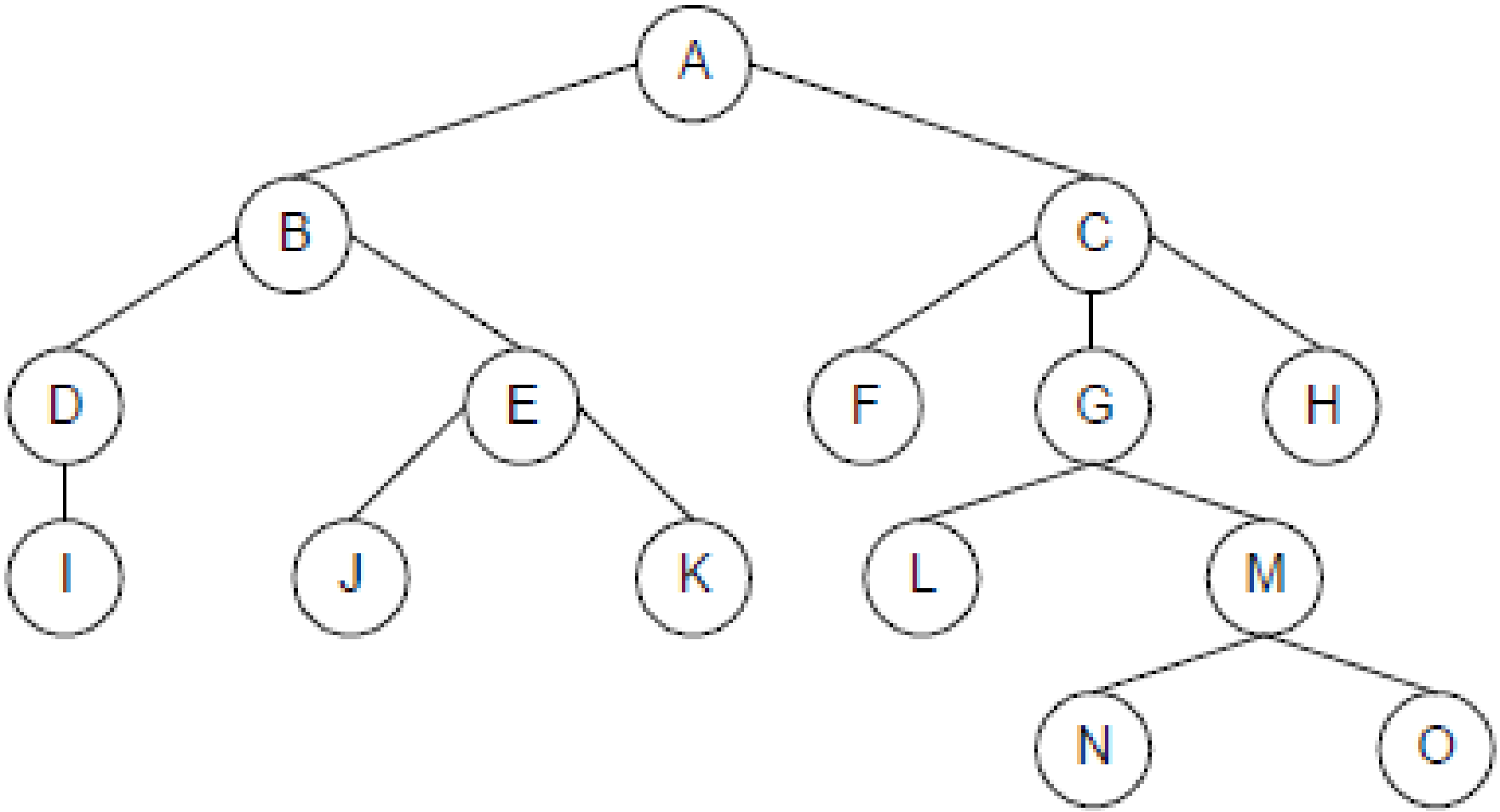
Tree

Brigida Arie Minartiningtyas, M.Kom

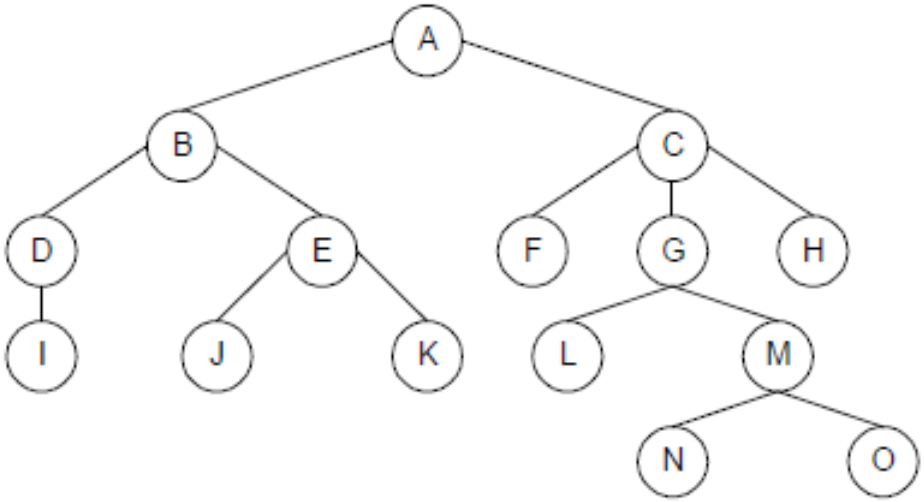
- Struktur data tak linier yg menggambarkan hubungan bersifat hirarkis antar elemen-elemen
- Kumpulan elemen-elemen yg memiliki salah satu elemennya adalah **akar** dan sisanya adalah **simpul**

Contoh Tree



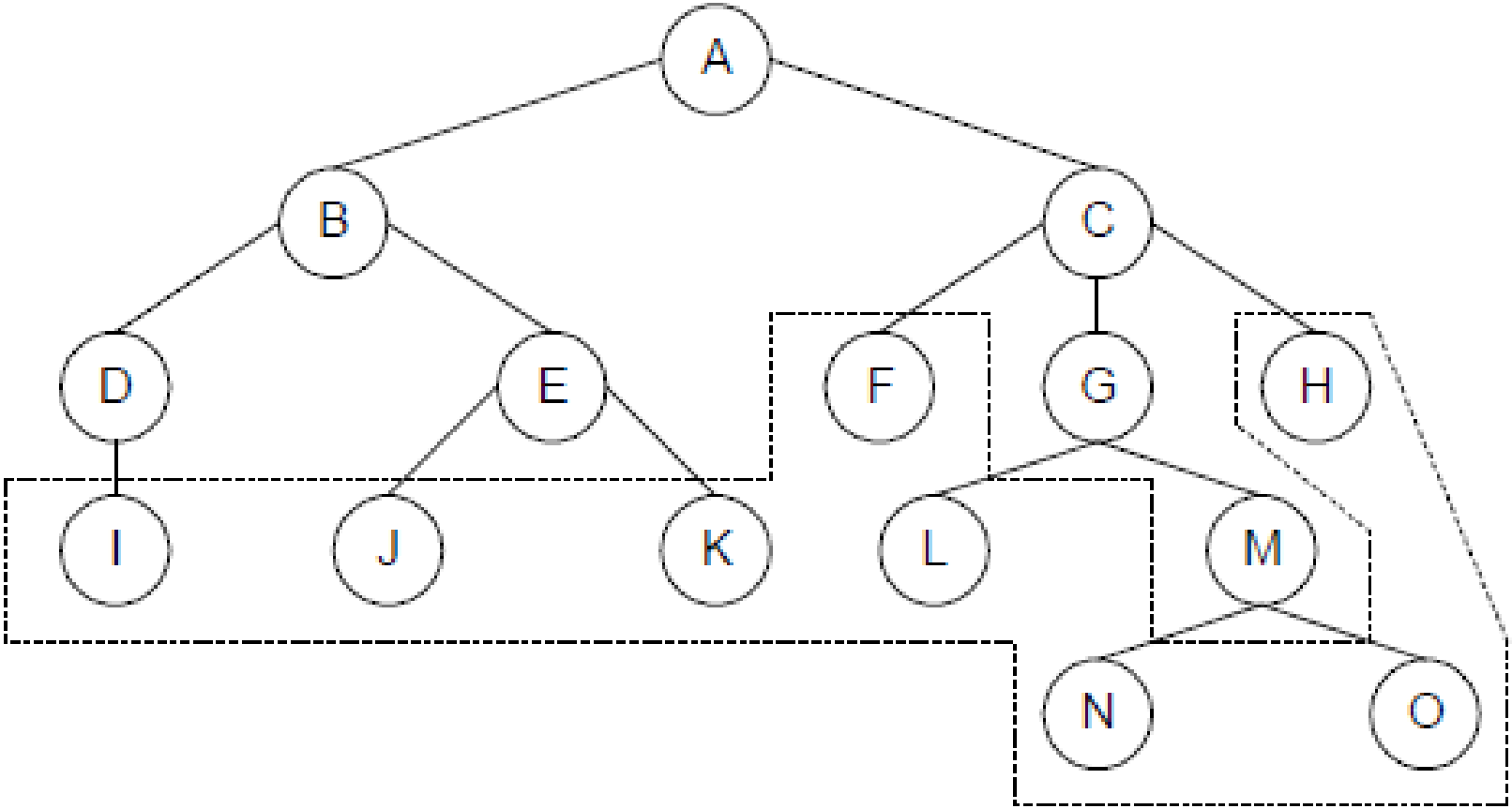
Derajad (degree)

- Derajad suatu simpul dinyatakan sebagai banyaknya anak atau turunan dari simpul tersebut.
  - simpul A mempunyai derajad 2,
  - simpul B mempunyai derajad 2,
  - simpul C berderajad 3



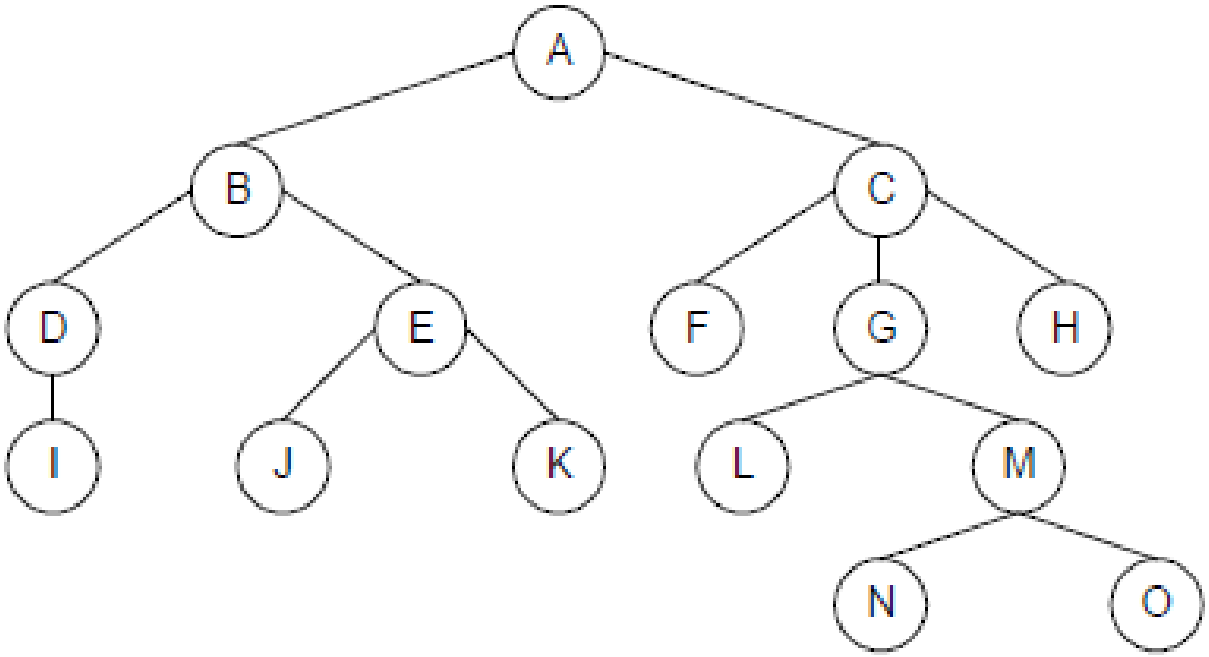
Daun (Leaf)

- Simpul-simpul F, H, I, J, K, L, N, O yang semuanya berderajad nol, disebut dengan daun (leaf)



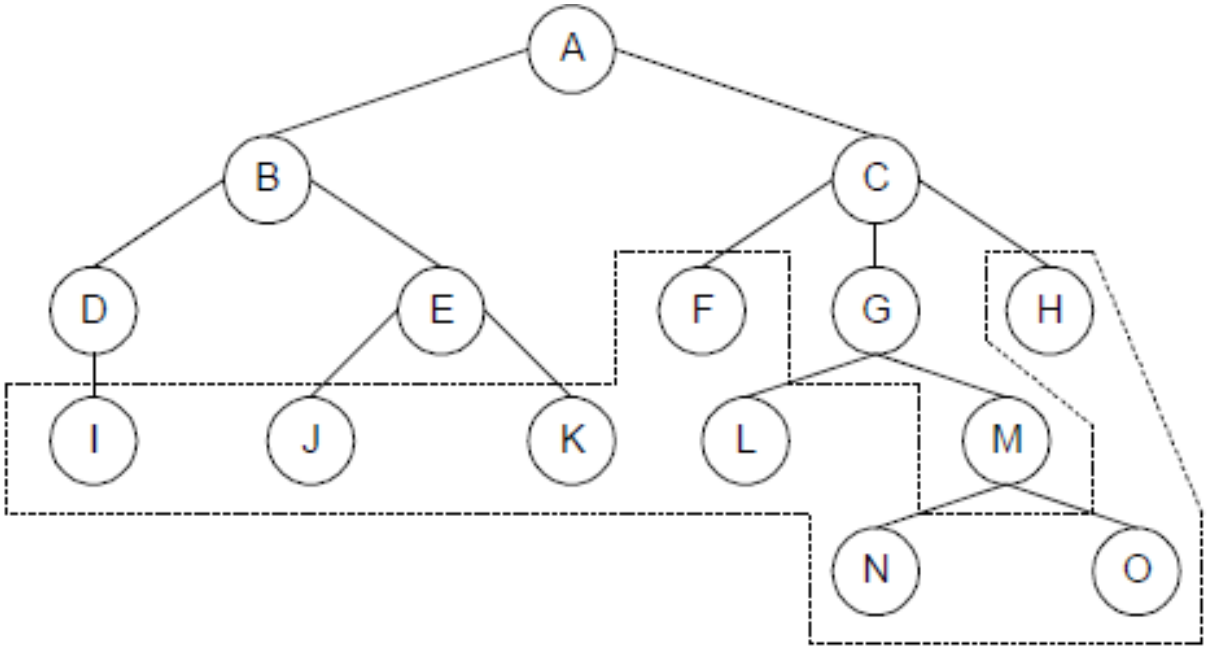
Tinggi (Height) / Kedalaman (Depth)

- Tingkat maksimum dari suatu pohon dikurangi dengan satu.
- Pohon mempunyai tinggi atau kedalaman sama dengan 4.



Hutan (Forest)

- Kumpulan sejumlah pohon yang tidak saling berhubungan.
- Dari gambar di atas jika kita menghapus simpul A maka akan terbentuk sebuah hutan.



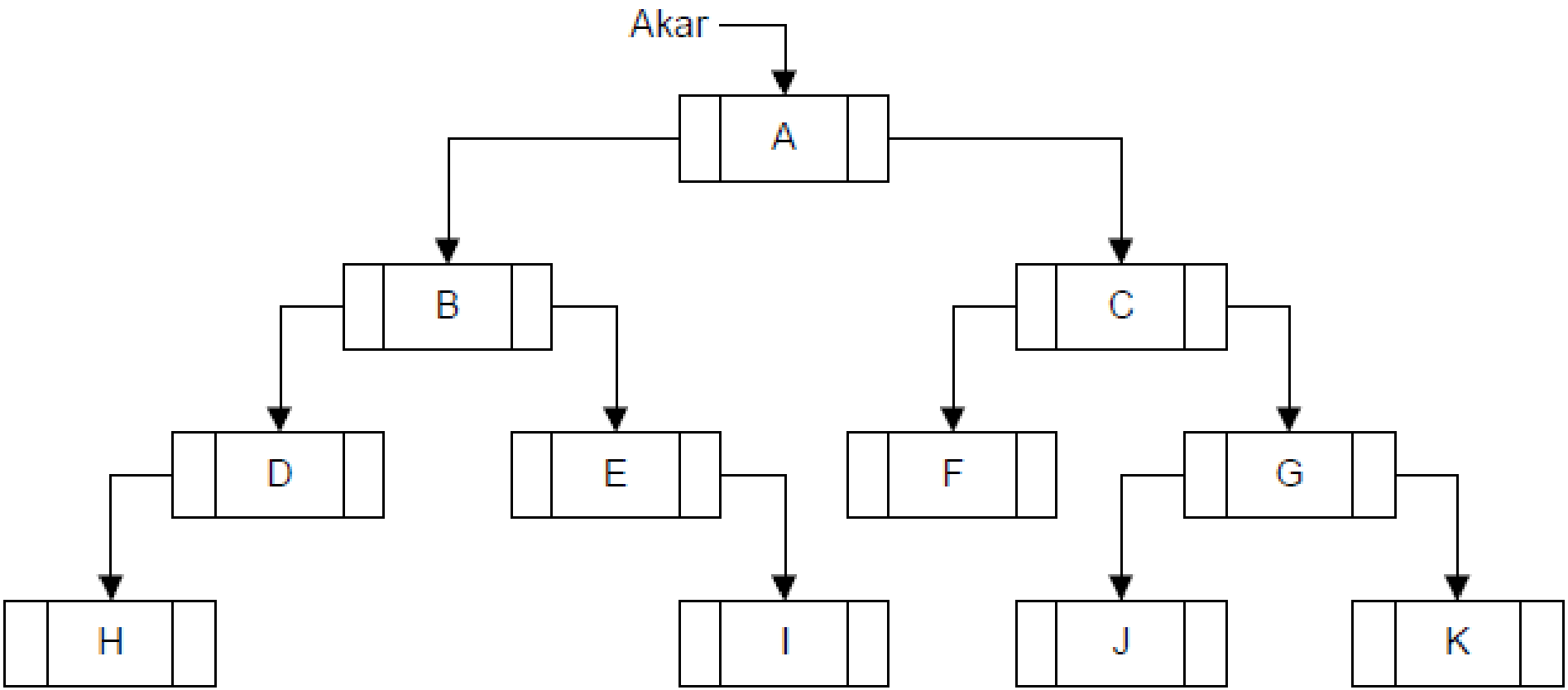
Pohon Biner (Binary Tree)

- Kumpulan simpul yang mungkin kosong atau mempunyai akar dan dua subpohon yang saling terpisah yang disebut dengan subpohon kiri dan sub pohon kanan.
- Subpohon disebut juga sebagai cabang.
- Karakteristik dari pohon biner ialah bahwa setiap simpul paling banyak hanya mempunyai dua buah anak.
- Dengan kata lain derajat tertinggi dari sebuah pohon biner adalah dua.
- Penyajian binary tree pada komputer di gunakan double link list.

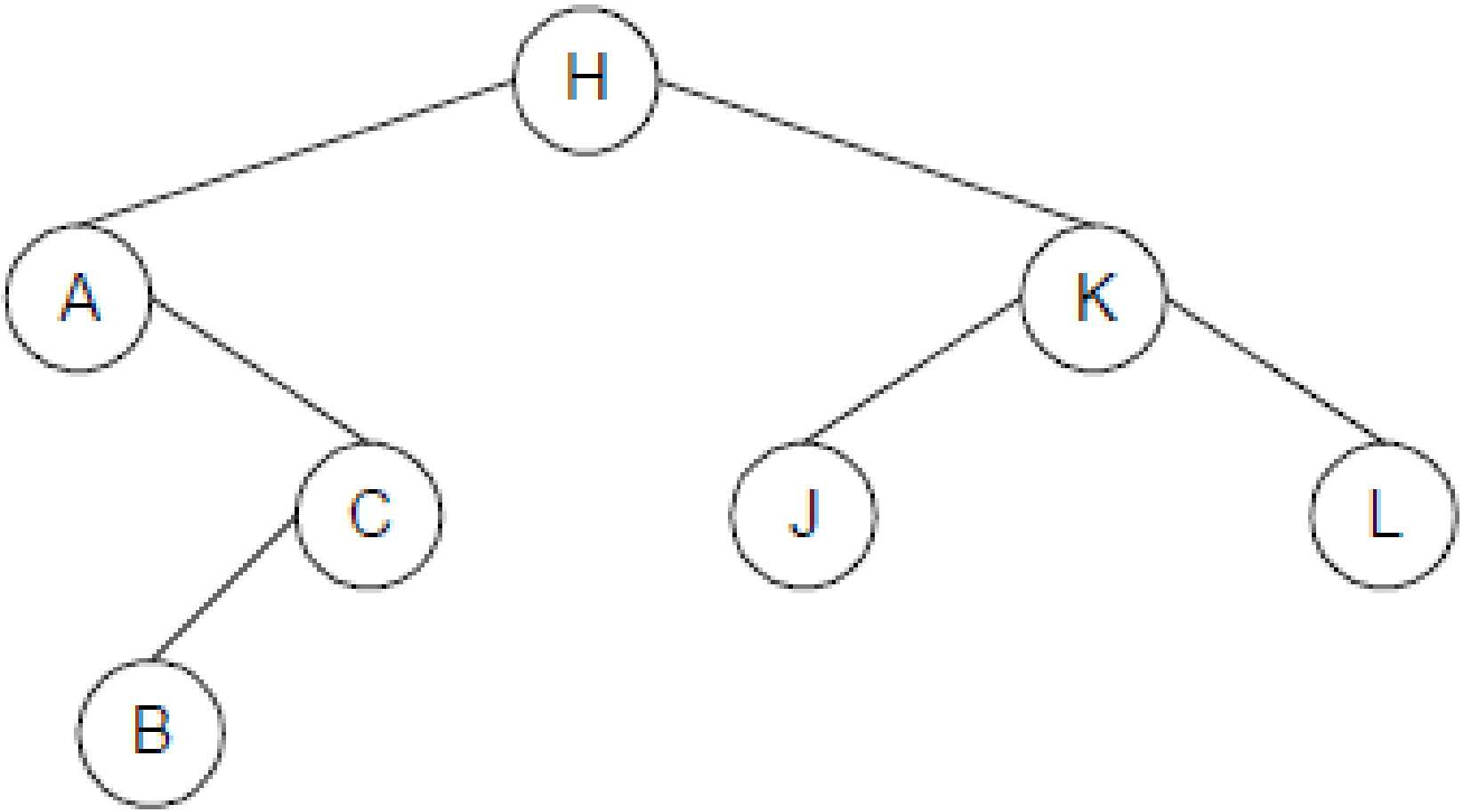
Deklarasi Pohon Biner

```
Type
  Tree = ^Simpul
  Simpul = Record
      Info : Tipe Data;
      Kiri : Tree;
      Kanan : Tree;
End;
```

Penyajian Pohon Biner



- HKACBLJ



Alokasi Simpul

```
Function BARU(Hrf : Char) : Tree;
  Var Temp : Tree;
  Begin
      New(Temp);
      Temp^.Info := Hrf;
      Temp^.Kiri := NIL;
      Temp^.Kanan := NIL;
      BARU := Temp;
  End;
```

Menempatkan dalam Pohon Biner

```
Procedure MASUK(Var Pohon : Tree; Hrf : Char);
  Begin
      If Pohon = NIL Then {Jika pohon masih kosong}
              Pohon := BARU(Hrf)
      Else
              Begin
                      If Pohon^.Info > Hrf {Jika Hrf lebih kecil, cabang kiri}
                              MASUK(Pohon^.Kiri,Hrf)
                      Else If Pohon^.Info < Hrf {Jika Hrf lebih besar, cabang kanan}
                              MASUK(Pohon^.Kanan,Hrf)
                      Else
                              Writeln('Karakter', Hrf, 'Sudah ada di
  Tree');
              End;
  End;
```

Kunjungan Pohon Biner

- Sebuah pohon biner memiliki operasi traversal yaitu suatu kunjungan pada suatu simpul tepat satu kali.
- Terdapat tiga jenis kunjungan pada pohon biner, yaitu :
  - PreOrder
  - InOrder
  - PostOrder

Pre Order

- Cetak isi simpul yang dikunjungi.
- Kunjungi cabang kiri.
- Kunjungi cabang kanan.

```
Procedure PREORDER(Temp : Tree);
        Begin
                If Temp <> NIL Then
                Begin
                        Write(Temp^.Info,' '); {Cetak isi simpul}
                        PREORDER(Temp^.Kiri); {Kunjungi cabang kiri}
                        PREORDER(Temp^.Kanan); {Kunjungi cabang kanan}
                End;
        End;
```

InOrder

- Kunjungi cabang kiri.
- Cetak isi simpul yang dikunjungi.
- Kunjungi cabang kanan.

```
Procedure INORDER(Temp : Tree);
        Begin
                If Temp <> NIL Then
                Begin
                        INORDER(Temp^.Kiri); {Kunjungi cabang kiri}
                        Write(Temp^.Info,' '); {Cetak isi simpul}
                        INORDER(Temp^.Kanan); {Kunjungi cabang kanan}
                End;
        End;
```

PostOrder

- Kunjungi cabang kiri.
- Kunjungi cabang kanan.
- Cetak isi simpul yang dikunjungi.

```
Procedure POSTORDER(Temp : Tree);
        Begin
                If Temp <> NIL Then
                Begin
                        POSTORDER(Temp^.Kiri); {Kunjungi cabang kiri}
                        POSTORDER(Temp^.Kanan); {Kunjungi cabang kanan}
                        Write(Temp^.Info,' '); {Cetak isi simpul}
                End;
        End;
```

STACK / TUMPUKAN

INFIX	POSTFIX	PREFIX
A + B	AB +	+ AB
A + B * C	ABC *+	*+ ABC

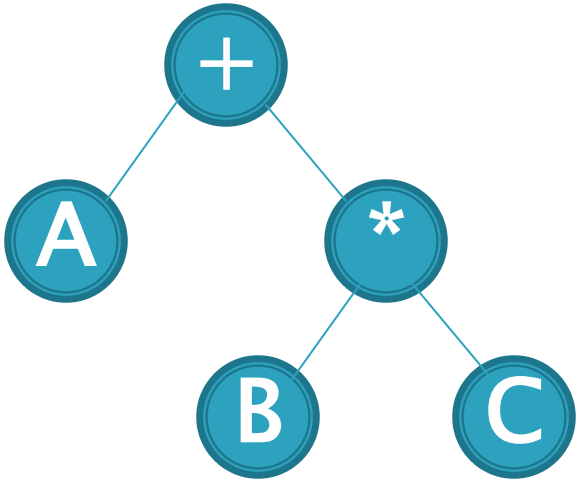
- **INFIX** = bentuk aritmatik yang Operatornya ada diantara (di dalam) dua buah Operand.
- **POSTFIX** = bentuk aritmatik yang Operatornya ada sesudah dua Operand.
- **PREFIX** = bentuk aritmatik yang Operatornya ada sebelum dua Operand.

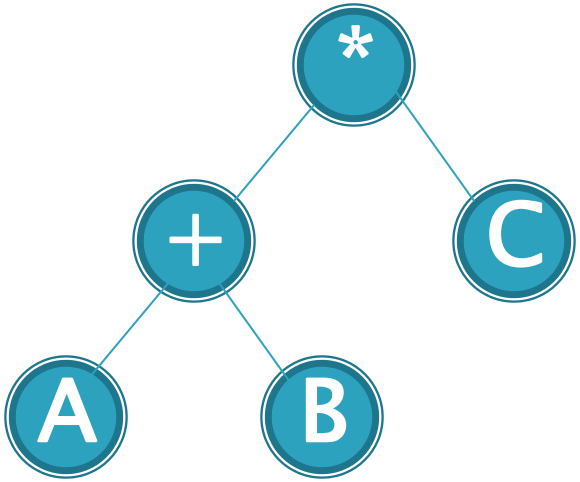
STACK / TUMPUKAN

- **<u>IN</u>FIX** =
- **<u>Pre</u>FIX** = AKAR – KIRI – KANAN
- **<u>Post</u>FIX** = KIRI – KANAN – AKAR

MENAMBAH SIMPUL

INFIX, PREFIX, POSTFIX





INFIX : A + B * C

PREFIX : + A * B C

POSTFIX : A B C * +

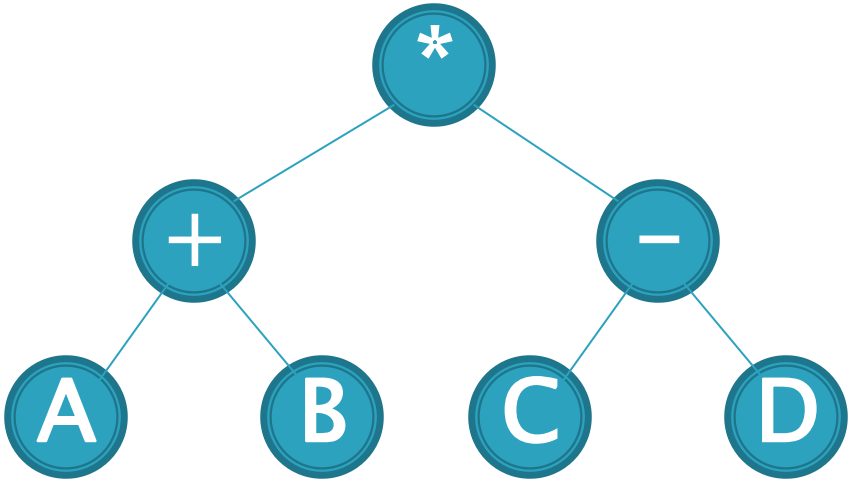
INFIX : (A + B) * C

PREFIX : * + A B C

POSTFIX : A B + C *

MENAMBAH SIMPUL

INFIX, PREFIX, POSTFIX



INFIX : (A + B) * (C–D)

PREFIX : * + A B – C D

POSTFIX : A B + C D – *