

VERTEX COVER MENGGUNAKAN METODE GREEDY UNTUK OPTIMASI PENEMPATAN TEMPAT SAMPAH (STUDI KASUS AREA GEDUNG PERKULIAHAN)

**Helena Nurramdhani Irmada, Mayanda Mega Santoni,
Ria Astriratma**

Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
email: helenairmada@upnvj.ac.id
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia

Abstrak

Pengelolaan sampah merupakan hal yang sangat penting dan menjadi tanggung jawab semua pihak di lingkungan Universitas. Tempat sampah biasanya ditempatkan secara acak disetiap sudut/persimpangan jalan / koridor sehingga tidak efisien atau bahkan terdapat area yang tidak terjangkau oleh tempat sampah tersebut. Untuk mengefisienkan jumlah tempat sampah yang digunakan, perlu adanya pengaturan peletakan tempat sampah sedemikian sehingga tempat sampah ditempatkan di beberapa titik yang dapat menjangkau area-area sekitarnya. Permodelan dilakukan dengan menggunakan graf, dan solusi optimasinya menggunakan *vertex cover* dengan metode *greedy*. *Vertex cover* dengan metode *greedy* dapat menyelesaikan permasalahan pengaturan penempatan tempat sampah di beberapa titik di area gedung perkuliahan, sehingga dapat efisien dibandingkan dengan penentuan letak secara manual (acak).

Kata kunci: sistem permodelan, graf, *vertex cover*, algoritma *greedy*

1. PENDAHULUAN

Pengelolaan sampah didefinisikan sebagai kegiatan yang sistematis, menyeluruh, dan berkesinambungan yang meliputi pengurangan dan penanganan sampah (UU No. 18 tahun 2008). Pengelolaan sampah meliputi proses pengumpulan, pengangkutan, pemrosesan, hingga pada proses pendaurulangan. Peningkatan jumlah penduduk membuat pengelolaan sampah ini menjadi hal yang sangat penting karena sampah yang tidak dikelola dengan baik akan berdampak pada kesehatan lingkungan seperti timbulnya bau, mengurangi estetika serta berkembangnya wabah penyakit. Pengelolaan sampah dilakukan disemua lingkungan masyarakat, termasuk lingkungan kampus.

Di lingkungan kampus pengelolaan sampah merupakan tanggung jawab semua pihak, terutama dalam hal pengumpulan sampah. Di luar ruangan kampus, sampah dikumpulkan dalam suatu tempat sampah besar. Setiap tempat sampah sebaiknya ditempatkan dibanyak tempat agar setiap warga kampus tidak kesulitan jika akan membuang sampah, namun permasalahannya jika tidak diatur dengan baik akan terjadi pemborosan tempat sampah, bahkan terdapat beberapa area yang tidak terjangkau oleh tempat sampah. Tempat sampah biasanya ditempatkan disetiap sudut/persimpangan jalan/koridor. Sehubungan dengan permasalahan tersebut, perlu adanya pengaturan peletakkan tempat sampah sedemikian sehingga agar ditempatkan di beberapa titik yang dapat menjangkau semua area sehingga penggunaan tempat sampah menjadi efisien. Penelitian ini bertujuan mengimplementasikan model *vertex cover* menggunakan

algoritma *greedy* dalam masalah penempatan tempat sampah di area gedung perkuliahan.

Vertex Cover merupakan sebuah himpunan *vertex* dari suatu graf sederhana, sisi-sisi dari graf tersebut bersisian dengan satu simpul dari himpunan simpul tersebut. Beberapa aplikasi dari *vertex cover* antara lain dalam bidang *biochemistry* yaitu untuk *SNP Assembly problem*, *Computer Network Security*, *Combinatorial Worm*, dan *Network Security* (Winata, 2010).

2. METODOLOGI PENELITIAN

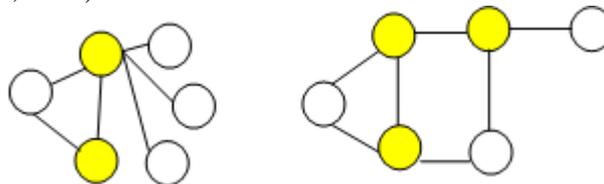
2.1. Teori Graf

Graf merupakan pasangan himpunan (V,E) dengan V yaitu himpunan tidak kosong dari simpul dan E yaitu himpunan sisi yang menghubungkan antara satu simpul dan simpul lainnya. Graf merupakan salah satu model matematika yang dapat diterapkan dalam beberapa kasus yaitu diantaranya rangkaian listrik, isomer senyawa kimia karbon dan lain-lain. Graf dapat direpresentasikan dengan matriks ketetanggaan (*adjacency matrix*) maupun matriks bersisian (*incidency matrix*). Dalam matriks ketetanggaan a_{ij} diberi nilai 1, jika simpul i dan j bertetangga dan 0, jika simpul i dan j tidak bertetangga, sedangkan dalam matriks bersisian a_{ij} diberi nilai 1 jika simpul i dan j bersisian dan 0 jika simpul i dan j tidak bersisian (Munir, 2006).

Penelitian ini menggunakan permodelan graf dengan verteks (simpul) merepresentasikan persimpangan jalan, dan *edge* merepresentasikan jalan. Jalan disini dalam artian jalan kecil di area kampus.

2.2. Vertex Cover Problem dengan Metode Greedy

Vertex cover dari sebuah graf sederhana merupakan sebuah himpunan simpul dari graf, dimana semua sisi dari graf akan berisikan dengan paling tidak dengan satu sudut yang berada dalam himpunan tersebut. *Minimum vertex cover* yaitu *vertex cover* dengan himpunan simpul yang minimum (Winata, 2010).



Gambar 9 Contoh vertex cover, tanda merah sebagai vertex cover dari graf 1 dan 2 (Winata, 2010)

Definisi permasalahan, diberikan input graf $G = (V, E)$ (graf tak berbobot). *Vertex cover* dari suatu graf adalah himpunan bagian dari verteks $C \subseteq V$ sedemikian sehingga untuk setiap sisi $(u, v) \in E$ atau $v \in C$ (atau keduanya). Keluarannya berupa *vertex cover* dengan ukuran minimum (Gao et all, 2018). *Vertex cover* dapat diselesaikan dengan beberapa metode diantaranya : *metode greedy*, aproksimasi, *linear programming*, dan lain sebagainya. Dalam penelitian ini, minimum *vertex cover* dengan metode *greedy* digunakan sebagai optimasi dalam penempatan tempat sampah di area gedung perkuliahan.

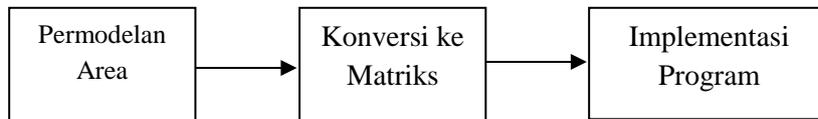
Ide dasar dari metode *greedy* untuk menyelesaikan permasalahan *vertex cover* yaitu : menemukan verteks(simpul) dimana dapat menjangkau beberapa jumlah maksimum edges (sisi). Berdasarkan hal tersebut berikut ini algoritmanya (Lap Chi Lau, 2008):

1. Temukan *vertex* v dengan derajat paling maksimum dalam suatu graph
2. Tambahkan v ke dalam solusi dan hapus v beserta semua sisi yang bersisian dengannya dari graf
3. Ulang hingga semua sisi tertutupi.

Algoritma ini mendekati solusi optimal meskipun tidak dalam semua kasus.

2.3. *Vertex Cover* untuk Permasalahan Penempatan Tempat Sampah

Terdapat beberapa tahap dalam penempatan tempat sampah dengan *vertex cover* metode *greedy* yang digambarkan dalam diagram blok pada Gambar 2.



Gambar 10 Diagram Blok proses *vertex cover* untuk penempatan tempat sampah

Tahap pertama dalam penelitian ini adalah memodelkan area Gedung perkuliahan sebagai graf, dengan setiap persimpangan direpresentasikan dengan *vertex* dan jalan area kampus sebagai *edges*. Tahap kedua, peneliti merepresentasikan graf dengan matriks ketetanggaan, memberikan nilai 1, jika simpul *i* dan *j* bertetangga dan 0, jika simpul *i* dan *j* tidak bertetangga. Tahap ketiga, peneliti mengembangkan program komputer untuk mensimulasikan program komputer *vertex cover* dengan bahasa C. Pendekatan yang digunakan yaitu dengan metode *greedy*. Tahap keempat, peneliti validasikan data yang sudah diperoleh dan selanjutnya membandingkan antar kondisi *existing* dengan hasil simulasi penempatan tempat sampah yang dioptimasi dengan *greedy vertex cover*.

3. HASIL DAN PEMBAHASAN

Sistem Penelitian ini akan memanfaatkan *vertex cover* dengan metode *greedy* untuk masalah penempatan tempat sampah studi kasus di area gedung perkuliahan, agar tempat sampah yang digunakan lebih efisien. Pseudocode *vertex cover* dengan metode *greedy* dapat dilihat pada Gambar 3.

```
PROGRAM VertexCover
DEKLARASI
i, j, v, e, input1, input2, temp_derajat, flag, derajat_max, kandidat, temp_jumlah :
integer
matriks[ ][ ], derajat[ ] : array of integer
ALGORITMA
READ (v,e)
{inisialisasi matriks dengan 0}
FOR i ← 0 TO v DO
    FOR j ← 0 TO v DO
        matriks[i][j] ← 0
    ENDFOR
ENDFOR
{input ke matriks}
FOR i ← 0 TO e DO
    READ(input1, input2)
    matriks[input1 - 1][input2 - 1] ← 1
    matriks[input2 - 1][input1 - 1] ← 1
ENDFOR
{update derajat}
FOR i ← 0 TO v DO
    temp_derajat ← 0
    FOR j ← 0 TO v DO
        IF matriks[i][j] = 1 THEN
            temp_derajat ← temp_derajat + 1
        ENDIF
    ENDFOR
ENDFOR
flag ← 0
WHILE flag = 0 DO
    temp_jumlah ← 0
    derajat_max ← 0
    FOR i ← 0 TO v DO
        IF derajat[i] > derajat_max THEN
            derajat_max ← derajat[i]
            kandidat ← i
        ENDIF
    ENDFOR
```

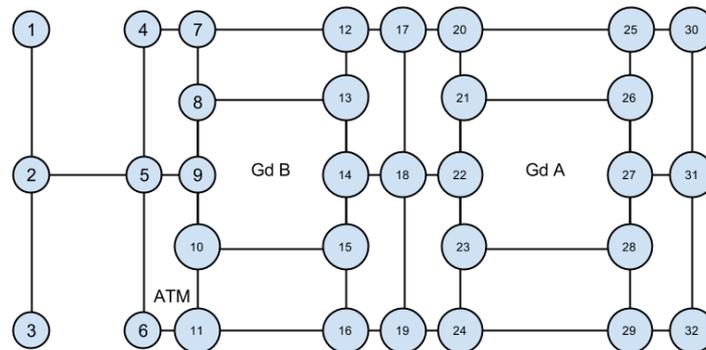
```

ENDFOR
WRITE (kandidat+1)
{update matriks}
FOR i ← 0 TO v DO
  IF matriks[kandidat][i] = 1 THEN
    matriks[kandidat][i] ← 0
  ENDIF
  IF matriks[kandidat][i] ← 1 THEN
    matriks[i][kandidat] ← 0
  ENDIF
ENDFOR
{update derajat}
FOR i ← 0 TO v DO
  temp_derajat ← 0
  FOR j ← 0 TO v DO
    IF matriks[i][j] = 1 THEN
      temp_derajat ← temp_derajat + 1
    ENDIF
  ENDFOR
  derajat[i] ← temp_derajat
ENDFOR
{cek coverage}
FOR i ← 0 TO v DO
  IF derajat[i] = 0 THEN
    temp_jumlah ← temp_jumlah + 1
  ENDIF
ENDFOR
IF temp_jumlah ← v THEN
  flag ← 1
ENDIF
ENDWHILE

```

Gambar 11 Pseudocode minimum vertex cover dengan algoritma greedy

Tahap awal dilakukan permodelan area kampus dengan menggunakan model graf dimana setiap persimpangan direpresentasikan dengan verteks dan jalan direpresentasikan dengan *edges*, area yang diambil adalah area gedung A dan gedung B. Dari tahap pertama diperoleh sebuah model graf dari area gedung perkuliahan dengan contoh area gedung A dan gedung B seperti pada Gambar 4.



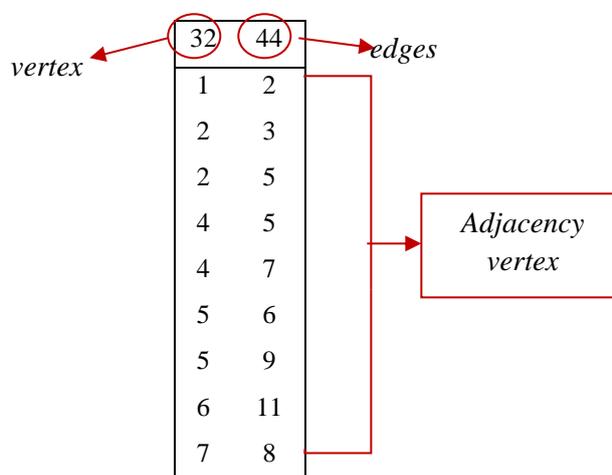
Gambar 12 Model graf area perkuliahan

Graf yang sudah dibuat selanjutnya direpresentasikan dengan matriks ketetanggaan, dengan memberikan nilai 1, jika simpul *i* dan *j* bertetangga dan 0, jika simpul *i* dan *j* tidak bertetangga. Matriks ini menjadi inputan dari program *vertex cover* dengan metode *greedy*, dan selanjutnya output program yang akan didapat yaitu beberapa *vertex* yang telah menjangkau *vertex* lainnya. Kemudian diperoleh matriks ketetanggaannya seperti pada Gambar 5.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

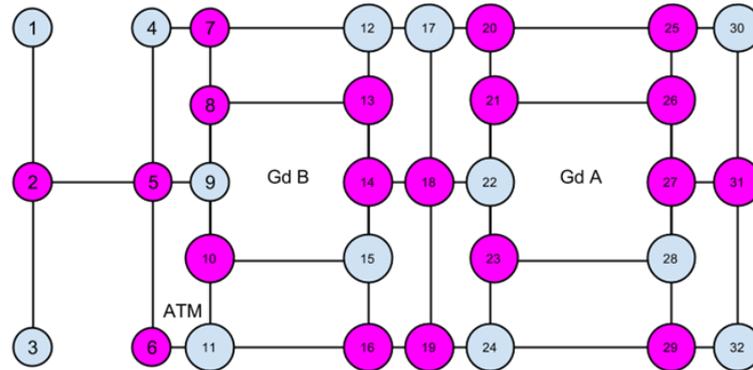
Gambar 13 Matriks ketetangaan

Tahap selanjutnya yaitu mengembangkan program komputer menggunakan bahasa C untuk menyelesaikan solusi *vertex cover* dengan menggunakan metode *greedy*. Input program ini yaitu banyaknya *vertex* dan banyaknya *edge*, setelah itu pengguna diminta untuk menginputkan *vertex* mana saja yang saling terhubung. Contoh masukan dari kasus ini akan diujikan seperti pada Gambar 6.



Gambar 14 Contoh masukan

Output yang dihasilkan program yaitu sebanyak 20 *vertex* yaitu 5, 18, 7, 10, 13, 16, 20, 23, 26, 29, 31, 2, 6, 8, 12, 14, 19, 21, 25, 27. Hal ini mengindikasikan bahwa terdapat 20 tempat sampah yang digunakan, hal ini lebih sedikit dibanding dengan kondisi existing yaitu sebanyak 32 tempat sampah. Dengan demikian penempatan tempat sampah di area perkuliahan gedung A dan gedung B dapat dilihat pada Gambar 7.



Gambar 15 Output model graf penempatan tempat sampah

4 KESIMPULAN

Penelitian ini bertujuan untuk menganalisis penggunaan *Vertex cover* sebagai optimasi Penempatan Tempat Sampah dengan menggunakan Metode *Greedy*. Permodelan yang digunakan dalam penelitian ini yaitu model graf, dengan *vertex* yang merepresentasikan persimpangan dan *edges* merepresentasikan sebagai jalan di area kampus. *Minimum vertex cover* dengan metode *greedy* dapat menyelesaikan permasalahan pengaturan penempatan tempat sampah di beberapa titik di area perkuliahan gedung A dan gedung B, sehingga jumlahnya dapat efisien dan dapat menjangkau semua area di sekitarnya. Metode ini dapat digunakan untuk menentukan peletakan tempat sampah dengan hasil yang lebih baik dibandingkan dengan penentuan letak tempat sampah secara manual.

Referensi

- Gao, W., Friedrich, T., Neumann, F., & Hercher, C. (2018). Randomized Greedy Algorithms for Covering Problems, <https://cs.adelaide.edu.au/~ec/research/randomgreedy.pdf>, diakses: 13 November 2018.
- Lap Chi Lau. (2008). *Combinatorial Optimization and Approximation Algorithms*, www.cse.cuhk.edu.hk/~chi/csc5160-2008/notes/L15-cover.pdf, diakses : 13 November 2018.
- Winata, Kevin. (2010). *Algoritma Vertex Cover dan Aplikasinya*, <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2011-2012/Makalah2011/Makalah-IF2091-2011-068.pdf>, diakses : 13 November 2018.
- Munir, Rinaldi. (2006). Diktat Kuliah IF2153 *Matematika Diskrit*. Program Studi Teknik Informatika, Institut Teknologi Bandung.
- Republik Indonesia. (2008). *Undang – Undang No 18 Tahun 2008 tentang Pengelolaan Sampah*.