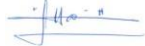




NASKAH UJIAN



☐ UTS ☒ UAS ☐ Susulan UTS/ UAS ☐ Lain-lain : [Ganjil/ Genap] 2020/2021

KMK - Mata Kuliah	: TIB11-STRUKTUR DATA	NIM	:
Kelas	: 2PTI1,2PTI2, 2PTI51	Nama Mahasiswa	:
Hari / Tanggal	: Selasa/8 Juni 2021		
Waktu Ujian	: 100 Menit (08:00-09:40)		
Sifat Ujian	: Buka/Tutup Buku/ Kamus / Take Home	Diperiksa oleh:	Tanda Tangan :
Lembar Jawaban	: Ya/ Tidak		
Laptop	: Ya / Tidak	(Lukman Hakim)	
Kalkulator	: Ya / Tidak	Tanggal :8-05-2021	

Naskah ujian harap dimasukkan kedalam lembar jawaban dan dikumpulkan kembali !!!

No	Jawablah Pertanyaan di bawah ini pada Lembar Jawaban Yang telah disediakan Untuk kendala Ujian dapat kirimkan ke email prodi: teknikinformatika.uas@ubm.ac.id	Bobot %
1	<p>Buatlah sebuah program dengan array yang sudah terinisialisasi pada saat deklarasi dimana array adalah sebagai berikut :</p> <pre>int a[20] = {12, 35, 41, 23, 22, 10, 99, 21, 50, 40, 39, 48, 24, 11, 66, 51, 55, 44, 33, 22};</pre> <p>program yang anda buat harus menyusun data-data tersebut pada sebuah binary tree secara left-to-right order, sementara menyusun pada binary tree program harus menampilkan alamat memory yang ditempati simpul tersebut dan alamat dari simpul yang menjadi parent dari simpul tersebut!</p> <p>(jika error tampilkan saja screenshot errornya dan tetap copas teks kode program nya!!!)</p> <p>Jawab:</p> <p>Copas jawaban anda di bawah ini dalam bentuk teks</p> <pre>// Jawaban #include <iostream> using namespace std; struct node{ int info; node* left, *right; }; class BinaryTree{ private: node *root, *temp, *before; public: BinaryTree(){ root = nullptr; temp = nullptr; } void inserts(int A){ temp = root; if (root == nullptr){ root = new node; cout << "Pemasukan data: " << A << '\n';</pre>	25%

```

        cout << "Alamat parent: " << root << '\n';
        cout << "Alamat simpul: " << root << '\n';
        root->info = A;
        root->right = nullptr;
        root->left = nullptr;
    }else{ //left to right
        while (true){
            if (A > temp->info){
                if (temp->right != nullptr){
                    temp = temp->right;
                    continue;
                }
                cout << "Pemasukan data: " << A << '\n';
                cout << "Alamat parent: " << temp << "
data parent : " << temp->info<< '\n';
                temp->right = new node;
                temp = temp->right;
                cout << "Alamat simpul: " << temp <<
'\n';

                temp->right = nullptr;
                temp->left = nullptr;
                temp->info = A;
                break;
            }else if (A < temp->info){
                if (temp->left != nullptr){
                    temp = temp->left;
                    continue;
                }
                cout << "Pemasukan data: " << A << '\n';
                cout << "Alamat parent: " << temp << "
data parent : " << temp->info<< '\n';
                temp->left = new node;
                temp = temp->left;
                cout << "Alamat simpul: " << temp <<
'\n';

                temp->right = nullptr;
                temp->left = nullptr;
                temp->info = A;
                break;
            }
            if (A == temp->info) break;
        }
        //artinya data yang dimasukan sdh terdapat di tree
    }
}

};

int main(){
int a[20] = {12, 35, 41, 23, 22, 10, 99, 21, 50, 40, 39, 48, 24,
11, 66, 51, 55, 44, 33, 22};
    BinaryTree Subaru;
    for (int i = 0; i < 20 ; i++){

```

```

        Subaru.inserts(a[i]);
        cout << '\n';
    }

    return 0;}

```

//Akhir kode program

Copas Screenshoot hasil
running program anda di
bawah ini

```

Pemasukan data: 12
Alamat parent: 0x1b1770
Alamat simpul: 0x1b1770

Pemasukan data: 35
Alamat parent: 0x1b1770 data parent : 12
Alamat simpul: 0x1b1790

Pemasukan data: 41
Alamat parent: 0x1b1790 data parent : 35
Alamat simpul: 0x1b19d0

Pemasukan data: 23
Alamat parent: 0x1b1790 data parent : 35
Alamat simpul: 0x1b19f0

Pemasukan data: 22
Alamat parent: 0x1b19f0 data parent : 23
Alamat simpul: 0x1b1a10

Pemasukan data: 10
Alamat parent: 0x1b1770 data parent : 12
Alamat simpul: 0x1b5b70

Pemasukan data: 99
Alamat parent: 0x1b19d0 data parent : 41
Alamat simpul: 0x1b5b90

Pemasukan data: 21
Alamat parent: 0x1b1a10 data parent : 22
Alamat simpul: 0x1b5bb0

Pemasukan data: 50
Alamat parent: 0x1b5b90 data parent : 99
Alamat simpul: 0x1b5bd0

Pemasukan data: 40
Alamat parent: 0x1b19d0 data parent : 41
Alamat simpul: 0x1b5bf0

Pemasukan data: 39
Alamat parent: 0x1b5bf0 data parent : 40
Alamat simpul: 0x1b5c10

Pemasukan data: 48
Alamat parent: 0x1b5bd0 data parent : 50
Alamat simpul: 0x1b5c30

Pemasukan data: 24
Alamat parent: 0x1b19f0 data parent : 23
Alamat simpul: 0x1b5e20

Pemasukan data: 11
Alamat parent: 0x1b5b70 data parent : 10
Alamat simpul: 0x1b5d40

Pemasukan data: 66
Alamat parent: 0x1b5bd0 data parent : 50
Alamat simpul: 0x1b5d00

Pemasukan data: 51
Alamat parent: 0x1b5d00 data parent : 66
Alamat simpul: 0x1b5cc0

Pemasukan data: 55
Alamat parent: 0x1b5cc0 data parent : 51
Alamat simpul: 0x1b5da0

```

```
Pemasukan data: 51
Alamat parent: 0x1b5d00 data parent : 66
Alamat simpul: 0x1b5cc0

Pemasukan data: 55
Alamat parent: 0x1b5cc0 data parent : 51
Alamat simpul: 0x1b5da0

Pemasukan data: 44
Alamat parent: 0x1b5c30 data parent : 48
Alamat simpul: 0x1b6020

Pemasukan data: 33
Alamat parent: 0x1b5e20 data parent : 24
Alamat simpul: 0x1b5f60

Process returned 0 (0x0)   execution time : 0.697 s
Press any key to continue.
```

- 2 Buatlah sebuah program dengan array yang **sudah terinisialisasi** pada saat deklarasi dimana array adalah sebagai berikut :
- ```
int a[20] = {12, 35, 41, 23, 22, 10, 99, 21, 50, 40, 39, 48, 24, 11, 66, 51, 55, 44, 33, 22};
```
- Masukkan data array tersebut ke dalam binary tree secara left to Right order, kemudian tampilkan seluruh isi binary tree secara **postOrder** Traversal  
(jika error tampilkan saja screenshot errornya dan tetap copas teks kode program nya!!!)

**25%****Jawab:**

Copas jawaban anda di bawah ini

```
// Jawaban
```

```
#include <iostream>
#include <vector>
using namespace std;
```

```
struct node{
 int info;
 node* left, *right;
};
```

```
class BinaryTree{
private:
 node *root, *temp, *before;
public:
 BinaryTree(){
 root = nullptr;
 temp = nullptr;
 }
```

```
void inserts(int A){
 temp = root;
 if (root == nullptr){
 root = new node;
 root->info = A;
 root->right = nullptr;
 root->left = nullptr;
 }else{ //left to right
 while (true){
 if (A > temp->info){
 if (temp->right != nullptr){
 temp = temp->right;
 continue;
 }
 temp->right = new node;
 temp = temp->right;
 temp->right = nullptr;
 temp->left = nullptr;
 temp->info = A;
 break;
 }else if (A < temp->info){
 if (temp->left != nullptr){
 temp = temp->left;
 continue;
 }
 temp->left = new node;
 temp->left->info = A;
 temp->left->right = nullptr;
 temp->left->left = nullptr;
 }
 }
 }
}
```

```

 temp = temp->left;
 temp->right = nullptr;
 temp->left = nullptr;
 temp->info = A;
 break;
 }
 if (A == temp->info) break;
}
}

void postOrderTraversal_mechanics(vector<int>& data,
node* ptr){
 if (ptr->left != nullptr){
 postOrderTraversal_mechanics(data,ptr->left);
 }
 if (ptr->right != nullptr){
 postOrderTraversal_mechanics(data,ptr->right);
 }
 data.push_back(ptr->info);
}

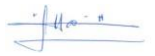
void postOrderTraversal(){
 vector<int>data;
 temp = root;
 postOrderTraversal_mechanics(data,temp);
 for (int i = 0; i < data.size(); i++){
 cout << data[i] << " ";
 }
 cout << '\n';
}

};

int main(){
int a[20] = {12, 35, 41, 23, 22, 10, 99, 21, 50, 40, 39, 48, 24,
11, 66, 51, 55, 44, 33, 22};
 BinaryTree Subaru;
 for (int i = 0; i < 20 ; i++){
 Subaru.inserts(a[i]);
 }
 Subaru.postOrderTraversal();

return 0;}

```

|                                                |                                                                                                        |
|------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>KMK - Mata Kuliah</b> : TIB11-STRUKTUR DATA | Diperiksa oleh:<br> |
| <b>Kelas</b> : 2PTI1,2PTI2, 2PTI51             |                                                                                                        |
| <b>Hari /Tanggal</b> : Selasa/8 Juni 2021      |                                                                                                        |

```
//Akhir kode program
```

Copas Screenshoot hasil running program anda di bawah ini

```
11 10 21 22 33 24 23 39 40 44 48 55 51 66 50 99 41 35 12
Process returned 0 (0x0) execution time : 0.475 s
Press any key to continue.
```

3. Buatlah sebuah program dengan array yang **sudah terinisialisasi** pada saat deklarasi dimana array adalah sebagai berikut :
- ```
int a[20] = {12, 35, 41, 23, 22, 10, 99, 21, 50, 40, 39, 48, 24, 11, 66, 51, 55, 44, 33, 22};
```
- susunlah pada sebuah binary tree secara left-to-right order, kemudian buatlah proses **postOrder** binary tree traversalnya, hasil visit di push() ke sebuah stack setelah semua data dari binary tree tersusun pada stack, lakukan pop() terhadap semua elemen dari stack dan cetaklah di layar semua element yang di pop() tersebut.
(jika error tampilkan saja screenshoot errornya dan tetap copas teks kode program nya!!!)

Jawab:

Copas jawaban anda di bawah ini

```
// Jawaban
#include <iostream>
#include <vector>
using namespace std;

struct node{
    int info;
    node* left, *right;
};

class stacks{
private:
    int info[100], index;
public:
    stacks(){
        index = 0;
    }

    bool isEmpty(){return (index == 0) ;}

    void push(int A){
        info[index] = A;
        index++;
    }
    void pop(){
        index--;
    }
    int topEL(){
```

25
%

```

        return info[index - 1];
    }
};

class BinaryTree{
private:
    node *root, *temp, *before;
public:
    BinaryTree(){
        root = nullptr;
        temp = nullptr;
    }

    void inserts(int A){
        temp = root;
        if (root == nullptr){
            root = new node;
            root->info = A;
            root->right = nullptr;
            root->left = nullptr;
        }else{ //left to right
            while (true){
                if (A > temp->info){
                    if (temp->right != nullptr){
                        temp = temp->right;
                        continue;
                    }
                    temp->right = new node;
                    temp = temp->right;
                    temp->right = nullptr;
                    temp->left = nullptr;
                    temp->info = A;
                    break;
                }else if (A < temp->info){
                    if (temp->left != nullptr){
                        temp = temp->left;
                        continue;
                    }
                    temp->left = new node;
                    temp = temp->left;
                    temp->right = nullptr;
                    temp->left = nullptr;
                    temp->info = A;
                    break;
                }
                if (A == temp->info) break;
            }
        }
    }

    void postOrderTraversal_mechanics(vector<int>& data,
node* ptr){
        if (ptr->left != nullptr){
            postOrderTraversal_mechanics(data, ptr->left);
        }
        if (ptr->right != nullptr){

```



```

        postOrderTraversal_mechanics(data,ptr->right);
    }
    data.push_back( ptr->info );
}

void postOrderTraversal(){
    vector<int>data;
    temp = root;
    postOrderTraversal_mechanics(data,temp);
    stacks stako;
    for (int i = 0; i < data.size(); i++){
        cout << data[i] << " ";
        stako.push(data[i]);
    }
    cout << "\nHasil pop() dan topEL() pada stacks: \n";
    while(!stako.isEmpty()){
        cout<< stako.topEL() << '\n';
        stako.pop();
    }
    cout << '\n';
}

};

int main(){
int a[20] = {12, 35, 41, 23, 22, 10, 99, 21, 50, 40, 39, 48, 24,
11, 66, 51, 55, 44, 33, 22};
    BinaryTree Subaru;
    for (int i = 0; i < 20 ; i++){
        Subaru.inserts(a[i]);
    }
    Subaru.postOrderTraversal();

return 0;}

//Akhir kode program

```

Copas Screenshoot hasil running program anda di bawah ini

```
11 10 21 22 33 24 23 39 40 44 48 55 51 66 50 99 41 35 12
Hasil pop() dan topEL() pada stacks:
12
35
41
99
50
66
51
55
48
44
40
39
23
24
33
22
21
10
11
```

- 4 Buatlah sebuah program dengan array yang **sudah terinisialisasi** pada saat deklarasi dimana array adalah sebagai berikut :
- ```
int a[20] = {12, 35, 41, 23, 22, 10, 99, 21, 50, 40, 39, 48, 24, 11, 66, 51, 55, 44, 33, 22};
```
- push() lah semua data dari array a mulai dari array ke 0 sampai array terakhir. Pada saat akan melakukan push, harus diperiksa dahulu data pada topStack. Jika didapati data pada topStack lebih besar dari data yang akan di push(), maka pop() lah data tersebut dari topStack dan susun pada binary tree secara left-to-right order, kemudian push() lah data dari array tersebut. Jika data yang akan di push() lebih kecil dari topStack, langsung di push() saja ke stack.
- setelah semua data dari array a berhasil diproses, pop() lah satu persatu data yang tersisa pada stack, dan susun data yang di pop() tersebut ke binary tree secara left-to-right order.
- lakukan tree traversal dari binary tree tersebut dan tampilkan di layar secara **postOrder** (jika error tampilkan saja screenshoot errornya dan tetap copas teks kode program nya!!!)

25

**Jawab:**

Copas jawaban anda di bawah ini

```
// Jawaban
#include <iostream>
#include <vector>
using namespace std;

struct node{
 int info;
 node* left, *right;
};

class stacks{
private:
 int info[100], index;
public:
 stacks(){
 index = 0;
 }
}
```

```

bool isEmpty(){return (index == 0) ;}

void push(int A){
 info[index] = A;
 index++;
}
void pop(){
 index--;
}
int topEL(){
 return info[index - 1];
}
};

class BinaryTree{
private:
 node *root, *temp, *before;
public:
 BinaryTree(){
 root = nullptr;
 temp = nullptr;
 }

 void inserts(int A){
 temp = root;
 if (root == nullptr){
 root = new node;
 root->info = A;
 root->right = nullptr;
 root->left = nullptr;
 }else{ //left to right
 while (true){
 if (A > temp->info){
 if (temp->right != nullptr){
 temp = temp->right;
 continue;
 }
 temp->right = new node;
 temp = temp->right;
 temp->right = nullptr;
 temp->left = nullptr;
 temp->info = A;
 break;
 }else if (A < temp->info){
 if (temp->left != nullptr){
 temp = temp->left;
 continue;
 }
 temp->left = new node;
 temp = temp->left;
 temp->right = nullptr;
 temp->left = nullptr;
 temp->info = A;
 break;
 }
 }
 }
 }
}

```

```

 if (A == temp->info) break;
 //artinya data yang dimasukan sdh terdapat di
tree
 }
 }
}

void postOrderTraversal_mechanics(vector<int>& data,
node* ptr){
 if (ptr->left != nullptr){
 postOrderTraversal_mechanics(data,ptr->left);
 }
 if (ptr->right != nullptr){
 postOrderTraversal_mechanics(data,ptr->right);
 }
 data.push_back(ptr->info);
}

void postOrderTraversal(){
 vector<int>data;
 temp = root;
 postOrderTraversal_mechanics(data,temp);
 stacks stako;
 for (int i = 0; i < data.size(); i++){
 cout << data[i] << " ";
 }
 cout << '\n';
}

};

int main(){
int a[20] = {12, 35, 41, 23, 22, 10, 99, 21, 50, 40, 39, 48, 24,
11, 66, 51, 55, 44, 33, 22};
BinaryTree Subaru;
stacks Reinhard;
Reinhard.push(a[0]);
for (int i = 1; i < 20 ; i++){

 while(Reinhard.topEL() > a[i]){
 Subaru.inserts(Reinhard.topEL());
 Reinhard.pop();
 }

 Reinhard.push(a[i]);
}
while(!Reinhard.isEmpty()){
 Subaru.inserts(Reinhard.topEL());
 Reinhard.pop();
}
Subaru.postOrderTraversal();

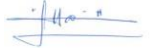
return 0;}

```

```
//Akhir kode program
```

Copas Screenshoot hasil running program anda di bawah ini

```
10 11 21 12 22 33 24 23 39 40 35 44 48 51 55 66 50 99 41
Process returned 0 (0x0) execution time : 0.623 s
Press any key to continue.
```

|                          |                       |                                                                                                        |
|--------------------------|-----------------------|--------------------------------------------------------------------------------------------------------|
| <b>KMK - Mata Kuliah</b> | : TIB11-STRUKTUR DATA | Diperiksa oleh:<br> |
| <b>Kelas</b>             | : 2PTI1,2PTI2, 2PTI51 |                                                                                                        |
| <b>Hari /Tanggal</b>     | : Selasa/8 Juni 2021  |                                                                                                        |