Rancangan Kuesioner Revisi 1.0

# Pre-tes

## Soal 1 – ID: pre1

Apa yang dimaksud dengan graf ?

**Kunci Jawaban: B**

1. Graf adalah objek dasar pelajaran dalam teori graf.
2. Graf adalah himpunan dari objek-objek yang dinamakan titik, simpul, atau sudut dihubungkan oleh penghubung yang dinamakan garis atau sisi.
3. Graf adalah gambar kurva (garis atau sisi) yang membentuk bangun datar berupa persegi panjang.
4. Graf adalah gambar grafik untuk kepentingan visualisasi data.
5. Graf adalah objek dasar yang terdiri dari bangun datar dan garis.

## Soal 2 – ID: pre2

Hitunglah komponen berikut ini berdasarkan pada gambar di bawah ini:

|  |  |
| --- | --- |
| Simpul = ... buah  Sisi = ... buah  Derajat pada simpul X = ... buah   1. 3, 11, dan 5 2. 5, 6, dan 5 3. 9, 18, dan 6 4. 9, 14, dan 5 5. 10, 14, dan 5 | **X** |

**Kunci Jawaban: D**

## Soal 3 – ID: pre3

Apakah Anda familiar dengan teori graf ? Jawab: Ya, Tidak, Tidak tahu

## Soal 4 – ID: pre4

Apakah Anda familiar dengan bahasa pemrograman C ? Jawab: Ya, Tidak, Tidak tahu

## Soal 4.1 – ID: pre4.1 (Jika soal 4 dijawab ‘Ya’)

Apakah Anda pernah memprogram struktur data graf dengan bahasa pemrograman C ? Jawab: Pernah, Tidak pernah, Lupa/Tidak tahu

## Soal 4.2 – ID: pre4.2 (Jika soal 4.1 dijawab ‘Pernah’)

Kapan Anda kali terakhir menggunakan bahasa pemrograman C ?

# Latihan

## Soal 1 – ID: latihanOPT

Silakan perbaiki kesalahan sintaks pada kode program berikut ini !

|  |  |
| --- | --- |
| Berapa nilai L yang dicetak ?   1. 4928 2. 4892 3. 4298 4. 4829 5. Tidak ada jawaban | int x(r){  return 3.14\*r\*r;  }  int main(){  int L;  L = x(37);  print(“L = %d”, L);  return 0  } |

**Kunci Jawaban: C**

## Soal 2 – ID: latihanCodeViz

Silakan perbaiki kesalahan sintaks pada kode program berikut ini !

|  |  |
| --- | --- |
| Berapa nilai M[2][3] ?   1. 29 2. 5 3. 1 4. 0 5. null | int main(){  int M[5][5] = {0};  int y = 29;  Int z = y \* M[2][3]  retun 0;  } |

**Kunci Jawaban: D**

# Simulasi 1

## Soal 1 – ID: simulasi1OPT1

Silakan perbaiki kesalahan sintaks pada kode program berikut ini !

|  |  |
| --- | --- |
| Berapa bobot antara simpul 7 dan 1 dari struktur data graf pada kode program tersebut ?   1. 10 2. 4 3. 7 4. 3 5. Tidak dapat ditentukan   **Kunci Jawaban: E** | #include “stdio.h”  int ar[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);  print("\n\nMinimum cost is %d\n ", cost);  return 0;  } |

## Soal 2 – ID: simulasi1OPT2

Silakan perbaiki kesalahan sintaks pada kode program berikut ini !

|  |  |
| --- | --- |
| Apa kegunaan fungsi XYZ dibuat ?   1. Menghitung derajat simpul 2. Menghitung jumlah sisi 3. Menghitung simpul yang berdekatan 4. Menghitung jarak antar simpul 5. Tidak dapat ditentukan   **Kunci Jawaban: D** | #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 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 += grf[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 n;  }  int main()  {  srand(time(NULL));  Z();  printf("\n");  fungsiku(0);  printf("\n\nMin: %d\n ", c);  return 0;  } |

## Soal 3 – ID: simulasi1OPT3

Silakan perbaiki kesalahan sintaks pada kode program berikut ini !

|  |  |
| --- | --- |
| Berapa jumlah simpul pada kode program berikut ini ?   1. 3 2. 8 3. 7 4. 2 5. 4   **Kunci Jawaban: E** | #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()  {  print("\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;  } |

## Soal 1 – ID: simulasi1CodeViz1

Silakan perbaiki kesalahan sintaks pada kode program berikut ini !

|  |  |
| --- | --- |
| Apa tujuan fungsi M dibuat ?   1. Menghitung jarak antar simpul 2. Menghitung jarak antar sisi 3. Menghitung angka random 4. Menghitung angka ganjil / genap 5. Tidak diketahui   **Kunci Jawaban: C** | #include stdio.h  #include time.h  #define infiniti 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];  n = 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;  } |

## Soal 2 – ID: simulasi1CodeViz2

Silakan perbaiki kesalahan sintaks pada kode program berikut ini !

|  |  |
| --- | --- |
| Berapa jumlah simpul yang berderajat 3 dalam kode program tersebut ?   1. 4 2. 5 3. 20 4. 12 5. 0   **Kunci Jawaban: A** | #include stdio.h  int baris = 5, kolom = 4;  int x, y;  int main()  {  Int G[5][4] = { { 0, 2, 3, 1},  { 2, 0, 7, 9},  { 3, 7, 0, 8},  { 1, 9, 8, 0}  };  print("\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;  } |

## Soal 3 – ID: simulasi1CodeViz3

Silakan perbaiki kesalahan sintaks pada kode program berikut ini !

|  |  |
| --- | --- |
| Berapa bobot minimum antara simpul 1 dan 4 pada kode program tersebut ?   1. 2 2. 3 3. 7 4. 5 5. Tidak ada jawaban   **Kunci Jawaban: E** | #include stdo.h  int row = 3, col = 3;  int i, j;  Int Main()  {  int Graf[3][3] = { { 0, 2, 3},  { 2, 0, 7},  { 3, 7, 0},  };  printf("\nGraf:\n");  for (i = 0; i < row; i++)  for (j = 0; j < col; j++)  print("G[%d][%d]: %d\n",i,j,G[i][j]);  return 0;  } |

# Post-tes 1

## Soal 1 – ID: post1

Berapa total simpul pada kode program di bawah ini ?

|  |  |
| --- | --- |
| 1. 7 2. 6 3. 5 4. 4 5. Tidak ada jawaban   **Kunci Jawaban: B** | #include "stdio.h"  #define X 99  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, X, 5, X, X, X, 8},  { 0, 5, X, 7, X, 2, X},  { 0, X, 7, X, 9, X, X},  { 0, X, X, 9, X, 4, X},  { 0, X, 2, X, 4, X, X},  { 0, 8, X, X, X, X, X}  };    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;  } |

# Simulasi 2

## Soal 1 – ID: simulasi2CodeViz1

Silakan perbaiki kesalahan sintaks pada kode program berikut ini !

|  |  |
| --- | --- |
| Berapa derajat (degree) yang dimiliki oleh simpul 3 pada kode program tersebut ?   1. 1 2. 2 3. 3 4. 4 5. 5   **Kunci Jawaban: C** | #include "stdio.h"  int row = 5, col = 4;  int i, j;  int main()  {  Int graf[5][4] = { { 0, 2, 3, 1},  { 2, 0, 7, 9},  { 3, 7, 0, 8},  { 1, 9, 8, 0}  };  print("\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;  } |

## Soal 2 – ID: simulasi2CodeViz2

Berapa jumlah minimum biaya (cost) yang diperlukan dari simpul 1 hingga simpul 7 dari kode program berikut ini ?

|  |  |
| --- | --- |
| 1. 14 2. 13 3. 12 4. 11 5. Tidak dapat dihitung   **Kunci Jawaban: B** | #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;  } |

## Soal 3 – ID: simulasi2CodeViz3

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

|  |  |
| --- | --- |
| 1. 1 2. 2 3. 3 4. 4 5. 5   **Kunci Jawaban: C** | #include<stdio.h>  int row = 3, col = 3;  int i, j;  int main()  {  int G[3][3] = { { 0, 2, 3},  { 2, 0, 7},  { 3, 7, 0},  };  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;  } |

## Soal 1 – ID: simulasi2OPT1

Silakan perbaiki kesalahan sintaks pada kode program berikut ini !

|  |  |
| --- | --- |
| apa nama algoritma yang paling sesuai untuk kode program tersebut ?   1. Dijkstra 2. Bellman Ford 3. Depth First Search (DFS) 4. Breadth First Search (BFS) 5. Tidak diketahui   **Kunci Jawaban: A** | #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);    print("\nthe cost matrix:\n");  int cost[7][7] = { { 0, 0, 0, 0, 0, 0, 0},  { 0, infinity, 5, infinity, infiniti, infinity, 8},  { 0, 5, infinity, 7, infinity, 2, infinity},  { 0, infinity, 7, infiniti, 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;  } |

## Soal 2 – ID: simulasi2OPT2

Berapa bobot minimum dari simpul 2 ke simpul 3 pada kode program berikut ini ?

|  |  |
| --- | --- |
| 1. 2 2. 3 3. 4 4. 8 5. 11   **Kunci Jawaban: C** | #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;  } |

## Soal 3 – ID: simulasi2OPT3

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

|  |  |
| --- | --- |
| 1. 6 2. 7 3. 8 4. 9 5. 13   **Kunci Jawaban: A** | #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;  } |

# Post-tes 2

## Soal 1 – ID: post2

Berapa jumlah simpul dari kode program berikut ini ?

|  |  |
| --- | --- |
| 1. 6 2. 5 3. 4 4. 7 5. Tidak ada jawaban   **Kunci Jawaban: B** | #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;  } |

# Penutup

## Soal 1 – ID: close1

Informasi apa saja yang Anda butuhkan ketika sedang menyelesaikan soal kode program struktur data graf saat menggunakan 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: ...

## Soal 2 – ID: close2

Animasi graf seperti apa yang Anda harapkan agar efektif dalam memahami eksekusi kode program struktur data graf ? atau informasi seperti apa yang dibutuhkan ketika Anda mencoba memahami eksekusi kode program struktur data graf ?

## Soal 3 – ID: close3

Apakah Anda bersedia untuk wawancara ketika dibutuhkan ?

Saya Bersedia / Tidak Bersedia