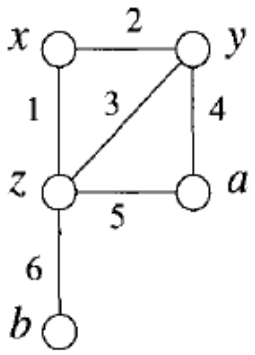
PERTEMUAN MINGGU KE 13, 14, 15, DAN 16

Materi Pembelajaran : Definisi-definisi dan komponen-komponen dalam sebuah graf, *vertex degree*, *paths*, *cycles*, graf planar, Euler *trails* dan *circuits*, Hamilton *paths* dan *cycles*, *Trees*, *Spanning trees*.

A. Pengertian Graf

Graf G adalah pasangan $(V(G), X(G))$, dimana $V(G)$ adalah himpunan berhingga yang elemen-elemennya disebut titik $(vertex)$ dan $X(G)$ adalah himpunan pasangan-pasangan tak berurut dari elemen-elemen $V(G)$ yang berbeda yang disebut sisi $(edge)$.

Contoh dari graf yaitu :



$V(G) = \{a, b, x, y, z\}$ himpunan dari $vertex$ graf di atas.

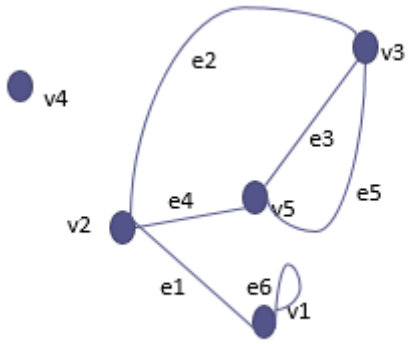
$E(G) = \{1,2,3,4,5,6\}$ himpunan dari $edges$ graf di atas.

B. Incident dan Adjacent

Dua buah $vertex$ dikatakan $adjacent$ apabila kedua $vertex$ tersebut terhubung secara langsung oleh sebuah $edge$.

Contoh :

Perhatikan graf berikut :



Apakah $v2$ dan 3 $adjacent$? Berikan alasan. Buatlah $adjacent$ matriks dari graf tersebut.

Penyelesaian :

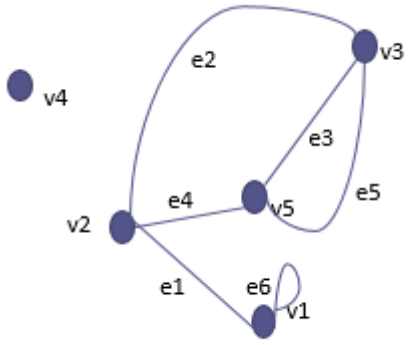
$v2$ dan $v3$ $adjacent$ karena terhubung secara langsung oleh $edge$ $e2$. $Adjacent$ matriksnya adalah sebagai berikut :

	v1	v2	v3	v4	v5
v1	2	1	0	0	0
v2	1	0	1	0	1
v3	0	1	0	0	2
v4	0	0	0	0	0
v5	0	1	2	0	0

Jika e merupakan $edge$ dengan $vertex$ v dan w yang ditulis $e = (v, w)$, maka v dan w disebut terletak pada e dan e disebut $incident$ dengan v dan w.

Contoh :

Perhatikan graf berikut :



Buatlah $incident$ matriks dari graf tersebut.

Penyelesaian :

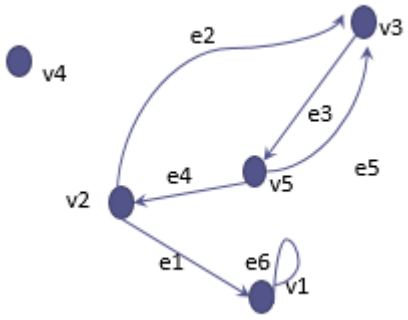
$Incident$ matriks dari graf tersebut adalah :

	e1	e2	e3	e4	e5	e6
v1	1	0	0	0	0	2
v2	1	1	0	1	0	0
v3	0	1	1	0	1	0
v4	0	0	0	0	0	0
v5	0	0	1	1	1	0

C. Directed Graph, Undirected Graph, dan Multigraph

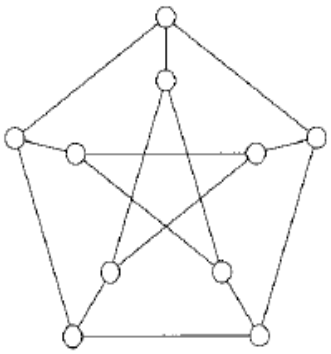
$Directed$ graf adalah suatu graf G yang terdiri dari himpunan $vertex$ $V = \{v1, v2, \dots, vp\}$ dan himpunan edge $E = \{e1, e2, e3, \dots, eq\}$ dan suatu pemetaan f yang memetakan setiap edge onto terhadap pasangan berurut dari $vertex$ (vi, vj), yang mana $vertex$ disajikan dengan titik dan $edge$ disajikan dengan sepotong garis berarah dari vi ke vj.

Contoh graf berarah :



$Undirected$ graf adalah suatu graf G yang terdiri dari pasangan himpunan $\{V, E\}$ yang mana V adalah himpunan berhingga tak kosong yang anggotanya disebut $vertex$ dan E adakah himpunan berhingga yang anggotanya disebut $edge$, yang merupakan potongan garis tak berarah yang menghubungkan pasangan $vertex$ tersebut.

Contoh $undirected$ graf :



Petersen Graph.

$Size$ dari graf adalah jumlah $vertex$ yang terdapat pada sebuah graf. $Order$ adalah jumlah $edge$ yang terdapat pada sebuah graf.

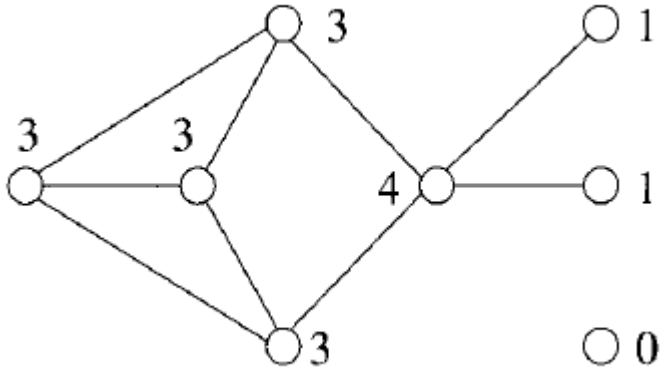
Contoh :

$Size$ dari Petersen Graph yaitu bernilai 10 sedangkan $order$ dari Petersen Graph yaitu bernilai 10.

D. Degree dari Vertex

$Degree$ atau derajat dari vertex vi adalah banyaknya $edge$ yang $incident$ pada $vertex$ vi dalam graf G dan loop dihitung dua kali. $Degree$ dinotasikan dengan $d(vi)$. $Vertex$ yang berderajat 1 disebut $end\ vertex$ $(pendant\ vertex)$ sedangkan $vertex$ yang berderajat nol disebut dengan $isolated\ vertex$ $(vertex\ terasing)$.

Berikut adalah graf beserta $degree$ di setiap $vertex$



E. Path dan Cycle dalam Graf

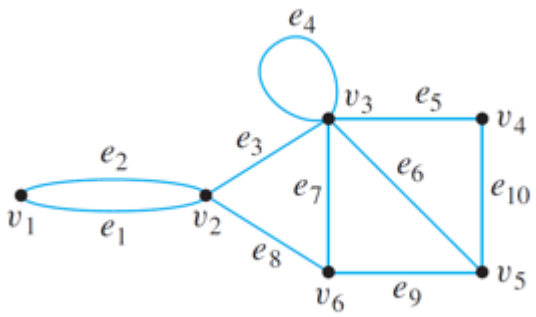
$Path\ v$ ke w merupakan sebuah walk yang tidak terdapat $edges$ maupun $vertex$ yang berulang. Sedangkan $cycle$ merupakan $path$ yang tertutup artinya $vertex$ yang pertama sama dengan $vertex$ yang terakhir.

***Teorema** : Jika terdapat sebuah walk yang dimulai dari vertex y ke z pada sebuah graf G, dimana y tidak sama dengan z, maka terdapat sebuah path pada graf tersebut dengan y merupakan vertex pertama dan z merupakan vertex yang terakhir.*

***Teorema** : Sebuah graf disebut bipartite jika dan hanya jika graf tersebut tidak terdapat cycle yang memiliki panjang ganjil.*

Contoh :

Tentukan $path$ dan $cycle$ yang terdapat pada graf berikut :



Penyelesaian :

Contoh $path$ pada graf tersebut adalah $v1e1v2e8v6e9v5e10v4e5v3$

Contoh $cycle$ pada graf tersebut adalah $v2e8v6e9v5e10v4e5v3e3v2$

F. Graf Planar

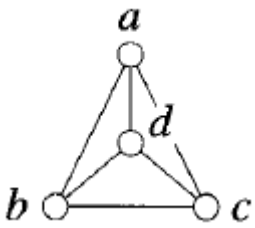
Sebuah graf $G = (v, e)$ disebut graf planar apabila graf tersebut dapat digambarkan dalam sebuah bidang datar tanpa ada $edge$ yang saling berpotongan (kecuali sisi-sisi berpotongan pada sebuah $vertex$)

***Teorema Kuratowski** : Graf G bersifat planar jika dan hanya jika ia tidak mengandung subgraph yang sama dengan salah satu graf kuratowski atau humomorfis dengan salah satunya.*

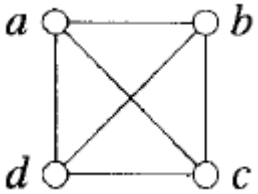
Sifat dari graf $Kuratowski$ adalah :

1. Kedua graf $Kuratowski$ adalah graf teratur.
2. Kedua graf $Kuratowski$ adalah graf non planar.
3. Penghapusan $edge$ atau $vertex$ dari graf $Kuratowski$ menyebabkan menjadi graf planar.
4. K_5 adalah graf non planar dengan jumlah simpul minimum, $K_{3,3}$ adalah graf non planar dengan jumlah sisi minimum.

Berikut merupakan contoh sebuah graf planar



Sedangkan contoh graf non planar yaitu :



G. Graf Euler

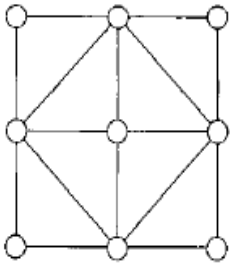
Euler path merupakan *path* yang melewati setiap *edges* tepat satu kali. Sedangkan *euler circuit* merupakan *euler path* yang dimulai dan diakhiri oleh *vertex* yang sama.

***Teorema 1 Euler** : Jika sebuah graf terdapat paling tidak ada satu vertex yang degreenya ganjil maka tidak terdapat euler circuit didalamnya.*
Jika sebuah graf connected dan setiap vertex memiliki degree genap maka terdapat paling tidak satu euler circuit didalamnya.

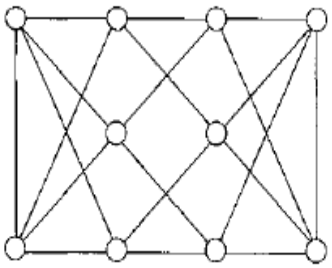
Contoh :

Apakah graf-graf berikut merupakan *euler circuit* atau *euler path* atau tidak keduanya ?

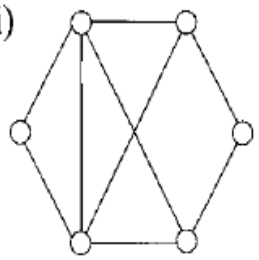
(i)



(ii)



(iii)



Penyelesaian

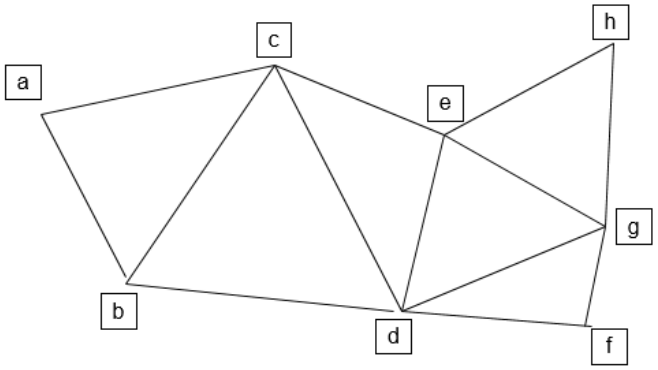
i) Bukan merupakan $euler\ path$ maupun $euler\ circuit$ karena terdapat 4 $vertex$ yang memiliki$degree$ ganjil.

ii) Graf ini terdapat $euler\ circuit$ karena setiap $vertex$ memiliki $degree$ genap.

iii) Graf ini terdapat $euler\ path$ karena memiliki tepat 2 $vertex$ yang $degree$nya ganjil.

***Teorema 2 Euler** : Jika sebuah graf terdapat lebih dari dua vertex yang degreenya ganjil maka tidak terdapat euler path dalam graf tersebut.*
Jika sebuah graf terhubung satu sama lain dan terdapat dua vertex yang degreenya ganjil, maka graf tersebut paling tidak terdapat satu euler path. Path tersebut harus dimulai dari salah satu vertex yang degreenya ganjil dan berakhir pada vertex ganjil yang lain.

Contoh :



Apakah graf tersebut merupakan $euler\ path$? Jika ya tentukan rutenya.

Penyelesaian :

Graf tersebut merupakan graf yang memiliki $euler\ path$ karena terdapat tepat dua $vertex$ yang $degree$nya ganjil. Rutenya yaitu $b, a, c, b, d, c, e, h, g, e, d, g, f, d$.

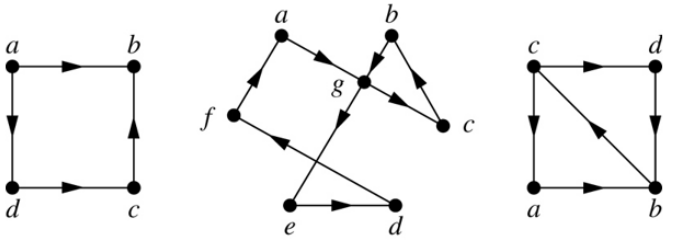
Euler Walks untuk graf berarah

Jika setiap $vertex$ yang jumlah $degree$ arahnya ke dalam = jumlah $degree$ yang arahnya keluar maka terdapat $euler\ circuit$.

Jika terdapat satu $vertex$ yang jumlah $degree$ arahnya ke dalam = jumlah $degree$ arah keluar $+1$, dan terdapat satu $vertex$ yang jumlah $degree$ arahnya ke dalam = jumlah $degree$ arah keluar -1, dan semua $vertex$ lainnya memiliki jumlah $degree$ arah ke dalam = jumlah $degree$ arah keluar maka terdapat $euler\ path$.

Contoh :

1.



Apakah graf-graf tersebut merupakan *euler circuit* dan/atau *euler path*? Jika ya, tentukan rutenya.

Penyelesaian :

H_1 : bukan merupakan *euler circuit* maupun *euler path*

H_2 : merupakan *euler circuit* dan *euler path*. Rute *euler circuit* yaitu $a, g, c, b, g, e, d, f, a$. Rute *euler path* yaitu $a, g, c, b, g, e, d, f, a$.

H_3 : merupakan *euler path* namun tidak terdapat *euler circuit*. Rute dari *euler path* yaitu c, a, b, c, d, b.

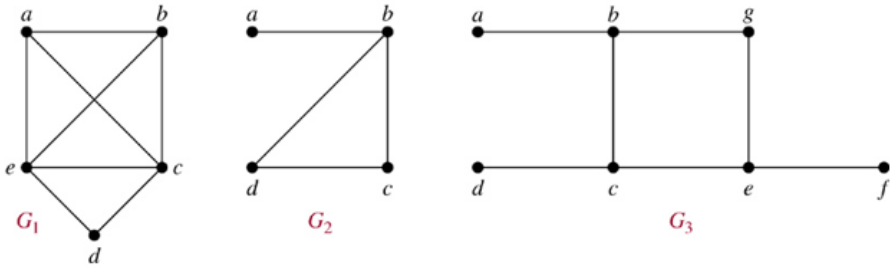
H. Hamilton Walks

Hamilton path dalam sebuah graf G adalah sebuah *path* yang mengunjungi setiap *vertex* di G tepat satu kali. *Hamilton Circuit* merupakan *hamilton path* yang kembali ke titik awalnya.

***Syarat Cukup** : Jika G merupakan graf dengan v vertex,v ≥3, dan setiap vertex memiliki degree paling sedikit $\frac{v}{2}$, maka G adalah Hamiltonian.*

***Teorema :** Sebuah bipartite graf dengan himpunan vertex yang berukuran m dan n dapat terkandung Hamilton Cycle jika m=n, dan dapat terkandung Hamilton Path hanya jika m dan n memiliki selisih 1.*

Contoh :



Manakah dari graf di atas yang memiliki $Hamilton\ Circuit$? Jika bukan $Hamilton\ Circuit$ apakah $Hamilton\ Path$? Jika ya, tentukan rutenya.

Penyelesaian :

G_1 : Merupakan $on\ Circuit$: *a, b, c, d, e, a.*

G_2 : Tidak memiliki $Hamilton\ Circuit$ namun memiliki $on\ Path$: *a, b, c, d.*

G_3 : Tidak memiliki $Hamilton\ Circuit$ maupun $Hamilton\ Path$.

I. Graph Trees dan Spanning Trees

Graph Trees

Pohon (tree) telah digunakan sejak tahun 1857 oleh matematikawan inggris yang bernama Arthur Cayley untuk menghitung jumlah senyawa kimia. Silsilah keluarga biasanya juga digambarkan pasa bentuk pohon. Pohon (tree) adalah merupakan graf yang tak berarah terhubung yang tidak memuat sirkuit sederhana. Diagram pohon dapat digunakan sebagai alat untuk memecahkan masalah dengan menggambarkan semua alternative pemecahan.

Jadi, dapat disimpulkan bahwa pohon adalah suatu graph yang banyak vertexnya sampai dengan $n = (n > 1)$, jika :

- Graph tersebut tidak mempunyai lingkar (cycle free) dan banyaknya rusuk $(n - 1)$.
- Graph tersebut terhubung .

Berikut adalah beberapa sifat pohon :

1. Misalkan G merupakan suatu graf dengan n buah simpul dan tepat $n - 1$ buah sisi.
2. Jika G tidak mempunyai sirkuit t maka G merupakan pohon.
3. Suatu pohon dengan n buah simpulnya mempunyai $n - 1$ buah sisi.
4. Setiap pasang simpul didalam suatu pohon terhubung dengan lintasan tunggal.

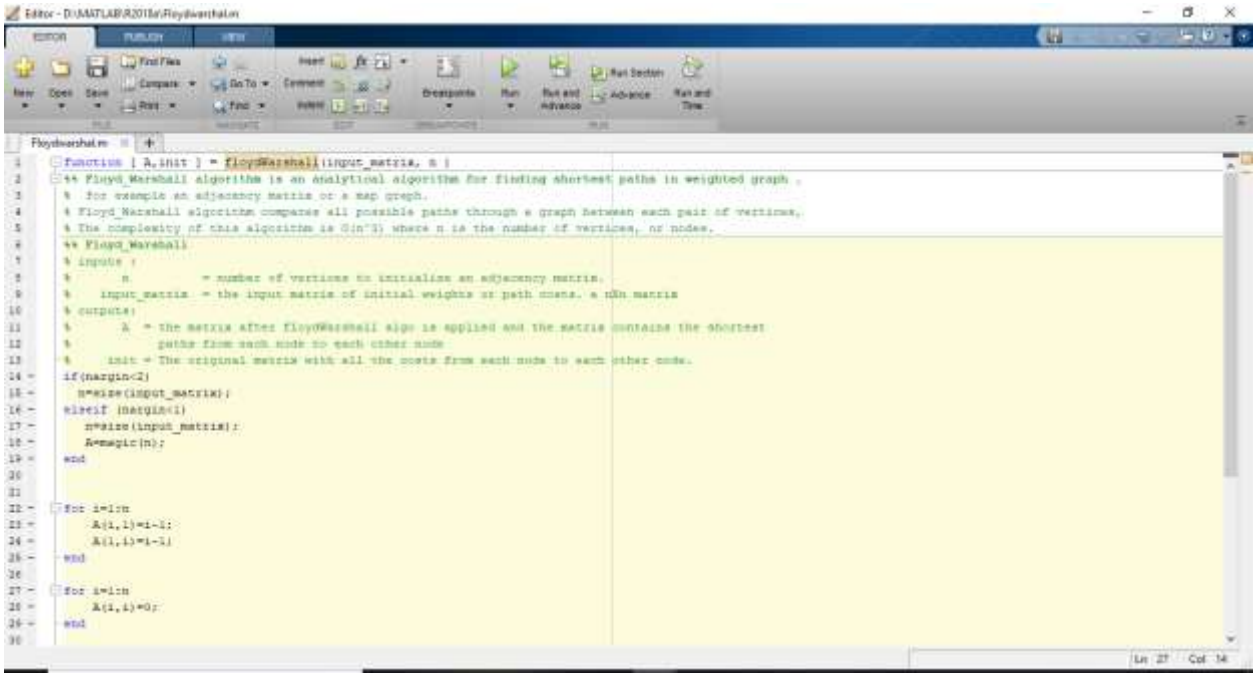
5. Misalkan G adalah graf sederhana dengan jumlah simpul, jika G tidak mengandung sirkuit maka penambahan satu sisi pada graf hanya akan membuat satu sirkuit.

Spanning Tree

Spanning Tree adalah subgraph G merupakan pohon dan mencakup semua titik dari G. Pohon merentang diperoleh dengan cara menghilangkan sirkuit didalam graf tersebut.

5.1. Program Komputer Floyd Warshall (Luaran)

Algoritma Floyd Warshall dikerjakan dalam bahasa pemrograman untuk menyederhanakan proses perhitungan. Bagian dari program yang telah dibuat dapat dilihat pada gambar 10.



Gambar 10. Coding *Floyd Warshall*

Coding program lengkap pada gambar 10 adalah sebagai berikut:

M Script

```
function [ A,init ] = floydWarshall(input_matrix, n )
%% Floyd_Warshall algorithm is an analytical algorithm for finding shortest paths in weighted graph
,
%  for example an adjacency matrix or a map graph.
% Floyd_Warshall algorithm compares all possible paths through a graph between each pair of
vertices,
% The complexity of this algorithm is O(n^3) where n is the number of vertices, or nodes.
%% Floyd_Warshall
% inputs :
%       n          = number of vertices to initialize an adjacency matrix.
%    input_matrix  = the input matrix of initial weights or path costs. a nXn matrix
% outputs:
```

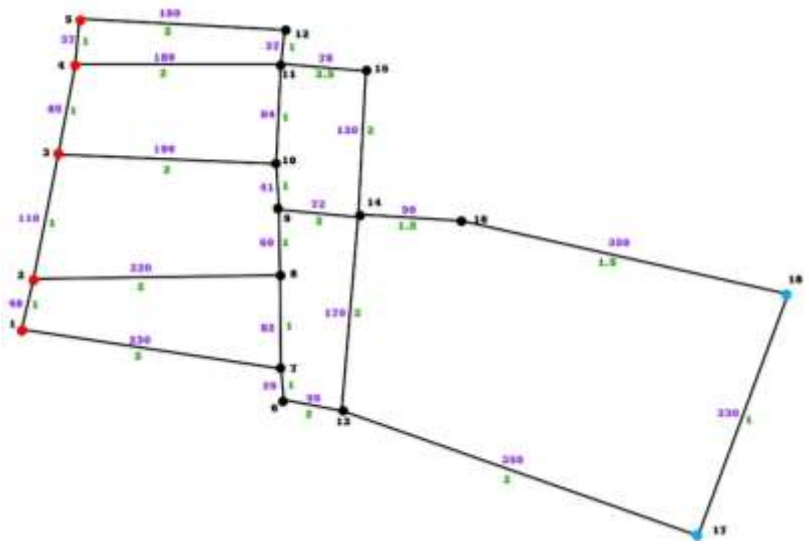
```
%       A  = the matrix after floydWarshall algo is applied and the matrix contains the shortest
%            paths from each node to each other node
%     init = The original matrix with all the costs from each node to each other node.
if(nargin<2)
  n=size(input_matrix);
elseif (nargin<1)
   n=size(input_matrix);
   A=magic(n);
end


for i=1:n
    A(i,1)=i-1;
    A(1,i)=i-1;
end

for i=1:n
    A(i,i)=0;
end

for i=2:n
    for j=2:n
        A(i,j)=input_matrix(i,j);  % input matrix, values
    end
end


init=A;   % temp variable to store the old matrix
for int_city=2:n
    for city1=2:n
        for city2=2:n
            A(city1,city2)=min(A(city1,city2),A(city1,int_city)+A(int_city,city2));
        end  % floyd-warshall
    end
end
```

5.2. Perhitungan dan Analisis Hasil

Rute evakuasi tsunami dalam penelitian ini difokuskan pada jalan. Data yang digunakan diambil dari peta masing-masing kelurahan di Kota Manado yang diperkirakan bisa mengalami dampak langsung jika terjadi Tsunami di Kota Manado.

Gambar 11. Peta salah satu kelurahan yang menjadi objek penelitian

Bobot (weight) dari graf diperoleh dengan mengalikan terlebih dahulu jarak antar verteks (titik) dan akses jalan. Bobot akses jalan mempunyai 5 kategori, yaitu bobot 1 untuk jalan yang bisa dilewati oleh 4 mobil dan ada gedung/rumah di sisi jalan, bobot 1.5 untuk jalan yang bisa dilewati 2-3 mobil dan ada gedung/rumah di sisi jalan, bobot 2 untuk jalan yang bisa dilewati oleh 1-2 mobil dan ada gedung/rumah di sisi jalan, bobot 2.5 untuk jalan yang bisa dilewati oleh 1 mobil dan ada gedung/rumah di sisi jalan, dan bobot 3 untuk jalan yang tidak bisa dilewati oleh mobil dan ada gedung/rumah di sisi jalan. Bobot ini ditentukan dengan memperhitungkan akses jalan yang berbeda-beda. Jalan yang lebar dan mudah dilewati memiliki bobot yang lebih kecil dibandingkan jalan yang sempit dan sulit dilewati. Pemilihan bobotnya sendiri tidak bisa terlalu besar agar tidak mempengaruhi rasio perbandingan jalan. Graf pada gambar 12 dibentuk dari peta (gambar 11), bobot jarak ditandai dengan warna ungu dan bobot akses jalan ditandai dengan warna hijau.



Gambar 12. Jalur Evakuasi
● : Lokasi awal ● : Tujuan

Jalur evakuasi terpendek yang terbentuk setelah setiap bobot pada gambar 12 diolah pada program yang telah dibuat dapat dilihat pada gambar 13.

1→17 (1295)
1 → 7 → 6 → 13 → 17

2→17 (1343)
2 → 1 → 7 → 6 → 13 → 17

3→17 (1398)
3 → 10 → 9 → 8 → 7 → 6 → 13 → 17

4→17 (1462)
4 → 11 → 10 → 9 → 8 → 7 → 6 → 13 → 17

5→17 (1499)
5 → 4 → 11 → 10 → 9 → 8 → 7 → 6 → 13 → 17

1→18 (1352)
1 → 2 → 8 → 9 → 14 → 16 → 18

2→18 (1304)
2 → 8 → 9 → 14 → 16 → 18

3→18 (1225)
3 → 10 → 9 → 14 → 16 → 18

4→18 (1289)
4 → 11 → 10 → 9 → 14 → 16 → 18

5→18 (1326)
5 → 4 → 11 → 10 → 9 → 14 → 16 → 18

Gambar 13. Hasil berupa jalur evakuasi terpendek
dari setiap titik awal ke setiap titik tujuan