

Contoh 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  void heap(int canarray[], int n, int i) {
5      int temp;
6      int canbesar = i;
7      int kiri = 2 * i + 1;
8      int kanan = 2 * i + 2;
9
10     if (kiri < n && canarray[kiri] > canarray[canbesar])
11         canbesar = kiri;
12     if (kanan < n && canarray[kanan] > canarray[canbesar])
13         canbesar = kanan;
14
15     if (canbesar != i) {
16         temp = canarray[i];
17         canarray[i] = canarray[canbesar];
18         canarray[canbesar] = temp;
19         heap(canarray, n, canbesar);
20     }
21 }
22
23 void sortheap(int canarray[], int n) {
24     int temp;
25     for (int i = n / 2 - 1; i >= 0; i--)
26         heap(canarray, n, i);
27     for (int i = n - 1; i >= 0; i--) {
28         temp = canarray[0];
29         canarray[0] = canarray[i];
30         canarray[i] = temp;
31         heap(canarray, i, 0);
32     }
33 }
34

```

```

34
35 int main() {
36     int canarray[] = {22, 7, 66, 28, 11, 63, 24, 12, 77, 99};
37     int n = 10;
38     int i;
39
40     cout << "Menampilkan Data Sebelum Diurutkan" << endl;
41     cout << "-----" << endl;
42     for (i = 0; i < n; i++)
43         cout << canarray[i] << " ";
44     cout << endl;
45
46     sortheap(canarray, n);
47
48     cout << endl;
49     cout << "Menampilkan Data Setelah Diurutkan - HEAP SORT" << endl;
50     cout << "-----" << endl;
51     for (i = 0; i < n; i++)
52         cout << canarray[i] << " ";
53     cout << endl;
54
55     return 0;
56 }
57

```

```

Command Prompt

F:\>g++ asd12.cpp -o 1

F:\>1
Menampilkan Data Sebelum Diurutkan
-----
22 7 66 28 11 63 24 12 77 99

Menampilkan Data Setelah Diurutkan - HEAP SORT
-----
7 11 12 22 24 28 63 66 77 99

F:\>

```

Pseudocode :

Kamus/Deklarasi Variabel fungsi heap
canarray[], n, i, temp, canbesar, kiri, kanan = int

Algoritma/Deskripsi fungsi heap(canarray[],n,i)
canbesar = i
kiri = 2*i+1
kanan = 2*i+2
if (kiri < n && canarray[kiri] > canarray[canbesar])
 canbesar = kiri
if (kanan < n && canarray[kanan] > canarray[canbesar])
 canbesar = kanan
if (canbesar != i)
 temp = canarray[i]
 canarray[i] = canarray[canbesar]
 canarray[canbesar] = temp
 heap(canarray, n, canbesar)
endif

Kamus/Deklarasi Variabel fungsi sortheap
canarray[], n, temp, i = int

Algoritma/Deskripsi fungsi sortheap(canarray[], n)
for (int i = n / 2 - 1; i >= 0; i--)
 heap(canarray, n, i)
for (int i = n - 1; i >= 0; i--)
 temp = canarray[0]
 canarray[0] = canarray[i]
 canarray[i] = temp
 heap(canarray, i, 0)
endfor

Kamus/Deklarasi Variabel fungsi utama
canarray[],n,i = int

Algoritma/Deskripsi fungsi utama
canarray[] = {22, 7, 66, 28, 11, 63, 24, 12, 77, 99}
n = 10
for (i = 0; i < n; i++)
 print canarray[i]
sortheap(canarray, n);

for (i = 0; i < n; i++)
 print canarray[i]
return 0

Algoritma :

1. Membuat fungsi heap (canarray[],n,i)
2. canbesar=i
3. kiri = 2*i+1
4. kanan = 2*i+2
5. Jika (kiri < n && canarray[kiri] > canarray[canbesar])
6. canbesar = kiri
7. Jika (kanan < n && canarray[kanan] > canarray[canbesar])
8. canbesar = kanan
9. Jika (canbesar != i) maka kerjakan baris 10 s.d 13
10. temp = canarray[i]
11. canarray[i] = canarray[canbesar]
12. canarray[canbesar] = temp
13. Memanggil fungsi heap(canarray, n, canbesar)
14. Membuat fungsi sortheap(canarray[],n)
15. Selama (i = n / 2-1)
16. Memanggil fungsi heap(canarray,n,i)
17. i-
18. Selama (i = n - 1) maka kerjakan baris 19 s.d 23
19. temp = canarray[0]
20. canarray[0] = canarray[i]
21. canarray[i] = temp
22. heap(canarray, i, 0)
23. i-
24. Membuat fungsi utama
25. canarray[] = {22, 7, 66, 28, 11, 63, 24, 12, 77, 99};
26. n = 10
27. Selama (i = 0)
28. Mencetak/Menampilkan Nilai canarray[i]
29. Memanggil fungsi sortheap(canarray,n)
30. i++
31. Selama (i = 0)
32. Mencetak/Menampilkan Nilai canarray[i]
33. i++
34. Selesai