

Penerapan Algoritma Greedy pada Permainan “Chain Reaction”

Kevin Basuki / 13516071

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516071@std.stei.itb.ac.id

Abstract—Permainan *Chain Reaction* adalah permainan yang bertipe strategi yang dapat dimainkan di *mobile* dengan jumlah pemain 2 sampai 8 orang untuk sekedar bersenang-senang maupun adu strategi. Strategi dengan menggunakan algoritma *greedy* dapat digunakan untuk permainan ini. Strategi dengan menggunakan algoritma *greedy* dapat mengarahkan kita untuk memenangkan permainan dengan memilih optimum lokal. Pengembangan algoritma untuk strategi permainan ini dapat dilakukan dengan memperhitungkan jumlah pemain yang banyak dan algoritma lain yang sekiranya dapat ataupun lebih dapat mencapai optimum global yaitu memenangkan permainan.

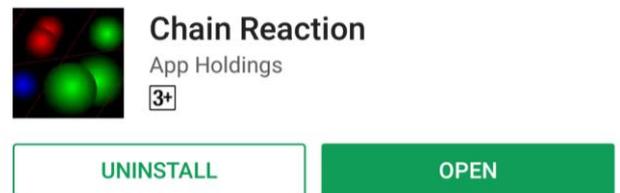
Keywords—*Chain Reaction*; *greedy*; *optimum lokal*; *optimum global*

I. PENDAHULUAN

Pada zaman sekarang ini, permainan (*game*) telah berkembang pesat dan disukai oleh banyak orang. Banyak permainan yang tak terhitung jumlahnya telah muncul dalam berbagai macam *genre* dan dapat dimainkan di berbagai *platform*.

Dari berbagai banyak permainan tersebut, salah satunya adalah *mobile game* yang berjudul “*Chain Reaction*”. Permainan ini adalah permainan ber-*genre arcade strategy* dan dapat dimainkan di *mobile*. Permainan ini telah dirilis pada tanggal 12 Mei 2011 dan di-*update* terakhir pada tanggal 28 September 2017 sebagai versi ke 1.7. Permainan ini mendapat *rating* 3+ (dapat dimainkan oleh manusia berumur 3 tahun keatas) dan mendapat ulasan 4.4 dari 5 bintang. Permainan ini dapat diunduh di *Google Play* dan sudah diunduh lebih dari 5 juta kali.^[1]

Permainan ini cukup disukai banyak orang karena tampilannya yang ringan, ukuran aplikasinya yang kecil (hanya sekitar 4 MB), dan *gameplay*-nya yang seru, menyenangkan, butuh berpikir, dan tidak terduga. *Gameplay* yang membutuhkan 2 sampai 8 orang pemain membuat permainan ini cocok untuk dimainkan bersama teman sekedar untuk bersenang-senang maupun untuk uji kecerdasan ataupun strategi.



Gambar 1. Permainan *Chain Reaction* di *Google Play*

(Sumber:

https://play.google.com/store/apps/details?id=com.BuddyMattEnt.ChainReaction&hl=en_US diakses 13 Mei 2018 pukul 15.00)

Pada makalah ini, akan dibahas cara untuk memenangkan permainan ini dengan menggunakan salah satu strategi yaitu algoritma *Greedy*.

II. DASAR TEORI

A. Algoritma Greedy

Algoritma *greedy* merupakan metode yang paling populer untuk memecahkan persoalan optimasi dengan prinsipnya: “*take what you can get now!*”. Arti kata *greedy* adalah rakus, tamak, roba. Pendekatan yang digunakan di dalam algoritma *greedy* adalah membuat pilihan yang “tampaknya” memberikan perolehan terbaik, yaitu dengan membuat pilihan optimum lokal (*local optimum*) pada setiap langkah dengan harapan bahwa sisanya mengarah ke solusi optimum global (*global optimum*).

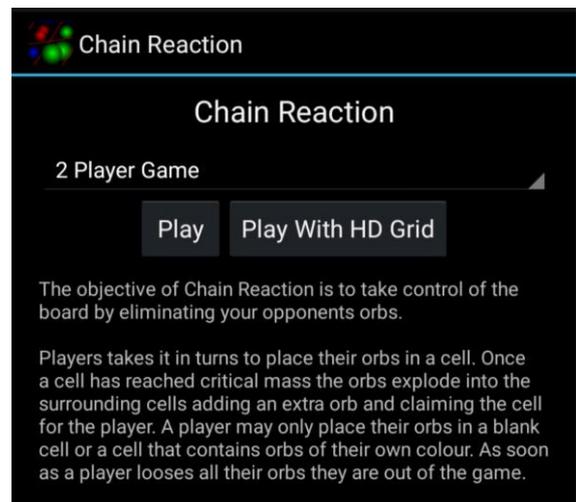
Algoritma *greedy* disusun oleh elemen-elemen berikut:

- a. Himpunan Kandidat
Berisi elemen-elemen pembentuk solusi.
- b. Himpunan Solusi
Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
- c. Fungsi Seleksi (*selection function*)
Memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
- d. Fungsi Kelayakan (*feasible*)
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.
- e. Fungsi Obyektif
Fungsi yang memaksimalkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain).

Pada beberapa masalah, algoritma *greedy* tidak selalu berhasil memberikan solusi yang benar-benar optimum. Jika jawaban terbaik mutlak (benar-benar optimum) tidak diperlukan, maka algoritma *greedy* sering berguna untuk menghasilkan solusi yang menghampiri (*approximation*) optimum, daripada menggunakan algoritma yang lebih rumit untuk menghasilkan solusi yang eksak.^[2]

B. Aturan dan Gameplay Permainan “Chain Reaction”

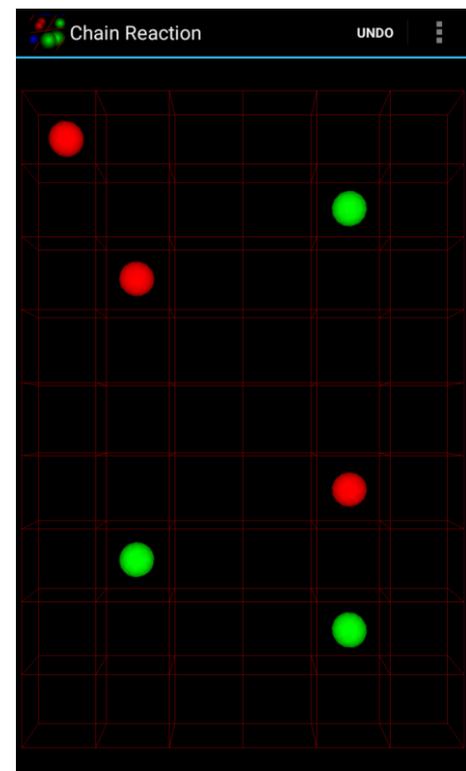
Permainan “Chain Reaction” adalah permainan strategi yang dapat dimainkan oleh 2 sampai 8 orang. Setiap pemain akan secara bergantian meletakkan bola *orb* pada sebuah papan dengan 9 baris dan 6 kolom. Warna *orb* dapat dipilih secara bebas oleh pemain berdasarkan warna RGB.



Gambar 2. Halaman Awal Permainan

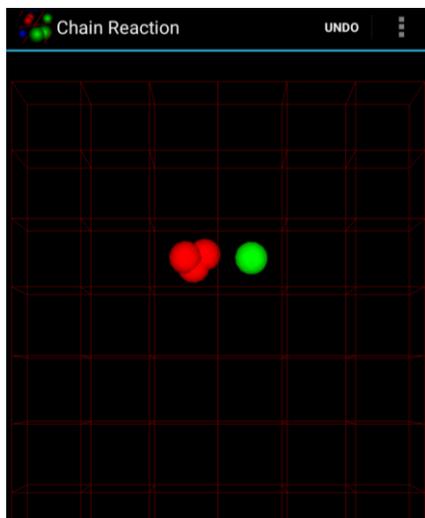
Berikut adalah aturan permainannya:^[1]

1. Pemain hanya dapat meletakkan *orb* pada petak yang kosong atau petak yang berisi *orb* dengan warna yang sama dengan warna yang dimiliki pemain. Pemain tidak dapat meletakkan *orb* ke petak yang telah berisi *orb* pemain lain.
2. Maksimal *orb* yang dapat ditampung di petak ujung (misal petak (0,0)) adalah 1 *orb*, maksimal *orb* yang dapat ditampung di petak pinggir (misal (3,0)) adalah 2 *orb*, dan maksimal *orb* yang dapat ditampung petak sisanya adalah 3 *orb*.

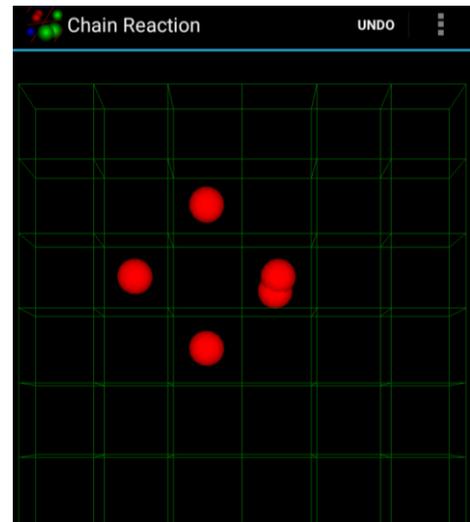


Gambar 3. Orb pada papan (Petak di ujung berisi jumlah orb yang maksimum yaitu 1 buah)

3. Petak yang telah mengandung jumlah *orb* yang maksimal akan terlihat bergetar. Jika pemain meletakkan lagi sebuah orb di petak yang telah mengandung jumlah *orb* maksimal, *orb* di petak tersebut akan meledak ke petak-petak di sampingnya (atas, bawah, kiri, kanan), menambah jumlah *orb* yang ada di petak sampingnya, dan membuat petak-petak samping yang terkena imbas ledakan menjadi milik pemain tersebut (jika awalnya adalah milik pemain lain).
4. Jika pada saat ledakan ada petak samping yang juga mempunyai jumlah *orb* yang maksimum, maka ketika orb tersebut tertambahkan ke petak samping tersebut, akan terjadi ledakan tambahan yang disebabkan oleh meledaknya petak samping walaupun pemain hanya meletakkan satu *orb* di satu petak. Begitu seterusnya jika di sampingnya ada lagi petak yang mempunyai jumlah *orb* yang maksimum yang akhirnya akan meledak dan membentuk sebuah reaksi berantai (*chain reaction*).
5. Pemain akan kalah jika semua *orb* yang dimilikinya habis diambil pemain lain.
6. Pemain akan menang jika semua *orb* yang tersisa di papan adalah miliknya.



Gambar 4. Petak berisi 3 orb sebelum meledak



Gambar 5. Keadaan papan setelah petak berisi 3 orb meledak

III. PENERAPAN STRATEGI ALGORITMA *GREEDY* PADA PERMAINAN

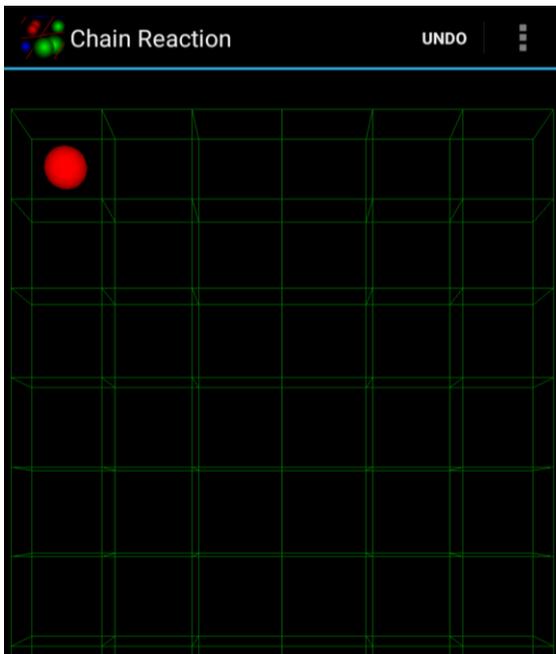
Strategi *greedy* yang dipakai dikaitkan dengan pengambilan petak milik lawan, karena fungsi objektif penerapan algoritma *greedy* adalah memenangkan permainan, yaitu mengambil semua petak milik pemain lain.

Secara umum, strateginya adalah mempersiapkan petak-petak untuk meledak dan membuat reaksi berantai secepat mungkin, karena kita hanya dapat mengambil petak berisi *orb* milik pemain lain dengan cara meledakkan petak dan membuat reaksi berantai.

Strategi lainnya adalah dengan meledakkan petak yang sudah siap diledakkan ketika ada petak musuh yang sudah siap meledak juga di samping petak kita tersebut untuk mendapatkan reaksi berantai memanfaatkan dan mendapatkan petak pemain lain.

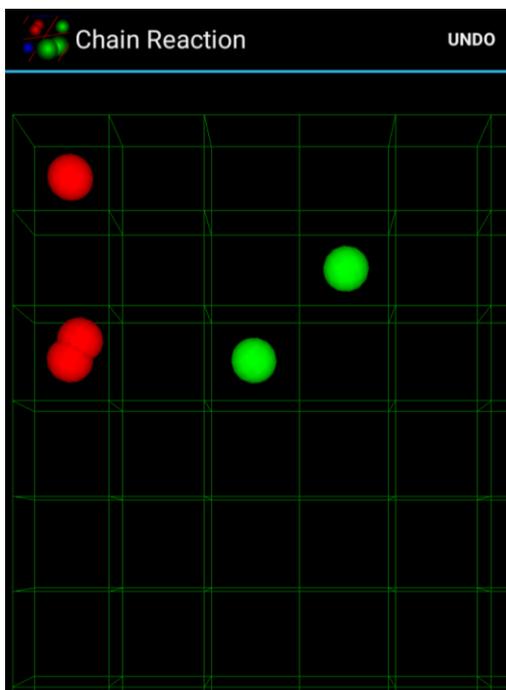
Langkah-langkah strategi *greedy* pada permainan ini adalah sebagai berikut:

1. Pada awal permainan, letakkan *orb* pada petak-petak ujung terlebih dahulu, karena maksimal *orb* yang dapat ditampung pada petak ujung hanya 1 buah *orb*, dengan demikian *orb* pada petak ujung tersebut sudah siap untuk meledak.



Gambar 6. Meletakkan orb pada petak ujung di awal permainan

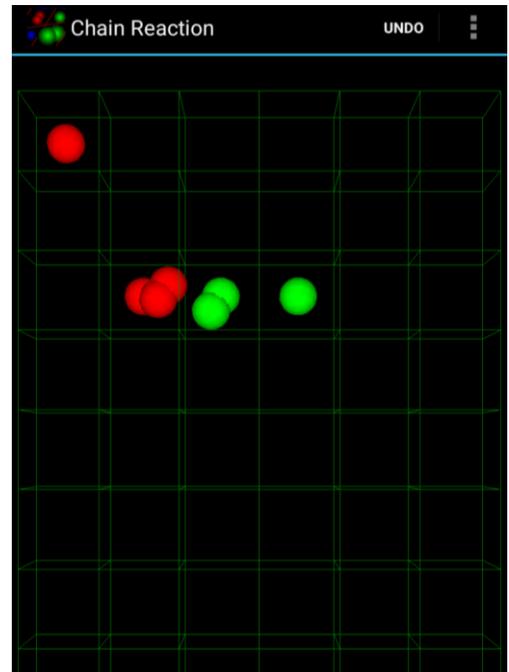
- Untuk selanjutnya, lebih baik letakkan orb pada petak-petak pinggir terlebih dahulu dan letakkan sampai satu petak berisi 2 orb, karena maksimal orb yang dapat ditampung petak pinggir adalah 2 buah orb.



Gambar 7. Meletakkan orb di pinggir setelah petak ujung terisi

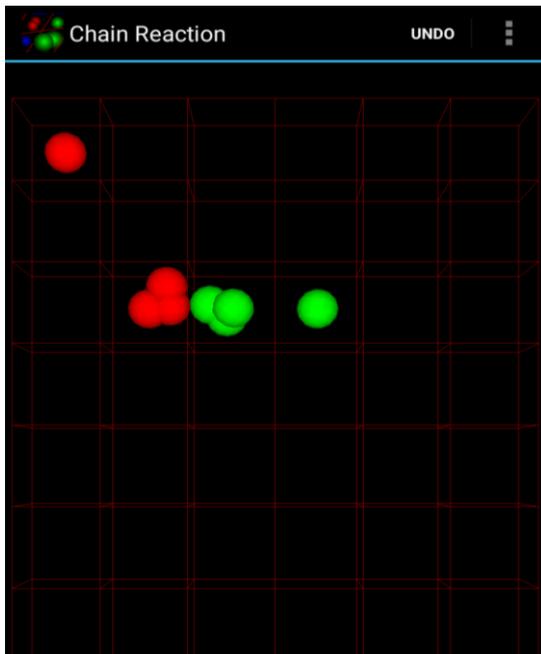
- Untuk selanjutnya barulah petak-petak tengah menjadi prioritas terakhir peletakan orb.

- Pada langkah-langkah di atas, gunakan juga prinsip “Selangkah lebih maju”. Yaitu jika ada orb milik pemain di sebelah petak kita, pastikan orb yang ada di petak kita sudah siap untuk meledak lebih dulu daripada petak pemain lain yang ada di sebelah petak kita dengan cara menambahkan orb di petak tersebut jika jumlah orb di kedua petak sama. Hal ini untuk memastikan agar petak musuh yang bergetar dapat kita manfaatkan untuk membuat reaksi berantai.

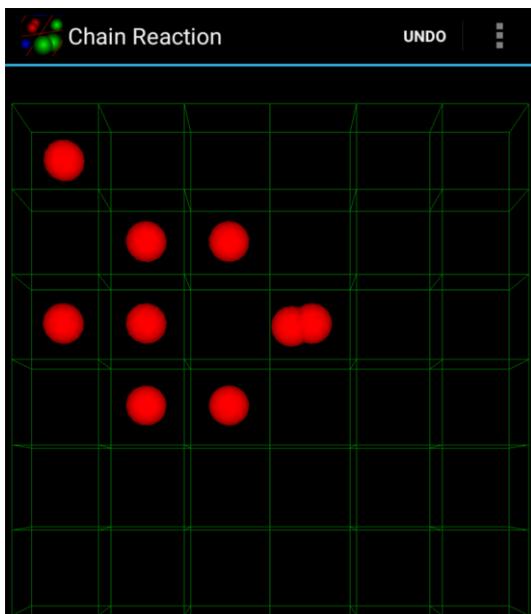


Gambar 8. Prinsip “Selangkah lebih maju”

- Berdasarkan prinsip “Selangkah lebih maju” pula, jika ada petak pemain lain yang sudah siap untuk meledak di samping petak kita yang masih belum siap untuk meledak, jangan tambahkan orb ke petak kita tersebut. Hal ini untuk memastikan agar petak kita tidak dimanfaatkan musuh untuk membuat reaksi berantai.
- Langkah *greedy* lainnya, ketika ada petak musuh yang sudah siap untuk meledak di sebelah petak kita yang sudah siap untuk meledak pula, langsung ledakkan petak kita tersebut untuk membuat reaksi berantai dan mengambil petak milik lawan.



Gambar 9. Petak musuh siap meledak di samping petak kita yang sudah siap untuk meledak pula



Gambar 10. Kondisi setelah petak diledakkan menyebabkan reaksi berantai

Himpunan Solusi di algoritma ini adalah petak-petak yang telah diletakkan *orb*.

c. Fungsi Seleksi

Fungsi Seleksi di algoritma ini adalah memilih petak untuk diletakkan *orb* berdasarkan prioritas jumlah *orb* paling sedikit yang dibutuhkan agar petak tersebut meledak.

Fungsi Seleksi lainnya adalah dengan menggunakan prinsip “Selangkah lebih maju” yaitu menambahkan *orb* di petak milik kita yang bersebelahan dengan petak pemain lain agar kita tetap selangkah lebih maju untuk meledakkan petak.

d. Fungsi Kelayakan

Fungsi Kelayakan yang ada di algoritma ini adalah membuang kandidat-kandidat yang tidak layak yaitu petak-petak milik pemain lain.

e. Fungsi Obyektif

Fungsi Obyektif yang ada di algoritma ini adalah memenangkan permainan, yaitu mengambil semua petak milik pemain lain dengan cara meledakkan petak milik kita dan membuat reaksi berantai.

IV. ANALISIS STRATEGI

A. Ketidakpastian Optimum Global

Seperti yang sudah disebutkan di dasar teori, strategi dengan menggunakan algoritma *greedy* tidak selalu menghasilkan solusi yang optimal. Pilihan pada algoritma adalah optimum lokal dengan harapan (tetapi belum tentu) mengarah ke optimum global.

Dengan kata lain, dengan memilih petak-petak yang akan diletakkan *orb* dengan langkah-langkah strategi di atas kita dapat menghasilkan solusi optimum lokal (mengambil petak milik lawan) dengan harapan (tetapi belum tentu) kita memenangkan permainan (optimum global).

B. Strategi Lebih Optimal untuk Permainan dengan 2 pemain, tetapi masih kurang optimal untuk pemain dengan jumlah banyak

Strategi yang diterapkan hanya memikirkan permainan dengan 2 pemain yaitu kita dan satu pemain lain. Permainan dengan jumlah pemain yang banyak akan membuat permasalahan lebih kompleks dan membuat algoritma menjadi sedikit kurang efektif.

Sebagai contoh, salah satu permasalahan yang bertambah seiring dengan bertambahnya jumlah pemain adalah kesempatan untuk melakukan aksi lebih sedikit sedangkan kesempatan untuk diserang lebih besar. Dengan kata lain, dalam satu putaran pergantian giliran pemain, kita hanya dapat melakukan satu aksi tetapi dapat diserang oleh pemain lain lebih dari sekali. Hal ini menyebabkan prinsip “Selangkah lebih maju” menjadi kurang efektif karena aksi yang dilakukan pemain lain sampai ke giliran kita selanjutnya lebih dari sekali.

Elemen-elemen algoritma *Greedy* yang dipakai di permainan ini adalah:

a. Himpunan Kandidat

Himpunan Kandidat di algoritma ini adalah petak-petak berukuran 9 baris dan 6 kolom yang dapat diletakkan *orb*.

b. Himpunan Solusi

V. KESIMPULAN

Algoritma *Greedy* adalah salah satu algoritma untuk memecahkan masalah, salah satunya dapat dipakai untuk membentuk strategi dalam permainan. Algoritma *Greedy* dapat dipakai untuk strategi permainan "*Chain Reaction*". Strategi *Greedy* yang digunakan di permainan ini belum tentu menghasilkan optimum global (belum tentu memenangkan permainan) tetapi langkah-langkah yang dilakukan menghasilkan optimum lokal dan mengarah ke optimum global. Algoritma dapat dikembangkan dengan memperhitungkan jumlah pemain yang banyak (lebih dari 2 pemain) ataupun dengan menerapkan algoritma lain selain *greedy* sehingga dapat mencapai solusi yang optimum ataupun lebih mengarah ke optimum global.

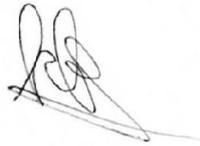
REFERENCES

- [1] https://play.google.com/store/apps/details?id=com.BuddyMattEnt.ChainReaction&hl=en_US. Diakses 13 Mei 2018 pukul 15.00 WIB.
- [2] R. Munir, *Bahan Kuliah ke-3 IF2251 Strategi Algoritmik Algoritma Greedy*. 2004.

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, 14 Mei 2018



Kevin Basuki / 13516071