

APLIKASI ALGORITMA GREEDY DAN PROGRAM DINAMIS (*DYNAMIC PROGRAMMING*) PADA PERMAINAN *GREEDY SPIDERS*

Ika Zulhidayati ⁽¹⁾,
Kartika Yulianti ⁽²⁾

ABSTRAK

Semakin pesatnya perkembangan teknologi, perkembangan *games* pun semakin maju. *Greedy Spiders* merupakan salah satu *game* untuk *smartphone* yang berbasis seperti Android yang dikembangkan oleh Blyts sebuah *software house*. Pada *game* ini terdapat laba-laba yang ingin memakan serangga yang terjebak di jaring laba-laba tersebut. Laba-laba tersebut merupakan *Artificial Intelligence* (AI) pada permainan ini, yang akan menjadi lawan bagi pemain dalam menyelesaikan permainan ini. Tugas pemain adalah menyelamatkan serangga yang terjebak sehingga laba-laba tidak dapat memakan serangga tersebut dengan cara memutuskan beberapa sisi pada jaring laba-laba. Algoritma Greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Algoritma Greedy membentuk solusi langkah per langkah. Program Dinamis (*Dynamic Programming*) adalah suatu metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. Pada program dinamis, rangkaian keputusan yang optimal dibuat dengan menggunakan Prinsip Optimalitas, dengan prinsip optimalitas ini dijamin bahwa pengambilan keputusan pada suatu tahap adalah keputusan yang benar untuk tahap-tahap selanjutnya. Algoritma Greedy digunakan untuk mengetahui pergerakan laba-laba, kemudian program dinamis (*dynamic programming*) pada permainan *Greedy Spiders* digunakan untuk mencari solusi optimal dari pergerakan laba-laba ke serangga, yaitu mencari jalur terpendek dari pergerakan laba-laba ke serangga sehingga pemain dapat memutuskan jaring mana yang akan dipotong untuk memperoleh hasil yang maksimal atau dalam hal ini pemain dapat memperoleh skor tertinggi.

Kata Kunci: Algoritma Greedy, Program Dinamis (*Dynamic Programming*), serangga, jaring laba-laba, laba-laba.

Pendahuluan

Kehadiran berbagai perangkat *smartphone* ternama berbasis Android dipasaran tampaknya sudah berhasil menjangkau berbagai kalangan pengguna. Android merupakan sistem operasi besutan Google. Sebanyak 39,63 persen pembuat *games* yang disurvei mengaku tertarik untuk mengembangkan *game* pada sistem operasi tersebut (tekno.liputan6.com). Seiring majunya perkembangan teknologi, perkembangan *games* pun seolah tidak mau ketinggalan. Para vendor bersaing membuat *games* yang dapat merebut hati para *gamers* melalui *games* besutannya. Para penyuka *games* pun semakin mewabah dan

memasuki berbagai kalangan umur. *Greedy Spiders* merupakan salah satu *game* untuk *smartphone* yang berbasis seperti Android. Pada *game* ini terdapat laba-laba yang ingin memakan serangga yang terjebak di jaring laba-laba tersebut. Laba-laba tersebut merupakan *Artificial Intelligence* (AI) pada permainan ini yang akan menjadi lawan bagi pemain dalam menyelesaikan permainan ini. Tugas pemain adalah menyelamatkan serangga yang terjebak sehingga laba-laba tidak dapat memakan serangga tersebut.

Dalam kehidupan sehari-hari, banyak persoalan yang menuntut pencarian solusi optimum. Persoalan optimasi adalah persoalan yang tidak hanya sekedar mencari solusi, melainkan mencari solusi terbaik. Algoritma Greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Algoritma Greedy membentuk solusi langkah per langkah. Sesuai dengan arti tersebut, algoritma ini akan mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan. Program Dinamis (*Dynamic Programming*) adalah suatu metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. Pemograman dinamis menggambarkan proses pemecahan masalah untuk menemukan suatu keputusan yang terbaik. Program dinamis akan memecahkan masalah, kemudian solusi optimal yang dihasilkan dapat dipandang sebagai suatu deret keputusan dari masalah tersebut. Untuk menyelesaikan *game* ini penulis akan membahas aplikasi Algoritma Greedy untuk menentukan bagaimana pergerakan laba-laba yang merupakan AI dalam permainan ini dengan objektif untuk mendekati serangga yang terperangkap dan Program Dinamis (*Dynamic Programming*) untuk mencari solusi optimal dalam menyelesaikan permainan ini, yang akan dibahas pada bab selanjutnya.

Greedy Spiders

Greedy Spiders merupakan salah satu *game* untuk *smartphone* yang berbasis seperti Android yang dikembangkan oleh Blyts sebuah *software house*. Konsep *puzzle game* yang sangat sederhana, dengan tantangannya yang sulit

kemudian dikemas dengan tampilan *game* yang menyenangkan membuat *game* ini populer. *Game* ini seperti gabungan antara catur dan tebak-tebakan angka, karena dalam penyelesaiannya harus memikirkan langkah-langkah yang dibuat dengan hati-hati.

Pada *game* ini terdapat laba-laba yang ingin memakan serangga yang terjebak di jaring laba-laba tersebut. Tugas pemain adalah menyelamatkan serangga yang terjebak sehingga laba-laba tidak dapat memakan serangga tersebut. Laba-laba yang merupakan lawan pemain akan mencari jalan terbaik untuk mendekati serangga yang akan dimangsanya, sebelum pemain dapat membebaskan serangga yang terjebak di jaring laba-laba. Apabila pemain dapat membebaskan serangga sebelum dimakan oleh laba-laba, maka pemain sudah memenangkan level tersebut dan dapat melanjutkan ke level selanjutnya. Akan tetapi, jika pemain gagal membebaskan serangga yang terjebak sehingga serangga dapat dimangsa oleh laba-laba maka pemain akan kalah dan tidak dapat melanjutkan level selanjutnya.

Konsep Dasar Teori Graf

Definisi Graf

Untuk memahami teori graf kita perlu mengetahui beberapa notasi dan terminologi yang digunakan.

Definisi 2.2.1

Sebuah graf G adalah sebuah himpunan terhingga tak kosong yang memuat obyek-obyek yang disebut **simpul** (titik atau *verteks*), dan himpunan pasangan tak urut antara simpul-simpul yang berlainan, yang disebut **sisi** (rusuk atau *edge*). Himpunan simpul dari graf G ditulis dengan $V(G)$, sedangkan himpunan sisi dari G dinyatakan dengan $E(G)$ (Kusuma, 1997).

Lintasan

Sebuah jalan dalam graf G adalah sebuah urutan tak nol $W = v_0e_1v_1e_2v_2\dots e_iv_i\dots e_kv_k$, yang suku-sukunya bergantian antara simpul dan sisi, demikian sehingga $1 \leq i \leq k$, ujung dari e_i adalah v_{i-1} dan v_i .

W adalah sebuah jalan dari v_0 (simpul asal) ke v_k (simpul terminus), atau cukup ditulis dengan jalan (v_0, v_k) . Banyaknya sisi pada W merupakan panjang jalan tersebut. Jika simpul-simpul $v_0e_1v_1e_2v_2\dots e_iv_i\dots e_kv_k$ dari jalan W berlainan, maka W disebut **lintasan** (Kusuma, 1997).

Sisi Pemotong

Pada pohon terdapat konsep sisi pemotong dari sebuah graf, yaitu sisi yang penghapusannya membuat graf tersebut menjadi tak terhubung.

Definisi 2.4.1 :

Sebuah sisi pemotong pada graf G adalah sebuah sisi e, demikian sehingga $\omega(G - e) > \omega(G)$, dengan $\omega(G)$ sebagai banyaknya komponen dari G. Pada Gambar 2-0-9 sisi yang tampak lebih tebal adalah sisi-sisi pemotong dari graf G. Jika salah satu dari keempat sisi tersebut dihapus, G akan menjadi graf tidak terhubung (Kusuma, 1997).

Keterhubungan sisi

Persoalan-persoalan dalam teori graf dapat diformulasikan ke- dalam topik keterhubungan sisi. Dalam konsep ini, persoalan diterjemahkan sebagai pertanyaan, berapa sisikah paling sedikit yang dapat dihapus agar sebuah graf menjadi graf tak terhubung. Dalam penulisan ini dibahas mengenai keterhubungan sisi, karena hanya pemutusan sisi saja yang dilakukan untuk menyelesaikan permainan *Greedy Spiders*.

Definisi 2.5.1.1

Keterhubungan sisi pada graf G, yang dilambangkan dengan $\lambda(G)$, adalah banyaknya sisi paling sedikit yang dapat dihapus, demikian sehingga graf G menjadi graf tak terhubung. Jika $\lambda(G) \geq k$, maka G disebut graf terhubung dalam k-sisi.

Definisi 2.5.1.2

Sebuah himpunan pemotong (*Cutset*) pada sebuah graf terhubung G adalah sebuah himpunan S yang memuat sisi-sisi dengan sifat-sifat berikut:

- a. Penghapusan semua sisi pada S membuat G menjadi tak terhubung,
- b. Penghapusan beberapa sisi pada S tapi tidak semuanya tidak mengakibatkan G menjadi tak terhubung (Kusuma, 1997).

Algoritma Greedy

Algoritma Greedy merupakan metode yang paling populer dalam memecahkan persoalan optimasi. Hanya ada dua macam persoalan optimasi, yaitu maksimasi dan minimasi. Algoritma Greedy adalah algoritma yang memecahkan masalah langkah per langkah (Munir).

Skema Umum Algoritma Greedy

Persoalan optimasi dalam konteks algoritma greedy disusun oleh elemen-elemen sebagai berikut :

1. Himpunan kandidat, C .
Himpunan ini berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.
2. Himpunan solusi, S .
Himpunan ini berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan solusi adalah himpunan bagian dari himpunan kandidat.
3. Fungsi seleksi (*selection function*)
Fungsi ini dinyatakan dengan predikat seleksi. Merupakan fungsi yang pada setiap langkah 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 (*feasible*)
Fungsi ini dinyatakan dengan predikat layak. Fungsi kelayakan ini merupakan fungsi yang 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 yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

5. Fungsi obyektif

Fungsi obyektif ini merupakan sebuah fungsi yang memaksimalkan atau meminimumkan nilai solusi (Munir).

Program Dinamis (Dynamic Programming)

Di dalam ilmu matematika dan ilmu komputer, program dinamis (*dynamic programming*) didefinisikan sebagai suatu metode penyelesaian masalah untuk permasalahan yang memiliki sifat-sifat *overlapping subproblems* dan *optimal substructure*. Program Dinamis digunakan untuk menggambarkan proses penyelesaian masalah dimana solusi yang diinginkan harus benar-benar paling optimum atau biasanya disebut sebagai optimum global (*global optimum*). (Suyanto, 2010).

Konsep Dasar

Optimal substructure solusi optimal untuk submasalah-submasalah (*subproblems*) dapat digunakan untuk menemukan solusi optimal yang utuh untuk masalah yang dihadapi. Sebagai contoh, dalam suatu graf, jalur terpendek (*shortest path*) dari suatu simpul asal (misalnya S) menuju *goal* (misalnya G) dapat ditemukan dengan terlebih dulu menghitung sub jalur terpendek dari semua simpul tetangga S menuju *goal*, dan selanjutnya menggunakan sub jalur terpendek tersebut untuk mendapatkan jalur terpendek yang utuh.

Secara umum, untuk menyelesaikan suatu masalah dengan *optimal substructure* dapat digunakan tiga langkah berikut :

1. Pecah masalah menjadi submasalah-submasalah yang lebih kecil
2. Selesaikan submasalah-submasalah secara optimal menggunakan tiga langkah ini secara rekursif
3. Gunakan solusi-solusi optimal untuk submasalah-submasalah tersebut untuk membangun solusi optimal yang utuh untuk masalah yang dihadapi.

Program dinamis biasanya menggunakan salah satu dari dua pendekatan berikut ini :

- ***Top-down***

Masalah dipecah menjadi beberapa submasalah, kemudian submasalah-submasalah tersebut diselesaikan dan solusi-solusinya diingat (dicatat) agar dapat digunakan kembali. Dalam hal ini, *recursion* dan *memoization* dikombinasikan secara bersamaan.

- ***Bottom-up***

Semua submasalah-submasalah, yang mungkin diperlukan, diselesaikan terlebih dahulu dan kemudian digunakan untuk membangun solusi-solusi untuk masalah-masalah yang lebih besar. Pendekatan ini sedikit lebih baik dalam hal ruang tumpukan (*stack space*) dan jumlah *function calls*, tetapi kadang-kadang tidak intuitif untuk menggambarkan semua submasalah yang diperlukan untuk menyelesaikan masalah yang diberikan (Suyanto, 2010).

Skema Umum Program Dinamis

Program Dinamis (*Dynamic Programming*) merupakan suatu metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. Karakteristik penyelesaian persoalan dengan program dinamis:

1. Terdapat sejumlah berhingga pilihan yang mungkin
2. Solusi pada setiap tahap dibangun dari hasil solusi tahap sebelumnya
3. Menggunakan persyaratan optimasi dan kendala untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap (Munir).

Prinsip Optimalitas

Dikatakan prinsip optimalitas, jika solusi total optimal, maka bagian solusi sampai tahap ke- k juga optimal. Prinsip optimalitas berarti bahwa jika dari

tahap k ke tahap $k+1$, dapat menggunakan hasil optimal dari tahap k tanpa harus kembali ke tahap awal.

$$\begin{aligned} \text{Ongkos pada tahap } k+1 = & \text{(ongkos yang dihasilkan pada tahap } k) \\ & + \\ & \text{(ongkos dari tahap } k \text{ ke tahap } k+1) \end{aligned} \quad (1)$$

Dengan prinsip optimalitas ini dijamin bahwa pengambilan keputusan pada suatu tahap adalah keputusan yang benar untuk tahap-tahap selanjutnya (Munir).

Karakteristik Persoalan Program Dinamis

Ada beberapa karakteristik persoalan yang dalam penyelesaian dengan menggunakan program dinamis yaitu:

1. Persoalan dapat dibagi menjadi beberapa tahap (*stage*), yang pada setiap tahap hanya diambil satu keputusan.
2. Masing-masing tahap terdiri dari sejumlah status (*state*) yang berhubungan dengan tahap tersebut. Secara umum, status merupakan bermacam kemungkinan masukan yang ada pada tahap tersebut.
3. Hasil dari keputusan yang diambil pada setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. Ongkos (*cost*) pada suatu tahap meningkat secara teratur (*steadily*) dengan bertambahnya jumlah tahapan.
5. Ongkos pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah berjalan dan ongkos pada tahap tersebut.
6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya.
7. Adanya hubungan rekursif yang mengidentifikasi keputusan terbaik untuk setiap status pada tahap k memberikan keputusan terbaik untuk setiap status pada tahap $k+1$.
8. Prinsip optimalitas berlaku pada persoalan tersebut.

Dua Pendekatan Program Dinamis

Dua pendekatan yang digunakan dalam program dinamis:

1. Program dinamis maju (*forward* atau *up-down*)
2. Program dinamis mundur (*backward* atau *bottom-up*)

Misalkan x_1, x_2, \dots, x_n menyatakan peubah (*variable*) keputusan yang harus dibuat masing-masing untuk tahap 1, 2, ..., n . Maka,

1. Program dinamis maju. Program dinamis bergerak mulai dari tahap 1, terus maju ke tahap 2, 3, dan seterusnya sampai tahap n . Runtunan peubah keputusan adalah x_1, x_2, \dots, x_n .
2. Program dinamis mundur. Program dinamis bergerak mulai dari tahap n , terus mundur ke tahap $n-1, n-2$, dan seterusnya sampai tahap 1. Runtunan peubah keputusan adalah x_n, x_{n-1}, \dots, x_1 .

- Prinsip optimalitas pada program dinamis maju:

$$\text{ongkos pada tahap } k+1 = (\text{ongkos yang dihasilkan pada tahap } k) + (\text{ongkos dari tahap } k \text{ ke tahap } k+1), \quad (2)$$

dengan $k = 1, 2, \dots, n-1$.

- Prinsip optimalitas pada program dinamis mundur:

$$\text{ongkos pada tahap } k = (\text{ongkos yang dihasilkan pada tahap } k+1) + (\text{ongkos dari tahap } k+1 \text{ ke tahap } k), \quad (3)$$

dengan $k = n, n-1, \dots, 1$ (Munir).

Langkah-langkah Pengembangan Algoritma Program Dinamis

Adapun bagaimana langkah-langkah dalam pengembangan algoritma program dinamis:

1. Karakteristikkan struktur solusi optimal.
2. Definisikan secara rekursif nilai solusi optimal.
3. Hitung nilai solusi optimal secara maju atau mundur.
4. Konstruksi solusi optimal.

Studi Kasus

Deskripsi Masalah

Jaring laba-laba diasumsikan sebagai graf yang memiliki bobot yang sama. Laba-laba akan mencari jarak terpendek dari titik dia berada ke titik dimana serangga berada dan bergerak ke titik dimana kemungkinan dia dapat memakan serangga lebih besar. Sedangkan pemain akan memutuskan jaring laba-laba yang mungkin akan dilalui oleh laba-laba tersebut, dengan cara memotong jaring seminimal mungkin agar memperoleh skor tertinggi.

Setelah mengetahui titik-titik kemungkinan terbesar dengan menggunakan algoritma greedy, dengan begitu jalur terpendek yang akan dilalui oleh laba-laba untuk mencapai serangga yang akan dimangsanya dapat diketahui. Saat titik kemungkinan terbesar dapat dicari, dengan begitu pemain dapat memutuskan jaring mana yang terbaik yang akan dipotong sehingga laba-laba tidak dapat mendekati serangga tersebut. Memilih jaring mana yang terbaik untuk dipotong dimaksudkan agar, jaring yang dipotong dapat seminimal mungkin sehingga pemain dapat menyelamatkan semua serangga dan dapat memperoleh skor tertinggi. Untuk itulah digunakan program dinamis (*dynamic programming*) untuk mencari jalur terpendek dengan begitu pemotongan jaring dapat dilakukan seminimal mungkin.

Aplikasi Algoritma Greedy dan Program Dinamis

Penerapan algoritma greedy digunakan untuk mengetahui titik kemungkinan terbesar dan mengetahui pergerakan laba-laba atau jalur yang akan dilalui oleh laba-laba untuk mendekati titik kemungkinan tersebut, kemudian program dinamis (*dynamic programming*) pada permainan *Greedy Spiders* digunakan untuk mencari solusi optimal dari pergerakan laba-laba ke serangga, yaitu mencari jalur terpendek dari pergerakan laba-laba ke serangga sehingga pemain dapat memutuskan jaring mana yang akan dipotong untuk memperoleh hasil yang maksimal atau dalam hal ini pemain dapat memperoleh skor tertinggi.

Untuk mengetahui pergerakan laba-laba yang mencari serangga terbanyak yang dapat ia makan dengan menggunakan algoritma greedy, perhatikan seluruh jaring laba-laba yang terdapat serangga, kemudian setelah mengetahui jaring-

jaring di mana serangga berada, aplikasi algoritma greedy tersebut adalah sebagai berikut:

- Himpunan Kandidat
Seluruh titik pada jaring laba-laba yang dapat ditempati laba-laba untuk mendekati serangga.
- Himpunan Solusi
Titik yang memiliki dua sisi atau lebih yang terdapat serangga yang dapat dimakan oleh laba-laba.
- Fungsi Layak
Berupa aturan, bilamana titik yang memiliki dua sisi atau lebih yang terdapat serangga yang dapat dimakan oleh laba-laba, memiliki minimal satu serangga yang dapat dimakan. Titik tersebut merupakan titik solusi.

Setelah diperoleh titik kemungkinan terbesar dan sisi yang berjarak satu langkah dari serangga, akan dicari jarak terdekat dari posisi laba-laba berada ke serangga. Pencarian jarak terdekat dari posisi laba-laba ke serangga dimaksudkan agar pemotongan jaring dapat dilakukan seminimal mungkin, untuk pencarian jarak terdekat ini digunakan program dinamis (*dynamic programming*).

Langkah-langkah pengembangan program dinamis (*dynamic programming*) pada persoalan ini, adalah sebagai berikut:

- Tahap (k) adalah proses memilih sisi tujuan berikutnya.
- Status (s) yang berhubungan dengan masing-masing tahap adalah sisi-sisi di dalam graf.
- Hitung nilai solusi optimal secara maju atau mundur. Program dinamis yang digunakan adalah program dinamis maju, program dinamis bergerak mulai dari tahap 1, terus maju ke tahap 2, 3, dan seterusnya sampai tahap n.

Seperti yang telah dijelaskan sebelumnya, jaring laba-laba diasumsikan sebagai graf yang memiliki bobot yang sama.

Tahap 1 :

Algoritma Greedy : dari algoritma greedy diperoleh titik kemungkinan terbesar, yaitu titik yang memiliki 2 sisi atau lebih yang menghubungkan ke 2 atau lebih serangga.

Tahap 2 :

Program Dinamis : dari program dinamis akan diperoleh *shortest path*.

Tampilan *Hint* dengan Program Delphi

Menggunakan program Delphi akan ditampilkan *hint* dari level-level permainan yang telah dibahas, yaitu level-level permainan setelah diaplikasikan oleh kedua algoritma (algoritma greedy dan program dinamis).

Kesimpulan dan Saran

Kesimpulan

1. Pada metode algoritma greedy hanya satu rangkaian keputusan yang pernah dihasilkan, oleh karena itu mengapa algoritma greedy digunakan untuk menentukan bagaimana pergerakan laba-laba yang merupakan AI dalam permainan *Greedy Spiders*. Sedangkan pada metode program dinamis lebih dari satu rangkaian keputusan yang dihasilkan. Hanya rangkaian keputusan yang memenuhi prinsip optimalitas yang akan dihasilkan, oleh karena itu mengapa program dinamis digunakan untuk mencari solusi optimal, yaitu usaha untuk mencapai skor tertinggi dalam menyelesaikan suatu level dengan cara pemutusan jaring laba-laba seminimal mungkin dalam menyelesaikan permainan ini. Sehingga pemain dapat menyelesaikan permainan ini dengan skor tertinggi yaitu memperoleh 3 bintang pada setiap levelnya.
2. Jaring laba-laba diasumsikan sebagai graf yang memiliki bobot yang sama. Laba-laba akan mencari jarak terpendek dari titik dia berada ke titik dimana serangga berada dan bergerak ke titik dimana kemungkinan dia dapat memakan serangga lebih besar. Pemain akan memutuskan jaring laba-laba yang mungkin akan dilalui oleh laba-laba tersebut. Titik dimana laba-laba memiliki kemungkinan memakan serangga lebih besar adalah ketika laba-laba berada pada titik dimana dia memiliki dua atau lebih serangga yang dapat dimangsa dalam satu waktu dengan sekali langkah. Apabila laba-laba berada di titik tersebut dengan sekali langkah lagi, laba-laba dapat memilih dua serangga untuk dimakan yang dapat menyebabkan

permainan berakhir dan pemain gagal dalam menyelesaikan level tersebut. Oleh karena itu digunakan algoritma greedy, untuk mengetahui titik-titik tersebut dalam setiap level permainan pada level dasar *time to eat*. Semakin tinggi suatu level, terdapat level-level yang memiliki titik kemungkinan terbesar lebih dari satu titik. Setelah mengetahui titik-titik tersebut dengan menggunakan algoritma greedy, dengan begitu jalur terpendek yang akan dilalui oleh laba-laba untuk mencapai titik tersebut dapat diketahui. Maka pemain dapat memutuskan jaring mana yang akan dipotong sehingga laba-laba tidak dapat mendekati titik tersebut. Memilih jaring mana yang terbaik untuk dipotong dimaksudkan agar, jaring yang dipotong dapat seminimal mungkin sehingga pemain dapat menyelamatkan semua serangga dan dapat memperoleh skor tertinggi. Untuk itulah digunakan program dinamis (*dynamic programming*) untuk mencari jalur terpendek dengan begitu pemotongan jaring dapat dilakukan seminimal mungkin.

Saran

1. Pada penulisan ini penerapan algoritma greedy dan program dinamis hanya pada level dasar permainan yaitu, *time to eat* yang memiliki titik kemungkinan terbesar. Pada penulisan skripsi selanjutnya kedua algoritma di atas dapat diterapkan atau diaplikasikan pada level-level berikutnya.
2. Untuk penulisan skripsi selanjutnya, algoritma greedy dan program dinamis dapat diterapkan pada permainan lainnya, tidak hanya pada *game smartphone*, dan dalam menyelesaikannya dapat membandingkan dengan algoritma lain selain kedua algoritma di atas.

DAFTAR PUSTAKA

Anonim.(2013).[Online].Tersedia:<http://tekno.liputan6.com/read/527945/dibandingkan-android-ios-lebih-diminati-pengembang-game>.

- Kusuma, Y. (1997). *Matematika Diskrit*. Bandung : Bahan Ajar IKIP Bandung dan University Press : Tidak diterbitkan.
- Jungnickel, D. (2008). *Graphs, Network and Algorithms Third Edition*. Germany : Springer.
- J. A. Bondy and Murty, U. S. R. (1976). *Graph Teory with Application*. Ontario: North-Holland
- Munir, R. *Diktat Kuliah IF2251 (Algortima Greedy) Strategi Algoritmik*. Bandung : Program Studi Informatika Sekolah Teknik Elektro dan Informatika ITB : Tidak diterbitkan.
- Bednorz, W. (2008). *Advances in Greedy Algorithms*. Croatia : In-Teh.
- Neapolitan, R dan Naimipour, K. (2011). *Foundations of Algorithms Fourth Edition*. Canada : Jones and Bartlett Publishers, LLC.
- Munir, R. *Diktat Kuliah IF3051 (Program Dinamis) Strategi Algoritmik*. Bandung : Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika ITB : Tidak diterbitkan.
- Suyanto. *Algoritma Optimasi Deterministik atau Probabilitik*. (2010). Yogyakarta : Graha Ilmu.
- Bellman, R. (1957). *Dynamic Programming*. New Jersey : Princeton University Press.
- Kamus Besar Bahasa Indonesia (KBBI) : Online.