Aplikasi Graf dalam Pembuatan Game

Felicia Christie / 13512039
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512039@std.stei.itb.ac.id

***Abstrak*—Sekarang perkembangan komunikasi dan informasi sedang sangat pesatnya, salah satu dampaknya adalah tingginya popularitas *game*. Oleh karena itu, semakin banyak pula perusahaan-perusahaan yang membuat *game* dan kemudian dijual di *application store*. Makalah ini akan membahas penggunaan struktur diskrit graf dalam pembuatan suatu *game*.**

Kata kunci – konsep game, graf.

I. PENDAHULUAN

Pada era dimana perkembangan teknologi komunikasi dan informasi sedang pesat seperti sekarang ini, masyarakat luas mempunyai banyak pilihan akan jenis rekreasi yang mereka bisa dapatkan. Dari pilihan-pilihan rekreasi tersebut, termasuk didalamnya adalah *game* yang berbasis konsol.

Game pada zaman sekarang sangatlah mudah didapat, dikarenakan maraknya penjualan game secara resmi maupun tidak melalui media internet, dibandingkan dulu dimana kita harus mempunyai *hardcopy* dari suatu game, dan mendapatkan *hardcopy* tersebut bukanlah hal yang mudah, apalagi untuk para *gamer* yang menginginkan game yang tidak dilokalisasi ke Bahasa Inggris sehingga tidak memasuki pasasr Indonesia, dan mengakibatkan mereka harus memesan dari vendor-vendor luar negeri, yang lebih banyak memakan waktu dan biaya. Kemudahan memperoleh rekreasi yang berupa *game* ini juga membantu popularitas *game* kepada masyarakat luas.

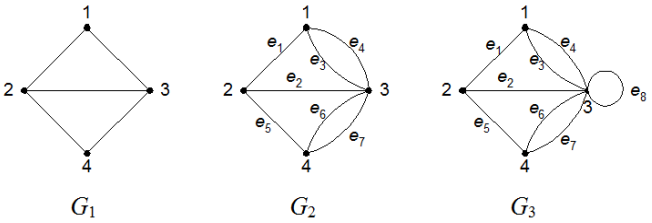
Hubungan antara *game* dengan bidang informatika yang sekarang penulis tekuni adalah, kebanyakan jenis permainan ini dimainkan di konsol, contohnya di *smartphone*, tablet PC, *PlayStation Portable*, *Nintendo DS*, dan *game* konsol tersebut tidak lain adalah hasil karya dari lulusan-lulusan jurusan informatika, atau *Computer Science*. Oleh karena ketenaran *game* dan minat penulis pada bidang ini, penulis memilih topik ini sebagai bahan makalah untuk mata kuliah Matematika Diskrit.

II. LANDASAN TEORI

Graf, secara matematis, didefinisikan sebagai pasangan himpunan simpul (disimbolkan V untuk *vertex*) dan sisi (disimbolkan dengan E untuk *edge*). Notasi matematis dari suatu graf G adalah G = (V, E). Himpunan simpul V tidak boleh kosong, sementara himpunan sisi E boleh kosong. Graf yang hanya mempunyai satu simpul tanpa sisi disebut graf trivial.

Graf dapat dikelompokkan ke dalam beberapa kategori berdasarkan kriteria tertentu. Pengelompokan bisa dilakukan berdasarkan ada atau tidaknya sisi ganda, jumlah simpul, ataupun orientasi arah sisi.

Berdasarkan ada atau tidaknya gelang dan atau sisi ganda pada suatu graf, secara umum graf dapat dikelompokkan menjadi dua jenis yaitu **graf sederhana** dan **graf tidak sederhana**.



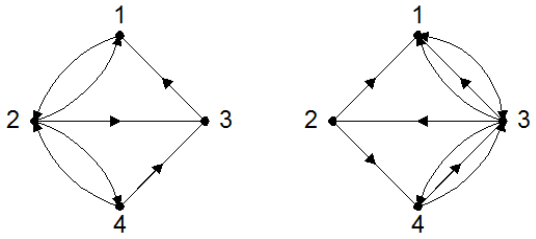
Gambar 2.1 Contoh graf berdasarkan pengelompokan adanya gelang dan sisi ganda

Graf sederhana (*simple graph*) adalah graf yang tidak mengandung gelang ataupun sisi ganda. Maksud dari sisi ganda adalah terdapat dua buah sisi yang menghubungkan sepasang simpul yang sama. Gelang adalah sisi yang menghubungkan suatu simpul dengan simpul itu sendiri. Contoh dari sebuah graf sederhana dapat dilihat pada graf G_1 pada gambar 2.1 diatas.

Graf tidak sederhana (*unsimple-graph*) adalah graf yang mengandung gelang ataupun sisi ganda. Graf tidak sederhana ini terbagi menjadi dua macam, yaitu **graf ganda** dan **graf semu**. Sesuai dengan namanya, graf ganda adalah graf yang memiliki sisi ganda. Contoh dari sebuah graf ganda dapat dilihat pada graf G_2 pada gambar 2.1 di atas.

Graf semu adalah graf yang memiliki sisi gelang. Graf yang memiliki sisi ganda dan gelang masuk di kategori graf semu. Contoh dari graf semu dapat dilihat pada G_3 pada gambar 2.1 di atas.

Selain pengelompokan di atas, juga terdapat pengelompokan graf menurut orientasi arah yang membedakan graf secara umum menjadi dua jenis, yaitu **graf tidak berarah** dan **graf berarah**.



(a) G_4 (b) G_5

Gambar 2.2 Contoh graf berarah

Graf tidak berarah adalah graf yang sisinya tidak mempunyai orientasi arah. Urutan pasangan simpul dihubungkan oleh sebuah sisi tidak diperhatikan. Contohnya, sisi yang menghubungkan v_j dengan v_k adalah sisi yang sama dengan yang menghubungkan v_k dengan v_j. Graf-graf pada gambar 2.1 juga termasuk graf tidak berarah.

Graf berarah adalah graf yang setiap sisinya memiliki orientasi arah. Sisi yang berarah biasa disebut juga busur. Busur yang menghubungkan v_j dengan v_k tidak sama dengan busur yang menghubungkan v_k dengan v_j.

Sisi dari sebuah graf biasa dinotasikan dengan (v_j , v_k). Pada graf berarah, busur (v_j , v_k) menyatakan bahwa v_j adalah simpul asal sementara v_k adalah simpul terminal. Maka dari itu pada graf berarah, busur (v_j , v_k) memiliki arti yang berbeda dengan busur (v_k , v_j). Sementara pada graf tidak berarah, busur (v_j , v_k) dengan busur (v_k , v_j) adalah busur yang sama.

Ada beberapa terminologi yang berkaitan erat dengan graf dan akan sering digunakan.

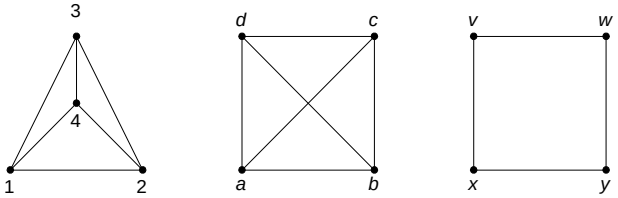
1. Bertetangga (*Adjacent*)
   Dua buah simpul pada graf tidak berarah G dikatakan bertetangga apabila keduanya dihubungkan oleh satu sisi. Pada graf berarah, untuk busur (v_j , v_k) maka dikatakan bahwa v_j bertetangga dengan v_k dan simpul v_k disebut tetangga dari v_j.
2. Bersisian *(Incident*)
   Sebuah simpul dan sebuah sisi dikatakan bersisian jika sisi tersebut menghubungkan simpul tersebut dengan simpul lain.
3. Simpul Terpencil (*Isolated Vertex*)
   Simpul terpencil adalah simpul yang tidak mempunyai sisi yang bersisian dengannya, dan akibatnya adalah simpul juga tidak akan bertetangga dengan simpul lain.
4. Graf Kosong (*Null/Empty Graph*)
   Graf kosong adalah graf yang hanya memiliki simpul, dan tidak memiliki sisi. Graf seperti ini masih terdefinisi karena definisi dari Graf adalah minimal memiliki satu simpul, namun tidak harus memiliki sisi.
5. Derajat (*Degree*)
   Derajat dari sebuah simpul pada graf tidak berarah adalah jumlah sisi yang bersisian dengan simpul tersebut. Pada graf berarah, suatu simpul memiliki dua jenis derajat yaitu derajat masuk dan derajat keluar. Sesuai dengan namanya, derajat masuk adalah jumlah busur yang keluar dari simpul sementara derajat keluar adalah jumlah busur yang masuk ke simpul tersebut.
6. Lintasan (*Path*)
   Panjang lintasan adalah jumlah sisi yang dilewati sebuah lintasan. Lintasan itu sendiri adalah barisan berselang-seling simpul dan sisi yang dilewati simpul awal ke simpul tujuan.
   Lintasan sederhana adalah lintasan yang terdiri dari simpul-simpul yang berbeda (tiap sisi dilalui hanya satu kali).
   Lintasan tertutup adalah lintasan yang berawal dan berakhir pada simpul yang sama. Sebaliknya, lintasan terbuka adalah lintasan yang simpul awal dan simpul akhirnya berbeda.
7. Siklus/Sirkuit (*Cycle / Circuit*)
   Sirkuit/siklus memiliki definisi yang sama dengan lintasan tertutup. Graf siklik adalah graf yang mengandung sebuah siklus didalamnya.
8. Terhubung (*Connected*)
   Dua buah simpul dikatakan terhubung apabila terdapat lintasan dari simpul pertama ke simpul kedua atau seebaliknya. Graf terhubung adalah graf dengan semua simpulnya saling terhubung. Jika terdapat sebuah pasangan simpul yang tidak terhubung, maka suatu graf dinyatakan bukan graf terhubung.
9. Upagraf (*Subgraph*)
   Misalkan G = (V, E) adalah sebuah graf. Maka G_1 adalah upagraf dari G jika V1 adalah elemen dari V, dan E1 adalah elemen dari E.

Selain terminologi-terminologi di atas, ada pula terminologi-terminologi lain yang umum dipakai, namun tidak dipakai dalam pembahasan makalah ini.

Ada beberapa jenis graf sederhana yang khusus, yang akan dijelaskan di bawah ini.

Graf lengkap (*complete graph*) adalah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Setiap simpul dari graf yang bersimpul *n* memiliki sisi sebanyak (*n – 1*). Graf lengkap dengan *n* buah simpul dilambangkan dengan $\boldsymbol{K}_n$.

Graf lingkaran adalah graf sedehana yang tiap simpulnya berderajat dua. Graf lingkaran bersimpul sebanyak *n* dilambangkan dengan C_n. Salah satu contoh kasus yang bisa direpresentasikan dengan graf lingkaran adalah topologi cincin dari jaringan komputer area lokal (*local area connection*).



(a) G_1 (b) G_2 (c) G_3

Gambar 2.3 Contoh Graf Isomorfik

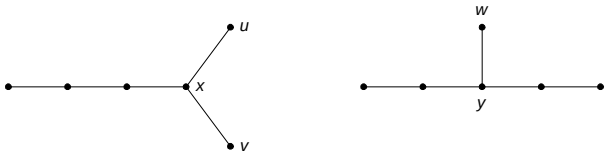
Graf isomorfik (*Isomorphic Graph*) adalah dua buah

graf yang secara geometri berbeda, namun sebenarnya kedua graf tersebut itu sama. Sebagai contoh, graf G_1 isomorfik dengan graf G_2, namun tidak isomorfik dengan graf G_3.

Dari definisi isomorfik, dapat disimpulkan bahwa graf isomorfik harus memenuhi syarat-syarat berikut:

- Mempunyai jumlah simpul yang sama
- Mempunyai jumlah sisi yang sama
- Mempunyai jumlah simpul yang sama yang berderajat tertentu

Namun ketiga syarat tersebut masih belum menjamin keisomorfikan dan masih harus dilakukan peninjauan secara visual untuk mengecek.



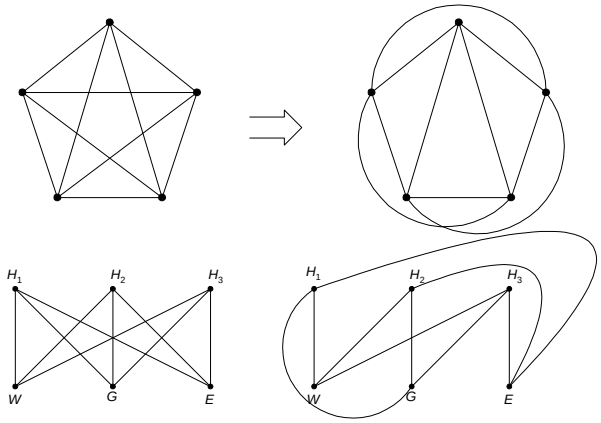
Gambar 2.4 Contoh graf yang tidak isomorfik

Pada gambar 2.4 di atas, jika ditinjau ketiga syarat di atas, maka kedua graf adalah graf isomorfik, namun ketika ditinjau secara visual, graf tersebut bukanlah graf isomorfik.

Graf planar (*Planar Graph*) adalah graf yang dapat digambarkan pada suatu bidang datar dengan sisi-sisi yang tidak saling berpotongan. Graf bidang (*Plane Graph*) adalah bentuk planar dari sebuah graf planar.

Untuk menentukan apakah suatu graf planar atau tidak, dapat dilakukan dengan teorema Kuratowski. Dalam teori graf, dikenal dua buah graf tidak planar yang khusus, yang disebut graf Kuratowski. Graf Kuratowski pertama, yaitu graf lengkap yang mempunyai lima simpul (K_5). Graf Kuratowski kedua adalah graf terhubung teratur dengan 6 buah simpul dan 9 buah sisi ($K_{3,3}$). Teorema Kuratowski berbunyi:

> Suatu graf G dinyatakan tidak planar jika dan hanya jika ia mengandung upagraf yang sama dengan K_5 atau $K_{3,3,}$ atau homeomorfik dengan salah satu dari keduanya.



Gambar 2.5 Graf K_5 (atas) dan $K_{3,3}$ (bawah)

Dua graf dikatakan homeomorfik jika salah satu dari kedua graf tersebut dapat diperoleh dari graf lain dengan cara menyisipkan dan/atau membuang secara berulang simpul-simpul yang berderajat 2.

III. Aplikasi Graf dalam Game

A. Pengonsepan Awal Game

Dengan menerapkan teori-teori graf, kita dapat membuat konsep-konsep dasar dari suatu *game*. Contohnya adalah penyusunan letak kota-kota dan rintangan pada game ber-*genre Adventure*. Kategori permainan *Adventure* termasuk kategori yang cukup populer. Anda sebagai pemain akan berusaha menjelajahi seluruh seluk-beluk 'dunia' game, dan tujuan anda menjelajahi berbeda-beda tergantung *game*-nya.

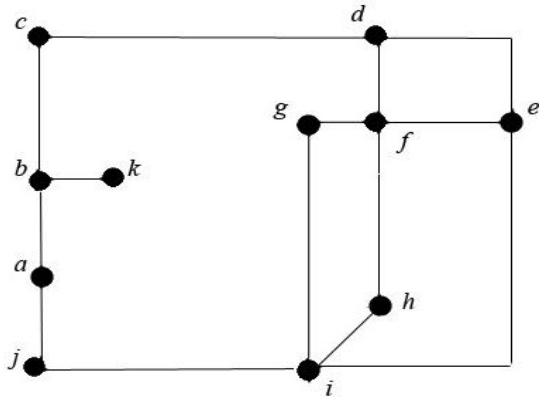
Sebagai contoh, pada setiap *game* dalam seri *Pokemon*, pemain akan bertualang ke kota-kota lain untuk mengalahkan ketua *Gym* di setiap kota, kemudian hanya setelah pemain mengalahkan semua ketua *Gym* barulah pemain boleh mengajukan tantangan ke *Pokemon League* untuk menjadi juara, yang merupakan *goal* utama dari *game* ini.

Gambar 3.1 Tampilan Peta Daerah *Kanto* pada *Pokemon FireRed*

Pada gambar 3.1, dapat dilihat bahwa sebenarnya jalan secara fisik dari kota awal (ditandai dengan tulisan START berwarna putih pada gambar) pemain ke *goal* ada. Namun untuk sampai ke *goal* tersebut, terdapat sebuah prasyarat yaitu pemain harus mengumpulkan lencana-lencana yang hanya bisa didapat dengan mengalahkan ketua-ketua *Gym*. Prasyarat-prasyarat tidak hanya ada pada *goal*, saat menuju suatu kota, pemain juga harus menyelesaikan prasyarat-prasyarat yang ada. Berikut merupakan graf simplifikasi peta tersebut.

Pada permainan-permainan petualangan, biasa terdapat banyak macam rintangan di jalan menuju daerah yang berikutnya harus dicapai, begitu pula dengan game *Pokemon* ini. Dalam *game* seperti ini, urutan penempatan kota-kota maupun rintangan yang akan dihadapi saat menuju kota sangatlah penting. Sebagai pemain, misalkan anda ingin pergi ke kota *j*, dapat dilihat pada gambar 3.2 di bawah bahwa jalan menuju *j* hanya ada dari kota *a* dan kota *i*. Namun kedua jalan menuju *j* adalah jalan air, anda harus melewati laut untuk mencapainya. Pendekatan

masalah ini sebagai pemain adalah untuk mencari *Pokemon* yang bisa berenang di air untuk menempuh jalan tersebut.



Gambar 3.2 Graf dari gambar 3.1

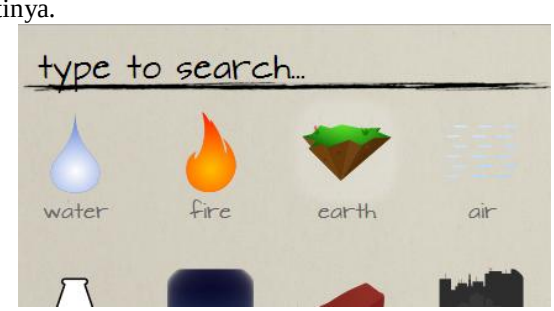
Pendekatan sebagai seorang *developer* yang membuat game ini berbeda dari pendekatan pemain. Sebagai pembuat game, tentunya anda tidak mungkin menyusun skenario sedemikian rupa sehingga untuk mencapai kota *j*, anda harus mendapatkan sesuatu yang hanya bisa didapatkan di kota *j* atau di kota yang hanya bisa dicapai setelah anda mencapai *j*. Kejadian ini dapat diilustrasikan dengan graf siklik.

Menggunakan konsep dasar dari *topological sort*, dengan adanya suatu siklus dalam graf, maka kita tidak bisa menentukan simpul (atau dalam kasus ini, tindakan yang harus dilakukan) mana yang harus dilakukan. Hal ini karena untuk memilih suatu simpul untuk dilakukan terlebih dahulu, simpul yang dipilih pertama haruslah simpul tanpa prasyarat (*predecessor*), namun dalam suatu siklus, tiap simpul memiliki prasyarat dan juga menjadi prasyarat.

Namun ada pula kejadian apabila dalam game, anda sebagai *developer* tidak membuat suatu solusi untuk mencapai kota *j*. Kejadian yang kedua ini dapat diilustrasikan dengan graf dengan simpul terpencil.

Jika terdapat simpul terpencil, berarti dalam *game* tersebut ada sesuatu, baik itu skenario, area, maupun sesuatu yang lain, yang sebetulnya ada pada *game* itu, namun tidak bisa diakses dengan cara apapun dalam *game* tersebut. Jika tidak dipakai, data tersebut hanya akan berperan sebagai data sampah yang akan memperbesar ukuran program *game* anda sebagai data yang tidak bisa diakses. Oleh karena itu, adanya simpul terpencil dalam suatu graf konsep *game* adalah sesuatu yang seharusnya tidak terjadi.

Contoh *game* lain yang konsepnya menggunakan aspek-aspek yang sama dari graf adalah *Little Alchemy*[1]. Pada game ini, anda harus menggabungkan dua elemen untuk menghasilkan elemen-elemen lain. Jumlah variasi elemen yang ada di game ini tidaklah sedikit. Anda memulai game ini dengan hanya 4 elemen dasar, yaitu *Fire, Water, Wind, Earth*. Jika anda mengombinasikan elemen-elemen yang tepat, anda bahkan bisa membuat elemen-elemen unik seperti *Werewolf, Vampire, Frankenstein*. Tidak lupa juga disertakan elemen-elemen dasar seperti *Brick* (batu bata yang dapat membentuk *House*, yang kemudian dapat membentuk *Village* dan *City* nantinya.



Gambar 3.3 Elemen-elemen awal dari Little Alchemy

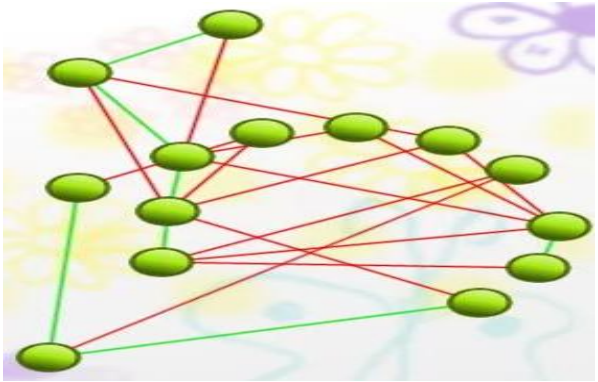
Pembentuk sebuah elemen tidak bersifat unik, dalam arti suatu elemen bisa dibentuk oleh elemen A dan B, namun juga bisa dibentuk oleh elemen C dan D, ataupun A dengan C.

Misalkan pada suatu pembaharuan versi dari game *Little Alchemy* tersebut, *developer* ingin menambahkan suatu elemen, contohnya *Biscuit*, maka dengan memodelkan semua elemen dalam sebuah graf, pertama *Biscuit* harus ditambahkan terlebih dahulu sebagai sebuah simpul, kemudian baru kita membuat sebuah busur yang mengarahkan elemen-elemen pembentuk (*predecessor*) ke *Biscuit*, contohnya *Dough* dan *Fire*.

Sama seperti kedua kejadian pada pembahasan sebelumnya, jika kita sebelumnya tidak menghubungkan elemen-elemen pembentuk ke elemen hasil, maka elemen hasil itu akan bertindak sebagai simpul terpencil dan tidak akan bisa dibentuk. Selain itu, kesalahan bisa juga terjadi jika pembentuk suatu elemen hanya satu pasangan elemen, dan sedikitnya salah satu dari pembentuk itu adalah elemen itu sendiri. (A ← A + B).

B. Sebagai Bagian Permainan

UntangleMe[1] adalah permainan *freeware* yang bisa dimainkan di konsol *smartphone*. *Game* ini dirilis pada *PlayStore* dan terakhir kali diperbaharui pada Oktober tahun 2012.



Gambar 3.4 Contoh tampilan *UntangleMe* (dipotong)

Pada game ini, pemain berusaha melepas kawat-kawat kusut yang menghubungkan beberapa titik dengan menggeser titik-titik hijau. Dapat dilihat pada gambar 3.3

di atas, kawat yang saling bersilangan ditandai dengan warna merah sementara yang tidak bersilangan dengan kawat lain ditandai dengan warna hijau.

Game ini menggunakan graf, seperti bisa dilihat dari gambar 3.3. Kawat pada *game* ini dapat ditinjau juga sebagai sisi, sementara titik-titik ditinjau sebagai simpul. Dengan meninjau tujuan *game* ini, dapat disimpulkan bahwa persoalan graf yang diberikan dalam game ini tidak boleh berupa graf yang tidak planar. Dengan kata lain, permainan ini bertujuan untuk mencari graf bidang dari graf planar tersebut. Agar suatu graf dapat dinyatakan planar, maka menurut teorema Kuratowski, graf tersebut tidak boleh mengandung upagraf $K_{3,3}$ atau K_5. Maka untuk membuat graf sebagai persoalan *game* ini, tidak boleh menggunakan graf $K_{3,3}$ atau K_5

IV. KESIMPULAN

Penggunaan graf dalam bidang informatika maupun diluar bidang informatika sangatlah banyak. Penerapan yang dicantumkan pada makalah ini hanya sebagian kecil dari penerapan yang mungkin pada dunia informatika. Graf dapat dipakai sebagai 'alat' untuk penyusunan ide konsep suatu *game*. Selain itu, graf juga bisa operasi-operasi yang umum diterapkan pada graf juga bisa dibuat menjadi permainan yang menarik, contohnya pada permainan *Untangle Me*. Penerapan graf lain juga mencakup perhitungan jalur terpendek untuk memperpendek waktu permainan.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada orang-tua yang telah melahirkan penulis. Tidak lupa penulis mengucapkan terima kasih sebesar-besarnya kepada Ir. Rinaldi Munir, M.T., selaku dosen pengajar mata kuliah Matematika Diskrit, Program Studi Teknik Informatika, Institut Teknologi Bandung, untuk ilmu-ilmu yang telah didapatkan dari beliau selama setengah tahun terakhir, dan untuk kesempatan yang telah diberikan untuk membuat makalah ini, dan juga orang-orang yang telah memberikan bantuan dalam berbagai bentuk demi terselesaikannya makalah ini.

REFERENSI

[1] http://littlealchemy.com/
Little Alchemy, game *browser*. Diakses pada 16 Desember 2013, pukul 17.34

[2] https://play.google.com/store/apps/details?id=softkos.untangleme&hl=en
Untangle Me Free pada PlayStore . Diakses pada 16 Desember 2013, pukul 17.07

[3] R. Munir, Diktat Kuliah IF2120 Matematika Diskrit edisi keempat. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2006.

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, 16 Desember 2013

ttd

Felicia Christie / 13512039