

Penerapan Pohon Biner dalam Menerjemahkan Kode Morse

Ayu Rifanny Margareth / 13517038
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517038@std.stei.itb.ac.id

Abstract—Kode Morse sudah dikenal sejak akhir abad 19. Kode Morse merupakan representasi huruf, angka dan tanda baca yang menggunakan kombinasi antara titik (*dot* ‘.’) atau tanda hubung (*dash* ‘-’). Salah satu cara menerjemahkan Kode Morse yaitu dengan merepresentasikan Kode Morse dengan pohon biner. Pohon biner tersebut dapat digunakan untuk membuat program enkripsi atau deskripsi Kode Morse.

Keywords—Kode Morse, Pohon biner, simpul, pointer.

I. PENDAHULUAN

Kode Morse diciptakan oleh Samuel F. B. Morse di akhir abad 19. Pada awalnya, Kode Morse digunakan untuk komunikasi dengan telegraf. Sebelum terciptanya telegraf, pesan yang seseorang ingin sampaikan kepada orang lain harus dibawa oleh kurir yang akan mengingat ataupun menuliskan pesan tersebut untuk disampaikan kepada penerima pesan. Dengan metode tersebut, pesan akan tersampaikan setelah penerima pesan menunggu hingga kurir datang untuk memberikan pesan. Namun, metode ini membutuhkan waktu yang lama. Maka diciptakanlah telegraf untuk mempermudah komunikasi.

Telegraf hadir dan membawa suasana baru di bidang komunikasi. Pesan yang disampaikan melalui telegraf dikirim dengan menggunakan sinyal pendek (*short signal*) dan sinyal panjang (*long signal*). Sinyal pendek bisa direpresentasikan dengan titik (*dot*) dan sinyal panjang direpresentasikan dengan tanda hubung (*dash*). Penyampaian pesan dengan telegraf dinilai sangat cepat dibandingkan penyampaian pesan melalui kurir.

Namun, zaman sekarang Kode Morse tidak lagi digunakan sebagai kebutuhan komunikasi pokok semua orang. Walau demikian, Kode Morse masih digunakan sebagai komunikasi antar kapal. Komunikasi yang dimaksud adalah komunikasi dengan menggunakan sinyal cahaya. Cahaya kapal akan dipancarkan dan membentuk Kode Morse untuk memberi peringatan kepada kapal lainnya apabila akan terjadi bahaya. Komunikasi dengan Kode Morse sangat berguna di lautan apalagi saat keadaan gelap. Selain itu digunakan di perkapalan, Kode Morse juga masih digunakan dalam kegiatan kepramukaan siswa sekolah.

Pada penerapan di Matematika Diskrit, Kode Morse dapat dinyatakan sebagai pohon biner. Huruf-huruf maupun angka-angka yang dapat dinyatakan dengan Kode Morse akan disusun sedemikian sehingga membentuk sebuah pohon biner. Representasi ini yang akan membantu kita untuk mengetahui

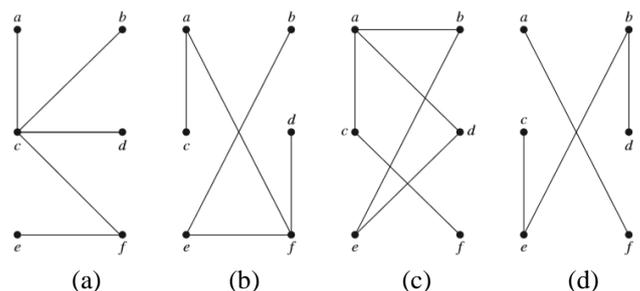
makna pesan yang disampaikan melalui Kode Morse. Dengan pohon biner, dapat dibuat sebuah program untuk menerjemahkan Kode Morse ke teks. Di era dimana Kode Morse sudah jarang dihapal oleh banyak orang, pembuatan program jenis ini akan sangat membantu untuk berbagai keperluan terkait penggunaan Kode Morse.

II. POHON

A. Definisi Pohon

Pohon merupakan suatu graf terhubung yang tidak memiliki sirkuit. Oleh karena itu, sebuah pohon harus selalu memiliki jalur (*path*) ke dari simpul ke simpul lainnya karena pohon merupakan graf terhubung. Jika $G = (V, E)$ merupakan graf tak-berarah sederhana dengan jumlah simpul n , maka semua pernyataan di bawah memiliki arti yang sama:

1. G adalah pohon.
2. Setiap pasangan simpul pada G dihubungkan oleh lintasan tunggal.
3. Misalkan m adalah jumlah sisi pada G , maka banyaknya m sebanding dengan $n-1$.
4. G tidak memiliki sirkuit.
5. Jika pada G ditambahkan sebuah sisi, maka akan membentuk sebuah sirkuit.
6. Semua sisi pada G merupakan jembatan.

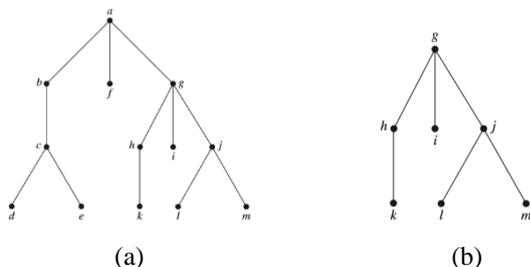


Gambar 1. (a) dan (b) merupakan pohon, sedangkan (c) dan (d) bukan merupakan pohon^[1]

B. Pohon Berakar

Sebuah pohon dapat disebut pohon berakar apabila pohon tersebut mengandung simpul yang ditetapkan sebagai akar. Dalam hal ini, akar merupakan sebuah simpul yang memiliki derajat-masuk sama dengan nol. Dari akar, simpul manapun yang terdapat pada pohon dapat dicapai. Untuk mencapai simpul-simpul tersebut tentu saja harus melewati lintasan. Selain akar, terdapat simpul-simpul daun. Daun memiliki derajat-keluar sama dengan nol. Artinya, daun tidak memiliki sisi lain selain dari sisi dari simpul yang menuju padanya. Selain

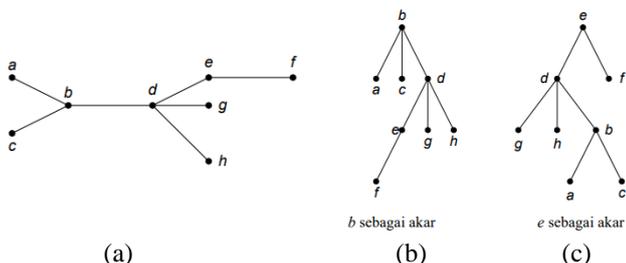
itu, daun juga sering disebut sebagai simpul terminal. Pada pohon berakar, simpul yang memiliki derajat-keluar lebih dari nol bisa disebut sebagai simpul cabang.



Gambar 2. (a) merupakan pohon berakar dan (b) merupakan simpul cabang^[1]

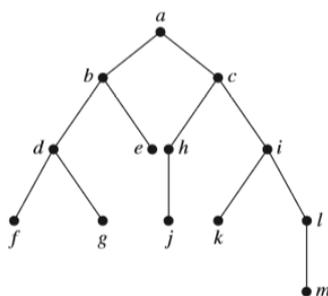
Setiap simpul yang terdapat pada pohon dapat dicapai dengan menelusuri lintasan (*path*) dari akar pohon tersebut. Untuk setiap simpul, terdapat sebuah lintasan tunggal (unik) yang dapat dicapai dari akar.

Apabila terdapat sebuah sembarang pohon tak-berakar, maka pohon tersebut dapat diubah menjadi pohon berakar dengan menetapkan sebuah simpul sebagai akarnya.



Gambar 3. (a) merupakan pohon tak berakar, (b) merupakan pohon (a) dengan simpul *b* sebagai akar, (c) merupakan pohon (a) dengan simpul *e* sebagai akar.^[1]

C. Terminologi pada Pohon Berakar

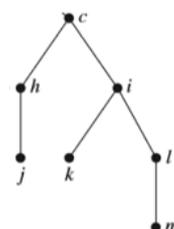


Gambar 4. Pohon berakar untuk menjelaskan terminologi pohon

Berikut merupakan beberapa terminologi dari pohon berakar:

1. Orangtua (*parent*) dan Anak (*child*)
Misalkan terdapat dua buah simpul x dan y , sebuah simpul y dikatakan anak simpul x jika terdapat sisi dari simpul x ke y . Demikian sebaliknya, simpul x dapat dikatakan sebagai orangtua (*parent*) y .
2. Lintasan (*path*)
Lintasan dari sebuah simpul ke simpul lain merupakan sejumlah urutan simpul yang dilalui dari simpul asal ke simpul tujuan. Misalkan pada Gambar 4, untuk mencapai simpul k dari simpul a dapat melalui lintasan $a, c, i,$ dan k .

3. Keturunan (*descendant*) dan Leluhur (*ancestor*)
Misalkan terdapat simpul x dan y , simpul x dapat dikatakan sebagai leluhur dari simpul y bila terdapat lintasan dari x yang mencapai y . Sebaliknya, bila x merupakan leluhur dari y , maka y merupakan keturunan dari x .
4. Saudara kandung (*sibling*)
Dua buah simpul dikatakan saudara kandung bila simpul-simpul tersebut memiliki orangtua yang sama. Pada Gambar 4, b dan c merupakan saudara kandung, sedangkan d dengan i bukan merupakan saudara kandung.
5. Upapohon (*subtree*)
Misalkan x adalah sebuah simpul pada pohon T . Upapohon dari x yaitu upagraf $T' = (V', E')$ dimana V' merupakan himpunan simpul dari keturunan x , dan E' mengandung seluruh lintasan yang berasal dari x . Dengan kata lain, upapohon merupakan sebagian cabang dari sebuah pohon.



Gambar 5. Upapohon dari Gambar 4.^[2]

6. Derajat (*degree*)
Jumlah upapohon (atau jumlah anak) pada sebuah pohon dapat dikatakan sebagai derajat-keluar dari sebuah simpul pada pohon berakar.
7. Daun (*leaf*)
Daun merupakan simpul yang tidak memiliki derajat-keluar. Selain itu, daun sering didefinisikan sebagai simpul yang berderajat nol. Pada Gambar 4, daun ditunjukkan oleh simpul f, g, j, k dan m .
8. Simpul dalam (*internal nodes*)
Simpul dalam merupakan sebutan bagi simpul yang memiliki anak.
9. Aras (*level*)
Aras (*level*) dari sebuah simpul adalah $1 +$ panjang lintasan yang dilalui untuk mencapai simpul tersebut dari akar. Sebagai konvensi, aras untuk akar adalah 0 .
10. Tinggi (*height*) atau Kedalaman (*depth*)
Kedalaman atau tinggi merupakan nilai aras yang paling besar yang dimiliki sebuah pohon. Pada Gambar 4, tinggi dari pohon tersebut adalah 4 .

D. Pohon m -ary

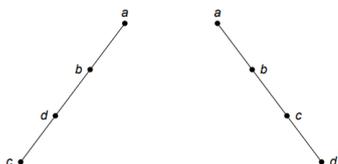
Pohon m -ary didefinisikan sebagai pohon berakar yang memiliki n buah anak. Sebagai contoh, pohon dengan $n = 2$ disebut pohon biner (*binary tree*), pohon dengan $n = 3$ disebut pohon 3 -ary. Pohon m -ary sangat banyak diimplementasikan di kehidupan dan juga di berbagai disiplin ilmu. Salah satu contoh kegunaan pohon m -ary adalah pemodelan bagan organisasi di sebuah perusahaan maupun pemerintahan.

E. Pohon Biner

Pohon biner paling sering digunakan di bidang informatika dan *computer science*. Pohon biner merupakan pohon m -ary

dengan $n = 2$. Oleh karena itu, sebuah pohon biner maksimal hanya memiliki dua buah anak. Pohon biner juga disebut sebagai pohon berurut, karena setiap anak pada pohon memiliki perbedaan urutan. Anak pada pohon biner sering disebut sebagai anak kiri dan anak kanan. Demikian juga upapohon pada pohon biner dapat disebut sebagai upapohon kiri dan upapohon kanan.

Selanjutnya, pohon biner yang hanya memiliki simpul di salah satu sisi saja disebut pohon condong. Terdapat dua jenis pohon condong, yaitu pohon condong kiri dan pohon condong kanan. Pohon condong kiri (*skew left*) memiliki simpul-simpul yang seluruhnya terletak di bagian kiri. Sedangkan pohon condong kanan (*skew right*) merupakan pohon yang seluruh simpulnya terletak di bagian kanan pohon.



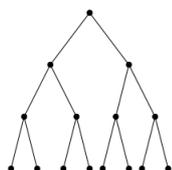
Gambar 6. Pohon biner condong kiri dan condong kanan^[2]

Pohon biner disebut pohon biner penuh (*full binary tree*) apabila setiap simpulnya memiliki tepat dua buah anak.

Sebuah pohon biner penuh memiliki 2^h daun, dimana h merupakan tinggi/kedalaman dari pohon tersebut. Untuk mengetahui jumlah seluruh simpul pada pohon biner penuh, kita dapat menjumlahkan seluruh jumlah daun pada tiap aras pohon, atau dapat disimplifikasi dengan rumus deret :

$$S = 2^0 + 2^1 + 2^2 + \dots + 2^h = 2^{h+1} - 1$$

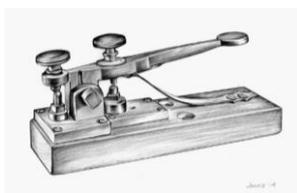
Selain pohon biner penuh, terdapat sebuah jenis pohon biner lainnya yaitu pohon biner seimbang (*balanced binary tree*). Pohon biner seimbang berarti pohon yang memiliki perbedaan tinggi antara upapohon kiri dan upapohon kanan maksimal 1. Selain itu, upapohon yang dimaksud juga tidak boleh condong (*skewed tree*).



Gambar 7. Pohon Biner penuh^[2]

III. KODE MORSE

A. Sejarah Kode Morse



Gambar 8. Telegraf

<https://stillunfold.com/science/surprising-facts-about-first-telegraph-first-message-morse-code>, diakses tanggal 8 Desember 2018 pukul 22.07

Kode Morse merupakan sebuah sistem representasi huruf,

angka dan symbol dengan menggunakan titik (*dot*), tanda hubung (*dash*) dan juga jeda (*space*). Salah satu penggunaannya yaitu dengan sinyal visual seperti cahaya, sinyal bunyi seperti peluit, maupun teks tertulis. Penemu Kode Morse ini adalah Samuel F.B. Morse, seorang penemu berkebangsaan Amerika. Pada awalnya, Kode Morse dianggap kurang cocok untuk teks karena kurangnya kode untuk huruf yang memiliki tanda diakritik. Maka dibuatlah sebuah variasi baru yang disebut *International Morse Code*, yaitu Kode Morse Internasional yang sekarang biasa digunakan. Terdapat beberapa perbedaan antara Kode Morse orisinal dan Kode Morse internasional. Pada Kode Morse yang orisinal, pola yang dibentuk adalah pola titik (*dot*) dan spasi (*space*). Sedangkan pada Kode Morse internasional membentuk pola dengan titik (*dot*) dan tanda hubung (*dash*).

Awal mula penggunaannya, Kode Morse digunakan untuk mengirimkan pesan singkat melalui telegraf. Pada telegraf terdapat sebuah tombol/kunci Morse yang dapat dinyalakan atau dimatikan sesuai dengan aturan Kode Morse. Saat pesan tiba di tujuan, maka arus elektrik akan membentuk elektromagnetik. Selanjutnya telegraf mempunyai tinta dan kertas dan langsung menuliskan pola-pola huruf tersebut pada kertas yang tersedia.

Penggunaan Kode Morse yang paling sering digunakan adalah kata 'SOS' yang memiliki kode '...---...'. SOS ini sering digunakan jika seseorang berada dalam bahaya. Pada umumnya, pesan SOS digunakan oleh pelaut yang sedang dalam situasi darurat.

B. Metode dan Cara Penggunaan Kode Morse

International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	• —	U	• • —
B	— • • •	V	• • • —
C	— • — •	W	— • —
D	— • •	X	— • • —
E	•	Y	— • — •
F	• • — •	Z	— — • •
G	— — •		
H	• • • •		
I	• •		
J	• — — —		
K	— • —		
L	• — • •		
M	— —		
N	— •		
O	— — —		
P	• — — •		
Q	— • — •		
R	• — •		
S	• • •		
T	—		
		1	• — — — —
		2	• • — — —
		3	• • • — —
		4	• • • • —
		5	• • • • •
		6	• • • • •
		7	— • — • •
		8	— • — • •
		9	— • — • •
		0	— — — — —

Gambar 9. Kode Morse Internasional

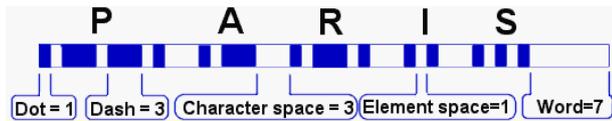
<https://orari8.wordpress.com/orari/kode-morse/>, diakses pada tanggal 8 Desember 2018 pukul 20.11

Kode Morse yang paling umum digunakan adalah Kode Morse Internasional. Dalam penggunaannya, Kode Morse memiliki durasi pengiriman yang diukur dengan *Words Per Minute* atau disingkat *WPM* yang berkisar antara 8-50 *WPM*.

Terdapat beberapa aturan terkait durasi antar huruf maupun kata dalam Kode Morse, yaitu:

1. Durasi untuk titik adalah satu satuan waktu.

2. Durasi untuk tanda hubung adalah tiga satuan waktu.
3. Untuk jeda pada setiap titik atau tanda hubung di alfabet yang sama memiliki durasi satu satuan waktu.
4. Jeda untuk setiap alfabet adalah tiga satuan waktu.
5. Jeda untuk setiap kata adalah tujuh satuan waktu.



Gambar 10. Visualisasi aturan durasi pada Kode Morse

http://www.nu-ware.com/NuCode%20Help/index.html?morse_code_structure_and_timing.htm

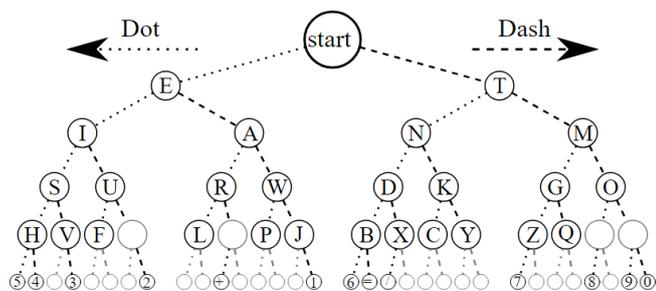
Diakses tanggal 8 Desember 2018, pukul 20.02

Gambar di atas merepresentasikan bagaimana Kode Morse dituliskan maupun diterjemahkan. Kata PARIS memiliki Kode Morse ‘-. - . -’.

IV. PEMBAHASAN

Pembahasan ini hanya membahas penerjemahan Kode Morse berupa teks dengan program sederhana. Kita tidak akan membahas tentang program yang membaca Kode Morse berupa suara ataupun dengan sinyal cahaya.

Kode Morse dapat direpresentasikan dengan pohon biner. Gambar 11 menunjukkan representasi Kode Morse dalam pohon biner.



Gambar 11. Pohon biner pada Kode Morse

<https://upload.wikimedia.org/wikipedia/commons/1/19/Morse-code-tree.svg>, diakses tanggal 8 Desember 2018 pukul 22.33

Tinggi dari pohon biner tersebut adalah lima. Selain itu, pohon biner ini memiliki akar yang ‘kosong’. Tidak ada huruf atau simbol apapun yang berada di aras nol. Hal ini benar, karena jika tidak ada alfabet, maka tidak ada juga Kode Morse yang mengandung sebuah titik atau mengandung sebuah tanda hubung.

Setiap simpul pada pohon biner menyimpan informasi karakter. Setiap simpul mengandung karakter yang berbeda, tergantung lintasan yang dilalui. Lintasan ke anak kiri direpresentasikan dengan titik (*dots*) dan lintasan ke anak kanan direpresentasikan dengan tanda hubung (*dash*). Dari penjelasan di atas, dapat diketahui bahwa Kode Morse untuk setiap karakter adalah lintasan dari akar menuju simpul karakter tersebut. Sebagai contoh, jika diberikan Kode Morse ‘-.’, maka lintasan dari akar adalah kiri-kanan-kanan-kiri, di mana setelah melewati lintasan tersebut kita akan mencapai simpul dengan karakter P.

Setiap simpul pada pohon biner mengandung informasi berupa karakter. Karakter-karakter tersebut yang nantinya akan menjadi keluaran dari program.

Perlu dicatat, pohon biner untuk Kode Morse ini bukan merupakan pohon pencarian biner (*binary search tree*), sehingga setiap informasi pada simpul tidak membentuk pola seperti pohon pencarian biner.

Sebelum melakukan pembacaan Kode Morse, pohon biner harus dibentuk terlebih dahulu. Pohon biner dapat dibentuk dengan membuat sebuah struktur data pohon biner. Jika diimplementasikan ke bahasa pemrograman C maka struktur data akan menjadi sebagai berikut:

```
typedef struct tNode *addrNode;
typedef struct tNode {
    char info;
    addrNode left;
    addrNode right;
} Node;
typedef addrNode BinTree;
```

Fig. 1: Struktur data pohon biner

Info merupakan informasi karakter pada sebuah simpul, left merupakan pointer ke anak simpul kiri dan right merupakan pointer ke anak simpul kanan. Tipe bentukan BinTree merupakan tipe untuk pohon itu sendiri

Dalam membentuk pohon biner, terdapat dua komponen yang penting yaitu inisialisasi pohon serta menambahkan daun. Secara garis besar, langkah-langkah membentuk pohon biner untuk Kode Morse adalah sebagai berikut:

1. Inisialisasi pohon, yaitu dengan membentuk sebuah pohon biner dengan akar yang memiliki info *dummy*.
2. Tambahkan daun-daun pada pohon biner sesuai dengan Gambar 11 hingga membentuk pohon biner Kode Morse yang sesuai.
3. Untuk simpul pada Gambar yang tidak memiliki karakter, saat menambahkan daun tersebut dapat juga menggunakan info *dummy*.

Pada tahapan kedua, daun-daun bisa di tambahkan sehingga membentuk pohon biner seperti Gambar 11. Perlu diperhatikan bahwa susunan dan letak karakter di simpul itu penting. Misalnya untuk karakter ‘S’ harus di letakkan menjadi anak simpul dari simpul ‘I’.

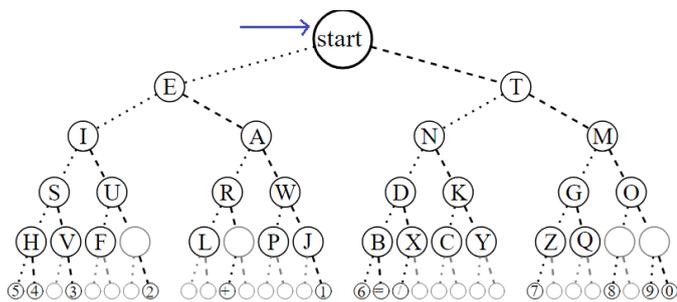
Pada Gambar 11 terdapat beberapa simpul yang kosong (tidak memiliki informasi karakter di simpulnya). Hal ini terjadi karena memang tidak ada karakter yang di representasikan apabila proses melewati lintasan tersebut. Namun tentu saja simpul tersebut tidak bisa dilewatkan (tidak dibuat). Hal ini karena terdapat representasi angka maupun simbol berada di aras yang paling besar yaitu lima. Untuk dapat mengakses simpul yang berisi angka, perlu melewati simpul-simpul yang kosong di aras empat. Karena kebutuhan itulah, informasi simpul yang kosong dapat menggunakan info *dummy*. Info

dummy dapat berupa karakter apapun yang penting tidak menimbulkan ambiguitas pada pembentukan pohon biner. Disarankan untuk dalam memilih info *dummy* karakter yang dipakai merupakan symbol yang jarang digunakan. Info *dummy* tidak dianggap sebagai sebuah karakter yang akan dikeluarkan. Sehingga, apabila pembacaan Kode Morse berakhir di simpul yang mengandung info *dummy*, berarti masukkan (*input*) salah.

Apabila pohon biner berupa informasi Kode Morse telah terbentuk, langkah selanjutnya adalah membuat kode program agar dapat membaca Kode Morse dan menerjemahkan kode tersebut menjadi sebuah teks. Adapun langkah-langkah penerjemahannya adalah:

1. Pembacaan string dari masukan *user*.
2. Inisialisasi pointer ke akar pohon biner.
3. Iterasi karakter *c* hingga mencapai karakter '\0' (akhir dari string):
 - a. Jika *c* sama dengan titik ('.'), ubah pointer ke anak simpul kiri.
 - b. Jika *c* sama dengan tanda hubung ('-'), ubah pointer ke anak simpul kanan.
 - c. Jika *c* sama dengan spasi (' ') atau karakter '\0', cetak info pada simpul yang ditunjuk pointer (simpul terakhir yang ditunjuk pointer), kemudian kembalikan pointer ke akar dari pohon biner.
4. Sebagai catatan, apabila ternyata *user* salah memberikan masukan, misalnya masih ada tanda titik atau tanda hubung saat pointer sudah menunjuk di aras pohon yang maksimum, atau pointer berhenti di simpul yang memiliki *dummy* info, maka berikan pesan kesalahan dan keluar dari program.

Misalkan masukkan dari terminal adalah `'... - - -.'`. Setelah membaca masukkan dari terminal, tahapan selanjutnya adalah inisialisasi pointer di akar pohon biner. Proses ini akan digambarkan sebagai berikut.

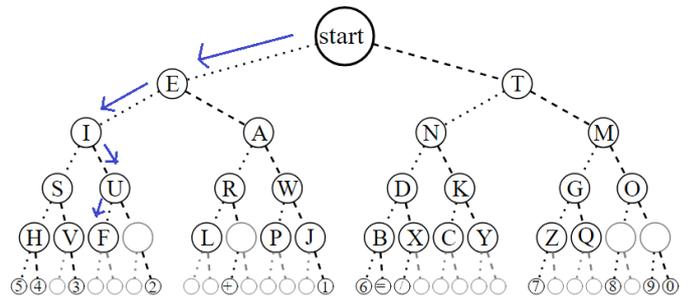


Gambar 12. Pointer menunjuk ke akar pohon biner

Proses selanjutnya ialah membaca karakter pertama. Karakter pertama merupakan tanda titik (*dot*) sehingga pointer harus menunjuk ke simpul anak kiri dari akar. Di proses ini, pointer sudah menunjuk ke simpul dengan karakter 'E'. Namun, karakter 'E' tidak akan dikeluarkan (*output*) karena karakter selanjutnya yang dibaca bukan merupakan spasi maupun akhir dari string. Sehingga pembacaan karakter dilanjutkan ke karakter berikutnya.

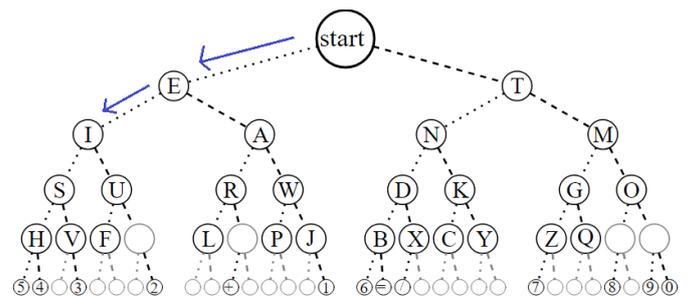
Dari string masukkan, karakter selanjutnya yang dibaca adalah titik. Sama seperti langkah sebelumnya, pointer akan menunjuk ke daun kiri dari simpul yang terakhir ditunjuk. Sehingga pointer akan menunjuk ke simpul dengan karakter 'I'.

Kemudian karakter yang akan di proses berikutnya adalah karakter tanda hubung (*dash*). Berarti pointer akan bergerak ke anak simpul kanan. Dari anak simpul kanan dari 'I' adalah 'U'. Selanjutnya, karakter yang akan dibaca adalah karakter titik. Dari simpul 'U' pointer akan bergerak ke kiri dan berhenti di karakter 'F'. Karena karakter selanjutnya adalah spasi, maka karakter 'F' akan dikeluarkan ke terminal. Proses yang telah dilalui sampai tahap ini adalah sebagai berikut.



Gambar 13. Pemrosesan Kode Morse untuk huruf pertama

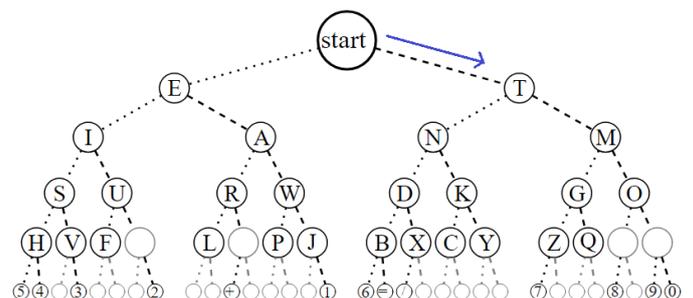
Karakter selanjutnya adalah karakter spasi, sehingga pointer harus dikembalikan ke akar pohon biner. Sehingga, pada tahap ini, pointer akan menunjuk ke akar pohon sesuai dengan Gambar 12. Kemudian, proses yang dilakukan adalah membaca karakter setelah spasi. Karakter yang dibaca adalah titik, sehingga pointer kembali menunjuk ke anak simpul kiri dari akar, yaitu simpul dengan info 'E'. Selanjutnya, karakter yang dibaca adalah titik, sehingga pointer menunjuk ke simpul 'I'. Sampai pada tahapan ini, proses yang dilalui digambarkan sebagai berikut.



Gambar 14. Pemrosesan Kode Morse untuk huruf kedua

Maka setelah melalui proses ini, karakter 'I' akan dikeluarkan karena pada string masukkan karakter selanjutnya merupakan spasi. Sehingga karakter pada simpul terakhir dikeluarkan dan pointer harus dikembalikan ke akar pohon biner.

Proses pembacaan selanjutnya adalah pembacaan karakter tanda hubung. Pada tahapan ini, simpul yang ditunjuk oleh pointer adalah simpul 'T'. Proses ini dapat digambarkan sebagai berikut.



Gambar 15. Pemrosesan Kode Morse untuk huruf ketiga

- [3] <https://www.britannica.com/topic/Morse-Code>, diakses tanggal 8 Desember 2018 pukul 15.03
- [4] <https://www.cryptomuseum.com/radio/morse/>, diakses tanggal 9 Desember 2018 pukul 8.37

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Desember 2018

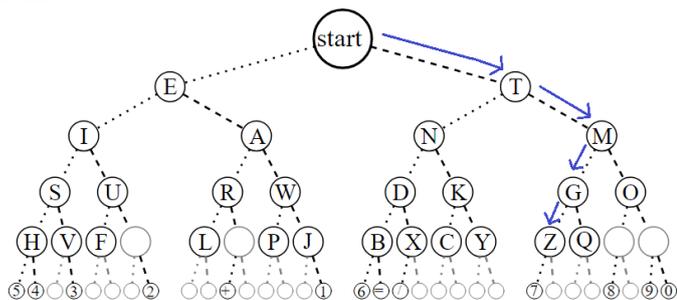
Ttd



Ayu Rifanny Margareth
13517038

Karakter selanjutnya yang dibaca merupakan karakter spasi. Sama seperti langkah sebelumnya, ketika karakter yang dibaca adalah spasi, maka proses yang dilakukan adalah mengeluarkan karakter pada simpul terakhir yang ditunjuk ke terminal dan mengembalikan pointer agar menunjuk ke akar.

Kemudian pembacaan karakter dilanjutkan dengan membaca karakter tanda hubung. Dari akar, pointer akan menunjuk ke simpul 'T'. Selanjutnya, karakter yang dibaca merupakan tanda hubung lagi. Maka pointer akan menunjuk ke anak simpul kanan dari simpul 'T', yaitu simpul 'M'. Dilanjutkan dengan pembacaan karakter titik sehingga pointer akan menunjuk ke anak kiri dari simpul 'M' yaitu simpul 'G'. Pembacaan masih berlanjut ke karakter terakhir. Karakter terakhir yang dibaca adalah karakter titik, maka pointer akan menunjuk ke simpul dengan karakter 'Z' yang merupakan anak simpul kiri dari 'G'. Proses yang dilalui sampai tahap ini digambarkan sebagai berikut.



Gambar 15. Pemrosesan Kode Morse untuk huruf keempat

Setelah berada di tahap ini, semua karakter telah di proses, sehingga iterasi akan dihentikan. Hasil terjemahan dari Kode Morse '...-...-...' adalah FITZ. Hasil tersebut akan di tampilkan di terminal.

V. SIMPULAN

Kode Morse dapat diterjemahkan dengan menggunakan program komputer. Dalam memudahkan proses penerjemahan, penggunaan pohon biner merupakan pendekatan yang baik untuk digunakan. Pohon biner dapat membantu proses pencarian karakter yang tepat melalui lintasan yang telah dilalui hingga simpul karakter yang dituju.

VII. UCAPAN TERIMA KASIH

Pertama-tama penulis mengucapkan syukur kepada Tuhan Yang Maha Esa, karena atas segala berkat dan rahmatNya penulis dapat menyelesaikan makalah ini. Tak lupa, penulis juga menyampaikan terima kasih kepada Ibu Dra. Harlili, sebagai dosen mata kuliah Matematika Diskrit, atas bimbingan dan ilmu yang diberikan selama ini. Kemudian terimakasih juga kepada orang tua dan teman-teman atas dukungan dan doa sehingga makalah ini dapat diselesaikan dengan baik.

REFERENSI

- [1] Munir, Rinaldi, "Diktat Kuliah IF2120 Matematika Diskrit", Program Studi Teknik Informatika ITB, 2006.
- [2] Kenneth H. Rosen, "Discrete Math and Its Applications 7th Edition.", New York: McGraw-Hill Companies, 2012.