

Tugas Praktikum Algoritma dan Struktur Data



Nama : Agil Deriansyah Hasan
Nim : 4522210125

Dosen Pengajar :

Dra.SRI REZEKI CANDRA NURSARI,M.Kom
Prak. Algoritma dan Struktur Data - I

**S1-Teknik Informatika
Fakultas Teknik
Universitas Pancasila 2023/2024**

```

1  #include <iostream>
2  using namespace std;
3
4  struct node {
5      node* left;
6      char data;
7      node* right;
8  };
9
10 typedef node simpul;
11 simpul* root, *P, *current;
12 simpul* Q[129];
13
14 void inisialisasi() {
15     root = NULL;
16     P = NULL;
17 }
18
19 void buatSimpul(char X) {
20     P = new simpul;
21     if (P != NULL) {
22         P->data = X;
23         P->left = NULL;
24         P->right = NULL;
25     } else {
26         cout << "Memory Heap Full" << endl;
27         exit(1);
28     }
29 }
30
31 void buatSimpulakar() {
32     if (root == NULL) {
33         if (P != NULL) {
34             root = P;
35             root->left = NULL;
36             root->right = NULL;
37         } else {
38             cout << "Simpul Belum Dibuat" << endl;
39         }
40     } else {
41         cout << "Pohon Sudah Ada!!!!!!" << endl;
42     }
43 }
44

```

```

44
45 void inserturutnomor() {
46     int i = 1, j = 1, flaq = 0;
47     char X;
48     Q[i] = root;
49     while (flaq == 0 && j < 129) {
50         cout << "Masukkan karakter (0 untuk berhenti): ";
51         cin >> X;
52         if (X != '0') {
53             buatSimpul(X);
54             current = Q[i];
55             if (current->left == NULL) {
56                 current->left = P;
57             } else if (current->right == NULL) {
58                 current->right = P;
59             } else {
60                 cout << "Subtree penuh!" << endl;
61                 continue;
62             }
63             j++;
64             Q[j] = P;
65         } else {
66             flaq = 1;
67             j++;
68             Q[j] = NULL;
69         }
70         i++;
71     }
72 }
73

```

```
pasd11-6.cpp | cnthprak12-1.cpp | cnthprak12-2.cpp |
73
74 void bacaurutnomor() {
75     int i = 1, j = 1, n = 1, counter = 0;
76     while (Q[i] != NULL) {
77         current = Q[i];
78         cout << current->data << " ";
79         counter++;
80         if (counter == n) {
81             cout << endl;
82             counter = 0;
83             n = n + 2;
84         }
85         if (current->left != NULL) {
86             j++;
87             Q[j] = current->left;
88         }
89         if (current->right != NULL) {
90             j++;
91             Q[j] = current->right;
92         }
93         i++;
94     }
95     cout << endl;
96 }
97
98 int main() {
99     inisialisasi();
100     char akar;
101     cout << "Masukkan karakter untuk akar: ";
102     cin >> akar;
103     buatSimpul(akar);
104     buatSimpulakar();
105     inserturutnomor();
106     bacaurutnomor();
107     return 0;
108 }
109
```

```
CA Command Prompt X + v
F:\>g++ pasd11-6.cpp -o 1
F:\>1
Masukkan karakter untuk akar: A
Masukkan karakter (0 untuk berhenti): B
Masukkan karakter (0 untuk berhenti): C
Masukkan karakter (0 untuk berhenti): D
Masukkan karakter (0 untuk berhenti): E
Masukkan karakter (0 untuk berhenti): F
Masukkan karakter (0 untuk berhenti): G
Masukkan karakter (0 untuk berhenti): H
Masukkan karakter (0 untuk berhenti): -
Masukkan karakter (0 untuk berhenti): =
Masukkan karakter (0 untuk berhenti): 0
A
B C D
E F G H -
=
F:\>
```

Pseudocode :

Kamus/Deklarasi Variabel fungsi Inisialisasi

-

Algoritma/Deskripsi fungsi inisialisasi

root = NULL

p = NULL

Kamus/Deklarasi Variabel fungsi buat simpul

X = char

Algoritma/Deskripsi fungsi simpul(X)

P = new simpul

if (P != NULL)

 P->data = X

 P->left = NULL

 P->right = NULL

else

 print ("Memory Heap Full")

 exit(1)

endif

Kamus/Deklarasi Variabel fungsi buatsimpulakar

-

Algoritma/Deskripsi fungsi buatsimpulakar

if (root == NULL)

 if (P != NULL)

 root = P

 root->left = NULL

 root->right = NULL

 else

 print ("Simpul Belum Dibuat")

 endif

 else

 print ("Pohon Sudah Ada!!!!!!")

endif

Kamus/Deklarasi Variabel fungsi inserturutnomor

i,j,flaq = int

X = char

Algoritma/Deskripsi fungsi inserturutnomor

i = 1

j = 1

flaq = 0

Q[i] = root

while (flaq == 0 && j < 129)

 input X

 if (X != '0')

 buatSimpul(X);

 current = Q[j]

 if (current->left == NULL)

 current->left = P

 endif

 else

 if (current->right == NULL)

 current->right = P

 endif

 else

 cout << "Subtree penuh!" << endl

 continue

 endif

 j++

 Q[j] = P

else

 flaq = 1

 j++

 Q[j] = NULL

endif

 i++

endwhile

Kamus/Deklarasi Variabel fungsi bacaurutnomor
i,k,n,counter = int

Algoritma/Deskripsi fungsi bacaurutnomor

```
i = 1, j = 1, n = 1, counter = 0
while (Q[i] != NULL)
    current = Q[i]
    print current->data
    counter++
    if (counter == n)
        cout << endl
        counter = 0
        n = n + 2
    endif
    if (current->left != NULL)
        j++
        Q[j] = current->left
    endif
    if (current->right != NULL)
        j++
        Q[j] = current->right
    endif
    i++
endwhile
```

Kamus/Deklarasi Variabel fungsi utama
akar, data = char
left , right = node*

Algoritma/Deskripsi fungsi utama

```
struct node{ left, data, right}
typedef node simpul
simpul* root, *P, *current
simpul* Q[129]
inisialisasi()
    input akar
    buatSimpul(akar)
    buatSimpulakar()
    inserturutnomor()
    bacaurutnomor()
```

Algoritma :

1. Membuat fungsi inisialisasi
2. root = NULL
3. p = NULL
4. Membuat fungsi buatsimpul(X)
5. p = new simpul
6. Jika (P!=NULL) maka kerjakan baris 7 s.d 9 kalau tidak 9 s.d 11
7. P->data = X
8. P->left = NULL
9. P->right = NULL
10. Mencetak "Memory Heap Full"
11. exit(1)
12. Membuat fungsi buatsimpulakar
13. Jika (root == NULL) maka kerjakan baris 14 s.d 18 kalau tidak baris 19
14. Jika (P != NULL) maka kerjakan baris 15 s.d 17 kalau tidak baris 18
15. root = P
16. root->left = NULL
17. root->right = NULL
18. Mencetak "Simpul Belum Dibuat"
19. Mencetak "Pohon Sudah Ada!!!!!!"
20. Membuat fungsi inserurutnomor
21. i = 1
22. j = 1
23. flaq = 0
24. Q[i] = root
25. Selama (flaq == 0 && j < 129) maka kerjakan baris 26 s.d 41
26. Menginput/Memasukkan Nilai X
27. Jika (X != '0') maka kerjakan baris 28 s.d 38 kalau tidak baris 38 s.d 40
28. Memanggil fungsi buatsimpul(X)
29. cureent = Q[i]
30. Jika (current -> left == NULL) Maka kerjakan baris 31 kalau tidak baris 35

31. current -> left = P
32. Jika (current -> right == NULL) maka kerjakan baris 33 s.d 35
33. current -> right = P
34. Mencetak "Subtree Penuh!")
35. continue
36. j++
37. Q[j] = P
38. flaq=1
39. j++
40. Q[j] = NULL
41. i++
42. Membuat fungsi bacaurutnomor
43. i = 1
44. j = 1
45. n = 1
46. counter= 0
47. Selama (Q[i] != NULL) maka kerjakan baris 48 s.d 60
48. current = Q[i]
49. Mencetak/Menampilkan Nilai current -> data
50. counter++
51. Jika (counter == n) maka kerjakan baris 52 s.d 53
52. counter = 0
53. n = n+2
54. Jika (current -> left != NULL) maka kerjakan baris 55 s.d 56
55. j++
56. Q[j] = current -> left
57. Jika (current -> right != NULL) maka kerjakan baris 58 s.d 59
58. j++
59. Q[j] = current -> right
60. i++

61. Membuat fungsi utama
62. Deklarasi sturktur (struct node{ left , data, right})
63. Pendefinisian node simpul
64. simpul* root, *P, *current
65. simpul* Q[129]
66. Memanggil fungsi inisialisasi
67. Menginput/Memasukkan Nilai akar
68. Memanggil fungsi buatsimpul(akar)
69. Memanggil fungsi buatsimpulakar
70. Memanggil fungsi inserturutnomor
71. Memanggil fungsi bacaurutnoor
72. Selesai