# Penerapan Algoritma Greedy dalam Pergerakan Troops pada Permainan Strategi Clash of Clans

Michael Hans 13518056[1]
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*[1]13518056@std.stei.itb.ac.id*

***Abstract*—Algoritma memegang peranan penting dalam pengembangan suatu aplikasi atau perangkat lunak. Salah satu algoritma yang banyak digunakan dalam pengembangan *game* adalah algoritma Greedy. Algoritma Greedy berperan dalam menentukan pergerakan *computer player* atau *bot* sehingga dapat bergerak dengan sendirinya dengan pergerakan yang cerdas. Pergerakan *bot* tersebut dimanfaatkan dalam permainan strategi terkenal bernama Clash of Clans, yang di dalamnya bot diimplementasikan pada pasukan atau *troops* yang telah dikeluarkan oleh player.**

***Keywords*—algoritma, greedy, euclidean, minimum**

## I. PENDAHULUAN

Algoritma memegang peranan penting dalam pengembangan suatu aplikasi atau perangkat lunak. Beberapa algoritma memiliki teknik-teknik tertentu agar dapat berjalan seefisien mungkin. Salah satu algoritma yang cukup banyak digunakan adalah algoritma Greedy. Algoritma Greedy memiliki prinsip utama, yaitu mengambil nilai terbaik dari setiap keputusan yang tersedia. Salah satu penggunaan algoritma Greedy adalah membuat tingkah laku dari suatu bot atau computer player dalam hal menentukan langkah-langkah selanjutnya dalam suatu permainan.

Permainan yang seringkali memanfaatkan bot dalam penggunaannya adalah permainan berbasis strategi. Pada bagian ini, permainan strategi yang akan diambil adalah permainan cukup terkenal sejak tahun 2013, yaitu Clash of Clans. Pada dasarnya, Clash of Clans adalah salah satu permainan strategi yang bertujuan untuk memenangkan permainan dengan menghancurkan suatu base berdasarkan strategi attack yang telah disiapkan di awal permainan.

*Gambar 1. Logo dari Permainan Clash of Clans*

Pengguna hanya mampu mengeluarkan *troops* atau skill-skill tertentu tanpa bisa mengatur secara langsung gerak-gerik masing-masing troops. Setiap troops yang sudah dideploy akan berjalan dengan sendirinya sesuai dengan konfigurasi bot yang sudah disetup dalam permainan. Salah satu perilaku bot yang cukup unik adalah perilaku dalam menentukan aksi yang pertama kali harus dilakukan dan seterusnya. Aksi ini dibuat berdasarkan suatu algoritma, yang cocok dengan implementasi algoritma Greedy. Dalam makalah ini, akan dibahas secara mendalam cara kerja bot pada *troops* dalam berperilaku dalam permainan strategi tersebut.

## II. LANDASAN TEORI

### A. Algoritma Greedy

Algoritma greedy adalah algoritma pemecahan persoalan dengan mencari solusi optimum dari setiap langkah yang dilakukannya. Persoalan optimasi bisa dibagi menjadi dua jenis, yaitu persoalan maksimasi dan persoalan minimasi. Persoalan minimasi adalah persoalan dimana kita menginginkan suatu persoalan A dijalankan untuk menghasilkan proses paling minimum. Langkah-langkah yang dilakukan untuk mencapai penyelesaian A kita lakukan dengan cara mencari langkah yang paling optimum. Contoh persoalan minimasi adalah proses penukaran uang dengan jumlah paling minimum koin. Sedangkan sebaliknya proses maksimisasi adalah proses dimana kita ingin menyelesaikan suatu persoalan A untuk menghasilkan proses akhir paling maksimum. Contoh persoalan ini adalah persoalan *knapsack* dimana ada banyak jenis barang dan kita ingin membawa barang dengan nominal harga yang paling besar.

Algoritma greedy memiliki beberapa karakteristik dalam penyelesaiannya. Prinsip algoritma ini adalah "ambil yang kamu bisa dapatkan sekarang". Algoritma greedy akan menjalankan solusi langkah per langkah. Setiap langkah yang dijalani, akan terdapat banyak pilihan langkah. Oleh karena itu, algoritma Greedy akan memilih langkah terbaik yang bisa diambil di setiap *step* atau langkahnya. Sehingga kita akan mendapatkan optimum lokal dari setiap langkah dan berharap akan mengarahkan kita pada solusi optimum global pada akhir solusi.

Namun, nyatanya nilai optimum lokal tidak selalu membawa kita pada solusi akhir optimum global. Karena ada masalah, yang solusi awalnya kurang optimum namun jika dilihat keseluruhan, walaupun tidak optimum di awal, namun optimum secara keseluruhan. Hal ini merupakan salah satu kelemahan

algoritma Greedy. Algoritma Greedy tidak melakukan operasi atau pemeriksaan secara menyeluruh terhadap semua alternatif solusi yang ada, seperti *exhaustive search.* Sehingga algoritma ini tidak selalu berhasil memberi solusi optimal.

Elemen-elemen algoritma greedy adalah sebagai berikut.

1. Himpunan kandidat (C)
   Berisi elemen-elemen pembentuk solusi.

2. Himpunan solusi (S)
   Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.

3. 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.

4. Fungsi kelayakan (feasibility function)
   Memerika 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).

5. Fungsi objektif
   Fungsi yang memaksimukan atau meminimumkan nilai solusi. Contohnya adalah panjang lintasan, keuntungan, dan lain-lain.

Dengan kata lain: Algoritma greedy melibatkan pencarian sebuah himpunan bagian S, dari himpunan kandidat C, yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimasi oleh fungsi objektif.

B. Permainan Clash of Clans

Clash of Clans adalah permainan berjenis strategi video game yang dikembangkan dan dikeluarkan oleh sebuah pengembang game bernama Supercell. Permainan ini pertama kali dikeluarkan pada platform iOS pada 2 Agustus 2012, kemudian dikeluarkan pada platform Android pada 7 Oktober 2013.


*Gambar 2. Cover game dari permainan Clash of Clans*

Permainan ini mengambil tema *fantasy* dengan pemain sebagai kepala suku dari suatu wilayah. Sebagai kepala suku, pemain diminta untuk mengembangkan desa menjadi desa yang penuh kekuatan dan berkuasa seiring dengan kenaikan level dalam permainan. Untuk meraih tujuan tersebut, pemain diharuskan untuk menyerang kota lawan untuk memperoleh sumber daya yang cukup untuk mengembangkan desa tersebut.

Beberapa objek penting yang digunakan dalam permainan ini adalah sebagai berikut.

1. Pasukan (troops)
   Pasukan merupakan bagian inti dalam permainan untuk menyerang desa lawan dengan strategi dan komposisi pasukan tertentu.

2. Bangunan bertahan (defense building)
   Bangunan bertahan merupakan bagian inti dalam permainan yang merupakan lawan dari troops, yaitu desa yang akan diserang oleh pasukan tersebut.

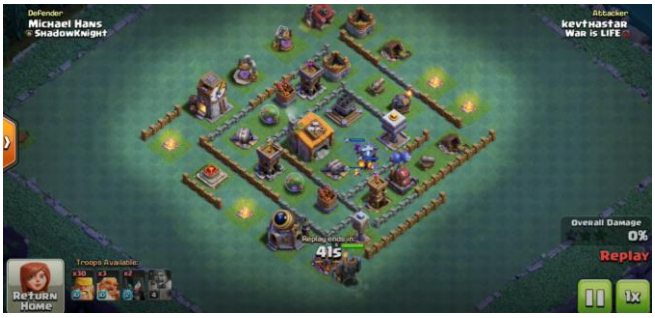
3. Ramuan (spells)
   Ramuan (spells) merupakan suatu kemampuan tambahan yang digunakan dalam permainan untuk meningkatkan kemampuan dalam menyerang menggunakan troops.

4. Suku (clan)
   Suku (clan) adalah kumpulan pemain yang berada dalam satu kelompok sebagai wadah untuk berkomunikasi dengan pemain lain dan meminta pasukan bantuan dari pemain lain, yang dalam hal ini adalah kepala desa dalam suku yang sama.

Alur permainan secara umum dalam permainan Clash of Clans adalah sebagai berikut.

1. Menyiapkan pasukan dengan melakukan training pada Barracks sebagia tempat membaangkitkan pasukan baru sesuai dengan kapasitas dan pilihan yang ada.
2. Menekan tombol Find Match untuk menjelajahi desa-desa yang ingin diserang. Umumnya, desa yang akan diserang adalah desa dengan sumber daya terbanyak.
3. Menyusun strategi untuk menghancurkan desa lawan.
4. Menjalankan strategi tersebut dengan mengeluarkan pasukan-pasukan sesuai dengan preferensi penyerang.
5. Pasukan akan bergerak dan menyerang desa lawan berdasarkan tingkah laku yang sudah disetup pada setiap jenis pasukan yang ada.

Berikut ini adalah tampilan permainan Clash of Clans dalam melakukan penyerangan terhadap desa lawan.




*Gambar 3. Tampilan Fase Attack terhadap desa lawan*

Dari sisi algoritma, setiap objek tentu akan memiliki

algoritma-algoritma tertentu dalam merepresentasikan permainan strategi. Salah satu bagian yang ingin dikupas adalah perilaku dari pasukan.

Pemain hanya bertugas dalam mendeploy pasukan sesuai dengan posisi yang ditunjuk pemain. Untuk selanjutnya, semuanya dikembalikan kepada perilaku dari setiap pasukan yang dideploy. Salah satu perilaku pasukan adalah menyerang bangunan dengan jarak terdekat dari posisi awal. Konsep ini merupakan penerapan konsep minimisasi yang terdapat pada algoritma Greedy yang akan dikupas lebih lanjut pada bagian Hasil dan Pembahasan.

C. Jarak Euclidean (*Euclidean Distance*)

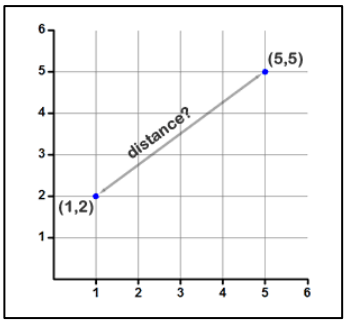
*Euclidean distance* adalah perhitungan jarak antara 2 buah titik dalam *Euclidean space*. *Euclidean space* diperkenalkan oleh Euclid seorang matematikawan dari Yunani sekitar tahun 300 SM untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini berkaitan dengan Teorema Phytagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi.

**Pada 2 dimensi**

Untuk mencari jarak euclidean antara dua buah titik pada 2 dimensi, berikut ini adalah rumus mencari jarak euclidean tersebut.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Keterangan:
$x_1$ = posisi horizontal x dari titik awal pada koordinat kartesius
$x_2$ = posisi horizontal x dari titik akhir pada koordinat kartesius
$y_1$ = posisi vertikal y dari titik awal pada koordinat kartesius
$y_2$ = posisi vertikal y dari titik akhir pada koordinat kartesius




*Gambar 3. Ilustrasi Jarak Euclidean pada Koordinat Kartesius*

Misalkan titik pertama mempunyai kordinat (1,2). Titik kedua ada di kordinat (5,5). Caranya adalah kurangkan setiap kordinat titik kedua dengan titik yang pertama. Yaitu, (5-1,5-2) sehingga menjadi (4,3). Kemudian pangkatkan masing-masing sehingga memperoleh (16,9). Kemudian tambahkan semuanya sehingga memperoleh nilai 16+9 = 25. Hasil ini kemudian diakarkan menjadi 5. Sehingga jarak euclideannya adalah 5.

## III. HASIL DAN PEMBAHASAN

Seperti yang telah dijelaskan sebelumnya, algoritma Greedy merupakan algoritma yang memiliki prinsip memaksimumkan atau meminimumkan suatu fungsi objektif. Pada permainan Clash of Clans, algoritma Greedy diterapkan dalam perilaku pasukan (troops) yang dalam hal ini bertindak sebagai bot atau computer player. Perilaku tersebut adalah menyerang bangunan dengan jarak terpendek terlebih dahulu dari seluruh opsi bangunan yang bisa diserang.

Pada bagian ini, penulis telah membuat simulasi sederhana terkait perilaku gerak dari pasukan dalam menyerang bangunan terdekat pada permainan. Berikut ini adalah implementasi simulasi sederhana menggunakan algoritma Greedy. Implementasi akan dibuat dalam bahasa pemrograman Python.

A. Tahap Persiapan Simulasi

Tahap Persiapan Simulasi adalah tahap penulis akan memaparkan rencana penyusunan algoritma Greedy beserta komponen-komponen dalam algoritma Greedy tersebut.

1. Himpunan kandidat (C)
   Himpunan kandidat merupakan semua kemungkinan bangunan yang dapat diserang dari posisi semula.

2. Himpunan solusi (S)
   Himpunan solusi merupakan pilihan langkah yang dapat mengoptimalkan fungsi objektif yang ada. Dalam hal ini, himpunan solusi berisi bangunan yang akan diserang dengan jarak terpendek.

3. Fungsi seleksi (selection function)
   Fungsi seleksi adalah memilih jarak terpendek dari jarak posisi awal pasukan terhadap posisi setiap bangunan yang terdapat dalam permainan agar waktu yang dipakai bisa sedikit mungkin. Bangunan dengan jarak terpendek akan dikunjungi dan bila sudah dikunjungi, akan dihapus dari bangunan-bangunan yang tersedia dalam senarai.

4. Fungsi kelayakan (feasibility function)
   Fungsi kelayakan adalah mengecek ketersediaan bangunan yang terdapat dalam senarai yang berisi koordinat bangunan.

5. Fungsi objektif
   Fungsi objektif adalah menghancurkan sebanyak mungkin bangunan dengan rute terpendek dan waktu tercepat.

Arena permainan direpresentasikan dalam sebuah peta yang berbentuk segi empat. Pada simulasi sederhana ini, penulis mepresentasikan peta dalam bentuk matriks berukuran X, Y dengan X sebagai panjang peta dan Y sebagai lebar peta. Secara umum, algoritma Greedy yang akan diterapkan pada permainan adalah sebagai berikut.

1. Misalkan P adalah koordinat bangunan terdekat.
   Inisialisasi variabel penyimpan koordinat pada pasukan

dan senarai bangunan. Inisialisasi nilai minDistance sebesar mungkin.
2. Jika senarai bangunan kosong, kembalikan None atau tidak ada koordinat
3. Jika senarai bangunan tidak kosong, lakukan pengulangan sebanyak jumlah bangunan yang ada dalam senarai bangunan.
4. Hitung jarak euclidean dari koordinat pasukan dengan bangunan yang ada dalam senarai bangunan.
5. Jika jarak euclidean yang diperoleh lebih kecil dari jarak euclidean minimum sebelumnya, gantikan jarak euclidean minimum dengan nilai tersebut dan gantikan P dengan koordinat bangunan dengan jarak terdekat.
6. Kembali ke langkah (2).

Algoritma Greedy diterapkan pada fungsi Get Nearest Defense dengan pseudocode sebagai berikut.

```
function GetNearestDefense(input B: list of Point, T: Point) → Point
{Mengembalikan Point dengan jarak terpendek antara T dengan salah satu Point dari B}

Kamus Data:
    P: Point
    MinDistance: float
    Distance: float
    i, Length: integer

Algoritma:
    P ← None
    MinDistance ← None
    Length ← B.length
    for i:= 1 to Length do
        Distance ← B[i].distance(T)
        if Distance < MinDistance then
            MinDistance ← Distance
            P ← B[i]
        endif
    endfor
    → P
```

Kompleksitas dari Algoritma Greedy yang diterapkan pada fungsi GetNearestDefense adalah O(n) karena pengulangan dilakukan sebanyak n kali tergantung dari banyak elemen Point dari senarai B.

B. Tahap Pelaksanaan Simulasi

Untuk pelaksanaan simulasi, berikut ini adalah daftar variabel yang digunakan dalam simulasi.

- B = sebuah senarai yang terdiri atas koordinat (x,y) dari setiap bangunan bertahan yang berada dalam permainan.
- Archer = sebuah koordinat (x,y), troop sebagai pasukan yang akan bergerak berdasarkan algoritma Greedy dan menyerang bangunan bertahan dalam permainan
- Map = sebuah matriks berukuran 10 x 10 untuk merepresentasikan posisi setiap building dalam B dan troop Archer.

Representasi dari Map yang akan ditampilkan dalam simulasi mengikuti aturan sebagai berikut.

| Y\X | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | T | | | | |
| 1 | | | | | |
| 2 | | | D | | |
| 3 | | | | | |
| 4 | | | | | D |

*Tabel 1. Gambaran Representasi dari Map (X,Y)*

Keterangan:

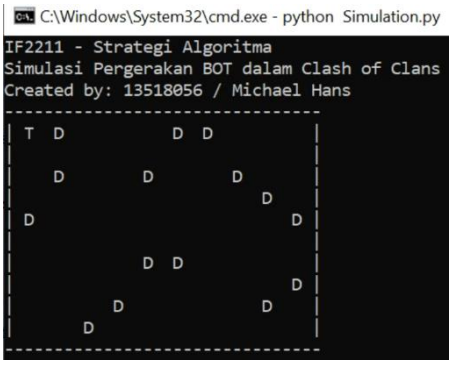
T = posisi koordinat (x,y) dari pasukan (troop)
D = posisi koordinat (x,y) dari bangunan

Berikut ini adalah alur kerja dari simulasi pergerakan bot dalam permainan:

1. Inisialisasi awal variabel-variabel yang digunakan dalam permainan.
   Archer = Point (0,0)
   B = RandomBuildingPosition(15)
2. Jika senarai bangunan kosong, program dihentikan.
3. Jika senarai bangunan tidak kosong, tentukan koordinat (x,y) dari bangunan dengan jarak terpendek dari posisi Archer awal dengan menggunakan algoritma Greedy.
4. Hitung Δ x dan Δ y dengan titik akhir adalah posisi bangunan terdekat dan titik awal adalah posisi Archer saat ini.
5. Menggunakan Δ x dan Δ y yang diperoleh untuk memindahkan Archer satu per satu satuan menuju posisi bangunan terdekat tersebut. Δ x dan Δ y mempengaruhi pergerakan dengan aturan sebagai berikut.
   - Jika Δ x bernilai positif, posisi Archer bergerak 1 satuan ke arah timur sebanyak Δ x kali
   - Jika Δ x bernilai negatif, posisi Archer bergerak 1 satuan ke arah barat sebanyak abs(Δ x) kali
   - Jika Δ y bernilai positif, posisi Archer bergerak 1 satuan ke arah selatan sebanyak Δ y kali
   - Jika Δ y bernilai negatif, posisi Archer bergerak 1 satuan ke arah utara sebanyak abs(Δ y) kali
6. Bila bangunan terdekat sudah dikunjungi, hapus dari senarai bangunan.
7. Kembali ke langkah (2).

C. Tahap Pengujian Simulasi

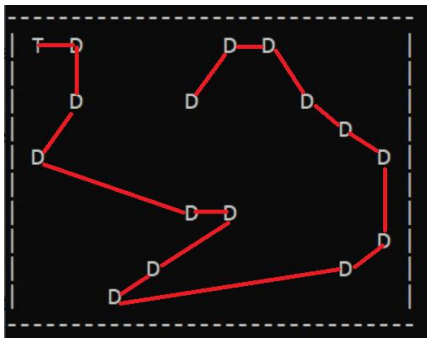
Berikut ini adalah tampilan awal dari simulasi.




*Gambar 4. Tampilan Awal Simulasi*

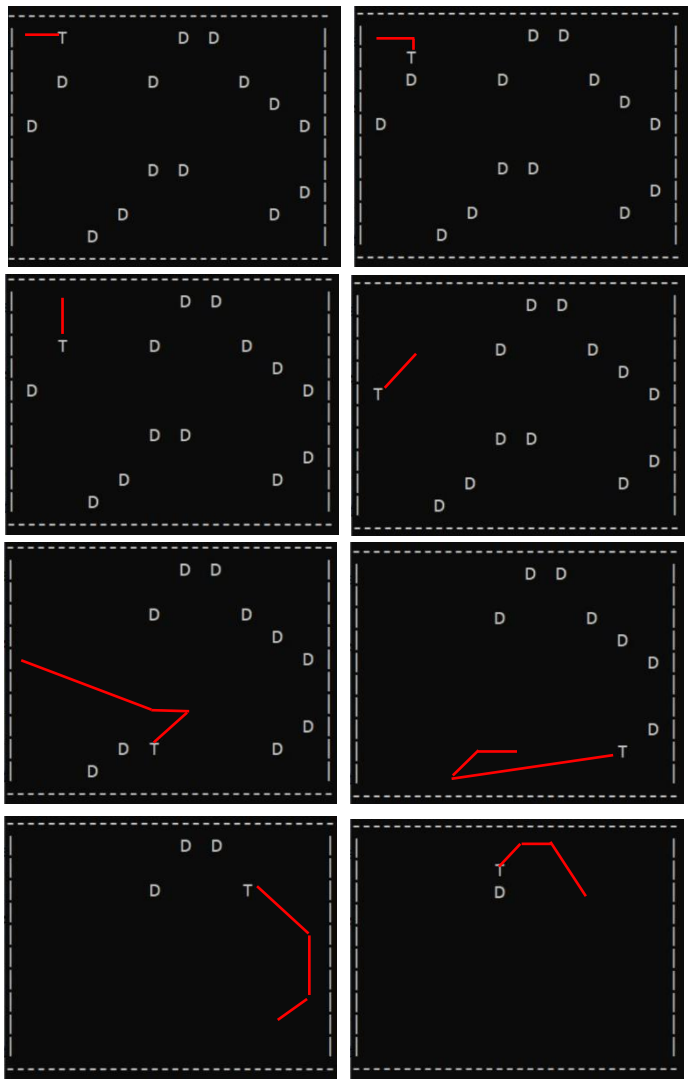
Pada tampilan pada gambar 4, terdapat character T yang melambangkan posisi pasukan awal dan character D yang melambangkan bangunan yang akan diserang. Berdasarkan algoritma Greedy, pasukan akan menyerang bangunan terdekat terlebih dahulu menurut perhitungan jarak euclidean terpendek.

Secara teori, algoritma Greedy akan membuat pergerakan pasukan T seperti pada tampilan di bawah ini.



*Gambar 5. Algoritma Greedy yang akan dijalankan*

Dari hasil pengujian, algoritma Greedy berjalan dengan tahapan-tahapan digambarkan dalam bentuk tampilan per tampilan sebagai berikut.

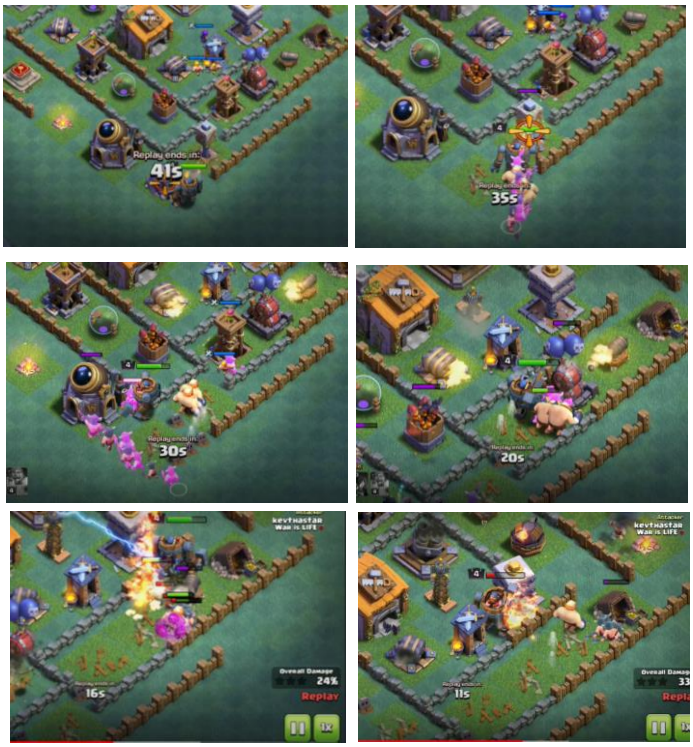


*Gambar 6. Langkah per langkah pergerakan T pada peta*

Dari hasil pengujian diatas, terbukti bahwa pergerakan karakter T mengikuti posisi bangunan terdekat dengan menghitung jarak terpendek antara posisi T terhadap setiap bangunan pada permainan, yaitu D. Selengkapnya terkait simulasi algoritma greedy dalam Clash of Clans, dapat dilihat lebih lanjut pada laman berikut ini.
https://github.com/michaellhans/Greedy-In-ClashOfClans

D. Perbandingan antara Simulasi dan Permainan Asli

Simulasi ini serupa dengan cara kerja dari pasukan yang dikeluarkan oleh *player* saat memainkan penyerangan pada permainan Clash of Clans. Berikut ini beberapa *screenshot* yang diambil dalam permainan Clash of Clans untuk menunjukkan pergerakkan pasukan.



*Gambar 7. Langkah per langkah pergerakan pasukan-pasukan dari sebagian kecil fase penyerangan.*

Perhatikan bahwa pasukan-pasukan tersebut bergerak menurut posisi bangunan terdekat masing-masing pasukan. Terdapat beberapa jenis pasukan yang memang telah diatur akan menyerang bangunan bertahan terlebih dahulu seperti Giant, namun pada dasarnya tetap menyerang bangunan terdekat. Contoh lebih konkret bisa dilihat pada pasukan dengan nomor level yang berada di atasnya, yaitu Hero. Pasukan berjenis ini terlihat jelas pada gambar terakhir dari gambar 7 bahwa ia memutari tembok untuk menyerang bangunan terdekat dari titik awal baru ketika berhasil menghancurkan bangunan terakhir.

Jika dibandingkan dengan simulasi, tembok atau *wall* menjadi opsi terakhir untuk diserang pada permainan aslinya. Namun pada simulasi, untuk menunjukkan fokus algoritma yang mengutamakan jarak terdekat, maka yang perlu ditampilkan adalah sebuah troop yang akan menyerang seluruh bangunan yang ada dalam peta tanpa adanya tembok atau *wall* tersebut dan

mengasumsikan bahwa bangunan bertahan tidak menyerang balik terhadap troop yang menyerang tersebut.

Pergerakan pasukan memanfaatkan algoritma Greedy agar bisa mengikuti cara kerja bila pemain memegang kontrol penuh atas pergerakan pasukan. Pertimbangan strategi yang diinginkan oleh pemain umumnya adalah sebagai berikut.

1. Menyerang dari sembarang titik yang dianggap menjadi titik yang cocok untuk memulai titik penyerangan.
2. Menginginkan agar pasukan tidak mati secara sia-sia, yakni menghancurkan sebanyak mungkin bangunan.
3. Berhasil menghancurkan pertahanan lawan secara penuh dengan pasukan yang cukup dalam penyerangan.

Dengan tiga pertimbangan tersebut, pasukan akan berjalan kurang lebih dengan memanfaatkan algoritma Greedy. Untuk memenuhi keinginan pemain dalam menghancurkan bangunan sebanyak-banyaknya, pasukan diatur sedemikian rupa agar menyerang bangunan terdekat terlebih dahulu. Seiring berjalannya waktu, pasukan memungkinkan untuk mati lebih awal atau tetap hidup karena *health points* setiap pasukan juga akan berkurang.

Untuk pembahasan lebih lanjut terkait penerapan Greedy pada permainan Clash of Clans, penulis menyiapkan video pembahasan algoritma tersebut pada laman video berikut ini.
https://youtu.be/FC3z3QRWfyQ

## IV. Kesimpulan

Algoritma Greedy banyak digunakan dalam pengembangan aplikasi berbasis permainan. Dalam praktek sehari-hari, algoritma Greedy digunakan dalam pembuatan pergerakan dari *computer player* atau *bot* yang lebih cerdas. Simulasi sederhana yang penulis implementasikan mampu merepresentasikan pergerakan utama dari pergerakan bot dalam permainan strategi Clash of Clans. Pergerakan pasukan dalam menyerang bangunan-bangunan mengikuti prioritas bangunan terdekat dari posisi awal pasukan saat menyerang desa lawan.

## V. Ucapan Terima Kasih

Penulis mengucapkan puji dan syukur yang sebesar-besarnya kepada rahmat Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga makalah berjudul “Penerapan Algoritma Greedy dalam Permainan Strategi Clash of Clans” dapat diselesaikan. Penulis mengucapkan terima kasih kepada selaku dosen pengajar IF2211 Strategi Algoritma, Dr. Nur Ulfa Maulidevi ST, M.Sc. yang telah memberikan bimbingan dan ilmu terkait materi Strategi Algoritma ini, khususnya algoritma Greedy dan pengaplikasiannya. Penulis juga mengucapkan terima kasih kepada teman-teman yang telah memberikan dukungan kepada penulis untuk menuliskan makalah ini. Tak lupa penulis juga mengucapkan terima kasih kepada para penulis sumber referensi yang telah memberikan penulis ilmu yang dibutuhkan untuk menyelesaikan makalah ini.

## Referensi

[1] Munir, Rinaldi. Algoritma Greedy (2020). https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Algoritma-Greedy-(2020).pdf. Diakses tanggal 25 April 2020, pukul 16.40 GMT +7.

[2] Rinaldi Munir, Diktat kuliah IF2251 Strategi Algoritmik, Teknik. Informatika ITB.

[3] Anton, Howard. 2014. Elementary Linear Algebra. United States of America : Wiley

[4] Tim Penulis Fandom Clash of Clans. https://clashofclans.fandom.com/wiki/Clash_of_Clans_Wiki. Diakses pada 25 April 2020, pukul 19.56 GMT +7.

[5] Tim Penulis Blogs ITB. https://blogs.itb.ac.id/anugraha/2014/09/10/teoripengukuran-jarak/. Diakses pada 25 April 2020, pukul 17.50 GMT+7.

## 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, 27 April 2020

Michael Hans
13518056