



PROGRAM STUDI

S1 SISTEM KOMPUTER

UNIVERSITAS DIPONEGORO

MULTIMEDIA

Kompresi

Oky Dwi Nurhayati, ST, MT
email: okydn@undip.ac.id

Definisi

- memampatkan/mengecilkan ukuran
- proses mengkodekan informasi menggunakan bit yang lain yang lebih rendah daripada representasi data yang tidak terkodekan dengan suatu sistem enkoding (penyandian) tertentu.

Aturan Kompresi

- Pengiriman data hasil kompresi dapat dilakukan jika pihak pengirim (yang melakukan kompresi) dan pihak penerima (yang melakukan dekompresi) memiliki aturan yang sama dalam hal kompresi data.
- Pihak pengirim harus menggunakan algoritma kompresi data yang sudah baku dan pihak penerima juga menggunakan teknik dekompresi data yang sama dengan pengirim sehingga data yang diterima dapat dibaca/didekode kembali dengan benar

Dasar-dasar teknik kompresi

- Sumber kompresi adalah lossy, yang berarti terjadi beberapa bagian komponen dari data yang hilang akibat dari proses kompresi.
- Kompresi entropy adalah lossless, yang berarti tidak ada data yang hilang selama proses kompresi (huffman coding).
- Hybrid merupakan kombinasi lossy dan lossless

- Kompresi entropy adalah media yg bebas mengambil secara urutan data per bit (byte stream)
- Huffman coding adalah contoh yg menggunakan teknik kompresi entropy.

- Entropy yang berasal dari sumber informasi menurut Shannon :
- $$H(S) = \eta = \sum_i p_i \log_2 \frac{1}{p_i}$$

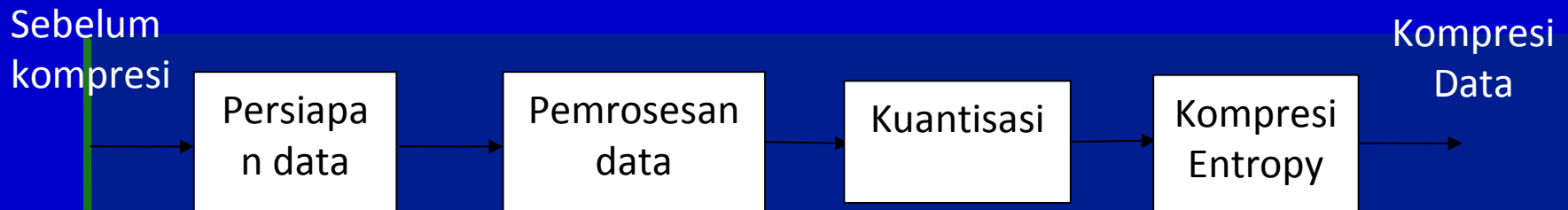
- Dimana p_i = nilai probabilitas S_i

$$\log_2 \frac{1}{p_i}$$

mengindikasikan adanya jumlah informasi yg terdapat di S_i , yaitu banyak bit yg diperlukan untuk dikodekan mjd S_i

contoh : tingkat intensitas gray, $p_i = 1/256$ shg jumlah bit yg diperlukan untuk mengkodekan tingkatan masing gray adalah 8 bit. Entropy dari gambar tersebut adalah 8 bit. 5

Diagram alir pemrosesan kompresi



Persiapan data : konversi analog ke digital, memberi pendekatan mengenai data yang akan dikompresi

Jika suatu daerah di dalam data akan dilakukan proses kompresi maka daerah tersebut harus diubah

Kuantisasi menetapkan pemetaan granularity dari angka sebenarnya seperti pecahan kedalam bilangan bulat

Kompresi entropy biasanya digunakan pada langkah terakhir

Teknik Entropy Encoding

- Run-length Encoding
- Repetition Suppression
- Pattern Substitution
- Huffman Coding

Keuntungan Kompresi

- Kompresi data menjadi sangat penting karena memperkecil kebutuhan penyimpanan data, mempercepat pengiriman data, memperkecil kebutuhan lebar-bidang (*bandwidth*).
- Teknik kompresi bisa dilakukan terhadap data teks/biner (zip), gambar (JPEG, PNG, TIFF), audio (MP3, AAC, RMA, WMA), dan video (MPEG, H261, H263).

http://en.wikipedia.org/wiki/Audio_codec

v · d · e		Data compression methods
Lossless compression methods	Theory	Entropy · Complexity · Redundancy
	Entropy encoding	Huffman · Adaptive Huffman · Arithmetic (Shannon-Fano · Range) · Golomb · Exp-Golomb · Universal (Elias · Fibonacci)
	Dictionary	RLE · LZ77/78 · LZW · LZWL · LZO · DEFLATE · LZMA · LZX
	Others	CTW · BWT · PPM · DMC
Audio compression methods	Theory	Convolution · Sampling · Nyquist–Shannon theorem
	Audio codec parts	LPC (LAR · LSP) · WLPC · CELP · ACELP · A-law · μ -law · MDCT · Fourier transform · Psychoacoustic model
	Others	Dynamic range compression · Speech compression · Sub-band coding
Image compression methods	Terms	Color space · Pixel · Chroma subsampling · Compression artifact
	Methods	RLE · Fractal · Wavelet · SPIHT · DCT · KLT
	Others	Bit rate · Test images · PSNR quality measure · Quantization
Video compression	Terms	Video Characteristics · Frame · Frame types · Video quality
	Video codec parts	Motion compensation · DCT · Quantization
	Others	Video codecs · Rate distortion theory (CBR · ABR · VBR)

Timeline of information theory, data compression, and error-correcting codes

See [Compression Formats and Standards](#) for formats and [Compression Software Implementations](#) for codecs

Jenis Kompresi

- Lossy Compression
(Kompresi Berugi)
- Lossless Compression
(Kompresi Tak-Berugi)

Lossy Compression

- Teknik kompresi dimana data hasil dekompresi tidak sama dengan data sebelum kompresi namun sudah “cukup” untuk digunakan.
- membuang bagian-bagian data yang tidak begitu berguna, tidak begitu dirasakan, tidak begitu dilihat oleh manusia → masih beranggapan bahwa data masih bisa digunakan.
- Kelebihan: ukuran file lebih kecil dibanding loseless namun masih tetap memenuhi syarat untuk digunakan.
- Contoh: MP3, streaming media, JPEG, MPEG, dan WMA
- Image asli berukuran 12,249 bytes, kompresi JPEG kualitas 30 dan berukuran 1,869 bytes → image 85% lebih kecil dan ratio kompresi 15%.

Lossless Compression

- Teknik kompresi dimana data hasil kompresi dapat didekompres lagi dan hasilnya tepat sama seperti data sebelum proses kompresi.
- Contoh aplikasi: ZIP, RAR, GZIP, 7-Zip, beberapa image seperti GIF dan PNG

Kriteria Kompresi

- Kualitas data hasil enkoding: ukuran lebih kecil, data tidak rusak untuk kompresi lossy.
- Kecepatan, ratio, dan efisiensi proses kompresi dan dekompresi
- Ketepatan proses dekompresi data: data hasil dekompresi tetap sama dengan data sebelum dikompres (kompresi loseless)

Teknik Kompresi

- Kompresi teks : RLE, kode huffman statik
- Kompresi gambar : JPEG, PNG, BMP, TIF, dan JPEG 2000
- Kompresi video : H261, H262, dan MPEG

RLE (Run-Length-Encoding)

- Kompresi data teks dilakukan jika ada beberapa huruf yang sama yang ditampilkan berturut-turut
 - Contoh ABCCCCCCCCDEFGGGG = 17 karakter
 - RLE tipe 1 (min. 4 huruf sama) : ABC!8DEFG!4 = 11 karakter

Run-length Encoding

- Beberapa elemen angka yang sama diwakilkan dengan satu buah elemen angka yang diberikan jumlahnya
- contoh:
111333322222211111 \Rightarrow (1,3),(3,4),(2,6),(1,5)
atau dengan contoh yang lain
10000001 dikompresi menjadi 10!61

Repetition Suppression

- Mengubah angka atau hurup yang berulang-ulang diwakilkan dengan satu hurup dengan jumlahnya
- Example:
98400000000000000000000000000000 = 984A24

Pattern Substitution

- Melakukan substitusi kata-kata menjadi huruf maupun simbol
- contoh:
This book is an exemplary example of a book....
This \Rightarrow 1 、 is \Rightarrow 2 、 an \Rightarrow 3 、 of \Rightarrow 4 、 a \Rightarrow 5
book \Rightarrow b*....

1b*23exemplary example45b*

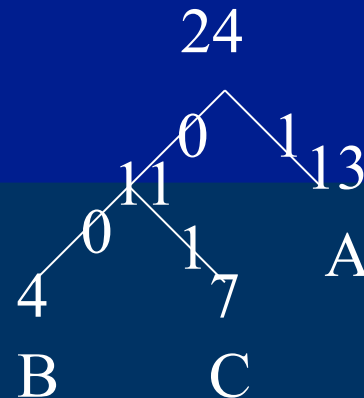
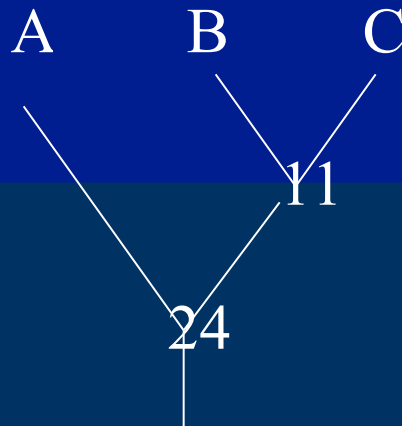
Pattern Substitution

- This book is an exemplary example of a book on multimedia and networking. Nowhere else will you find this kind of coverage and completeness. This is truly a one-stop-shop for all that you want to know about multimedia and networking.
- a:1, about:2, all:3, an:4, and:5, for:6, is:7, of:8, on:9, that:+, this:&, to:=, will:#
 - &b o o k 7 4 e x e m p l a r y s p e x a m p l e 8 1 b o o k 9 m * 5 n * . N o w h e r e s p e l s e # y o...
 - 129 : 193 = 0.6684 → 33.16% compression

Huffman Coding

- Berdasarkan frekuensi kejadian pada karakter yang diberikan

- A:13, B:4, C:7 \Rightarrow A \rightarrow 1, B \rightarrow 00, C \rightarrow 01



Pemampatan Data dengan Algoritma Huffman

Prinsip kode Huffman:

- karakter yang paling sering muncul di dalam data dengan kode yang lebih pendek;
- sedangkan karakter yang relatif jarang muncul dikodekan dengan kode yang lebih panjang.

Fixed-length code

Karakter	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Frekuensi	45%	13%	12%	16%	9%	5%
Kode	000	001	010	011	100	111

‘bad’ dikodekan sebagai ‘001000011’

Pengkodean 100.000 karakter
membutuhkan 300.000 bit.

Variable-length code (Huffman code)

Karakter	a	b	c	d	e	f
Frekuensi	45%	13%	12%	16%	9%	5%
Kode	0	101	100	111	1101	1100

'bad' dikodekan sebagai '1010111'

Pengkodean 100.000 karakter membutuhkan
 $(0,45 \times 1 + 0,13 \times 3 + 0,12 \times 3 + 0,16 \times 3 + 0,09 \times 4 + 0,05 \times 4) \times 100.000 = 224.000$ bit

Nisbah pemampatan:

$$(300.000 - 224.000) / 300.000 \times 100\% = 25,3\%$$

Algoritma Greedy untuk Membentuk Kode Huffman:

- § Baca semua karakter di dalam data untuk menghitung frekuensi kemunculan setiap karakter. Setiap karakter penyusun data dinyatakan sebagai pohon bersimpul tunggal. Setiap simpul di-*assign* dengan frekuensi kemunculan karakter tersebut.
- § Terapkan strategi *greedy* sebagai berikut: gabungkan dua buah pohon yang mempunyai frekuensi terkecil pada sebuah akar. Akar mempunyai frekuensi yang merupakan jumlah dari frekuensi dua buah pohon penyusunnya.
- 7. Ulangi langkah 2 sampai hanya tersisa satu buah pohon Huffman.

Kompleksitas algoritma Huffman: $O(n \log n)$ untuk n karakter.

- **Contoh:**

Karakter	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>

Frekuensi	45	13	12	16	9	5

1.

$f:5$

$e:9$

$c:12$

$b:13$

$d:16$

$a:45$

2.

$c:12$

$b:13$

$fe:14$

$d:16$

$a:45$

$f:5$

$e:9$

3.

$fe:14$

$d:16$

$cb:25$

$a:45$

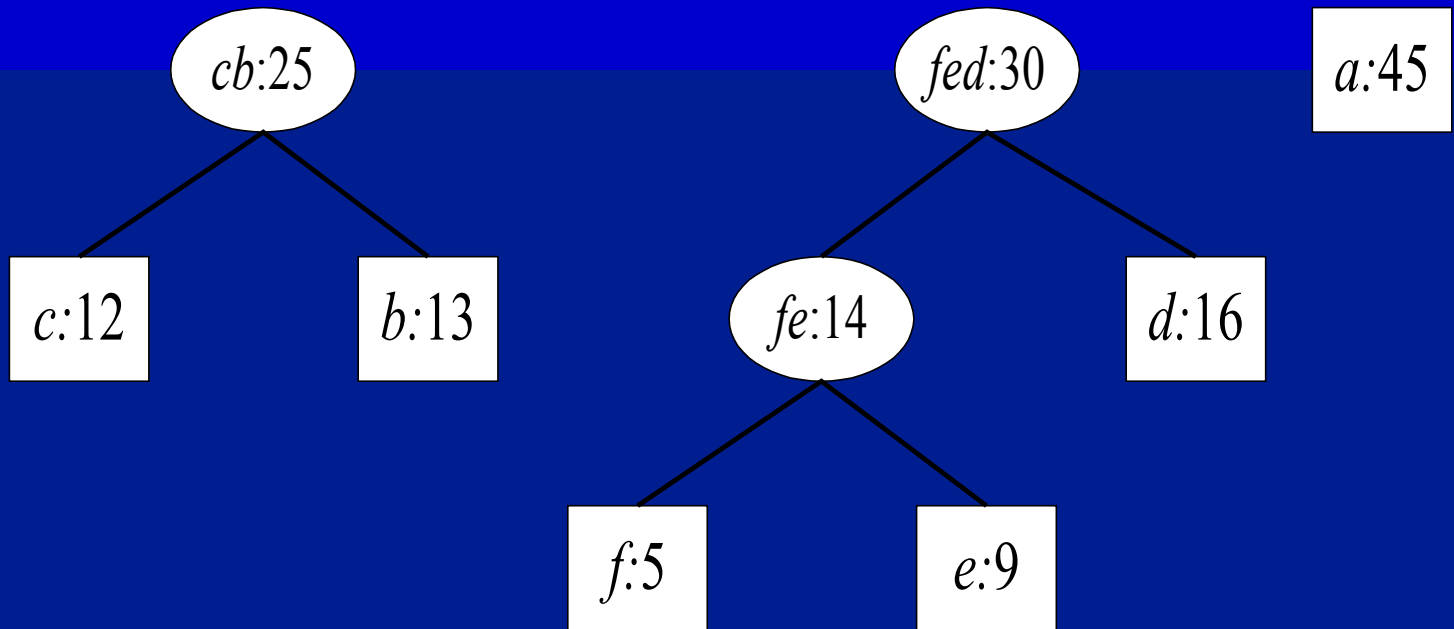
$f:5$

$e:9$

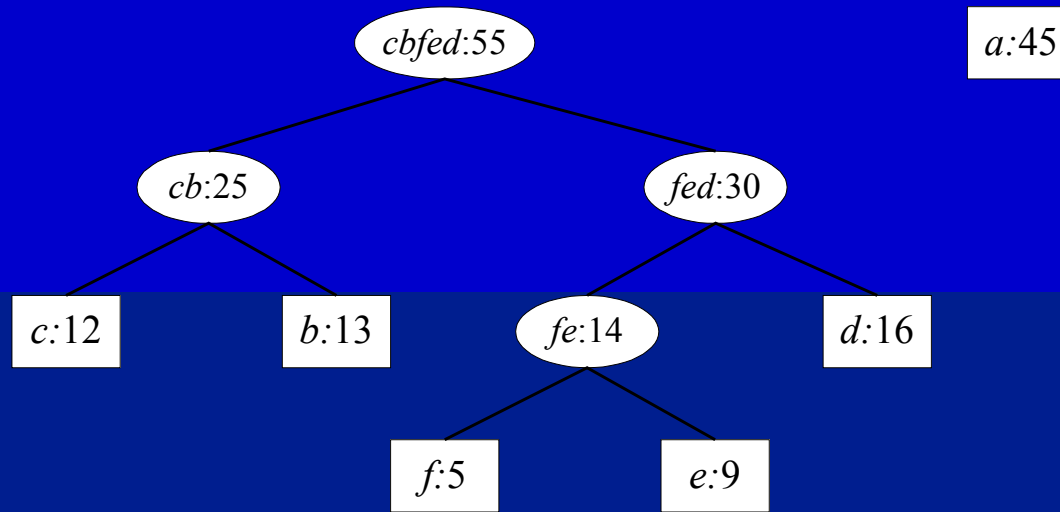
$c:12$

$b:13$

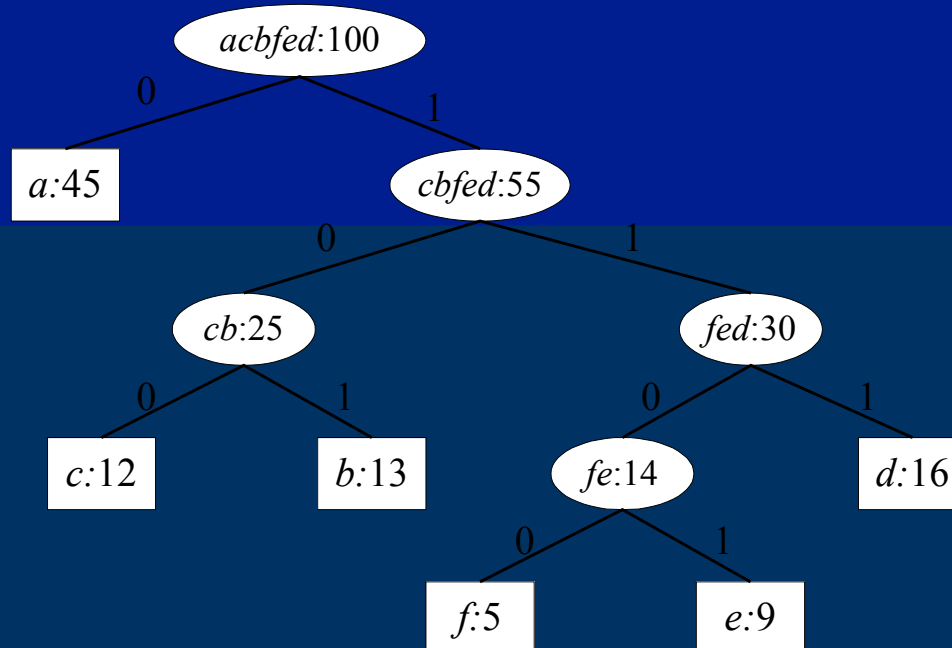
4.



5.



6



Huffman Coding

A 15 1100	B 3 0101011	C 2 0010011	D 7 01000	E 18 1011	F 4 0001100
G 3 0101000	H 6 101010	I 14 00111	J 0 00000111	K 6 101011	L 11 1000
M 7 00010	N 16 1111	O 21 1110	P 5 0100111	Q 0 000011000	R 7 10100
S 10 0011	T 15 001000	U 6 0000111	V 1 0000111	W 6 110101	X 2 0100100
Y 4 0000011	Z 0 0000100	. 3 0000100	Sp 39 0111		

Static Huffman Coding

- Frekuensi karakter dari string yang akan dikompres dianalisis terlebih dahulu. Selanjutnya dibuat pohon huffman yang merupakan pohon biner dengan **root awal** yang diberi nilai 0 (sebelah kiri) atau 1 (sebelah kanan), sedangkan selanjutnya untuk dahan **kiri** selalu diberi nilai 1(kiri)
- 0(kanan) dan di dahan **kanan** diberi nilai 0(kiri) – 1(kanan)
- A bottom-up approach = frekuensi terkecil dikerjakan terlebih dahulu dan diletakkan ke dalam leaf(daun).
- Kemudian leaf-leaf akan dikombinasikan dan dijumlahkan probabilitasnya menjadi root di atasnya.

Proses Encoding

Encoding adalah cara menyusun *string biner* dari teks yang ada. Proses *encoding* untuk satu karakter dimulai dengan membuat pohon Huffman terlebih dahulu. Setelah itu, kode untuk satu karakter dibuat dengan menyusun nama *string biner* yang dibaca dari akar sampai ke daun pohon Huffman.

Langkah-langkah untuk men-*encoding* suatu *string biner* adalah sebagai berikut :

1. Tentukan karakter yang akan di-*encoding*
2. Mulai dari akar, baca setiap bit yang ada pada cabang yang bersesuaian sampai ketemu daun dimana karakter itu berada
3. Ulangi langkah 2 sampai seluruh karakter di-*encoding*

LZW (Lempel-Ziv-Welch)

- Menggunakan teknik adaptif dan berbasiskan “kamus”.
- Pendahulu LZW adalah LZ77 dan LZ78 yang dikembangkan oleh Jacob Ziv dan Abraham Lempel pada tahun 1977 dan 1978.
- Terry Welch mengembangkan teknik tersebut pada tahun 1984.
- LZW banyak dipergunakan pada UNIX, GIF, V.42 untuk modem

Algoritma Kompresi LZW

```
BEGIN
  s = next input character;
  while not EOF
  {
    c = next input character;
    if s + c exists in the dictionary
      s = s + c
    else
    {
      Output the code for s;
      Add string s + c to the dictionary with a new code
      s = c;
    }
  }
END
```

Contoh Kompresi LZW

Code	String
1	A
2	B
3	C

Data : ABABBABCABABBA

s	c	output	code	string
			1	A
			2	B
			3	C
A	B	1	4	AB
B	A	2	5	BA
A	B			
AB	B	4	6	ABB
B	A			
BA	B	5	7	BAB
B	C	2	8	BC
C	A	3	9	CA
A	B			
AB	A	4	10	ABA
A	B			
AB	B			
ABB	A	6	11	ABBA
A	EOF	1		

Algoritma Dekompresi

```
BEGIN
  S = NULL;
  while not EOF{
    K = NEXT INPUT CODE;
    Entry = dictionary entry for K;
    Ouput entry;
    if(s != NULL)
      add string s + entry[0] to dictionary with new code
    S = Entry;
  }
END
```

Dekompresi : 124523461

S	K	Entry/output	Code	String
			1	A
			2	B
			3	C
NULL	1	A		
A	2	B	4	AB
B	4	AB	5	BA
AB	5	BA	6	ABB
BA	2	B	7	BAB
B	3	C	8	BC
C	4	AB	9	CA
AB	6	ABB	10	ABA
ABB	1	A	11	ABBA
A	EOF			

Output : **ABABBABCABABBA**

Aplikasi Kompresi

- ZIP File Format
 - Ditemukan oleh Phil Katz untuk program PKZIP kemudian dikembangkan untuk WinZip, WinRAR, 7-Zip.
 - Berekstensi *.zip dan MIME application/zip
 - Dapat menggabungkan dan mengkompresi beberapa file sekaligus menggunakan bermacam-macam algoritma.

Aplikasi Kompresi

- Method Zip
 - Shrinking : merupakan metode variasi dari LZW
 - Reducing : merupakan metode yang mengkombinasikan metode same byte sequence based dan probability based encoding.
 - Imploding : menggunakan metode byte sequence based dan Shannon-Fano encoding.
 - Deflate : menggunakan LZW

Aplikasi Kompresi

- Oleh Eugene Roshal, pada 10 Maret 1972 di Rusia
- **RAR** → Roshal Archive.
- Berekstensi .rar dan MIME (Multipurpose Internet Mail Extensions-MIME) application/x-rar-compressed
- Proses kompresi lebih lambat dari ZIP tapi ukuran file hasil kompresi lebih kecil.
- Aplikasi: WinRAR yang mampu menangani RAR dan ZIP, mendukung volume split, enkripsi AES.

Latihan

2. Tuliskan bilangan biner nama kamu yang di hasilkan dari proses pengkodean dengan huffman coding dan berpa faktor kompresinya dibandingkan dengan format ASCII
3. Tuliskan nilai dekompresi dari pengkodean kompresi:
 - 73d5f3
 - 83!5a!9B