

# Penerapan Algoritma Greedy dalam Mengurus Restoran di Permainan Diner Dash

Felicia Gojali 13518101

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
13518101@std.stei.itb.ac.id

**Abstract**—Algoritma adalah logika, metode, dan tahapan “urutan” sistematis yang digunakan untuk memecahkan suatu permasalahan. Dalam makalah ini, salah satu dari banyak permasalahan yang akan dibahas adalah sebuah permainan yang sering dimainkan dahulu. *Diner Dash* merupakan sebuah permainan dimana pemain menjadi seorang pelayan yang bersusah payah mengurus restoran miliknya. Permasalahan permainan ini akan diselesaikan dengan algoritma Greedy.

**Keywords**—algoritma greedy; permasalahan; *Diner Dash*

## I. PENDAHULUAN

Algoritma adalah suatu metode khusus yang tepat dan terdiri dari serangkaian langkah yang terstruktur dan dituliskan secara matematis yang akan dikerjakan untuk menyelesaikan suatu masalah dengan bantuan komputer. Jadi berdasarkan definisi ini, dapat dikatakan bahwa algoritma merupakan langkah penyelesaian suatu masalah yang menghasilkan solusi dalam bentuk program komputer. Namun penting diketahui bahwa algoritma tidaklah tergantung oleh suatu bahasa pemrograman tertentu, artinya suatu algoritma harus dapat diwujudkan oleh bahasa pemrograman komputer apapun.

Terdapat berbagai jenis algoritma, yaitu Algoritma *BruteForce*, Algoritma *Greedy*, Algoritma *Divide and Conquer*, *Dynamic Programming*, dan lain-lain.

Algoritma dapat disajikan dalam dua bentuk yaitu dalam bentuk tulisan/bahasa dan dalam bentuk gambar. Penyajian algoritma dalam bentuk tulisan haruslah menggunakan bahasa yang dapat dimengerti manusia dalam menyajikan langkah-langkah algoritma. Penyajian algoritma dalam bentuk tulisan juga dapat dilakukan menggunakan pseudocode.

Algoritma memiliki beberapa fungsi, yaitu algoritma bisa digunakan untuk memecahkan program yang rumit, bisa menjadikan program yang besar menjadi program yang lebih sederhana, bisa digunakan secara berulang atau lebih dari satu kali penggunaan, dan bisa memecahkan masalah dengan logika dan urutan. Dapat dilihat dari fungsi-fungsi yang disebutkan tadi bahwa intinya algoritma dapat digunakan untuk memecahkan suatu masalah dan dalam makalah ini

masalah yang akan dipecahkan berupa permainan yang disebut dengan *Diner Dash*.



Gambar 1. *Diner Dash*

*Diner Dash* adalah sebuah permainan berbasis strategi dan manajemen waktu yang sempat populer pada tahun 2000-an. Permainan ini memiliki banyak *stages* atau *levels* yang semakin lama akan semakin meningkat kesulitannya. Namun, permainan ini juga memberikan banyak fitur tambahan seiring meningkatnya kesulitan permainannya yang dapat memudahkan pemain untuk memecahkan masalah di tiap *stage*.

Dalam makalah ini, algoritma yang akan digunakan untuk memecahkan masalah salah satu *stage* yang ada dalam permainan *Diner Dash* adalah Algoritma Greedy. Algoritma tersebut akan membantu pemain mengatur strategi dan meningkatkan kemampuan untuk dapat melakukan manajemen waktu dalam permainan tersebut.

## II. LANDASAN TEORI

### A. Algoritma *Greedy*

Algoritma greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Greedy sendiri diambil dari bahasa Inggris yang artinya rakus, tamak atau

serakah .Prinsip algoritma greedy adalah: “take what you can get now!”

Algoritma *greedy* membentuk solusi langkah per langkah dan pada setiap langkah ada banyak hal yang perlu dievaluasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan.

Pada setiap langkah, kita membuat pilihan optimum lokal dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global. Apabila kita melihat dari definisi dan pengertian algoritma *greedy*, maka bisa kita simpulkan bahwa kelebihan dari algoritma *greedy* adalah cepat dalam bertindak alias fast response. Apabila anda membutuhkan penyelesaian masalah secara instan dan juga cepat, algoritma *greedy* adalah salah satu metode yang tepat. Algoritma *greedy* tidak membutuhkan waktu lama untuk memikirkan opsi – opsi lain yang bisa dilakukan, serta tidak perlu mempertimbangkan baik buruk serta konsekuensi dari apa yang diputuskan.

Meskipun demikian, algoritma *greedy* juga memiliki beberapa kelemahan, yaitu walaupun merupakan bentuk pemikiran pemecahan masalah secara logis yang cepat, namun algoritma greedy ini memiliki kelemahan berupa hasil akhir yang tidak sebaik algoritma *bruteforce*. Hal ini tentu saja disebabkan karena pemilihan opsi yang ditiadakan, sehingga dapat *negative* ataupun konsekuensi dari pemilihan keputusan tersebut tidak dapat dipertanggungjawabkan secara penuh. Algoritma ini juga menekankan pada memilih pilihan yang menyebabkan optimum lokal dengan harapan bahwa dapat membawa kita kepada hasil optimum global. Namun, pada kenyataannya, hasil optimum lokal tidak dapat memastikan akan membawa kepada hasil optimum global, sehingga algoritma ini tidak dapat selalu memberikan hasil yang optimal secara 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 (constraints) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

## 5. Fungsi objektif

Fungsi yang memaksimumkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain

```
function greedy(input C: himpunan_kandidat) → himpunan_kandidat
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy
Masukan: himpunan_kandidat C
Keluaran: himpunan solusi yang bertipe himpunan_kandidat
}
Deklarasi
x : kandidat
S : himpunan_kandidat
Algoritma:
S ← {} { inisialisasi S dengan kosong }
while (not SOLUSI(S) and (C ≠ {})) do
x ← SELEKSI(C) { pilih sebuah kandidat dari C }
C ← C - {x} { elemen himpunan kandidat berkurang satu }
if LAYAK(S ∪ {x}) then
S ← S ∪ {x}
endif
endwhile
{ SOLUSI(S) or C = {} }
if SOLUSI(S) then
return S
else
write('tidak ada solusi')
endif
```

## B. Permainan *Diner Dash*

*Diner Dash* adalah permainan video strategi dan manajemen waktu yang awalnya dikembangkan oleh studio pengembangan permainan Amerika Gamelab yang berbasis di New York City dan diterbitkan oleh PlayFirst yang berbasis di San Francisco. Sekarang dimiliki dan diterbitkan oleh PlayFirst. Ini adalah salah satu game yang dapat diunduh paling laris sepanjang masa, tersedia dalam berbagai platform seperti PC, Mac, konsol, dan seluler.



Gambar 2 . Cover Art *Diner Dash*

Dalam permainan ini, pemain akan memainkan sebuah karakter bernama Flo, di mana Flo adalah seorang pekerja keras di sebuah perusahaan pasar saham besar di Dinertown.

Dia lelah melakukan semua pekerjaan dan merasa tidak terpenuhi. Flo berhenti dari pekerjaannya, berharap dia bisa bekerja di tempat lain. Dia melihat restoran tua yang sudah rusak, yang dia beli. Dia harus menghasilkan cukup uang untuk memperbaiki restoran, dan untuk mendapatkan uang yang banyak, pemain harus menyelesaikan *stage-stage* yang ada dengan tingkat kesulitan yang akan meningkat.

Cara bermain permainan ini adalah dengan memainkan atau menggerakkan Flo. Tujuan dari setiap *stage* dalam permainan ini adalah *keep the customers satisfied* yang artinya pemain tidak boleh membiarkan pelanggan sedih karena terlalu lama menunggu dan akhirnya keluar dari restoran itu.

Pelanggan akan menjadi *satisfied* ketika dia diurus oleh pemain dan tidak menunggu terlalu lama. Pada setiap *stage*, pemain akan memainkan Flo yang akan menjadi seorang pelayan dalam restorannya dan ada beberapa hal yang harus dikerjakan oleh dia, yaitu:

1. Mendudukan pelanggan

Pelanggan yang datang ke restoran Flo akan mengantri dan pemain harus *click and drag* pelanggan-pelanggan tersebut ke meja yang sudah ada di restoran itu. Hal yang perlu diingat adalah jumlah pelanggan yang hadir dalam satu grup dan jumlah kursi pada meja. Pemain harus mencocokkan secara strategis pada meja mana pelanggan tersebut harus ditaruh. Pelanggan juga hanya bisa ditaruh di meja yang sudah bersih dan tidak ada yang menduduki. Selain itu, pelanggan akan datang dengan warna baju berbeda-beda yang akan menempel tempat duduk yang didudukinya. Pemain dapat mencocokkan warna baju pelanggan dan warna bangku yang didudukinya untuk meningkatkan hati pelanggan tersebut. Pemain dapat melakukan aksi ini dengan *mouse* sendiri dan tidak membutuhkan Flo.



Gambar 3. Medudukan pelanggan

2. Mengambil pesanan

Setelah pelanggan duduk di meja makan, mereka akan mengangkat tangan mereka untuk menandakan bahwa mereka ingin memesan makanan. Pemain harus *click* pelanggan agar Flo mendatangi mereka untuk mencatat pesanan makanan mereka, setelah itu pemain juga harus *click spinning ticket station* agar Flo menaruh pesanan di tempat tersebut agar dimasak oleh koki restoran.



Gambar 4. Mengambil pesanan

3. Mengantar pesanan

Setelah pesanan ditaruh di *spinning ticket station*, koki akan memasak pesanan dan menaruh makanan di meja panjang. Pemain harus *click* makanan tersebut dan menaruhnya ke meja pelanggan yang memesannya. Jangan takut untuk tidak tahu pesanan siapakah makanan tersebut karena di setiap makanan akan dicantumkan nomor meja. Aksi ini membutuhkan Flo untuk dilakukan.



Gambar 5. Mengantarkan pesanan

4. Melakukan transaksi

Setelah makan, pelanggan akan mengangkat tangan mereka lagi untuk menandakan bahwa mereka sudah selesai makan dan ingin meminta *bill* pesanan mereka. Pemain harus *me-click* pelanggan tersebut agar Flo menghampiri mereka dan memberikan *bill* untuk dibayar oleh mereka.



Gambar 6. Melakukan transaksi

5. Membersihkan meja

Setelah membayar, pelanggan akan pergi dan meninggalkan meja tempat mereka makan dalam keadaan kotor. Meja harus dibersihkan agar meja tersebut dapat menampung pelanggan selanjutnya, pemain harus *me-click* meja agar Flo akan datang untuk membersihkan meja tersebut dan menaruh piring bekas ke tempatnya dengan *me-click* box yang sudah disediakan.



Gambar 6. Membersihkan meja

Satu *stage* permainan akan berakhir ketika semua pelanggan pada hari itu sudah selesai dilayani dan bintang pemain masih ada. Bintang pemain dapat hilang jika ada pelanggan yang pergi karena tidak *satisfied*. Tambahan adalah Flo dapat membawa dua barang sekaligus.

### III. HASIL DAN PEMBAHASAN

#### A. Perancangan Algoritma *Greedy*

Seperti yang dijelaskan sebelumnya, algoritma *greedy* berarti mengambil langkah yang dianggap optimum pada suatu saat tertentu dengan harapan setiap optimum lokal dapat menghasilkan hasil yang optimum secara global.

Langkah-langkah yang dilakukan untuk mengimplementasikan algoritma tersebut dalam permainan *Diner Dash* adalah menentukan elemen-elemen algoritma *greedy*.

1. Himpunan kandidat

Himpunan ini berisi segala hal yang dapat pemain lakukan pada saat tertentu untuk setiap pelanggan yang ada, misalkan mendudukan pelanggan yang mengantri, mengambil pesanan pelanggan, mengantarkan pesanan, melakukan transaksi dan membersihkan meja. Untuk memudahkan pemain, untuk aksi melakukan transaksi dan membersihkan meja akan digabung karena menurut penulis, penggabungan kedua hal tersebut akan memudahkan pemain.

2. Himpunan solusi

Himpunan berisi langkah-langkah dalam himpunan kandidat yang dipilih.

3. Fungsi seleksi

Fungsi untuk memilih solusi dalam himpunan kandidat. Fungsi yang penulis buat untuk permainan ini adalah:

- Jika aksi yang dapat dilakukan dalam himpunan kandidat  $\leq 2$  (dua), maka segera melakukan aksi tersebut.
- Jika ada pelanggan yang hatinya  $\leq 3$  dan sedang dalam keadaan dapat dilayani (seperti dalam himpunan kandidat), maka pemain harus melayani pelanggan tersebut
- Jika ada aksi melakukan transaksi dan membersihkan meja dalam himpunan kandidat, maka lakukan aksi tersebut, dan jika ada aksi yang sama dengan meja yang berbeda  $\geq 2$ , lakukan kedua aksi tersebut sekaligus (membersihkan meja 1 - membersihkan meja 2 - menaruh piring di tempatnya)
- Jika ada aksi mendudukan pelanggan yang mengantri dalam himpunan kandidat dan terdapat tempat duduk yang jumlah kursinya sesuai dengan jumlah pelanggan dalam satu grup (pilih meja dengan jumlah kursi yang paling sedikit yang sesuai serta utamakan kursi yang warnanya sama dengan warna baju pelanggan), maka lakukan aksi tersebut
- Jika ada aksi mengantar makanan dalam himpunan kandidat, maka lakukan aksi tersebut, dan jika ada aksi yang sama dengan

pelanggan yang berbeda  $\geq 2$ , pilih pelanggan yang memiliki jumlah hati terkecil untuk mengantar makanan ke mereka sekaligus ( mengambil makanan 1 - mengambil makanan 2 - mengantar makanan 1 - mengantar makanan 2)

- f. Jika ada aksi mengambil pesanan dari pelanggan dalam himpunan kandidat, maka lakukan aksi tersebut, dan jika ada aksi yang sama dengan pelanggan yang berbeda  $\geq 2$ , pilih dua pelanggan dengan jumlah hati terkecil untuk melakukan aksi mengambil pesanan dari mereka ( ambil pesanan 1 - ambil pesanan 2 - menaruh pesanan di *spinning ticket station* )
- g. Jika tidak ada kondisi yang memenuhi segala di atas, maka pemain diam dan menunggu

#### 4. Fungsi kelayakan

Fungsi kelayakannya adalah semua langkah sudah pasti layak.

#### 5. Fungsi objektif

Fungsi objektif adalah *keep the customers satisfied* dengan melayani mereka.

Dapat diperhatikan bahwa pembentukan kandidat solusi dan penggunaan fungsi seleksi dilakukan ketika pemain dan Flo sudah *free* dan dapat melakukan suatu aksi.

Algoritma *greedy* pun sudah selesai dibuat mengikuti elemen-elemen yang sudah dibuat di atas.

#### B. Hasil Percobaan dan Analisis

Pada bagian ini, penulis akan menjelaskan cara memainkan permainan *Diner Dash* dengan menerapkan algoritma *greedy* yang sudah penulis buat.

Pada awal permainan, dapat dilihat bahwa hanya ada satu pelanggan saja, maka solusi optimal yang dipilih mengikuti fungsi seleksi bagian (a)



Selama berjalannya permainan, ada kasus berikut yang harus dipilih solusi optimalnya



Dapat dilihat dari gambar di atas bahwa ada 3 hal yang dapat dilakukan oleh pemain, yang berarti himpunan kandidatnya adalah mendudukan pelanggan, mengambil pesanan, dan mengantar makanan, di sini pemain harus memilih aksi mana yang akan menghasilkan hasil optimum lokal. Menurut fungsi seleksi yang sudah dibuat, pemain akan mengutamakan untuk mendudukan pelanggan yang datang ke meja yang kosong dan kemudian melanjutkan untuk membentuk himpunan kandidat yang baru.



Kasus selanjutnya dapat dilihat dalam gambar di atas, himpunan kandidatnya adalah mengantar makanan, mengambil pesanan dan melakukan transaksi serta membersihkan meja. Dilihat dari fungsi seleksi yang sudah dibuat, maka aksi yang dipilih adalah melakukan transaksi dan membersihkan meja.

Setelah memainkan *stage* ini, dapat dilihat bahwa pemain dapat menyelesaikan *stage* ini dengan algoritma *greedy*



Dengan mengimplementasikan algoritma *greedy* yang sudah dibuat, fungsi objektif dapat terpenuhi. Namun untuk pengembangan implementasi yang lebih baik dan akurat, pemain dapat digantikan dengan bot yang sudah diatur untuk memilih keputusan dengan algoritma *greedy*. Oleh karena pemain adalah manusia biasa, maka ada beberapa hal yang terlambat karena dibutuhkan waktu untuk berpikir yang tidak akan terjadi apabila pemain adalah sebuah bot yang sudah diprogram sedemikian rupa.

#### IV. KESIMPULAN

Algoritma *greedy* merupakan salah satu algoritma yang sering diimplementasikan dan digunakan untuk memecahkan masalah namun tidak dapat dibantah bahwa algoritma *greedy* tidak dapat menjamin hasil yang dihasilkan merupakan hasil yang optimal. Penulis juga merasa untuk dapat melihat algoritma *greedy* yang diimplementasikan dalam permainan *Diner Dash* lebih baik dapat dikembangkan dengan cara menggantikan pemain dengan bot yang sudah diprogram dengan algoritma *greedy*.

#### V. LINK VIDEO DI YOUTUBE

Untuk pembahasan lebih jelas mengenai penerapan algoritma *Greedy* dalam permainan *Diner Dash*, penulis menyiapkan video berikut <https://youtu.be/erXGaWoGiI0>

#### VI. UCAPAN TERIMA KASIH

Saya berterima kasih kepada Tuhan Yang Maha Esa atas rahmat-Nya, saya dapat menyelesaikan makalah ini, saya juga

berterima kasih kepada orang tua saya karena sudah memberikan dukungan dan semangat kepada saya selama saya mengerjakan makalah ini. Tak lupa juga saya berterima kasih sebesar-besarnya kepada Ibu Dr. Nur Ulfa Maulidevi ST, M.Sc. selaku dosen Strategi Algoritma saya yang sudah membimbing saya dan mengajarkan saya berbagai materi di Strategi Algoritma sehingga saya paham dan dapat mengerjakan makalah ini. Tak lupa juga saya mengucapkan terima kasih kepada penulis-penulis sumber referensi yang telah memberikan saya ilmu untuk dapat menulis makalah ini.

#### REFERENSI

- [1] Tim Penulis Dosen Pendidikan, "Contoh Algoritma"(2020) <https://www.dosenpendidikan.co.id/contoh-algoritma/> diakses pada tanggal 30 April 2020, pukul 16.00
- [2] Rinaldi Munir, Algoritma Greedy (2017) [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Greedy-\(2018\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Greedy-(2018).pdf) diakses pada tanggal 30 April 2020, pukul 17.00
- [3] Wikipedia, Diner Dash (2020) , [https://en.wikipedia.org/wiki/Diner\\_Dash](https://en.wikipedia.org/wiki/Diner_Dash) diakses pada tanggal 30 April 2020, pukul 17.30

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 30 April 2020

Felicia Gojali  
13518101