

PERTEMUAN 15

POHON PENCARIAN

BINARY SEARCH TREE (BST)/POHON PENCARIAN BINER

Pohon pencarian biner salah pohon pencarian yang sangat penting. Khususnya pada persoalan yang banyak melakukan operasi pencarian, penyisipan, dan penghapusan elemen.

Aturan :

Nilai di subpohon kiri $< node$

Nilai di subpohon kanan $> node$

Setiap node memiliki maksimal 2 *child*

Cara Pencarian pada BST

1. Mulai dari root
2. Bandingkan nilai yang dicari dengan *node* saat ini
3. Jika :
 - nilai < *node* → ke kiri
 - nilai > *node* → ke kanan
4. Ulangi sampai
 - Nilai ditemukan, atau
 - Node = null (tidak ada)

POHON PENCARIAN

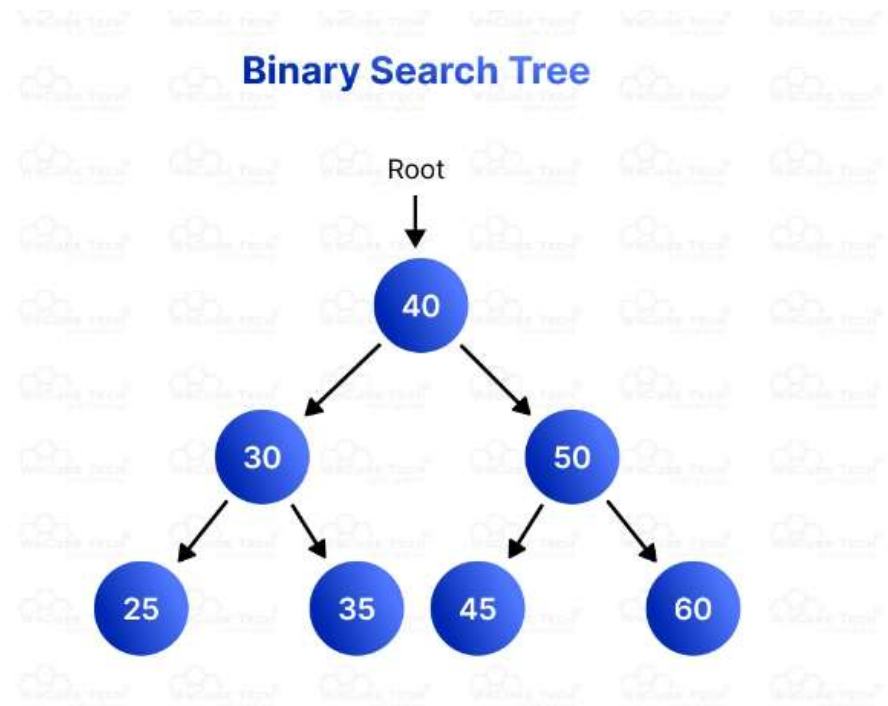
Contoh :

Mencari 25

1. Root = 40 → ke kiri
2. Node = 30 → ke kiri
3. Node = 25 ditemukan

Mencari 60

1. Root = 40 → ke kanan
2. Node = 50 → ke kanan
3. Node = 60 ditemukan



Fungsi Utama

Menyimpan data terurut secara dinamis (menambah-menghilangkan).

Digunakan pada :

- Struktur data kamus (Dictionary)
- Manajemen data sederhana
- Sistem pencarian berbasis nilai

KELEBIHAN

Pencarian data lebih cepat dari array biasa.

Insert dan delete data lebih flexible (tidak perlu menggeser elemen).

Implementasi mudah.

Efisien untuk data acak.

KEKURANGAN

Bisa tidak seimbang → Pencarian lambat

BINARY TREE TRAVERSAL (KUNJUNGAN POHON BINER)

Kunjungan pada Pohon Binar merupakan salah satu operasi yang sering dilakukan pada suatu Pohon Binar tepat satu Kali (Binary Tree Traversal). Operasi ini terbagi menjadi 3 bentuk :

1. Kunjungan secara Preorder (Depth First Order),
mempunyai urutan :
 - a. Cetak isi simpul yang dikunjungi (Simpul Akar)
 - b. Kunjungi Cabang Kiri
 - c. Kunjungi Cabang Kanan

2. Kunjungan secara Inorder (Symetric Order), mempunyai urutan :
 - a. Kunjungi Cabang Kiri
 - b. Cetak isi simpul yang dikunjungi (Simpul Akar)
 - c. Kunjungi Cabang Kanan

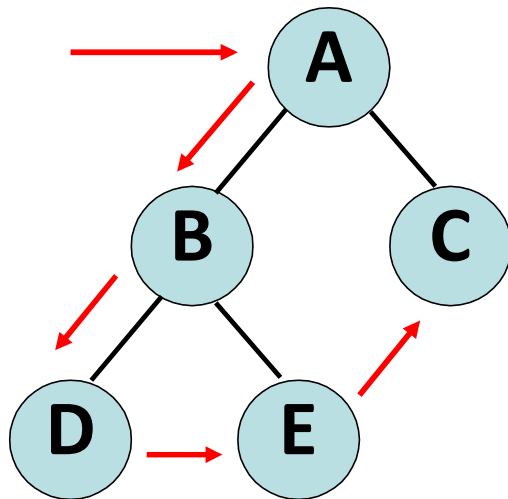
3. Kunjungan secara Postorder, mempunyai urutan :
 - a. Kunjungi Cabang Kiri
 - b. Kunjungi Cabang Kanan
 - c. Cetak isi simpul yang dikunjungi (Simpul Akar)

Pada ketiga cara kunjungan diatas, kunjungan ke Cabang Kiri dilakukan terlebih dahulu, baru kemudian kunjungan ke Cabang Kanan. Dengan orientasi semacam ini, Ketiga kunjungan diatas disebut dengan Left To Right Oriented (LRO).

Jika kunjungan ke Cabang Kanan dilakukan lebih dahulu baru kemudian kunjungan ke Cabang Kiri, maka Orientasi semacam ini disebut Right To Left Oriented (RLO).

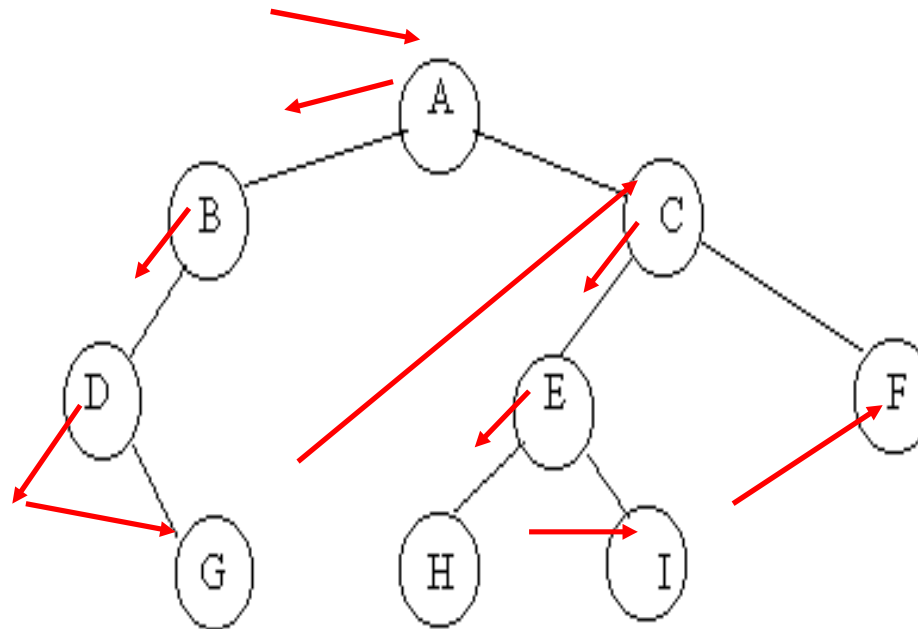
POHON PENCARIAN

1. Kunjungan secara Preorder (Depth First Order), mempunyai urutan :
 - a. Cetak isi simpul yang dikunjungi (Simpul Akar)
 - b. Kunjungi Cabang Kiri
 - c. Kunjungi Cabang Kanan



A B D E C

Kunjungan PreOrder

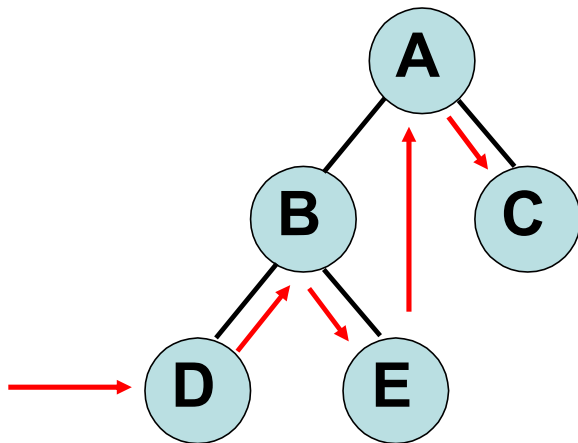


**Contoh
PreOrder**

Hasil : A B D G C E H I F

2. Kunjungan secara Inorder (Symetric Order), mempunyai urutan :

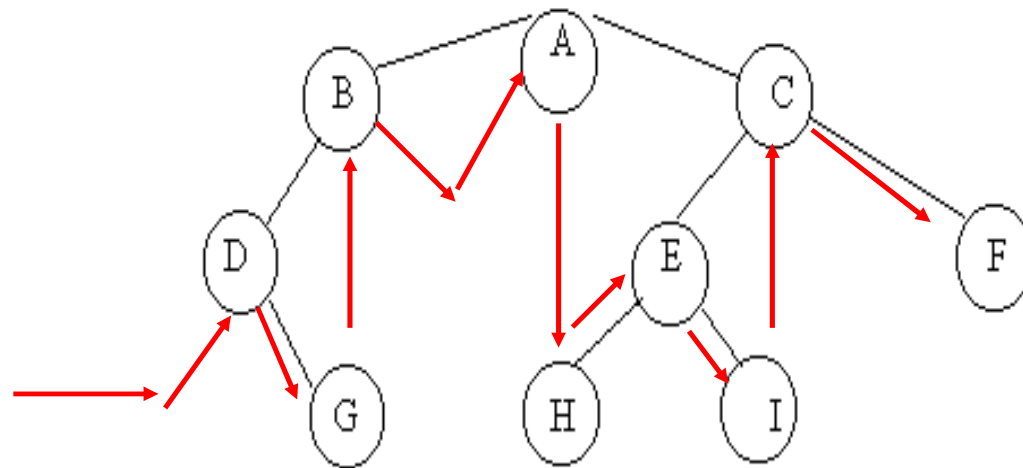
- a. Kunjungi Cabang Kiri
- b. Cetak isi simpul yang dikunjungi (Simpul Akar)
- c. Kunjungi Cabang Kanan



D B E A C

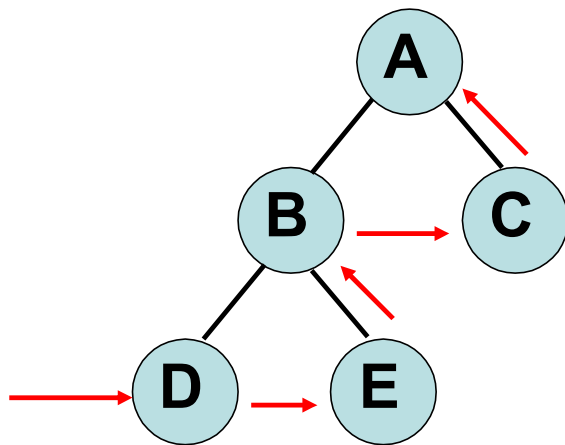
Kunjungan InOrder

**Contoh
InOrder**



Hasil : D G B A H E I C F

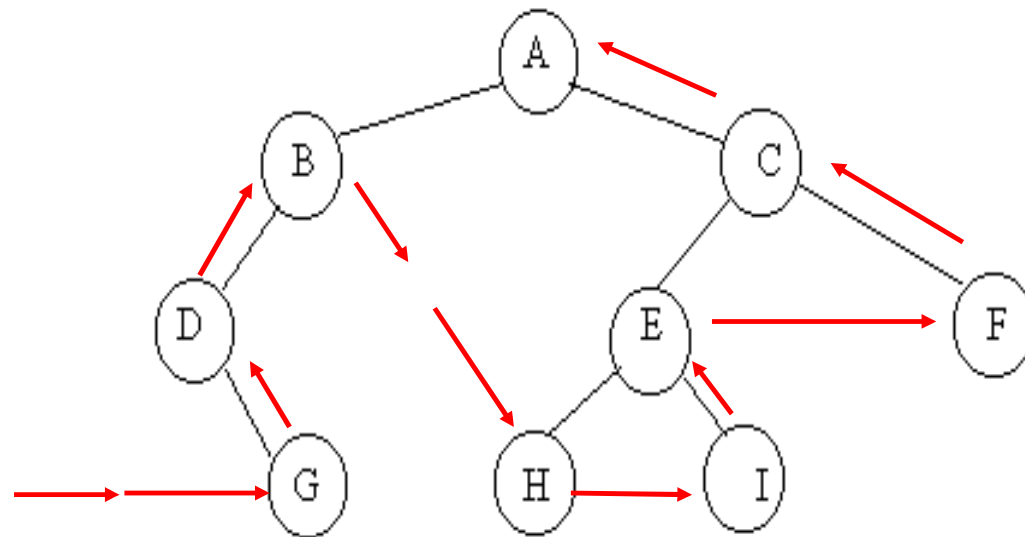
3. Kunjungan secara Postorder, mempunyai urutan :
- a. Kunjungi Cabang Kiri
 - b. Kunjungi Cabang Kanan
 - c. Cetak isi simpul yang dikunjungi (Simpul Akar)



D E B C A

Kunjungan PostOrder

**Contoh
PostOrder**



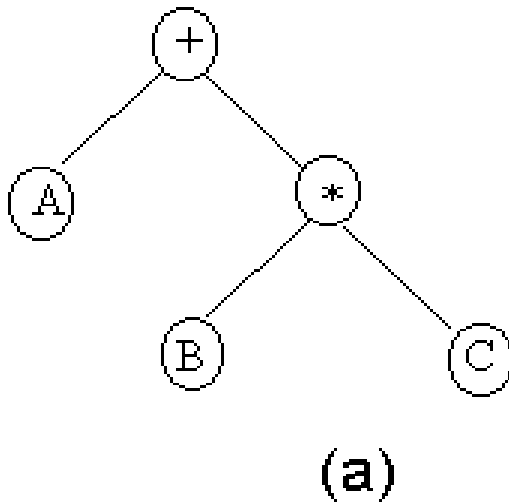
Hasil : G D B H I E F C A

APLIKASI POHON BINER

NOTASI PREFIX, INFIX DAN POSTFIX

Pada bagian ini akan dibahas tentang bagaimana menyusun sebuah Pohon Binar yang apabila dikunjungi secara PreOrder akan menghasilkan Notasi Prefix, kunjungan secara InOrder menghasilkan Notasi Infix, dan kunjungan PostOrder menghasilkan Notasi Postfix.

POHON BINER

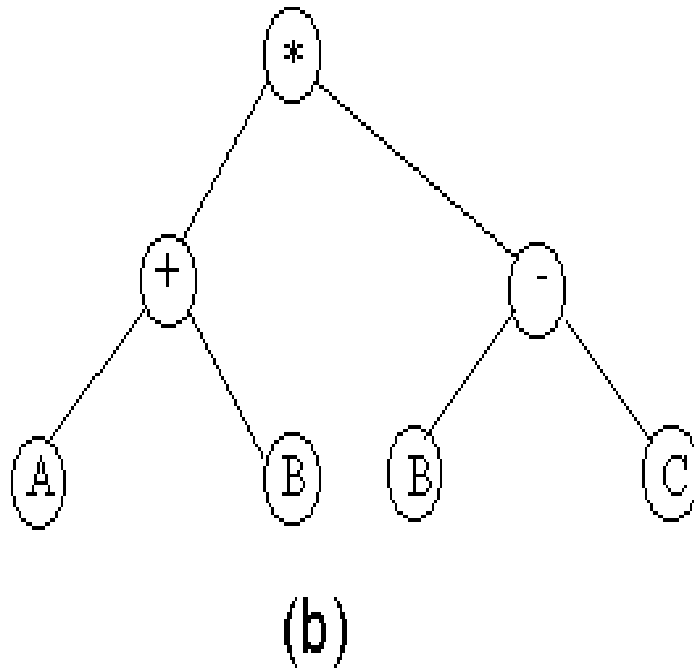


$+A*BC$ = Pre Order (Prefix)

$(A+(B*C))$ = In Order (Infix)

$ABC*+$ = Post Order (Postfix)

POHON BINER



*+AB-BC = Pre Order (Prefix)

((A+B) * (B-C)) = In Order (Infix)

AB+BC-* = Post Order (Postfix)

POHON BINER

Pre Order (Prefix)

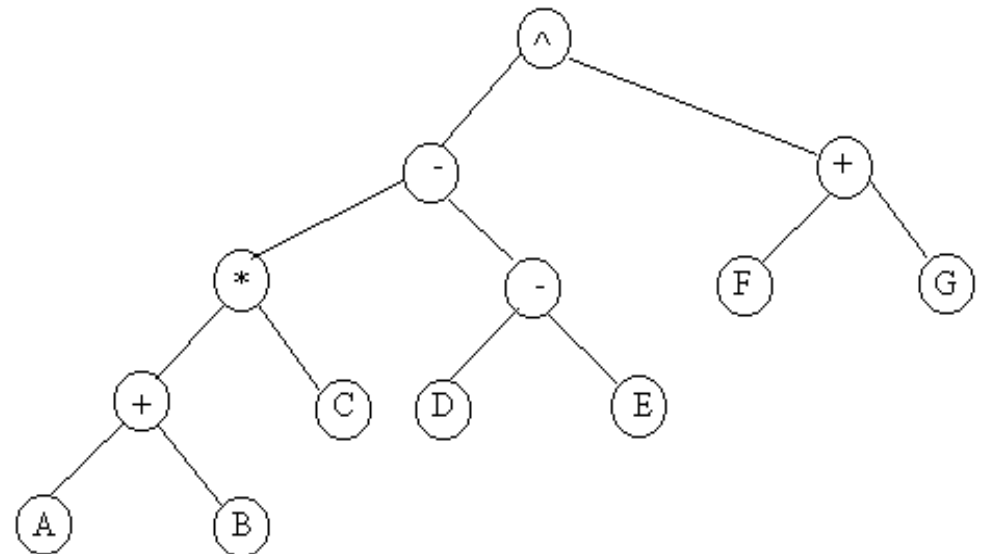
$^-*+ABC-DE+FG$

In Order (Infix)

$((((A+B) * C) - (D-E))^+(F+G))$

Post Order (Post fix)

$AB+C*DE--FG+^$



(c)

LATIHAN SOAL

1. Gunakan pohon berakar untuk menggambarkan semua kemungkinan hasil dari pertandingan tenis antara dua orang pemain, Amir dan Rian, yang dalam hal ini pemenangnya Adalah pemain yang pertama memenangkan 2 set berturut-turut atau pemain yang pertama memenangkan total 3 set.
2. Diberikan masukan berupa rangkaian karakter dengan urutan sebagai berikut:
 - a. Gambarkan pohon pencarian
 - b. Tentukan hasil penelusuran preorder, InOrder dan PostOrder.



Terimakasih