KUNJUNGAN POHON BINER

- **Pohon (**dalam struktur data)

  ➔ struktur berisi sekumpulan elemen dimana salah satu elemen adalah **akar** (**root**) dan elemen-elemen lain adalah bagian-bagian pohon yang membentuk susunan hirarki dengan akar sebagai awal mula.

- Elemen-elemen Pohon disebut **simpul** (**node**).

Definisi

Struktur pohon telah biasa digunakan dalam kehidupan sehari-hari seperti :

- Silsilah keluarga
- Daftar isi buku
- Struktur organisasi
- Pohon keputusan

Definisi Pohon Biner

- **Pohon biner** adalah bentuk graf yang terhubung yang tidak memiliki sirkuit dan pada pohon biner selalu terdapat *path* atau jalur yang menghubungkan dua simpul dalam pohon.

Terminologi Pohon Biner

Beberapa terminologi pada pohon biner :

- **Simpul akar (*root*)**

  simpul pohon dengan tingkatan tertinggi

- **Simpul daun (*leaf*)**

  simpul-simpul pada pohon yang **tidak lagi** memiliki simpul anak (*child*)

- **Induk (*parent*)**

  simpul yang merupakan induk dari children-nya

- **Anak dari simpul x**

  akar-akar (*root*) dari subpohon–subpohon dari simpul x adalah anak dari x

- ***Siblings***

  anak dari induk yang sama

Terminologi Pohon Biner

Beberapa terminologi pada pohon biner :

- **Moyang (*anchestor*)**

  simpul-simpul disepanjang jalur dari simpul ke *root*

- **Level suatu *node***

  jika simpul pada level p, maka *children*-nya adalah berada pada level p + 1

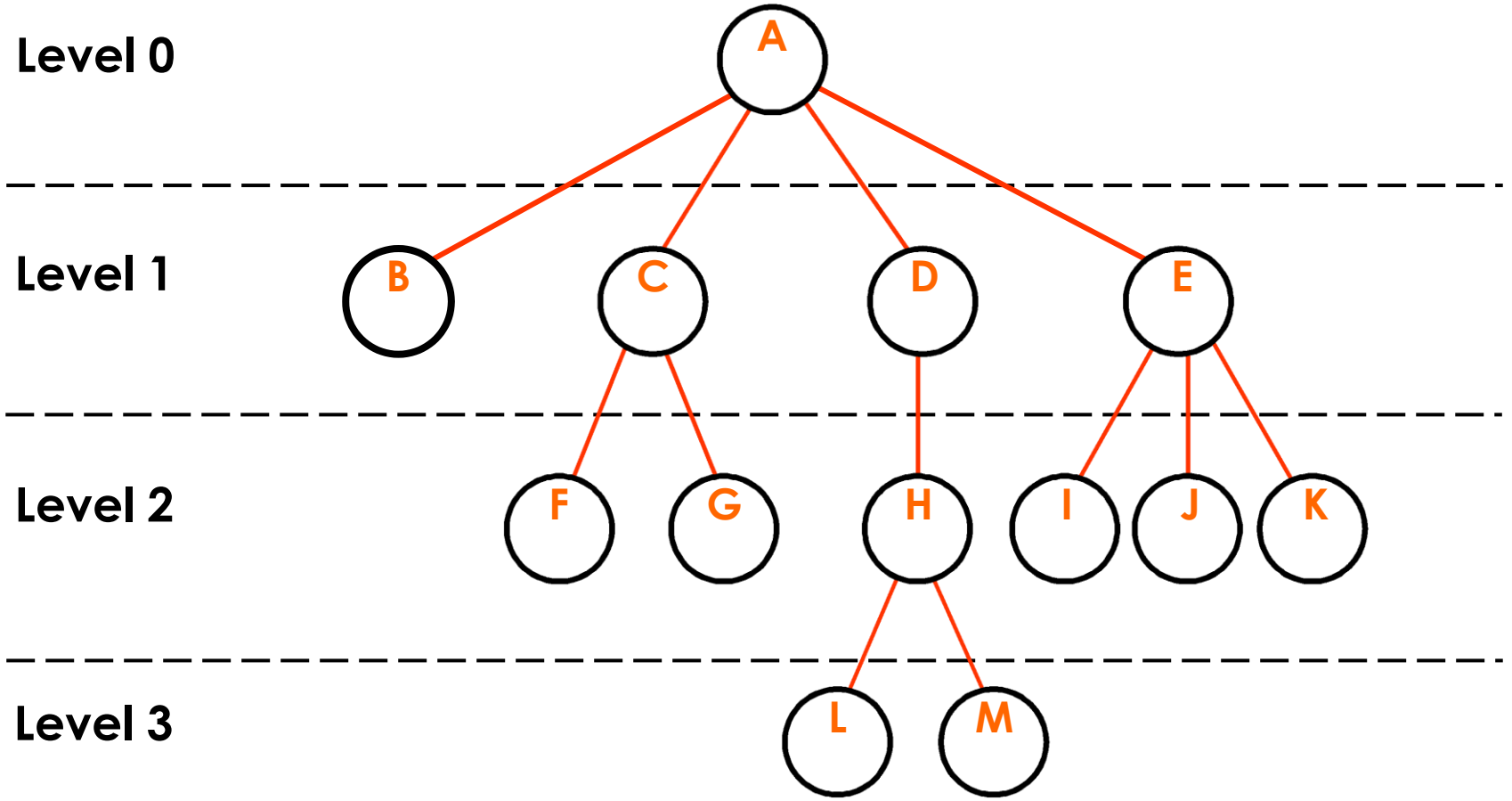
- ***Height* atau *depth***

  Pohon memiliki ketinggian (height) atau kedalaman yang merupakan level tertinggi + 1.

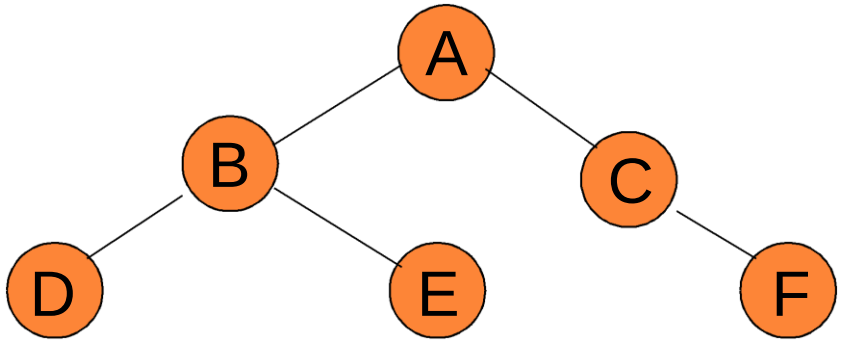
- ***Weight***

  Pohon memiliki berat (weight) yang merupakan banyaknya Daun pada Pohon.

Terminologi Pohon Biner



CONTOH :



Dari contoh diatas diperoleh :

Banyak simpul (*node*)	:	6 (n)
Banyak ruas (*edge*)	:	5 (n-1)
Root	:	A
Leaf	:	D, E dan F
Parent	:	A parent dari B dan C
		B parent dari D dan E
Level	:	0 = A
		1 = B dan C
		2 = D, E dan F
Height (ketinggian)	:	level tertinggi + 1
		2 + 1 = 3

BINARY TREE (POHON BINER)

- Jumlah maximum tingkatan simpul dari pohon biner adalah 2, jika ada penambahan dilakukan penelusuran ke kiri dan ke kanan yang ditetapkan sebagai subpohon.

BINARY TREE (POHON BINER)

- Jumlah maximum tingkatan simpul dari pohon biner adalah 2, jika ada penambahan dilakukan penelusuran ke kiri dan ke kanan yang ditetapkan sebagai subpohon.

- Dua pohon biner bisa dikatakan sama jika keduanya memiliki struktur yang sama.

- Dua pohon biner dikatakan equivalent jika keduanya sama dan berisi informasi yang sama.

Pohon Umum dan Pohon Biner

- Pohon Umum
  - Pohon yang simpulnya terhubung lebih dari 2 simpul anak
  - Pohon umum tidak dapat diproses komputer dan harus dijadikan pohon biner

- Algoritma untuk mengubah pohon umum ke pohon biner
  1. Hubungkan semua simpul yang bersaudara 1 parent
  2. Hapus ruas yang terhubung ke setiap simpul anak, kecuali ruas yang paling kiri
  3. Ruas mendatar hasil penambahan, diputar searah jarum jam sebesar 45 derajat (1/8 lingkaran)

Sequential Search

- Proses mengunjungi melalui satu pohon dengan cara setiap simpul dikunjungi hanya satu kali yang disebut tree travesal (kunjungan pohon)
- Aktifitas sequential search adalah Mengunjungi akar / root, Menelusuri subtree kiri, dan Menelusuri subtree kanan dari sebuah pohon biner.

BINARY SEARCH TREE SEBAGAI INDEKS

- Pohon juga berguna untuk menggambarkan sekumpulan data yang memiliki cabang struktur logik.
- Contoh pohon biner yang menggambarkan pernyataan aritmatika :

REVIEW

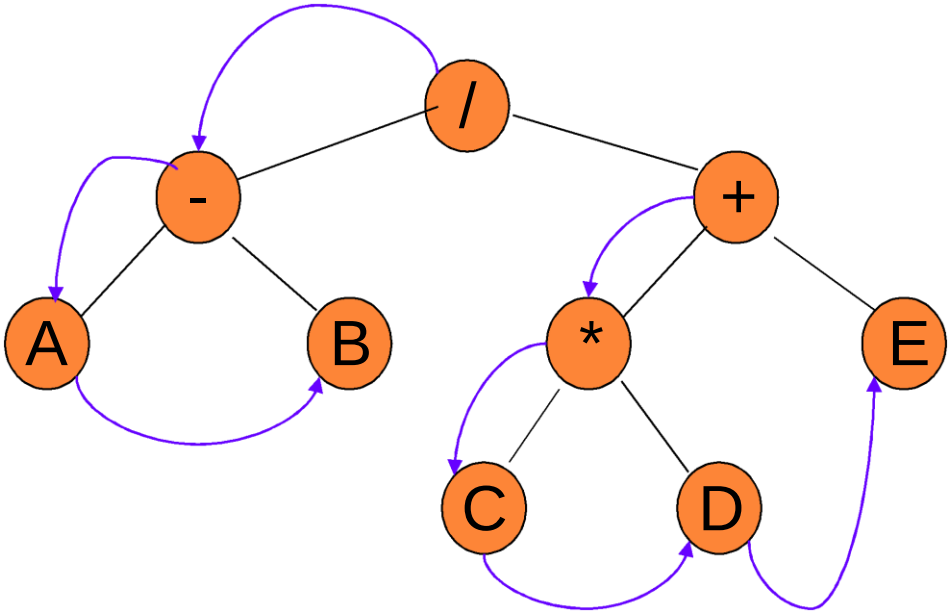
INFIX	POSTFIX	PREFIX
A + B	A B +	+ A B
A – B * C	A B C * -	- A * B C
A – B * C ^ D	A B C D ^ * -	- A * B ^ C D

Metode Sequential Search (1)

- Traversal PRE-ORDER
  1. Kunjungi akar
  2. Menelusuri subtree kiri dalam pre-order
  3. Menelusuri subtree kanan dalam pre-order

TRAVERSAL IN-ORDER (PREFIX)

- Notasi infix : ((A – B) / (C * D) + E)



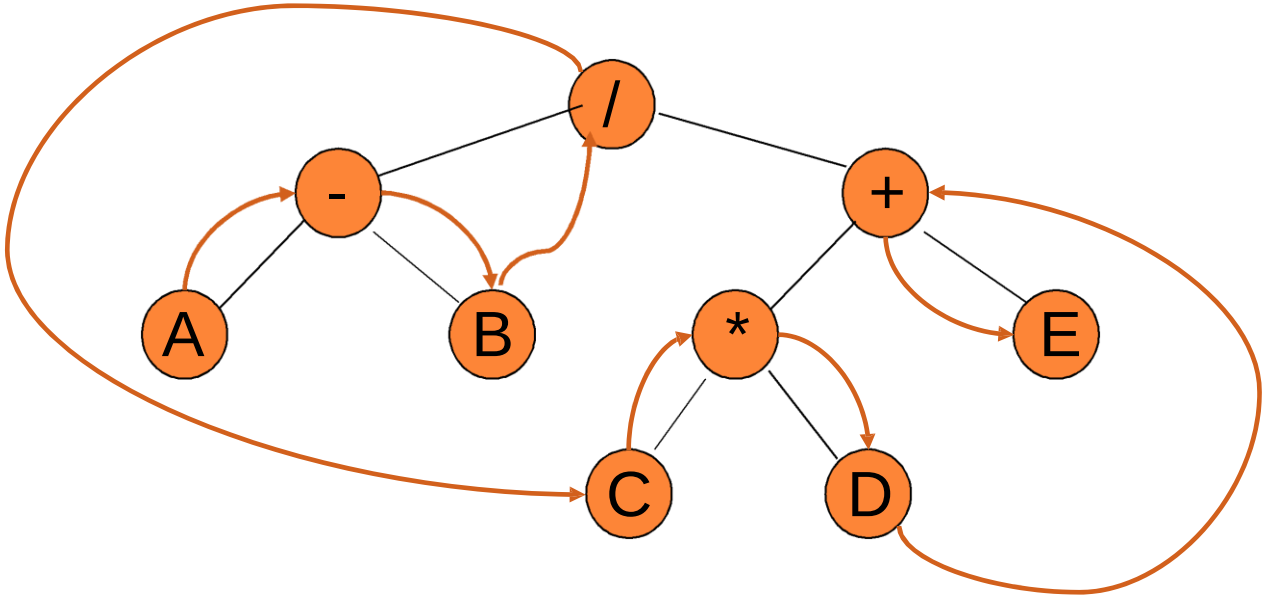
tentukan transversal pre-order (prefix) pohon biner nya, yaitu dari atas ke bawah (**akar-kiri-kanan**), dinyatakan dengan urutan simpul bertanda panah, mulai dari awal yaitu dari root sampai akhir simpul yaitu simpul E.

Dengan urutan pre-order (prefix) : / - A B + * C D E.

Metode Sequential Search (2)

- Traversal IN-ORDER
  1. Menelusuri subtree kiri dalam in-order
  2. Kunjungi akar
  3. Menelusuri subtree kanan dalam in-order

TRAVERSAL IN-ORDER (INFIX)

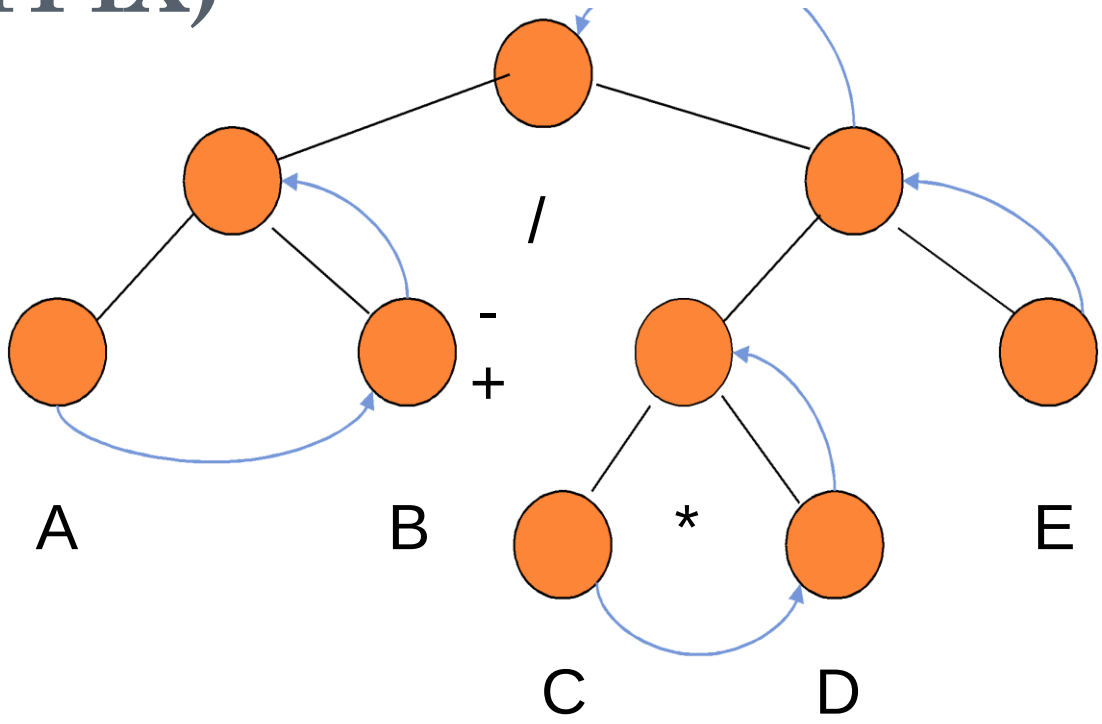


- Dari pohon tentukan transversal yaitu berturut-turut dari kiri ke tengah dan ke kanan (**kiri-akar-kanan**).
- Dari panah yang menunjukkan urutan simpul tersebut didapat transversal in-order (infix)
- Dengan urutan : A – B / C * D + E

Metode Sequential Search (3)

- Traversal POST-ORDER
  1. Menelusuri subtree kiri dalam post-order
  2. Menelusuri subtree kanan dalam post-order
  3. Kunjungi akar

TRAVERSAL POST-ORDER (POSTFIX)



- Dari urutan transversal post-order (postfix) dapat kita tentukan dari simpul bawah yaitu A sampai simpul atas yaitu bagi (/), (**kiri-kanan-akar**) dapat kita lihat urutannya yaitu
- A B – C D * E + /