

# PENERAPAN ALGORITMA *GREEDY* UNTUK PENENTUAN JALUR DISTRIBUSI BANTUAN PASCA BENCANA ALAM

Filman Ferdian

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Jalan Ganesha 10 Bandung  
if17091@students.if.itb.ac.id

## ABSTRAK

Distribusi bantuan pasca bencana alam merupakan salah satu pertimbangan penting yang harus diambil setelah terjadi suatu bencana alam. Hal ini menjadi penting karena korban dari suatu bencana alam tentu mengalami trauma dan juga membutuhkan bantuan secara cepat berupa makanan, tenda dan obat-obatan. Kecepatan dalam mendistribusikan bantuan menimbulkan tantangan untuk menentukan jalur yang efisien. Tantangan pada kasus ini dilihat sebagai suatu masalah perumusan strategi algoritma untuk menyelesaikan masalah ini. Proses perumusan tersebut akan melalui dua tahap antara lain, pemodelan masalah, penentuan jalur dengan suatu algoritma *greedy*. Pemodelan masalah ini erat kaitannya dengan graf yang memiliki simpul dan sisi yang memiliki bobot atau nilai. Pada makalah ini akan digambarkan metode pemodelan desa menjadi simpul, jalur antar desa menjadi sisi beserta pemberian bobot pada komponen-komponen tersebut. Setelah memiliki gambaran, maka pendekatan algoritma *greedy* diterapkan pada kasus tersebut untuk menentukan jalur yang harus ditempuh untuk melalui setiap simpul dengan mempertimbangkan bobotnya. Permasalahan ini merupakan pendekatan terhadap kasus penentuan lintasan hamilton dengan kriteria bobot berdasarkan graf masalah yang ada.

**Kata kunci:** algoritma *greedy*, jalur distribusi, lintasan hamilton.

## 1. PENDAHULUAN

Gempa dan banjir merupakan dua kasus bencana alam yang cukup sering terjadi akhir-akhir ini di Indonesia. Di balik penanganan kasus gempa dan banjir, ternyata ada suatu fenomena yang cukup dekat dengan bidang informatika. Kasus itu adalah distribusi bantuan pasca gempa. Permasalahan jalur distribusi terletak pada kondisi posko atau desa yang bisa didekati sebagai simpul

dari suatu graf dan jalur yang bisa didekati sebagai sisi dari suatu graf. Simpul dan sisi pun memiliki bobot yang dapat ditentukan berdasarkan beragam kondisi yang ada di desa atau di jalur bencana alam. Distribusi pasca gempa yang ada selama ini dirasa belum memiliki rumusan yang terstruktur sehingga perlu digagas suatu rumusan yang lebih terstruktur khususnya dalam penentuan jalur distribusi dengan pendekatan ilmiah strategi algoritma.

Algoritma *greedy* dipilih sebagai pendekatan pada masalah ini karena kasus jalur yang memiliki kedekatan karakter seperti *Travelling Salesman Problem* belum memiliki solusi yang non polinomial sehingga diambil kesimpulan bahwa pada kasus jalur distribusi ini pun belum bisa dirumuskan algoritma paling efisien. Algoritma *Greedy* pun sering digunakan sebagai pendekatan untuk penyelesaian kasus yang mirip dengan kasus ini. Oleh karena itu, algoritma *greedy* dirasa merupakan pendekatan yang cukup baik untuk menyelesaikan kasus ini meskipun tidak memberikan hasil yang optimal.

Makalah ini disusun untuk memberikan sebuah kontribusi terhadap penanganan gempa khususnya daam hal penentuan jalur distribusi bantuan pasca bencana alam secara efisien karena kasus bencana alam dapat menimbulkan trauma dan penderitaan fisik maupun mental pada korban yang harus ditangani dengan cepat sehingga perlu ditentukan strategi distribusi yang efisien. Selain itu, makalah ini diharapkan mampu memberi inspirasi pada orang lain untuk mampu menyusun algoritma yang menghasilkan jalur lebih efisien dengan pendekatan lain dibandingkan algoritma *greedy* atau mampu menggunakan strategi algoritma untuk menyelesaikan aspek lain, khususnya dalam kasus penanganan bencana alam. Secara umum, tinjauan ini juga bisa memberikan referensi dalam memperhitungkan penyelesaian lintasan hamilton paling efisien pada graf berbobot sisi dan simpul.

Pada makalah ini fokus permasalahan terletak pada penentuan metode penentuan jalur distribusi dengan menerapkan pendekatan algoritma *greedy* untuk

menyelesaikan permasalahan penentuan jalur dsitribusi bantuan. Pada makalah ini terdapat beberapa pembatasan variabel yang terlibat pada kasus ini dan dijabarkan pada bagian analisis.

## 2. LANDASAN TEORI

Landasan teori berisi teori-teori yang ada terkait makalah ini. Masalah yang dibahas pada makalah ini membutuhkan tinjauan teori terkait kondisi-kondisi yang ada pasca suatu bencana alam, graf berbobot yang memiliki bobot pada sisi dan simpul serta tinjauan algoritma *greedy* secara umum. Setelah paparan umum, akan dijabarkan penarikan kedekatan antara teori dengan masalah yang ada.

### 2.1. PASCA BENCANA ALAM

Kondisi pasca bencana alam adalah kondisi yang ada setelah terjadinya suatu bencana alam. Hal-hal yang terdapat pada penanganan bencana alam setelah bencana alam antara lain pendistribusian bantuan, *trauma healing*, dan rekonstruksi kehidupan secara struktural setelah suatu bencana terjadi.

Bencana alam yang terjadi dapat menggoyongkan banyak aspek dalam tata kehidupan sosial di masyarakat khususnya ekonomi sehingga pemulihan ekonomi merupakan salah satu agenda penting setelah terjadi suatu bencana. Pemulihan ekonomi sendiri umumnya dilakukan dengan memberdayakan kembali pemerintahan di daerah dan mulai memberikan stimulus-stimulus untuk memutar roda ekonomi. Pembangunan sarana dan infrastruktur pun juga menjadi suatu agenda penting. Namun, ada kondisi lain setelah bencana alam yang harus dilakukan dengan cepat yaitu distribusi bantuan.

Suatu bencana alam, yang merusak infrastruktur menyebabkan sebagian masyarakat kehilangan tempat tinggal dan kehilangan kemampuan ekonomi secara sesaat. Sebagian besar korban bencana akan berada di suatu tenda atau posko bencana dan untuk menyambung kehidupan di sana tentu juga dibutuhkan bantuan pangan dan pakaian yang perlu disalurkan agar masyarakat dapat hidup dengan cukup baik sebelum usaha rekonstruksi dan rehabilitasi dilakukan.

Pada makalah ini, kondisi yang ada pada posko dan jalur akan menjadi referensi yang cukup berguna bagi tahapan analisis dari solusi permasalahan penentuan jalur distribusi bantuan dengan pendekatan algoritma *greedy*.

### 2.2. GRAF BERBOBOT

Sebuah graf  $G = (V, E)$  disebut sebuah graf berbobot (*weight graph*), apabila terdapat sebuah fungsi bobot bernilai real  $W$  pada himpunan  $E$ ,

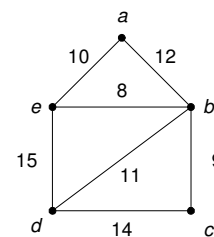
$$W : E \rightarrow \mathbb{R},$$

nilai  $W(e)$  disebut bobot untuk sisi  $e$ ,  $\forall e \in E$ . Graf berbobot tersebut dinyatakan pula sebagai  $G = (V, E, W)$ .

Graf berbobot  $G = (V, E, W)$  dapat menyatakan

- suatu sistem jaringan komputer, di mana
  - $V$  = himpunan komputer
  - $E$  = himpunan jalur komunikasi langsung antar dua komputer
  - $W$  = fungsi bernilai real pada  $E$  yang menyatakan jarak atau ongkos atau waktu

Dengan adanya bobot pada setiap sisi, maka timbul persoalan menentukan lintasan terpendek (*shortest path*) antara dua verteks dalam sebuah graf berbobot. Lintasan terpendek dapat berarti jalur termurah atau terdekat atau tercepat tergantung pada makna dari fungsi bobot yang bersangkutan.



Gambar 1. Contoh graf berbobot

Dari landasan teori yang ada belum ditemukan adanya graf yang memiliki bobot pada simpul. Namun, pada pemodelan kasus ini jikaditinjau dengan pendekatan teori graf maka dibutuhkan suatu bobot pada simpul. Pada makalah ini akan tetap digunakan graf dengan bobot pada sisi dan juga pada simpul karena pendekatan graf adalah pendekatan yang paling mudah untuk memodelkan kondisi dari masalah yang ada.

### 2.3. ALGORITMA GREEDY

Algoritma *Greedy* merupakan algoritma yang erat kaitannya dengan persoalan optimisasi. Persoalan optimasi adalah persoalan yang menuntut pencarian solusi optimum. Solusi optimum adalah solusi yang bernilai minimum atau maksimum dari sekumpulan alternatif solusi yang mungkin. Pada kasus optimasi terdapat elemen persoalan seperti:

1. kendala
2. fungsi objektif atau fungsi optimasi

Solusi yang memenuhi semua kendala disebut solusi layak. Solusi layak yang mengoptimalkan fungsi optimasi disebut solusi optimum.

Algoritma *greedy* merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Prinsip *greedy* adalah: “*take what you can get now!*”. Algoritma *greedy* membentuk solusi langkah per langkah. Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya. Pendekatan yang digunakan di dalam algoritma *greedy* adalah membuat pilihan yang “tampaknya” memberikan perolehan terbaik, yaitu dengan membuat pilihan optimum lokal pada setiap langkah dengan harapan bahwa sisanya mengarah ke solusi optimum global.

Algoritma *greedy* disusun oleh elemen-elemen berikut:

1. Himpunan kandidat, berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi, berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
3. Fungsi seleksi, memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
4. Fungsi kelayakan, memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.
5. Fungsi obyektif, yaitu fungsi yang memaksimalkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain).

Pseudo-code algoritma *greedy* adalah sebagai berikut:

```
procedure greedy (input C:  
himpunan_kandidat, output S:  
himpunan_solusi)
```

Deklarasi:  
X: kandidat

Algoritma:  
S ← {}

```
while (belum SOLUSI(S)) and (C ≠  
{})  
do  
  x ← SELEKSI(C)  
  C ← C - {x}  
  if LAYAK(S ∪ {x}) then  
    S ← S ∪ {x}  
  endif  
endwhile
```

### 3. ANALISIS

Bagian analisis adalah proses perumusan solusi berdasarkan tinjauan teori yang ada. Pada bagian ini, akan dibahas pemodelan kondisi-kondisi pasca gempa khususnya yang terkait dengan distribusi bantuan pasca bencana alam. Setelah dipatikan sebuah graf berbobot, akan dilakukan proses penerapan algoritma *greedy* sampai dapat ditentukan jalur-jalur yang harus dilewati.

Berdasarkan literatur yang ada dapat diambil kesimpulan bahwa ketika terjadi suatu bencana alam maka kita akan melakukan distribusi bantuan melalui berbagai titik yang mampu dicapai dengan cepat. Pada makalah ini, pembahasan dibatasi pada pembahasan penentuan suatu jalur. Sedangkan, penentuan daerah pembagian tidak diatur pada kasus ini. Pada makalah ini akan dijabarkan cara penentuan jalur melalui satu titik keberangkatan dengan satu kendaraan atau banyak kendaraan dengan memanfaatkan jalur yang sama. Selain itu ada beberapa simplifikasi yang digunakan antara lain proses penentuan bobot.

Pada kasus jalur distribusi terdapat dua aspek yang perlu dipertimbangkan yaitu kondisi posko atau RW atau desa sebagai basis untuk pembagian kepada masyarakat dan jalur yang ditempuh dari posko menuju posko lain. Pada posko, yang dapat berupa perwakilan desa atau RW, hal-hal yang mempengaruhi antara lain, jumlah penduduk, tingkat kerusakan, jumlah korban dan jumlah kebutuhan bantuan. Sedangkan pada jalur, hal-hal yang mempengaruhi antara lain jarak antar posko, keberadaan infrastruktur dan waktu tempuh yang dihabiskan dari satu posko ke posko lainnya yang terletak pada suatu basis masyarakat seperti kantor pemerintahan desa atau RW.

#### 3.1. PEMODELAN GRAF

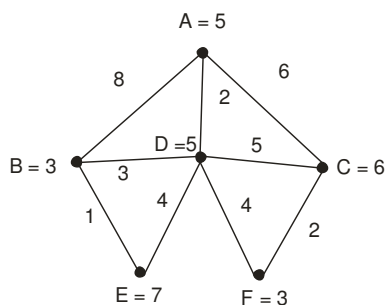
Pemodelan graf dilakukan untuk mengangkat masalah yang ada menjadi suatu graf yang memiliki bobot pada sisi dan simpul agar bisa dievaluasi dengan pendekatan algoritma *greedy*.

Kasus menarik yang perlu dicermati pada pemodelan graf adalah menentukan bobot dari simpul dan sisi. Sebelum masuk pada pembahasan tersebut akan diambil beberapa kesepakatan antara lain simpul merupakan desa yang dianggap sebagai posko pembagian bantuan. Sisi merupakan jalur yang menghubungkan antar simpul.

Simpul merupakan representasi dari suatu desa. Desa tersebut memiliki komponen penilaian seperti jumlah penduduk, dampak kerusakan oleh bencana, ketersediaan cadangan makanan, tenda dan obat-obatan serta kondisi pemerintahan di desa tersebut. Bobot dari komponen yang ada akan menentukan seberapa pentingnya desa tersebut untuk disalurkan bantuan. Desa dengan bobot yang paling membutuhkan bantuan akan diprioritaskan pada bagian awal jalur distribusi. Semakin besar bobot maka semakin penting kondisi di desa tersebut untuk dibantu. Pada pengerjaan makalah ini, bobot dari simpul akan berada pada kisaran 0-100 begitu juga bobot dari sisi.

Sisi merupakan representasi dari suatu jalur yang menghubungkan dua desa. Jalur tersebut memiliki komponen penilaian seperti jarak antara desa dan kondisi dari jalur tersebut. Bobot dari komponen yang ada akan menentukan seberapa mudahnya jalur tersebut untuk dilewati. Jalur dengan bobot yang paling mudah akan diprioritaskan untuk dilalui sebagai jalur distribusi. Semakin besar bobot maka semakin sulit kondisi dari suatu jalur untuk dilalui.

Berdasarkan hasil pembobotan yang ada diharapkan bahwa mekanisme penentuan bobot pada sisi dan simpul proposional sehingga jika bobot ditukarkan antar suatu sisi dan simpul maka nilainya memiliki tingkat kualitatif yang sama. Misalnya, jika simpul A memiliki bobot 4 dan sisi X memiliki bobot 4. Jika bobot pada sisi melambangkan jarak dan bobot pada simpul melambangkan jumlah penduduk maka perbandingan setiap 1 orang dengan  $\frac{1}{4}$  jarak dari sisi X harus sama secara kualitatif. Penilaian kualitatif tersebut tidak dibahas pada makalah ini.



Gambar 2. Contoh Graf berbobot pada simpul dan sisi

Gambar diatas adalah graf yang akan digunakan untuk penerapan algoritma *greedy*.

### 3.2. PENERAPAN ALGORITMA

Penerapan algoritma pada kasus ini akan langsung ditunjukkan dengan algoritma umumnya dan contoh pada graf yang ada pada gambar-2 secara singkat.

Algoritma ini, menerima input suatu matriks yang berisi bobot dari sisi. Lalu, kita juga membutuhkan 2 array yang berisi bobot dari simpul dan jarak dari simpul asal ke simpul lain. Array simpul merupakan bagian dari input sedangkan array bobot sisi diciptakan pada saat algoritma dijalankan. Hasil akhir dari algoritma ini adalah array yang berisi tahapan-tahapan jalur akhir. Selama proses berlangsung perlu diingat pula jalur terdekat antar setiap simpul dan juga array antara menyimpan bobot total dari suatu simpul.

Algoritma untuk menentukan jalur adalah sebagai berikut:

1. Jika sudah tidak ada simpul yang belum dikunjungi maka keluar. Jika belum lakukan tahap kedua.
2. Dari titik keberangkatan, lakukan algoritma dijkstra. Hasil bobot algoritma dijkstra disimpan di array bobot sisi yang menyimpan bobot dari total sisi yang harus dilewati untuk menuju suatu simpul.
3. Selisishkan bobot dari array simpul dan array bobot sisi untuk setiap simpul dengan memanfaatkan array antara.
4. Pilih bobot dengan nilai tertinggi masukan simpul tersebut ke array output. Ulangi langkah 1.

Pada algoritma ini, kita akan memanfaatkan algoritma dijkstra yang merupakan salah satu bentuk algoritma *greedy* untuk menghasilkan lintasan terpendek dari simpul asal ke simpul lain. Bobot dari lintasan terpendek itu akan menjadi pertimbangan dalam menentukan jalur distribusi. Algoritma dijkstra akan memiliki masukan suatu simpul dan menghasilkan array bobot sisi.

Berikut pseudo-code dari algoritma penentuan jalur dengan titik awal A:

```

Procedure JalurDistribusi
(input MCost: matriks, input SCost:
array, input nHop : integer, output
Jalur: matriks)

arrSisi : array {array bobot sisi}

```

```
arrSimpul : array {array bobot simpul}
arrAntara : array {array bobot sisi}
```

```
Inisialisasi(Jalur)
{set matriks menjadi 0}
```

```
Dijkstra(A, arrSisi)
{melakukan dijkstra pada simpul A}
```

```
i iterate [1..nHop]
  SelisihArray(arrSisi, arrSimpul, arr
  Antara)
  {Menyelisihkan array sisi dan
  simpul menyimpan hasilnya ke array
  antara}
```

```
MaxBobot(arrAntara, x)
{Mencari nilai max dari arrAntara
dan mengembalikan nilai nya di x}
```

```
MasukkanSimpul(Jalur, x)
{Memasukkan simpul x ke array
jalur beserta pathnya untuk indeks
ke 0. Indeks ke 1 mengeset status
simpul pemberhentian atau bukan}
```

```
Dijkstra(x, arrSisi)
```

Sekarang, digambarkan contoh penyelesaian suatu masalah yang telah dimodelkan dengan gambar 2 dan dimulai dari titik A.

Pada titik A, kita melakukan algoritma dijkstra dan menentukan bobot yang paling rendah dan hasil dijkstra menunjukkan array sisi sebagai berikut:

B	C	D	E	F
5	6	2	6	6

Gambar 3. Array sisi dari A

Dengan kondisi simpul seperti yang ada di gambar maka didapatkan array antara sebagai berikut:

B	C	D	E	F
-2	0	3	1	-3

Gambar 4. Array antara dari A

Dari matriks antara yang dihasilkan maka dapat dilihat bahwa jalur pertama akan menuju D terlebih dahulu. D dan jalur menuju D disimpan ke matriks jalur yang menyimpan jalur-jalur yang harus dilewati. Di sini kita dapat melihat bahwa meskipun bobot sisi dari A ke E dan ke F sama tetapi dengan kondisi simpul yang berbeda, kita dapat melihat bahwa simpul E harus didahulukan.

Sekarang, kita akan melakukan dijkstra pada simpul D dan menghasilkan array sebagai berikut.

A	B	C	E	F
~	3	5	4	4

Gambar 5. Array sisi dari D

Dengan kondisi simpul seperti yang ada di gambar maka didapatkan array antara sebagai berikut:

A	B	C	E	F
- (~)	0	1	3	-1

Gambar 6. Array antara dari D

Berdasarkan kondisi ini diputuskan bahwa simpul yang akan dikunjungi berikutnya adalah E.

Sekarang, kita akan melakukan dijkstra pada simpul E dan menghasilkan array sebagai berikut.

A	B	C	D	F
~	1	9	~	8

Gambar 7. Array sisi dari E

Dengan kondisi simpul seperti yang ada di gambar maka didapatkan array antara sebagai berikut:

A	B	C	E	F
- (~)	2	-3	- (~)	-5

Gambar 8. Array antara dari E

Berdasarkan kondisi ini diputuskan bahwa simpul yang akan dikunjungi berikutnya adalah B.

Proses ini dilakukan sampai semua simpul telah dilalui dan akan menghasilkan matriks yang harus dikunjungi sebagai berikut dengan tujuan:

Simpul	D	E	B	F	C
Via - 1	D	E	B	D	C
Via - 2				F	

Gambar 9. Matriks Jalur

Maka jalur yang harus dilewati adalah sebagai berikut:

A - D - E - B - D - F - C

#### 4. KESIMPULAN

Berdasarkan proses yang telah dijabarkan maka dapat diambil beberapa kesimpulan antara lain:

1. Strategi algoritma perlu digunakan sebagai pendekatan untuk menentukan jalur distribusi banuan pasca suatu bencana alam.

2. Graf berbobot yang digunakan pada pemodelan jalur distribusi adalah graf yang memiliki bobot pada sisi dan simpul.
3. Pendekatan algoritma *greedy* digunakan dengan mengambil kemungkinan terbaik pada setiap pemilihan. Pada setiap pemilihan, akan diterapkan algoritma dijkstra untuk menentukan jarak antar simpul saat ini dengan simpul lain. Penentuan keputusan dilakukan dengan mempertimbangkan jumlah bobot dari simpul yang akan dituju ditambah nilai dari simpul tersebut.

Selain itu, terdapat beberapa saran yang bisa diberikan antara lain:

1. Pembahasan tentang pembagian jalur dapat disusun dengan pendekatan strategi algoritma sehingga jika pembagian dapat dilakukan secara efisien maka efisiensi proses distribusi bencana alam pun akan meningkat.
2. Pada kasus keberangkatan dari satu titik dengan dua kendaraan, pembagian dapat dilakukan dengan cara satu kendaraan mengikuti jalur yang sudah ditentukan sedangkan mobil yang lain melalui jalur yang berkebalikan dengan jalur dari mobil pertama.

Sesuai tujuan dari penulisan makalah ini, makadiharapkan proses penentuan solusi dari strategi dari distribusi bantuan bencana alam dapat dilanjutkan dengan melengkapi hal-hal berikut seperti: kajian tentang variabel yang terlibat untuk penentuan bobot, kajian tentang metode pembagian jalur dengan banyak titik awal serta pemanfaatan pendekatan algoritma lain yang dapat meningkatkan efisiensi.

## REFERENSI

- [1] Munir, Rinaldi. 2004. Strategi Algoritma. Bandung.
- [2] Munir, Rinaldi. 2005. Matematika Diskrit. Bandung: Penerbit Informatika.
- [3] Humas Bappenas. 2009. Bappenas lakukan penilaian pasca bencana alam. [www.bappenas.go.id/node/116/2077/bappenas-lakukan-penilaian-pasca-bencana-alam](http://www.bappenas.go.id/node/116/2077/bappenas-lakukan-penilaian-pasca-bencana-alam). Waktu akses: 2 Januari 2009 pukul 12.00
- [4] Sigit Iko Sugondo. 2009. Perspective. [www.actforhumanity.or.id/perspective](http://www.actforhumanity.or.id/perspective). Waktu akses: 2 Januari 2009 pukul 12.00