

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**

contoh prak 6-1

```
cnthprak06-1.cpp | cnthprak06-2.cpp | cnthprak07-1.cpp |
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  void Q_Sort(int[], int, int);
6
7  int main() {
8      int numlist[9] = {65, 2, 44, 26, 19, 22, 5, 3, 12};
9      cout << "Baca Sebelum Diurutkan" << endl;
10     cout << "~~~~~" << endl;
11     for (int d = 0; d < 9; d++) {
12         cout << setw(3) << numlist[d];
13     }
14     cout << endl << endl;
15     Q_Sort(numlist, 0, 9);
16     cout << "Data Setelah Diurutkan" << endl;
17     cout << "~~~~~" << endl;
18     for (int iii = 0; iii < 9; iii++) {
19         cout << setw(3) << numlist[iii];
20     }
21     cin.get();
22     return 0;
23 }
24
25 void Q_Sort(int angka[], int kiri, int kanan) {
26     int pivot, kiri_hold, kanan_hold;
27     kiri_hold = kiri;
28     kanan_hold = kanan;
29     pivot = angka[kiri];
30     while (kiri < kanan) {
31         while (angka[kanan] >= pivot && kiri < kanan)
32             kanan--;
33         if (kiri != kanan) {
34             angka[kiri] = angka[kanan];
35             kiri++;
36         }
37         while (angka[kiri] <= pivot && kiri < kanan)
38             kiri++;
39         if (kiri != kanan) {
40             angka[kanan] = angka[kiri];
41             kanan--;
42         }
43     }
44     angka[kiri] = pivot;
45     pivot = kiri;
46     kiri = kiri_hold;
47     kanan = kanan_hold;
48     if (kiri < pivot)
49         Q_Sort(angka, kiri, pivot - 1);
50     if (kanan > pivot)
51         Q_Sort(angka, pivot + 1, kanan);
52 }
53
```

```
Command Prompt - 1
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\agild>F:

F:\>g++ cnthprak06-1.cpp -o 1

F:\>1
Baca Sebelum Diurutkan
~~~~~
65  2 44 26 19 22  5  3 12

Data Setelah Diurutkan
~~~~~
2  3  5 12 19 22 26 44 65
```

Pseudocode

Kamus/Deklarasi Variabel
d, iii, numlist[9] : int

Algoritma/Deskripsi
numlist[9] = {65, 2, 44, 26, 19, 22, 5, 3, 12}
for(d = 0; d < 9; d++)
 print (numlist[d])
endfor
Q_Sort(numlist, 0, 9)
for (iii = 0; iii < 9; iii++)
 print (numlist[iii])
endfor
return 0

Kamus/Deklarasi Variabel fungsi Q_Sort
angka[], kiri, kanan : int
pivot, kiri_hold, kanan_hold : int

Algoritma/Deskripsi Fungsi Q_Sort(angka[], kiri, kanan)
 kiri_hold = kiri
 kanan_hold = kanan
 pivot = angka[kiri]
 while (kiri < kanan)
 while (angka[kanan] >= pivot && kiri < kanan)
 kanan--
 if (kiri != kanan)
 angka[kiri] = angka[kanan]
 kiri++
 endif
 while (angka[kiri] <= pivot && kiri < kanan)
 kiri++
 if (kiri != kanan)
 angka[kanan] = angka[kiri]
 kanan--
 endif
 endwhile
 angka[kiri] = pivot
 pivot = kiri
 kiri = kiri_hold
 kanan = kanan_hold
 if (kiri < pivot)
 Q_Sort(angka, kiri, pivot - 1)
 if (kanan > pivot)
 Q_Sort(angka, pivot + 1, kanan)
 endif
endif

Algoritma

1. numlist[9] = {65, 2, 44, 26, 19, 22, 5, 3, 12}
2. Selama (d=0) Maka kerjakan baris 3 s.d 4
3. Mencetak/Menampilkan Nilai numlist[d]
4. d++
5. Memanggil fungsi Q_Sort(numlist, 0,9)
6. Selama (iii=0) maka kerjakan baris 7 s.d 8
7. Mencetak/Menampilkan Nilai numlist[iii]
8. iii++
9. Membuat fungsi Q_Sort(angka[], kiri, kanan)
10. kiri_hold = kiri
11. kanan_hold = kanan
12. pivot = angka[kiri]
13. Selama (kiri<kanan) maka kerjakan baris 14 s.d 23
14. selama (angka[kanan] >= pivot && kiri < kanan)
15. kanan--
16. Jika (kiri<=kanan) maka kerjakan baris 17 s.d 18
17. angka[kiri] = angka[kanan]
18. kiri++
19. selama (angka[kiri] <= pivot && kiri < kanan))
20. kiri++
21. Jika (kiri<=kanan) maka kerjakan baris 22 s.d 23
22. angka[kanan] = angka[kiri]
23. kanan--
24. angka[kiri] = pivot
25. pivot = kiri
26. kiri = kiri_hold
27. kanan = kanan_hold
28. Jika (kiri<pivot)
29. Q_Sort(angka, kiri, pivot - 1)
30. Jika (kanan>pivot)
31. Q_Sort(angka, pivot + 1, kanan)

contoh prak 6-2

```
cnthprak06-1.cpp  cnthprak06-2.cpp  cnthprak07-1.cpp
1  #include <iostream>
2  using namespace std;
3
4  void merge(int arr[], int kiri, int mid, int kanan) {
5      int n1 = mid - kiri + 1;
6      int n2 = kanan - mid;
7
8      int L[n1], R[n2];
9
10     for (int i = 0; i < n1; i++) {
11         L[i] = arr[kiri + i];
12     }
13     for (int j = 0; j < n2; j++) {
14         R[j] = arr[mid + 1 + j];
15     }
16
17     int i = 0;
18     int j = 0;
19     int k = kiri;
20
21     while (i < n1 && j < n2) {
22         if (L[i] <= R[j]) {
23             arr[k] = L[i];
24             i++;
25         } else {
26             arr[k] = R[j];
27             j++;
28         }
29         k++;
30     }
31     while (i < n1) {
32         arr[k] = L[i];
33         i++;
34         k++;
35     }
36     while (j < n2) {
37         arr[k] = R[j];
38         j++;
39         k++;
40     }
41 }
```

```
cnthprak06-1.cpp  cnthprak06-2.cpp  cnthprak07-1.cpp
36     while (j < n2) {
37         arr[k] = R[j];
38         j++;
39         k++;
40     }
41 }
42
43 void mergesort(int arr[], int kiri, int kanan) {
44     if (kiri < kanan) {
45         int mid = kiri + (kanan - kiri) / 2;
46         mergesort(arr, kiri, mid);
47         mergesort(arr, mid + 1, kanan);
48         merge(arr, kiri, mid, kanan);
49     }
50 }
51
52 void cetakdata(int arr[], int size) {
53     for (int i = 0; i < size; i++) {
54         cout << arr[i] << " ";
55     }
56     cout << endl;
57 }
58
59 int main() {
60     int arr[] = {65, 2, 44, 26, 19, 22, 5, 3, 12};
61     int arrsize = sizeof(arr) / sizeof(arr[0]);
62
63     cout << "Data Sebelum diurutkan:" << endl;
64     cout << "~~~~~" << endl;
65     cetakdata(arr, arrsize);
66
67     mergesort(arr, 0, arrsize - 1);
68     cout << endl << endl;
69     cout << "Data Setelah diurutkan:" << endl;
70     cetakdata(arr, arrsize);
71
72     return 0;
73 }
```

```
Command Prompt
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\agild>f:

F:\>G++ cnthprak06-2.cpp -o 1

F:\>1
Data Sebelum diurutkan:
~~~~~
65 2 44 26 19 22 5 3 12

Data Setelah diurutkan:
2 3 5 12 19 22 26 44 65

F:\>|
```

Pseudocode

Kamus/Deklarasi Variabel Fungsi merge
i, j, k, n1, n2, L[n1], R[n2] = int

Algoritma/Deskripsi Fungsi merge (arr[], kiri, mid, kanan)
n1 = mid - kiri + 1
n2 = kanan - mid
for (i = 0; i < n1; i++)
 L[i] = arr[kiri + i]
endfor
for (int j = 0; j < n2; j++)
 R[j] = arr[mid + 1 + j]
endfor
i = 0
j = 0
k = kiri
while (i < n1 && j < n2)
 if (L[i] <= R[j])
 arr[k] = L[i]
 i++
 else
 arr[k] = R[j]
 j++
 endif
 k++
endwhile
while (i < n1)
 arr[k] = L[i]
 i++
 k++
endwhile
while (j < n2)
 arr[k] = R[j]
 j++
 k++
endwhile

Kamus/Deklarasi Variabel fungsi mergesort
arr[], kiri, kanan, mid = int

Algoritma/Deskripsi Fungsi mergesort (arr[], kiri, mid, kanan)
if (kiri < kanan)
 mid = kiri + (kanan - kiri) / 2
 mergesort(arr, kiri, mid)
 mergesort(arr, mid + 1, kanan)
 merge(arr, kiri, mid, kanan)
endif

Kamus/Deklarasi cetakdata
arr[], size, i = int

Algoritma/Deskripsi fungsi cetakdata (arr[], size)
for (i = 0; i < size; i++)
 print (arr[i])
endfor

Kamus/Deklarasi Variabel
arr[], arrsize = int

Algoritma/Deskripsi
arr[] = {65, 2, 44, 26, 19, 22, 5, 3, 12}
arrsize = sizeof(arr) / sizeof(arr[0])
cetakdata(arr, arrsize)
mergesort(arr, 0, arrsize - 1)
cetakdata(arr, arrsize)
return 0

Algoritma

1. Membuat fungsi merge (arr[], kiri, mid, kanan)
2. n1 = mid - kiri + 1
3. n2 = kanan - mid
4. Selama (i=0) maka kerjakan baris 5 s.d 6
5. L[i] = arr[kiri + i]
6. i++
7. Selama (j=0) maka kerjakan baris 8 s.d 9
8. R[j] = arr[mid + 1 + j]
9. j++
10. i=0
11. j=0
12. k=kiri
13. Selama (i < n1 && j < n2) maka kerjakan baris 14 s.d 20
14. Jika(L[i]<=R[j]) maka kerjakan baris 15 s.d 16 kalau tidak kerjakan 17 s.d 19
15. arr[k] = L[i]
16. i++
17. else
18. arr[k]=R[j]
19. j++
20. k++
21. Selama (i<n1) maka kerjakan baris 22 s.d 24
22. arr[k]=L[i]
23. i++
24. k++
25. Selama (j<n2) maka kerjakan baris 26 s.d 28
26. arr[k]=R[j]
27. j++
28. k++
29. Membuat fungsi mergesort (arr[],kiri,kanan)
30. Jika (kiri<kanan) maka kerjakan baris 31 s.d 34
31. mid = kiri + (kanan - kiri)/2
32. Memanggil fungsi mergesort(arr, kiri, mid)
33. Memanggil fungsi mergesort(arr, mid+1, kanan)
34. Memanggil fungsi mergesort(arr, kiri, mid, kanan)

35. Membuat fungsi cetakdata(arr[], size)
36. Selama (i=0) maka kerjakan baris 37 s.d 36
37. Mencetak/Menampilkan Nilai arr[i]
38. i++
39. Membuat fungsi utama
40. arr[]={65, 2, 44, 26, 19, 22, 5, 3, 12}
41. arrsize = sizeof(arr) / sizeof(arr[0])
42. Memanggil fungsi cetakdata(arr,arrsize)
43. Memanggil fungsi mergesort(arr,0,arrsize-1)
44. Memanggil fungsi cetakdata(arr,arrsize)
45. Selesai