

# Pemanfaatan Algoritma Greedy dalam Grinding Cash didalam game Need For Speed ProStreet

Vincent Hasiholan / 13518108  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
wildarms372@gmail.com

**Abstract** – Makalah ini akan membahas implementasi algoritma Greedy pada game tua, khususnya pada game *Racing*. Setiap game *racing* pasti memerlukan uang atau cash untuk dapat mengembangkan atau membeli kendaraan untuk dapat memenangkan sebuah lomba balapan. Tetapi, untuk mendapatkan uang ini memerlukan waktu dan terkadang panjang lintasan dalam balapan tersebut juga dapat mempengaruhi bagaimana pemain harus dapat mengalahkan musuh dalam lintasan tersebut.

**Keywords** – Greedy , game, racing

## I. Pendahuluan

Bagi pecinta game bertema *racing* dan suka *franchise* dari Need for Speed pasti mengenal salah satu game ini, yaitu Need for Speed ProStreet. Game ini dirilis setelah game Need for Speed Carbon pada tahun 2007. Game ini mengenalkan *system car damage* yang dapat pemain merasa kesal karena walau kelihatan kemenangan didepan mata, jika *car damage* mencapai level *totaled*, maka kemenangan itu akan hilang.

Need for Speed Pro Street dirilis of EA dengan cover menggunakan mobil Nissan GT-R. Game ini menggambarkan *legal racing* yang tidak melibatkan kejar – kejaran dengan polisi seperti pada game Need for Speed : Most Wanted dan Need for Speed Carbon. Game ini mengembangkan pengembangan permofance mesin dan tampilan mesin yang dapat

disesuaikan dengan keinginan pemain. Kemudian performa pada mesin tersebut dapat disetel dengan arena yang akan digunakan dalam balapan tersebut.

Untuk mendapatkan semua mobil yang tersedia, pemain harus mengalahkan balapan dengan status balapan yang memberikan promotion. Tetapi, ada beberapa mobil dan upgrade mobil yang tidak bisa didapatkan dengan metode ini. Mereka hanya bisa didapatkan dengan memasukkan kode rahasia untuk mendapatkannya. Salah satu contohnya adalah McLaren F1. Performa yang dapat dibuka dengan memasukkan kode rahasia ini adalah stage 4, yaitu upgrade paling terakhir pada mobil dan upgrade ini sangatlah langka untuk didapatkan dengan metode pemilihan reward pada saat win.

Game ini dapat dimainkan pada PS2, DS, PSP, PS3, Xbox 360, PC, dan Wii. Tetapi, untuk permainan secara online hanya dapat dimainkan selain dari platform Wii dan PS2.



Gambar 1 : Cover Need for Speed Pro Street  
(sumber

[https://www.google.com/search?q=need+for+speed+pro+street&safe=strict&rlz=1C1CHBF\\_enlD873ID873&hl=en&sxsrf=ALeKk00r7pXxHhEY\\_y9JeSkFdkffclx8Q:1587897723517&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiOtv\\_984XpAhUXOisKHTfhD4UQ\\_AUoAXoECD8QAw&biw=1366&bih=663](https://www.google.com/search?q=need+for+speed+pro+street&safe=strict&rlz=1C1CHBF_enlD873ID873&hl=en&sxsrf=ALeKk00r7pXxHhEY_y9JeSkFdkffclx8Q:1587897723517&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiOtv_984XpAhUXOisKHTfhD4UQ_AUoAXoECD8QAw&biw=1366&bih=663))

Game ini memiliki empat mode secara umum, yaitu Grip, Drag, Drift, dan Speed. Keempat mode memiliki *king*, yang memegang status juara dalam setiap mode dan sebuah *king* yang memegang semua status tersebut. Cara untuk membuka peluang untuk melawan setiap *king* pada mode masing – masing adalah mengalahkan *track record* yang ada pada setiap mode dan mencapai jumlah tertentu. Kemudian, stage untuk melawan *king* tersebut akan terbuka.

- Grip



Gambar 2 : Contoh Mobil - Grip  
(Sumber : dokumentasi sendiri)

Grip mode adalah balapan circuit yang memiliki empat mode didalamnya, yaitu *grip*, *grip class*, *time attack*, dan *sector shootout*. *Grip* adalah mode circuit normal dengan menggunakan mobil yang disiapkan oleh pemain dan balapan akan dilakukan dengan lintasan yang sama dari dua sampai empat kali putaran. Selain dari *grip*, mode *grip* yang lain dilakukan selama tiga kali putaran. Untuk *grip class*, delapan mobil balapan akan dibagi kedalam dua tim dan pemain akan ditandingkan oleh pemain yang berada didalam satu tim dan tim disusun dengan melihat kekuatan mobil. *Time Attack*

melihat waktu terbaik setiap putaran. *Sector shootout* melihat waktu terbaik dari setiap segment didalam lintasan circuit yang digunakan.

Bonus score pada tipe balapan ini dipisah menjadi dua, yaitu normal (*grip* dan *grip class*), *time attack*, dan *sector shootout*. Bonus score diperoleh dengan membandingkan total waktu yang diperoleh dengan perkiraan total waktu yang telah disiapkan. Untuk *Time Attack*, bonus score dilihat dari waktu terbaik per lap yang diselisihkan dengan waktu perkiraan untuk balapan tersebut. Bonus score pada *sector shootout* melihat total point yang diperoleh selama melakukan balapan tersebut.

- Drag

Drag adalah balapan dari poin ke poin dengan jarak 0.25 mile atau 0.5 mile. Didalam drag ini juga terdapat mode *wheelie* yang melejit seberapa jauh mobil pemain dapat melakukan *wheelie* dalam jarak 0.25 mile. Didalam balapan drag, bonus score didapat dari selisih waktu terbaik yang telah diset pada balapan dengan waktu terbaik yang diperoleh diri sendiri. Sebelum balapan dimulai, pemain disiapkan terlebih dahulu dengan *burnout* tire untuk menaikkan nilai *grip* untuk launch saat hendak memulai balapan.



Gambar 3 : Contoh Mobil - Drag

(Sumber : dokumentasi sendiri)

- Drift

Drift adalah mode balapan yang melihat poin terbaik yang dapat diperoleh melalui lamanya drift, speed, dan angle pada drift yang dilakukan pemain. Pemenang ditentukan dari nilai terbaik yang diperoleh dalam tiga kali drift. Tetap diperhatikan bahwa terdapat beberapa arena balapan yang dapat membuat mobil menjadi totaled. Bonus score pada mode ini didapat dari total score terbaik yang diperoleh dari tiga kali percobaan drift.



Gambar 4 : Contoh Mobil - Drift

(Sumber : dokumentasi sendiri)

- Speed

Speed adalah mode balapan yang melalui lintasan poin ke poin yang panjang dan pemain harus berhati – hati karena pemain akan melakukan balapan ini akan melalui batas 200 mph. Jika tidak berhati – hati, maka mobil pemain akan totaled dan pemain akan dinyatakan kalah. Didalam mode ini ada mode normal dan top speed run dengan melihat total kecepatan yang dilihat dari setiap poin dalam segment lintasan.



Gambar 5 : Contoh Mobil – Speed

(Sumber : dokumentasi sendiri)

Pada game ini, transisi pada setiap mode balapan dapat diatur. Transisi yang disediakan adalah Automatic, Manual, dan Manual Clutch. Perbedaan antara Manual dan Manual Clutch adalah waktu perpindahan transisi tiap gigi. Manual Clutch tidak melihat waktu perpindahan tersebut dan mobil dapat langsung berlari tanpa jeda walau dalam permainan harus menekan dua tombol yang hamper secara bersamaan.

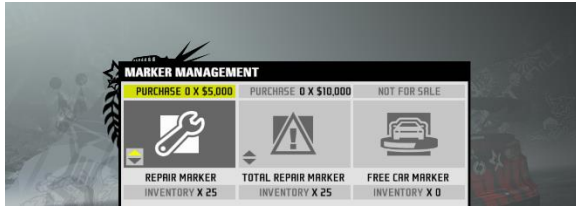
Game ini juga mengenalkan *system car damage* yang dapat memberikan efek pada mobil yang akan digunakan pada balapan. Tingkat damage yang diterima mobil adalah Light (penurunan minim pada performa mesin), Heavy (penurunan drastis pada performa mesin), dan Totaled (dikeluarkan dari balapan / dinyatakan kalah dalam kondisi apapun). Perlu diperhatikan bahwa mobil yang sedang balapan dan mengalami kondisi Totaled dari kondisi normal ke Heavy Damage dapat digunakan untuk balapan dengan menggunakan Total Repair Marker.



Gambar 6 : Mobil yang sudah Totaled

(Sumber : dokumentasi sendiri)

Didalam permainan ini, terdapat tiga buah item yang dapat menolong ketika cash yang dimiliki sedang sedikit, yaitu Repair Marker, Total Repair Marker, dan Free Car Marker. Free Car Marker ini sangat jarang digunakan dikarenakan ketika mendapatkan marker ini, biasanya cash yang sudah dimiliki sudah cukup untuk membeli mobil yang paling mahal beserta pembelian *upgrade* dan pemasangan visual pada mobil tersebut.



Gambar 7 : Marker

(Sumber : Dokumentasi Sendiri)

### 1. Repair Marker

Repair Marker adalah marker yang digunakan untuk membetulkan mobil dari kondisi Light damage atau Heavy damage. Marker ini tidak bisa digunakan untuk kondisi Totaled. Marker ini bisa dibeli seharga \$5000 cash in game atau didapatkan dari Win race reward secara acak. Jumlah maksimal yang dapat dipegang oleh pemain adalah 25.

### 2. Total Repair Marker

Total Repair Marker adalah marker yang hanya bisa digunakan untuk membetulkan mobil dengan kondisi Totaled dan bukan untuk kondisi yang lain. Marker ini bisa dibeli seharga \$10000 cash in game atau didapatkan dari Win race reward secara acak. Jumlah yang dapat dipegang oleh pemain adalah 25.

### 3. Free Car Marker

Free Car Marker adalah marker yang digunakan di Car Lot untuk membeli mobil secara gratis tanpa melihat seberapa tinggi harganya. Marker ini hanya bisa didapatkan dengan keberuntungan dalam memilih win reward pada stage sebelum *last boss* saja.

Selain digunakan untuk membeli Marker, cash digame ini digunakan untuk meng-upgrade mobil secara performa ataupun visual. Hal ini dapat dilakukan di Customization Menu. Customization Menu ini merupakan menu yang menampilkan status mobil yang digunakan pada tipe balapan tertentu.



Gambar 8 : Customization Screen

(Sumber : Dokumentasi sendiri)

Didalam menu terdapat beberapa pilihan yang dapat diberikan untuk meningkatkan performa mobil, yaitu Quick dan Custom Upgrade. Perbedaannya adalah Quick Upgrade memberikan sebuah paket yang sudah berisikan upgrade untuk performa mobil, sedangkan Custom Upgrade adalah pilihan untuk meng-upgrade mobil dengan melihat kondisi cash sendiri dan pilihan ini memberikan pilihan untuk mengubah visual pada mobil.



Gambar 9 : McLaren F1 setelah diupgrade dan diubah visual

(Sumber : dokumentasi sendiri)

Di bawah pilih Custom Upgrade, terdapat pilihan Tune, Dyno Test, Wind Tunnel, dan Update Blueprint. Tune ini berfungsi untuk menyetel performa dari setiap part pada mobil sesuai dengan keinginan sendiri. Dyno Test berfungsi untuk menguji coba bagaimana performa mobil setelah dipasang dan dilakukan tuning pada setiap bagian mesin. Update Blueprint adalah fungsi yang digunakan untuk memasang semua upgrade dan visual pada mobil yang akan digunakan pada balapan.



## II. Dasar Teori

- Algoritma Greedy

Algoritma greedy merupakan salah satu metode penyelesaian masalah yang cukup populer dalam persoalan optimasi. Yang dimaksud persoalan optimasi sendiri ialah pencarian solusi optimum dalam suatu permasalahan.

Ada dua macam optimasi yaitu maksimasi dan minimasi.

### a) Maksimasi

Mengambil suatu langkah yang menyebabkan nilai yang diinginkan maksimum.

### b) Minimasi

Mengambil suatu langkah yang menyebabkan nilai yang diinginkan minimal.

Prinsip greedy adalah mengambil suatu hal permasalahan yang kecil dengan optimum dan berharap untuk mendapat nilai yang optimal. Greedy membentuk solusi dengan cara langkah per langkah (step by step). Dengan prinsip tersebut algoritma greedy selalu mengambil langkah optimal dalam semua langkah.

Algoritma greedy disusun oleh beberapa elemen berikut:

### a) Himpunan kandidat (C).

Himpunan kandidat berisi dengan semua elemen kandidat yang masih mungkin untuk membentuk solusi.

### b) Himpunan solusi(S).

Himpunan solusi berisi dengan beberapa elemen saja yang merupakan nilai optimum dalam suatu permasalahan kecil (Optimum lokal).

### c) Fungsi seleksi

Pemilihan kandidat yang paling memungkinkan dalam mencapai solusi akhir optimal (Optimum Global). Karena solusi berbentuk himpunan, solusi yang sudah dipilih

tidak akan dipilih lagi pada semua langkah berikutnya.

### d) Fungsi kelayakan

Memeriksa apakah kandidat solusi yang telah terpilih sebelumnya dapat memberikan solusi yang layak, yaitu semua kandidat solusi yang sudah terbentuk bersama sama tidak melanggar constraint yang ada pada permasalahan. Jadi kandidat yang tidak layak akan dibuang dari solusi dan tidak pernah dipertimbangkan lagi.

### e) Fungsi objektif

Fungsi yang memaksimalkan atau meminimumkan nilai solusi contohnya: meminimumkan biaya perjalanan, memaksimalkan keuntungan dalam penjualan barang.

```
function greedy (input C: himpunan kandidat) -> himpunan kandidat
f Menemukan solusi dari persoalan optimasi dengan algoritma greedy
Masukan: himpunan kandidat C
Keluaran: himpunan solusi yang ber tipe himpunan kandidat
}
Deklarasi
x : kandidat
S : himpunan kandidat
Algoritma:
S ← {} f inisialisasi S dengan kosong
while (not SOLUSI(S) and (C ≠ {})) do
  x ← SELEKSI(C) f pilih sebuah kandidat dari C
  C ← C - {x} f 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
```

Gambar 10 : Slide kuliah Rinaldi Munir

Pada permasalahan yang diselesaikan oleh algoritma greedy tidak selalu menghasilkan jawaban paling optimal dikarenakan yang diambil secara optimal hanyalah sub permasalahan / suatu langkah saja (Optimum lokal). Jadi penyusunan elemen - elemen dalam penyusunan algoritma greedy sangat menentukan apakah permasalahan itu cocok atau tidak dalam menyelesaikan permasalahan tersebut.

## III. Implementasi

Untuk mendapatkan mobil yang diinginkan (dianggap mobil dan part untuk meng-upgrade mobil sudah terbuka semua), diperlukan cash yang sangat banyak. Kasus ini

akan melihat dari waktu lamanya balapan dan panjang lintasan yang dilewati karena ada kemungkinan lintasannya panjang tapi waktunya kecil diakibatkan tipe lintasan adalah mode Speed.

```

107 private void greedyView(Greedy_Budget[] farming, double time){
108     quickSort_time(farming, farming.length());
109     double useTime = 0.0;
110     int sumCash = 0;
111     double totalCourseLength = 0.0;
112
113     for(int i = 0; i < farming.length(); i++){
114         useTime += farming[i].time;
115         sumCash += farming[i].cash;
116         totalCourseLength += farming[i].courseLength;
117         farming[i].usable = false;
118     }
119
120     for(int a = 0; a < farming.length(); a++){
121         if(farming[a].usable){
122             Console.WriteLine("(+) + " + farming[a].raceType + " Cash = $" + farming[a].cash);
123         }
124         continue;
125     }
126 }
    
```

Gambar 12 : Algoritma Greedy menurut lamanya balapan

(Sumber : dokumentasi sendiri)

Untuk langkah menggunakan algoritma greedy, dilakukan sebagai berikut :

1. *Filtering* dari semua lomba untuk menentukan lomba mana yang akan dilakukan.
2. Memilih apakah ingin melihat dari sisi waktu atau panjang lintasan.
3. Memasukkan n banyaknya balapan yang akan dipilih.
4. Mengurutkan sesuai dengan pandangan yang ingin dilihat pada nomor 2.
5. Mengeluarkan semua lintasan yang menampilkan hasil terbaik yang disarankan.

Source code :

```

128 private void greedyCourseLength(Greedy_Budget[] farming, double courseLength){
129     quickSort_Length(farming, farming.length());
130     double useTime = 0.0;
131     int sumCash = 0;
132     double totalCourseLength = 0.0;
133
134     for(int i = 0; i < farming.length(); i++){
135         useTime += farming[i].time;
136         sumCash += farming[i].cash;
137         totalCourseLength += farming[i].courseLength;
138         farming[i].usable = false;
139     }
140
141     for(int a = 0; a < farming.length(); a++){
142         if(farming[a].usable){
143             Console.WriteLine("(+) + " + farming[a].raceType + " Cash = $" + farming[a].cash);
144         }
145         continue;
146     }
147 }
    
```

Gambar 12 : Algoritma Greedy menurut panjang Lintasan (Sumber : dokumentasi sendiri)

#### IV. Kesimpulan

Algoritma greedy cocok dengan permainan ini dikarenakan pemain dapat melihat dari dua sisi dimana pemain ingin mendapatkan cash dalam waktu singkat atau lintasan singkat dalam permainan Need for Speed Pro Street. Algoritma ini dapat digunakan dalam kehidupan sehari – hari, misalnya penentuan pekerjaan yang memiliki waktu kecil sehingga dapat melakukan banyak pekerjaan yang lain.

#### V. Saran

Dalam permainan ini sebaiknya disesuaikan dengan mobil yang digunakan, dikarenakan mobil yang tidak sesuai mungkin dapat menyebabkan kekalahan karena kalah performa atau salah *setting* sehingga mobil dapat mengalami kondisi Totalled karena pemain kurang mampu dalam membuat *setting* yang sesuai dengan mobil.

#### VI. Referensi

1. [https://en.wikipedia.org/wiki/Need\\_for\\_Speed:\\_ProStreet](https://en.wikipedia.org/wiki/Need_for_Speed:_ProStreet)
2. [https://informatika.stei.itb.ac.id/~rinald.i.munir/Stmik/2019-2020/Algoritma-Greedy-\(2020\).pdf](https://informatika.stei.itb.ac.id/~rinald.i.munir/Stmik/2019-2020/Algoritma-Greedy-(2020).pdf)
3. [https://youtu.be/R\\_LxapT0sEo](https://youtu.be/R_LxapT0sEo) (video)

#### VII. 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.

Jakarta, 26 April 2020

Vincent Hasiholan