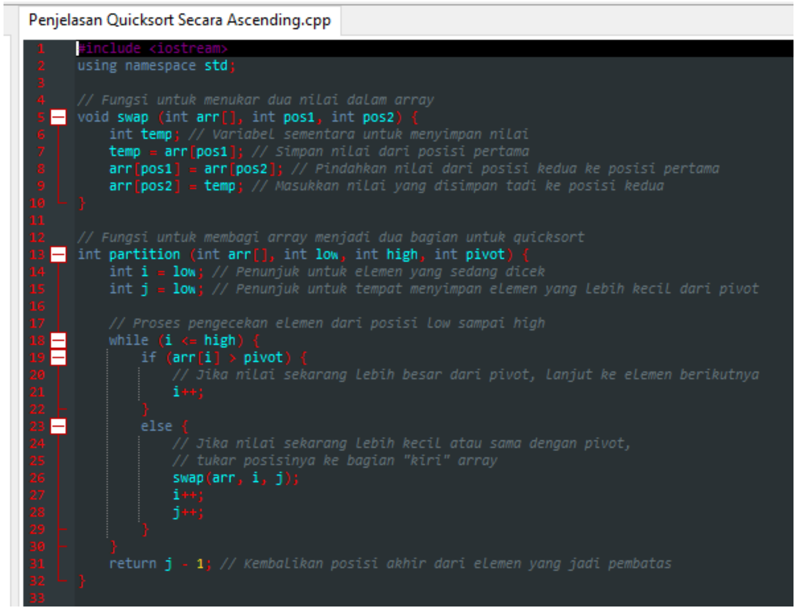
Nama : Muchamad Rofii

NPM : 242310030

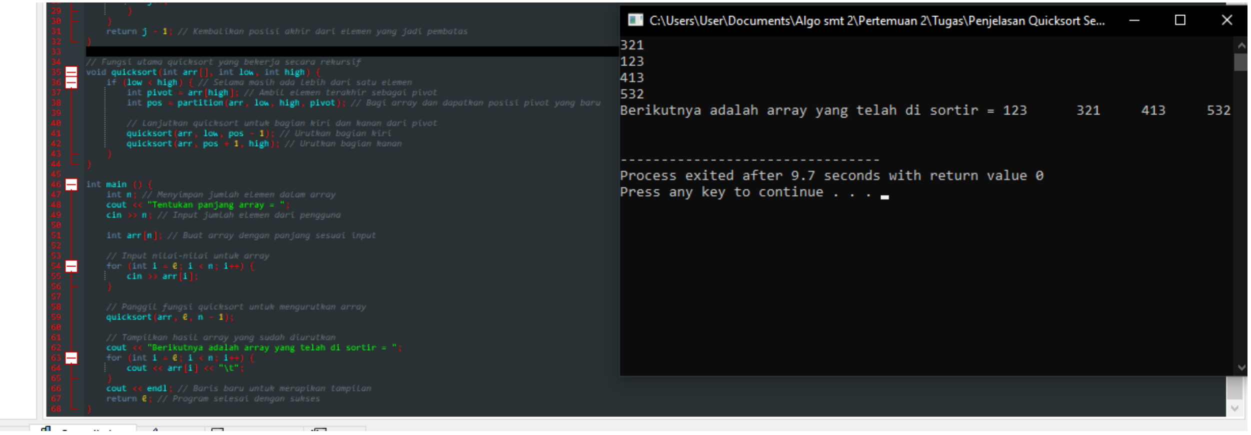
Kelas : TI-24-PA

Matkul : Desain Analisis Algoritma

Quicksort Asc:



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

// Fungsi untuk menukar dua nilai dalam array
void swap (int arr[], int pos1, int pos2) {
    int temp; // Variabel sementara untuk menyimpan nilai
    temp = arr[pos1]; // Simpan nilai dari posisi pertama
    arr[pos1] = arr[pos2]; // Pindahkan nilai dari posisi kedua ke posisi pertama
    arr[pos2] = temp; // Masukkan nilai yang disimpan tadi ke posisi kedua
}

// Fungsi untuk membagi array menjadi dua bagian untuk quicksort
int partition (int arr[], int low, int high, int pivot) {
    int i = low; // Penunjuk untuk elemen yang sedang dicek
    int j = low; // Penunjuk untuk tempat menyimpan elemen yang lebih kecil dari pivot

    // Proses pengecekan elemen dari posisi low sampai high
    while (i <= high) {
        if (arr[i] > pivot) {
            // Jika nilai sekarang lebih besar dari pivot, lanjut ke elemen berikutnya
            i++;
        }
        else {
            // Jika nilai sekarang lebih kecil atau sama dengan pivot,
            // tukar posisinya ke bagian "kiri" array
            swap(arr, i, j);
            i++;
            j++;
        }
    }
    return j - 1; // Kembalikan posisi akhir dari elemen yang jadi pembatas
}

```



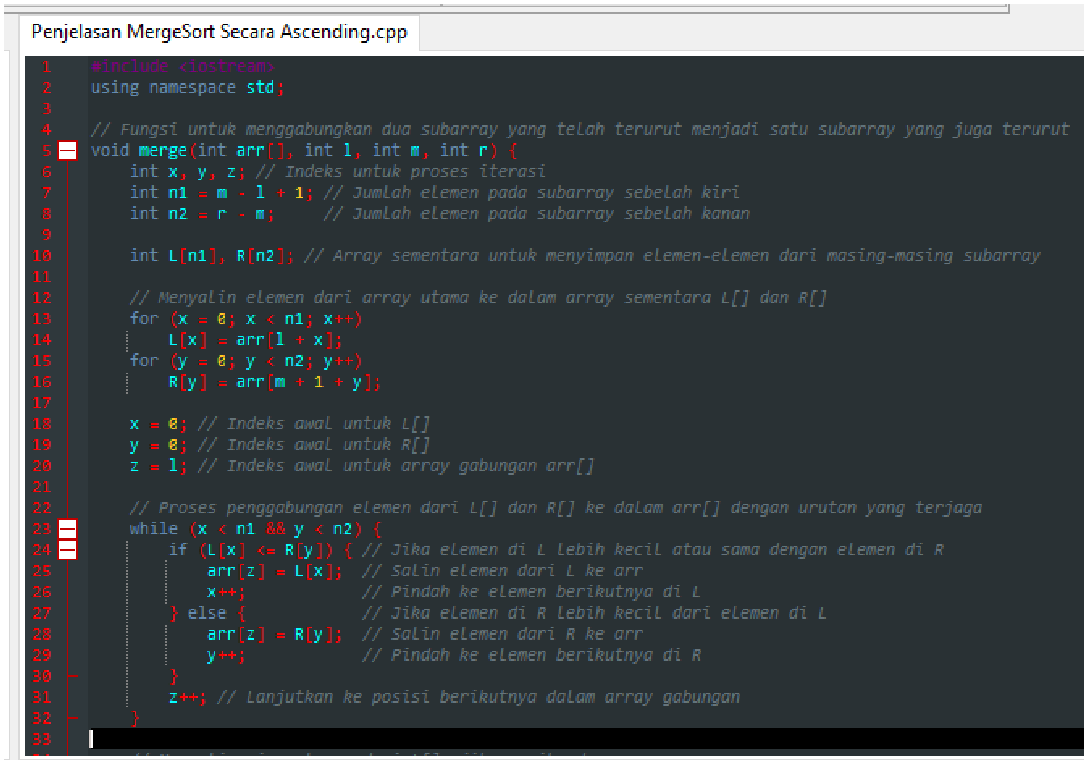
```cpp
        }
    }
    return j - 1; // Kembalikan posisi akhir dari elemen yang jadi pembatas
}
```

```cpp

// Fungsi utama quicksort yang bekerja secara rekursif
void quicksort(int arr[], int low, int high) {
    if (low < high) { // Selama masih ada lebih dari satu elemen
        int pivot = arr[high]; // Ambil elemen terakhir sebagai pivot
        int pos = partition(arr, low, high, pivot); // Bagi array dan dapatkan posisi pivot yang baru

        // Lanjutkan quicksort untuk bagian kiri dan kanan dari pivot
        quicksort(arr, low, pos - 1); // Urutkan bagian kiri
        quicksort(arr, pos + 1, high); // Urutkan bagian kanan
    }
}

int main () {
    int n; // Menyimpan jumlah elemen dalam array
    cout << "Tentukan panjang array = ";
    cin >> n; // Input jumlah elemen dari pengguna

    int arr[n]; // Buat array dengan panjang sesuai input

    // Input nilai-nilai untuk array
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    // Panggil fungsi quicksort untuk mengurutkan array
    quicksort(arr, 0, n - 1);

    // Tampilkan hasil array yang sudah diurutkan
    cout << "Berikutnya adalah array yang telah di sortir = ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << "\t";
    }
    cout << endl; // Baris baru untuk merapikan tampilan
    return 0; // Program selesai dengan sukses
}
```

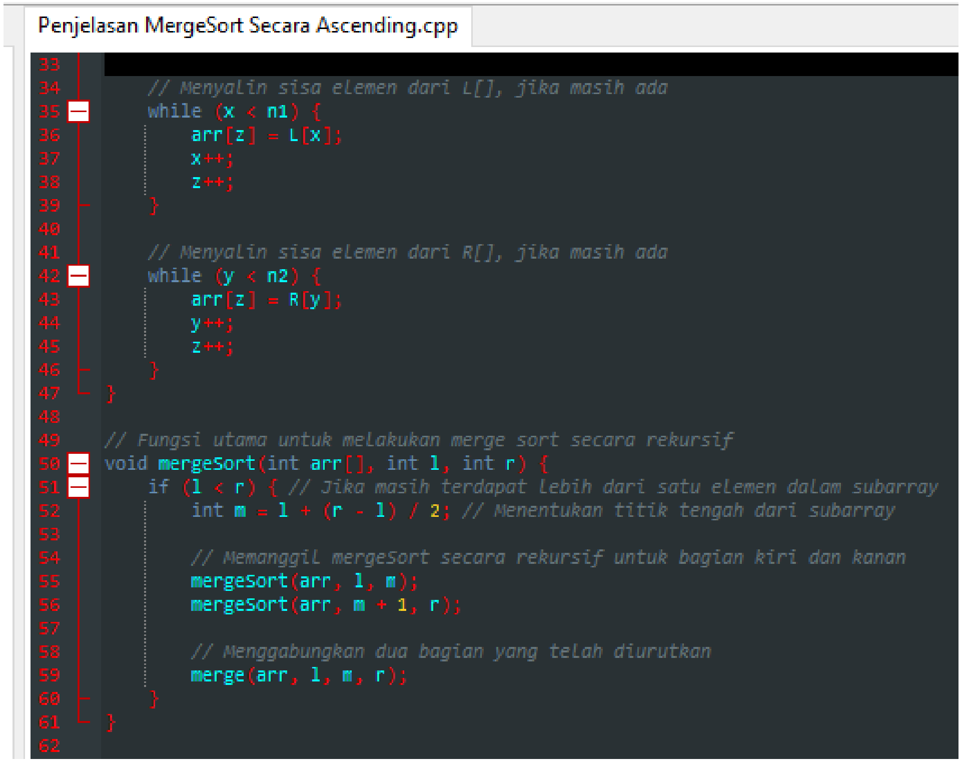
C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\Penjelasan Quicksort Se...

```
321
123
413
532
Berikutnya adalah array yang telah di sortir = 123      321     413     532


--------------------------------
Process exited after 9.7 seconds with return value 0
Press any key to continue . . .
```

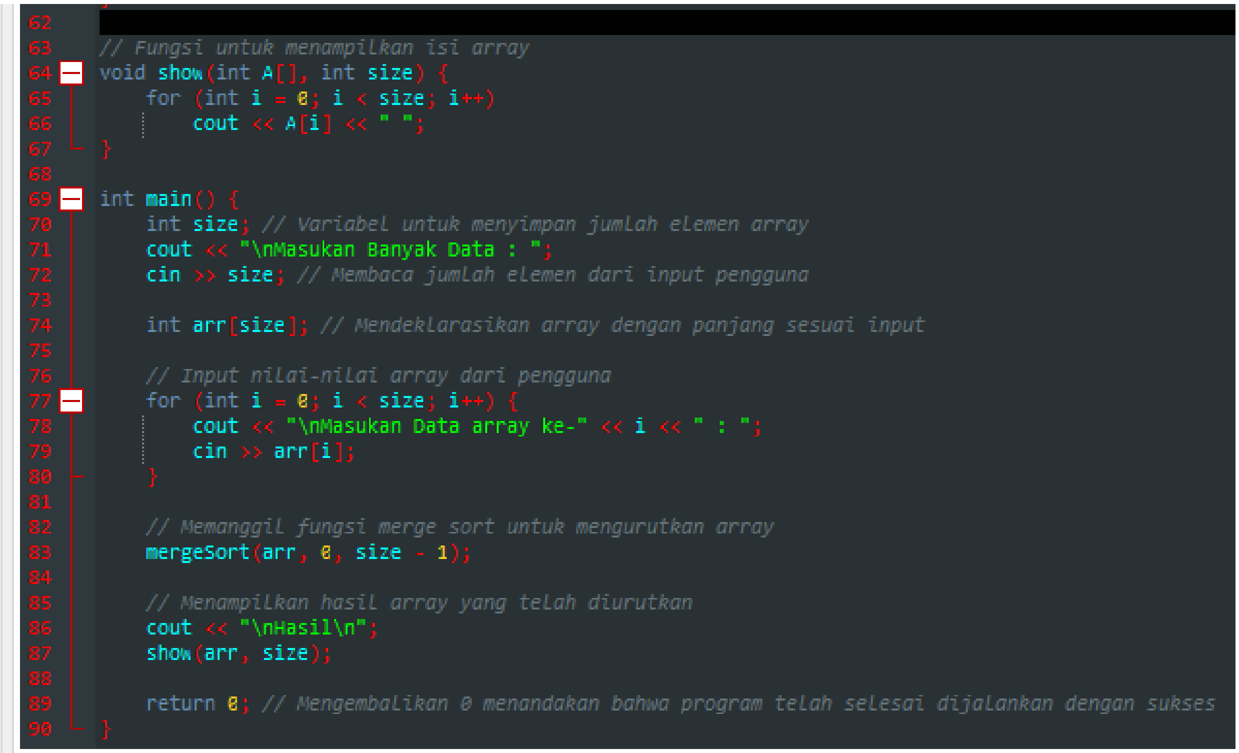
MergeSort:



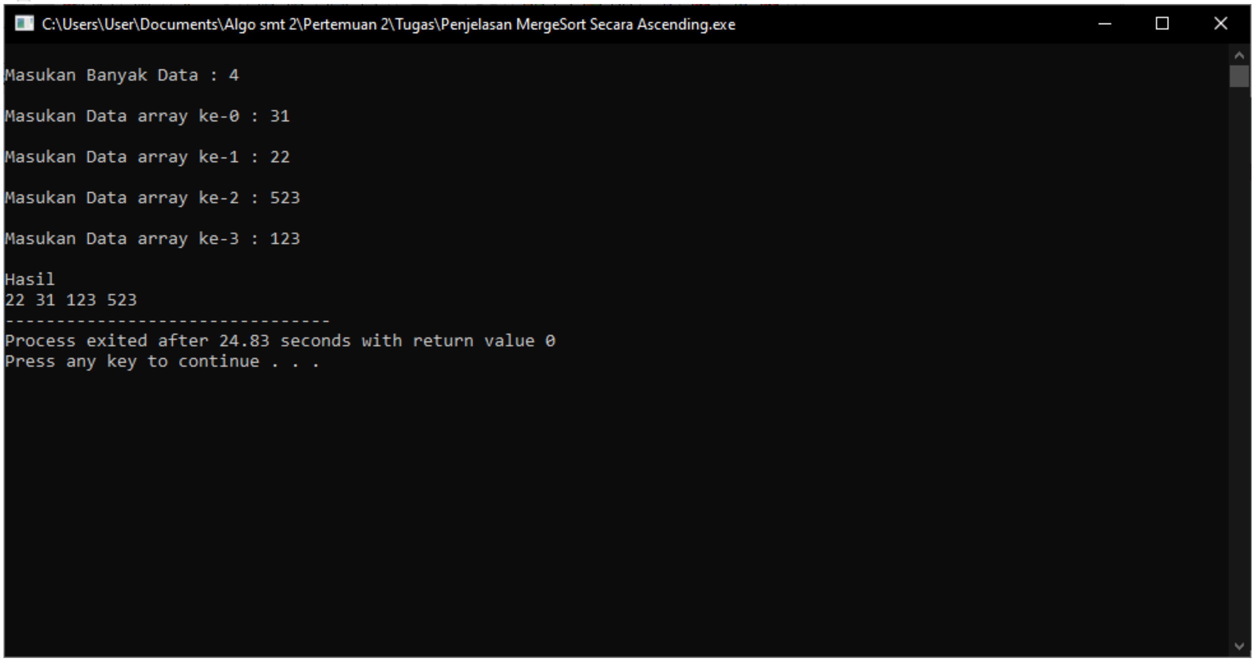
```cpp
#include <iostream>
using namespace std;

// Fungsi untuk menggabungkan dua subarray yang telah terurut menjadi satu subarray yang juga terurut
void merge(int arr[], int l, int m, int r) {
    int x, y, z; // Indeks untuk proses iterasi
    int n1 = m - l + 1; // Jumlah elemen pada subarray sebelah kiri
    int n2 = r - m;     // Jumlah elemen pada subarray sebelah kanan

    int L[n1], R[n2]; // Array sementara untuk menyimpan elemen-elemen dari masing-masing subarray

    // Menyalin elemen dari array utama ke dalam array sementara L[] dan R[]
    for (x = 0; x < n1; x++)
        L[x] = arr[l + x];
    for (y = 0; y < n2; y++)
        R[y] = arr[m + 1 + y];

    x = 0; // Indeks awal untuk L[]
    y = 0; // Indeks awal untuk R[]
    z = l; // Indeks awal untuk array gabungan arr[]

    // Proses penggabungan elemen dari L[] dan R[] ke dalam arr[] dengan urutan yang terjaga
    while (x < n1 && y < n2) {
        if (L[x] <= R[y]) { // Jika elemen di L lebih kecil atau sama dengan elemen di R
            arr[z] = L[x];  // Salin elemen dari L ke arr
            x++;            // Pindah ke elemen berikutnya di L
        } else {            // Jika elemen di R lebih kecil dari elemen di L
            arr[z] = R[y];  // Salin elemen dari R ke arr
            y++;            // Pindah ke elemen berikutnya di R
        }
        z++; // Lanjutkan ke posisi berikutnya dalam array gabungan
    }

```



```cpp

    // Menyalin sisa elemen dari L[], jika masih ada
    while (x < n1) {
        arr[z] = L[x];
        x++;
        z++;
    }

    // Menyalin sisa elemen dari R[], jika masih ada
    while (y < n2) {
        arr[z] = R[y];
        y++;
        z++;
    }
}

// Fungsi utama untuk melakukan merge sort secara rekursif
void mergeSort(int arr[], int l, int r) {
    if (l < r) { // Jika masih terdapat lebih dari satu elemen dalam subarray
        int m = l + (r - l) / 2; // Menentukan titik tengah dari subarray

        // Memanggil mergeSort secara rekursif untuk bagian kiri dan kanan
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);

        // Menggabungkan dua bagian yang telah diurutkan
        merge(arr, l, m, r);
    }
}

```



```cpp

// Fungsi untuk menampilkan isi array
void show(int A[], int size) {
    for (int i = 0; i < size; i++)
        cout << A[i] << " ";
}

int main() {
    int size; // Variabel untuk menyimpan jumlah elemen array
    cout << "\nMasukan Banyak Data : ";
    cin >> size; // Membaca jumlah elemen dari input pengguna

    int arr[size]; // Mendeklarasikan array dengan panjang sesuai input

    // Input nilai-nilai array dari pengguna
    for (int i = 0; i < size; i++) {
        cout << "\nMasukan Data array ke-" << i << " : ";
        cin >> arr[i];
    }

    // Memanggil fungsi merge sort untuk mengurutkan array
    mergeSort(arr, 0, size - 1);

    // Menampilkan hasil array yang telah diurutkan
    cout << "\nHasil\n";
    show(arr, size);

    return 0; // Mengembalikan 0 menandakan bahwa program telah selesai dijalankan dengan sukses
}
```

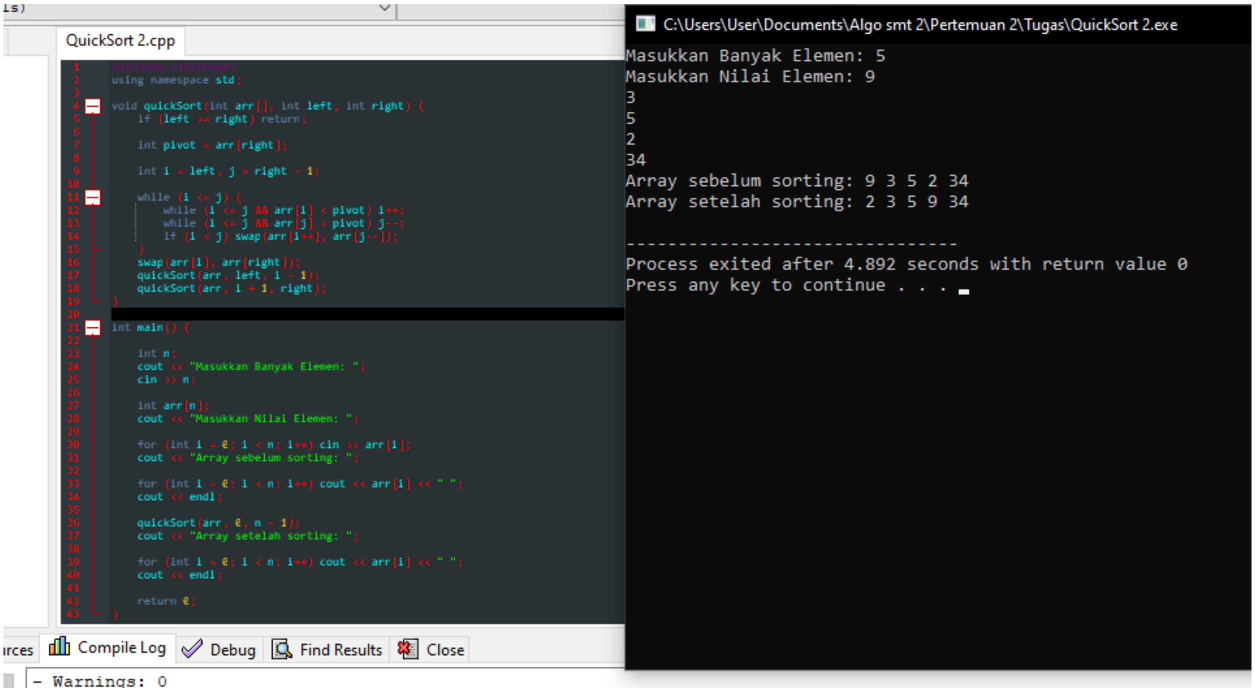


C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\Penjelasan MergeSort Secara Ascending.exe

```

Masukan Banyak Data : 4

Masukan Data array ke-0 : 31

Masukan Data array ke-1 : 22

Masukan Data array ke-2 : 523

Masukan Data array ke-3 : 123

Hasil
22 31 123 523
--------------------------------
Process exited after 24.83 seconds with return value 0
Press any key to continue . . .
```

B.

QuickSort:

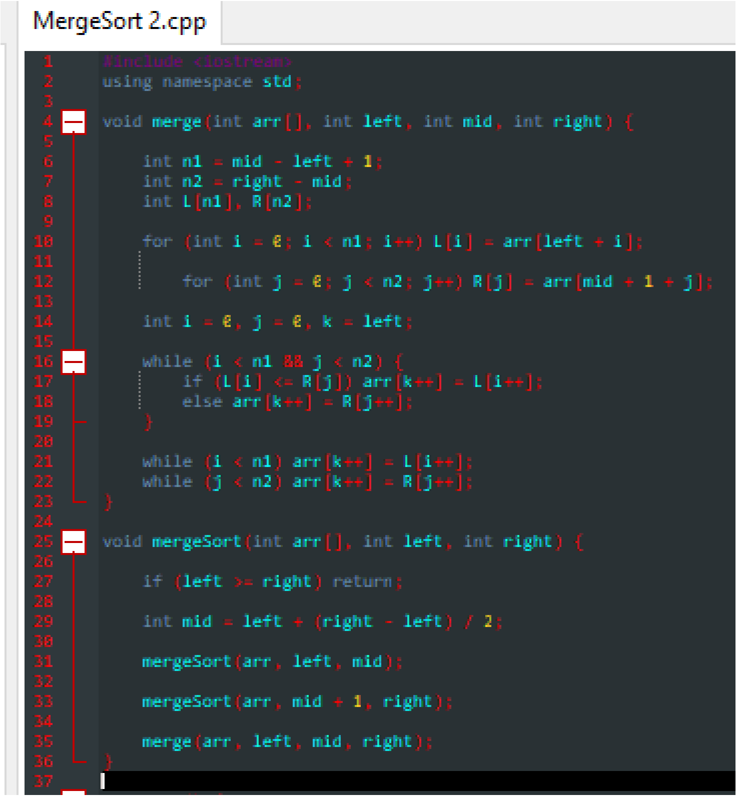


```cpp
#include <iostream>
using namespace std;

void quickSort(int arr[], int left, int right) {
    if (left >= right) return;

    int pivot = arr[right];

    int i = left, j = right - 1;

    while (i <= j) {
        while (i <= j && arr[i] < pivot) i++;
        while (i <= j && arr[j] > pivot) j--;
        if (i < j) swap(arr[i++], arr[j--]);
    }
    swap(arr[i], arr[right]);
    quickSort(arr, left, i - 1);
    quickSort(arr, i + 1, right);
}

int main() {

    int n;
    cout << "Masukkan Banyak Elemen: ";
    cin >> n;

    int arr[n];
    cout << "Masukkan Nilai Elemen: ";

    for (int i = 0; i < n; i++) cin >> arr[i];
    cout << "Array sebelum sorting: ";

    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    cout << endl;

    quickSort(arr, 0, n - 1);
    cout << "Array setelah sorting: ";

    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    cout << endl;

    return 0;
}
```

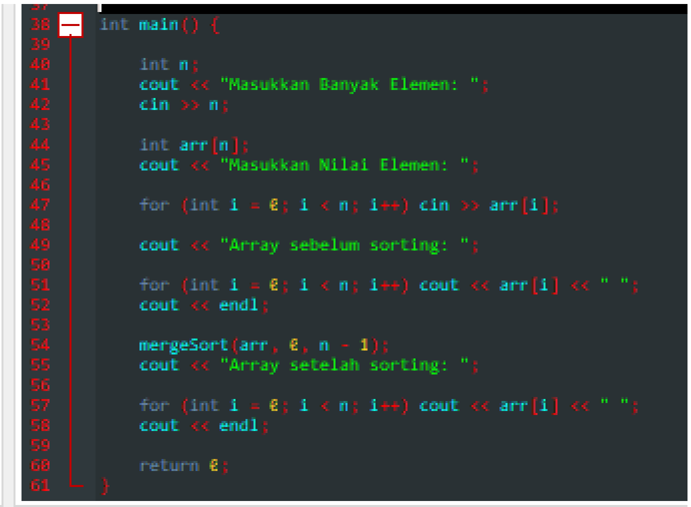
```
- Warnings: 0
```

```
C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\QuickSort 2.exe
Masukkan Banyak Elemen: 5
Masukkan Nilai Elemen: 9
3
5
2
34
Array sebelum sorting: 9 3 5 2 34
Array setelah sorting: 2 3 5 9 34

--------------------------------
Process exited after 4.892 seconds with return value 0
Press any key to continue . . .
```

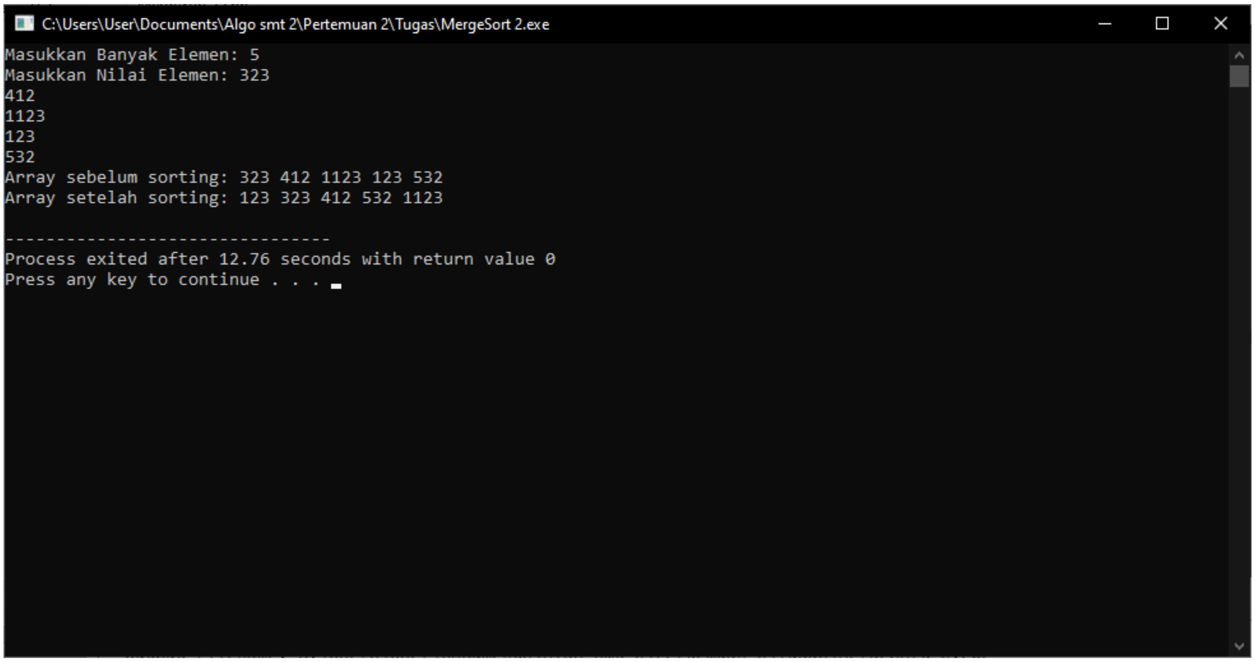
MergeSort:



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

void merge(int arr[], int left, int mid, int right) {

    int n1 = mid - left + 1;
    int n2 = right - mid;
    int L[n1], R[n2];

    for (int i = 0; i < n1; i++) L[i] = arr[left + i];

        for (int j = 0; j < n2; j++) R[j] = arr[mid + 1 + j];

    int i = 0, j = 0, k = left;

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

    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = R[j++];
}

void mergeSort(int arr[], int left, int right) {

    if (left >= right) return;

    int mid = left + (right - left) / 2;

    mergeSort(arr, left, mid);

    mergeSort(arr, mid + 1, right);

    merge(arr, left, mid, right);
}
```



```cpp
int main() {

    int n;
    cout << "Masukkan Banyak Elemen: ";
    cin >> n;

    int arr[n];
    cout << "Masukkan Nilai Elemen: ";

    for (int i = 0; i < n; i++) cin >> arr[i];

    cout << "Array sebelum sorting: ";

    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    cout << endl;

    mergeSort(arr, 0, n - 1);
    cout << "Array setelah sorting: ";

    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    cout << endl;

    return 0;
}
```

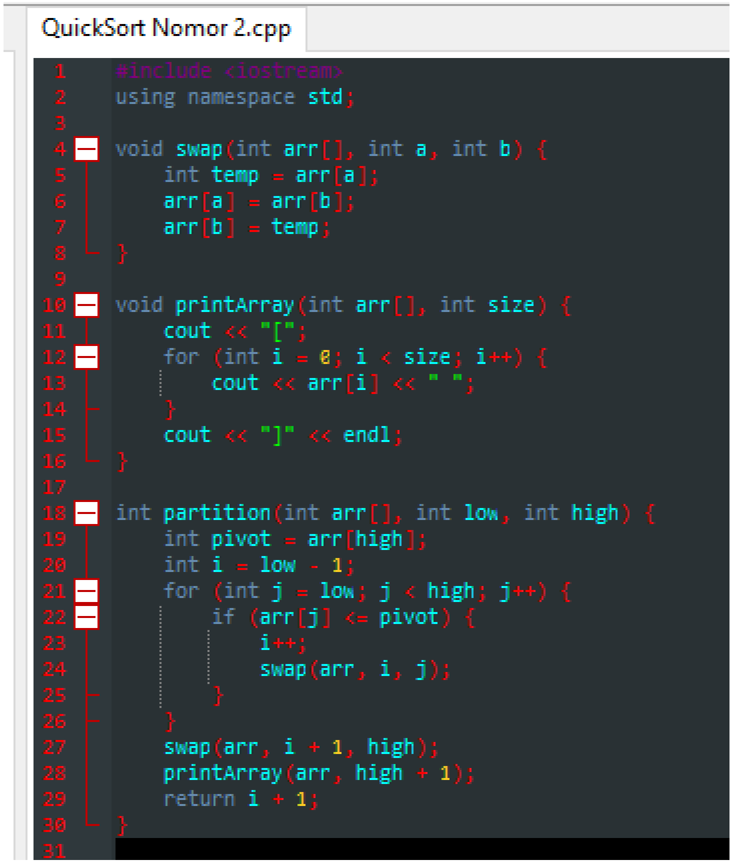


C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\MergeSort 2.exe

```
Masukkan Banyak Elemen: 5
Masukkan Nilai Elemen: 323
412
1123
123
532
Array sebelum sorting: 323 412 1123 123 532
Array setelah sorting: 123 323 412 532 1123

--------------------------------
Process exited after 12.76 seconds with return value 0
Press any key to continue . . .
```

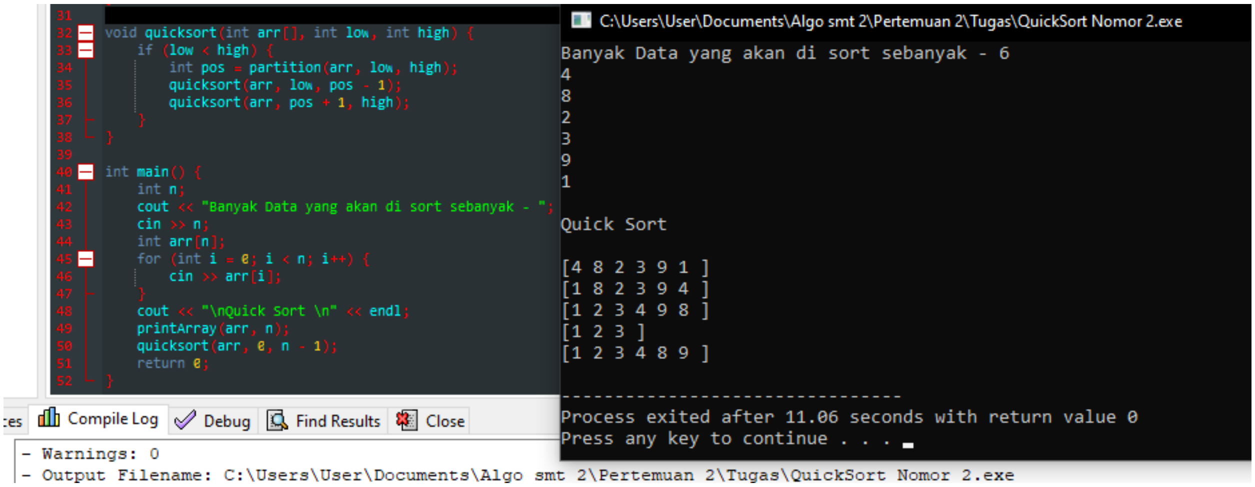
Nomor 2:

QuickSort:



QuickSort Nomor 2.cpp

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

void swap(int arr[], int a, int b) {
    int temp = arr[a];
    arr[a] = arr[b];
    arr[b] = temp;
}

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

int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = low - 1;
    for (int j = low; j < high; j++) {
        if (arr[j] <= pivot) {
            i++;
            swap(arr, i, j);
        }
    }
    swap(arr, i + 1, high);
    printArray(arr, high + 1);
    return i + 1;
}

```



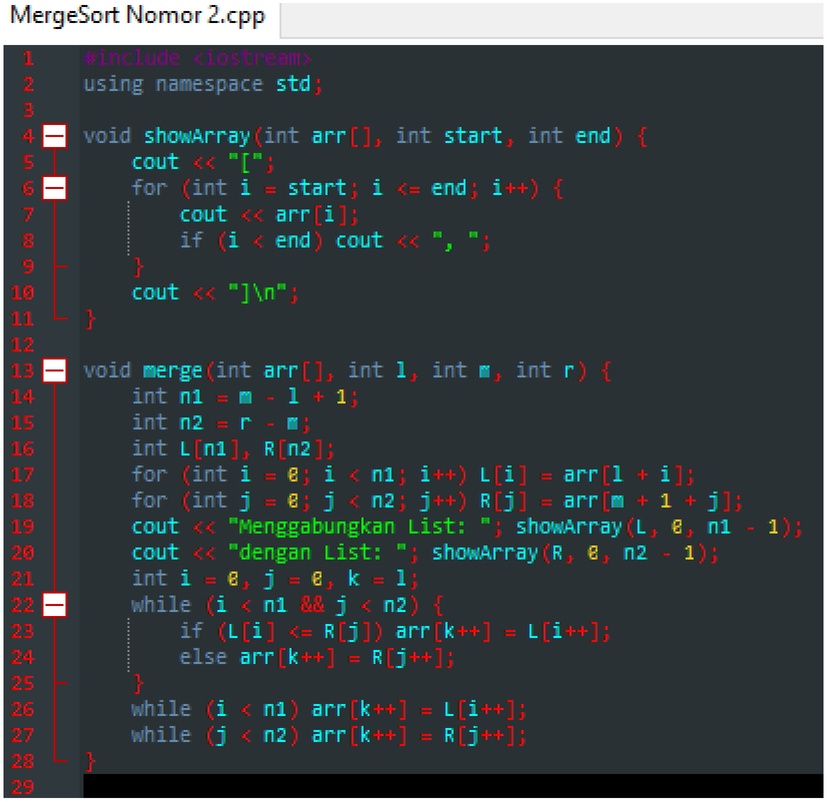
```cpp

void quicksort(int arr[], int low, int high) {
    if (low < high) {
        int pos = partition(arr, low, high);
        quicksort(arr, low, pos - 1);
        quicksort(arr, pos + 1, high);
    }
}

int main() {
    int n;
    cout << "Banyak Data yang akan di sort sebanyak - ";
    cin >> n;
    int arr[n];
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    cout << "\nQuick Sort \n" << endl;
    printArray(arr, n);
    quicksort(arr, 0, n - 1);
    return 0;
}
```

C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\QuickSort Nomor 2.exe

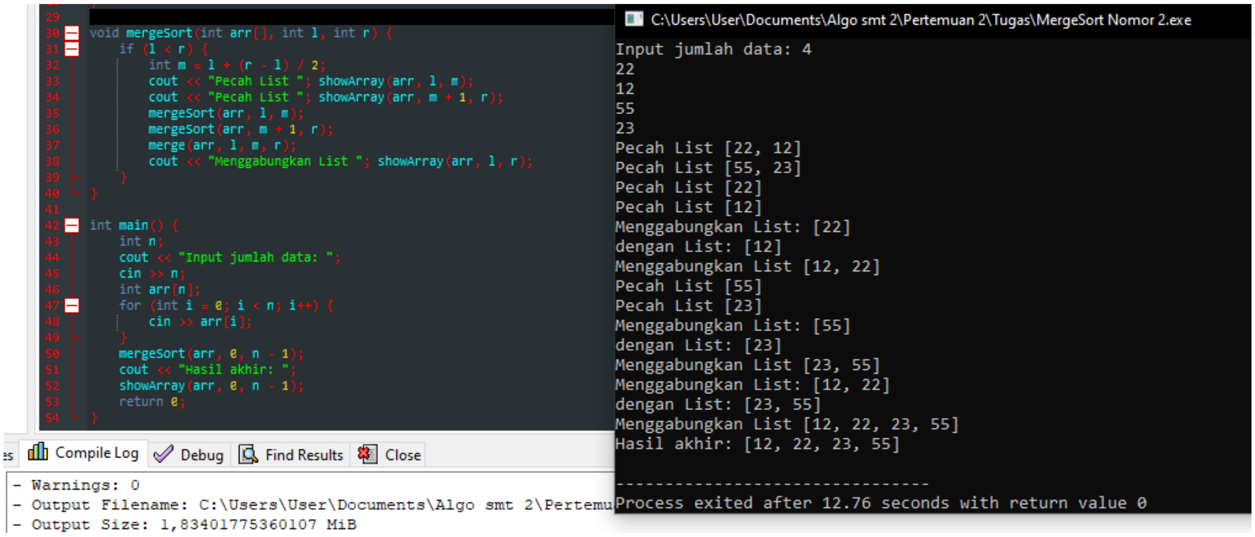
```
Banyak Data yang akan di sort sebanyak - 6
4
8
2
3
9
1

Quick Sort

[4 8 2 3 9 1 ]
[1 8 2 3 9 4 ]
[1 2 3 4 9 8 ]
[1 2 3 ]
[1 2 3 4 8 9 ]

--------------------------------
Process exited after 11.06 seconds with return value 0
Press any key to continue . . .
```

```
- Warnings: 0
- Output Filename: C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\QuickSort Nomor 2.exe
```

MergeSort:



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

void showArray(int arr[], int start, int end) {
    cout << "[";
    for (int i = start; i <= end; i++) {
        cout << arr[i];
        if (i < end) cout << ", ";
    }
    cout << "]\n";
}

void merge(int arr[], int l, int m, int r) {
    int n1 = m - l + 1;
    int n2 = r - m;
    int L[n1], R[n2];
    for (int i = 0; i < n1; i++) L[i] = arr[l + i];
    for (int j = 0; j < n2; j++) R[j] = arr[m + 1 + j];
    cout << "Menggabungkan List: "; showArray(L, 0, n1 - 1);
    cout << "dengan List: "; showArray(R, 0, n2 - 1);
    int i = 0, j = 0, k = l;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) arr[k++] = L[i++];
        else arr[k++] = R[j++];
    }
    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = R[j++];
}
```


C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\MergeSort Nomor 2.exe
Compile Log
Debug
Find Results
Close

```cpp
void mergeSort(int arr[], int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        cout << "Pecah List "; showArray(arr, l, m);
        cout << "Pecah List "; showArray(arr, m + 1, r);
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);
        merge(arr, l, m, r);
        cout << "Menggabungkan List "; showArray(arr, l, r);
    }
}

int main() {
    int n;
    cout << "Input jumlah data: ";
    cin >> n;
    int arr[n];
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    mergeSort(arr, 0, n - 1);
    cout << "Hasil akhir: ";
    showArray(arr, 0, n - 1);
    return 0;
}
```

```
Input jumlah data: 4
22
12
55
23
Pecah List [22, 12]
Pecah List [55, 23]
Pecah List [22]
Pecah List [12]
Menggabungkan List: [22]
dengan List: [12]
Menggabungkan List [12, 22]
Pecah List [55]
Pecah List [23]
Menggabungkan List: [55]
dengan List: [23]
Menggabungkan List [23, 55]
Menggabungkan List: [12, 22]
dengan List: [23, 55]
Menggabungkan List [12, 22, 23, 55]
Hasil akhir: [12, 22, 23, 55]

--------------------------------
Process exited after 12.76 seconds with return value 0
```

```
- Warnings: 0
- Output Filename: C:\Users\User\Documents\Algo smt 2\Pertemu
- Output Size: 1,83401775360107 MiB
```