Pohon (*Tree*)

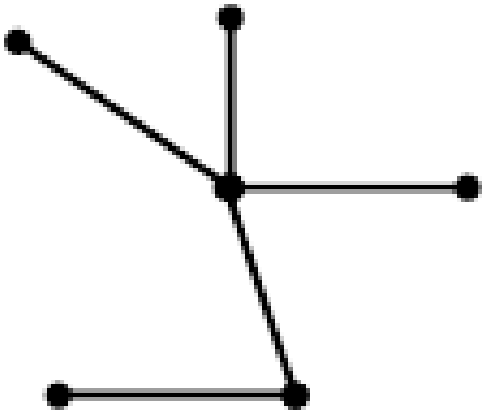
Anindita Septiarini

- ***Tree (Pohon) :*** merupakan suatu graf tak berarah yang terhubung dan tidak mengandung sirkuit.
- ***Forest (Hutan) :*** merupakan himpunan tree yang terpisan satu sama lain.
- ***Leaf (Daun) :*** suatu simpul terminal yang merupakan vertek berderajat 1 dalam sebuah tree.
- ***Branch (Cabang) :*** merupakan vertek berderajat > 1 dalam sebuah tree.

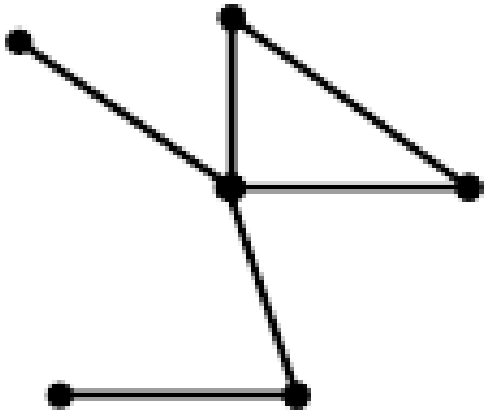
Gambarkan Tree pada problem berikut :

- Suatu komplek perumahan terdiri dari 15 keluarga. Suatu saat ketua RT ingin memberitahukan suatu berita pada seluruh warga komplek tersebut. Misal semua keluarga mempunyai telepon dan berita tersebut disebarluaskan dengan cara berikut :
  - Ketua RT hanya menelpon 2 keluarga, kemudian masing-masing keluarga yang telah ditelpon menelpon 2 keluarga lainnya yang belum ditelpon demikian seterusnya sampai semua keluarga menerima berita tersebut.

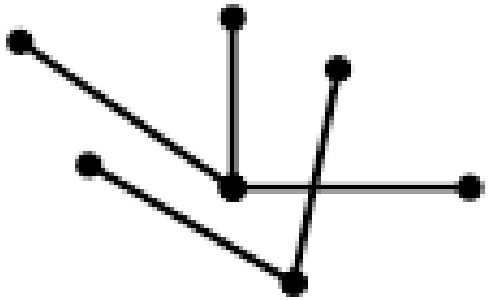
Pohon dan Bukan Pohon



A. Pohon



B. Bukan Pohon



C. Bukan Pohon

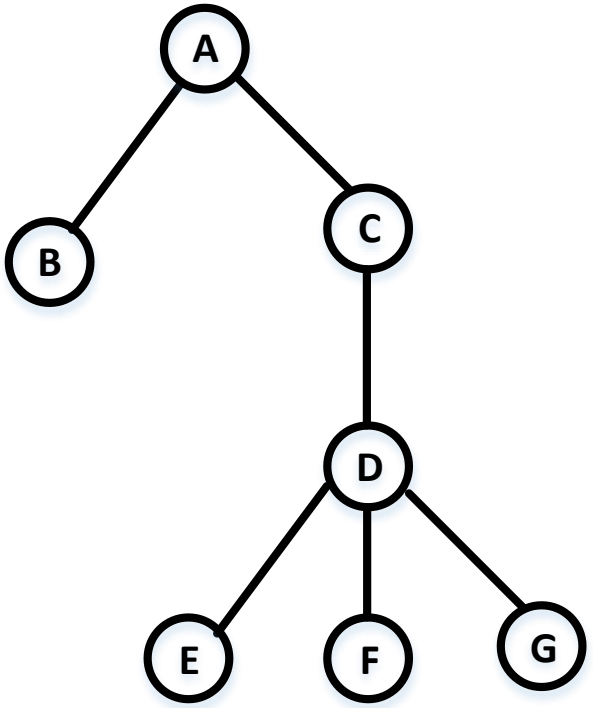
Sifat-sifat Tree :

- Ada sebuah path tunggal antara 2 vertek.
- Dalam sebuah pohon jumlah vertek satu lebih banyak daripada jumlah rusuk.
- Tree dengan lebih dari 2 vertek mempunyai daun ≥ 2.
- **Pohon Berakar :** pohon berarah yang mempunyai tepat 1 vertek yang mempunyai din = 0 dan semua vertek lainnya mempunyai din = 1.
  - Vertek dengan din = 0 disebut AKAR pohon tersebut.

Pada pohon berakar :

- Daun = vertek yang mempunyai dout = 0.
- Cabang = vertek yang mempunyai dout ≠ 0.
- Tinggi Pohon = panjang lintasan maksimum pada suatu pohon.
- Child (Anak) = ?
- Parent (Orang Tua) = ?
- Sibling (Saudara Kandung) = ?

Terminologi dalam Pohon



- Simpul E, F, dan G disebut **anak (child)** dari simpul D
- Simpul D disebut **orang tua (parent)**
- B dan C disebut **sibling** atau saudara kandung
- Daun adalah simpul paling ujung dalam sebuah pohon. Simpul B, E, F, dan G adalah daun.
- Aras maksimum dari suatu pohon disebut tinggi atau kedalaman pohon tersebut. Pada pohon disamping aras mak = 3

Pohon BINER

- Merupakan suatu pohon di mana setiap cabangnya mempunyai maksimum 2 anak.
- Anak pohon pertama disebut ANAK POHON KIRI (LEFT SUBTREE) dan anak pohon kedua disebut ANAK POHON KANAN (RIGHT SUBTREE).

- Teorema 1 :

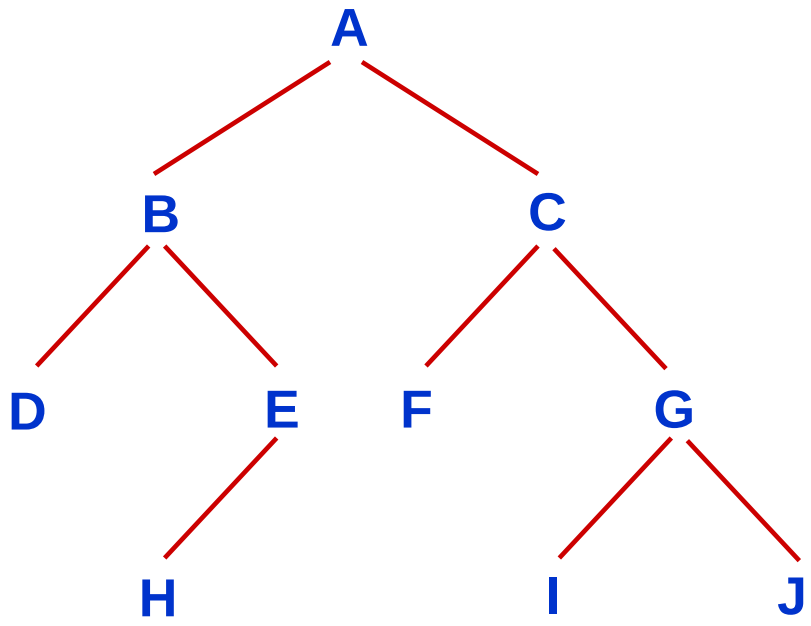
  Pohon biner beraturan dengan vertek lebih dari 2 mempunyai tepat 1 vertek berderajat 2 dan vertek lainnya berderajat 1 atau 3.

- Teorema 2 :

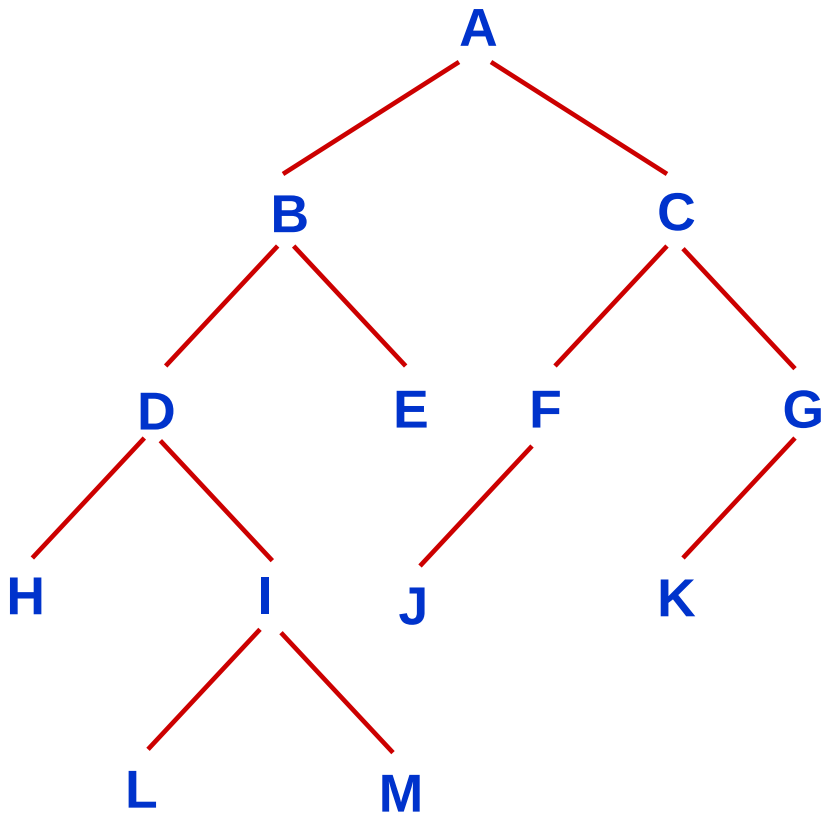
  Banyaknya vertek dalam suatu pohon biner beraturan selalu ganjil.

Traversal Pohon Biner

- Ada tiga cara untuk mengunjungi simpul-simpul pada pohon biner, misal (P=Parent, C1=Left Child dan C2=Right Child) :
  - PREORDER
    - Kunjungi → P
    - Kunjungi → C1
    - Kunjungi → C2
  - INORDER
    - Kunjungi → C1
    - Kunjungi → P
    - Kunjungi → C2
  - POSTORDER
    - Kunjungi → C1
    - Kunjungi → C2
    - Kunjungi → P



- Preorder : A-B-D-E-H-C-F-G-I-J
- Inorder : D-B-H-E-A-F-C-I-G-J
- Postorder : D-H-E-B-F-I-J-G-C-A



- Preorder : ?
- Inorder : ?
- Postorder : ?

Kode Huffman

- Salah satu algoritma yang biasa digunakan dalam kompresi data adalah algoritma pengkodean Huffman
- Pada Algoritma pengkodean Huffman simbol yang mempunyai probabilitas paling besar diberi kode paling pendek (jumlah bit kode sedikit) dan simbol dengan probabilitas paling kecil akan memperoleh kode paling panjang (jumlah bit kode banyak).
- Kode tersebut diperoleh dengan cara menyusun sebuah pohon Huffman untuk masing-masing simbol berdasarkan nilai probabilitasnya

Algoritma Pohon Huffman

- Berdasarkan daftar simbol dan probabilitas, **buatlah dua buah node dengan frekuensi paling kecil**.
- **Buatlah node parent** dari node tersebut dengan bobot parent merupakan jumlah dari probabilitas kedua node anak tersebut.
- **Masukkan node parent tersebut beserta bobotnya ke dalam daftar**, dan kemudian kedua node anak beserta probabilitasnya dihapus dari daftar.
- Salah satu node anak dijadikan jalur (dilihat dari node parent) untuk pengkodean 0 sedangkan lainnya digunakan untuk jalur pengkodean 1.

Contoh

Buatlah kode Huffman untuk “SCIENCE”

Solusi

- Buatllah daftar frekuensi kemunculan simbol-simbol dalam data tersebut.

- Urutkan berdasarkan frekuensi dari terkecil ke terbesar, jika ada yang frekuensinya sama maka urutan berdasarkan urutan huruf pada kata yang dimaksud

Huruf	Frekuensi
S	1
I	1
N	1
C	2
E	2

Contoh

Buatlah kode Huffman untuk “SCIENCE”

Solusi

- Buatllah daftar frekuensi kemunculan simbol-simbol dalam data tersebut.

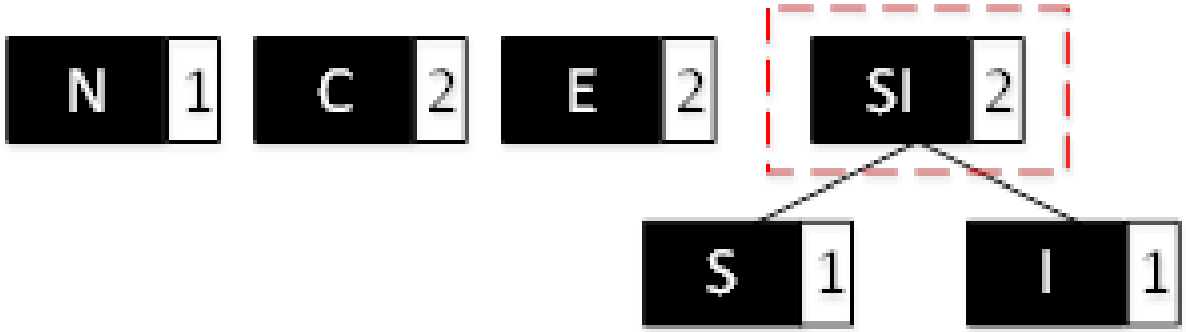
- Urutkan berdasarkan frekuensi dari terkecil ke terbesar, jika ada yang frekuensinya sama maka urutan berdasarkan urutan huruf pada kata yang dimaksud

Huruf	Frekuensi
S	1
I	1
N	1
C	2
E	2

Solusi

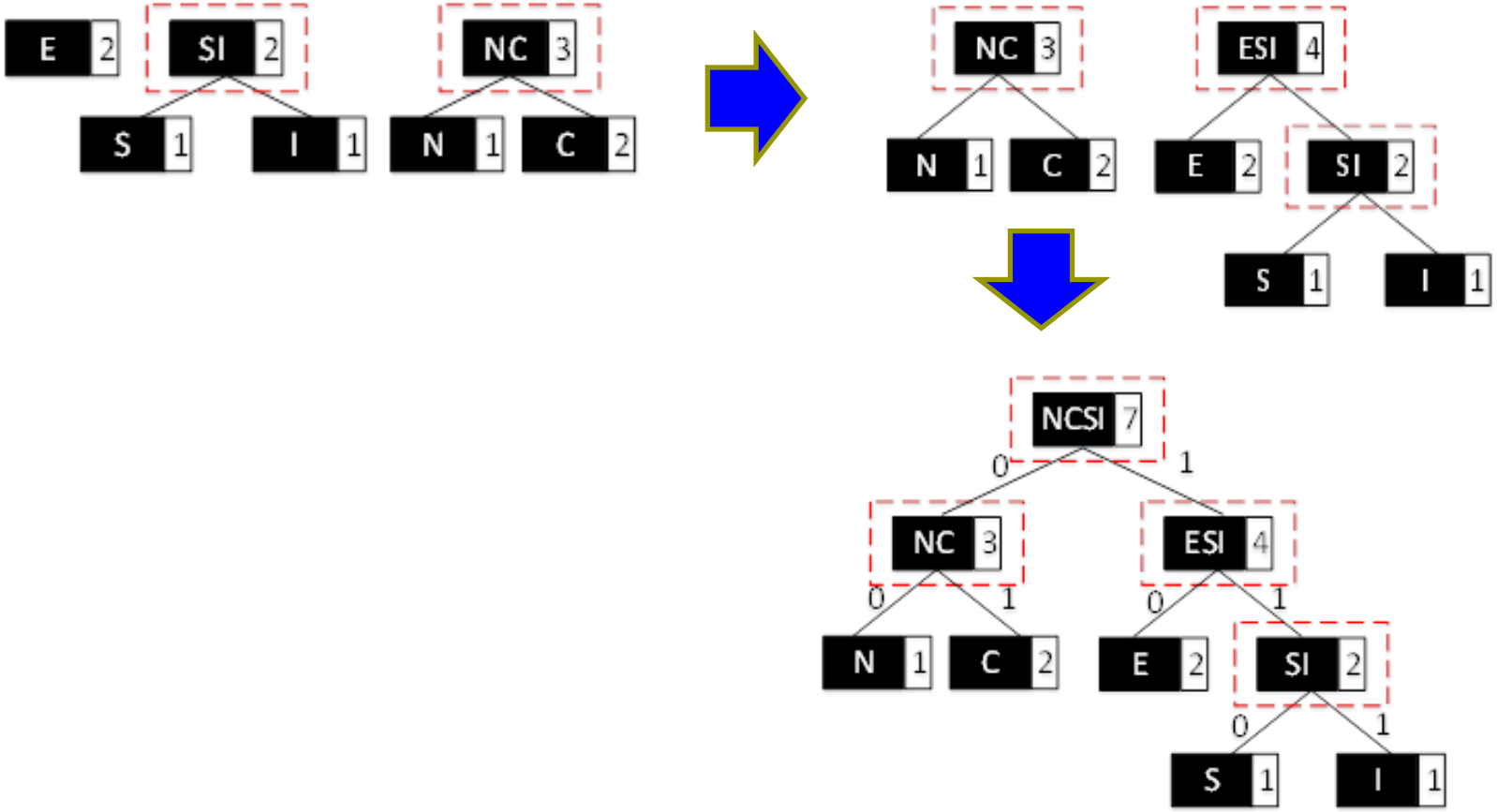
- Berdasarkan daftar frekuensi tersebut, kita buat daun-daun yang mewakili setiap simbol serta mengasosiasikan daun tersebut dengan frekuensi kemunculan simbol.

- Dari daun S dan I kita buat simpul baru SI yang akan menjadi orangtua dari simpul S dan I dan menyisipkannya ke dalam daftar sesuai dengan urutan frekuensinya.



Solusi

- Selanjutnya, kita lakukan hal yang sama secara berulang-ulang sehingga terbentuk satu pohon biner Huffman



Solusi

- Dengan menelusuri pohon biner Huffman yang telah dibuat, kita dapat membuat tabel kode Huffman sebagai berikut:

Huruf	Kode Huffman
S	110
I	111
N	00
C	01
E	10

- Sehingga kode Huffman untuk string “SCIENCE” adalah **110011111000110**.
- Dengan menggunakan kode ASCII memori yang dipakai adalah sebesar 56 bit sedangkan dengan menggunakan kode Huffman memori yang dipakai adalah sebesar 16 bit.

Contoh

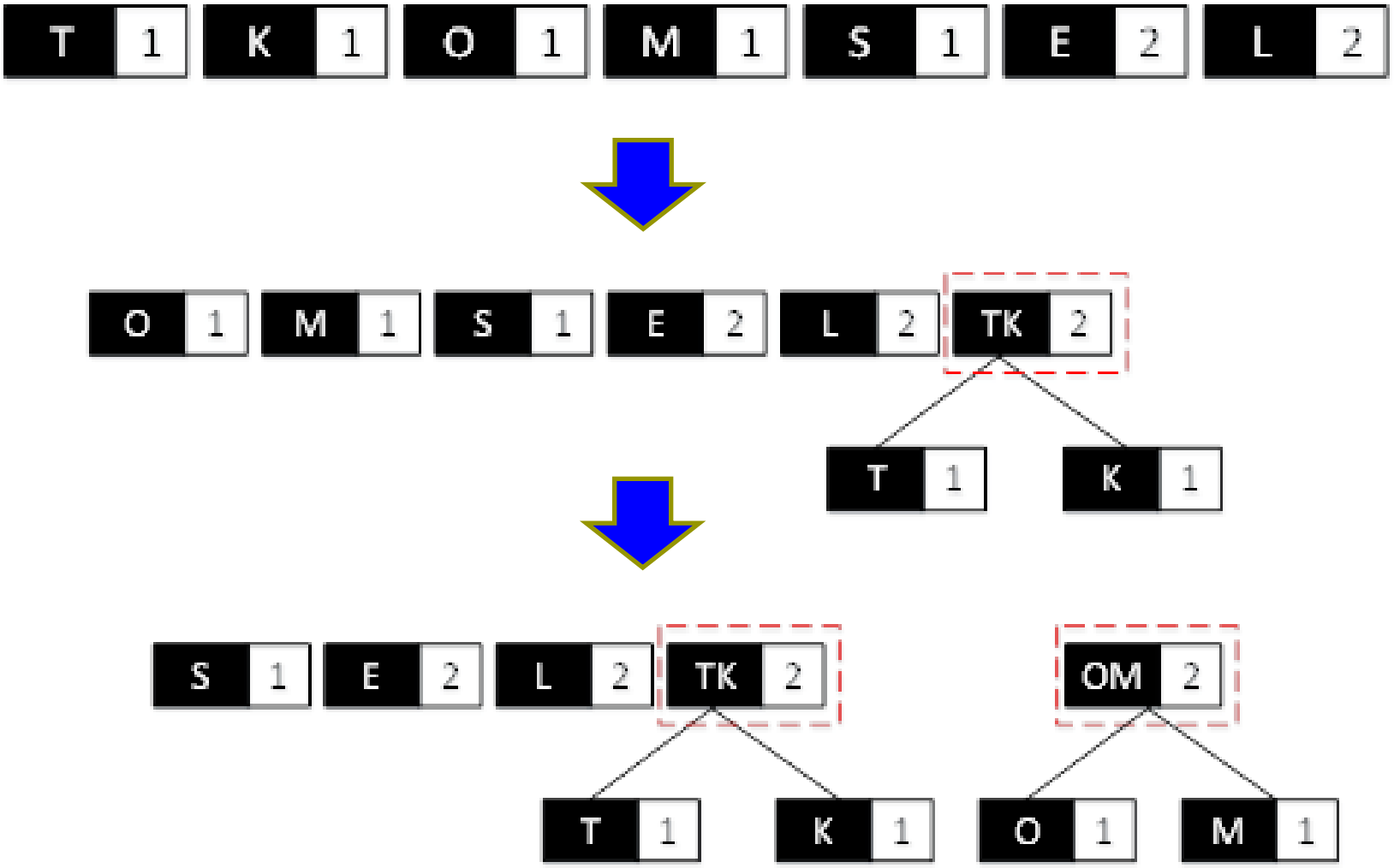
- Buatlah kode Huffman untuk "TELKOMSEL"

Solusi

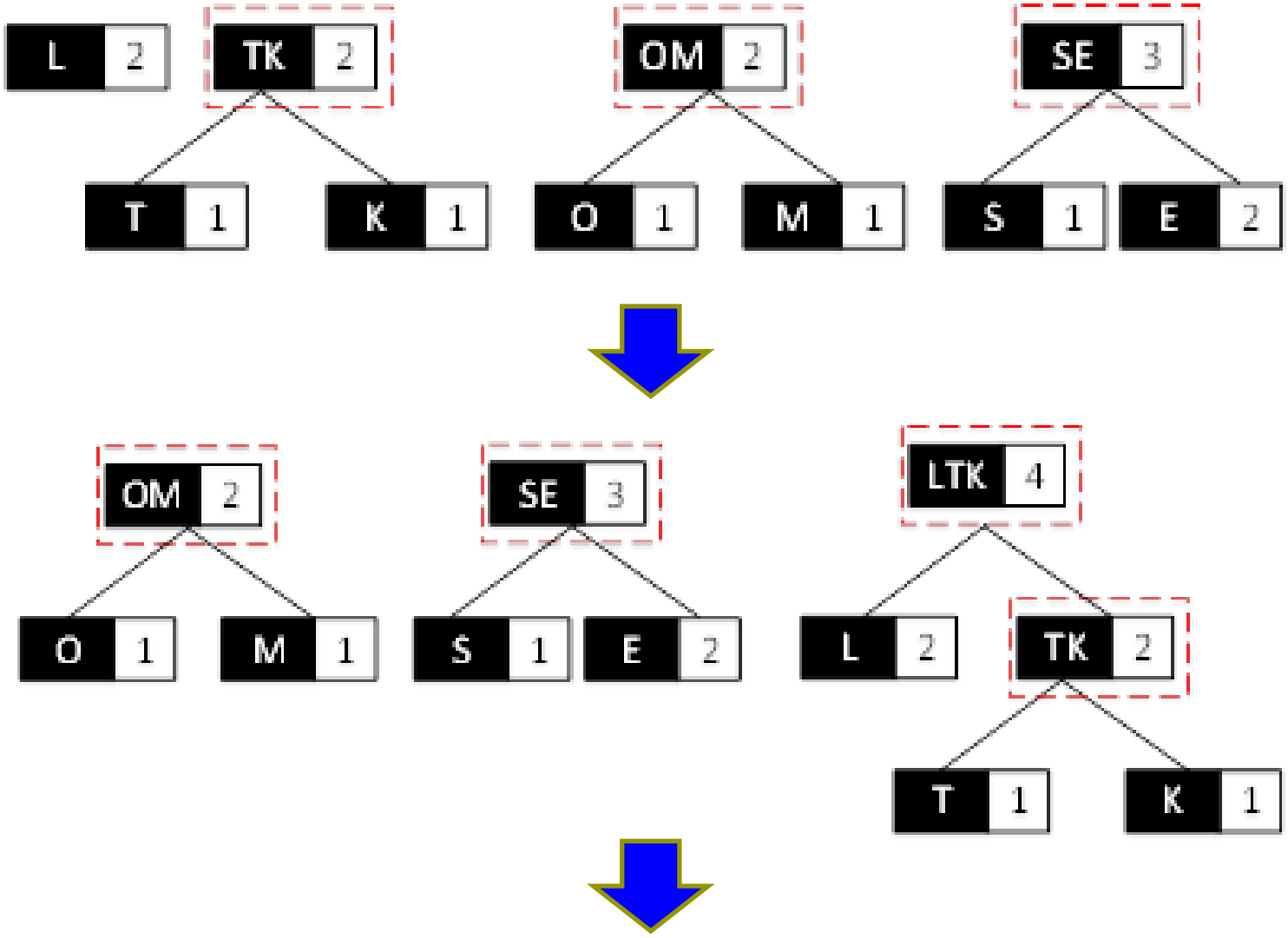
- Buatlah daftar frekuensi kemunculan simbol-simbol dalam data tersebut.
- Urutkan berdsarkan frekuensi dari terkecil ke terbesar, jika ada yang frekuensinya sama maka urutan berdasarkan urutan huruf pada kata yang dimaksud.

Huruf	Frekuensi
T	1
K	1
O	1
M	1
S	1
E	2
L	2

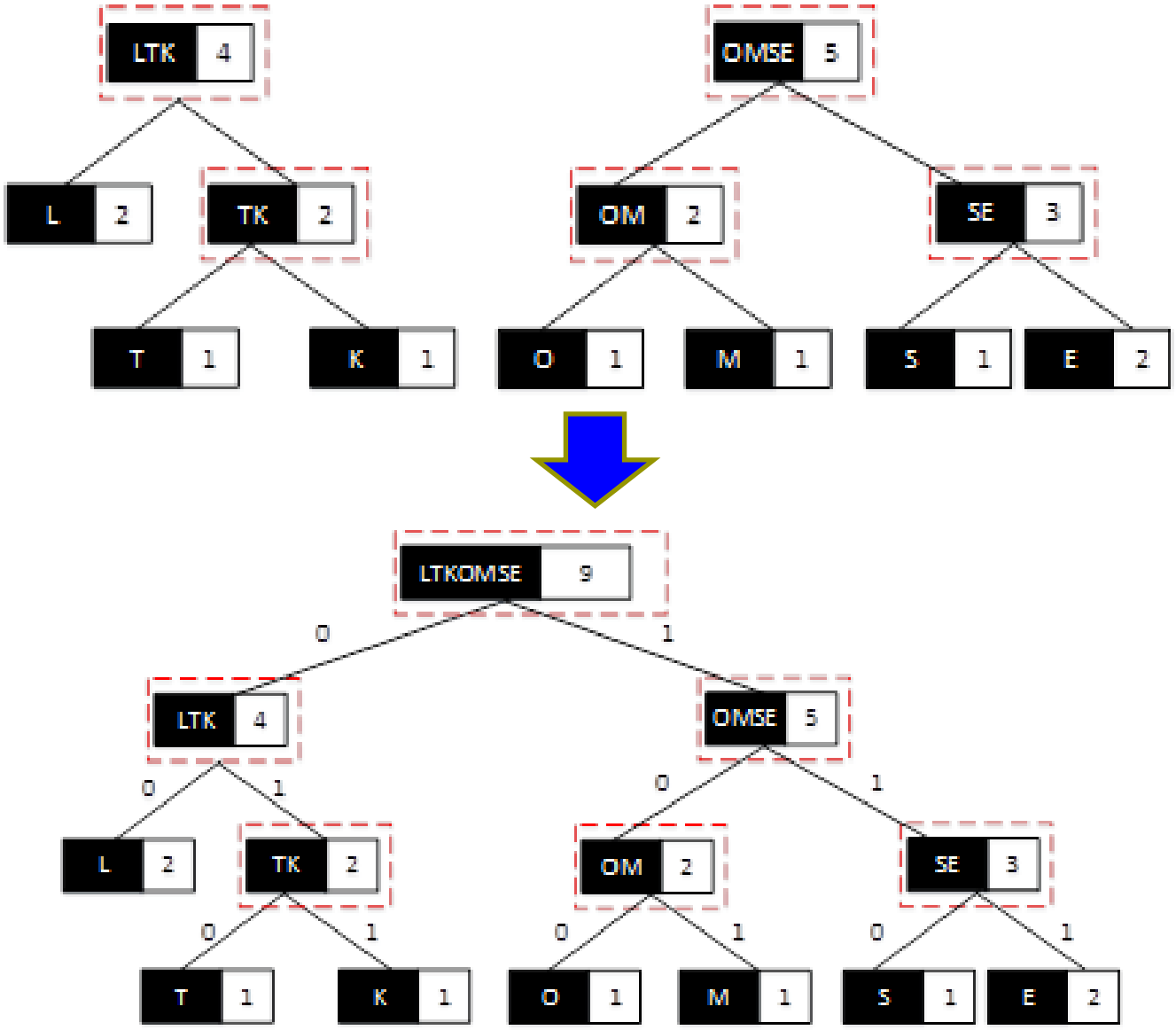
Solusi


T 1
K 1
O 1
M 1
S 1
E 2
L 2
O 1
M 1
S 1
E 2
L 2
TK 2
T 1
K 1
S 1
E 2
L 2
TK 2
OM 2
T 1
K 1
O 1
M 1

Solusi


L 2
TK 2
T 1
K 1
OM 2
O 1
M 1
SE 3
S 1
E 2
OM 2
O 1
M 1
SE 3
S 1
E 2
LTK 4
L 2
TK 2
T 1
K 1

Solusi


LTK 4
L 2
TK 2
T 1
K 1
OMSE 5
OM 2
SE 3
O 1
M 1
S 1
E 2
LTKOMSE 9
0
1
LTK 4
OMSE 5
0
1
0
1
L 2
TK 2
OM 2
SE 3
0
1
0
1
0
1
T 1
K 1
O 1
M 1
S 1
E 2

Solusi

- Dengan menelusuri pohon biner Huffman yang telah dibuat, kita dapat membuat tabel kode Huffman sebagai berikut:

Huruf	Kode Huffman	Huruf	Kode Huffman
T	010	E	111
K	011	L	00
O	100		
M	101		
S	110		

- Sehingga kode Huffman untuk string “Telkomsel” adalah **010111000111001011011100** = 25 bit

Contoh :

Dengan menggunakan String sebagai berikut :

- ABACCDA
- IKAN BAKAR DAN SAYUR ASAM
- SAYA SUKA MAKAN AYAM BAKAR
- MAKAN MALAM MENU NASI GORENG SAJA
- FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
- MATAKULIAH YANG PALING SAYA SUKAI ADALAH MATEMATIKA DISKRIT