

**Ujian Akhir Semester
Praktikum Struktur Data dan
Pemograman**



Oleh :

Nama : Muhammad Yusuf rismanda gaja

NPM : 1910631170109

Smstr/Kelas : 3G

Prodi : Teknik Informatika

**Universitas Singaperbangsa Karawang
Fakultas Ilmu Komputer Teknik
Informatika
2020**

A. Dana donasi

1. Kodingan

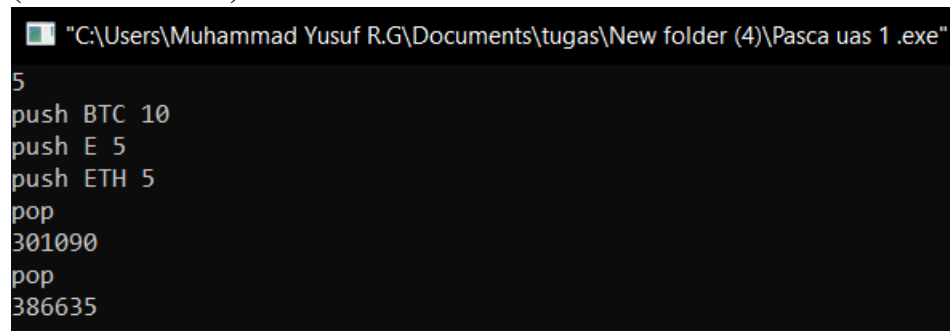
```
1  #include <iostream>
2  #include <stdlib.h>
3  #include <string>
4
5  #define Nil NULL
6
7  using namespace std;
8
9  struct Node {
10     int info;
11     struct Node* next;
12 };
13
14 typedef Node* Nod;
15
16 void Push(Nod* List, int input);
17 int Pop(Nod* List);
18 int konversieuro(int value);
19 int konversidollar(int value);
20 int konversibitcoin(int value);
21 int konversiethereum(int value);
22
23 void Push(Nod* List, int input) {
24     Nod tmp = *List;
25     *List = (Nod)malloc(sizeof(**List));
26     if (*List != Nil) {
27         (*List)->info = input;
28         (*List)->next = tmp;
29     }
30 }
31
32 int Pop(Nod* List) {
33     int result = -1;
34     Nod tmp = *List;
35     if (*List == Nil) {
36         return result;
37     }
38     else
39     {
40         for (Nod current = tmp; current != Nil; current = current->next)
41         {
42             if (current->next == Nil) {
43                 result = current->info;
44                 tmp->next = Nil;
45                 return result;
46             }
47             tmp = current;
48         }
49     }
50 }
51
52 int konversieuro(int value) {
53     return value * (17000 + 109);
54 }
55
56 int konversidollar(int value) {
57     return value * (14000 + 109);
58 }
59
60 int konversibitcoin(int value) {
61     return value * (30000 + 109);
62 }
63
64 int konversiethereum(int value) {
65     return value * (25000 + 109);
66 }
```

```

67
68 int main()
69 {
70     Node* Uang = Nil;
71     int P = 0, input, jumlah = 0;
72     string Perintah, Matauang;
73     cout<<" ";
74     cin>>P;
75
76     for(int i = 0; i < P; i++)
77     {
78         cin>>Perintah;
79         if (Perintah == "push" ){
80             cin>>Matauang;
81             cin>>input;
82             if (Matauang == "E")
83                 Push(&Uang, konversieuro(input));
84             else if (Matauang == "D")
85                 Push(&Uang, konversidollar(input));
86             else if (Matauang == "BTC")
87                 Push(&Uang, konversibitcoin(input));
88             else if (Matauang == "ETH")
89                 Push(&Uang, konversiethereum(input));
90         }
91         else if (Perintah == "pop"){
92             jumlah += Pop(&Uang);
93             cout<<jumlah<<endl;
94         }
95     }
96 }

```

2. Keluaran Program (1910631170109)



```

"C:\Users\Muhammad Yusuf R.G\Documents\tugas\New folder (4)\Pasca uas 1 .exe"
5
push BTC 10
push E 5
push ETH 5
pop
301090
pop
386635

```

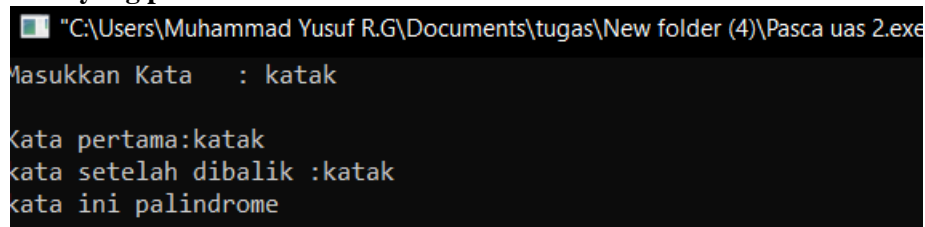
B. Palindrom

1. Kodingan

```
1 #include <iostream>
2 #include <algorithm>
3
4 using namespace std;
5
6 int main() {
7     string abc;
8     string kata;
9
10    // meminta user menginputkan kata yang akan dicek
11    // contoh: BaKaD
12
13    cout << "Masukkan Kata\t: ";
14    getline(cin, kata);
15
16    // buat variable baru dengan value sama dengan apa yang diinputkan user
17    // lalu ubah menjadi lowercase
18    // hasil setelah lowercase: baksd
19
20    string originalKata = kata;
21    transform(originalKata.begin(), originalKata.end(), originalKata.begin(), ::tolower);
22
23    // buat variable baru lagi dengan value yang sudah dijadikan lowercase
24    // kemudian reverse / balik nilainya
25    // hasil setelah dibalik: oskab
26
27    string reversedWord = originalKata;
28    reverse(reversedWord.begin(), reversedWord.end());
29    abc=reversedWord;
30
31    // bandingkan originalKata dengan reversedWord
32    // jika sama maka palindrome, selain itu bukan
33
34    if(originalKata == reversedWord) {
35        cout << "\nKata pertama:" << kata << "\nkata setelah dibalik : " << abc << "\nkata ini palindrome"
36        <<endl;
37    } else {
38        cout << "\nKata pertama:" << kata << "\nkata setelah dibalik : " << abc << "\nkata ini bukan
39        palindrome" <<endl;
40    }
41    return 0;
42 }
```

2. Keluaran

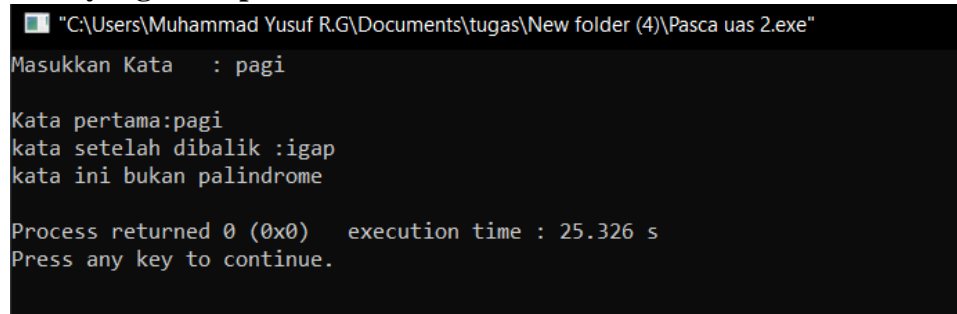
Kata yang palindrome



```
"C:\Users\Muhammad Yusuf R.G\Documents\tugas\New folder (4)\Pasca uas 2.exe"
Masukkan Kata    : katak

Kata pertama:katak
kata setelah dibalik :katak
kata ini palindrome
```

Kata yang tidak palindrome



```
"C:\Users\Muhammad Yusuf R.G\Documents\tugas\New folder (4)\Pasca uas 2.exe"
Masukkan Kata    : pagi

Kata pertama:pagi
kata setelah dibalik :igap
kata ini bukan palindrome

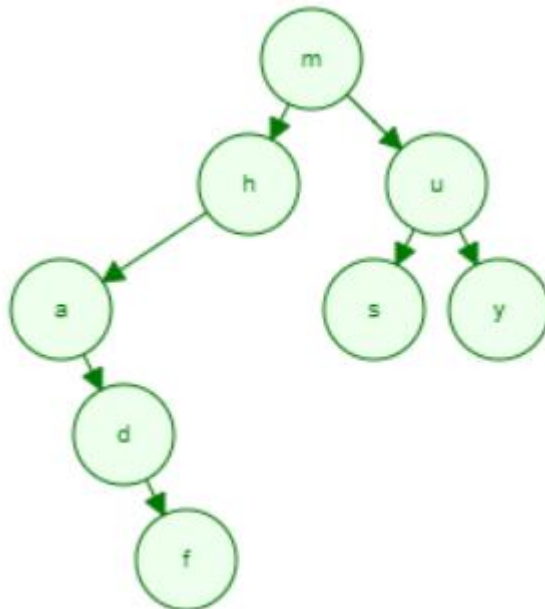
Process returned 0 (0x0) execution time : 25.326 s
Press any key to continue.
```

C. Pohon Karakter

1. Algoritma Konsep Tree

Buat pohon untuk menyusun penempatan nilai yang akan ditempatkan di posisi Right atau Left. Dan hitung secara manual untuk menentukan InOrder, PreOrder, PostOrder.

Muhammad Yusuf = muhad ysf



Preorder = MUHADYSF

Inorder = HUAMDSYF

Postorder = HAUSFYDM

2. Code Program Tree

- Header untuk hak akses Synthax

```
1 #include<iostream>
2 #include<malloc.h>
3 #define Nil NULL
```

- Fungsi yang digunakan
NodBaru() fungsi ini dipakai di AddNode() berfungsi untuk mengalokasikan memori

```

NOD* NodBaru(char item) {
    NOD* n;
    n = (NOD*)malloc(sizeof(NOD));

    if (n != Nil)
    {
        n->data = item;
        n->left = Nil;
        n->right = Nil;
    }

    return n;
}

```

- TambahkanNod() untuk menambahkan node

```

void TambahkanNod(NOD** p, char item)
{
    NOD* n;
    n = NodBaru(item);
    *p = n;
}

```

- Fungsi InOrder(), PreOrder(), PostOrder() untuk mengkomputasi urutan tree berdasarkan tipe yang terdapat pada order tersebut.

```
void preOrder(POKOK* T) {  
    if (!PokokKosong(T))  
    {  
        cout << " " << T->data;  
        preOrder(T->left);  
        preOrder(T->right);  
    }  
}  
  
void inOrder(POKOK* T) {  
    if (!PokokKosong(T))  
    {  
        inOrder(T->left);  
        cout << " " << T->data;  
        inOrder(T->right);  
    }  
}  
  
void postOrder(POKOK* T) {  
    if (!PokokKosong(T))  
    {  
        postOrder(T->left);  
        postOrder(T->right);  
        cout << " " << T->data;  
    }  
}
```

```

1  #include<iostream>
2  #include<malloc.h>
3  #define Nil NULL
4
5  using namespace std;
6
7  struct nod {
8      struct nod* left;
9      char data;
10     struct nod* right;
11 };
12
13 typedef struct nod NOD;
14 typedef NOD POKOK;
15
16 NOD* NodBaru(char item);
17 void BinaPokok(POKOK** T);
18 bool PokokKosong(POKOK* T);
19 void TambahNod(NOD** p, char item);
20 void preOrder(POKOK* T);
21 void inOrder(POKOK* T);
22 void postOrder(POKOK* T);
23
24 int main() { cout<<"Nama : Muhammad Yusuf Rismanda Gaja"<<endl<<endl;
25     POKOK* kelapa;
26     char buah;
27
28     BinaPokok(&kelapa);
29
30     TambahNod(&kelapa, buah = 'M');
31     TambahNod(&kelapa->left, buah = 'U');
32     TambahNod(&kelapa->left->left, buah = 'H');
33     TambahNod(&kelapa->left->right, buah = 'A');
34     TambahNod(&kelapa->right, buah = 'D');
35     TambahNod(&kelapa->right->right, buah = 'Y');
36     TambahNod(&kelapa->right->right->left, buah = 'S');
37     TambahNod(&kelapa->right->right->right, buah = 'F');
38
39     cout << "Tampilan secara PreOrder : ";
40     preOrder(kelapa);
41
42     cout << endl;
43     cout << "Tampilan secara InOrder : ";
44     inOrder(kelapa);
45
46     cout << endl;
47     cout << "Tampilan secara PostOrder : ";
48     postOrder(kelapa);
49
50
51     cout << endl;
52     cout << endl;
53
54 }
55
56 NOD* NodBaru(char item) {
57     NOD* n;
58     n = (NOD*)malloc(sizeof(NOD));
59
60     if (n != Nil) {
61         n->data = item;
62         n->left = Nil;
63         n->right = Nil;
64     }
65     return n;
66 }

```



```

67
68 void BinaPokok(POKOK** T) {
69     *T = Nil;
70 }
71
72 bool PokokKosong(POKOK* T) {
73     return ((bool) (T == Nil));
74 }
75
76 void TambahNod(NOD** p, char item) {
77     NOD* n;
78     n = NodBaru(item);
79
80     *p = n;
81 }
82
83 void preOrder(POKOK* T) {
84     if (!PokokKosong(T)) {
85         cout << " " << T->data;
86         preOrder(T->left);
87         preOrder(T->right);
88     }
89 }
90
91 void inOrder(POKOK* T) {
92     if (!PokokKosong(T)) {
93         inOrder(T->left);
94         cout << " " << T->data;
95         inOrder(T->right);
96     }
97 }
98
99 void postOrder(POKOK* T) {
100     if (!PokokKosong(T)) {
101         postOrder(T->left);
102         postOrder(T->right);
103         cout << " " << T->data;
104     }
105 }

```

3. Tampilan Hasil dari Program.

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\Users\Muhammad Yusuf R.G\Documents\tugas\sdp\tree.exe". The window has standard Windows window controls (minimize, maximize, close) on the right. The command prompt shows the following text:
Tampilan secara PreOrder : M U H A D Y S F
Tampilan secara InOrder : H U A M D S Y F
Tampilan secara PostOrder : H A U S F Y D M

Process returned 0 (0x0) execution time : 0.164 s
Press any key to continue.
The rest of the window is empty, waiting for a key press.