

Pertemuan 10

KUNJUNGAN PADA 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

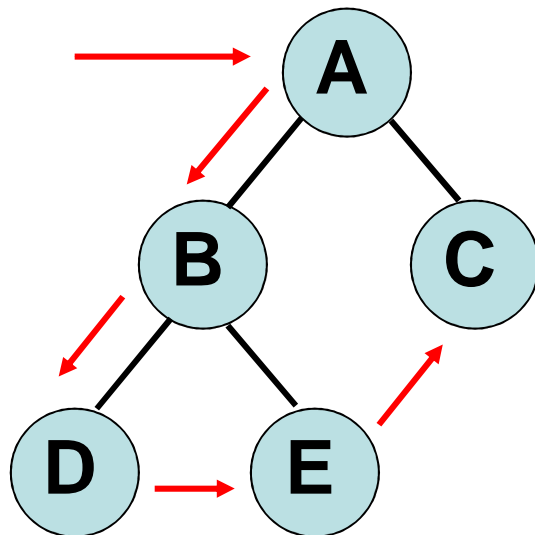
2. Kunjungan secara Inorder (Symetric Order), mempunyai urutan :
 - a. Kunjungi Cabang Kiri
 - b. Cetak isi simpul yang dikunjungi (Simpul Akar)
 - c. Kunjungi Cabang Kanan

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).

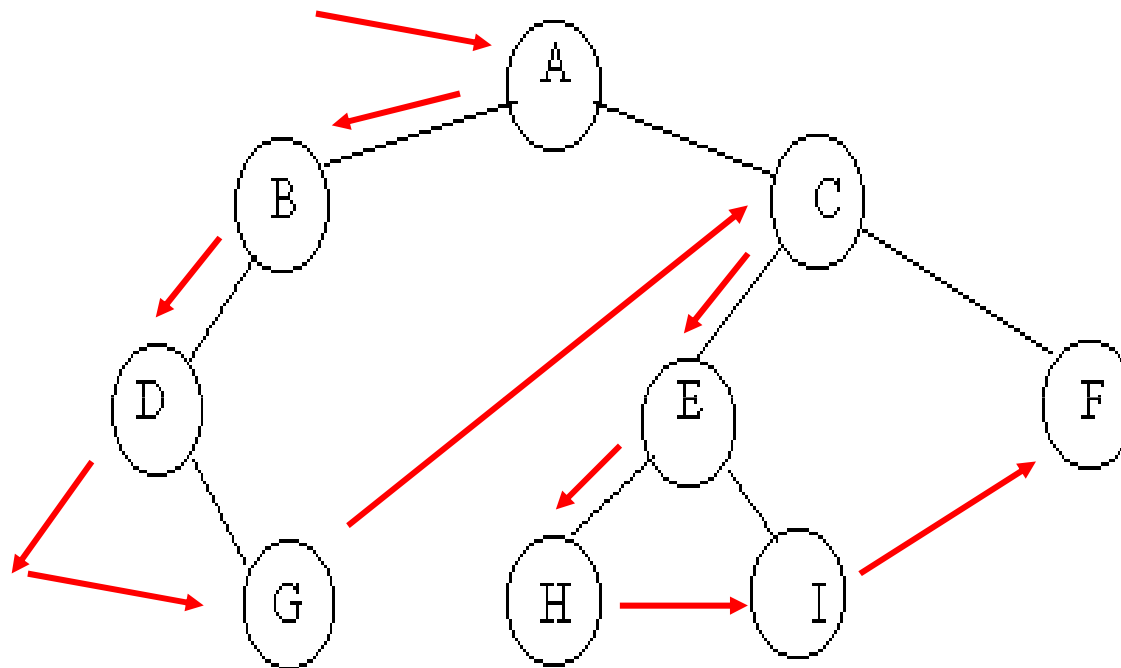
1. Kunjungan secara Preorder (Depth First Order), mempunyai urutan :
 - a. Cetak isi simpul yang dikunjungi (Simpul Akar)
 - b. Kunjungi Cabang Kiri
 - c. Kunjungi Cabang Kanan



A B D E C

Klik Animasi

Kunjungan PreOrder



Hasil : A B D G C E H I F

Klik Animasi

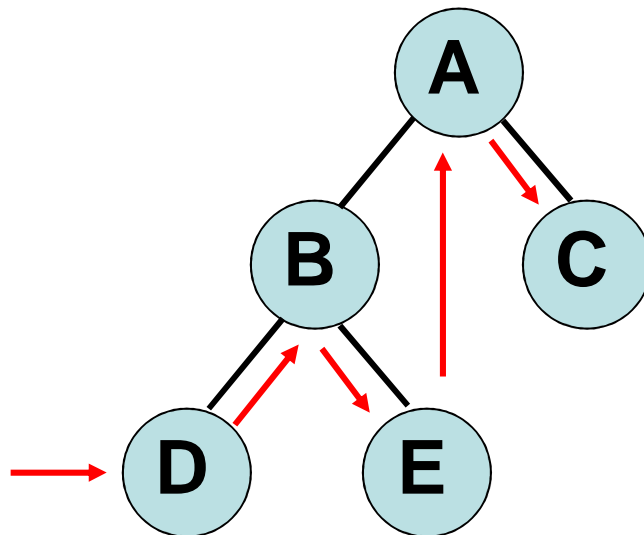
Kunjungan PreOrder dalam Program C++

```
void preOrder(Tree *root){
    if(root != NULL){
        printf("%d ",root->data);
        preOrder(root->left);
        preOrder(root->right);
    }
}
```

etric Order), mempunyai

urutan .

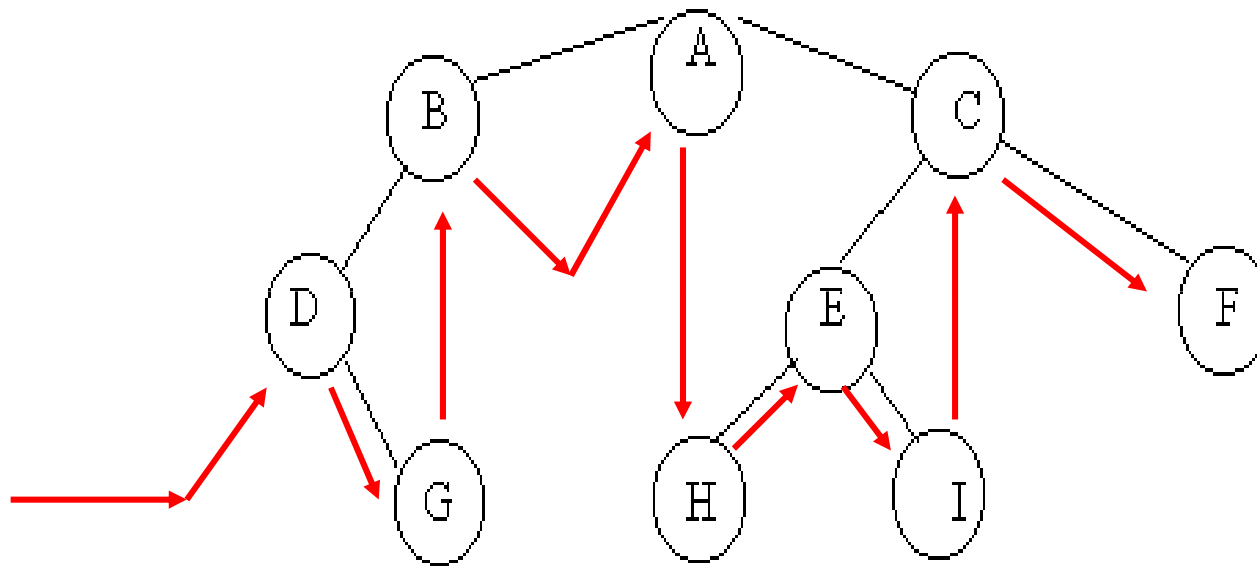
- a. Kunjungi Cabang Kiri
- b. Cetak isi simpul yang dikunjungi (Simpul Akar)
- c. Kunjungi Cabang Kanan



D B E A C

Klik Animasi

Kunjungan InOrder



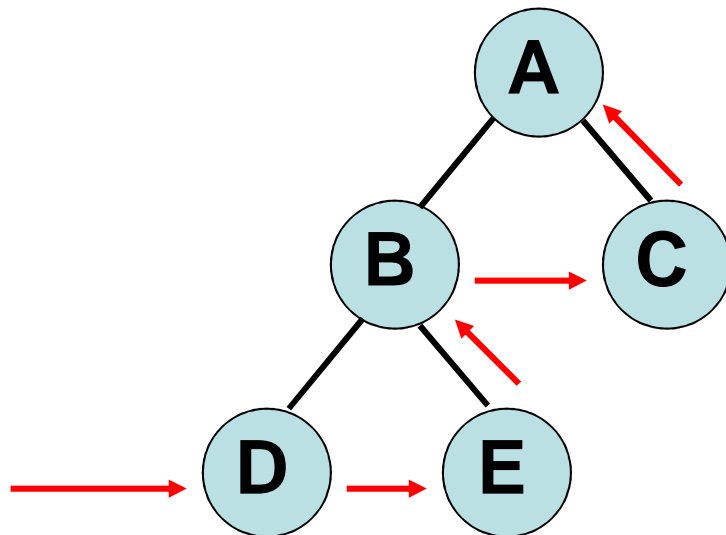
Hasil : D G B A H E I C F

Klik Animasi

Kunjungan InOrder dalam Program C++

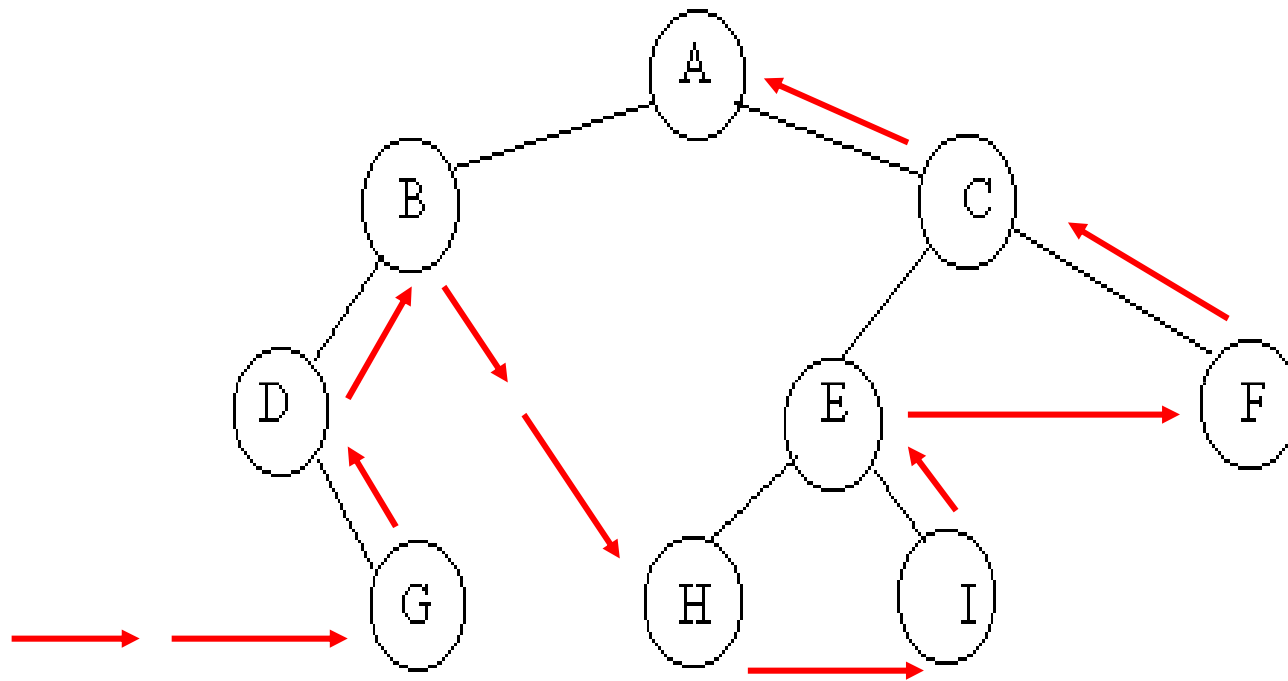
```
void inOrder(Tree *root) {  
    if (root != NULL) {  
        inOrder(root->left);  
        printf("%d ", root->data);  
        inOrder(root->right);  
    }  
}
```

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

Klik Animasi



Hasil : G D B H I E F C A

Klik Animasi

Kunjungan PostOrder dalam Program C++

```
void postOrder(Tree *root){  
    if(root != NULL){  
        postOrder(root->left);  
        postOrder(root->right);  
        printf("%d ", root->data);  
    }  
}
```

Kunjungan LevelOrder

Selain kunjungan yang dijelaskan diatas, masih ada satu macam kunjungan masih ada satu macam kunjungan lagi yaitu kunjungan LevelOrder.

Kunjungan dimulai dari simpul yang ada pada tingkat 1 (Akar), diteruskan pada simpul di tingkat 2, tingkat 3 dan seterusnya.

Secara singkat kunjungan Level Order ini dapat dijelaskan sebagai berikut.

1. Dimulai dengan memasukkan Akar kedalam antrean.
2. Kemudian mengeluarkan Akar tersebut keluar dari antrean.

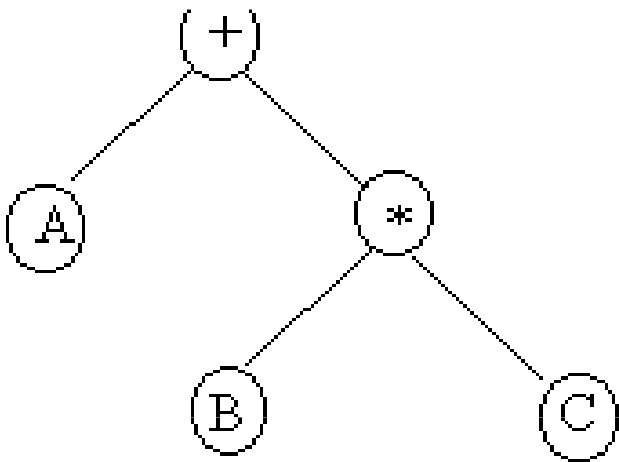
Pada saat Akar tersebut dikeluarkan dari antrean, cabang kiri dan cabang kanan secara berturut-turut dimasukkan dalam antrean.

Dengan kata lain jika suatu elemen dikeluarkan dari antrean, maka cabang kiri dan kanan dari elemen yang baru saja dikeluarkan dimasukkan kedalam antrean.

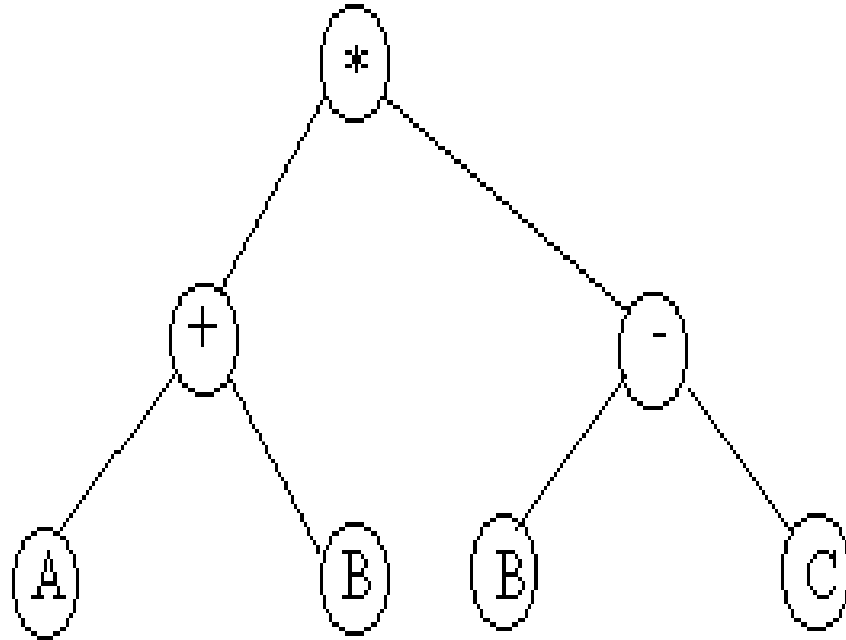
APLIKASI POHON BINER

NOTASI PREFIX, INFIX DAN POSTFIX

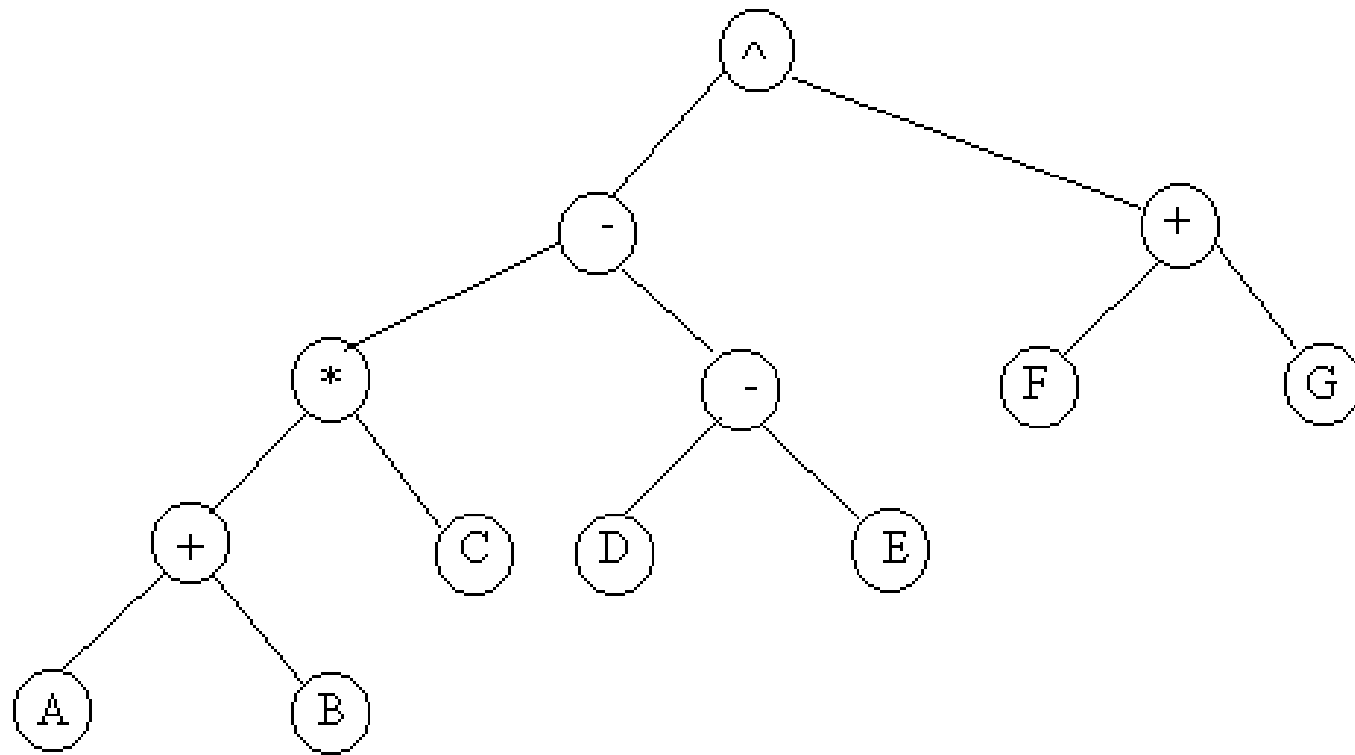
Pada bagian ini akan dibahas tentang bagaimana menyusun sebuah Pohon Binar yang apabila dikunjungi secara PreOrder akan menghasilkan Notasi Prefix, kunjungan secara InOrder menghasilkan Notasi Infix, dan kunjungan PostOrder menghasilkan Notasi Postfix.



(a)



(b)



(c)

Berdasarkan Gambar diatas, apabila dilakukan kunjungan secara PreOrder, maka akan diperoleh Notasi Prefix dari persamaan-persamaan yang digambarkan tersebut, yaitu :

$+A^*BC$ (Gambar.a)

$*+AB-BC$ (Gambar.b)

$^-*+ABC-DE+FG$ (Gambar.c)

Jika dilakukan kunjungan secara InOrder, akan diperoleh Notasi Infixnya, yaitu :

$(A+(B^*C))$ (Gambar.a)

$((A+B) * (B-C))$ (Gambar.b)

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

Jika dilakukan kunjungan secara PostOrder, akan diperoleh Notasi Postfixnya, yaitu :

ABC*+ (Gambar.a)

AB+BC-* (Gambar.b)

AB+C*DE-FG+^ (Gambar.c)