

IMPLEMENTASI KOMPRESI DATA TEXT MENGGUNAKAN HUFFMAN CODING

¹Devie R. Suchendra, ²Sandra Wulandari

¹Program Studi Sistem Informasi STMIK LPKIA

²Program Studi Teknik Informatika STMIK LPKIA

Jln. Soekarno Hatta No. 456 Bandung 40266, Telp. +62 22 75642823, Fax. +62 22 7564282

Email : sonifajar@gmail.com

Abstrak

Kompresi ialah proses pengubahan sekumpulan data menjadi suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data. Pendekatan teknik kompresi lossless adalah teknik kompresi data yang tidak mengalami kehilangan informasi selama proses kompresi. Kompresi lossless disebut kompresi *reversibel* karena data asli dapat dipulihkan sempurna oleh dekompresi. Teknik kompresi lossless banyak digunakan ketika data asli dari sumber sangat penting dan tidak mampu untuk kehilangan detail, dengan kata lain berarti teknik lossless merupakan salah satu teknik yang bermanfaat bagi seseorang dimana sifat dari kompresi data ini adalah dapat melakukan kompresi data tanpa menghilangkan informasi saat dilakukan kompresi, sehingga data kembali seperti semula ketika dilakukan dekompresi kembali, maka teknik ini menjadi alternatif kompresi data yang sangatlah bermanfaat.

Untuk mengimplementasikan kompresi data text ini akan lebih baik menggunakan salah satu algoritma yang bersifat lossless sehingga untuk mewujudkan itu, implementasi kompresi data text ini dapat menggunakan algoritma Huffman sebagai algoritma untuk melakukan proses kompresi dan dekompresi yang secara umum efisien dalam mengkompresi teks.

Kata kunci : *Kompresi, Text, Huffman Coding*

1. Pendahuluan

Kebutuhan manusia yang sangat memerlukan informasi telah memacu semangat para pengembang teknologi informasi untuk mencoba mengimbangi tuntutan tersebut. Salah satu bagian terpenting dalam dunia teknologi informasi adalah bagaimana cara mengolah data dari informasi-informasi yang semakin meningkat tersebut.

Data yang dikirim seringkali ukurannya sangat besar sehingga membutuhkan waktu pengiriman yang lama. Begitu juga dalam penyimpanan data, arsip atau file yang berukuran besar akan memakan ruang penyimpanan yang besar. Kedua permasalahan tersebut dapat diatasi dengan mengkodekan data atau isi sesingkat mungkin, sehingga waktu pengiriman data relatif cepat, dan ruang penyimpanan yang dibutuhkan juga lebih efisien. Pengkodean seperti ini disebut pemampatan atau kompresi data.

Kompresi data diharuskan dapat menghasilkan ukuran file menjadi lebih kecil dari ukuran semula yang mana tidak ada informasi data yang hilang atau berkurang selama proses kompresi, sehingga algoritma kompresi data yang dibutuhkan adalah kompresi data yang bersifat lossless yang mana setelah proses dekompresi, jumlah bit (byte) data atau informasi dalam keseluruhan file hasil sama persis dengan file aslinya.

Adapun permasalahan yang ditemukan pada implementasi kompresi data text menggunakan Huffman coding, antara lain adalah :

1. Bagaimana algoritma Huffman melakukan kompresi data untuk mengurangi redundancy kemunculan karakter pada file yang berukuran besar.
2. Bagaimana algoritma Huffman mengkompresi data pada karakter yang mempunyai frekuensi kemunculan yang sama.

Berdasarkan permasalahan yang ada di atas maka perlu membatasi ruang lingkup dari permasalahan tersebut. Adapun permasalahan yang akan dibahas meliputi :

1. Membahas teknik kompresi menggunakan algoritma Huffman pada data text.
2. Membahas prediksi algoritma Huffman pada data dengan frekuensi kemunculan karakter yang sama.

Adapun tujuan dari perancangan sistem yang baru adalah sebagai berikut :

1. Mengetahui proses kerja algoritma Huffman dalam melakukan kompresi data untuk mengurangi redundancy kemunculan karakter pada file yang berukuran besar
2. Mengetahui hasil prediksi penerapan algoritma Huffman pada data yang memiliki frekuensi kemunculan karakter sama.

1.1. Landasan Teori

Algoritma Huffman ditemukan oleh David Huffman pada tahun 1952. Algoritma ini menggunakan pengkodean yang mirip dengan kode

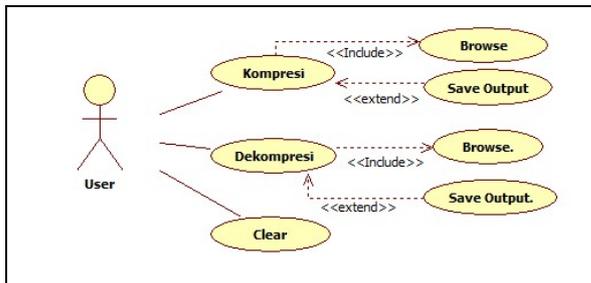
Morse. Berdasarkan tipe kode yang digunakan algoritma Huffman termasuk metode statistic. Sedangkan berdasarkan teknik pengkodeannya menggunakan metode symbolwise. Algoritma Huffman merupakan salah satu algoritma yang digunakan untuk mengompres teks.

2. Gambaran Perangkat Lunak

2.1 Aliran Proses

2.1.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sistem dan merepresentasikan interaksi antara actor dengan sistem.



Gambar 1 Use Case Diagram

Tabel 1 Use Case Skenario Browse

Nomor	1
Nama	Browse
Tujuan	User menentukan file yang akan dikompresi/ yang akan didekompresi
Deskripsi	Merupakan proses pemilihan file yang dalam kasus ini merupakan file berekstension .txt untuk dikompresi atau file berekstension .huff untuk didekompresi.
Aktor	User
Skenario:	
Aksi Aktor	Reaksi Sistem
1.Menekan tombol Browse 3.Memilih file	2.Memunculkan kotak dialog 4.Menampilkan history nama file input 5.Menampilkan alamat directory file terpilih 6.Menampilkan saran alamat directory file hasil kompresi/dekompresi akan tersimpan 7.Memberi caption kompresi/dekompresi pada label proses dan tombol action

Tabel 2Use Case Skenario Save Output

Nomor	2
Nama	Save Output

Tujuan	User mengetahui tempat penyimpanan file hasil kompresi/ dekompresi
Deskripsi	Merupakan proses pemilihan directory file hasil kompresi/ dekompresi akan disimpan
Aktor	User
Skenario:	
Aksi Aktor	Reaksi Sistem
1.Menekan tombol Save Output 3.Memasukan Nama file	2.Memunculkan kotak dialog 4.Menampilkan history nama file output 5.Menampilkan alamat directory file hasil kompresi/ dekompresi akan disimpan

Tabel 3Use Case Skenario Kompresi

Nomor	3
Nama	Kompresi
Tujuan	User menerima file yang telah dikompresi
Deskripsi	Merupakan proses pengkompresian untuk data yang telah diinputkan sebelumnya.
Aktor	User
Skenario:	
Aksi Aktor	Reaksi Sistem
1.Menekan tombol kompresi 4.Menerima file yang telah dikompresi dengan ekstensi .huff	2.melakukan proses kompresi dengan menggunakan Algoritma huffman, dengan mengubah string-string kedalam format biner 3.memberi informasi bahwa proses kompresi telah berhasil

Table 4Use Case Skenario Dekompresi

Nomor	4
Nama	Dekompresi
Tujuan	User menerima file yang telah didekompresi
Deskripsi	sistem menerima masukan file yang diinputkan oleh user dan melakukan proses dekompresi

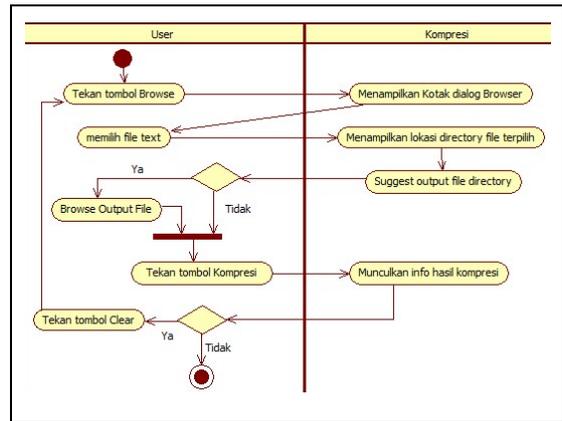
Aktor	User
Skenario:	
Aksi Aktor	Reaksi Sistem
1. Menekan tombol dekompresi	2. Melakukan proses decoding dengan algoritma huffman, dengan mengubah format biner menjadi string
4. Menerima file yang telah didekompresi dengan ekstensi .txt	3. Memberi informasi bahwa proses dekompresi telah berhasil

Tabel 5 Use Case Scenario Clear

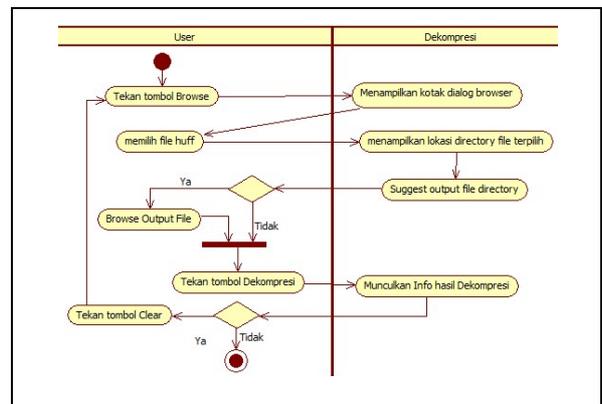
Nomor	5
Nama	Clear
Tujuan	Semua textbox yang telah terisi dalam keadaan bersih.
Deskripsi	Merupakan proses membersihkan semua text box yang telah terisi sebelumnya.
Aktor	User
Skenario:	
Aksi Aktor	Reaksi Sistem
1. Menekan tombol Clear	2. Membersihkan text-text yang terdapat dalam textbox/ textarea

2.2 Activity Diagram

Pada sub bab berikut ini akan dimodelkan aliran kegiatan yang terjadi dalam perangkat lunak kompresi data yang digambarkan dalam *Activity Diagram* dan secara garis besar adalah untuk memodelkan aliran kerja (*workflow*) atau aktivitas dan operasi dari perangkat lunak kompresi data.



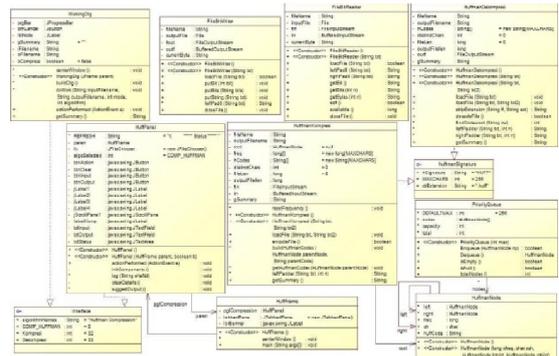
Gambar 2 Activity Diagram Kompresi



Gambar 3 Activity Diagram Dekompresi

2.2 Class Diagram

Class diagram adalah suatu diagram yang menyediakan sekumpulan class objek antar muka interface dan relasinya, dan juga untuk memodelkan database logic.



Gambar 4 Class Diagram

2.5 Perancangan Antar Muka

Perancangan antarmuka ini bertujuan untuk memberikan gambaran mengenai bentuk antarmuka dari perangkat lunak yang akan digunakan oleh user untuk berinteraksi dengan perangkat lunak. Rancangan antarmuka ini mempertimbangkan berbagai kemudahan dan fungsionalitas dari perangkat lunak itu sendiri.

2.4.1 Antar muka Kompresi

Judul Perangkat Lunak		
Input	<input type="text"/>	Browse File
Output	<input type="text"/>	Save Output
Proses	Kompresi	
Status	<input type="text"/>	
Tombol Kompresi		Tombol Clear

Gambar 13Antar muka kompresi

Penjelasan Antar muka Kompresi

1. Perangkat lunak dalam keadaan terbuka
2. Tampilan form kompresi muncul
3. Tekan tombol Browse file untuk mencari file yang akan di kompresi
4. Tekan tombol save output jika directory tempat penyimpanan file hasil, akan dirubah dari yang disarankan sistem sebelumnya.
5. Proses Kompresi akan muncul secara otomatis jika file Input berekstensi .txt
6. Text Area Status akan otomatis terisi dengan kegiatan yang telah dilakukan oleh user pada perangkat lunak.
7. Tekan tombol kompresi untuk melakukan proses kompresi.
8. Tekan tombol Clear untuk membersihkan textfield dan text area

Judul Perangkat Lunak		
Input	<input type="text"/>	Browse File
Output	<input type="text"/>	Save Output
Proses	Dekompresi	
Status	<input type="text"/>	
Tombol Dekompresi		Tombol Clear

Gambar 13Antar muka dekompresi

Penjelasan gambar antar muka Dekompresi

1. Perangkat lunak dalam keadaan terbuka
2. Tampilan form dekompresi muncul
3. Tekan tombol Browse file untuk mencari file yang akan di dekompresi
4. Tekan tombol save output jika directory tempat penyimpanan file hasil, akan dirubah dari yang disarankan sistem sebelumnya.
5. Proses Dekompresi akan muncul secara otomatis jika file Input berekstensi .huff
6. Text Area Status akan otomatis terisi dengan kegiatan yang telah dilakukan oleh user pada perangkat lunak.
7. Tekan tombol dekompresi untuk melakukan proses dekompresi.
8. Tekan tombol Clear untuk membersihkan textfield dan text area

3.Implementasi

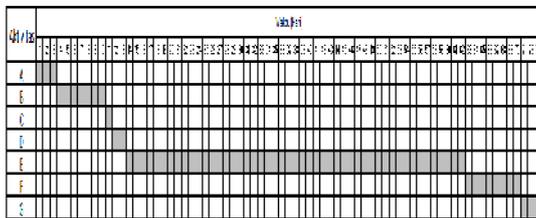
Sub bab ini akan menjelaskan langkah-langkah serta rencana jadwal dalam rangka mengimplementasikan kompresi data text menggunakan algoritma Huffman yang telah dirancang pada bab sebelumnya.

Daftar Kegiatan

Daftar perencanaan kegiatan pengimplementasian system yang akan dirancang sebagai berikut :

1. Analisis kebutuhan
Merupakan tahap awal menentukan kebutuhan apa saja yang diperlukan untuk mengimplementasikan algoritma pada perangkat lunak
2. Pengumpulan Data
Merupakan tahapan pengumpulan informasi, pencarian referensi sebagai acuan pemahaman algoritma, perancangan dan pembangunan perangkat lunak, dan tahap mempelajari kebutuhan akan perangkat keras dan perangkat lunak yang akan digunakan untuk pembuatan perangkat lunak
3. Instalasi Tools yang akan digunakan.
 1. Java SDK
 2. Netbeans 6.8
 Aktifitas ini dilakukan supaya dapat diketahui perkakas apa saja yang dibutuhkan untuk membuat perangkat lunak
Java SDK, merupakan Software Development Kit dari aplikasi java, digunakan karena kita membutuhkan Java Virtual Machine untuk mengembangkan aplikasi berbasis J2SE.
Netbeans 6.8, merupakan IDE (Interface Development Editor) yang digunakan untuk menuliskan kode dan membangun aplikasi Java.
4. Pembuatan Antarmuka
Pembuatan antarmuka ini dilakukan supaya tidak terlalu sulit dalam penentuan tahap penulisan kode program yang harus dilakukan terlebih dahulu.
5. Penulisan kode program
Penulisan kode program dilakukan secara bertahap Testing dan perbaikan pada perangkat lunak
6. Testing dan perbaikan pada perangkat lunak
Testing terhadap kode program dan perbaikan kesalahan saat program dieksekusi
7. Pengujian
Setelah perangkat lunak bebas dari kesalahan-kesalahan langkah selanjutnya adalah tahapan pengujian melakukan kompresi dan dekompresi data text, apakah sesuai harapan atau tidak

Kode Aktifitas	Aktifitas	Waktu (Hari)	Predecessor
A	Analisis kebutuhan	3	-
B	Pengumpulan Data	7	A
C	Install perkakas	1	B
D	Pembuatan Antarmuka	2	C
E	Penulisan kode	53	D
F	Testing dan Perbaikan pada perangkat lunak	8	E
G	Pengujian	3	F



Gambar 14 Jadwal kegiatan

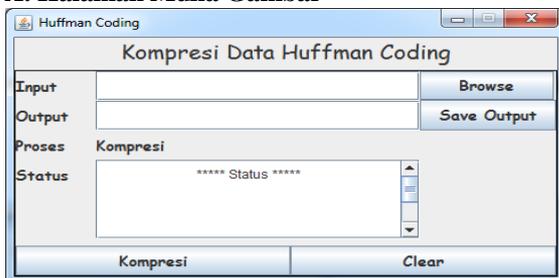
3.1 Lingkup dan Batasan Implementasi

Ruang lingkup dan batasan implementasi terdiri dari:

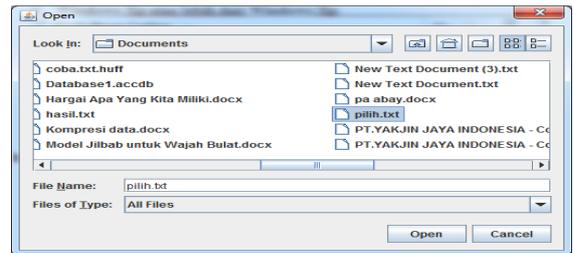
1. Penerapan kompresi data ini hanya pada media komputer
2. Penerapan algoritma Huffman pada kompresi data dengan bahasa pemrograman Java.
3. Penerapan algoritma Huffman pada kompresi data ini hanya dapat melakukan kompresi dan dekompresi pada data text.

3.3 Implementasi Antarmuka

A. Halaman Muka Gambar

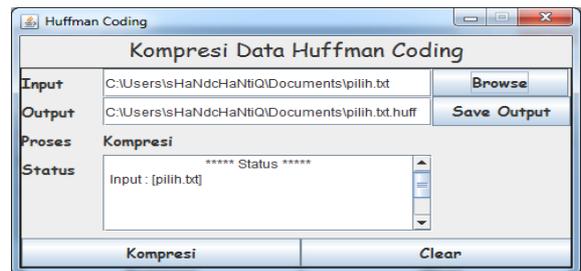


Gambar 15 Dialog Screen Tampilan Awal



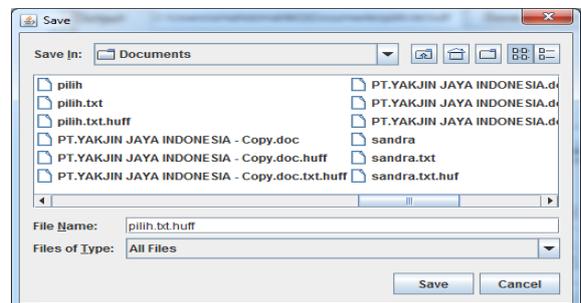
Gambar 15 Dialog Screen Browse

Tampilan tersebut akan muncul setelah menekan tombol Browse pada tampilan awal, lalu pilih text yang ingin di kompresi



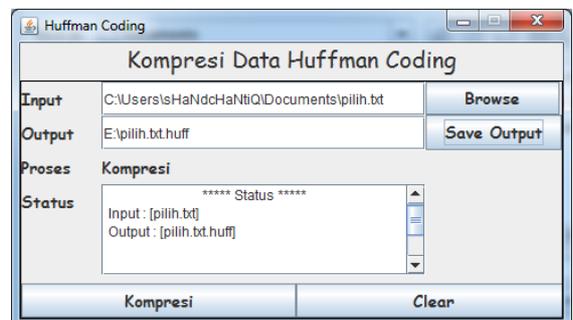
Gambar 15 Dialog Screen Suggest Output

Textarea Output akan otomatis terisi setelah file dipilih



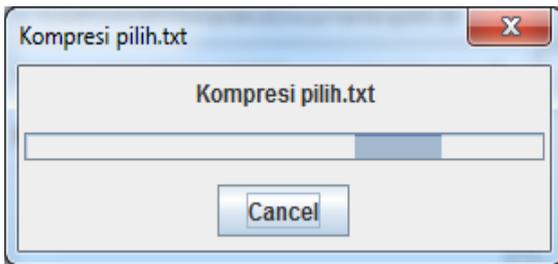
Gambar 15 Dialog Screen Save Output

Tampilan tersebut akan muncul setelah menekan tombol Save output pada tampilan awal untuk menentukan tempat file hasil kompresi akan disimpan



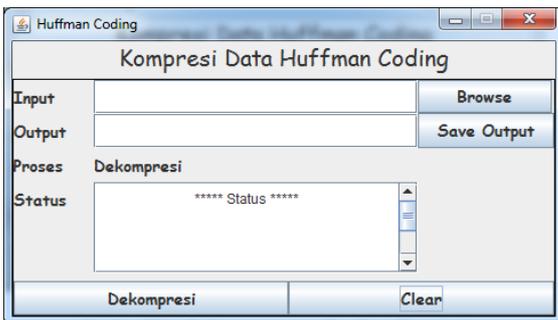
Gambar 15 Dialog Screen Kompresi

Tekan tombol Kompresi untuk memulai proses kompresi data sehingga muncul dialog seperti gambar berikut



Gambar 15 Dialog Screen Progress

Setelah tampilan dialog tersebut menghilang, maka proses pengkompresian data selesai



Gambar 15 Dialog Screen Dekompresi

Gambar berikut menunjukkan gambar untuk melakukan proses dekompresi.

4. Hasil Pengujian

Di bawah ini adalah hasil pengujian yang telah dilakukan pada perangkat lunak kompresi data text menggunakan algoritma Huffman

Jenis	Orginal File	After Compression	Ratio	Distinct char
Satu karakter	20 KB	1 KB	49%	1
Karakter berbeda frekuensi kemunculan sama	20 KB	9 KB	45,90%	9
Indonesia	20 KB	11 KB	58,60%	80
Inggris	20 KB	11 KB	54,90%	36

Gambar 16 Hasil pengujian

Setelah melalui beberapa tahap analisa, perancangan dan implementasi, maka diperoleh kesimpulan sebagai berikut :

1. Redundancy kemunculan karakter pada file ditangani algoritma Huffman dengan menggunakan binary tree sehingga karakter yang memiliki frekuensi kemunculan lebih banyak akan diwakili dengan codeword yang

lebih pendek, sedangkan karakter yang frekuensi kemunculannya sedikit akan diwakili dengan codeword yang lebih panjang. Sehingga jumlah bit berkurang secara signifikan.

2. Untuk karakter yang mempunyai frekuensi yang sama, file yang dihasilkan dari proses kompresi akan lebih besar dibandingkan dengan file hasil kompresi dengan satu karakter yang memiliki ukuran file asli yang sama, namun tidak menutup kemungkinan karakter yang mempunyai frekuensi yang sama file yang dihasilkan dari proses kompresi menjadi lebih besar dari file aslinya, dikarenakan algoritma Huffman melakukan proses kompresi dengan mendahulukan frekuensi terendah.

Saran atau masukan yang dapat kami berikan untuk menunjang atau pengembangan sistem selanjutnya, sebagai berikut:

1. Untuk meningkatkan rasio kompresi maka algoritma kompresi Huffman dapat digabungkan dengan rasio kompresi yang lain seperti LZW.
2. Untuk dapat lebih melihat dan membuktikan keefektifan dari algoritma Huffman, perlu diadakannya sebuah penelitian yang bertujuan membandingkan seluruh algoritma kompresi dalam mengkompresi berbagai data atau file.

DAFTAR PUSTAKA

1. Abu Zaid, N. H. (1995). *Teks Otoritas Kebenaran*. Yogyakarta: LKiS.
2. Linawati, & Panggabean, H. P. (2004, Maret). Perbandingan Kinerja Algoritma Kompresi Huffman, LZW, dan DMC pada beberapa tipe File. *Integral*, 9, 7-16.
3. Longkutoy, J. J. (1983). *Pengenalan Komputer*. Jakarta: Mutiara Offset.
4. Machali, R. (2009). *Pedoman bagi Penerjemah*. Bandung: Kaifa.
5. Munir, R. (2005). *Diktat Kuliah IF2251 Strategi Algoritmik*. Bandung: ITB.
6. Penyusun, T. (2008). *Kamus Bahasa Indonesia*. Jakarta: Pusat Bahasa.
7. Pu, I. M. (2005). *Fundamental Data Compression*. London: Butterworth-Heinemann.
8. Salomon, D. (2007). *Data Compression*. London: Springer.
9. Sayood, K. (2006). *Introduction to Data Compression*. San Francisco: Morgan Kaufman.