[TTG4J3] KODING DAN KOMPRESI

Lesson 5
Huffman Coding

Oleh :
Ledya Novamizanti
Astri Novianty

Prodi S1 Teknik Telekomunikasi
Fakultas Teknik Elektro
Universitas Telkom

Huffman Coding

- Optimal code pertama yang dikembangkan oleh David Huffman pada tahun 1952, sebagai bagian dari tugas kelas ketika menempuh pendidikan Ph.D di MIT
- Kelas tersebut yang pertama kalinya di bidang teori informasi dan diajarkan oleh Robert Fano
- Kode Huffman merupakan binary prefix code yang optimum
- Pembentukan diagram pohon dari Huffman coding dimulai dari daun ke akar (bottom-top)

Huffman Coding

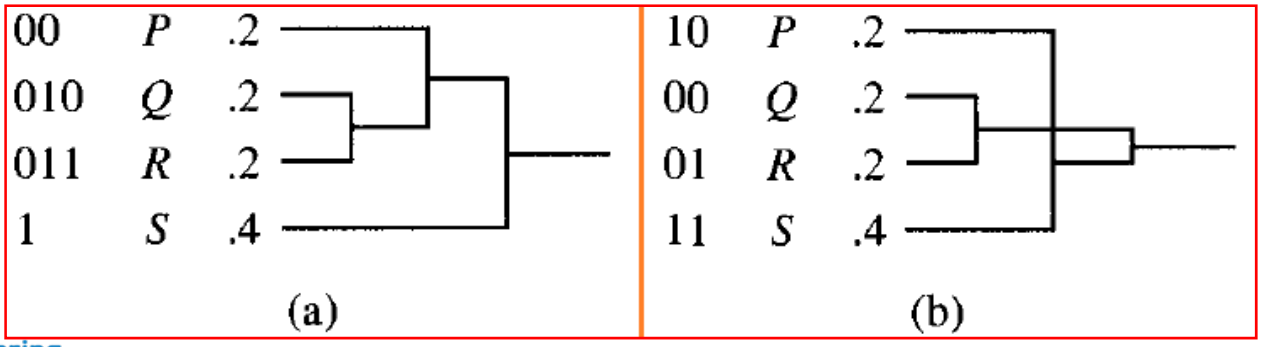
Untuk source $S = \{x_1, \ldots, x_n\}$; probabilitas $P = \{p_1, \ldots, p_n\}$; codewords $\{c_1, \ldots, c_n\}$ dengan panjang $\{l_1, \ldots, l_n\}$, terdapat optimal binary prefix code dengan karakteristik :

Teorema :

a) Jika $p_j > p_i$, *maka* $l_j \leq l_i$
b) Dua codeword dari dua simbol dg probabilitas terendah mempunyai panjang yg sama
c) Dua codeword terpanjang identik kecuali pada digit terakhir

Huffman Coding

- Kode Huffman tidak unik, bergantung pada :
  - Aturan (*rule*) ascending/descending order yang digunakan di dalam algoritmanya
  - Bagaimana memilih probabilitas terendah saat membangun diagram pohon Huffman (Huffman tree)
- Aturan tersebut bersifat opsional, tetapi harus konsisten diterapkan
- Panjang rata-rata codeword selalu sama untuk tree berbeda



Algoritma Standard Huffman Coding

1. Urutkan simbol/karakter berdasarkan probabilitasnya
   - Jika probabilitas sama, urutkan simbol berdasarkan indeks simbol tersebut
2. Ambil dua simbol dengan probabilitas terkecil, gabungkan menjadi simbol baru, dan jumlahkan probabilitasnya
3. Urutkan kembali simbol-simbol seperti langkah 1 dengan menyertakan simbol baru yang diperoleh dari langkah 2
   - Simbol baru ditempatkan **di bawah/ di kiri** simbol lama jika probabilitasnya sama
4. Ulangi langkah 2 dan 3 hingga diperoleh jumlah probabilitas = 1.0
5. Tentukan codewords setiap simbol dengan penelusuran bit

Representasi Huffman Coding

- Dapat menggunakan representasi diagram panah, atau pohon biner
- Probabilitas dan indeks huruf dapat diurut menaik (ascending) atau pun menurun (descending), tetapi harus konsisten di salah satu pilihan saat melakukan coding
- Di kelas ini, akan digunakan skema descending untuk skema diagram panah, dan ascending untuk pohon biner

Aturan Diagram Panah

- Simbol diurut berdasarkan probabilitasnya secara descending dari atas ke bawah
- Jika probabilitas sama maka :
  - Simbol diurut berdasarkan indeks karakter secara descending dari atas ke bawah
  - Simbol baru ditempatkan **di bawah** simbol lama
- Level atas diberi bit '1', level bawah diberi bit '0'

Aturan Pohon Biner

- Simbol diurut berdasarkan probabilitasnya secara ascending dari kiri ke kanan
- Jika probabilitas sama maka :
  - Simbol diurut berdasarkan indeks karakter secara ascending dari kiri ke kanan
  - Simbol baru ditempatkan **di kiri** simbol lama
- Pembentukan pohon biner dimulai dari daun ke akar
- Anak kiri diberi bit '0', anak kanan diberi bit '1',

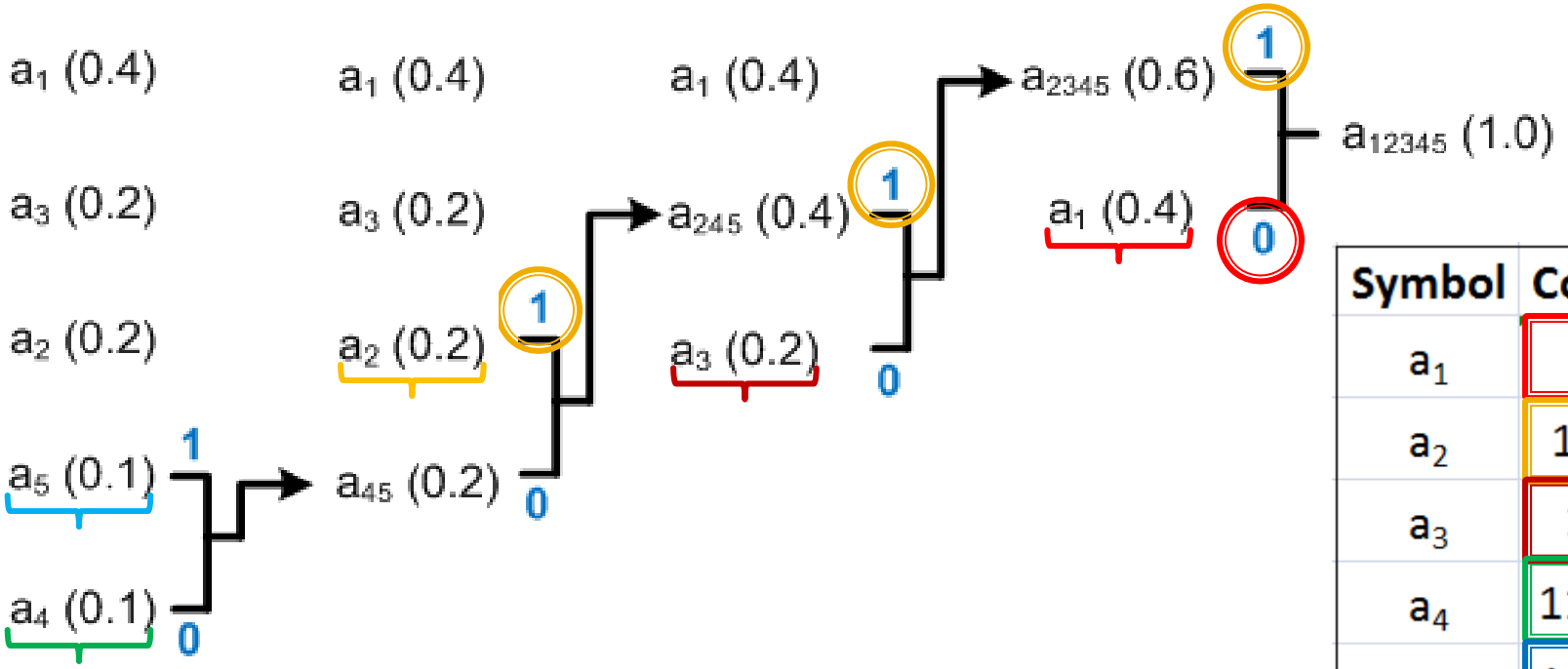
Contoh 1

Rancanglah kode Huffman untuk A = {a_1, a_2, a_3, a_4, a_5}, dengan probabilitas kemunculan P(a_1) = 0.4, P(a_2)=P(a_3) = 0.2, P(a_4) = P(a_5) = 0.1 menggunakan representasi :

a. Diagram panah
b. Pohon Biner

Solusi Contoh 1(a)

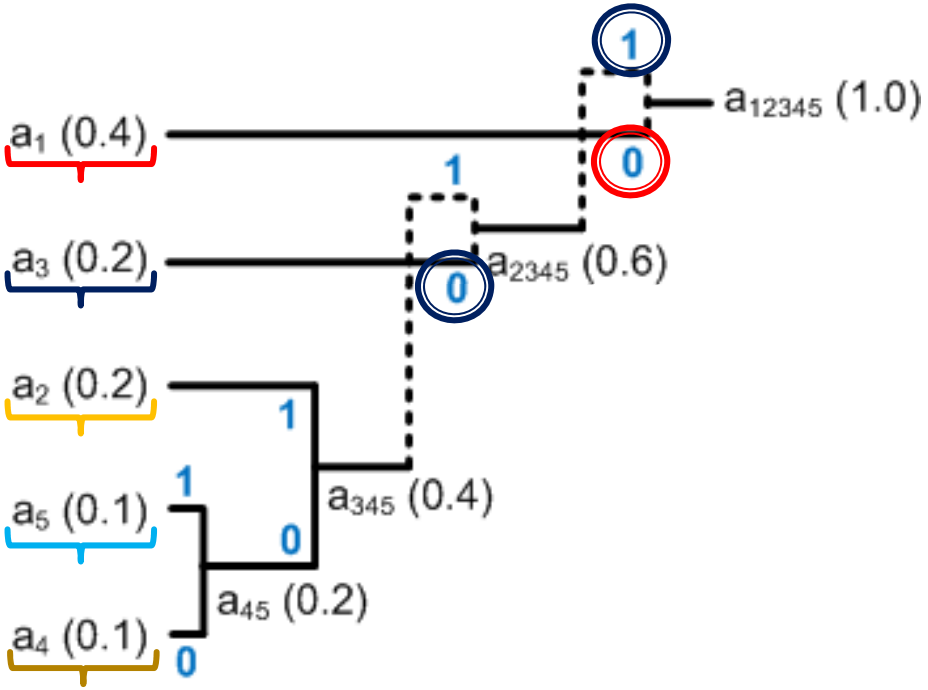
a. Pencarian Huffman code menggunakan diagram panah (cara 1)



Symbol	Code
a_1	0
a_2	111
a_3	10
a_4	1100
a_5	1101

Solusi Contoh 1(a)

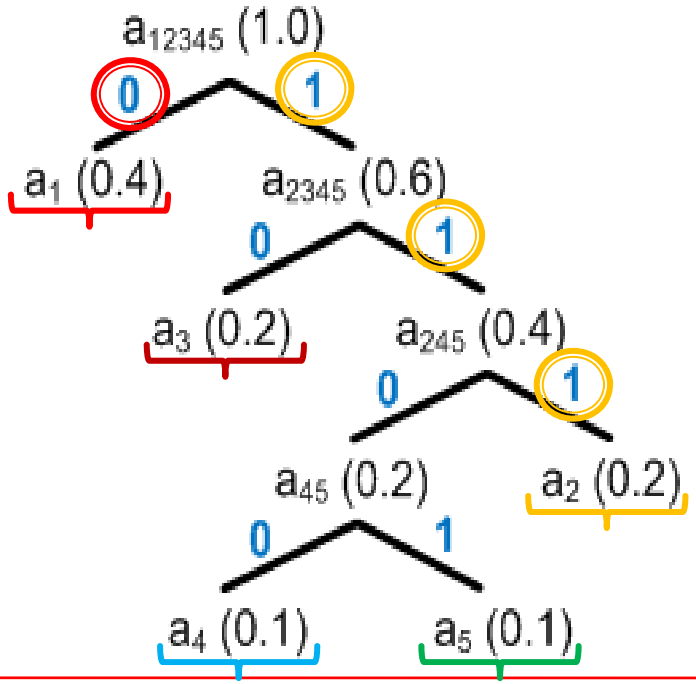
a. Pencarian Huffman code menggunakan diagram panah (cara 2)



Symbol	Code
a_1	0
a_2	111
a_3	10
a_4	1100
a_5	1101

Solusi Contoh 1(b)

b. Pencarian Huffman code menggunakan pohon biner



Symbol	Code
a_1	0
a_2	111
a_3	10
a_4	1100
a_5	1101

Contoh 2

Dari kode Huffman pada Contoh 1, hitung :

Symbol	Probability	Code
a_1	0.40	0
a_2	0.20	111
a_3	0.20	10
a_4	0.10	1100
a_5	0.10	1101

a. Average length
b. Entropy
c. Redundancy
d. Efficiency
e. Variansi kode

Solusi Contoh 2

a. Average length $L = \sum P(S_i) \times n(S_i)$

L = 0.4×1 + 0.2×2 + 0.2×3 + 0.1×4 + 0.1×4
= 2.2 bits/symbol

Symbol	Probability	Code
a_1	0.40	0
a_2	0.20	111
a_3	0.20	10
a_4	0.10	1100
a_5	0.10	1101

b. Entropy $H = -\sum P(S_i) \times log_b P(S_i)$

H = - (0.4 $\log_2$ 0.4 + 2 × 0.2 $\log_2$ 0.2 + 2 × 0.1 $\log_2$ 0.1 ≈ 2.12 bits/symbol

c. Redundancy, R=L-H=2.2 - 2.122 = 0.08 bits/symbol
(3.68% lebih banyak dari entropy)

Solusi Contoh 2

Average Length (L) 2.2 bits/symbol
dan Entropy (H) 2.12 bits/symbol

Symbol	Probability	Code
a_1	0.40	0
a_2	0.20	111
a_3	0.20	10
a_4	0.10	1100
a_5	0.10	1101

d. Efficiency $efficiency = \frac{H}{L} \times 100\%$

$$efficiency = \frac{2.12}{2.2} \times 100\% \approx 96.36\%$$

e. Variansi $V = \sigma^2 = E(a_i - A)^2 = \frac{1}{n}\sum (a_i - A)^2$

$$V = 0.4(1 - 2.2)^2 + 0.2(2 - 2.2)^2 + 0.2(3 - 2.2)^2 + 0.1(4 - 2.2)^2 + 0.1(4 - 2.2)^2 = 1.36$$

Tugas 1

Diketahui pesan string **ada ada saja**

a. Tentukan kode Huffman dari string tersebut
b. Hitung average length dan entropy
c. Hitung redundancy kode
d. Hitung variansi kode
e. Hitung efficiency
f. Hitung rasio kompresi

Minimum Variance Huffman Coding

- Kode Huffman dengan variansi panjang codewords minimum, sehingga lebih optimal dalam proses transmisi data
- Average length pada MVHC (Minimum Variance Huffman Coding) sama dengan average length SHC (Standard Huffman Coding)

Algoritma Minimum Variance HC

1. Urutkan simbol/karakter berdasarkan probabilitasnya
   - Jika probabilitas sama, urutkan simbol berdasarkan indeks simbol tersebut
2. Ambil dua simbol dengan probabilitas terkecil, gabungkan menjadi simbol baru, dan jumlahkan probabilitasnya
3. Urutkan kembali simbol-simbol seperti langkah 1 dengan menyertakan simbol baru yang diperoleh dari langkah 2
   - Simbol baru ditempatkan **di atas/ di kanan** simbol lama jika probabilitasnya sama
4. Ulangi langkah 2 dan 3 hingga diperoleh jumlah probabilitas = 1
5. Tentukan codewords setiap simbol dengan penelusuran bit

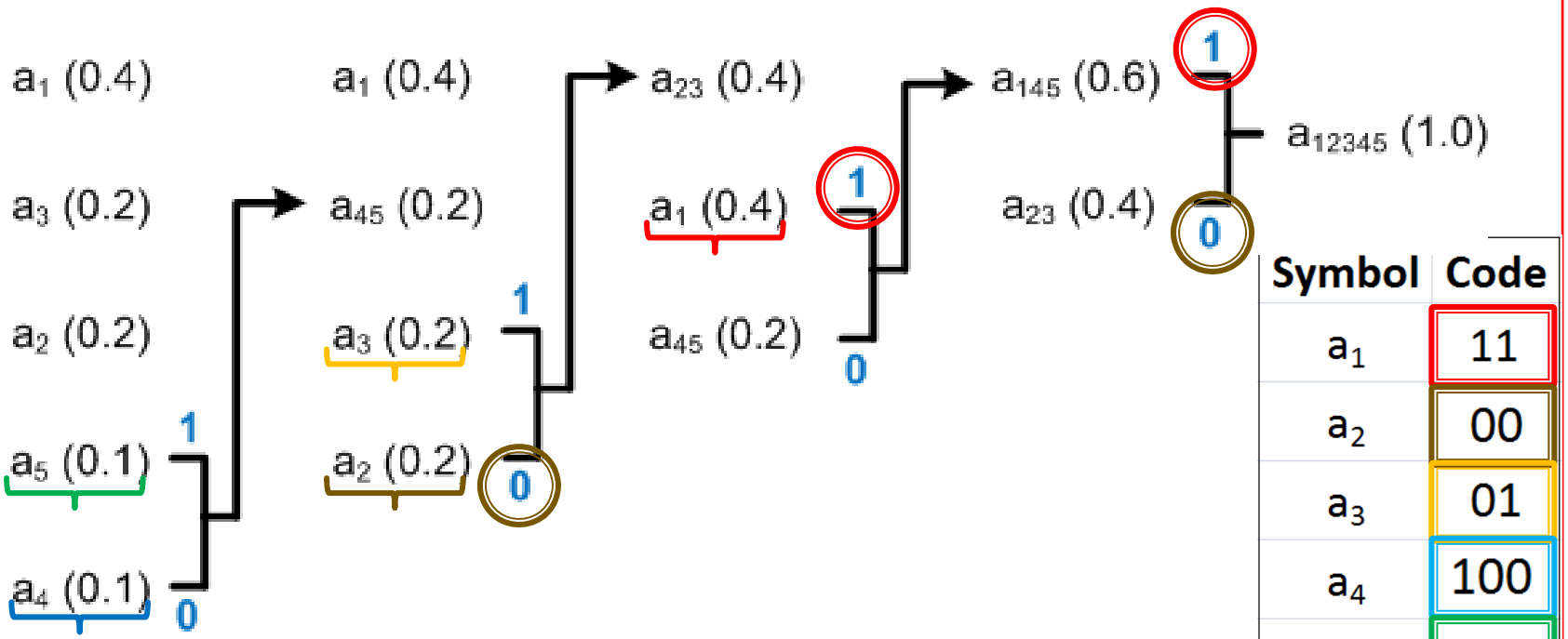
Contoh 3

Rancanglah kode Minimum Variance Huffman untuk $A = \{a_1, a_2, a_3, a_4, a_5\}$, dengan probabilitas kemunculan $P(a_1) = 0.4$, $P(a_2)=P(a_3) = 0.2$, $P(a_4)=P(a_5)= 0.1$ menggunakan representasi :

a. Diagram panah
b. Pohon Biner

Solusi Contoh 3(a)

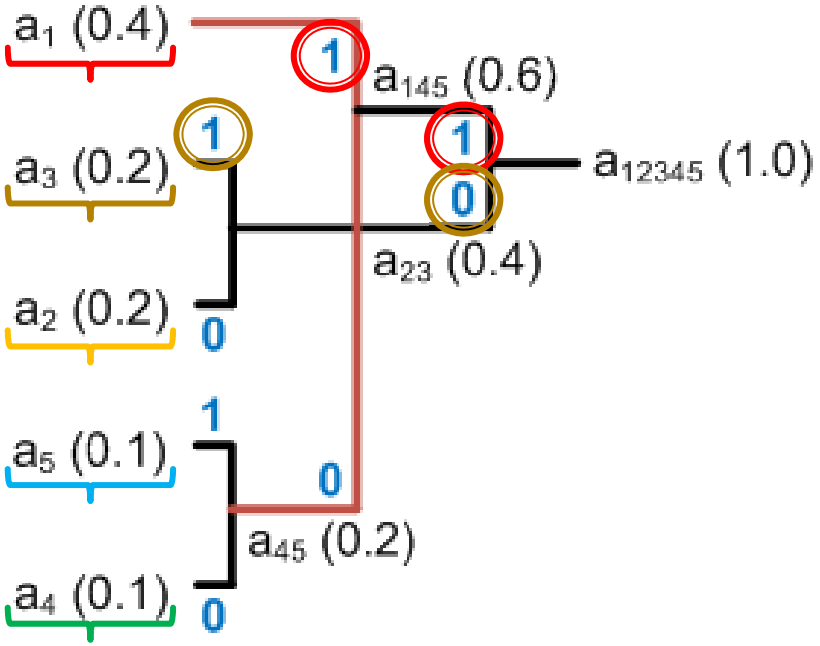
a. Pencarian Minimum Variance Huffman Code menggunakan diagram panah (cara 1)



Symbol	Code
a_1	11
a_2	00
a_3	01
a_4	100
a_5	101

Solusi Contoh 3(a)

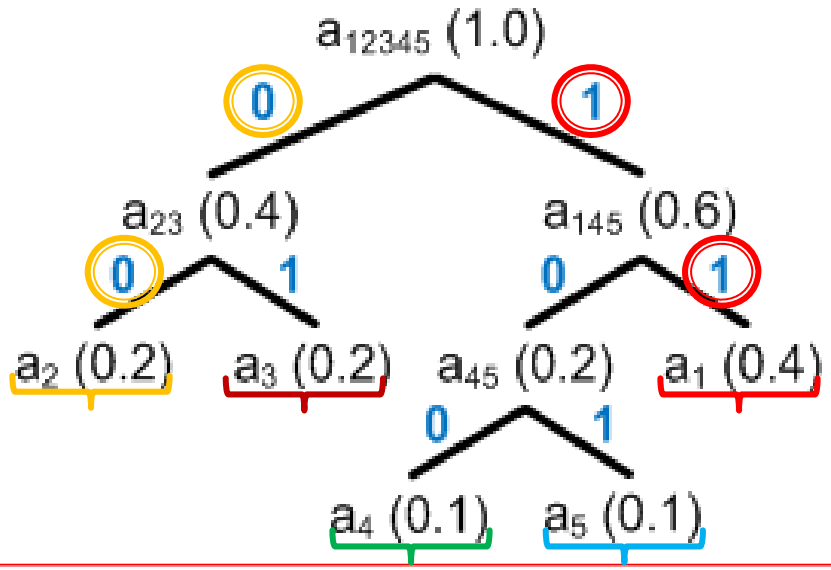
a. Pencarian Minimum Variance Huffman Code menggunakan diagram panah (cara 2)



Symbol	Code
a_1	11
a_2	00
a_3	01
a_4	100
a_5	101

Solusi Contoh 3(b)

b. Pencarian Minimum Variance Huffman Code menggunakan pohon biner



Symbol	Code
a_1	11
a_2	00
a_3	01
a_4	100
a_5	101

Contoh 4

Dari kode Minimum Variance Huffman pada Contoh 3, hitung :

Symbol	Probability	Code
a_1	0.40	11
a_2	0.20	00
a_3	0.20	01
a_4	0.10	100
a_5	0.10	101

a. Average length
b. Entropy
c. Redundancy
d. Efficiency
e. Variansi

Solusi Contoh 4

a. Average length $L = \sum P(S_i) \times n(S_i)$

L = 0.4×2 + 2×(0.2×2) + 2×(0.1×3) = 2.2 bits/symbol

b. Entropy $H = -\sum P(S_i) \times log_b P(S_i)$

H = - (0.4 $\log_2$ 0.4 + 2 × 0.2 $\log_2$ 0.2 + 2 × 0.1 $\log_2$ 0.1 ≈ 2.12 bits/symbol

c. Redundancy, R=L-H=2.2 - 2.122 = 0.08 bits/symbol

Symbol	Probability	Code
a_1	0.40	11
a_2	0.20	00
a_3	0.20	01
a_4	0.10	100
a_5	0.10	101

Average length pada MVHC sama dengan average length pada SHC (lihat hal 16)

Solusi Contoh 4

Average Length (L) 2.2 bits/symbol
dan Entropy (H) 2.12 bits/symbol

Symbol	Probability	Code
a_1	0.40	11
a_2	0.20	00
a_3	0.20	01
a_4	0.10	100
a_5	0.10	101

d. Efficiency $efficiency = \frac{H}{L} \times 100\%$

$$efficiency = \frac{2.12}{2.2} \times 100\% \approx 96.36\%$$

e. Variansi $V = \sigma^2 = E(a_i - A)^2 = \frac{1}{n}\sum (a_i - A)^2$

$$V = 0.4(2-2.2)^2 + 0.2(2-2.2)^2 + 0.2(2-2.2)^2 + 0.1(3-2.2)^2 + 0.1(3-2.2)^2 = 0.36$$

Pada kasus yang sama, menggunakan Huffman Coding, dihasilkan variansi 1.36 (lihat hal 17)

Tugas 2

Terdapat string **mana makan malam** (asumsi indeks spasi lebih dulu dari huruf 'a')

a. Cari kode Huffman nya (menggunakan skema panah dan pohon biner)
b. Hitung average length dan redundancy nya
c. Cari Minimum Variance Huffman Coding nya
d. Hitung rasio kompresi dan variansi kode menggunakan SHC dan MVHC

Catatan..

- Constraint pada Kode Huffman adalah panjang codeword (banyaknya bit/codeword), bukan nilai bit-nya ('0' atau '1')
- Sangat mungkin diperoleh representasi kode Huffman yang berbeda jika kesepakatan pada algoritma diubah (asal konsisten)
  - Misal: setelah simbol diurutkan, simbol atas diberi bit '0', simbol bawah bit '1'

Extended Huffman Code

- Jika jumlah simbol pada source-nya kecil, dan probabilitas kemunculan antara simbol sangat timpang, maka nilai probabilitas maksimum (p_{max}) bisa sangat besar dan kode huffman yang dihasilkan menjadi tidak efisien
- Alternatif solusi: Extended Huffman Coding

Extended Huffman Code

- Pada extended huffman coding, sebuah codeword tidak merepresentasikan satu simbol, melainkan sekumpulan simbol (lebih dari 1) atau block of symbols
- Bertujuan untuk mendapatkan nilai average length yang mendekati entropy
- Misal m=jumlah simbol pada source, n=jumlah rangkaian pada tiap simbol. Maka diperoleh simbol baru sebanyak m^n, yang akan dicari codewordnya

Contoh 5

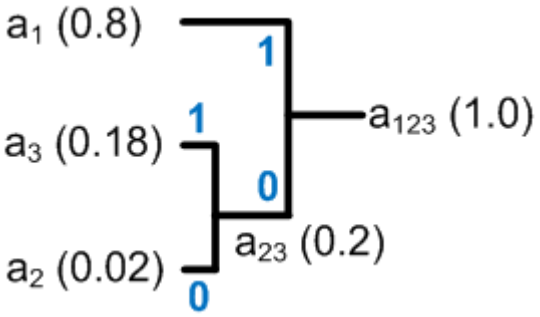
Diketahui sebuah source data A = {a_1, a_2, a_3}, dengan $P(a_1)$=0.8, $P(a_2)$ = 0.02, $P(a_3)$ = 0.18. Entropy source = 0.816 bits/symbol.

a. Rancanglah kode Huffman dari source data
b. Hitung redundancy dari kode Huffman
c. Rancanglah kode extended Huffman dari source data
d. Hitung redundancy dari kode extended Huffman

Solusi Contoh 5

Diketahui sebuah source data A = {a_1, a_2, a_3}, dengan $P(a_1)$=0.8, $P(a_2)$ = 0.02, $P(a_3)$ = 0.18. Entropy source = 0.816 bits/symbol.

- Huffman code yang dihasilkan:



Symbol	Code
a_1	1
a_2	00
a_3	01

- Average length $L = \sum P(S_i) \times n(S_i)$

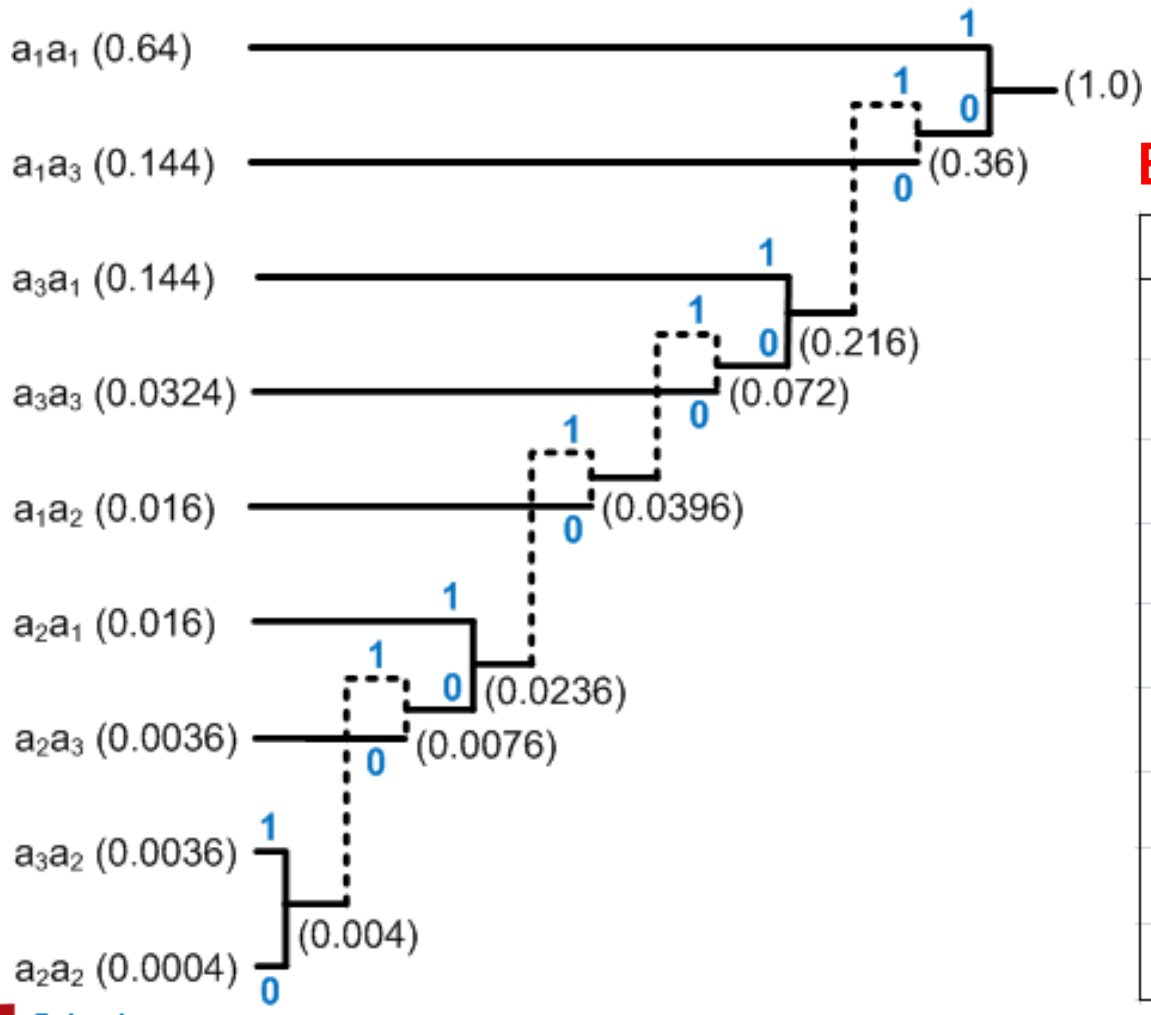
  L = 0.8×1 + 0.2×2 = 1.2 bits/symbol

- Redundancy = L – H = 0.384 bits/symbol (**47% dari entropy**)

Solusi Contoh 5

- A = {a_1, a_2, a_3}, dengan $P(a_1) = 0.8$, $P(a_2) = 0.02$, $P(a_3) = 0.18$.
- Dapat dibuat codeword yang merepresentasikan rangkaian 2 simbol
- Karena banyaknya simbol pada source, m=3, maka akan diperoleh $3^2 = 9$ simbol baru yang akan dicari codewordsnya

Solusi Contoh 5



Extended Huffman Code :

Symbol	Probability	Code
a_1a_1	0.64	1
a_1a_2	0.016	01010
a_1a_3	0.144	00
a_2a_1	0.016	010111
a_2a_2	0.0004	01011010
a_2a_3	0.0036	0101100
a_3a_1	0.144	011
a_3a_2	0.0036	01011011
a_3a_3	0.0324	0100

Solusi Contoh 5

Average length $L = \sum P(S_i) \times n(S_i)$

L = 0.64×1 + 0.016×5 + 0.144×2 + 0.016×6 + 0.0004×8 + 0.0036×7 + 0.144×3 + 0.0036×8 + 0.0324×4
= 1.7228 bits/symbol

Setiap simbol dalam extended, bersesuaian dengan 2 simbol dari simbol aslinya.
Sehingga average length dari kode aslinya,
L = 1.7228 /2 = 0.8614 bits/symbol

Symbol	Probability	Code
a_1a_1	0.64	1
a_1a_2	0.016	01010
a_1a_3	0.144	00
a_2a_1	0.016	010111
a_2a_2	0.0004	01011010
a_2a_3	0.0036	0101100
a_3a_1	0.144	011
a_3a_2	0.0036	01011011
a_3a_3	0.0324	0100

Solusi Contoh 5

Average length Extended HC = 0.8614 bits/symbol

Entropy source = 0.816 bits/symbol

Maka :

Redundancy = L – H= 0.8614 – 0.816 = 0.045 bits/symbol (**5.5% dari entropy**)

Bandingkan dengan Redundancy SHC/MVHC 0.384 bits/symbol (**47% dari entropy**)

Referensi

1. Adam Drozdek, *Elements of Data Compression*, Thomson Brooks/Cole, 2002
2. Khalid Sayood, *Introduction to Data Compression*, Academic Press, 2000.
3. T.M. Cover, J.A. Thomas, *Elements of Information Theory*, John Wiley&Sons.
4. M. Nelson and J.-L. Gailly. *The Data Compression Book. M&T Books, CA, 1996.*
5. D. Salomon. *Data Compression: The Complete Reference. Springer, 1998.*

Thank you ☺