

Buatlah ADT Non Binary Tree dengan alokasi statis / representasi Kontigu (menggunakan Array) dengan Struktur data (setiap node) tddr subvar :
 info : var tunggal beritpe Integer (boleh diganti karakter)
 posisi_parent, posisi_first_son, posisi_next_brother : var tunggal betipe integer
 (jadi menggunakan array bertipe komposit)

Spesifikasi (*.h) adalah sebagai berikut :

```
#ifndef nbtrees_h
#define nbtrees_h

#include "boolean.h"

//Maksimal node yang dapat ditampung dalam array
#define jml_maks 20

//Arah traversal

#define kebawah 1
#define keatas 2
#define kekanan 3
#define nil 0

/*****
/* Type data */
*****/

/* Indeks dalam bahasa C dimulai dengan 0; tetapi indeks 0 tidak
dipakai */

/* Kamus */

typedef char infotype;
typedef int address;
typedef struct { infotype info;
                address ps_fs, ps_nb, ps_pr;}nbtree;

typedef nbtree Isi_Tree[jml_maks+1];

/*****
/* S P E S I F I K A S I */
*****/

void Create_tree(Isi_Tree X, int Jml_Node);
// Create Non Binary Tree sebanyak Jml_Node
// Tujuan mengentrykan Non Binary Tree ke array Isi_Tree dengan pola
Level Order
// Jml Node adalah banyaknya elemen dalam Tree yang menjadi parameter
input
```

```

boolean IsEmpty (Isi_Tree P);
/* Mengirimkan true jika Isi_Tree KOSONG */

/***** Traversal *****/

void PreOrder (Isi_Tree P);
/* Traversal PreOrder */
/* IS : P terdefinisi */
/* FS : Semua simpul P sudah ditampilkan secara PreOrder : Parent,
fs, nb */

void InOrder (Isi_Tree P);
/* Traversal InOrder */
/* IS : P terdefinisi */
/* FS : Semua simpul P sudah ditampilkan secara InOrder : fs, Parent,
Sisa anak lain */

void PostOrder (Isi_Tree P);
/* Traversal PostOrder */
/* IS : P terdefinisi */
/* FS : Semua simpul P sudah ditampilkan secara PostOrder : fs, nb,
Parent */

void Level_order(Isi_Tree X, int Maks_node);
/* Traversal LevelOrder */
/* IS : P terdefinisi */
/* FS : Semua simpul P sudah ditampilkan secara Level Order */

void PrintTree (Isi_Tree P);
/* IS : P terdefinisi */
/* FS : Semua simpul P sudah ditulis ke layar */

/***** Search *****/

boolean Search (Isi_Tree P, infotype X);
/* Mengirimkan true jika ada node dari P yang bernilai X */

/***** Fungsi Lain *****/

int nbElmt (Isi_Tree P);
/* Mengirimkan banyak elemen (node) di pohon non biner P */

int nbDaun (Isi_Tree P);
/* Mengirimkan banyak daun (node) pohon non biner P */

int Level (Isi_Tree P, infotype X);
/* Mengirimkan level dari node X yang merupakan salah satu simpul */
/* dr pohon P. Akar (P) levelnya adalah 0. Pohon tidak kosong */

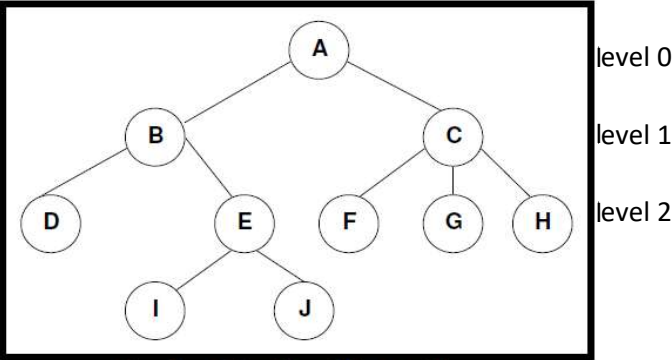
int Depth (Isi_Tree P);
/* Pohon Biner mungkin Kosong, mengirimkan 'depth' yaitu tinggi dari
Pohon */
/* Basis : Pohon Kosong, tingginya Nol */

int Max (infotype Data1, infotype Data2);
/* Mengirimkan Nilai terbesar dari dua data */

#endif

```

Misal akan dibentuk tree sbb :



Gambar 18. Non Binary Tree

Maka ilustrasi Final State situasi alokasi statis untuk array NBTS nya adalah sbb : (ilustrasi FS sampai inset node di level 2)

Array NBTS :

subvar :	info	fs	nb	pr	info	fs	nb	pr	info	fs	nb	pr	info	fs	nb	pr	info	fs	nb	pr	info	fs	nb	pr	info	fs	nb	pr	...	info	fs	nb	pr				
	'A'	2	0	0	'B'	4	3	1	'C'	6	0	1	'D'	0	5	2	'E'	0	0	2	'F'	0	7	3	'G'	0	8	3		'H'	0	0	3				
no indeks	1				2				3				4				5				6				7				8				...	20			