

Perancangan Aplikasi Untuk Kompresi Ayat-Ayat Di Dalam Qanun Aceh Menerapkan Algoritma Huffman

Ahmad Rafiqi^{1*}, Nelly Astuti Hasibuan¹, Imam Saputra¹

¹Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: ^{1*}ahmadrafiqi123@gmail.com

Abstrak—Data Qanun Aceh yang tersimpan dalam bentuk digital ini mempunyai beberapa kelebihan dibanding dengan data berbentuk tercetak, antara lain adalah data aman dari serangga pengerat, data bisa berpindah-pindah dengan mudah, namun qanun yang berbentuk digital memiliki masalah, banyaknya pasal-pasal dan ayat-ayat mengakibatkan ukuran file menjadi besar sehingga pemakaian memori yang sangat besar. Adapun solusi dalam permasalahan diatas adalah bagaimana file yang besar tersebut dapat dikompresi guna untuk mempercepat perpindahan dan menyimpan file tersebut. Kompresi pada file teks dilakukan dengan memperkecil ukuran file tersebut dengan proses mengurangi bit pada file yang di simpan. Dengan menerapkan algoritma huffman untuk melakukan teknik kompresi, file yang besar akan berkurang ukurannya sehingga dapat menghemat ruang penyimpanan. Dalam penelitian ini dengan menggunakan algoritma tersebut, hasil kompresi akan menguntungkan dalam melakukan pengiriman dan pemindahan file teks akan semakin mudah.

Kata Kunci: Aplikasi, Kompresi, Ayat Qanun Aceh, Huffman

Abstract—The Aceh Qanun data stored in digital form has several advantages compared to printed data, including data that is safe from rodents, data can move around easily, but digital qanuns have problems, there are many articles and verses. paragraph causes the file size to be large so that it uses a very large memory. The solution to the problem above is how large files can be compressed in order to speed up moving and saving the file. The compression of the text file is done by reducing the size of the file by reducing the bits in the saved file. By applying the Huffman algorithm to perform compression techniques, large files will be reduced in size so as to save storage space. In this study, using this algorithm, the compression results will be beneficial in making sending and moving text files easier.

Keywords: Application, Compression, Aceh Qanun Verse, Huffman.

1. PENDAHULUAN

Perkembangan jaman yang semakin pesat membuat komputerisasi pada kehidupan sehari-hari semakin wajar. Data-data yang dahulu hanya tersimpan dalam bentuk tercetak, saat ini disimpan dalam bentuk digital. Data yang disimpan dalam bentuk digital ini mempunyai beberapa kelebihan dibanding dengan data tercetak, antara lain adalah data akan aman dari serangga pengerat, data bisa dipindah-pindahkan dengan mudah. Namun semakin hari data yang ada akan semakin banyak, hal itu menyebabkan adanya kebutuhan akan memori penyimpanan semakin besar. Kebutuhan terhadap kapasitas penyimpanan yang semakin besar merupakan penyebab munculnya berbagai teknik kompresi data.

Qanun adalah peraturan perundang-undangan sejenis peraturan daerah yang mengatur penyelenggaraan pemerintah dan kehidupan masyarakat di Provinsi Aceh. Qanun mempunyai banyak pasal-pasal dan di dalam pasal terdapat ayat-ayat. Qanun berbentuk cetakan sehingga banyak menghabiskan kertas. Untuk mengatasi masalah tersebut maka perlu untuk membuat qanun yang berbentuk digital, qanun yang berbentuk digital mempermudah penyebarluasannya dan mempermudah mencari ayat qanun lebih cepat, namun qanun yang berbentuk digital memiliki masalah, banyaknya pasal-pasal dan ayat-ayat mengakibatkan ukuran file besar sehingga pemakaian memori yang besar. File qanun berbentuk digital dengan ukuran file terlalu besar maka membutuhkan teknik untuk mengecilkan file. Dengan melakukan kompresi sehingga dapat menghemat kapasitas penyimpanan.

Kompresi merupakan teknik yang dapat digunakan untuk mengecilkan atau memampatkan ukuran. Kompresi data adalah teknik untuk mengecilkan data sehingga dapat diperoleh file dengan ukuran yang lebih kecil daripada ukuran file aslinya. Dengan melakukan kompresi, data yang besar akan berkurang ukuran sehingga dapat menghemat kapasitas penyimpanan dan waktu untuk pertukaran atau transmisi data.

Ada dua teknik yang dapat dilakukan dalam melakukan kompresi data yaitu Lossless Compression dan Lossy Compression. Lossless Compression merupakan kompresi data dimana hasil dekompresi dari data yang terkompresi sama dengan data aslinya dan tidak ada informasi yang hilang. Sedangkan Lossy Compression adalah kompresi data dimana hasil dekompresi dari data yang terkompresi tidak sama dengan data aslinya karena ada informasi yang hilang, tetapi masih dapat ditolerir oleh persepsi mata. Beberapa teknik kompresi menggunakan peluang kemunculan suatu karakter untuk menghasilkan kode kompres, misalnya pengkodean huffman.

Huffman encoding sangat ampuh untuk pesan teks yang memiliki terjadinya karakter yang sama[1]. Dan untuk rasio kompresi yang dihasilkan cukup bervariasi tergantung jenis file yang dikompres. Untuk file audio dan image rasio kompresi yang dihasilkan lebih kecil sedangkan untuk file teks rasio kompresi yang dihasilkan lebih besar[2]. Kemudian Tingkat keamanan data setelah dikompresi cukup terjaga, dengan kata lain file tidak berkurang atau mengalami kerusakan setelah proses kompresi data dilakukan[2]. Berdasarkan hasil dari penelitian sebelumnya maka sesuai untuk menerapkan algoritma huffman dalam mengompres Ayat Qanun Aceh. Didalam Qanun Aceh banyak terdapat karakter yang sama dan tidak memiliki image.

2. METODOLOGI PENELITIAN

2.1 Kompresi

Kompresi data merupakan proses mereduksi ukuran suatu data untuk menghasilkan representasi digital yang padat atau mampat (compact) namun tetap dapat mewakili kuantitas informasi yang terkandung pada data tersebut. Pada citra, video, dan audio, kompresi mengarah pada minimasi jumlah bit rate untuk merepresentasi digital. Pada beberapa literatur, istilah kompresi sering disebut juga source coding, data compression, bandwidth compression, dan signal compression[3].

2.2 Algoritma Huffman Coding

Algoritma Huffman, yang dibuat oleh seorang mahasiswa MIT bernama David Huffman pada tahun 1952, merupakan salah satu metode paling lama dan paling terkenal dalam kompresi teks. Algoritma Huffman menggunakan prinsip pengkodean yang mirip dengan Morse, yaitu tiap karakter (simbol) dikodekan hanya dengan rangkain beberapa bit, dimana karakter yang sering muncul dikodekan dengan rangkain bit yang pendek dan karakter yang jarang muncul dikodekan dengan rangkain bit yang lebih panjang. Berdasarkan tipe peta kode yang digunakan untuk mengubah pesan awal (isi data yang diinputkan) menjadi sekumpulan codeword, algoritma Huffman termasuk kedalam kelas algoritma yang menggunakan metode statik. Metode statik adalah metode yang selalu menggunakan peta kode yang sama, metode ini membutuhkan dua fase (two-pass), fase pertama untuk menghitung probabilitas kemunculan tiap simbol dan menentukan peta kodenya, dan fase kedua untuk mengubah pesan menjadi kumpulan kode yang akan di transmisikan. Sedangkan berdasarkan teknik pengkodean simbol yang digunakan, algoritma Huffman menggunakan metode symbolwise. Metode symbolwise adalah metode yang menghitung peluang kemunculan dari setiap simbol dalam satu waktu, dimana simbol yang lebih sering muncul diberi kode lebih pendek dibandingkan simbol yang jarang muncul[2].

Kompresi data secara umum merupakan proses yang dapat mengubah sebuah data masukan (sumber data asli) ke data yang lain (data keluaran) yang memiliki ukuran yang lebih kecil, tujuan dari kompresi data adalah untuk merepresentasikan suatu data digital dengan sesedikit mungkin bit, tetapi tetap mempertahankan kebutuhan minimum untuk membentuk kembali data aslinya. Sedangkan Enkripsi merupakan bagian dari algoritma kriptografi yang akan merahasiakan informasi dengan menyandikannya kedalam bentuk yang tidak dimengerti lagi maknanya[6].

File akan dirubah strukturnya dengan proses pemampatan menggunakan algoritma Huffman dan kemudian dienkripsi menggunakan algoritma Huffman yang digunakan. Proses pengembalian file yang sudah dimampatkan dan proses deskripsi menggunakan algoritma Huffman akan mengembalikan file yang telah di dimampatkan dan di enkripsi ke bentuk file semula dan juga akan mengembalikan ukuran file ke ukuran semula.

Kode Huffman pada dasarnya merupakan kode prefiks (prefix code). Kode prefiks adalah himpunan yang berisi sekumpulan kode biner yang dalam hal ini tidak ada kode biner yang menjadi awal bagi kode biner yang lain. Kode prefiks biasanya direpresentasikan sebagai pohon biner yang berlabel, dimana setiap sisi diberi label 0 (cabang kiri) atau 1 (cabang kanan). Rangkaian bit yang terbentuk pada setiap lintasan dari akar ke daun merupakan kode prefiks untuk karakter yang berpadanan. Pohon biner ini biasa disebut pohon Huffman. Langkah-langkah pembentukan pohon Huffman sebagai berikut[6] :

1. Baca semua karakter di dalam data untuk menghitung frekuensi kemunculan setiap karakter. Urutkan karakter dari yang paling kecil frekuensi kemunculannya.
2. Gabungkan dua buah pohon yang mempunyai frekuensi terkecil pada sebuah akar. Akar mempunyai frekuensi yang merupakan jumlah dari frekuensi dua buah pohon penyusunnya.
3. Ulangi langkah 2 sampai hanya tersisa satu buah pohon Huffman. Agar pemilihan dua pohon yang akan digabungkan berlangsung cepat, maka semua pohon yang ada selalu terurut menaik berdasarkan frekuensi.
4. Kodekan tiap karakter dengan susunan bit biner hasil dari pohon Huffman.
5. Ganti susunan karakter (kalimat) awal dengan susunan bit biner yang dihasilkan dari pohon biner dilangkah ke 2.

Enkripsi merupakan bagian dari algoritma kriptografi yang akan merahasiakan informasi dengan menyandikannya kedalam bentuk yang tidak dimengerti lagi maknanya. proses mengganti susunan karakter menjadi susunan bit merupakan proses enkripsi yang dimiliki oleh algoritma Huffman, selain untuk membuat file yang lebihkecil dari bentuk awal proses peenggantian susunan karakter dengan susunan bit merupakan proses enkripsi.

Dasar pemikiran algoritma kompresi Huffman adalah bahwa setiap karakter ASCII biasanya diwakili oleh 8 bits. Jadi misalnya suatu file berisi deretan karakter "ABACAD" maka ukuran file tersebut adalah $6 \times 8 \text{ bits} = 48 \text{ bit}$ atau 6 bytes. Jika setiap karakter tersebut di beri kode lain secara sembarang, misalkan A=1, B=00, C=010, dan D=011, berarti hanya perlu file dengan ukuran 11 bits (10010101011). Untuk menentukan kode-kode pengganti dari karakter ASCII yang tersebut dengan kriteria bahwa kode harus unik dan karakter yang sering muncul dibuat kecil jumlah bitnya, hal tersebut dapat menggunakan algoritma Huffman.

2.3 Qanun

Qanun dalam Kamus Bahasa Indonesia dikenal dengan istilah "kanun" yang artinya undang-undang, peraturan, kitab undang-undang, Hukum, Kaidah. Istilah kanun tersebut juga ditemukan dalam kamus Aceh-Indonesia I, yakni "kanun", yang diartikan, peraturan, Undang-undang, hukum atau adat kebiasaan. Kanun diartikan juga sebagai peraturan atau ketentuan-ketentuan raja yang sedang memerintah[7].

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Analisa yang dilakukan yaitu mengkompresi data teks dengan menerapkan algoritma huffman. Kompresi merupakan pengurangan ukuran suatu data menjadi ukuran yang lebih kecil dari aslinya. Pengompresian data teks ini sangat menguntungkan manakala terdapat suatu data yang berukuran besar dan data di dalamnya mengandung banyak pengulangan karakter. Adapun teknik dari kompresi ini adalah dengan mengganti karakter yang berulang-ulang tersebut dengan suatu pola tertentu sehingga berkas tersebut dapat meminimalisasi ukurannya. Algoritma tersebut dimulai dengan memberikan rangkain string sebagai masukan, bagaimana menghasilkan keluaran algoritma berupa string biner atau kode yang menterjemahkan setiap string masukan agar string tersebut mempunyai jumlah bit yang sedikit dibandingkan dengan string yang tidak dimampatkan. Dengan demikian, masalahnya adalah bagaimana memperoleh kode tersebut dengan karakter yang telah diurutkan dan tabel frekuensinya sebagai masukan dan kode biner yang lebih pendek sebagai keluaran.

Dalam hal ini penerapan algoritma huffman dilakukan terhadap ayat pada aplikasi Qanun Aceh yang memiliki keterbatasan memori sehingga menggunakan memori yang relatif besar. Mengompres ayat pada aplikasi Qanun Aceh merupakan teknik kompresi sehingga memperkecil ukuran aplikasi Qanun Aceh dan dapat digunakan pada memori yang kecil. Sehingga diperoleh ukuran memori yang lebih kecil dari aslinya.

Adapun tujuan dari analisa terhadap sistem yang dibangun yaitu untuk mengetahui dan merumuskan kebutuhan dari sistem serta membantu meminimalisir sumber daya yang berlebih. Berikut merupakan proses kompresi ayat pada aplikasi Qanun Aceh.

3.1.1. Penerapan Algoritma Huffman

Algoritma huffman merupakan salah satu teknik kompresi lossless dengan cara menggantikan suatu deret simbol input dengan suatu data teks dengan sebuah bilangan. Semakin panjang karakter di dalam aplikasi Qanun Aceh, maka akan semakin banyak bit yang diperlukan guna untuk proses kompresi dan dekompresi data teks. Contoh kata yang di kompresi adalah “mpu bersidang sekurang-kurangnya 3 kali dalam 1 tahun”. Bila dikodekan menggunakan kode huffman kemunculan tiap-tiap simbol dan urutkan dari yang terbesar hingga yang terkecil. Buat daftar frekuensi kemunculan tiap simbol dan urutkan dari yang terbesar hingga terkecil.

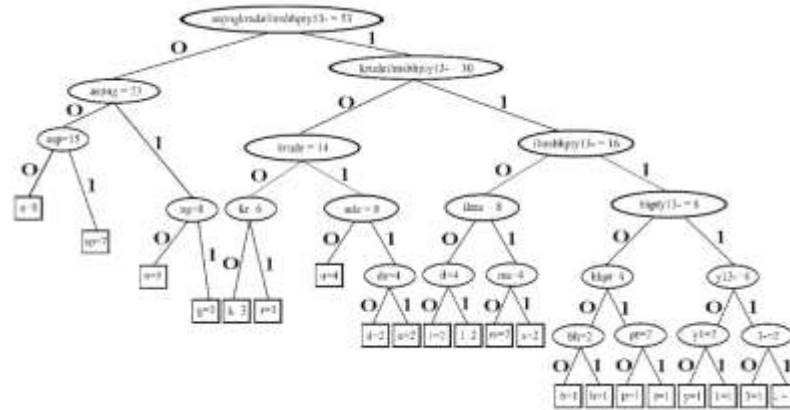
Tabel 1. Probalitas kemunculan

String	Frekuensi	Bit	F x Bit
a	8	01100001	64
sp	7	00100000	56
n	5	01101110	40
u	4	01110101	32
g	3	01100111	24
k	3	01101011	24
r	3	01110010	24
d	2	01100100	16
e	2	01100101	16
i	2	01101001	16
l	2	01101100	16
m	2	01101101	16
s	2	01110011	16
b	1	01100010	8
h	1	01101000	8
p	1	01110000	8
t	1	01110011	8
y	1	01111001	8
1	1	00110001	8
3	1	00110011	8
-	1	00101101	8
Total Bit			424

1. Proses kompresi

Maka, rangkain bit untuk string “mpu bersidang sekurang-kurangnya 3 kali dalam 1 tahun” adalah: 01101101 01110000 01110101 00100000 01100010 01100101 01110010 01110011 01101001 01100100 01100001 01101110 01100111 00100000 01110011 01100101 01101011 01101011 01110101 01110010 01100001 01101110 01100111 01101110 01111001 00100000 00110011 01101011 01101011 01101100 01101001 00100000 01100100 01100001 01101100 01100001 01101101 00100000 00110001 01110011 01100001 01101000 01101110. Berdasarkan metode pengkodean ASCII, setiap huruf direpresentasikan dalam 8-bit sehingga untuk merepresentasikan 53 karakter akan membutuhkan $53 \times 8 = 424$ bit.

Sebagai contoh, dalam kode ASCII string 53 karakter akan membutuhkan ruang sebesar 424 bit atau 53 byte, karena setiap 1 karakter menempati 8 bit atau 1 byte, maka susunan huruf “mpu bersidang sekurang-kurangnya 3 kali dalam 1 tahun” akan mempunyai ukuran file sebesar 53 byte. Dengan algoritma huffman akan dibentuk pohon biner huffman setiap karakter akan diganti dengan kode biner sebagai berikut:



Gambar 1. Pohon Huffman

Adapun kode-kode biner yang di hasilkan dari pohon huffman tersebut dapat di lihat dari tabel 2.

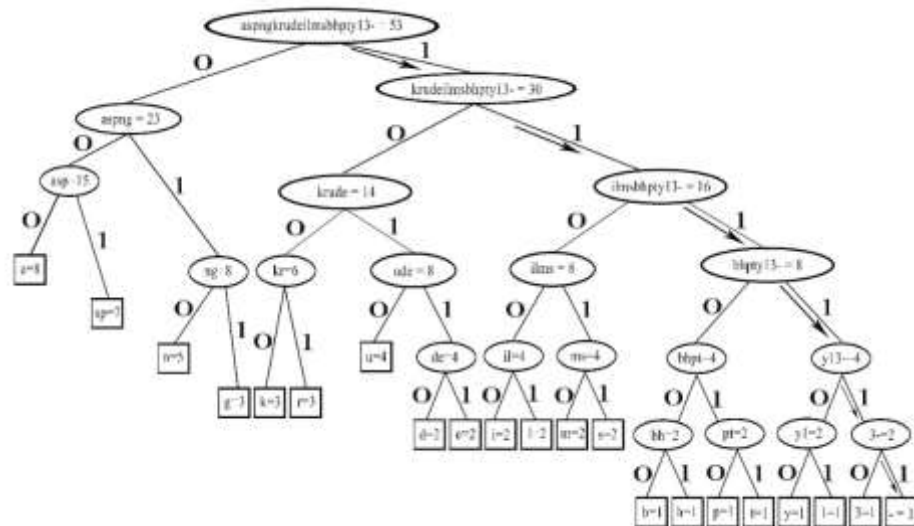
Tabel 2. kode pengganti berdasarkan pohon huffman

Simbol	Frekuensi	Peluang	Bit	F x Bit
a	8	8/53	000	24
sp	7	7/53	001	21
n	5	5/53	010	15
u	4	4/53	1010	16
g	3	3/53	011	9
k	3	3/53	1000	12
r	3	3/53	1001	12
d	2	2/53	10110	10
e	2	2/53	10111	10
i	2	2/53	11000	10
l	2	2/53	11001	10
m	2	2/53	11010	10
s	2	2/53	11011	10
b	1	1/53	111000	6
h	1	1/53	111001	6
p	1	1/53	111010	6
t	1	1/53	111011	6
y	1	1/53	111100	6
1	1	1/53	111101	6
3	1	1/53	111110	6
-	1	1/53	111111	6
Total Bit				217

Dari tabel 2. diperoleh kode huffman string “mpu bersidang sekurang-kurangnya 3 kali dalam 1 tahun” adalah: 11010 111010 1010 001 111000 10111 1001 11011 11000 10110 000 010 011 001 11011 10111 1000 1010 1001 000 010 011 111111 1000 1010 1001 000 010 011 010 111100 000 001 111110 001 1000 000 11001 11000 001 10110 000 11001 000 11010 001 111101 001 111011 000 111001 1010 010. Susunan biner-biner yang sudah terkompres, menghasilkan karakter “×Tx¼i””P*B ÅHMxÆpl24} =Ži”. Untuk bit yang di hasilkan = 217 bit, maka rasio kompresinya adalah. Rasio = (424 - 217) / 424 x 100% = 48.82%.

2. Proses Dekompresi

Untuk melakukan dekompresi dengan biner, 11010 111010 1010 001 111000 10111 1001 11011 11000 10110 000 010 011 001 11011 10111 1000 1010 1001 000 010 011 010 111111 1000 1010 1001 000 010 011 010 111100 000 001 111110 001 1000 000 11001 11000 001 10110 000 11001 0001 11010 001 111101 001 111011 000 111001 1010 010, maka digunakan pohon huffman untuk menelusuri string di atas sebagai berikut:



Gambar 2. Proses Dekompresi

Sebagai contoh medekomresi string bernilai 11111, setelah ditelusuri menggunakan pohon biner mulai dari akar, maka diketahui string bernilai 111111 adalah karakter – (lihat tabel 2). cara dekomresi kedua adalah dengan cara memperhatikan tabel kode huffman hasil dari pohon biner huffman (lihat tabel 2), maka proses dekomresi dari susunan biner dapat dilakukan dengan mudah, prosesnya sebagai berikut, baca susunan biner 11010 111010 1010 001 111000 10111 1001 11011 11000 10110 000 010 011 001 11011 10111 1000 1010 1001 000 010 011 111111 1000 1010 1001 000 010 011 010 111100 000 001 111110 001 1000 000 11001 11000 001 10110 000 11001 000 11010 001 111101 001 111011 000 111001 1010 010 bit pertama dari susunan tersebut adalah 11010 yang merupakan pengganti dari karakter m, baca bit berikutnya, yaitu 1, tidak ada kode huffman 1 pada tabel maka baca bit berikutnya sehingga menjadi 11, 11 juga tidak ditemukan, maka baca lagi bit berikutnya sampai bit tersusun menjadi 111010, rangkain bit 111010 diketahui adalah pengganti dari karakter p, baca bit berikutnya yaitu 1, 1 tidak ada kode huffman pada tabel maka baca bit berikutnya sampai rangkain bit menjadi 1010 yang merupakan pengganti dari karakter u, lakukan pembacaan bit sampai bit terakhir sehingga proses dekomresi akan menghasilkan susunan karakter semula yaitu “mpu bersidang sekurang-kurangnya 3 kali dalam 1 tahun”. Dengan susunan biner 8 bit dari string tersebut 01101101 01110000 01110101 00100000 01100010 01100101 01110010 01110011 01101001 01100100 01100001 01101110 01100111 00100000 01110011 01100101 01101011 01110101 01110010 01100001 01101110 01100111 01101110 01111001 00100000 00110011 01101011 01101011 01101100 01101001 00100000 01100100 01100001 01101100 01100001 01101101 00100000 00110001 01110011 01100001 01101000 01101110.

3.2. Implementasi

Kebutuhan sistem ini mencakup dari spesifikasi kebutuhan tentang perangkat yang digunakan seperti software (perangkat lunak) dan hardware (perangkat keras). Implementasi suatu proses pelaksanaan dan penerapan yang menuju aktivitas kegiatan yang terencana dalam perancangan sistem yang diterapkan dalam bahasa pemrograman. Implementasi suatu sistem sangat berpengaruh pada spesifikasi komputer yang digunakan, dimana nantinya program agar bisa di jalankan dengan baik, adapun kebutuhan sistem yang digunakan sebagai berikut:

1. Perangkat Keras (Hardware)

Perangkat keras adalah suatu peralatan komputer yang dilihat secara fisik dan dapat disentuh secara langsung.

Perangkat keras yang dibutuhkan untuk menjalankan sistem dikomputer dengan spesifikasi komputer.

a. Processor : Intel®Core™i3-4005UCPU@1.70GHz(4 CPUs),~1,7GHz

b. Momory (RAM) : 4069MB RAM

c. Harddisk

2. Perangkat Lunak (Software)

Perangkat lunak adalah sistem yang digunakan untuk mengolah data atau aplikasi tertentu, adapun perangkat lunak (software) yang digunakan agar aplikasi ini berjalan dengan baik adalah sebagai berikut:

a. Sistem operasi merupakan suatu sistem yang mengendalikan seluruh komponen yang ada didalam sebuah komputer, sehingga sistem operasi yang digunakan hal ini adalah minimal MS-Windows 7.

b. Software yang digunakan dalam menjalankan program berupa software dengan bahasa pemrograman. Berdasarkan pada sub bab sebelumnya untuk membangun atau merancang suatu program menggunakan aplikasi Microsoft Visual Studio 2008

Tampilan aplikasi Qanun Aceh ini merupakan tampilan yang dimana akan menjelaskan beberapa form seperti, form Qanun Aceh untuk melakukan pencarian aya-ayat, form About menerangkan informasi profil penulis. Tampilan menu utama merupakan form yang pertama kali muncul pada saat aplikasi dijalankan. Form menu utama ini memiliki

beberapa sub menu diantaranya adalah form Qanun Aceh, form About, Keluar. Adapun tampilan halaman menu utama pada aplikasi dapat dilihat sebagai berikut:



Gambar 3. form Menu Utama

Form Qanun Aceh merupakan form yang digunakan untuk melakukan pencarian ayat Qanun Aceh, dimana didalam form terdapat textbox cari judul UUD, cari pasal, cari ayat dan button cari, batal, keluar. Terlihat seperti digambar 4.



Gambar 4. Form Qanun Aceh

Pada gambar 5. melakukan penginputan kata, dimana kata yang di cari yaitu, cari judul UUD, Pasal, Ayat. Setelah melakukan penginputan kata pada setiap textbox dan hanya 1 (satu) textbox yang tidak melakukan penginputan yaitu textbox hasil pencarian, untuk melihat hasil yang dicari terutama mengklik tombol cari, seperti gambar berikut:



Gambar 5. form Qanun Aceh Input

dari penjelasan pada gambar 5. yang sudah melakukan penginputan kata, kemudian melakukan mengklik tombol cari akan menampilkan hasil pencarian pada textbox yang tidak melakukan penginputan, seperti gambar berikut:



Gambar 6. form Qanun Aceh Hasil Pencarian

Kemudian pada textbox cari judul UUD melakukan penginputan kata yang di masukkan salah, akan menampilkan pesan pada textbox hasil pencarian atau textbox yang di melakukan penginputan, dan begitu juga pada setiap textbox cari pasal, textbox cari ayat, seperti gambar berikut:



Gambar 7. form Qanun Aceh Penginputan Salah

Tombol batal, fungsi dari tombol batal mengosongkan semua isi textbox seperti gambar berikut:



Gambar 8. form Qanun Aceh Fungsi Tombol

4. KESIMPULAN

Setelah mengimplementasikan perangkat lunak kompresi ayat qanun aceh dengan algoritma huffman, maka dapat diambil beberapa kesimpulan dari bab-bab sebelumnya Qanun Aceh dapat disajikan dalam bentuk digital. Prosedur kompresi teks dengan menerapkan algoritma huffman telah berhasil melakukan proses kompresi yang sesuai dengan langkah-langkah untuk melakukan proses kompresi dan dekompresi, sehingga proses dekompresi berjalan sesuai dengan teknik kompresi. Kompresi teks ayat qanun aceh dapat dirancang dan dibangun dengan menggunakan bahasa pemrograman Microsoft Visual Studio 2008, dengan menerapkan algoritma huffman sehingga bisa diharapkan untuk mempermudah pengguna. Rasio yang dihasilkan dalam mengompresi teks menggunakan algoritma huffman adalah 48.82%.

REFERENCES

- [1] A. Putera, U. Siahaan, U. Pembangunan, and P. Budi, "Implementasi Teknik Kompresi Teks Huffman," no. August, 2016.
- [2] A. Wibowo, "Kompresi data menggunakan metode huffman," vol. 2012, no. Semantik, pp. 47–51, 2012.

- [3] D. Putara, Pengolahan Citra Digital. Yogyakarta: ANDI OFFSET, 2010.
- [4] D. Salomon, Data Compression. USA: Springer, 2010.
- [5] A. Pahdi, "Algoritma Huffman Dalam Pemampatan Dan Enkripsi Data," vol. 6, no. 3, pp. 1–7, 2017.
- [6] D. A. N. Materi and M. Qanun, "'power tends to corrupt and absolute power corrupts absolutely,'" vol. 05, no. 01, pp. 17–33, 2017.
- [7] Ihsan and D. P. Utomo, "Analisis Perbandingan Algoritma Even-Rodeh Code Dan Algoritma Subexponential Code Untuk Kompresi File Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
- [8] S. R. Saragih and D. P. Utomo, "Penerapan Algoritma Prefix Code Dalam Kompresi Data Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
- [9] Lamsah and D. P. Utomo, "Penerapan Algoritma Stout Codes Untuk Kompresi Record Pada Databade Di Aplikasi Kumpulan Novel," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.