**Metode Penyebaran Kuesioner:**

1. Sebarkan URL beserta kode unik yang valid hingga 3 jam berikutnya. Jika kuota responden terpenuhi, maka kuesioner ditutup.
2. Jika responden pertama sedang mengerjakan, kemudian ada responden lain yang akan mengerjakan dengan kuota responden yang dibutuhkan maka responden yang lebih dahulu submit datanya akan terekam.
3. Kuota responden yang dibutuhkan adalah 7 orang.

**Metode Pengerjaan Survei:**

1. Responden menjawab soal pretest. Tujuannya adalah untuk mengetahui pengetahuan yang dimiliki oleh responden sebelumnya. Hal ini akan berkorelasi dengan seberapa efektif visualisasi akan membantu responden dalam memahami kode program.

2. Simulasi terbagi menjadi dua bagian, yaitu latihan tes dan tes inti. Tujuannya adalah agar responden dapat adaptasi menggunakan kakas OPT maupun CodeViz dengan fitur-fitur yang tersedia. Kemudian responden mengisi tes inti yang berkaitan dengan graf kode program. Sedangkan latihan tes berisi soal kode program sederhana yang memiliki aspek penilaian hampir sama dengan tes inti namun tingkat kesulitan lebih rendah dibanding soal pada tes inti. Untuk soal pada tes ini akan digunakan metode analisis sebagai berikut.

|  |  |  |
| --- | --- | --- |
| **Total 7 responden** | **7 responden** | **7 responden** |
| **Latihan**  4 soal | Latihan Simulasi OPT -> CodeViz | |
| **Skenario**  Jumlah soal:  5 OPT, 5 CodeViz  5 Post-Tes  x 2 skenario  Total = 30 soal. | * OPT -> CodeViz * Post-Tes | * CodeViz -> OPT * Post-Tes |
| **Teknik Analisis**  Berdasarkan correctness dan timing | Jika hasil survei **OPT lebih rendah** dibanding hasil survei CodeViz (makin meningkat), maka **visualisasi efektif** | Jika hasil survei **CodeViz lebih tinggi** dibanding hasil survei OPT (makin menurun), maka **visualisasi efektif** |
| Jika hasil survei **OPT lebih tinggi** dibanding hasil survei CodeViz (menurun), maka **visualisasi tidak efektif** | Jika hasil survei **CodeViz lebih rendah** dibanding hasil survei OPT (meningkat), maka **visualisasi tidak efektif** |

\* menghindari *memory effect ?*

**Reward:**

|  |  |
| --- | --- |
| Jumlah Jawaban Yang Benar dari total soal | Pulsa Yang akan Diperoleh + Presisi jumlah soal sbg poin domain |
| 25 - 30 | 25K |
| 20 - 24 | 20K |
| 15 - 19 | 15K |
| 10 - 14 | 10K |
| 1 - 9 | 5K |
| 0 | Tidak dapat pulsa – Tidak ada poin |

3. Post-tes berisi soal terkait graf kode program. Responden diberikan soal terkait teori graf dan beberapa potongan graf kode program tanpa perlu menggunakan kedua kakas tersebut.

**Aspek penilaian:**

Berkaitan dengan tujuan visualisasi yaitu **‘memahami’** terdapat empat poin:

1. Pengguna mampu menelusuri dan menjelaskan alur eksekusi graf kode program

2. Pengguna mampu memperbaiki kesalahan yang ada pada graf kode program

3. Pengguna mampu menjelaskan algoritma yang digunakan di dalam graf kode program

4. Pengguna mampu menjelaskan setiap prosedur, operasi, langkah, dan fungsi yang terdapat di dalam graf kode program

**Kuantitatif:**

- Correctness

- Timing

# Bagian I - Pretest

1. Apa yang dimaksud dengan graf ?

a. Graf adalah objek dasar pelajaran dalam teori graf.

b. Graf adalah himpunan dari objek-objek yang dinamakan titik, simpul, atau sudut dihubungkan oleh penghubung yang dinamakan garis atau sisi.

c. Graf adalah gambar kurva (garis atau sisi) yang membentuk bangun datar berupa persegi panjang.

d. Graf adalah gambar grafik untuk kepentingan visualisasi data.

e. Graf adalah objek dasar yang terdiri dari bangun datar dan garis.

1. Apa yang dimaksud dengan simpul (node) pada graf ?

a. Simpul (node) adalah graf ganda

b. Simpul (node) adalah graf tidak sederhana

c. Simpul (node) adalah graf semu

d. Simpul (node) adalah sisi yang menghubungkan pasangan titik yang sama

e. Simpul (node) adalah himpunan titik

1. Apa yang dimaksud dengan sisi (edge) pada graf ?

a. Sisi (edge) adalah bagian penting graf

b. Sisi (edge) adalah himpunan graf

c. Sisi (edge) adalah jembatan

d. Sisi (edge) adalah sisi yang membentuk persegi

e. Sisi (edge) adalah garis yang menghubungkan dua simpul atau lebih

1. Apa yang dimaksud dengan derajat (degree) pada graf ?

a. Derajat (degree) adalah ukuran (size) graf

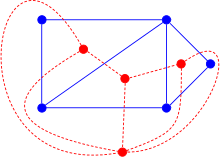
b. Derajat (degree) adalah bentuk sudut dari beberapa sisi

c. Derajat (degree) adalah banyaknya sisi yang bersisian dengan suatu simpul

d. Derajat (degree) adalah banyaknya simpul yang terhubung

e. Derajat (degree) adalah sisi genap dan sisi ganjil

1. Ada berapa simpul pada gambar graf berikut ini ?



a. 5

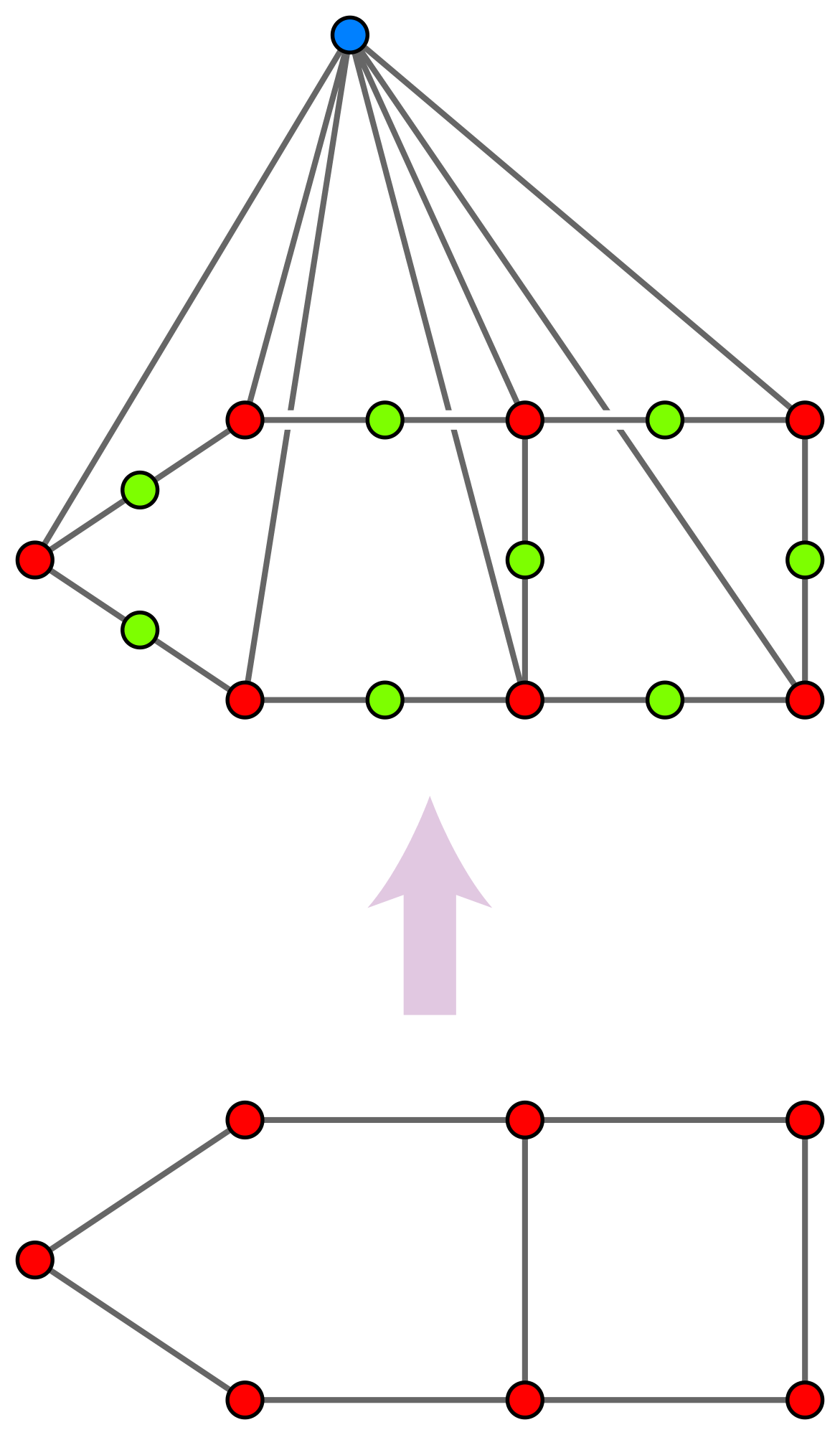
b. 7

c. 8

d. 9

e. 10

1. Ada berapa sisi pada gambar graf berikut ini ?

a. 6

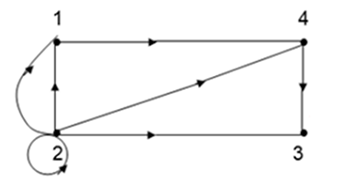
b. 7

c. 8

d. 9

e. 10

1. Berapa derajat pada simpul nomor 2 yang dimiliki oleh graf berikut ?

a. 4

b. 5

c. 6

d. 7

e. 8

1. Apakah Anda familiar dengan bahasa pemrograman C ? Ya/tidak

Jika jawab Ya:

8.1 Apakah Anda pernah memprogram graf dengan bahasa pemrograman C ? Ya Pernah/Tidak Pernah/Lupa, tidak tahu

8.2 Kapan Anda kali terkahir menggunakan bahasa pemrograman C ? hari ini, kemarin, 2 hari lalu, lebih dari 3 hari lalu, Tidak tahu/lupa

**Latihan | *Warming Up:***

**[Skenario 1 - OPT]** Berisi empat soal kode program sederhana dengan bahasa pemrograman C menggunakan OPT.

1. Dari kode program berikut, ketika variabel x di print bernilai ...

a. 41 b. 10 c. 0 d. 31 e. 62

|  |
| --- |
| Int main(){  int x = 10;  int y = 31;  int z = 0;  x = y;  y = x;  x = x + y;  printf(‘x = %d’, x);  return 0;  } |

2. Silakan perbaiki kesalahan sintaks pada kode program berikut ini!

|  |
| --- |
| Int main(){  int graf = [7][7];  int graf[7][8] = 0;  printf(‘graf’, graf);  return 0;  } |

3. Apa nama algoritma yang paling cocok untuk kode program berikut ini ?

a. menghitung luas b. Menghitung luas segitiga

c. menghitung luas persegi panjang d. Menghitung luas lingkaran

e. tidak dapat ditentukan

|  |
| --- |
| int main(){  int p = 7;  int l = 9;  int L = p \* l;  return 0;  } |

4. Fungsi X berguna untuk ...

a. menghitung perkalian b. Menghitung luas lingkaran

c. menghitung luas persegi d. Menghitung nilai phi

e. tidak berfungsi / error

|  |
| --- |
| int x(int r){  return 3.14\*r\*r;  }  int main(){  int L;  L = x(7);  printf(‘L = %d’, L);  return 0;  } |

**[Skenario 2 - CodeViz]** Berisi empat soal kode program sederhana dengan bahasa pemrograman C menggunakan CodeViz.

1 Silakan perbaiki kesalahan sintaks pada kode program berikut ini!

Output Program: z : 493

|  |
| --- |
| Int main(){  int x = 17;  int y = 29;  int z = y \* x;  printf(z);  return 0;  } |

2 Prosedur xyz berguna untuk ...

* 1. Mencetak hasil d. Operasi print
  2. Operasi perkalian m dan n, kemudian mencetaknya e. Tidak ada hasil
  3. Operasi penjumlahan

|  |
| --- |
| void xyz(int a, int b){  int z = a\*b;  printf("z = %d", z);  }  int main(){  int m = 33;  int n = 99;  xyz(m,n);  return 0;  } |

3 Dari kode program berikut ini, berapa nilai M[2][3] pada baris ke-4 ?

a. 29 b. 0 c. 1 d. 5 e. null

|  |
| --- |
| int main(){  int M[5][5] = {0};  int y = 29;  int z = y \* M[2][3];  return 0;  } |

4 Apa kegunaan kode program berikut ini ?

* + - * 1. Menghitung nilai array d. Menghitung panjang array
        2. Menghitung luas persegi e. Tidak dapat didefinisikan
        3. Menentukan nilai ganjil atau genap

|  |
| --- |
| int main(){  int x[5], y, i;  x[0] = 9;  x[1] = 11;  x[2] = 17;  x[3] = 19;  x[4] = 20;  for(i = 0; i < 5; i++){  y = x[i] % 2;  if (y == 0) printf("%d adalah ganjil\n", x[i]);  else printf("%d adalah genap\n", x[i]);  }  return 0;  } |

# Bagian II - Simulasi

**Tes Inti | *Core Test:***

**[Skenario 1]**

**[Sesi 1]** Berisi empat buah soal graf kode program dengan bahasa pemrograman C menggunakan OPT.

Tips: silakan menggunakan alat tulis untuk membantu dalam penyelesaian soal.

1 Berapa bobot antara simpul 7 dan 1 dalam graf kode program berikut ini ?

* 1. 3
  2. 4
  3. 0
  4. 1
  5. Tidak ada

|  |
| --- |
| #include<stdio.h>  int main()  {  int row = 4, col = 4;  int i, j,  cost[4][4] = { { 0, 1, 1, 1},  { 1, 0, 1, 1},  { 1, 1, 0, 1},  { 1, 1, 1, 0}  };    printf("\nthe matrix:\n");  for (i = 0; i < row; i++)  for (j = 0; j < col; j++)  printf("cost[%d][%d]: %d\n",i,j,cost[i][j]);  return 0;  } |

2 Silakan perbaiki kesalahan pada graf kode program berikut ini agar dapat *running* secara normal.

|  |
| --- |
| #include<stdio.h>  int row = 4, col = 4;  int i, j,  cost[4][4];  int main()  {  printf("\nthe matrix:\n")  cost[0][1] = 2  cost[0][2] = 3;  cost[0][3] = 1;  cost[1][2] = 7;  cost[1][3] = 9;  cost[2][3] = 8;    for (i = 0; i < row; i++)  for (j = 0; j <= col; j++)  printf("cost[%d][%d]: %d\n",i,j,cost[i][j]);  return 0  } |

3 Apa nama algoritma yang paling cocok digunakan untuk menggambarkan proses eksekusi graf kode program berikut ini ?

* + - * 1. Travelling Salesman Problem (TSP) c. Binary Search Tree (BST) e. Tidak diketahui
        2. Bellman Ford d. Breadth First Search (BFS)

|  |
| --- |
| #include <stdio.h>  int ary[10][10], completed[10], n, cost = 0;  void takeInput()  {  int i, j;  printf("the number of villages: ");  n = 4;  printf("\nthe Cost Matrix\n");  for (i = 0; i < n; i++)  {  printf("\nElements of Row: %d\n", i + 1);  ary[i][0] = 3;  ary[i][1] = 7;  ary[i][2] = 4;  ary[i][3] = 9;  completed[i] = 0;  }  printf("\n\nThe cost list is:");  for (i = 0; i < n; i++)  {  printf("\n");  for (j = 0; j < n; j++)  printf("\t%d", ary[i][j]);  }  }  void mincost(int city)  {  int i, ncity;  completed[city] = 1;  printf("%d--->", city + 1);  ncity = least(city);  if (ncity == 999)  {  ncity = 0;  printf("%d", ncity + 1);  cost += ary[city][ncity];  return;  }  mincost(ncity);  }  int least(int c)  {  int i, nc = 999;  int min = 999, kmin;  for (i = 0; i < n; i++)  {  if ((ary[c][i] != 0) && (completed[i] == 0))  if (ary[c][i] + ary[i][c] < min)  {  min = ary[i][0] + ary[c][i];  kmin = ary[c][i];  nc = i;  }  }  if (min != 999)  cost += kmin;  return nc;  }  int main()  {  takeInput();  printf("\n\nThe Path is:\n");  mincost(0);  printf("\n\nMinimum cost is %d\n ", cost);  return 0;  } |

4 Apa kegunaan fungsi XYZ dibuat pada graf kode program berikut ini?

a. menghitung derajat simpul c. Menghitung simpul yang berdekatan e. Tidak dapat ditentukan

b. menghitung jumlah sisi d. Menghitung jarak antar simpul

|  |
| --- |
| #include <stdio.h>  #include <time.h>  #define infinity 999  int graf[10][10], X[10], n, c = 0;  int R(const int nMin, const int nMax)  {  return rand() % (nMax - nMin) + nMin;  }  void Z()  {  int i, j;  n = R(3, 9);  for (i = 0; i < n; i++)  {  for (j = 0; j < n; j++)  {  graf[i][j] = R(0, 17);  if (graf[i][j] == 0)  {  graf[i][j] = infinity;  graf[j][i] = infinity;  }  else  {  graf[j][i] = graf[i][j];  }  }  X[i] = 0;  }  for (i = 0; i < n; i++)  {  printf("\n");  for (j = 0; j < n; j++)  printf("\t%d", graf[i][j]);  }  }  void fungsiku(int s)  {  int i, n;  X[s] = 1;  printf("%d--->", s + 1);  n = xyz(s);  if (n == 999)  {  n = 0;  printf("%d", n + 1);  c += graf[s][n];  return;  }  fungsiku(n);  }  int xyz(int c)  {  int i, nc = 999;  int min = 999, kmin;  for (i = 0; i < n; i++)  {  if ((graf[c][i] != 0) && (X[i] == 0))  if (graf[c][i] + graf[i][c] < min)  {  min = graf[i][0] + graf[c][i];  kmin = graf[c][i];  nc = i;  }  }  if (min != 999)  c += kmin;  return nc;  }  int main()  {  srand(time(NULL));  Z();  printf("\n");  fungsiku(0);  printf("\n\nMin: %d\n ", c);  return 0;  } |

5. Berapa jumlah simpul dalam graf kode program berikut ini?

a. 3 b. 7 c. 4 d. 9 e. 10

|  |
| --- |
| #include <stdio.h>  int G[10][10], completed[10], n, cost = 0;  void takeInput()  {  int i, j;  printf("the number of villages: ");  n = 4;  printf("\nthe Cost Matrix\n");  for (i = 0; i < n; i++)  {  printf("\nElements of Row: %d\n", i + 1);  G[i][0] = 3;  G[i][1] = 7;  G[i][2] = 4;  G[i][3] = 9;  completed[i] = 0;  }  printf("\n\nThe cost list is:");  for (i = 0; i < n; i++)  {  printf("\n");  for (j = 0; j < n; j++)  printf("\t%d", G[i][j]);  }  }  void mincost(int city)  {  int i, ncity;  completed[city] = 1;  printf("%d--->", city + 1);  ncity = least(city);  if (ncity == 999)  {  ncity = 0;  printf("%d", ncity + 1);  cost += G[city][ncity];  return;  }  mincost(ncity);  }  int least(int c)  {  int i, nc = 999;  int min = 999, kmin;  for (i = 0; i < n; i++)  {  if ((G[c][i] != 0) && (completed[i] == 0))  if (G[c][i] + G[i][c] < min)  {  min = G[i][0] + G[c][i];  kmin = G[c][i];  nc = i;  }  }  if (min != 999)  cost += kmin;  return nc;  }  int main()  {  takeInput();  printf("\n\nThe Path is:\n");  mincost(0);  printf("\n\nMinimum cost is %d\n ", cost);  return 0;  } |

**[Sesi 2]** Berisi empat buah soal graf kode program dengan bahasa pemrograman C menggunakan CodeViz.

1 Apa nama algoritma yang paling cocok digunakan untuk menggambarkan proses eksekusi graf kode program berikut?

a. Bellman Ford b. BST c. BFS d. TSP e. Tidak diketahui

|  |
| --- |
| #include<stdio.h>  int row = 4, col = 4;  int i, j,  graf[4][4] = { { 0, 2, 3, 1},  { 2, 0, 7, 9},  { 3, 7, 0, 8},  { 1, 9, 8, 0}  };  int main()  {  printf("\nGraf:\n");  for (i = 0; i < row; i++)  for (j = 0; j < col; j++)  printf("graf[%d][%d]: %d\n",i,j,graf[i][j]);  return 0;  } |

2 Apa tujuan fungsi M dibuat dalam graf kode program ini ?

a. menghitung jarak antar simpul c. Menghitung angka random e. Tidak diketahui

b. menghitung jarak antar sisi d. Menghitung angka ganjil / genap

|  |
| --- |
| #include <stdio.h>  #include <time.h>  #define infinity 999  int G[10][10], X[10], n, c = 0;  void Z()  {  int i, j;  n = M(1, 7);  for (i = 0; i < n; i++)  {  for (j = 0; j < n; j++)  {  G[i][j] = M(0, 9);  if (G[i][j] == 0)  {  G[i][j] = infinity;  G[j][i] = infinity;  }  else  {  G[j][i] = G[i][j];  }  }  X[i] = 0;  }  for (i = 0; i < n; i++)  {  printf("\n");  for (j = 0; j < n; j++)  printf("\t%d", G[i][j]);  }  }  void F(int s)  {  int i, n;  X[s] = 1;  printf("%d--->", s + 1);  n = xyz(s);  if (n == 999)  {  n = 0;  printf("%d", n + 1);  c += G[s][n];  return;  }  F(n);  }  int M(const int x, const int y){  return rand() % (y - x) + x;  }  int xyz(int c)  {  int i, nc = 999;  int min = 999, kmin;  for (i = 0; i < n; i++)  {  if ((G[c][i] != 0) && (X[i] == 0))  if (G[c][i] + G[i][c] < min)  {  min = G[i][0] + G[c][i];  kmin = G[c][i];  nc = i;  }  }  if (min != 999)  c += kmin;  return nc;  }  int main()  {  srand(time(NULL));  Z();  printf("\n");  F(0);  printf("\nMinimun cost: %d\n ", c);  return 0;  } |

3 Berapa jumlah simpul yang berderajat 3 dalam graf kode program berikut ini ?

a. 3 b. 4 c. 5 d. 6 e. 7

|  |
| --- |
| #include<stdio.h>  int main()  {  int row = 4, col = 4;  int i, j,  cost[4][4] = { { 0, 2, 3, 1},  { 2, 0, 7, 9},  { 3, 7, 0, 8},  { 1, 9, 8, 0}  };  printf("\nthe matrix:\n");  for (i = 0; i < row; i++)  for (j = 0; j < col; j++)  printf("cost[%d][%d]: %d\n",i,j,cost[i][j]);  return 0;  } |

4 Ubahlah nilai bobot antara simpul 1 dan 3 menjadi 9 dari graf kode program berikut ini !

|  |
| --- |
| #include <stdio.h>  #include <time.h>  #define infinity 999  int G[10][10], X[10], n, c = 0;  void Z()  {  int i, j;  n = M(4, 7);  for (i = 0; i < n; i++)  {  for (j = 0; j < n; j++)  {  G[i][j] = M(0, 9);  if (G[i][j] == 0)  {  G[i][j] = infinity;  G[j][i] = infinity;  }  else  {  G[j][i] = G[i][j];  }  }  X[i] = 0;  }  G[1][3] = 9;  G[3][1] = 9;  for (i = 0; i < n; i++)  {  printf("\n");  for (j = 0; j < n; j++)  printf("\t%d", G[i][j]);  }  }  void F(int s)  {  int i, n;  X[s] = 1;  printf("%d--->", s + 1);  n = xyz(s);  if (n == 999)  {  n = 0;  printf("%d", n + 1);  c += G[s][n];  return;  }  F(n);  }  int M(const int x, const int y){  return rand() % (y - x) + x;  }  int xyz(int c)  {  int i, nc = 999;  int min = 999, kmin;  for (i = 0; i < n; i++)  {  if ((G[c][i] != 0) && (X[i] == 0))  if (G[c][i] + G[i][c] < min)  {  min = G[i][0] + G[c][i];  kmin = G[c][i];  nc = i;  }  }  if (min != 999)  c += kmin;  return nc;  }  int main()  {  srand(time(NULL));  Z();  printf("\n");  F(0);  printf("\nMinimun cost: %d\n ", c);  return 0;  } |

5. Berapa bobot minimum antara simpul 1 dan 4 pada graf kode program berikut ini ?

a. 7 b. 8 c. 9 d. 10 e. 11

|  |
| --- |
| #include<stdio.h>  int baris = 5, kolom = 5;  int i, j,  cost[5][5];  int main()  {  printf("\nthe matrix:\n");  cost[0][1] = 2;  cost[0][2] = 3;  cost[0][3] = 1;  cost[1][2] = 7;  cost[1][3] = 9;  cost[2][3] = 8;  cost[2][4] = 4;    for (i = 0; i < baris; i++)  for (j = 0; j < kolom; j++)  printf("cost[%d][%d]: %d\n",i,j,cost[i][j]);  return 0;  } |

# Bagian III - Post-Test Skenario 1:

Tidak diperkenankan menggunakan bantuan kakas apapun.

Silakan persiapkan alat tulis untuk membantu dalam menghitung.

**Graf Kode Program:**

1. Berapa total simpul dalam graf kode program di bawah ini ?

a. 7 b. 1 c. Antara 1 dan 7 d. Antara 0 dan 10 e. Tidak dapat ditentukan

|  |
| --- |
| #include <stdio.h>  #include <time.h>  #define infinity 999  int G[10][10], X[10], n, c = 0;  void Z()  {  int i, j;  n = M(1, 7);  for (i = 0; i < n; i++)  {  for (j = 0; j < n; j++)  {  G[i][j] = M(0, 9);  if (G[i][j] == 0)  {  G[i][j] = infinity;  G[j][i] = infinity;  }  else  {  G[j][i] = G[i][j];  }  }  X[i] = 0;  }  for (i = 0; i < n; i++)  {  printf("\n");  for (j = 0; j < n; j++)  printf("\t%d", G[i][j]);  }  }  void F(int s)  {  int i, n;  X[s] = 1;  printf("%d--->", s + 1);  n = xyz(s);  if (n == 999)  {  n = 0;  printf("%d", n + 1);  c += G[s][n];  return;  }  F(n);  }  int M(const int x, const int y){  return rand() % (y - x) + x;  }  int xyz(int c)  {  int i, nc = 999;  int min = 999, kmin;  for (i = 0; i < n; i++)  {  if ((G[c][i] != 0) && (X[i] == 0))  if (G[c][i] + G[i][c] < min)  {  min = G[i][0] + G[c][i];  kmin = G[c][i];  nc = i;  }  }  if (min != 999)  c += kmin;  return nc;  }  int main()  {  srand(time(NULL));  Z();  printf("\n");  F(0);  printf("\nMinimun cost: %d\n ", c);  return 0;  } |

2 Berapa derajat (degree) yang dimiliki oleh simpul 3 pada graf kode program berikut ini ?

a. 1 b. 2 c. 3 d. 4 e. Tidak diketahui

|  |
| --- |
| #include<stdio.h>  int main()  {  int row = 5, col = 5;  int i, j,  G[5][5] = { { 0, 6, 3, 1, 3},  { 6, 0, 9, 9, 9},  { 3, 9, 0, 8, 8},  { 1, 9, 8, 0, 7},  { 1, 9, 8, 7, 0}  };  printf("\nthe matrix:\n");  for (i = 0; i < row; i++)  for (j = 0; j < col; j++)  printf("G[%d][%d]: %d\n",i,j,G[i][j]);  return 0;  } |

3. Berapa jumlah minimum biaya (cost) yang diperlukan dari simpul 1 hingga simpul 4 dari graf kode program berikut ini ?

a. 7 b. 9 c. 11 d. 10 e. 12

|  |
| --- |
| #include<stdio.h>  int baris = 5, kolom = 5;  int i, j,  cost[5][5];  int main()  {  printf("\nthe matrix:\n")  cost[0][1] = 2  cost[0][2] = 3;  cost[0][3] = 1;  cost[1][2] = 7;  cost[1][3] = 9;  cost[2][3] = 8;  cost[2][4] = 4;  cost[2][1] = 7;    for (i = 0; i < row; i++)  for (j = 0; j <= col; j++)  printf("cost[%d][%d]: %d\n",i,j,cost[i][j]);  return 0  } |

4. Berapa jumlah sisi (edge) yang dimiliki oleh graf kode program berikut ini ?

a. 1 b. 2 c. 3 d. 4 e. 5

|  |
| --- |
| #include<stdio.h>  int row = 3, col = 3;  int i, j,  G[3][3] = { { 0, 2, 3},  { 2, 0, 7},  { 3, 7, 0},  };  int main()  {  printf("\nGraf:\n");  for (i = 0; i < row; i++)  for (j = 0; j < col; j++)  printf("G[%d][%d]: %d\n",i,j,G[i][j]);  return 0;  } |

5. Ada berapa kesalahan sintaks pada kode program berikut ini ?

a. 5 b. 6 c. 7 d. 8 e. 9

Sebutkan pada baris berapa saja? ...

|  |
| --- |
| #include<stdio.h>  int baris = 5, kolom = 5;  int i, j,  cost[5][5];  int main()  {  printf("\nthe matrix:\n")  cost[0][1] = 2  cost[0][2] = 3;  cost[0][3] = 1;  cost[1][2] = 7;  cost[1][3] = 9;  cost[2][3] = 8;  cost[2][4] = 4;  cost[2][1] = 7;    for (i = 0; i < row; i++)  for (j = 0; j <= col; j++)  printf("cost[%d][%d]: %d\n",i,j,cost[i][j]);  return 0  } |

**[Skenario 2]**

**Sesi 1** Berisi empat buah soal graf kode program dengan bahasa pemrograman C menggunakan CodeViz.

Ubahlah bobot antara simpul 2 dan 4 menjadi nilai 7 pada graf kode program berikut ini.

|  |
| --- |
| #include<stdio.h>  int main()  {  int row = 4, col = 4;  int i, j,  cost[4][4] = { { 0, 1, 1, 1},  { 1, 0, 1, 1},  { 1, 1, 0, 1},  { 1, 1, 1, 0}  };    printf("\nthe matrix:\n");  for (i = 0; i < row; i++)  for (j = 0; j < col; j++)  printf("cost[%d][%d]: %d\n",i,j,cost[i][j]);  return 0;  } |

Jawaban:

|  |
| --- |
| #include<stdio.h>  int main()  {  int row = 5, col = 5;  int i, j,  cost[5][5] = { { 0, 1, 1, 1, 1},  { 1, 0, 1, 1, 1},  { 1, 1, 0, 1, 7},  { 1, 1, 1, 0, 1},  { 1, 1, 7, 1, 0}  };    printf("\nthe matrix:\n");  for (i = 0; i < row; i++)  for (j = 0; j < col; j++)  printf("cost[%d][%d]: %d\n",i,j,cost[i][j]);  return 0;  } |

Apa kegunaan prosedur Q pada graf kode program berikut ini?

a. menghitung jarak minimum antar simpul

b. prosedur rekursif dengan memanggil fungsi xyz kemudian mencetak hasil

c. prosedur pelengkap untuk mencetak hasil jarak minimum

d. untuk menghitung dengan fungsi xyz

e. tidak diketahui

|  |
| --- |
| #include <stdio.h>  #include <time.h>  #define infinity 999  int graf[10][10], X[10], n, c = 0;  int R(const int nMin, const int nMax)  {  return rand() % (nMax - nMin) + nMin;  }  void Z()  {  int i, j;  n = R(3, 9);  for (i = 0; i < n; i++)  {  for (j = 0; j < n; j++)  {  graf[i][j] = R(0, 17);  if (graf[i][j] == 0)  {  graf[i][j] = infinity;  graf[j][i] = infinity;  }  else  {  graf[j][i] = graf[i][j];  }  }  X[i] = 0;  }  for (i = 0; i < n; i++)  {  printf("\n");  for (j = 0; j < n; j++)  printf("\t%d", graf[i][j]);  }  }  void Q(int s)  {  int i, n;  X[s] = 1;  printf("%d--->", s + 1);  n = xyz(s);  if (n == 999)  {  n = 0;  printf("%d", n + 1);  c += graf[s][n];  return;  }  Q(n);  }  int xyz(int c)  {  int i, nc = 999;  int min = 999, kmin;  for (i = 0; i < n; i++)  {  if ((graf[c][i] != 0) && (X[i] == 0))  if (graf[c][i] + graf[i][c] < min)  {  min = graf[i][0] + graf[c][i];  kmin = graf[c][i];  nc = i;  }  }  if (min != 999)  c += kmin;  return nc;  }  int main()  {  srand(time(NULL));  Z();  printf("\n");  Q(0);  printf("\n\nHasil: %d\n ", c);  return 0;  } |

Berapa derajat (degree) yang dimiliki oleh simpul 3 dari graf kode program berikut ini ? Jika ada error, silakan perbaiki!

a. 1 b. 2 c. 3 d. 4 e. 5

|  |
| --- |
| #include<stdio.h>  int row = 5, col = 4;  int i, j,  graf[5][4] = { { 0, 2, 3, 1},  { 2, 0, 7, 9},  { 3, 7, 0, 8},  { 1, 9, 8, 0}  };  int main()  {  printf("\nGraf:\n");  for (i = 0; i < row i++)  for (j = 0; j < cols; j++)  printf("graf[%d][%d]: %d\n",i,j,graf[i][j]);  return 0;  } |

Apa nama algoritma yang paling cocok digunakan untuk menggambarkan proses eksekusi graf kode program berikut ?

a. Dijkstra b. Bellman Ford c. Breadth First Search (BFS) d. Tree e. Tidak diketahui

|  |
| --- |
| #include<stdio.h>  int baris = 5, kolom = 5;  int i, j,  graf[5][5];  int main()  {  printf("\nthe matrix:\n");  graf[0][1] = 2;  graf[0][2] = 3;  graf[0][3] = 1;  graf[1][2] = 7;  graf[1][3] = 9;  graf[2][3] = 8;  graf[2][4] = 4;    for (i = 0; i < baris; i++)  for (j = 0; j < kolom; j++)  printf("graf[%d][%d]: %d\n",i,j,graf[i][j]);  return 0;  } |

5. Berapa bobot minimum dari simpul 1 ke simpul 7 pada graf kode program berikut ini ?

a. 14 b. 13 c. 12 d. 11 e. Tidak dapat dihitung

|  |
| --- |
| #include <stdio.h>  #include <time.h>  #define infinity 999  int graf[10][10], X[10], n, c = 0;  int R(const int nMin, const int nMax)  {  return rand() % (nMax - nMin) + nMin;  }  void Z()  {  int i, j;  n = 7;  for (i = 0; i < n; i++)  {  X[i] = 0;  }    graf[1][2] = 3;  graf[1][3] = 5;  graf[2][4] = 10;  graf[3][5] = 4;  graf[5][6] = 1;  graf[6][7] = 3;  graf[2][7] = 11;    for (i = 0; i < n; i++)  {  printf("\n");  for (j = 0; j < n; j++)  printf("\t%d", graf[i][j]);  }  }  void Q(int s)  {  int i, n;  X[s] = 1;  printf("%d--->", s + 1);  n = xyz(s);  if (n == 999)  {  n = 0;  printf("%d", n + 1);  c += graf[s][n];  return;  }  Q(n);  }  int xyz(int c)  {  int i, nc = 999;  int min = 999, kmin;  for (i = 0; i < n; i++)  {  if ((graf[c][i] != 0) && (X[i] == 0))  if (graf[c][i] + graf[i][c] < min)  {  min = graf[i][0] + graf[c][i];  kmin = graf[c][i];  nc = i;  }  }  if (min != 999)  c += kmin;  return nc;  }  int main()  {  srand(time(NULL));  Z();  printf("\n");  Q(0);  printf("\n\nHasil: %d\n ", c);  return 0;  } |

**Sesi 2** Berisi lima buah soal graf kode program dengan bahasa pemrograman C menggunakan OPT.

1. Berapa jumlah sisi (edge) yang dimiliki oleh graf kode program berikut ini ?

a. 5 b. 6 c. 7 d. 8 e. 10

|  |
| --- |
| #include<stdio.h>  int main()  {  int baris = 5, kolom = 5;  int x, y,  graf[5][5] = { { 0, 1, 1, 1, 1},  { 1, 0, 1, 1, 1},  { 1, 1, 0, 1, 1},  { 1, 1, 1, 0, 1},  { 1, 1, 1, 1, 0}  };    printf("\nGraf:\n");  for (x = 0; x < baris; x++)  for (y = 0; y < kolom; y++)  printf("graf[%d][%d]: %d\n",x,y,graf[x][y]);  return 0;  } |

1. Ubahlah graf kode program berikut ini agar jumlah simpul menjadi 5 buah.

|  |
| --- |
| #include <stdio.h>  #include <time.h>  #define infinity 999  int graf[4][4], X[4], n, c = 0;  int R(const int M, const int N){  return rand() % (N - M) + M;  }  void G(){  int i, j;  n = R(2, 4);  for (i = 1; i < n; i++)  {  for (j = 1; j < n; j++)  {  graf[i][j] = R(0, 17);  if (graf[i][j] == 0){  graf[i][j] = infinity;  graf[j][i] = infinity;  } else {  graf[j][i] = graf[i][j];  }  }  X[i] = 0;  }  for (i = 1; i < n; i++){  printf("\n");  for (j = 1; j < n; j++)  printf("\t%d", graf[i][j]);  }  }  void F(int s){  int i, n;  X[s] = 1;  printf("%d--->", s);  n = minim(s);  if (n == 999){  n = 1;  printf("%d", n);  c += graf[s][n];  return;  }  F(n);  }  int minim(int c){  int i, nc = 999;  int min = 999, g\_min;  for (i = 1; i < n; i++){  if ((graf[c][i] != 0) && (X[i] == 0))  if (graf[c][i] + graf[i][c] < min){  min = graf[i][0] + graf[c][i];  g\_min = graf[c][i];  nc = i;  }  }  if (min != 999)  c += g\_min;  return nc;  }  int main(){  srand(time(NULL));  G();  printf("\n");  F(1);  printf("\n\nHasil: %d\n ", c);  return 0;  } |

Kunci Jawaban:

|  |
| --- |
| #include <stdio.h>  #include <time.h>  #define infinity 999  int graf[5][5], X[5], n, c = 0;  int R(const int M, const int N){  return rand() % (N - M) + M;  }  void G(){  int i, j;  n = 5;  for (i = 0; i < n; i++)  {  for (j = 0; j < n; j++)  {  graf[i][j] = R(0, 10);  if (graf[i][j] == 0){  graf[i][j] = infinity;  graf[j][i] = infinity;  } else {  graf[j][i] = graf[i][j];  }  }  X[i] = 0;  }  printf("\n\nBiaya perjalanan:");  for (i = 0; i < n; i++){  printf("\n");  for (j = 0; j < n; j++)  printf("\t%d", graf[i][j]);  }  }  void F(int s){  int i, n;  X[s] = 1;  printf("%d--->", s);  n = minim(s);  if (n == 999){  n = 1;  printf("%d", n);  c += graf[s][n];  return;  }  F(n);  }  int minim(int c){  int i, nc = 999;  int min = 999, g\_min;  for (i = 0; i < n; i++){  if ((graf[c][i] != 0) && (X[i] == 0))  if (graf[c][i] + graf[i][c] < min){  min = graf[i][0] + graf[c][i];  g\_min = graf[c][i];  nc = i;  }  }  if (min != 999)  c += g\_min;  return nc;  }  int main(){  srand(time(NULL));  G();  printf("\n\nJalur:\n");  F(0);  printf("\n\nMinimal Biaya: %d\n ", c);  return 0;  } |

1. Apakah tujuan prosedur X dibuat pada graf kode program berikut ini ?

a. untuk menghitung jumlah simpul (node)

b. untuk menghitung jarak antar sisi (edge)

c. untuk menghitung jarak antar simpul (node)

d. untuk menghitung bobot pada sisi

e. tidak diketahui

|  |
| --- |
| #include<stdio.h>  void X(int graf[4][4], int m, int n){  printf("\nHitung jarak antar simpul %d ke %d\n", m, n);  }  int main(){  int row = 4, col = 4;  int i, j,  graf[4][4] = { { 0, 2, 3, 1},  { 2, 0, 7, 9},  { 3, 7, 0, 8},  { 1, 9, 8, 0}  };  printf("\nGraf:\n");  for (i = 0; i < row; i++)  for (j = 0; j < col; j++){  printf("graf[%d][%d]: %d\n",i,j,graf[i][j]);  X(graf, i, j);  }  return 0;  } |

1. Apa nama algoritma yang paling sesuai pada graf kode program berikut ini ?

a. Depth First Search (DFS) c. Dijkstra e. Tidak diketahui

b. Bellman Ford d. Breadth First Search

|  |
| --- |
| #include<stdio.h>  #define infinity 999  void XYZ(int n, int v, int cost[7][7], int dist[10]){  int i, u, count, w, flag[10], min;  for (i = 1; i <= n; i++){  flag[i] = 0;  dist[i] = cost[v][i];  }    count = 2;    while (count <= n){  min = 99;  for (w = 1; w <= n; w++){  if (dist[w] < min && !flag[w]){  min = dist[w];  u = w;  }  }    flag[u] = 1;  count++;  for (w = 1; w <= n; w++){  if ((dist[u] + cost[u][w] < dist[w]) && !flag[w]){  dist[w] = dist[u] + cost[u][w];  }  }  }  }  int main()  {  int n, v, i, dist[10] = {0};    n = 6;  printf("\nthe number of nodes: %d \n", n);    printf("\nthe cost matrix:\n");  int cost[7][7] = { { 0, 0, 0, 0, 0, 0, 0},  { 0, infinity, 5, infinity, infinity, infinity, 8},  { 0, 5, infinity, 7, infinity, 2, infinity},  { 0, infinity, 7, infinity, 9, infinity, infinity},  { 0, infinity, infinity, 9, infinity, 4, infinity},  { 0, infinity, 2, infinity, 4, infinity, infinity},  { 0, 8, infinity, infinity, infinity, infinity, infinity}  };    v = 2;  printf("\nthe source matrix: %d \n", v);    XYZ(n, v, cost, dist);    printf("\nShortest path:\n");  for (i = 1; i <= n; i++){  if (i != v){  printf("%d->%d,cost=%d\n", v, i, dist[i]);  }  }  return 0;  } |

1. Perbaiki kesalahan sintaks pada graf kode program berikut ini agar dapat *running* secara normal!

|  |
| --- |
| #include<stdio.h>  #define infinity 999  void someAlgorithm(int n, int v, int cost[][], int dist[10]){  int i, u, count, w, F[10], min;  for (i = 1; i <= n; i++){  F[i] = 0;  dist[i] = cost[v][i];  }    count = 2;    while (count <= n){  min = 99;  for (w = 1; w <= n; w++){  if (dist[w] < min && !F[w]){  min = dist[w];  u = w;  }  }    F[u] = 1;  count++;  for (w = 1; w <= n; w++){  if ((dist[u] + cost[u][w] < dist[w]) && !F[w]){  dist[w] = dist[u] + cost[u][w];  }  }  }  }  int main()  {  int n, v, i, dist[10] = {0};    n = 6;  printf("\nthe number of nodes: %d \n", n);    printf("\nthe cost matrix:\n");  int graf[7][7] = { { 0, 0, 0, 0, 0, 0, 0}  { 0, infinity, 5, infinity, infinity, infinity, 8},  { 0, 5, infinity, 7, infinity, 2, infinity},  { 0, infinity, 7, infinit, 9, infinity, infinity},  { 0, infinity, infinity, 9, infinity, 4, infinity},  { 0, infinity, 2, infinity, 4, infinity, infinity},  { 0, 8, infinity, infinity, infinity, infinity, infinity}  };    v = 2;  printf("\nthe source matrix: %d \n", v);    someAlgorithm(n, v, graf, dist);    printf("\nShortest path:\n");  for (i = 1; i <= n; i++){  if (i != v){  printf("%d->%d,graf=%d\n", v, i, dist[i]);  }  }  return 0;  } |

# Bagian III - Post-Test Skenario 2:

Tidak diperkenankan menggunakan bantuan kakas apapun.

Silakan persiapkan alat tulis untuk membantu dalam menghitung.

**Graf Kode Program:**

1. Berapa derajat (degree) yang dimiliki oleh simpul 3 pada graf kode program berikut ini ?

a. 3 b. 4 c. 5 d. 6 e. 7

|  |
| --- |
| #include<stdio.h>  int baris = 5, kolom = 5;  int i, j,  cost[5][5];  int main()  {  printf("\nthe matrix:\n");  cost[0][1] = 2;  cost[0][2] = 3;  cost[0][3] = 1;  cost[1][2] = 7;  cost[1][3] = 9;  cost[2][3] = 8;  cost[2][4] = 4;    for (i = 0; i < baris; i++)  for (j = 0; j < kolom; j++)  printf("cost[%d][%d]: %d\n",i,j,cost[i][j]);  return 0;  } |

1. Apakah tujuan fungsi Z dibuat pada graf kode program berikut ini ?

a. untuk menghitung nilai random

b. untuk menghitung nilai bobot pada sisi graf

c. untuk menghitung jarak antar simpul (node)

d. untuk menghitung sisi simpul dari dan ke simpul lain

e. tidak diketahui

|  |
| --- |
| #include<stdio.h>  int Z(int graf[4][4], int m, int n){  printf("\nHitung sisi simpul %d ke %d\n", m, n);  return graf[n][m];  }  int main(){  int row = 4, col = 4, sisi;  int i, j,  graf[4][4] = { { 0, 2, 3, 1},  { 2, 0, 7, 9},  { 3, 7, 0, 8},  { 1, 9, 8, 0}  };  printf("\nGraf:\n");  for (i = 0; i < row; i++)  for (j = 0; j < col; j++){  printf("graf[%d][%d]: %d\n",i,j,graf[i][j]);  sisi = Z(graf, i, j);  }  return 0;  } |

1. Berapa jumlah sisi (edge) yang dimiliki oleh graf kode program berikut ini jika tanpa simpul 5 ?

a. 1 b. 6 c. 5 d. 7 e. 0

|  |
| --- |
| #include<stdio.h>  int main(){  int x = 6, y = 6;  int i, j,  graf[6][6] = { { 0, 0, 0, 0, 0, 0},  { 0, 0, 11, 7, 5, 9},  { 0, 11, 0, 8, 4, 13},  { 0, 7, 8, 0, 2, 1},  { 0, 5, 3, 2, 0, 13},  { 0, 9, 3, 1, 13, 0}  };  printf("\nGraf:\n");  for (i = 0; i < x; i++)  for (j = 0; j < y; j++){  printf("graf[%d][%d]: %d\n",i,j,graf[i][j]);  }  return 0;  } |

1. Berapa jumlah simpul dari graf kode program berikut ini ?

a. 6 b. 7 c. 8 d. 9 e. 13

|  |
| --- |
| #include<stdio.h>  #define Z 999  void A(int n, int v, int cost[7][7], int dist[10]){  int i, u, count, w, F[10], min;  for (i = 1; i <= n; i++){  F[i] = 0;  dist[i] = cost[v][i];  }    count = 2;    while (count <= n){  min = 99;  for (w = 1; w <= n; w++){  if (dist[w] < min && !F[w]){  min = dist[w];  u = w;  }  }    F[u] = 1;  count++;  for (w = 1; w <= n; w++){  if ((dist[u] + cost[u][w] < dist[w]) && !F[w]){  dist[w] = dist[u] + cost[u][w];  }  }  }  }  int main()  {  int n, v, i, dist[10] = {0};    n = 6;  printf("\nthe number of nodes: %d \n", n);    printf("\nthe cost matrix:\n");  int graf[7][7] = { { 0, 0, 0, 0, 0, 0, 0},  { 0, Z, 5, Z, Z, Z, 8},  { 0, 5, Z, 7, Z, 2, Z},  { 0, Z, 7, Z, 9, Z, Z},  { 0, Z, Z, 9, Z, 4, Z},  { 0, Z, 2, Z, 4, Z, Z},  { 0, 8, Z, Z, Z, Z, Z}  };    v = 2;  printf("\nthe source matrix: %d \n", v);    A(n, v, graf, dist);    printf("\nShortest path:\n");  for (i = 1; i <= n; i++){  if (i != v){  printf("%d->%d,graf=%d\n", v, i, dist[i]);  }  }  return 0;  } |

1. Ada berapa kesalahan sintaks pada kode program berikut ini ? Sebutkan di bagian mana saja!

a. 1 b. 3 c. 5 d. 9 e. 0

|  |
| --- |
| #include<stdio.h>  #define Z 999  void A(int n, int v, int cost[7][7], int dist[10]){  int i, u, count, w, F[10], min;  for (i = 1; i <= n; i++){  F[i] = 0;  dist[i] = cost[v][i];  }    count = 2;    while (count <= n){  min = 99;  for (w = 1; w <= n; w++){  if (dist[w] < min && !F[w]){  min = dist[w];  u = w;  }  }    F[u] = 1;  count++;  for (w = 1; w <= n; w++){  if ((dist[u] + cost[u][w] < dist[w]) && !F[w]){  dist[w] = dist[u] + cost[u][w];  }  }  }  }  int main()  {  int n, v, i, dist[10] = {0};    n = 6;  printf("\nthe number of nodes: %d \n", n);    printf("\nthe cost matrix:\n");  int graf[7][7] = { { 0, 0, 0, 0, 0, 0, 0},  { 0, Z, 5, Z, Z, Z, 8},  { 0, 5, Z, 7, Z, 2, Z},  { 0, Z, 7, Z, 9, Z, Z},  { 0, Z, Z, 9, Z, 4, Z},  { 0, Z, 2, Z, 4, Z, Z},  { 0, 8, Z, Z, Z, Z, Z}  };    v = 2;  printf("\nthe source matrix: %d \n", v);    A(n, v, graf, dist);    printf("\nShortest path:\n");  for (i = 1; i <= n; i++){  if (i != v){  printf("%d->%d,graf=%d\n", v, i, dist[i]);  }  }  return 0;  } |

# Bagian IV – Selesai

1. Informasi apa saja yang Anda butuhkan ketika sedang menyelesaikan soal graf kode program di kakas CodeViz ?

[ ] gambar simpul

[ ] nama atau nilai suatu simpul

[ ] bentuk simpul (seperti lingkaran, persegi, atau bentuk bangun datar lainnya)

[ ] gambar sisi graf sebagai penghubung antar simpul

[ ] nilai bobot pada sisi graf berbobot

[ ] gambar anak panah untuk graf berarah

[ ] informasi derajat pada simpul

[ ] animasi proses pencarian dari satu simpul ke simpul lain

[ ] animasi proses pembobotan dari satu sisi graf ke sisi lain

[ ] lainnya: ...

1. [open/opini] Animasi graf seperti apa yang Anda harapkan agar efektif dalam memahami eksekusi graf kode program ? Informasi seperti apa yang dibutuhkan ketika Anda mencoba memahami eksekusi graf kode program ?
2. Apakah Anda bersedia untuk di wawancara ketika dibutuhkan? Ya, Saya Bersedia / Tidak Bersedia