Penggunaan Algoritma Greedy pada Scheduling Permainan Clash of Clans

Fitrandi Ramadhan, 13508065
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13508065@students.if.itb.ac.id

***Abstract*—Clash of Clans, sebuah permainan simulasi strategi dimana seorang pemain membangun desa dan pasukan untuk menghancurkan desa pemain lain. Pembangunan desa yang dilakukan pada permainan Clash of Clans dilakukan dalam satuan waktu nyata dan durasi yang relatif lama. Kesalahan-kesalahan yang dilakukan oleh pemain dapat mengakibatkan kerugian waktu yang relatif cukup lama. Untuk itu diperlukan suatu alat bantu untuk pemain dalam memudahkan mengelola jadwal pembangunan desanya.**

***Index Terms*—Clash of Clans, Strategy Simulation Games, Real-time Simulation, Algoritma Greedy, Machine Scheduling**

I. PENDAHULUAN

Dengan semakin semaraknya perkembangan mobile device, para pengembang perangkat lunak pun tidak mau kalah dengan terus menciptakan kreasi-kreasi barunya. Terlebih lagi pada domain game. Hampir setiap orang pada masa sekarang ini memiliki perangkat mobile pribadi, paling tidak sebuah telepon genggam. Telepon genggam pada masa sekarang (smartphone) telah memiliki kemampuan komputasi yang luar biasa, dimana telah dilengkapi dengan operating system yang lebih fluid dan dinamis. Dari sekian banyak permainan yang bermunculan ada beberapa game yang dapat dikatakan merajai persaingan ketat mobile games pada tahun 2013 ini, salah satu nya adalah Clash of Clans. Sebuah permainan simulation strategy yang dikembangkan oleh Supercell, sejak pertama kali dirilis pada platform iOS Clash of Clans langsung meroket dalam jumlah download dan in-game purchase. Tak hanya itu sejak tanggal 7 Oktober 2013 dirilis untuk platform Android sampai saat makalah ini ditulis Clash of Clans masih tak terkalahkan menjajaki peringkat satu top grossing game di Google Play.

Clash of clans menggunakan menggunakan mekanisme thin-client with thick back-end server dan real-time simulation. Pemain diminta untuk membangun sebuah kastil dengan berbagai functional dan non-functional structure. Semua structure atau bangunan yang dibuat pada permainan Clash of Clans dibuat dengan menggunakan resource dan durasi pembuatanya dilakukan dengan satuan waktu nyata. Apabila suatu bangunan dikatakan waktu pembuatanya adalah 30 menit maka bangunan itu akan selesai 30 menit real-time tanpa memperdulikan player sedang online ataupun offline. Begitu pula dengan durasi upgrade bangunan, jika waktu yang dibutuhkan tertera 7 hari, maka pemain harus menuggu 7 hari nyata sampai bangunan itu selesai dibangun.

Karena durasi yang "lama" ini pemain pun harus dapat mengelola waktu dan resource nya dengan baik untuk mendapatkan hasil yang maksimal. Untuk membantu pemain mengelola masalah dapat dikembangkan sebuah perencana atau scheduler untuk mendapatkan hasil permainan yang optimal, atau dalam konteks ini adalah waktu yang minimal. Algoritma greedy sebagai salah satu algoritma pemecahan masalah yang cukup populer dalam hal scheduling dapat digunakan sebagai core dalam planner atau scheduler permainan ini.

Mengingat basis pemain Clash of Clans ini terdiri dari rentang umur yang sangat bervariasi. Permainan ini dapat dimainkan oleh anak berusia tahun hingga orang tua sekalipun. Hal ini dikarenakan kesuksesan Supercell dalam mendevelop gameplay dan tidak lupa user interface yang sangat natural. Perbedaan antara pemain yang berusia sangat muda dan dewasa tentunya ada pada pola pikir dan strategi bermain yang mereka gunakan. Pada permainan ini setiap pemain dapat melakukan apa saja dengan urutan yang mereka tentukan sendiri. Perbedaan pola pikir ini tentunya akan menghasilkan perbedaan kesuksesan pemain dalam permainan ini. Karena itu sebuah scheduler yang mudah digunakan dan memberikan petunjuk strategi yang optimal menjadi salah satu solusi yang baik.

Sifat algoritma greedy yang to the point membuatnya menjadi pilihan utama dalam penyelesaian masalah seperti ini. Dengan memilih setiap langkahnya dengan pilihan secara optimum lokal karena memang domain permasalahan yang sangat spesifik dan kompleksitas yang sederhana. Algoritma ini diharapkan dapat menjadi solusi menemukan optimasi global dengan pendekatan pemilihan optimum lokal. Dengan makalah ini diharapkan

dapat tercapai pemahaman lebih mendalam mengenai algoritma greedy ini dalam aplikasinya secara nyata khususnya bidang machine scheduling.

Top Grossing Applications February 2013

iPhone, United States, apps released since January 2012

Grossing Rank	Application	All-time ARPD	Release date
1	Clash of Clans	$4.66	2012-06
2	Candy Crush Saga ®	$1.14	2012-10
3	Hay Day	$3.29	2012-05
4	MARVEL War of Heroes	$2.93	2012-09
5	The Simpsons™: Tapped Out	$2.14	2012-02
6	Big Fish Casino – Free Slots, Poker, Blackjack and More!	$6.80	2012-08
7	Rage of Bahamut	$7.04	2012-05
8	The Hobbit: Kingdoms of Middle-earth	$4.64	2012-10
9	What's the Word? - new quiz with pics and word	$0.37	2012-11
10	TurboTax SnapTax	$2.18	2012-01

iPhone ony, iPad revenues and downloads for universal apps not included in this analyis.

Fig 1.1 US Top Grossing Apps for iOS

II. DASAR TEORI

A. Algoritma Greedy

Kata greedy pada algoritma ini berarti tamak atau rakus. Sesuai dengan kata dasar ini algorimta greedy membentuk solusi langkah per langkah. Pada setiap langkah tersebut akan dipilih keputusan yang paling optimal. Keputusan tersebut tidak perlu memperhatikan keputusan selanjutnya yang akan diambil, dan keputusan tersebut tidak dapat diubah lagi pada langkah selanjutnya. Dapat dikatakan bahwa prinsip algoritma greedy ini adalah "Take what you can get now.". Maksud dari prinsip tersebut adalah sebagai berikut adalah pada setiap langkah dalam algoritma greedy, kita ambil keputusan yang paling optimal untuk langkah tersebut tanpa memperhatikan konsekuensi pada langkah selanjutnya. Kita namakan solusi tersebut dengan optimum lokal. Kemudian saat pengambilan nilai optimum lokal pada setiap langkah, diharapkan tercapai optimum global, yaitu tercapainya solusi optimum yang melibatkan keseluruhan langkah dari awal sampai akhir.

Elemen-elemen yang digunakan dalam penerapan algoritma greedy antara lain :

1. Himpunan Kandidat
   Himpunan yang berisi elemen pembentuk solusi.
2. Himpunan Solusi
   Himpunan yang terpilih sebagai solusi persoalan.
3. Fungsi Seleksi
   Fungsi yang memilih kandidat yang paling mungkin untuk mencapai solusi optimal.
4. Fungsi Kelayakan
   Fungsi yang memeriksa apakah suatu kandidat yang dipilih dapat memberikan solusi yang layak. Maksudnya yaitu apakah kandidat tersebut bersama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada.
5. Fungsi SolusiFungsi yang mengembalikan nilai boolean. True jika himpunan solusi yang sudah tebentuk merupakan solusi yang lengkap; False jika himpunan solusi belum lengkap.
6. Fungsi Objektif
   Fungsi yang mengoptimalkan solusi.

Skema Umum Algoritma Greedy adalah misal kita mengasumsikan elemen algoritma greedy sebagai berikut.

- Himpunan Kandidat = C,
- Himpunan Solusi = S,
- Fungsi Seleksi = select(),
- Fungsi Kelayakan = feasible(),
- Fungsi Solusi = solution(), dan
- Fungsi Obyektif = objective().

Skema umum dari algoritma greedy dapat kita tuliskan:

- Inisialisasi S dengan kosong.
- Pilih sebuah kandidat dari C (dengan select()).
- Kurangi C dengan kandidat yang telah terpilih di atas.
- Periksa apakah kandidat yang dipilih tersebut bersama-sama dengan S membentuk solusi yang layak (dengan feasible()). Jika ya, masukkan kandidat ke S; jika tidak buang kandidat tersebut dan tidak perlu ditelaah lagi.
- Periksa apakah S yang sudah terbentuk telah memberikan solusi yang lengkap (dengan solution()). Jika ya, berhenti; jika tidak, ulangi dari langkah 2.

```
function JobSchedulling1(input C : himpunan_job)
→ himpunan_job
{ Menghasilkan barisan job yang akan diproses
oleh mesin }

Deklarasi
  i : integer
  J : himpunan_job   { solusi }

Algoritma
  J ← {}
  while C ≠ {} do
     i ← job yang mempunyai p[i] terbesar
     C ← C - {i}
     if (semua job di dalam J ∪ {i} layak) then
        J ← J ∪ {i}
     endif
  endwhile
  { C = {} }
  return J
```

B. Clash of Clans

Clash of Clans adalah sebuah permainan strategi simulasi yang menggunakan waktu standar dunia sebagai satuan waktunya. Pada permainan ini pemain diminta untuk membangun suatu desa atau kastil. Pemain kemudian juga membangun pasukan untuk digunakan untuk menyerang desa atau kastil pemain lain. Jika

pemain telah menyerang desa dari pemain lain dan pemain memenangkan pertempuran dengan memenuhi salah sebagian atau seluruh dari 3 syarat kemenangan, pemain akan mendapatkan trophy. Ketiga syarat tersebut adalah.

1. Pemain menghancurkan 50% dari total semua bangunan lawan.
2. Pemain menghancurkan TownHall lawan.
3. Pemain menghancurkan seluruh bangunan lawan.

Untuk setiap syarat diatas yang terpenuhi pemain akan mendapatkan 1 bintang. Bintang ini merupakan representasi dari kesuksesan penyerangan tersebut dan semakin banyak bintang yang didapat oleh pemain semakin banyak pula trophy yang akan didapatkan. Pemain yang kalah pada suatu pertempuran, trophy nya akan dikurangi sejumlah dengan trophy yang didapatkan oleh pemain yang menang. Tujuan utama dari permainan ini adalah untuk mengumpulkan trophy sebanyak-banyaknya dan terus bersaing dengan pemain lain. Terdapat suatu liga atau klasemen-klasemen yang memperlihatkan statistik trophy dan kemenangan pemain untuk memperlihatkan pemain yang terbaik.

Selain trophy pemain juga akan mendapatkan resource yang dimiliki pemain lawan. Dengan menghancurkan Gold Storage, Elixir Storage, Dark Elixir Storage, dan kolektor-kolektornya. Sampai dengan Clash of Clans versi tertanggal 31 November 2013 aturan yang digunakan dalam perebutan resource ini adalah pemain mendapatkan 25% dari total tampungan seluruh storage yang dihancurkan dengan nominal maksimal ada 200000. Pemain juga bisa mendapatkan resource yang masih tersimpan pada kolektor yang sebesar 50% dari resource yang sedang ditampung oleh kolektor. Mekanisme ini menjadi suatu bagian yang penting karena kemampuan kolektor yang dimiliki oleh pemain didesain tidak dapat memenuhi kebutuhan pembangunan dan upgrade pemain.

Fig 2.1 Desa pada game Clash of Clans

Dalam perjalananya pemain dapat mengupgrade bangunan dan pasukan yang dimilikinya yang kemudian kemampuanya semakin meningkat. Bangunan ini digunakan untuk mempertahankan desa, membuat pasukan, mengumpulkan resource dan yang lainya. Berikut adalah daftar beberapa structure yang dapat dibangun oleh pemain dalam permainan ini.

1. Cannon

Bangunan yang digunakan untuk mempertahankan desa, dapat menyerang pasukan musuh yang masuk kedalam jarak tembaknya. Single target attack.

2. Mortar

Bangunan yang digunakan untuk pertahanan desa, dapat menyerang pasukan musuh yang masuk kedalam jarak tembaknya. Splash damage sebanyak 9 tile.

3. Gold Mine

Bangunan collector gold yang digunakan untuk mengumpulkan gold. Bangunan ini akan menambang gold dengan rate yang sesuai dengan level bangunan ini. Gold mine dengan level tertinggi yaitu 11 akan menambang 3000 gold dalam kurun waktu 1 jam.

4. Gold Storage

Bangunan untuk menyimpan gold pemain. Pemain tidak dapat memiliki gold melebihi dari daya tampung total semua gold storage nya.

Seperti yang telah dijelaskan diatas pemain membutuhkan resource untuk membangun desanya. Resource ini dapat dibagi menjadi 3 yaitu sumber daya alam dan sumber daya manusia. Sumber daya alam disini adalah gold, elixir, dan dark elixir yang mana dapat dianalogikan sebagai uang pada dunia nyata. Sumber daya manusia pada konteks ini adalah builder atau unit yang digunakan untuk membangun atau mengupgrade bangunan. Jumlah builder yang diberikan pada awal permainan ini adalah 2, pemain dapat menambah jumlah builder ini sampai maksimal 5 unit dengan cara membelinya dengan gem. Gem adalah satuan resource khusus yang dapat pemain dapatkan dengan membelinya secara online dengan mata uang nyata (rupiah atau US$).Pada proses pembangunan atau upgrade structure pemain membutuhkan sejumlah waktu tertentu sampai bangunan dapat selesai dibuat atau diupgrade. Pada proses pembangunan Gold Mine level 1 contohnya waktu yang dibutuhkan adalah selama 5 menit, sedangkan untuk upgrade gold mine level 11 pemain membutuhkan waktu selama 4 hari.

III. IMPLEMENTASI GREEDY

Untuk membantu pemain dalam mendapatkan hasil pengembangan desa yang paling optimal, dalam konteks ini pengembangan yang paling optimal adalah pengembangan yang membutuhkan waktu paling cepat. Untuk permasalahan ini algoritma greedy dapat digunakan menjadi core scheduler.

Seperti yang telah diutarakan sebelumnya bahwa pada permainan ini pemain membutuhkan resource untuk dapat membangun atau mengupgrade suatu bangunan seperti gold, elixir, dan dark elixir. Pada permainan ini terdapat

resource khusus yaitu gem yang dapat dibeli oleh pemain dengan menggunakan mata uang nyata seperti rupiah dan US$. Dengan menggunakan gem ini pemain dapat menggunakan gem untuk melakukan perintah luar biasa seperti mempersingkat waktu pembuatan structure, menukar gem dengan resource lain seperti gold, elixir, dan dark elixir, dan meningkatkan mining rate dari Gold Mine dan Elixir Collector. Akan tetapi penggunaan mata uang nyata sebagai metode memenangkan game ini tidak bisa dilakukan oleh setiap orang karena tidak setiap orang dapat atau mau mengeluarkan banyak mengeluarkan uang mereka. Untuk itu pada makalah ini akan dibuat batasan-batasan khusus agar scheduler ini dapat digunakan oleh semua orang secara general. Kemudian berikutnya mempertimbangkan faktor penyerangan antar pemain dimana dapat berakibat kurang nya resource yang dimiliki oleh pemain untuk mengupgrade suatu bangunan maka pada makalah ini akan dibuat batasan masalah dimana pemain diasumsikan memiliki jumlah resource (Sumber daya alam seperti gold, elixir, dan dark elixir) yang tak terbatas. Sehingga concern yang perlu dipertimbangkan selanjutnya hanyalah Sumber daya manusia (builder) dan waktu nya saja. Jumlah builder tentunya juga akan berpengaruh dalam proses scheduling ini akan tetapi dengan menggunakan proses yang tepat scheduler dapat memberikan keputusan sesuai dengan rule yang diberikan.

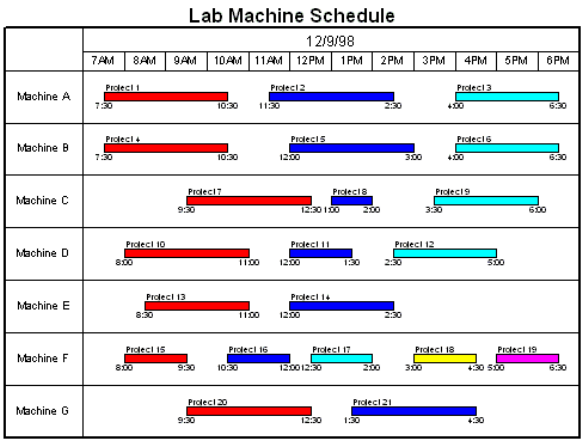

Fig 3.1 Contoh Machine Scheduler

Pada scheduler ini yang pertama harus diperhatikan adalah input dari fungsi yang dibuat. Elemen-elemen yang terdapat pada permainan Clash of Clans harus disesuaikan dengan algoritma yang akan digunakan untuk itu dilakukan sebuah pemrosesan awal atau preprocessing pada elemen-elem clash of clans yang akan dijadikan input pada algoritma greedy ini. Seperti yang telah dijelaskan sebelumnya, karena tujuan utama pada scheduler ini adalah mendapatkan jadwal atau schedule yang paling optimum secara waktu maka seluruh data durasi waktu pembuatan dan upgrade bangunan disimpan terlebih dahulu. Pada kali ini seluruh data akan disimpan pada sebuah file xml yang dapat diubah dan diedit sewaktu-waktu jika terjadi perubahan. Metode penyimpanan ini dipilih mempertimbangkan sifat utama game dimana developer akan selalu melakukan post-launch development untuk memperbaiki kekurangan-kekurangan yang dimiliki dan melakukan balancing-balancing untuk terus meningkatkan kualitas dari game tersebut.

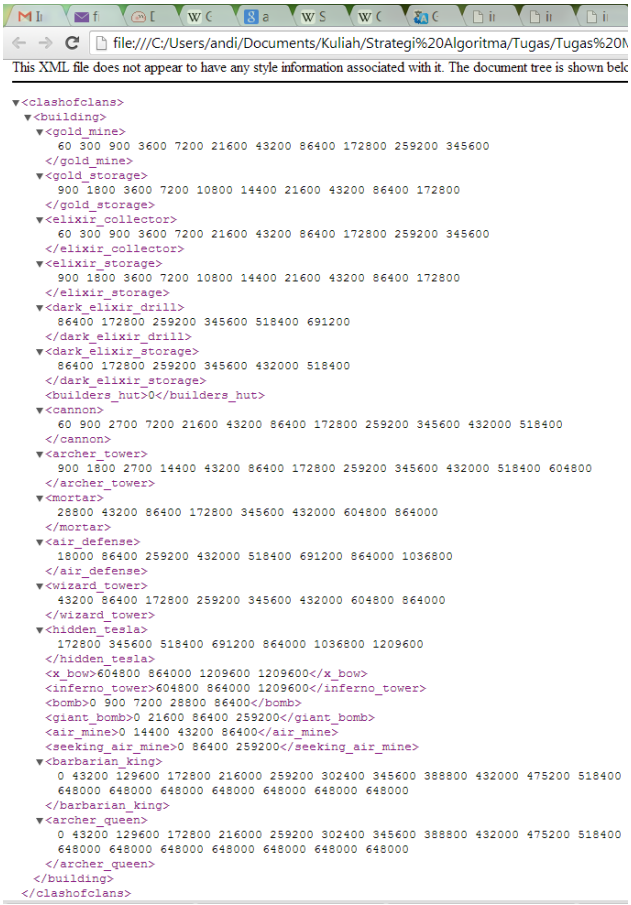

Fig 3.2 Clash of Clans Planner XML Database

Dalam scheduler ini input yang digunakan tentunya adalah daftar pekerjaan-pekerjaan yang akan dilakukan oleh pemain. Karena memang dalam game ini pemain dapat saja melakukan upgrade berkali-kali pada satu bangunan yang sama, maka untuk menangani masalah ini akan dilakukan preprocessing dengan menyatukan aksi atau proses upgrade pada bangunan yang sama menjadi satu proses sekuens dengan upgrade yang lebih dulu(yang level nya lebih kecil) dikerjakan terlebih dahulu. Misalkan pemain ingin mengupgrade Archer Tower nya dari level 1 ke level 2 dan dilanjutkan dengan upgrade ke level 3 (archer tower yang sama), maka pekerjaan ini akan disatukan menjadi pekerjaan yang sekuens. Dapat dianalogikan bahwa proses preprocessing ini adalah proses pembuatan suatu pita-pita pekerjaan yang akan dilakukan. Penyatuan pita-pita ini diperlukan untuk menghindari kesalahan pada proses scheduling lebih lanjut. Pita-pita ini kemudian masukkan kedalam suatu array atau dapat juga dengan menggunakan list sebagai himpunan kandidat.

```
    Job Scheduler::SelectCandidate() {
            Job tempCandidate(CandidateSet[0]);
            for(int i=1;i<nJob;i++) {
            if(CandidateSet[i].getDuration()>tempCandidate.getDuration
()) {
                                    tempCandidate = CandidateSet[i];
                    }
            }
            return tempCandidate();
}
```

Karena proses upgrade yang dilakukan membutuhkan builder sebagai unit yang melakukan upgrade ini. Maka disini dapat dianalogikan seorang builder sebagai sebuah mesing yang melakukan pekerjaan atas pita-pita pekerjaan tersebut. Dengan kata lain builder adalah sebuah mesin. Algoritma ini berjalan dengan memilih pekerjaan dari himpunan kandidat yang ada dengan menggunakan sebuah fungsi seleksi. Fungsi seleksi disini adalah memilih kandidat pekerjaan yang memiliki durasi yang paling lama. Pita pekerjaan yang memiliki kwatu paling lama ini akan dimasukkan kedalam list pekerjaan yang akan dikerjakan oleh mesin yang memiliki idle paling lebih dulu. Selanjutnya proses dilakukan secara rekursif sampai pita-pita pekerjaan pada himpunan kandidat sudah habis.

```
    int Scheduler::IdleEarliest() {
            Builder tempBuilder(BuilderSet[0]) {
            for(int i=1;i<nJBuilder;i++) {
                    if(BuilderSet[i].getIdle()<tempBuilder.getIdle()) {
                            tempBuilder = BuilderSet[i];
                    }
            }
            return tempBuilder.getID();
}
```

```
    Job* Scheduler::Schedule() {
            if(isCandidateEmpty()) {
                    return SolutionSet;
            }
            else {
                    PushSolution(IdleEarliest(), SelectCandidate());
            }
}
```

Output yang dihasilkan pada algoritma ini adalah scheduler pekerjaan yang harus dilakukan oleh mesin-mesin atau dalam hal ini builder yang tersedia. Hasil output ini dapat disajikan dalam satuan waktu nyata atau dapat juga dengan lebih sederhana yaitu sebagai sekuens pekerjaan yang harus dilakukan.

Keluaran ini kemudian dapat diproses lebih lanjut menyesuaikan dengan kebutuhan user itu sendiri. Apakah user merupakan user yang sudah cukup dewasa untuk dapat memahami output sederhana ataukah user menginginkan sesuatu dengan kurva pemahaman yang lebih rendah. Hal ini tidak akan dibahas lebih lanjut pada makalah ini karena konsentrasi utama pada makalah ini hanya menitikberatkan pada implementasi algoritma greedy pada scheduling permainan Clash of Clans ini saja.

Fig 3.3 Top Klasemen Clash of Clans

IV. PENJELASAN PENGGUNAAN

Untuk menggunakan alat bantu ini hal yang perlu diperhatikan oleh user sebagai berikut :

1. Masukkan input pekerjaan yang akan dilakukan.
2. Mulai proses penjadwalan.
3. Pahami dan lakukan hasil jadwal yang diberikan oleh aplikasi untuk hasil yang paling optimum secara waktu.

V. KESIMPULAN

Algoritma greedy dapat diimplementasikan dengan cukup mudah dan sederhana pada machine scheduling dengan constraint dan resource yang sederhana. Algoritma ini akan memberikan hasil optimum global pada kasus-kasus khusus yang sederhana. Salah satu kasus yang menghasilkan hasil optimum global jika diimplementasikan algoritma greedy adalah scheduling clash of clans ini.

Banyak algoritma lain yang lebih kompleks yang dapat digunakan untuk mendapatkan solusi paling optimal pada masalah machine-scheduling ini. Akan tetapi domain yang cukup sederhana membuat algoritma greedy lebih dipilih karena lebih mudah untuk diimplementasi demi mendapatkan hasil yang optimal. Masih banyak lagi aspek dari permainan Clash of Clans ini yang dapat dibantu dengan berbagai implementasi greedy seperti optimasi pembuatan pasukan yang membutuhkan waktu seperti pembangunan struktur.

REFERENCES

[1] G. Bendall and F. Margot, Greedy Type Resistance of Combinatorial Problems, Discrete Optimization 3 (2006), 288–298.

[2] Introduction to Algorithms (Cormen, Leiserson, and Rivest) 1990, Chapter 17 "Greedy Algorithms" p. 329.

[3] Black, Paul E. (2 February 2005). "greedy algorithm". Dictionary of Algorithms and Data Structures. U.S. National Institute of Standards and Technology (NIST). Retrieved 17 August 2012.
[4] G. Gutin, A. Yeo and A. Zverovich, Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. Discrete Applied Mathematics 117 (2002), 81–86.
[5] J. Bang-Jensen, G. Gutin and A. Yeo, When the greedy algorithm fails. Discrete Optimization 1 (2004), 121–127.
[6] http://www.supercell.net/games/view/clash-of-clans
[7] http://blogs.mervpolis.com/roller/yudhy/entry/pengertian_metode_greedy_dan_algoritma

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, 18 Desember 2013

Fitrandi Ramadhan, 13508065