

Penerapan Algoritma *Greedy* untuk Menempatkan Pelanggan dalam Permainan Video *Diner Dash*

Timotius Kevin Levandi | 13510056
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510056@std.stei.itb.ac.id

Abstrak—Algoritma *greedy* merupakan algoritma yang paling populer dalam menyelesaikan persoalan optimasi. Persoalan optimasi sendiri terdiri dari persoalan dalam mencari solusi paling maksimal atau persoalan dalam mencari solusi paling minimal.

Dalam permainan video *Diner Dash*, tujuan utama yang harus dipenuhi untuk melanjutkan permainan ke tingkat berikutnya adalah memenuhi minimal poin tertentu dalam satu giliran kerja. Poin didapat dari kombinasi dua mekanisme yaitu ketepatan menempatkan pelanggan dan kecepatan melayani pelanggan. Algoritma *greedy* dalam *Diner Dash* dapat dimanfaatkan untuk mencapai tujuan ini (mencari solusi paling maksimal). Pilihan yang ditinjau dalam melakukan *greedy* di setiap langkahnya adalah mekanisme ketepatan menempatkan pelanggan.

Makalah ini membahas pemanfaatan algoritma *greedy* untuk mendapatkan poin terbanyak dalam satu tingkat permainan *Diner Dash* dengan memerhatikan penempatan pelanggan. Aspek yang diperhatikan dalam menempatkan pelanggan adalah keterkaitan jumlah pelanggan dalam satu kelompok dibandingkan dengan jumlah kursi yang tersedia, penggunaan kursi dengan warna yang sama secara berturut-turut, serta pemilihan kelompok pelanggan yang harus dilayani terlebih dahulu berkaitan dengan toleransi tunggu masing-masing kelompok pelanggan.

Kata kunci—algoritma *greedy*, *Diner Dash*, solusi maksimal

I. PENDAHULUAN

Diner Dash adalah permainan video strategi dan pengelolaan waktu yang mengambil ide kehidupan seorang pelayan restoran dalam jam kerja sibuknya.

Diner Dash menggunakan sistem tingkat, yaitu permainan bertambah sulit seiring semakin tingginya tingkat yang dimainkan. Ada beberapa tingkat yang bisa dimainkan. Tingkatan permainan dimulai dari paling bawah dan pemain tidak bisa mengakses tingkat yang lebih tinggi jika belum menyelesaikan tingkat di bawahnya.

Untuk menyelesaikan satu tingkat, pemain perlu mengumpulkan uang dari kegiatan utama permainan ini, yaitu melayani berbagai jenis pelanggan selama satu giliran kerja. Dalam melayani pelanggan, pemain menentukan pilihan-pilihan apa yang perlu dibuat agar

bisa melayani semua pelanggan yang datang ke restoran.

Pilihan-pilihan yang ada dalam permainan aslinya sangat banyak dan kompleks. Contohnya seperti, apakah kita harus mendahulukan pelanggan yang tidak sabar untuk didudukkan terlebih dahulu daripada pelanggan yang lebih sabar. Atau apakah kita memilih untuk membiarkan pelanggan tidak terlalu terpuaskan dengan pelayanan yang ia berikan agar pelanggan tidak meminta tambahan *dessert* dan cepat pergi, sehingga meja dapat segera dipakai oleh pelanggan lain yang masih antri, dan lain-lain.

Hal yang harus sebisa mungkin dihindari dari permainan ini adalah pelanggan yang kabur karena parameter kepuasannya sudah mencapai nol akibat menunggu terlalu lama. Hal ini akan mengakibatkan jumlah total uang yang selama ini sudah didapat dikurangi sebagai kompensasi. Hal ini akan membuat kemungkinan total uang yang dihasilkan dalam satu giliran kerja sulit dan hampir tidak mungkin mencapai sasaran untuk melanjutkan ke tingkat selanjutnya.

Seiring bertambahnya tingkat, keberhasilan melayani semua pelanggan tanpa ada yang kabur menjadi semakin sulit dicapai. Bahkan kadang berhasil melakukannya pun tidak cukup untuk memenuhi sasaran jumlah uang yang diperlukan untuk melanjutkan ke level berikutnya. Hal ini dikarenakan ada beberapa hal lain yang juga sangat berpengaruh terhadap penambahan jumlah uang dari hasil melayani meja tersebut.

Hal yang perlu diperhatikan yang dapat menjadi bonus uang adalah tingkat kepuasan pelanggan pada saat selesai makan dan meninggalkan meja. Tingkat kepuasan pelanggan pada dasarnya menurun semakin lama mereka menunggu, dan naik ketika mereka dilayani. Oleh karena itu pemilihan set meja yang harus ditempati serta pelanggan mana yang harus ditempatkan saat ada suatu meja yang kosong menjadi penting. Setiap jenis pelanggan memiliki tingkat kesabaran yang berbeda-beda dalam menunggu. Pengelolaan yang baik akan menghindari adanya pelanggan yang menunggu terlalu lama dan akhirnya kabur.

Hal lainnya yang dapat menjadi bonus uang adalah penempatan pelanggan sesuai dengan warna kursi yang didudukinya. Pelanggan datang dengan warna baju

bervariasi antara 1-4 warna. Pada saat pelanggan didudukkan di kursi, warna kursi-kursi pada meja tersebut akan berubah sesuai warna pakaian mereka masing-masing, dan tetap seperti itu setelah mereka meninggalkan meja. Mendudukkan pelanggan berikutnya dengan warna pakaian yang sesuai dengan warna kursi akan memulai suatu *combo* yang akan mengalikan jumlah uang yang didapat dari pelanggan tersebut sejumlah *combo* kursi. Menempatkan pelanggan berikutnya dengan warna yang tidak sesuai akan mengembalikan berapa pun nilai *combo* yang didapat menjadi nol lagi yang tentunya sebisa mungkin dihindari.

Algoritma *greedy* dapat dimanfaatkan untuk memutuskan pilihan-pilihan yang harus diambil dalam maksimisasi uang yang dihasilkan, terutama pada kedua aspek yang telah disebutkan di atas.

Secara garis besar, dalam pengelolaan penempatan pelanggan, *greedy* diaplikasikan untuk memilih pelanggan yang akan didudukkan berdasarkan ketersediaan jumlah kursi, toleransi menunggu, dan kesamaan warna pakaian dengan kursi.

Mekanisme permainan yang perlu diperhatikannya meliputi beberapa hal sebagai berikut

- 1.) Pelanggan datang dalam set yang bisa terdiri dari 1-8 orang tiap set nya. Pelanggan menunggu untuk ditempatkan pada set meja kosong yang tersedia. Terdapat batas toleransi menunggu tiap jenis set pelanggan.
- 2.) Restoran terdiri dari beberapa set meja yang bisa terdiri dari 2-8 kursi.
- 3.) Untuk mulai melayani, satu set pelanggan didudukkan pada satu set meja. Set pelanggan hanya dapat didudukkan di set meja dengan jumlah kursi yang kurang dari atau sama dengan jumlah pelanggan dalam set tersebut.
- 4.) Pelanggan menentukan pesanan dengan interval waktu tertentu
- 5.) Pesanan dibuat dalam interval waktu tertentu, pelanggan menunggu
- 6.) Setelah makanan datang dan disajikan, pelanggan memakan makanan dalam interval waktu tertentu
- 7.) Pelanggan selesai makan, piring kotor diambil dan pemain mendapat uang.

II. TEORI DASAR

Algoritma *greedy* merupakan salah satu algoritma penyelesaian masalah dalam ilmu komputer. Algoritma *greedy* membentuk solusi dari tiap langkahnya. Pada tiap langkah terdapat banyak pilihan yang perlu dieksplorasi. Oleh karena itu, di tiap langkah perlu diputuskan untuk mengambil pilihan yang nantinya akan menghasilkan solusi yang paling optimum.

Prinsip algoritma *greedy* adalah solusi optimum lokal (keputusan menentukan pilihan pada setiap langkahnya) akan mengarah ke solusi optimum global. Dengan demikian, keputusan didasarkan oleh kriteria tunggal alih-

alih melakukan analisis global yang mengarah kepada dampak keputusan pada langkah-langkah yang akan datang (“*take what you can get now*”).

Sebagai suatu algoritma penyelesaian masalah, algoritma *greedy* tidak selalu “menyelesaikan masalah” dengan benar. Dalam mencari solusi suatu permasalahan optimasi, *greedy* hanya bisa menjamin solusi teroptimum bila dalam setiap titik keputusan, mengambil pilihan yang paling baik (paling minimal dalam kasus minimisasi dan paling maksimal dalam kasus maksimisasi) akan selalu mengarah ke tujuan akhir. Optimum global belum tentu merupakan solusi optimum (terbaik), tetapi *sub-optimum* atau *pseudo-optimum*. Hal ini dikarenakan tidak semua alternatif solusi akan ditinjau oleh *greedy*, hanya tiap elemen solusi dalam tiap langkah. Selain itu, hanya pemilihan fungsi seleksi pilihan yang tepat di setiap langkahnya yang akan membawa *greedy* menghasilkan solusi optimum.

Beberapa komponen dalam algoritma *greedy* adalah sebagai berikut:

- 1.) Himpunan kandidat: himpunan pilihan dari mana solusi dibuat
- 2.) Fungsi seleksi: memilih kandidat terbaik untuk ditambahkan ke dalam solusi
- 3.) Fungsi layak: digunakan untuk menentukan apakah sebuah kandidat dapat digunakan untuk berkontribusi pada solusi
- 4.) Fungsi objektif: menentukan nilai sebuah solusi
- 5.) Fungsi solusi: diindikasikan saat solusi yang lengkap telah ditemukan

III. ANALISIS DAN PEMBAHASAN

III.1 MEKANISME DASAR PERMAINAN

Berikut adalah beberapa mekanisme dasar permainan Diner Dash. Mekanisme asli permainan Diner Dash banyak dan kompleks. Namun di bawah ini akan dipaparkan mekanisme dasar

- 1.) Pelanggan datang dalam set yang bisa terdiri dari 1-8 orang tiap set nya. Pelanggan menunggu untuk ditempatkan pada set meja kosong yang tersedia. Terdapat batas toleransi menunggu tiap jenis set pelanggan.
- 2.) Restoran terdiri dari beberapa set meja yang bisa terdiri dari 2-8 kursi. Untuk mulai melayani, satu set pelanggan didudukkan pada satu set meja. Set pelanggan hanya dapat didudukkan di set meja dengan jumlah kursi yang kurang dari atau sama dengan jumlah pelanggan dalam set tersebut.
- 3.) Pelanggan menentukan pesanan dengan interval waktu tertentu
- 4.) Pesanan dibuat dalam interval waktu tertentu, pelanggan menunggu

- 5.) Setelah makanan datang dan disajikan, pelanggan memakan makanan dalam interval waktu tertentu
- 6.) Pelanggan selesai makan, piring kotor diambil, pemain mendapat uang, dan set meja kembali tersedia untuk pelanggan berikutnya.

Mekanisme 3-6 adalah inti utama game di mana pemain berkeliling restoran untuk melayani pelanggan dan melibatkan kegiatan mengambil dan mengantar. Mekanisme 3-6 melibatkan kemampuan pemain dalam ketelitian dan kecepatan refleks untuk melihat kebutuhan pelanggan saat itu, apakah pelanggan ingin pesannya diambil, makanannya diantar, bonnya diberikan, atau piringnya dibereskan. Karena ada banyak pelanggan yang harus diperhatikan, kemungkinan kebutuhan suatu pelanggan dalam suatu saat akan tertunda untuk memenuhi kebutuhan pelanggan lainnya. Semakin lama pemain membiarkan pelanggan menunggu, toleransi pelanggan akan menurun yang akan mempengaruhi uang tip akhir yang akan mereka berikan saat mereka meninggalkan meja. Karena mekanisme 3-6 tidak banyak mementingkan optimasi, maka mekanisme 3-6 tidak akan dibahas.

Mekanisme 1 dan 2 merupakan tahap inisiasi sebelum pelanggan dilayani di mana pemain mengambil satu set pelanggan dalam antrian untuk didudukkan pada salah satu set meja yang tersedia. Di sini terdapat dua kegiatan memilih yang akan mempengaruhi mekanisme 3-6 nantinya, yaitu set pelanggan mana yang akan didudukkan, dan pada set meja yang mana.

Kesalahan fatal dalam memilih pelanggan yang sering terjadi di tahap ini adalah mendahulukan mendudukkan pelanggan dengan toleransi tunggu tinggi daripada mereka yang memiliki toleransi tunggu rendah. Keputusan ini akan mengakibatkan habisnya toleransi pelanggan dengan toleransi tunggu rendah sehingga akhirnya mereka kabur dan kita kehilangan banyak uang.

Kesalahan fatal dalam memilih set meja adalah menempatkan set pelanggan dengan jumlah kecil pada set meja yang seharusnya dapat menampung set pelanggan dengan jumlah besar. Kesalahan ini biasanya sering terjadi di tengah giliran kerja di mana set pelanggan dengan jumlah kecil berada di paling depan antrian dan telah menunggu lama, sementara meja yang tersedia hanya yang berkapasitas banyak. Keputusan ini akan mengakibatkan terlantarnya set pelanggan yang berjumlah banyak dalam antrian karena biasanya set meja dengan kapasitas banyak hanya ada satu atau dua saja.

III.2 ALGORITMA GREEDY UNTUK MENENTUKAN PILIHAN

Untuk mempermudah dalam pembahasan, kita akan mengambil salah satu tingkat permainan sebagai contoh kasus. Tingkat yang akan diambil sebagai contoh kasus adalah 1-7. Berikut informasi yang bisa didapat dari tingkat tersebut

Waktu datang (a)	Jumlah tiap set (b)	Jenis (c)	Toleransi tunggu (d)	Warna (e)
00.00	3	Gadis	60	B, b, b
00.08	2	Senior	90	M, m
00.15	1	Gadis	60	B
00.30	2	Senior	90	B, b
00.47	2	Senior	90	B, b
01.02	2	Gadis	60	M, m
01.12	4	Gadis	60	M, b, b, m
01.22	3	Gadis	60	B, b, b
01.47	2	Senior	90	B, m
01.54	2	Gadis	60	M, b
02.54	4	Senior	90	M, m, m, m
02.11	2	Gadis	60	M, m
02.29	3	Senior	90	B, b, b
02.36	3	Senior	90	B, m, b
02.51	2	Senior	90	M, m
02.58	2	Gadis	60	B, m
03.05	3	Gadis	60	B, b, b

Tabel 1 – Inisiasi tabel pelanggan tingkat 1-7

Nomor (f)	Status (g)	Kapasitas (h)	Tersedia berikutnya (i)	Warna (j)
1	Bebas	2	0	-, -
2	Bebas	4	0	-, -, -, -
3	Bebas	2	0	-, -
4	Bebas	2	0	-, -
5	Bebas	4	0	-, -, -, -
6	Bebas	2	0	-, -

Tabel 2 – Inisiasi tabel meja tingkat 1-7

Kedua tabel akan selalu diperiksa seiring waktu. Nilai yang berubah dari Tabel 1 seiring waktu adalah kolom (d). Pada pelanggan yang sedang antri, nilai (d) akan terus turun hingga akhirnya menjadi 0. Nilai yang berubah dari Tabel 2 seiring waktu adalah kolom (g), (i), dan (j). Pada meja yang ditempati pelanggan, (g) berubah jadi 'terpakai', (i) akan berisi waktu hitung mundur sampai meja tersebut menjadi bebas kembali, (j) akan berisi warna sesuai pelanggan yang menempatnya.

Dari gambaran keadaan seperti tabel di atas dapat dibuat dibuat strategi greedy sebagai berikut untuk setiap saat

1. buat variabel akumulator nilai n1, inisiasi dengan 0.
2. masukkan waktu sekarang dalam variabel t.
3. masukkan semua pelanggan dimana $(a) + (d) > t$ ke dalam TabelAntri. (menentukan pelanggan mana yang sedang mengantri dengan membandingkan apakah waktu sekarang masih termasuk dalam toleransi tunggu pelanggan semenjak kedatangannya).
4. Untuk setiap pelanggan P dari TabelAntri, lakukan hal sebagai berikut

- a. Tambahkan nilai n_1 dengan $P(d)$
(semakin kecil nilainya artinya semakin mendesak kebutuhan pelanggan ini untuk segera didudukkan)
 - b. Untuk setiap meja M dengan $P(b) \leq M(h)$ dari Tabel 2, lakukan hal sebagai berikut
 - i. buat variabel akumulator nilai n_2 , inisiasi 0.
 - ii. tambahkan nilai $M(h) - P(b)$ ke dalam n_2 .
(selisih antara kapasitas meja dengan jumlah tiap set pelanggan, semakin kecil semakin bagus, artinya penggunaan meja semakin efektif)
 - iii. tambahkan nilai $M(i)$ ke dalam n_2 .
(semakin kecil nilainya menunjukkan bahwa meja tersebut hampir atau sedang bebas untuk digunakan)
 - iv. bandingkan $M(j)$ dan $P(e)$, tambahkan jumlah warna yang tidak cocok ke dalam n_2
(semakin kecil nilainya menunjukkan bahwa kesamaan kombinasi warna pakaian dan tempat duduk semakin tinggi)
- Cari M dengan n_2 terkecil.
Cari P dengan $(n_1 + n_2)$ terkecil.

III.3 CONTOH PENGAPLIKASIAN

Masih menggunakan contoh permainan pada tingkat 1-7 dan informasi tabel seperti tertera pada subbab III.2, misalkan waktu sekarang dari sejak dimulainya permainan adalah 02.55. Pada saat ini keadaan adalah sebagai berikut



Bagan 1: Permainan tingkat 1-7 pada menit 02.55

Waktu datang (a)	Jumlah tiap set (b)	Jenis (c)	Toleransi tunggu (d)	Warna (e)
00.00	3	Gadis	-	B, b, b
00.08	2	Senior	-	M, m
00.15	1	Gadis	-	B

00.30	2	Senior	-	B, b
00.47	2	Senior	-	B, b
01.02	2	Gadis	-	M, m
01.12	4	Gadis	-	M, b, b, m
01.22	3	Gadis	-	B, b, b
01.47	2	Senior	-	B, m
01.54	2	Gadis	-	M, b
02.54	4	Senior	-	M, m, m, m
02.11	2	Gadis	-	M, m
02.29	3	Senior	-	B, b, b
02.36	3	Senior	75	B, m, b
02.51	2	Senior	90	M, m
02.58	2	Gadis	60	B, m
03.05	3	Gadis	60	B, b, b

Tabel 3 – tabel pelanggan pada 02.55

Nomor (f)	Status (g)	Kapasitas (h)	Tersedia berikutnya (i)	Warna (j)
1	Bebas	2	0	b,b
2	Terpakai	4	15	m,m,m,m
3	Terpakai	2	10	m,b
4	Terpakai	2	10	m,m
5	Terpakai	4	40	-,b,b,b
6	Terpakai	2	5	b,m

Tabel 4 – tabel pelanggan pada 02.55

Berdasarkan strategi yang telah kita tetapkan sebelumnya, penyelesaian penempatan dapat dilakukan sebagai berikut

1. Pelanggan yang masuk TabelAntri adalah pelanggan ke -14 ($02.36 + 75 > 02.55$) dan ke-15 ($02.51 + 90 > 90$)
2. Untuk pelanggan ke-14:
 - a. $n_1 = 0 + P(d) = 75$
 - b. $P(b) = 3$, maka meja dengan $M(h) \geq P(b)$ adalah meja nomor 2 dan 5 yang berkapasitas 4
 - c. $n_{2_2} = 0 + M(h) - P(b) = 0 + 1 = 1$
 $n_{2_5} = 0 + M(h) - P(b) = 0 + 1 = 1$
 - d. $n_{2_2} = 1 + M(i) = 1 + 15 = 16$
 $n_{2_5} = 1 + M(i) = 1 + 40 = 41$
 - e. $n_{2_2} = 16 + 3 = 19$
 $n_{2_5} = 41 + 2 = 43$
 - f. nilai n_2 terkecil adalah untuk $n_{2_2} = 19$
3. Untuk pelanggan ke-15:
 - a. $n_1 = 0 + P(d) = 90$
 - b. $P(b) = 2$, maka meja dengan $M(h) \geq P(b)$ semua meja
 - c. $n_{2_1} = 0 + M(h) - P(b) = 0 + 0 = 0$
 $n_{2_2} = 0 + M(h) - P(b) = 0 + 2 = 2$
 $n_{2_3} = 0 + M(h) - P(b) = 0 + 0 = 0$
 $n_{2_4} = 0 + M(h) - P(b) = 0 + 0 = 0$
 $n_{2_5} = 0 + M(h) - P(b) = 0 + 2 = 2$
 $n_{2_6} = 0 + M(h) - P(b) = 0 + 0 = 0$
 - d. $n_{2_1} = 0 + M(i) = 0 + 0 = 0$
 $n_{2_2} = 2 + M(i) = 2 + 15 = 17$
 $n_{2_3} = 0 + M(i) = 0 + 10 = 10$
 $n_{2_4} = 0 + M(i) = 0 + 10 = 10$

$$n2_5 = 2 + M(i) = 2 + 40 = 42$$

$$n2_6 = 0 + M(i) = 0 + 5 = 5$$

e. $n2_1 = 0 + 2 = 2$

$$n2_2 = 17 + 2 = 19$$

$$n2_3 = 10 + 1 = 11$$

$$n2_4 = 10 + 0 = 10$$

$$n2_5 = 42 + 4 = 46$$

$$n2_6 = 5 + 1 = 6$$

f. nilai $n2$ terkecil adalah untuk $n2_1 = 2$

4. nilai $n1 + n2$ terkecil di antara kedua pelanggan tersebut adalah pelanggan ke-15 = 92

5. Dengan demikian keputusannya, saat ini tempatkan pelanggan ke-15 pada meja 1

IV. KESIMPULAN

Dari pembahasan di atas dapat dihasilkan kesimpulan sebagai berikut

1. Algoritma greedy dapat digunakan pada penyelesaian permasalahan optimasi yang melibatkan banyak kegiatan memilih pada setiap tahapnya.
2. Untuk menentukan strategi greedy yang baik, fungsi seleksi yang dibuat harus dipikirkan baik-baik berdasarkan apa yang perlu didahulukan dan batasan-batasan pemilihan, karena yang membedakan kebaikan hasil suatu algoritma greedy terletak pada pendefinisian fungsi seleksinya.

REFERENSI

- [1] Munir, Rinaldi. *Strategi Algoritma*. 2009. Bandung: Penerbit ITB, halaman 26-32
- [2] _____. "Algorithms/Greedy Algorithms". Sumber http://en.wikibooks.org/wiki/Algorithms/Greedy_Algorithms (Diakses 21 Desember 2012)
- [3] magic knight. "diner Dash: FAQ/Walkthrough". Sumber <http://www.gamefaqs.com/pc/926256-diner-dash/faqs/45187> (Diakses 19 Desember 2012)

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, 21 Desember 2012



Timotius Kevin Levandi | 13510056