Tugas Praktikum Algoritma dan Struktur Data

Nama : Agil Deriansyah Hasan
Nim : 4522210125

Dosen:

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

S1-Teknik Informatika
Fakultas Teknik
Universitas Pancasila 2023/2024

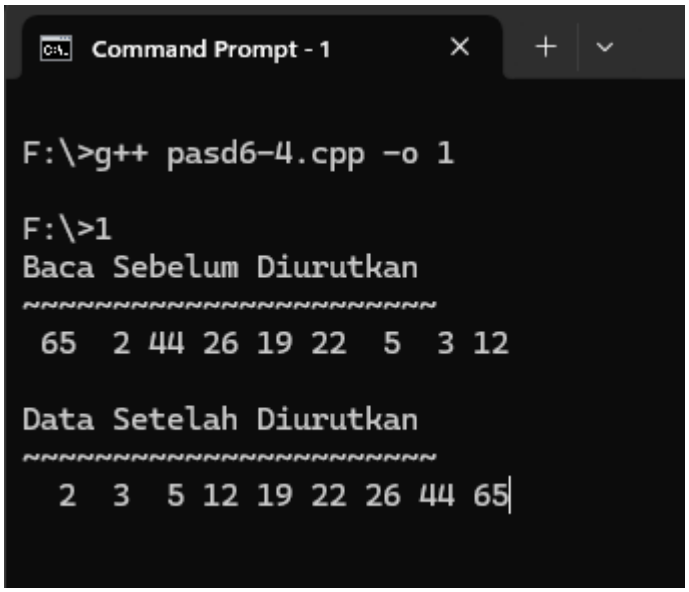
pasd6-4

cnthprak06-1.cpp | cnthprak06-2.cpp | cnthprak07-1.cpp | pasd6-4.cpp | pasd6-25.cpp

```cpp
#include <iostream>
#include <iomanip>
using namespace std;

struct Data {
    int value;
};

void Q_Sort(Data[], int, int);

int main() {
    Data dataList[9] = {65, 2, 44, 26, 19, 22, 5, 3, 12};

    cout << "Baca Sebelum Diurutkan" << endl;
    cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;
    for (int d = 0; d < 9; d++) {
        cout << setw(3) << dataList[d].value;
    }
    cout << endl << endl;

    Q_Sort(dataList, 0, 8);

    cout << "Data Setelah Diurutkan" << endl;
    cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;
    for (int iii = 0; iii < 9; iii++) {
        cout << setw(3) << dataList[iii].value;
    }

    cin.get();
    return 0;
}
```

cnthprak06-1.cpp | cnthprak06-2.cpp | cnthprak07-1.cpp | pasd6-4.cpp | pasd6-25.cpp

```cpp
void Q_Sort(Data data[], int left, int right) {
    int pivot, left_hold, right_hold;
    left_hold = left;
    right_hold = right;
    pivot = data[left].value;
    while (left < right) {
        while (data[right].value >= pivot && left < right)
            right--;
        if (left != right) {
            data[left].value = data[right].value;
            left++;
        }
        while (data[left].value <= pivot && left < right)
            left++;
        if (left != right) {
            data[right].value = data[left].value;
            right--;
        }
    }
    data[left].value = pivot;
    pivot = left;
    left = left_hold;
    right = right_hold;
    if (left < pivot)
        Q_Sort(data, left, pivot - 1);
    if (right > pivot)
        Q_Sort(data, pivot + 1, right);
}
```



```
Pseudocode

Kamus/Deklarasi Variabel fungsi Q_sort
-

Algoritma/Deskripsi fungsi Q_sort(Data[],int,int)

Kamus/Deklarasi Variabel
d, iii = int

Algoritma/Deskripsi

Data dataList[9] = {65, 2, 44, 26, 19, 22, 5, 3, 12}
for (d = 0; d < 9; d++)
        print ( dataList[d].value)
endfor
for (iii = 0; iii < 9; iii++)
        print (dataList[iii].value)
endfor
return 0

Kamus/Deklarasi Variabel fungsi Q_sort
data[] = data
left, right = int
pivot, left_hold, right_hold = int
```

```
Algoritma/Deskripsi fungsi Q_sort(data[], left,right)
left_hold = left
right_hold = right
pivot = data[left].value
while (left < right)
        while (data[right].value >= pivot && left < right)
          right–
        if (left != right)
          data[left].value = data[right].value
            left++
        endif
        while (data[left].value <= pivot && left < right)
          left++
        if (left != right)
          data[right].value = data[left].value
            right–
        endif
endwhile
data[left].value = pivot;
pivot = left;
left = left_hold;
right = right_hold;
        if (left < pivot)
                Q_Sort(data, left, pivot - 1);
        if (right > pivot)
                Q_Sort(data, pivot + 1, right)
```

Algoritma

1. Membuat fungsi utama
2. dataList[9] = {65, 2, 44, 26, 19, 22, 5, 3, 12}
3. Selama (d=0) maka kerjakan baris 4 s.d 5
4. Mencetak/Menampilkan dataList[d].value
5. d++
6. Memanggil fungsi Q_sort(dataList,0,8)
7. Selama (iii=0) maka kerjakan baris 8 s.d 9
8. Mencetak/Menampilkan dataList[iii].value
9. iii++
10. Membuat fungsi Q_Sort(data[], left, right)
11. left_hold = left
12. right_hold = right
13. pivot = data[left].value
14. Selama (left<right) maka kerjkaan baris 15 s.d 24
15. Selama (data[right].value >= pivot && left<right)
16. right–
17. Jika(left<= right) maka kerjkan baris 18 s.d 19
18. data[left].value = data[right].value
19. left++
20. Selama (data[left].value <= pivot && left < right)
21. left++
22. Jika (left<=right) maka kerjkan baris
23. data[right].value = data[left].value
24. right–
25. data[left].value = pivot
26. pivot = left
27. left = left_hold
28. right = right_hold
29. Jika (left<pivot)
30. Memanggil fungsi Q_sort(data,left,pivot-1)
31. Jika (right>pivot)
32. Memanggil fungsi Q_sort(data,pivot+1,right)

pasd6-25

cnthprak06-1.cpp | cnthprak06-2.cpp | cnthprak07-1.cpp | pasd6-4.cpp | pasd6-25.cpp

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

class MergeSort {
private:
    void merge(int arr[], int kiri, int mid, int kanan) {
        int n1 = mid - kiri + 1;
        int n2 = kanan - mid;

        int L[n1], R[n2];

        for (int i = 0; i < n1; i++) {
            L[i] = arr[kiri + i];
        }
        for (int j = 0; j < n2; j++) {
            R[j] = arr[mid + 1 + j];
        }

        int i = 0;
        int j = 0;
        int k = kiri;

        while (i < n1 && j < n2) {
            if (L[i] >= R[j]) {
                arr[k] = L[i];
                i++;
            } else {
                arr[k] = R[j];
                j++;
            }
            k++;
        }
        while (i < n1) {
            arr[k] = L[i];
            i++;
            k++;
        }
        while (j < n2) {
            arr[k] = R[j];
            j++;
            k++;
        }
    }

    void mergesort(int arr[], int kiri, int kanan) {
        if (kiri < kanan) {
            int mid = kiri + (kanan - kiri) / 2;
            mergesort(arr, kiri, mid);
            mergesort(arr, mid + 1, kanan);
            merge(arr, kiri, mid, kanan);
        }
    }

public:
    void sort(int arr[], int arrsize) {
        mergesort(arr, 0, arrsize - 1);
    }
};

void cetakdata(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main() {
    int arr[] = {65, 2, 44, 26, 19, 22, 5, 3, 12};
    int arrsize = sizeof(arr) / sizeof(arr[0]);

    cout << "Data Sebelum diurutkan:" << endl;
    cout << "~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;
    cetakdata(arr, arrsize);

    MergeSort sorter;
    sorter.sort(arr, arrsize);

    cout << endl << endl;
    cout << "Data Setelah diurutkan secara descending:" << endl;
    cetakdata(arr, arrsize);

    return 0;
}
```

```
Command Prompt

F:\>g++ pasd6-25.cpp -o 1

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


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

F:\>
```

Pseudcode

```
class MergeSort
private :
        Kamus/Deklarasi Variabel fungsi merge
        arr[],kiri,mid,kanan,n1,n2,L,R,i,j,k = 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,kanan)

        if(kiri<kanan)
                mid = kiri + (kanan - kiri) / 2
                mergesort(arr, kiri, mid)
                mergesort(arr, mid + 1, kanan)
                merge(arr, kiri, mid, kanan)
        endif

public :
        Kamus/Deklarasi Variabel fungsi sort
        arr[], arrsize = int

        Algoritma/Deskripsi fungsi sort(arr[],arrsize)
        mergesort(arr,0,arrsize-1)

Kamus/Deklarasi Variabel fungsi 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 sorter
   sorter.sort(arr, arrsize)
   cetakdata(arr, arrsize)
   return
```

Algoritma :

1. Membuat class MergeSort
2. Mendeklarasikan class MergeSort dengan kata kunci private
3. Membuat fungsi merge(arr[],kiri,mid,kanan)
4. n1 = mid - kiri+1
5. n2 = kanan - mid
6. Selama (i=0) maka kerjakan baris 7 s.d 8
7. L[i] = arr[kiri + i]
8. i++
9. Selama (j=0) maka kerjkan baris 10 s.d 11
10. R[j] = arr[mid + 1 + j]
11. j++
12. i=0
13. j=0
14. k=kiri
15. Selama (i<n1&&j<n2) maka kerjkan baris 16 s.d 21
16. Jika (L[i]>=R[j]) maka kerjkan baris 17 s.d 18 kalau tidak 19 s.d 20
17. arr[k]=L[i]
18. i++
19. arr[k]=R[j]
20. j++
21. k++
22. Selama (i<n1) maka kerjakan baris 23 s.d 25
23. arr[k]=L[i]
24. i++
25. k++
26. Selama (j<n2) maka kerjkan baris 27 s.d 29
27. arr[k]=R[j]
28. j++
29. k++
30. Membuat fungsi megesort(arr[],kiri,kanan)
31. Jika (kiri<kanan) maka kerjakan baris 32 s.d 35
32. mid = kiri + (kanan - kiri) / 2
33. mergesort(arr, kiri, mid)
34. mergesort(arr, mid + 1, kanan)
35. merge(arr, kiri, mid, kanan)

36. Mendeklarasikan class MergeSort dengan kata kunci Public
37. Membuat fungsi sort(arr[],arrsize)
38. Memanggil fungsi mergesort(arr,0,arrsize-1)
39. Membuat fungsi cetakdata(arr[],size)
40. Selama (i=0) maka kerjakan baris 41
41. Mencetak/Menampilkan Nilai arr[i]
42. Membuat fungsi utama
43. arr[] = {65, 2, 44, 26, 19, 22, 5, 3, 12}
44. arrsize = sizeof(arr) / sizeof(arr[0])
45. Membuat fungsi cetakdata (arr,arrsize)
46. Membuat fungsi Mergesort sorter
47. Membuat fungsi sorter.sort (arr,arrsize)
48. Membuat fungsi cetakdata (arr,arrsize)
49. Selesai