

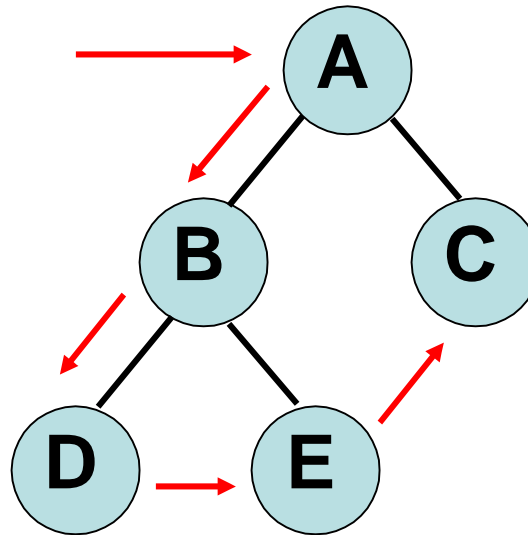
KUNJUNGAN PADA POHON BINER

KUNJUNGAN POHON BINER

- Kunjungan pada pohon biner merupakan salah satu operasi yang sering dilakukan pada pohon biner tepat satu kali (Binary Tree Traversal).
- Operasi pada kunjungan pohon biner terbagi menjadi 3 bentuk, yaitu:
 1. Kunjungan secara Preorder (Depth First Order)
 2. Kunjungan secara Inorder (Symetric Order)
 3. Kunjungan secara Postorder

KUNJUNGAN PREORDER (DEPTH FIRST ORDER)

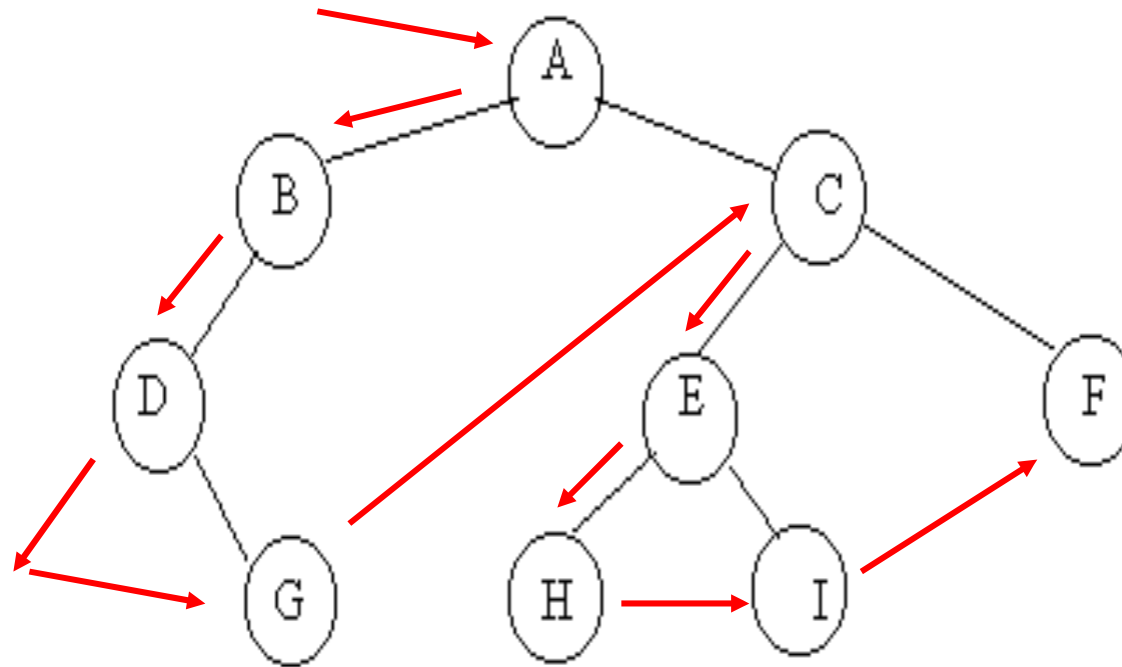
- Kunjungan secara Preorder (Depth First Order) memiliki urutan sebagai berikut:
 1. Cetak isi simpul yang dikunjungi (Simpul Akar)
 2. Kunjungi Cabang Kiri
 3. Kunjungi Cabang Kanan



A B D E C

KUNJUNGAN PREORDER (DEPTH FIRST ORDER)

Kunjungan PreOrder



Hasil : A B D G C E H I F

KUNJUNGAN PREORDER (DEPTH FIRST ORDER)

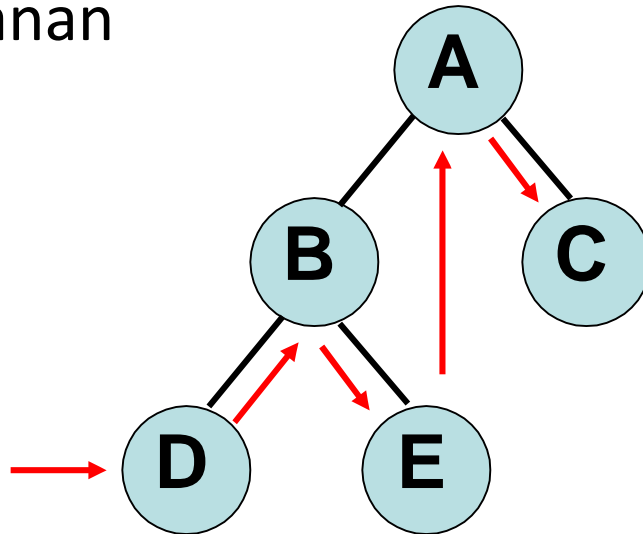
Contoh kode:

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

KUNJUNGAN INORDER (SYMETRIC ORDER)

- Kunjungan secara Inorder (Symetric Order) memiliki urutan sebagai berikut:

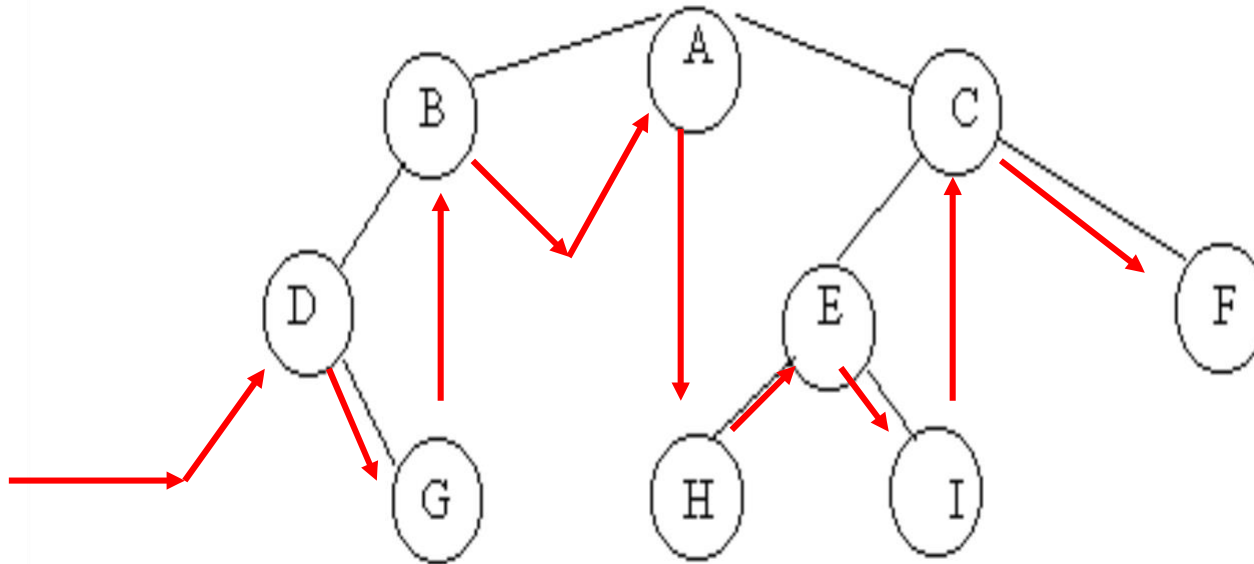
1. Kunjungi Cabang Kiri
2. Cetak isi simpul yang dikunjungi (Simpul Akar)
3. Kunjungi Cabang Kanan



D B E A C

KUNJUNGAN INORDER (SYMETRIC ORDER)

Kunjungan InOrder



Hasil : D G B A H E I C F

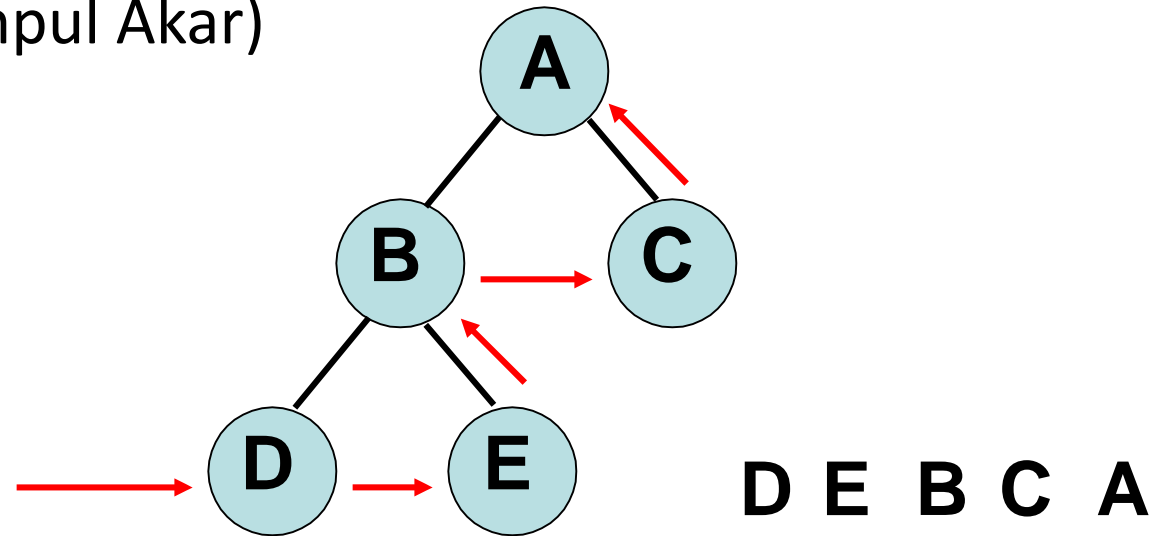
KUNJUNGAN INORDER (SYMETRIC ORDER)

Contoh kode:

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

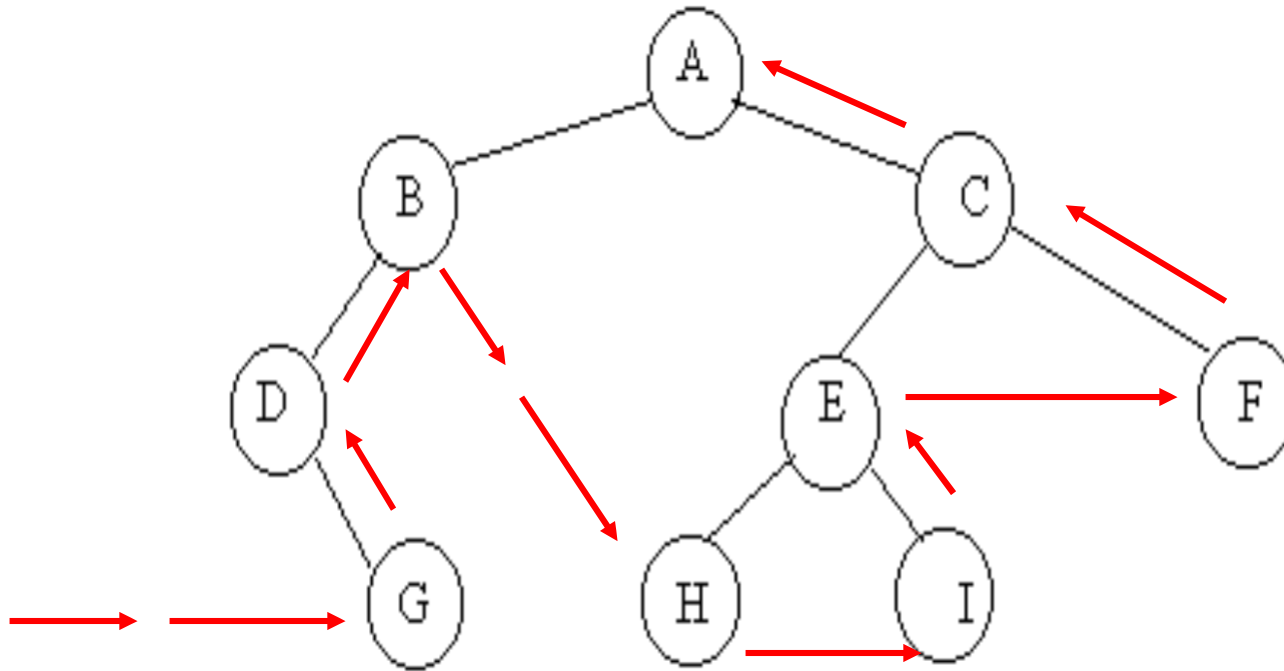

KUNJUNGAN SECARA POSTORDER

- Kunjungan secara Postorder memiliki urutan sebagai berikut:
 1. Kunjungi Cabang Kiri
 2. Kunjungi Cabang Kanan
 3. Cetak isi simpul yang dikunjungi (Simpul Akar)



KUNJUNGAN SECARA POSTORDER

Kunjungan PostOrder



Hasil : G D B H I E F C A

KUNJUNGAN SECARA POSTORDER

Contoh kode:

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

KUNJUNGAN POHON BINER

- Pada ketiga cara kunjungan pohon biner, kunjungan ke Cabang Kiri dilakukan terlebih dahulu, baru kemudian kunjungan ke Cabang Kanan. Dengan orientasi semacam ini 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).

KUNJUNGAN LEVEL ORDER

- Selain tiga bentuk kunjungan tersebut, masih ada satu macam kunjungan lagi yaitu kunjungan Level Order.
- Kunjungan dimulai dari simpul yang ada pada tingkat 1 (Akar), diteruskan pada simpul di tingkat 2, tingkat 3 dan seterusnya.

KUNJUNGAN LEVEL ORDER

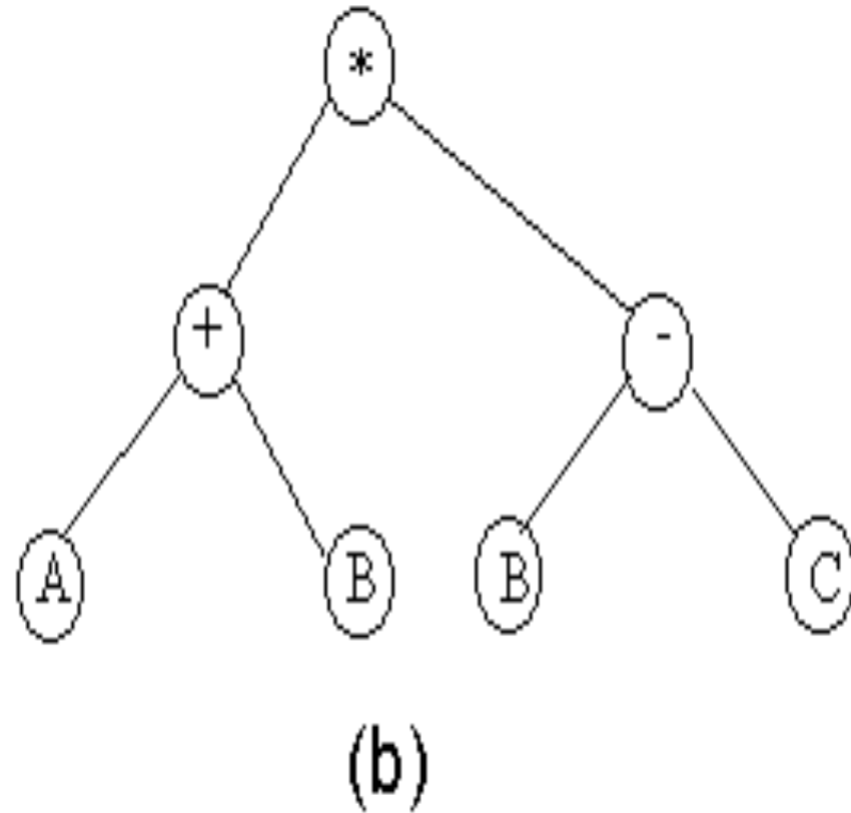
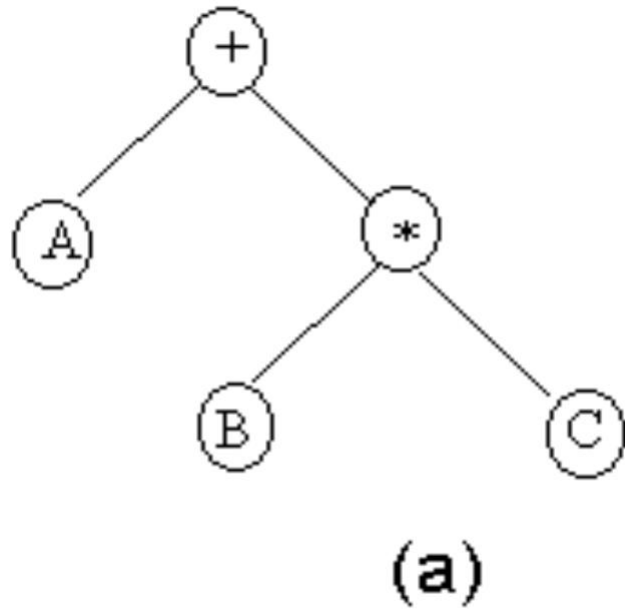
- Secara singkat kunjungan Level Order ini dapat dijelaskan sebagai berikut.
 1. Dimulai dengan memasukkan Akar ke dalam antrean.
 2. Kemudian mengeluarkan Akar tersebut keluar dari antrean.
- Pada saat Akar tersebut dikeluarkan dari antrean, cabang kiri dan cabang kanan secara berturut-turut dimasukkan 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

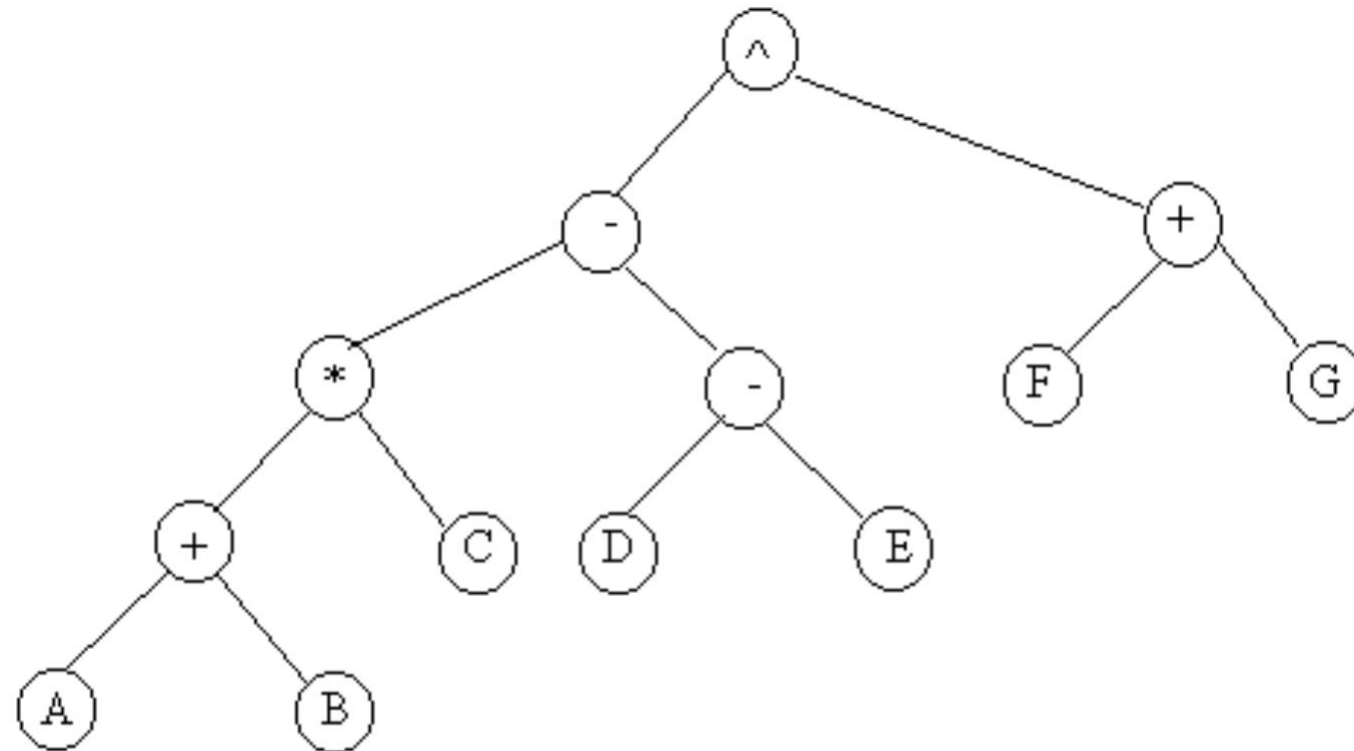
NOTASI PREFIX, INFIX DAN POSTFIX

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

APLIKASI POHON BINER



APLIKASI POHON BINER



(c)

APLIKASI POHON BINER

Berdasarkan Gambar sebelumnya, 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)

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

APLIKASI POHON BINER

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)

TERIMA KASIH