressData1) 0.1117 Total Execution Time 0.1337 GET DATA No GET data exists MEMORY USAGE 1,890,688 bytes
4 Indeks 1230	Tanpa Algoritma Huffman	BENCHMARKS Loading Time: Base Classes 0.0236 Controller Execution Time (Uji / Dataawal) 0.1100 Total Execution Time 0.1345 GET DATA No GET data exists MEMORY USAGE 1,799,512 bytes
	Dengan Algoritma Huffman 2 Pohon	BENCHMARKS Loading Time: Base Classes 0.0268 Controller Execution Time (Uji / KompressData) 0.0737 Total Execution Time 0.1012 GET DATA No GET data exists MEMORY USAGE 1,842,008 bytes
	Dengan Algoritma Huffman 1 Pohon	BENCHMARKS Loading Time: Base Classes 0.0216 Controller Execution Time (Uji / KompressData1) 0.1196 Total Execution Time 0.1419 GET DATA No GET data exists MEMORY USAGE 1,884,160 bytes

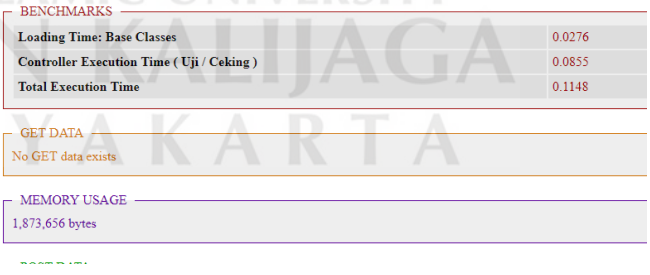
5 **Indeks 1230** **Percobaan ke 2 kali**	Tanpa Algoritma Huffman	BENCHMARKS Loading Time: Base Classes 0.0214 Controller Execution Time (Uji / Dataawal) 0.0987 Total Execution Time 0.1208 GET DATA No GET data exists MEMORY USAGE 1,799,512 bytes
	Dengan Algoritma Huffman 2 Pohon	BENCHMARKS Loading Time: Base Classes 0.0251 Controller Execution Time (Uji / KompressData) 0.0751 Total Execution Time 0.1015 GET DATA No GET data exists MEMORY USAGE 1,842,008 bytes
	Dengan Algoritma Huffman 1 Pohon	BENCHMARKS Loading Time: Base Classes 0.0206 Controller Execution Time (Uji / KompressData1) 0.1038 Total Execution Time 0.1251 GET DATA No GET data exists MEMORY USAGE 1,884,160 bytes
6 **10 data** **Indeks 2200 - 2210**	Tanpa Algoritma Huffman	BENCHMARKS Loading Time: Base Classes 0.0212 Controller Execution Time (Uji / Dataawal) 0.7416 Total Execution Time 0.7635 GET DATA No GET data exists MEMORY USAGE 1,820,568 bytes
	Dengan Algoritma Huffman 2 Pohon	BENCHMARKS Loading Time: Base Classes 0.0236 Controller Execution Time (Uji / KompressData) 0.8673 Total Execution Time 0.8915 GET DATA No GET data exists MEMORY USAGE 1,859,736 bytes

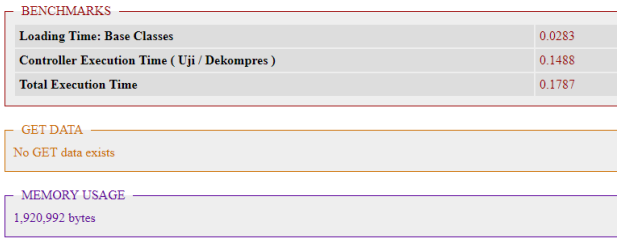
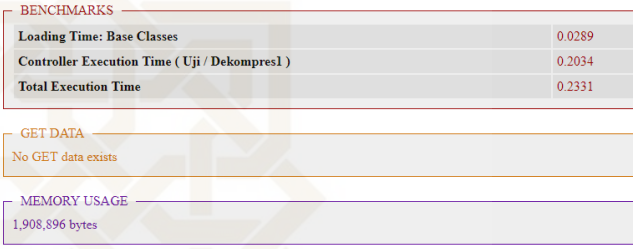
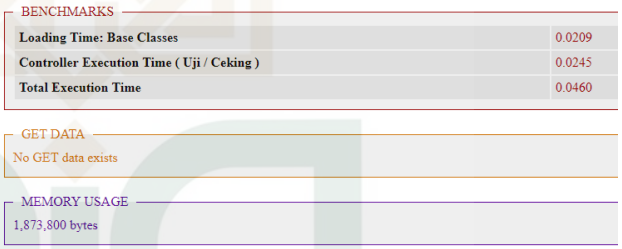
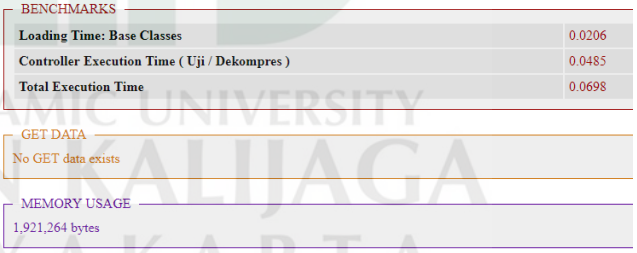
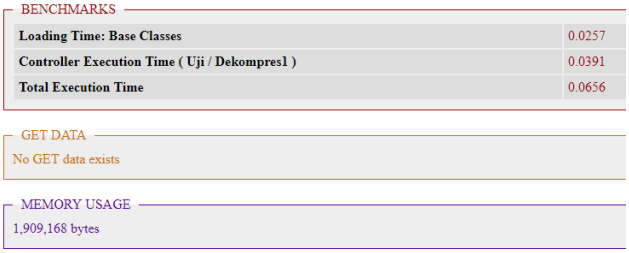
	Dengan Algoritma Huffman 1 Pohon	
7 50 data Indeks 2200-2250	Tanpa Algoritma Huffman	
	Dengan Algoritma Huffman 2 Pohon	
	Dengan Algoritma Huffman 1 Pohon	
8 100 data	Tanpa Algoritma Huffman	

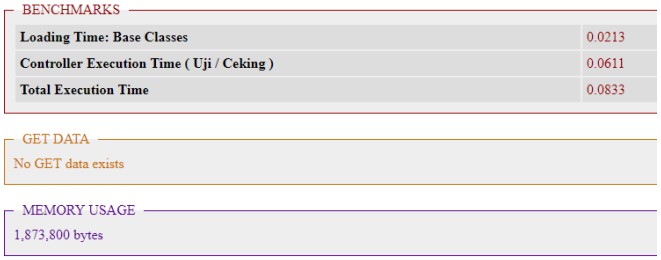
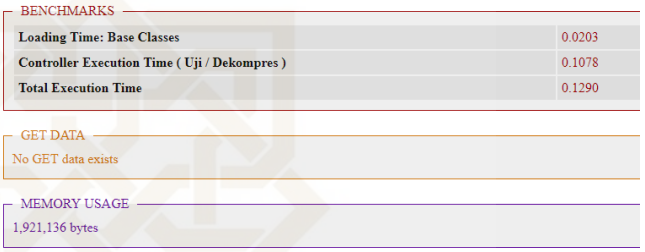
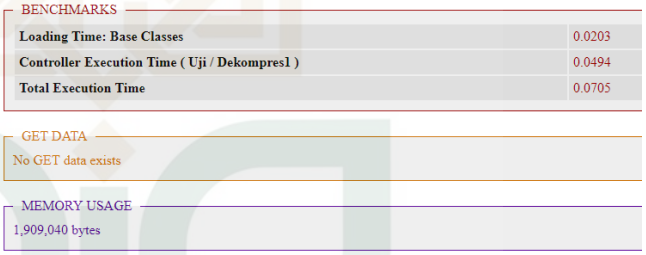
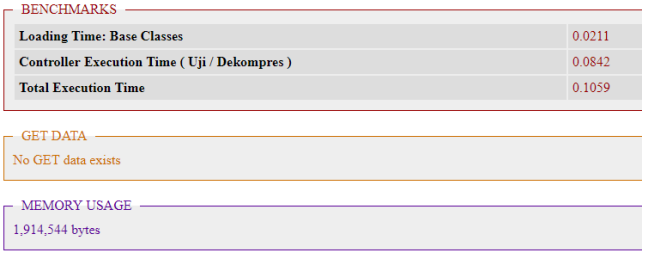
	Dengan Algoritma Huffman 2 Pohon	BENCHMARKS Loading Time: Base Classes 0.0256 Controller Execution Time (Uji / KompressData) 6.1379 Total Execution Time 6.1640 GET DATA No GET data exists MEMORY USAGE 1,973,400 bytes
	Dengan Algoritma Huffman 1 Pohon	BENCHMARKS Loading Time: Base Classes 0.0213 Controller Execution Time (Uji / KompressData1) 5.2073 Total Execution Time 5.2292 GET DATA No GET data exists MEMORY USAGE 2,023,360 bytes
9 **500 data**	Tanpa Algoritma Huffman	BENCHMARKS Loading Time: Base Classes 0.0245 Controller Execution Time (Uji / Dataawal) 35.7593 Total Execution Time 35.7843 GET DATA No GET data exists MEMORY USAGE 2,949,496 bytes
	Dengan Algoritma Huffman 2 Pohon	BENCHMARKS Loading Time: Base Classes 0.0261 Controller Execution Time (Uji / KompressData) 40.7502 Total Execution Time 40.7771 GET DATA No GET data exists MEMORY USAGE 2,491,640 bytes
	Dengan Algoritma Huffman 1 Pohon	BENCHMARKS Loading Time: Base Classes 0.0209 Controller Execution Time (Uji / KompressData1) 34.9816 Total Execution Time 35.0032 GET DATA No GET data exists MEMORY USAGE 2,542,368 bytes

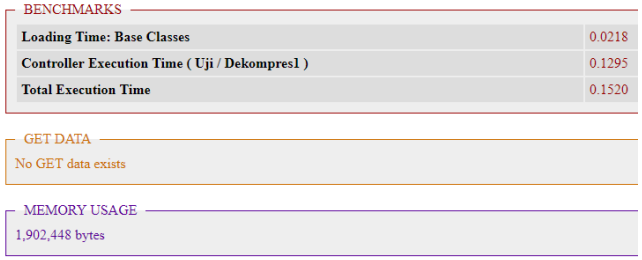
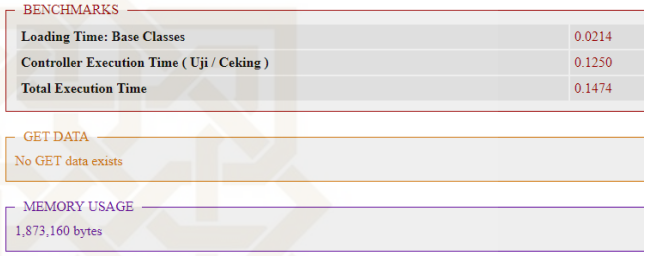
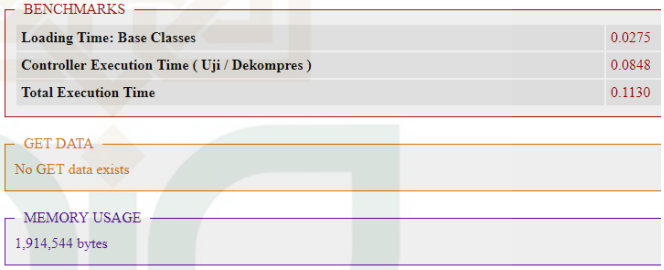
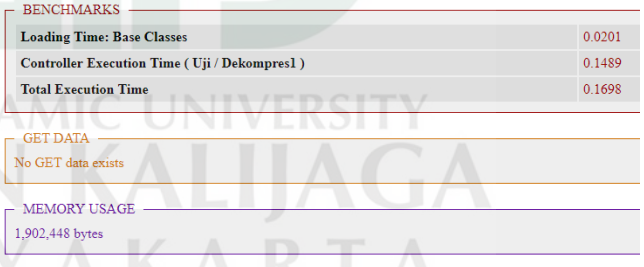
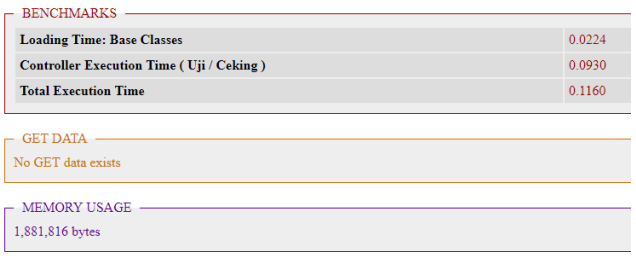
10 **1500 data**	Tanpa Algoritma Huffman	
	Dengan Algoritma Huffman 2 Pohon	
	Dengan Algoritma Huffman 1 Pohon	

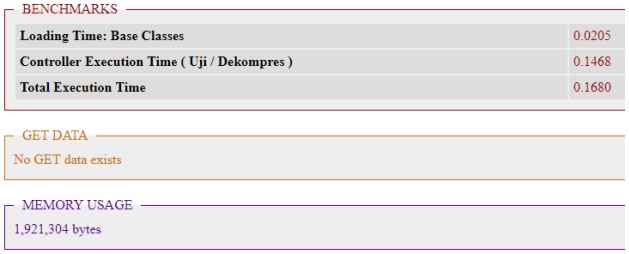
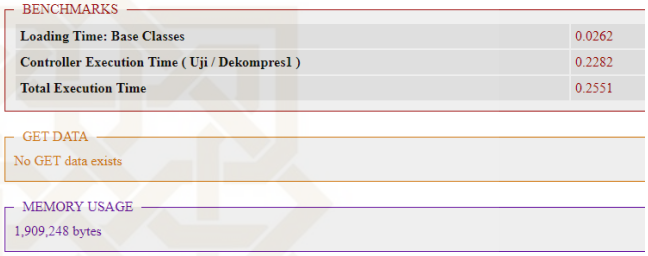
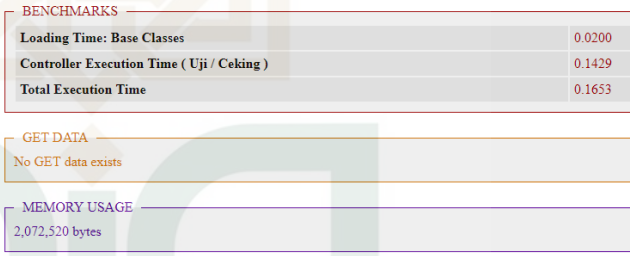
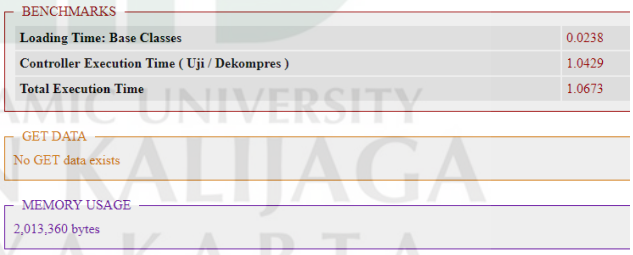
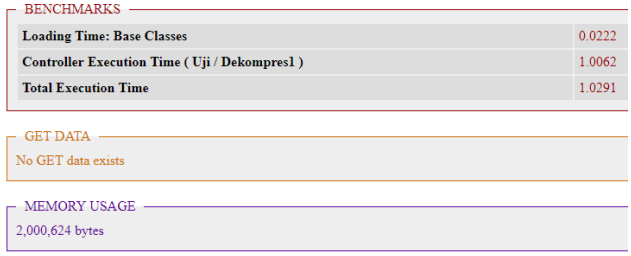
5. Screenhoot Pengujian Performa Menampilkan Data

Percobaan ke-		Hasil
1 **Indeks 10**	Tanpa Algoritma Huffman	

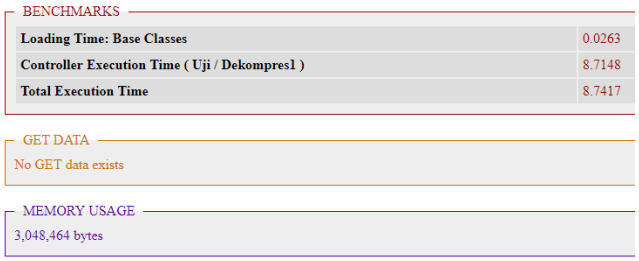
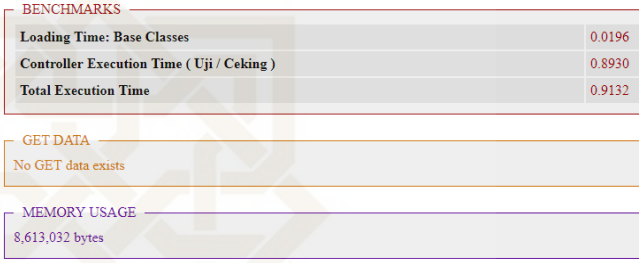
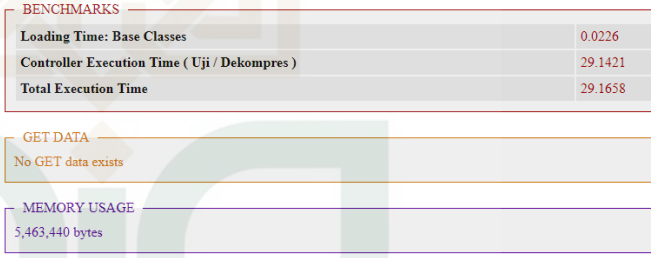
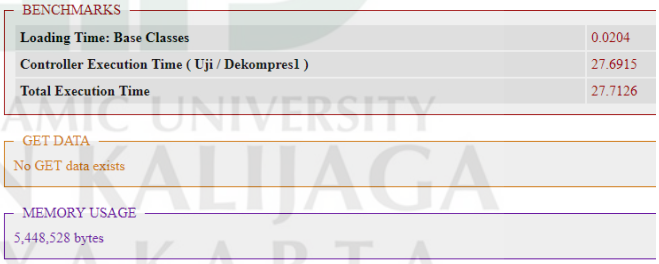
	Dengan Algoritma Huffman 2 Pohon	
	Dengan Algoritma Huffman 1 Pohon	
2 Indeks 11	Tanpa Algoritma Huffman	
	Dengan Algoritma Huffman 2 Pohon	
	Dengan Algoritma Huffman 1 Pohon	

3 **Indeks 100**	Tanpa Algoritma Huffman	
	Dengan Algoritma Huffman 2 Pohon	
	Dengan Algoritma Huffman 1 Pohon	
4 **Indeks 1230**	Tanpa Algoritma Huffman	
	Dengan Algoritma Huffman 2 Pohon	

	Dengan Algoritma Huffman 1 Pohon	
5 Indeks 1230 Percobaan ke 2 kali	Tanpa Algoritma Huffman	
	Dengan Algoritma Huffman 2 Pohon	
	Dengan Algoritma Huffman 1 Pohon	
6 10 data Indeks 1230 - 1240	Tanpa Algoritma Huffman	

	Dengan Algoritma Huffman 2 Pohon	
	Dengan Algoritma Huffman 1 Pohon	
7 50 data Indeks 2200-2250	Tanpa Algoritma Huffman	
	Dengan Algoritma Huffman 2 Pohon	
	Dengan Algoritma Huffman 1 Pohon	

8 **100 data**	Tanpa Algoritma Huffman	BENCHMARKS Loading Time: Base Classes 0.0199 Controller Execution Time (Uji / Ceking) 0.1528 Total Execution Time 0.1734 GET DATA No GET data exists MEMORY USAGE 2,305,640 bytes
	Dengan Algoritma Huffman 2 Pohon	BENCHMARKS Loading Time: Base Classes 0.0222 Controller Execution Time (Uji / Dekompres) 1.9905 Total Execution Time 2.0135 GET DATA No GET data exists MEMORY USAGE 2,138,064 bytes
	Dengan Algoritma Huffman 1 Pohon	BENCHMARKS Loading Time: Base Classes 0.0194 Controller Execution Time (Uji / Dekompres1) 1.8826 Total Execution Time 1.9027 GET DATA No GET data exists MEMORY USAGE 2,125,072 bytes
9 **500 data**	Tanpa Algoritma Huffman	BENCHMARKS Loading Time: Base Classes 0.0277 Controller Execution Time (Uji / Ceking) 0.2218 Total Execution Time 0.2506 GET DATA No GET data exists MEMORY USAGE 4,027,816 bytes
	Dengan Algoritma Huffman 2 Pohon	BENCHMARKS Loading Time: Base Classes 0.0224 Controller Execution Time (Uji / Dekompres) 9.4823 Total Execution Time 9.5053 GET DATA No GET data exists MEMORY USAGE 3,060,944 bytes

	Dengan Algoritma Huffman 1 Pohon	
10 1500 data	Tanpa Algoritma Huffman	
	Dengan Algoritma Huffman 2 Pohon	
	Dengan Algoritma Huffman 1 Pohon	

LAMPIRAN C : Tabel ASCII

Karakter	Nilai Unicode (heksadesimal)	Nilai ANSI ASCII (desimal)	Keterangan
NUL	0000	0	Null (tidak tampak)
SOH	0001	1	Start of heading (tidak tampak)
STX	0002	2	Start of text (tidak tampak)
ETX	0003	3	End of text (tidak tampak)
EOT	0004	4	End of transmission (tidak tampak)
ENQ	0005	5	Enquiry (tidak tampak)
ACK	0006	6	Acknowledge (tidak tampak)
BEL	0007	7	Bell (tidak tampak)

BS	0008	8	Menghapus satu karakter di belakang kursor (Backspace)
HT	0009	9	Horizontal tabulation
LF	000A	10	Pergantian baris (Line feed)
VT	000B	11	Tabulasi vertikal
FF	000C	12	Pergantian baris (Form feed)
CR	000D	13	Pergantian baris (carriage return)
SO	000E	14	Shift out (tidak tampak)
SI	000F	15	Shift in (tidak tampak)
DLE	0010	16	Data link escape (tidak tampak)
DC1	0011	17	Device control 1 (tidak tampak)
DC2	0012	18	Device control 2 (tidak tampak)
DC3	0013	19	Device control 3 (tidak tampak)
DC4	0014	20	Device control 4 (tidak tampak)
NAK	0015	21	Negative acknowledge (tidak tampak)
SYN	0016	22	Synchronous idle (tidak tampak)
ETB	0017	23	End of transmission block (tidak tampak)
CAN	0018	24	Cancel (tidak tampak)
EM	0019	25	End of medium (tidak tampak)
SUB	001A	26	Substitute (tidak tampak)
ESC	001B	27	Escape (tidak tampak)
FS	001C	28	File separator
GS	001D	29	Group separator
RS	001E	30	Record separator
US	001F	31	Unit separator
SP	0020	32	Spasi
!	0021	33	Tanda seru (exclamation)
"	0022	34	Tanda kutip dua
#	0023	35	Tanda pagar (kres)
$	0024	36	Tanda mata uang dolar
%	0025	37	Tanda persen
&	0026	38	Karakter ampersand (&)
'	0027	39	Karakter Apostrof
(	0028	40	Tanda kurung buka
)	0029	41	Tanda kurung tutup
*****	002A	42	Karakter asterisk (bintang)
+	002B	43	Tanda tambah (plus)
,	002C	44	Karakter koma
-	002D	45	Karakter hyphen (strip)
.	002E	46	Tanda titik
/	002F	47	Garis miring (*slash*)
0	0030	48	Angka nol
1	0031	49	Angka satu
2	0032	50	Angka dua
3	0033	51	Angka tiga
4	0034	52	Angka empat
5	0035	53	Angka lima
6	0036	54	Angka enam
7	0037	55	Angka tujuh
8	0038	56	Angka delapan
9	0039	57	Angka sembilan
:	003A	58	Tanda titik dua
;	003B	59	Tanda titik koma
<	003C	60	Tanda lebih kecil
=	003D	61	Tanda sama dengan
>	003E	62	Tanda lebih besar
?	003F	63	Tanda tanya
@	0040	64	A keong (@)
A	0041	65	Huruf latin A kapital
B	0042	66	Huruf latin B kapital
C	0043	67	Huruf latin C kapital

D	0044	68	Huruf latin D kapital
E	0045	69	Huruf latin E kapital
F	0046	70	Huruf latin F kapital
G	0047	71	Huruf latin G kapital
H	0048	72	Huruf latin H kapital
I	0049	73	Huruf latin I kapital
J	004A	74	Huruf latin J kapital
K	004B	75	Huruf latin K kapital
L	004C	76	Huruf latin L kapital
M	004D	77	Huruf latin M kapital
N	004E	78	Huruf latin N kapital
O	004F	79	Huruf latin O kapital
P	0050	80	Huruf latin P kapital
Q	0051	81	Huruf latin Q kapital
R	0052	82	Huruf latin R kapital
S	0053	83	Huruf latin S kapital
T	0054	84	Huruf latin T kapital
U	0055	85	Huruf latin U kapital
V	0056	86	Huruf latin V kapital
W	0057	87	Huruf latin W kapital
X	0058	88	Huruf latin X kapital
Y	0059	89	Huruf latin Y kapital
Z	005A	90	Huruf latin Z kapital
[	005B	91	Kurung siku kiri
****	005C	92	Garis miring terbalik (*backslash*)
]	005D	93	Kurung sikur kanan
^	005E	94	Tanda pangkat
_	005F	95	Garis bawah (underscore)
`	0060	96	Tanda petik satu
a	0061	97	Huruf latin a kecil
b	0062	98	Huruf latin b kecil
c	0063	99	Huruf latin c kecil
d	0064	100	Huruf latin d kecil
e	0065	101	Huruf latin e kecil
f	0066	102	Huruf latin f kecil
g	0067	103	Huruf latin g kecil
h	0068	104	Huruf latin h kecil
i	0069	105	Huruf latin i kecil
j	006A	106	Huruf latin j kecil
k	006B	107	Huruf latin k kecil
l	006C	108	Huruf latin l kecil
m	006D	109	Huruf latin m kecil
n	006E	110	Huruf latin n kecil
o	006F	111	Huruf latin o kecil
p	0070	112	Huruf latin p kecil
q	0071	113	Huruf latin q kecil
r	0072	114	Huruf latin r kecil
s	0073	115	Huruf latin s kecil
t	0074	116	Huruf latin t kecil
u	0075	117	Huruf latin u kecil
v	0076	118	Huruf latin v kecil
w	0077	119	Huruf latin w kecil
x	0078	120	Huruf latin x kecil
y	0079	121	Huruf latin y kecil
z	007A	122	Huruf latin z kecil
{	007B	123	Kurung kurawal buka
¦	007C	124	Garis vertikal (pipa)
}	007D	125	Kurung kurawal tutup
~	007E	126	Karakter gelombang (tilde)
DEL	007F	127	Delete
	0080	128	Dicadangkan
	0081	129	Dicadangkan

	0082	130	Dicadangkan
	0083	131	Dicadangkan
IND	0084	132	Index
NEL	0085	133	Next line
SSA	0086	134	Start of selected area
ESA	0087	135	End of selected area
	0088	136	Character tabulation set
	0089	137	Character tabulation with justification
	008A	138	Line tabulation set
PLD	008B	139	Partial line down
PLU	008C	140	Partial line up
	008D	141	Reverse line feed
SS2	008E	142	Single shift two
SS3	008F	143	Single shift three
DCS	0090	144	Device control string
PU1	0091	145	Private use one
PU2	0092	146	Private use two
STS	0093	147	Set transmit state
CCH	0094	148	Cancel character
MW	0095	149	Message waiting
	0096	150	Start of guarded area
	0097	151	End of guarded area
	0098	152	Start of string
	0099	153	Dicadangkan
	009A	154	Single character introducer
CSI	009B	155	Control sequence introducer
ST	009C	156	String terminator
OSC	009D	157	Operating system command
PM	009E	158	Privacy message
APC	009F	158	Application program command
	00A0	160	Spasi yang bukan pemisah kata
¡	00A1	161	Tanda seru terbalik
¢	00A2	162	Tanda sen (Cent)
£	00A3	163	Tanda Poundsterling
¤	00A4	164	Tanda mata uang (*Currency*)
¥	00A5	165	Tanda Yen
¦	00A6	166	Garis tegak putus-putus (*broken bar*)
§	00A7	167	Section sign
¨	00A8	168	Diaeresis
©	00A9	169	Tanda hak cipta (Copyright)
ª	00AA	170	Feminine ordinal indicator
«	00AB	171	Left-pointing double angle quotation mark
¬	00AC	172	Not sign
	00AD	173	Tanda strip (*hyphen*)
®	00AE	174	Tanda merk terdaftar
¯	00AF	175	Macron
°	00B0	176	Tanda derajat
±	00B1	177	Tanda kurang lebih (plus-minus)
²	00B2	178	Tanda kuadrat (pangkat dua)
³	00B3	179	Tanda kubik (pangkat tiga)
´	00B4	180	Acute accent
µ	00B5	181	Micro sign
¶	00B6	182	Pilcrow sign
·	00B7	183	Middle dot

LAMPIRAN D : Kode Sumber

File : Huffman.php

```
<?php
include "Sampel.php";
include "Node.php";
include "Tree.php";
include "Kompres.php";
include "Dekompres.php";
class Huffman{
    protected $kompres;
    protected $dekompres;
    protected $tree;
    protected $stringCount;
    protected $treeResult;
    protected $treeArrayResult;
    protected $direktory;
    protected $fileNameTree = "/Pohon.json";
    protected $fileNameTreeArray = "/Code.json";
    public function __construct() {
        $this->tree = new Tree(null,null);
        $this->stringCount = new Sampel();
        $this->direktory = dirname(__FILE__);
    }
    //fungsi untuk kompresi data menjadi kode prefix
    public function kompres($var)
    {
        $this->kompres = new Kompres();
       $this->kompres->setData($var);
       return $this->toASCIILagi($this->kompres->getKompres());
    }
    //get dekompres data
    public function dekompres($var)
    {
         $this->dekompres = new Dekompres();
        $this->dekompres->setData($this->toBinerLagi($var));
        return $this->dekompres->getString();
    }
    //untuk presentasi saja
    public function pptKompres($var)
    {
       $this->kompres = new Kompres();
       $this->kompres->setData($var);
       $hasil1 = $this->kompres->getKompres();
       $hasil2 = $this->toASCIILagi($hasil1);
       return array($hasil1,$hasil2);
    }
    public function pptDekompres($var)
    {
        $this->dekompres = new Dekompres();
        $hasil1 = $this->toBinerLagi($var);
        $this->dekompres->setData($hasil1);
        $hasil2 = $this->dekompres->getString();
       return array($hasil1,$hasil2);
    }
    //training data jadi pohon huffman
    public function buildTree($var)
    {
        if ($this->cekTreehuffman() == true) {
            $this->stringCount->setData($var);
            $jumChar = $this->stringCount->getJumlahKarakter();
            $this->getTree($jumChar[0],"0");
            $treeHuffman[] = $this->treeResult;
```

```
            $treeArrayHuffman[] = $this->treeArrayResult;
            $this->getTree($jumChar[1],"1");
            $treeHuffman[] = $this->treeResult;
            $treeArrayHuffman[] = $this->treeArrayResult;
            $this->saveTree($treeHuffman,$treeArrayHuffman);
            return true;
        }else{
            return false;
        }
    }
    //fungsi membentuk tree
    private function getTree($node, $code)
    {
        $this->tree->setData($node,$code);
        $this->tree->build();
        $this->tree->buildCode();
        $this->treeResult = $this->tree->getTree();
        $this->treeArrayResult = $this->tree->getTreeArray();
    }
    //save  tree
    private function saveTree($tree, $treeArray)
    {
        file_put_contents($this->direktory.$this-
>fileNameTreeArray,json_encode($treeArray));
        file_put_contents($this->direktory.$this-
>fileNameTree,json_encode($tree));
    }
    //cek tree huffmannya udah ada belum
    public function cekTreehuffman()
    {
        if (empty(json_decode(file_get_contents($this->direktory.$this-
>fileNameTree))) && empty(json_decode(file_get_contents($this-
>direktory.$this->fileNameTreeArray)))) {
            return false;
        }else{
            return true;
        }
    }
    //clean tree huffman nya
    public function cleanTreeHuffman()
    {
        $this->saveTree(null,null);
        return true;
    }
    //fungsi ubah lagi ke ascii
    public function toASCIILagi($hasilKompresi)
    {
        $my_byte_array = str_split($hasilKompresi);
        $text = "";
        for ($i=0; $i < count($my_byte_array) ; $i+=8) {
            $bit = '';
            for ($j=$i; $j < $i+8; $j++) {
                $bit .= $my_byte_array[$j];
            }
            if (count(str_split($bit)) == 8) {
                $text .= chr(bindec($bit));
            }else{
                $text .= $bit.count(str_split($bit));
            }
        }
        if (count($my_byte_array)%8 == 0) {
            $text .= "0";
        }
```

```
        return $text;
    }
    //dari ascii ke biner lagih
    public function toBinerLagi($text)
    {
        $my_text = str_split($text);
        $bin = "";
        $sisa = $my_text[count($my_text)-1];
        unset($my_text[count($my_text)-1]);
        for($i=0; $i < count($my_text); $i++ ) {
            if ($i < count($my_text)-$sisa) {
                $bin .= sprintf( "%08d", decbin(ord($my_text[$i])));
            }else{
                $bin .= $my_text[$i];
            }
        }
        return $bin;
    }
    public function getDir()
    {
        return $this->direktory;
    }
    public function getTreeHuffman()
    {
        return json_decode(file_get_contents($this->direktory.$this-
>fileNameTree));
    }
}
?>
```

File : Node.php

```
<?php
class Node
    {
        var $value;
        var $huruf;
        var $count;
        var $prefixCode;
        var $left;
        var $right;
        public function __construct($jml, $id) {
            $this->count = $jml;
            $this->value = $id;
            // new nodes are leaf nodes
            $this->left = null;
            $this->right = null;
            $this->prefixCode = null;
        }
        public function setLeaf($kiri, $kanan)
        {
            $this->left = $kiri;
            $this->right = $kanan;
        }
        public function setCode($code)
        {
            $this->prefixCode = $code;
        }
        public function setHuruf($huruf)
        {
            $this->huruf = $huruf;
        }
    }
```

```
?>
```

File : Sampel.php

```
<?php
    class Sampel
    {
        var $data;
        function __construct()
        {
          //parent::__construct();
        }
        public function setData(Array $par)
        {
          //$this->data = preg_replace('/[^A-Za-z0-9\s]/', '', $par);
          $data = "";
          foreach ($par as $key) {
              $data .= preg_replace('/[^A-Za-z\s]/', '',
trim($key,"\x0B\n\0\r\t"));
          }
          $this->data = $data;
          //$this->data = preg_replace('/[^A-Za-z\s]/', '', $par);
          //$this->data = $par;
        }
        public function getData(){
            return $this->data;
        }
        public function getJumlahKarakter()
        {
            $temp = strtolower($this->data);
            $temp1 = count_chars(strtolower($temp),1);
            arsort($temp1);
            $hasil = $this->bagi($temp1);
            return $hasil;
            //return $temp1;
        }
        private function bagi($data)
        {
            $index = 0;
            $satu;
            $dua;
            foreach ($data as $key => $value) {
                if ($index%2 == 0) {
                    $node = new Node($value,$key);
                    $node->setHuruf(chr($key));
                    $satu[] =  $node;
                }else{
                    $node = new Node($value,$key);
                    $node->setHuruf(chr($key));
                    $dua[] =  $node;
                }
                $index++;
            }
            $hasil = array(0 => $satu, 1 => $dua);
            return $hasil;
        }
    }
?>
```

File : Tree.php

```
<?php
```

```
class Tree
{
    protected $huffmanTree;
    protected $data;
    protected $preCode;
    protected $root;
    protected $treeArray;
    public function __construct($item,$code)
    {
        $this->huffmanTree = null;
        $this->data = $item;
        $this->preCode = $code;
    }
    public function setData($item,$code)
    {
        $this->huffmanTree = null;
        $this->data = $item;
        $this->preCode = $code;
    }
    public function isEmpty()
    {
        return $this->root === null;
    }
    public function sorting()
    {
        usort($this->data, function($a, $b)
        {
            return $a->count <=> $b->count;
        });
    }
    public function build()
    {
        $indexSumNode = 0;
        while (count($this->data) > 1)
        {
            $indexSumNode++;
            $this->sorting();
            $pertama = $this->data[0];
            $kedua = $this->data[1];
            unset($this->data[0]);
            unset($this->data[1]);
            $count = $pertama->count + $kedua->count;
            $newNode = new Node($count,null);
            if ($pertama->count > $kedua->count)
            {
                $newNode->setLeaf($kedua,$pertama);
            }else
            {
                $newNode->setLeaf($pertama,$kedua);
            }
            array_push($this->data,$newNode);
            $this->addInHasil($pertama);
            $this->addInHasil($kedua);
            $this->huffmanTree[] = $newNode;
            $this->root = $newNode;
        }
        return $this->huffmanTree;
    }
    public function addInHasil($array)
    {
            if (in_array($array,$this->huffmanTree))
            {
                echo "data is exist";
```

```
                }else{
                    $this->huffmanTree[] = $array;
                }
        }
        public function buildCode()
        {
            $this->addPrefixCode($this->root,$this->preCode);
            $this->toArray($this->root);
        }
        public function addPrefixCode(Node $node, $prefix)
        {
            $node->setCode($prefix);
            if ($node->left) {
                $prefixLeft = $prefix."0";
                $this->addPrefixCode($node->left,$prefixLeft);
            }
             if ($node->right) {
                $prefixRight = $prefix."1";
                $this->addPrefixCode($node->right,$prefixRight);
            }
        }
        public function getTree()
        {
            return $this->huffmanTree[count($this->huffmanTree)-1];
        }
        public function toArray(Node $node)
        {
            $this->treeArray[] = array("value" => $node->value,"huruf" =>
$node->huruf ,"prefixCode" => $node->prefixCode);
            if ($node->left) {
                $this->toArray($node->left);
            }
            if ($node->right) {
                $this->toArray($node->right);
            }
        }
        public function getTreeArray()
        {
            return $this->treeArray;
        }
    }
?>
```

File : Kompres.php

```
<?php
    class Kompres
    {
        protected $tree;
        protected $tree0;
        protected $tree1;
        protected $fileName = "\Code.json";
        protected $stringData;
        protected $stringCode = null;
        protected $direktory;
        public function __construct() {
            $this->direktory = dirname(__FILE__);
           $this->tree = json_decode(file_get_contents($this-
>direktory.$this->fileName));
           $this->tree0 = $this->tree[0];
           $this->tree1 = $this->tree[1];
        }
        public function printTree()
```

```
        {
            return $this->tree;
        }
        public function getCode($character)
        {
            $value = null;
           foreach ($this->tree0 as $key) {
               if ($key->value == $character) {
                   $value = $key;
                   break;
               }
           }
           if (empty($value)) {
               foreach ($this->tree1 as $key) {
                    if ($key->value == $character) {
                        $value = $key;
                        break;
                    }
                }
           }
           return $value->prefixCode;
        }
        public function setData($data)
        {
            $this->stringData = preg_replace('/[^A-Za-z\s]/', '', $data);
        }
        public function getKompres()
        {
            $char =str_split(strtolower($this->stringData));
            foreach ($char as $key) {
                $this->stringCode = $this->stringCode.$this-
>getCode(ord($key));
            }
            return $this->stringCode;
        }
    }
?>
```

File : Dekompres.php

```
<?php
class Dekompres
{
    protected $stringCode;
    protected $stringTemp = null;
    protected $stringHasil = null;
    protected $tree;
    protected $tree0;
    protected $tree1;
    protected $fileName = "\Code.json";
    protected $direktory;
    public function __construct() {
        $this->direktory = dirname(__FILE__);
        $this->tree = json_decode(file_get_contents($this-
>direktory.$this->fileName));
        $this->tree0 = $this->tree[0];
        $this->tree1 = $this->tree[1];
    }
    public function setData($data)
    {
        $this->stringCode = $data;
        $this->stringHasil = null;
    }
```

```
    public function getData()
    {
        return $this->stringCode;
    }
    public function getString()
    {
        $this->stringHasil = null;
        $code = str_split($this->stringCode);
        $treeCode = null;
        $huruf = null;
        foreach ($code as $key) {
            $this->stringTemp = $this->stringTemp.$key;
            if ( strlen($this->stringTemp) > 1) {
                $huruf = $this->cekToArrayTree($this-
>stringTemp,$treeCode);
                if (!empty($huruf->value)) {
                    $this->stringHasil = $this->stringHasil.$huruf-
>huruf;
                    $this->stringTemp = null;
                    //echo "hasilnya = ".$this->stringHasil."<br>";
                }
            }else{
                if ($this->stringTemp == 0) {
                    $treeCode = 0;
                }else{
                    $treeCode = 1;
                }
            }
        }
        return $this->stringHasil;
    }
    private function cekToArrayTree($code, $tree)
    {
        $hasil = null;
        if ($tree == 0) {
            foreach ($this->tree0 as $key) {
               if ($key->prefixCode === $code) {
                   $hasil = $key;
                   break;
               }
           }
        }else{
            foreach ($this->tree1 as $key) {
               if ($key->prefixCode === $code) {
                   $hasil = $key;
                   break;
               }
           }
        }
        return $hasil;
    }}?>
```

Nama	Ridwan Wulida Siam
Tempat Tanggal Lahir	Kulonprogo, 31 desember 1997
Alamat	Kopat, Karangsari, Pengasih, Kulonprogo, Daerah Istimewa Yogyakarta
Jenis kelamin	Laki-laki
Hp	085729785652
Email	rwsiam.rw@gmail.com
Website	kalau.web.id
Penguasaan bahasa	Indonesia, Jawa, Inggris
Computer	Ms Word, Excel, Powerpoint, Access, Coreldraw, Visual Studio, Netbean
	Mengerti Bahasa Pemrograman(C++ Dan Java), Html, Javascript, Css, Php, Database(Oracle Dan Mysql), Dan Dasar Jaringan Komputer

Pendidikan
UIN Sunan Kalijaga – Teknik Informatika (2014-2018)
SMA 1 Wates (2012-2014)
SMP 1 wates (2009-2012)
SD Percobaan 4 (2003-2009)

Organisasi
Kelompok Studi Linux Sunan Kalijaga – Devisi internal (2015-2017)
INFINITY Sunan Kalijaga – anggota (2015-2017)

Workshop/training
Workshop HTML – Infinity Sunan Kalijaga (2016)
Workshop Fiber Optic – Biznet Dan Insect Sunan Kalijaga (2016)
Workshop HereMap – Infinity Sunan Kalijaga (2016)
Training ict – Sunan Kalijaga (2015)

Pengalaman Kerja/Kegiatan
Magang di Global Top Career (2016)
Asisten Dosen (2015-2017)

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA