

LAPORAN TUGAS UAS STRUKTUR DATA



Muhammad Dhafa Jawadil Ubaid (21091397058)

UNIVERSITAS NEGERI SURABAYA

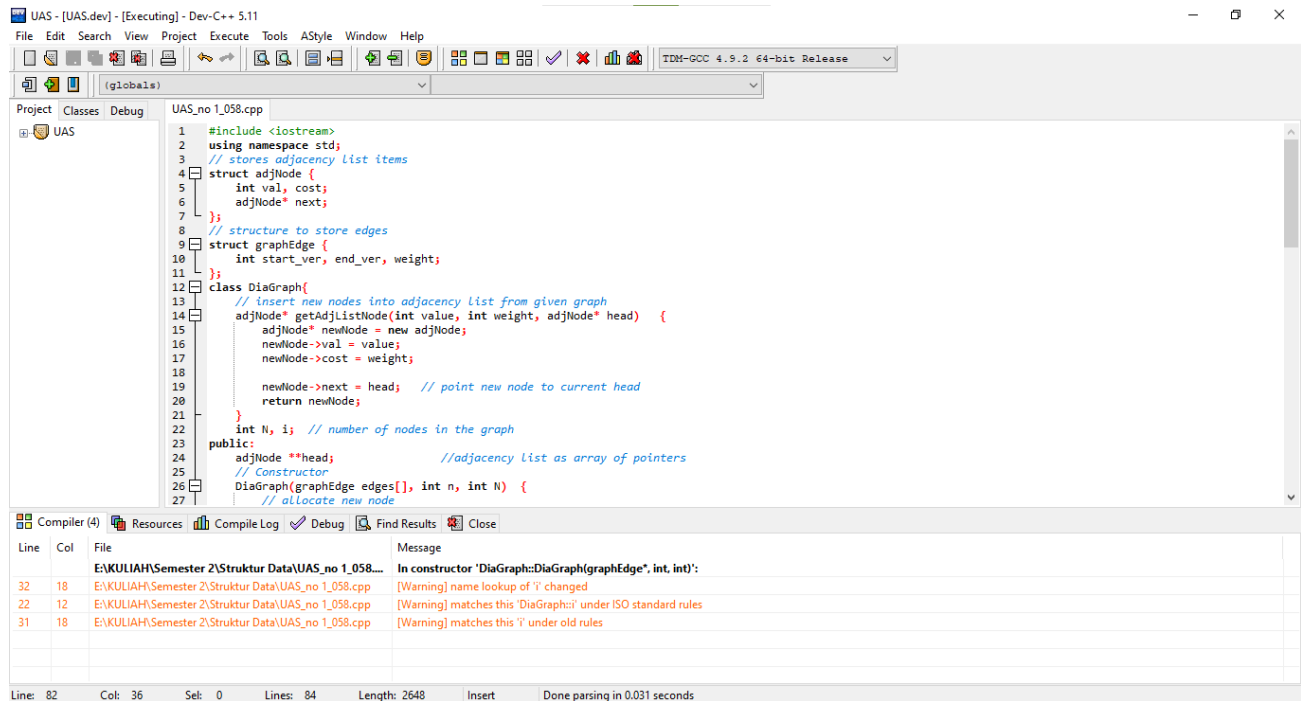
MANAJEMEN INFORMATIKA

2021

1. Input :

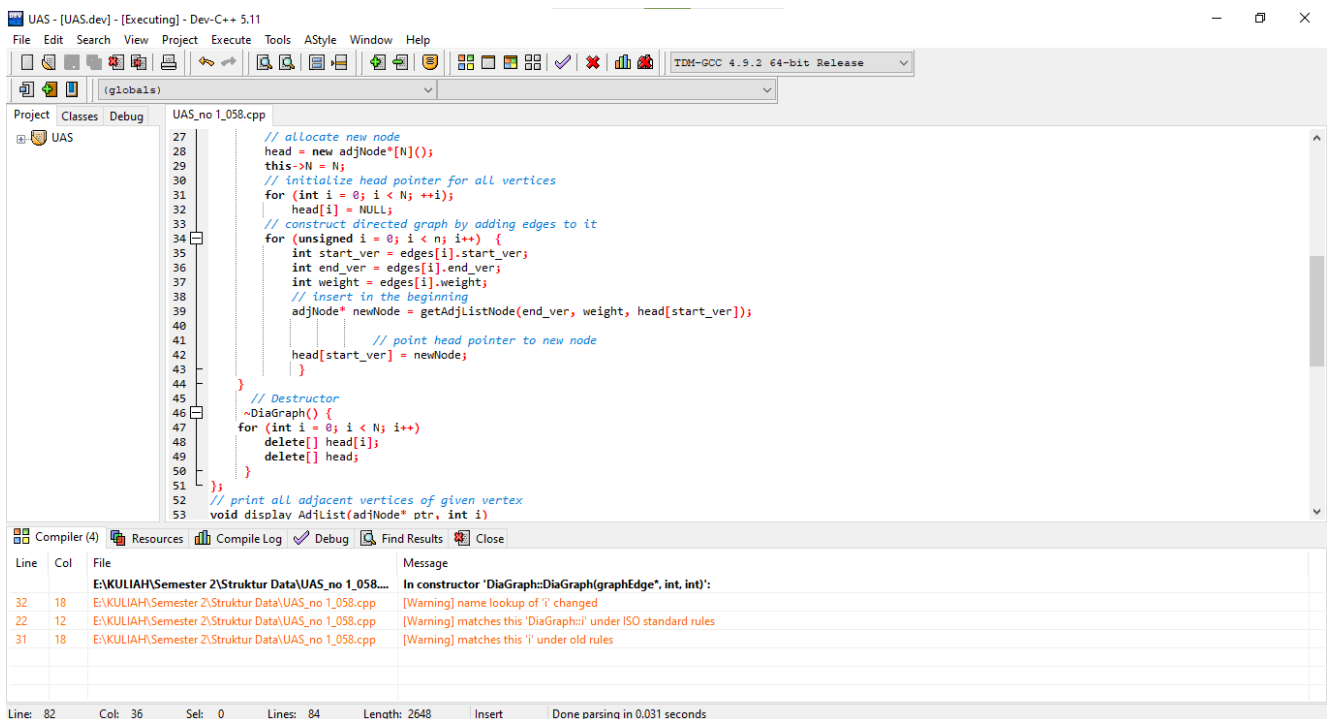
0. int jumlah vertex yang ada dalam graph

1. (x,y) dengan x = vertex 1, y = vertex 2, w = weight



The screenshot shows the Dev-C++ IDE with the file 'UAS_no 1_058.cpp' open. The code defines a directed graph structure. It includes `<iostream>` and uses the `std` namespace. It defines an `adjNode` struct with `int val, cost;` and a pointer `adjNode* next;`. Then it defines a `graphEdge` struct with `int start_ver, end_ver, weight;`. The `DiaGraph` class has a private member `adjNode* head;` and a public constructor `DiaGraph(graphEdge edges[], int n, int N)`. The constructor logic is partially visible, showing the initialization of the `head` array.

```
1 #include <iostream>
2 using namespace std;
3 // stores adjacency list items
4 struct adjNode {
5     int val, cost;
6     adjNode* next;
7 };
8 // structure to store edges
9 struct graphEdge {
10     int start_ver, end_ver, weight;
11 };
12 class DiaGraph{
13     // insert new nodes into adjacency list from given graph
14     adjNode* getAdjListNode(int value, int weight, adjNode* head) {
15         adjNode* newNode = new adjNode;
16         newNode->val = value;
17         newNode->cost = weight;
18         newNode->next = head; // point new node to current head
19         return newNode;
20     }
21     int N, i; // number of nodes in the graph
22 public:
23     adjNode** head; // adjacency list as array of pointers
24     // Constructor
25     DiaGraph(graphEdge edges[], int n, int N) {
26         // allocate new node
```



This screenshot shows the continuation of the C++ code from the previous block. It completes the constructor logic by iterating through the edges and adding them to the adjacency list. It also shows the destructor and a function to display the adjacency list.

```
27 // allocate new node
28 head = new adjNode*[N];
29 this->N = N;
30 // initialize head pointer for all vertices
31 for (int i = 0; i < N; ++i)
32     head[i] = NULL;
33 // construct directed graph by adding edges to it
34 for (unsigned i = 0; i < n; i++) {
35     int start_ver = edges[i].start_ver;
36     int end_ver = edges[i].end_ver;
37     int weight = edges[i].weight;
38     // insert in the beginning
39     adjNode* newNode = getAdjListNode(end_ver, weight, head[start_ver]);
40     // point head pointer to new node
41     head[start_ver] = newNode;
42 }
43
44 // Destructor
45 ~DiaGraph() {
46     for (int i = 0; i < N; i++)
47         delete[] head[i];
48     delete[] head;
49 }
50
51 // print all adjacent vertices of given vertex
52 void display AdjList(adjNode* ptr, int i)
```

UAS - [UAS.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Classes Debug UAS_no_1_058.cpp

```
53 void display_AdjList(adjNode* ptr, int i)
54 {
55     while (ptr != NULL) {
56         cout << "(" << i << ", " << ptr->val
57             << ", " << ptr->cost << ") ";
58         ptr = ptr->next;
59     }
60     cout << endl;
61 }
62 // graph implementation
63 int main()
64 {
65     // graph edges array.
66     graphEdge edges[] = {
67         // (x, y, w) -> edge from x to y with weight w
68         {1,2,5},{2,3,1},{4,1,3},{2,4,1},{3,1,1}
69     };
70     int N = 5; // Number of vertices in the graph
71     // calculate number of edges
72     int n = sizeof(edges)/sizeof(edges[0]);
73     // construct graph
74     DiaGraph diagraph(edges, n, N);
75     // print adjacency list representation of graph
76     cout<<"Graph adjacency list "<<endl<<"(start_vertex, end_vertex, weight):"<<endl;
77     for (int i = 0; i < N; i++)
78     {
79         // display adjacent vertices of vertex i
```

Compiler (4) Resources Compile Log Debug Find Results Close

Line	Col	File	Message
32	18	E:\KULIAH\Semester 2\Struktur Data\UAS_no_1_058...	In constructor 'DiaGraph::DiaGraph(graphEdge*, int, int)':
22	12	E:\KULIAH\Semester 2\Struktur Data\UAS_no_1_058.cpp	[Warning] name lookup of 'i' changed
31	18	E:\KULIAH\Semester 2\Struktur Data\UAS_no_1_058.cpp	[Warning] matches this 'DiaGraph::i' under ISO standard rules
			[Warning] matches this 'i' under old rules

Line: 82 Col: 36 Sel: 0 Lines: 84 Length: 2648 Insert Done parsing in 0.031 seconds

E:\KULIAH\Semester 2\Struktur Data\UAS_no_1_058.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Classes Debug UAS_no_1_058.cpp UAS_no_2_058.cpp

```
58     ptr = ptr->next;
59 }
60 cout << endl;
61 }
62 // graph implementation
63 int main()
64 {
65     // graph edges array.
66     graphEdge edges[] = {
67         // (x, y, w) -> edge from x to y with weight w
68         {1,2,5},{2,3,1},{4,1,3},{2,4,1},{3,1,1}
69     };
70     int N = 5; // Number of vertices in the graph
71     // calculate number of edges
72     int n = sizeof(edges)/sizeof(edges[0]);
73     // construct graph
74     DiaGraph diagraph(edges, n, N);
75     // print adjacency list representation of graph
76     cout<<"Graph adjacency list "<<endl<<"(start_vertex, end_vertex, weight):"<<endl;
77     for (int i = 0; i < N; i++)
78     {
79         // display adjacent vertices of vertex i
80         display_AdjList(diagraph.head[i], i);
81     }
82     return 0;
83 }
```

Compiler (4) Resources Compile Log Debug Find Results Close

Line	Col	File	Message
32	18	E:\KULIAH\Semester 2\Struktur Data\UAS_no_1_058...	In constructor 'DiaGraph::DiaGraph(graphEdge*, int, int)':
22	12	E:\KULIAH\Semester 2\Struktur Data\UAS_no_1_058.cpp	[Warning] name lookup of 'i' changed
31	18	E:\KULIAH\Semester 2\Struktur Data\UAS_no_1_058.cpp	[Warning] matches this 'DiaGraph::i' under ISO standard rules
			[Warning] matches this 'i' under old rules

Line: 83 Col: 2 Sel: 0 Lines: 83 Length: 2607 Insert Done parsing in 0.984 seconds

Output :

Satu per satu vertex, edge, dan weightnya

```
E:\KULIAH\Semester 2\Struktur Data\UAS_no 1_058.exe
Graph adjacency list
(start_vertex, end_vertex, weight):
(1, 2, 5)
(2, 4, 1) (2, 3, 1)
(3, 1, 1)
(4, 1, 3)

-----
Process exited after 0.7951 seconds with return value 0
Press any key to continue . . .
```

2. Input :

0. int jumlah vertex yang ada dalam graph

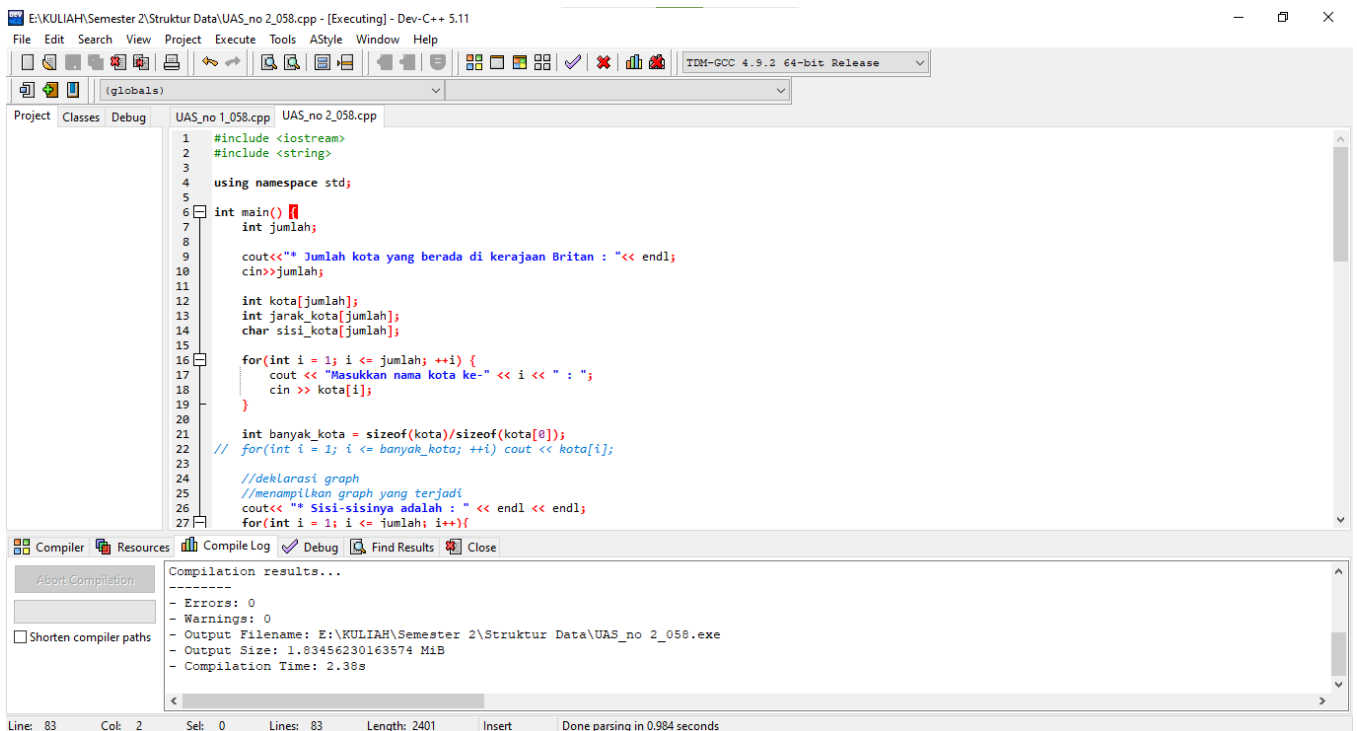
1. (x,y,w) dipisahkan dengan spasi

x = vertex 1, y = vertex 2, w = weight

2. Kota mana yang merupakan kota yang ditempati pedagang sekarang

3. Vertex mana yang merupakan kota yang diserang naga

4. Vertex mana yang merupakan kota tempat istana raja



The screenshot shows a C++ IDE with the following code in `UAS_no 2_058.cpp`:

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main()
7 {
8     int jumlah;
9
10    cout<<"* Jumlah kota yang berada di kerajaan Britan : "<< endl;
11    cin>>jumlah;
12
13    int kota[jumlah];
14    int jarak_kota[jumlah];
15    char sisi_kota[jumlah];
16
17    for(int i = 1; i <= jumlah; ++i) {
18        cout << "Masukkan nama kota ke-" << i << " : ";
19        cin >> kota[i];
20    }
21
22    int banyak_kota = sizeof(kota)/sizeof(kota[0]);
23    // for(int i = 1; i <= banyak_kota; ++i) cout << kota[i];
24
25    //deklarasi graph
26    //menampilkan graph yang terjadi
27    cout<<"* Sisi-sisinya adalah : " << endl << endl;
28    for(int i = 1; i <= jumlah; ++i){
```

The bottom panel shows the compilation results:

```
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: E:\KULIAH\Semester 2\Struktur Data\UAS_no 2_058.exe
- Output Size: 1.83456230163574 MIB
- Compilation Time: 2.38s
```

At the bottom, the status bar indicates: Line: 83, Col: 2, Sel: 0, Lines: 83, Length: 2401, Insert, Done parsing in 0.984 seconds.

E:\KULIAH\Semester 2\Struktur Data\UAS_no 2_058.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Classes Debug UAS_no 1_058.cpp UAS_no 2_058.cpp

```
24 //deklarasi graph
25 //menampilkan graph yang terjadi
26 cout<< "Sisi-sisinya adalah : " << endl << endl;
27 for(int i = 1; i <= jumlah; i++){
28     for(int j = 1; j <= jumlah; j++){
29         std::cout<< "(" << kota[i] << ", " << kota[j] << ")" << endl;
30         sisi_kota[i] = j;
31     }
32 }
33
34 int banyak_sisi = sizeof(sisi_kota)/sizeof(sisi_kota[0]);
35 cout<< endl << endl << "SISI KOTA ";
36 for(int i = 1; i <= banyak_sisi; ++i) cout << sisi_kota[i] << endl;
37
38 cout << endl << "Panjang jalan antar kota : " << endl;
39 cout<< "seluruh jalan yang ada dalam kerajaan britan dan panjang jalannya : " << endl;
40 for(int i = 1; i <= jumlah; i++){
41     for(int j = 1; j <= jumlah; j++){
42         std::cout<< "Panjang " << "(" << kota[i] << ", " << kota[j] << ") : " << endl;
43         // cin >> jarak_kota[k];
44         cin >> jarak_kota[i];
45         cout<< "(" << kota[i] << ", " << kota[j] << ") : " << jarak_kota[i] << endl;
46     }
47 }
48 // cout<<"panjang "<<kota1<<" ke "<<kota2<<" : "; cin>> hasil;
49
50
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Shorten compiler paths

Compilation results...

```
- Errors: 0
- Warnings: 0
- Output Filename: E:\KULIAH\Semester 2\Struktur Data\UAS_no 2_058.exe
- Output Size: 1.83456230163574 MiB
- Compilation Time: 2.38s
```

Line: 83 Col: 2 Sel: 0 Lines: 83 Length: 2401 Insert Done parsing in 0.984 seconds

E:\KULIAH\Semester 2\Struktur Data\UAS_no 2_058.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Classes Debug UAS_no 1_058.cpp UAS_no 2_058.cpp

```
51 //menampilkan tempat pedagang berada
52 cout<<" kota tempat pedagang sekarang berada : " << endl << endl;
53 cout<<kota[1];
54
55 cout<<endl<<endl;
56
57 //menampilkan kota yang diserang naga
58 cout<<" kota yang diserang naga : " << endl << endl;
59 cout<<kota[2];
60
61 cout<<endl<<endl;
62
63 //menampilkan kota yang terdapat kastil
64 cout<<" kota yang memiliki kastil : " << endl << endl;
65 int kota_terakhir = sizeof(kota) / sizeof(kota[0]);
66 cout<<kota_terakhir;
67
68 cout<<endl<<endl;
69
70 //menampilkan vertex tercepat untuk selamat
71 cout<<" jalur yang paling cepat ditempuh : " << endl << endl;
72 cout<<kota[1]<<"->"<<kota[2]<<"->"<<kota[3]<<endl;
73
74 cout<<endl<<endl;
75
76 //total edge yang harus ditempuh
77 cout<< " dengan jarak : " << endl << endl;
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Shorten compiler paths

Compilation results...

```
- Errors: 0
- Warnings: 0
- Output Filename: E:\KULIAH\Semester 2\Struktur Data\UAS_no 2_058.exe
- Output Size: 1.83456230163574 MiB
- Compilation Time: 2.38s
```

Line: 83 Col: 2 Sel: 0 Lines: 83 Length: 2401 Insert Done parsing in 0.984 seconds

```
58     cout<<"* kota yang diserang naga : "<<endl<<endl;
59     cout<<kota[2];
60
61     cout<<endl<<endl;
62
63     //menampilkan kota yang terdapat kastil
64     cout<<"* kota yang memiliki kastil : "<<endl<<endl;
65     int kota_terakhir = sizeof(kota) / sizeof(kota[0]);
66     cout<<kota_terakhir;
67
68     cout<<endl<<endl;
69
70     //menampilkan vertex tercepat untuk selamat
71     cout<<"* jalur yang paling cepat ditempuh : "<<endl<<endl;
72     cout<<kota[1]<<"- "<<kota[2]<<"- "<<kota[3]<<endl;
73
74     cout<<endl<<endl;
75
76     //total edge yang harus ditempuh
77     cout<<"* dengan jarak : "<<endl<<endl;
78     int a = sizeof(jarak_kota) / sizeof(jarak_kota[0]);
79     int jarakkota_terakhir = jarak_kota[a-1];
80     cout<<jarak_kota[1]*jarakkota_terakhir<<endl<<endl;
81
82     return 0;
83
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: E:\KULIAH\Semester 2\Struktur Data\UAS_no 2_058.exe
- Output Size: 1.83456230163574 MiB
- Compilation Time: 2.38s

Line: 83 Col: 2 Sek: 0 Lines: 83 Length: 2401 Insert Done parsing in 0.984 seconds

Output :

1. Jalur yang paling cepat ditempuh oleh pedagang untuk ke kastil tanpa melewati kota yang diserang naga
2. Jarak yang ditempuh

```
E:\KULIAH\Semester 2\Struktur Data\UAS_no 2_058.exe
* Jumlah kota yang berada di kerajaan Britan :
3
Masukkan nama kota ke-1 : 1
Masukkan nama kota ke-2 : 2
Masukkan nama kota ke-3 : 3
* Sisi-sisinya adalah :

(1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3)

SISI KOTA @
@
v

* Panjang jalan antar kota :
* seluruh jalan yang ada dalam kerajaan britan dan panjang jalannya :
Panjang (1,1) :
1
(1,1,1) Panjang (1,2) :
2
(1,2,2) Panjang (1,3) :
3
(1,3,3) Panjang (2,1) :
4
(2,1,4) Panjang (2,2) :
5
(2,2,5) Panjang (2,3) :
6
(2,3,6) Panjang (3,1) :
7
(3,1,7) Panjang (3,2) :
```

```
E:\KULIAH\Semester 2\Struktur Data\UAS_no 2_058.exe
(3,1,7) Panjang (3,2) :
8
(3,2,8) Panjang (3,3) :
9
(3,3,9) * kota tempat pedagang sekarang berada :
1
* kota yang diserang naga :
2
* kota yang memiliki kastil :
3
* jalur yang paling cepat ditempuh :
1-2-3
* dengan jarak :
9

-----
Process exited after 11.38 seconds with return value 0
Press any key to continue . . .
```