

Menelusuri Labirin *Tartarus* pada Permainan *Persona 3 Portable* Menggunakan Algoritma *Depth First Search*

Mohammad Dicky Andika Putra - 13515044¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13515044@std.stei.itb.ac.id

Abstract— *Persona* adalah permainan yang dikembangkan oleh Atlus. Permainan ini berfokus pada *Role-playing* nya yang dilakukan di dalam *dungeon* (suatu medan tarung pada game RPG) yang biasanya berupa labirin. dalam *Persona 3*, *Tartarus* adalah suatu menara labirin yang harus ditelusuri oleh pemain. Makalah ini menjelaskan penelusuran labirin *tartarus* ini menggunakan pendekatan algoritma *Depth First Search* (DFS).

Keywords— *Persona 3*, *tartarus*, *depth first search*, labirin

I. PENDAHULUAN

Persona 3 adalah suatu *Role-Playing based game* yang dibuat dan dikembangkan oleh Atlus. *Persona 3* bercerita tentang murid pindahan yang terpaksa harus menyelesaikan kasus tentang *Dark Hour* (atau jam kegelapan), yang terjadi pada pergantian hari pada jam 00.00, dan hanya beberapa orang dengan suatu potensi yang dapat mengetahuinya. Namun murid pindahan ini harus tetap menjaga kehidupan sosial dan sekolahnya selayaknya murid yang normal.

Persona 3 ini memiliki 2 versi lanjutan, yakni *Persona 3 FES* (P3F) dan *Persona 3 Portable* (P3P).

P3F adalah versi lanjutan dari *Persona 3* yang memiliki tambahan cerita lanjutan dari cerita utama *Persona 3*, serta beberapa perbedaan dibandingkan dengan *Persona 3* pada versi cerita tambahannya.

Persona 3 Portable(P3P), adalah varsi lain dari *Persona 3* yang dapat dimainkan di *platform PlayStation Portable*, sedangkan *Persona 3* dan *Persona 3 FES* menggunakan *platform PlayStation 2*. Pada P3P, antarmuka pengguna ditampilkan dalam bentuk *visual novel*, berbeda dengan varsi *Persona 3* atau P3F yang ditampilkan dalam bentuk 3D. Selain itu, dalam *Persona 3 Portable*, pemain dapat menentukan akan beemain sebagai pria atau wanita, tanpa mengubah inti dari keseluruhan cerita *Persona 3*.



Gambar 1.1 Cover *Persona 3 Portable*
(<http://www.mobygames.com>)

Baik *Persona 3*, P3F, maupun P3P, ketiganya bertema pada pertarungan bergilir (*role playing*). Pertarungan ini dilakukan pada sebuah *dungeon* yang disebut dengan *Tartarus*.

Tartarus adalah inti dari seluruh cerita dalam seri *persona 3*. *Tartarus* merupakan menara labirin yang hanya muncul pada jam kegelapan (*Dark hour*).

di dalam *tartarus*, terdapat total 264 lantai yang harus ditelusuri semuanya tidak termasuk dengan *dungeon* bonus (disebut dengan *monad* yang memiliki total 10 lantai dan memiliki labirin yang cukup lebih luas dan lebih rumit dibandingkan dengan *tartarus*) dan terbagi menjadi 10 sub-*dungeon* dengan pembagian yang tidak merata setiap sub-*dungeon*-nya.

Pada menara ini, terdapat berbagai hal: item, musuh, *access point* (untuk kembali keluar *tartarus*), dan tangga untuk menuju ke lantai berikutnya. pada saat-saat tertentu terdapat misi untuk menyelamatkan orang yang tersesat di dalam *Tartarus*.

Setiap labirin pada setiap lantai dalam *Tartarus* selalu berubah-ubah susunannya secara acak. Sehingga tidak dapat diketahui dengan pasti jalan keluar untuk setiap lantai. Walaupun untuk beberapa lantai (lantai untuk *mini-boss*) selalu memiliki susunan labirin yang sama setiap waktu.



Gambar 1.2 Tartarus pada Persona 3 (megamitensei.wikia.com)

Pada Persona 3 FES, terdapat *dungeon* baru pada varsi cerita lanjutannya yang disebut *Abyss of Time*. Mirip seperti Tartarus, namun *Abyss of Time* adalah jurang yang mendalam, juga tersusun atas labirin yang susunannya tak menentu. Sedangkan Tartarus adalah menara yang menjulang ke atas. Selain itu, *Abyss of Time* memiliki 8 sub-*dungeon* yang berbeda dan harus dieksplorasi semuanya.

Persona 4, seri game Persona dari *Shin Megami Tensei* setelah Persona 3 yang memiliki *gameplay* yang sama kecuali dari segi cerita utamanya, juga memiliki sistem permainan labirin. yang membedakan antara keduanya adalah Persona 4 memiliki beberapa *dungeon* utama yang berbeda-beda setiap waktu. Namun, keduanya memiliki hal yang serupa pada sisi labirin yang susunannya selalu berubah-ubah.

Pada makalah ini, akan dijelaskan penerapan algoritma DFS untuk menelusuri labirin Tartarus pada Persona 3 Portable, yang juga dapat diimplementasikan untuk seluruh seri Persona 3 dan Persona 4.

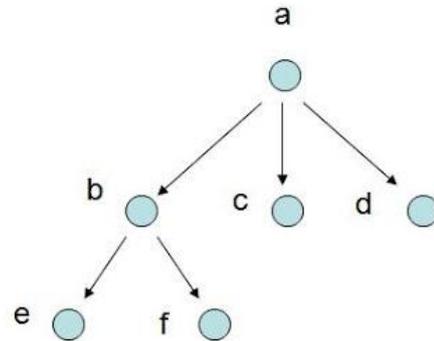
II. DASAR TEORI

Algoritma Depth First Search (DFS) adalah salah satu algoritma penelusuran graf. Algoritma ini juga sering disebut dengan pencarian mendalam. DFS mencari ke salah satu simpul cabang dahulu secara mendalam sampai bisa diputuskan apakah cabang tersebut memiliki solusi atau tidak.

Urutan penelusuran graf dengan algoritma DFS dapat diuraikan sebagai berikut:

- Kunjungi simpul awal (v).
- Kunjungi simpul yang bertetangga dengan v yang belum dikunjungi (v').
- Ulangi langkah (b) sampai dengan simpul x yang mana simpul x tidak memiliki simpul tetangga yang belum dikunjungi.

- Lakukan *backtracking* ke simpul sebelum simpul x dan kunjungi simpul lain yang bertetangga yang belum dikunjungi.
- Ulangi langkah (c) dan (d) sampai tidak ada satupun simpul yang belum dikunjungi



Gambar 2.1

(<http://dedy-froze.blogspot.co.id/>)

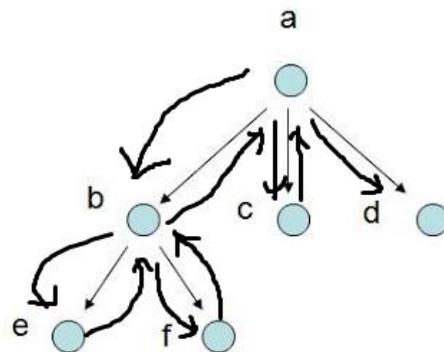
Sebagai contoh, apabila mencoba memulai penelusuran graf di atas menggunakan algoritma *Depth First Search*, mulai dari simpul (a), dan mendahulukan abjad yang terkecil terlebih dahulu, maka kita akan menuju simpul (b) untuk memulainya.

Dari (b), berbeda dengan algoritma *Breadth First Search*, kita tidak menelusuri simpul (c) dan (d) dahulu. Melainkan melanjutkan perjalanan dari simpul (b) ke simpul selanjutnya yang belum dikunjungi dengan abjad yang paling dahulu, yaitu simpul (e).

Karena dari simpul (e) tidak ada lagi simpul yang belum dikunjungi, lakukan *backtracking* ke simpul sebelum simpul (e), yakni simpul (b). lalu dari simpul (b) cek lagi apakah ada simpul lain yang belum dikunjungi.

Langkah seperti ini terus dilakukan sampai seluruh simpul di dalam graf telah dikunjungi semua.

Solusi penelusuran graf dengan DFS: a-b-e-f-c-d.



Gambar 2.2 Solusi penelusuran dengan DFS

Sebagai perbandingan, apabila melakukan penelusuran dengan *Breadth First Search* (atau disebut juga sebagai “pencarian melebar”) dari simpul (a), akan lakukan penelusuran dengan mencari semua simpul yang bertetangga dengan simpul (a) yang belum pernah dikunjungi. Didapat simpul (b), (c), dan (d).

Kemudian, dari ketiga simpul tersebut, ulangi langkah di atas untuk setiap simpul. Apabila mendahulukan simpul dengan abjad terkecil, maka dimulai untuk melakukan BFS pada simpul (b) dan didapatkan simpul (e) dan (f).

Lalu, giliran untuk melakukan pencarian BFS dari simpul (c). namun, simpul (c) tidak memiliki simpul tetangga yang belum pernah dikunjungi, sehingga bisa dianggap pencarian dengan BFS untuk simpul (c) telah berakhir. Begitu pula dengan simpul (d).

Lalu untuk setiap simpul anak dari (b) akan dilakukan BFS kembali, namun kedua simpul ini tidak memiliki simpul yang belum dikunjungi, seperti kasus yang terjadi pada simpul (c) dan (d).

Karena semua simpul telah dikunjungi, maka pemcarian BFS untuk graf ini telah berakhir.

Solusi penelusuran graf dengan BFS: a-b-c-d-e-f.

Kelebihan algoritma *Depth First Search* dibandingkan dengan *Breadth First Search* adalah kebutuhan memorinya yang lebih sedikit karena simpul yang telah diketahui tidak memiliki solusi akan dihapus dari memori.

Namun, pencarian dengan DFS lebih banyak memakan waktu dibandingkan dengan BFS apabila graf memiliki kedalaman yang cukup besar. Selain itu, algoritma DFS tidak menjamin akan menemukan solusi dalam graf apabila graf memiliki kedalaman tak berhingga, sedangkan solusi yang dicari berada pada simpul lain yang ternyata tidak dipilih. Sedangkan pada BFS, solusi pasti ketemu selama solusi tersebut memang ada sebab tiap simpul ditelusuri perlahan-lahan. Jadi, tidak ada simpul yang tidak terpilih.

III. IMPLEMENTASI ALGORITMA DFS PADA PENELUSURAN TARTARUS

Pada Persona (dalam hal ini, Persona 3), labirin Tartarus tidak memberikan peta yang jelas pada awalnya. Denah pada permainan kemudian memberikan informasi tentang labirin sejalan ketika player bergerak menelusuri labirin. Jadi, bisa dikatakan bahwa player sedang melakukan *uninformed search* dalam labirin Tartarus.

Dalam prosesnya, pemain akan sering kali dihadang oleh musuh ketika sedang berjalan. Namun, kali ini tidak akan dibahas mengenai pertarungan dengan musuh karena tidak akan menimbulkan efek pada permainan.



Gambar 3.1 tampilan awal saat memasuki labirin

Sebagai contoh, ketika baru memasuki suatu lantai pada Tartarus (pada gambar di atas, lantai 26), peta yang ditampilkan di sebelah kiri menunjukkan informasi yang dapat diketahui. Dapat dilihat bahwa pemain dapat berbelok ke kiri maupun ke kanan. Dan gambar 3.2 di bawah adalah tampilan akhir dari labirin yang telah ditelusuri.



Gambar 2.2 peta labirin yang telah utuh

Untuk menelusuri labirin dari Gambar 3.1 ke Gambar 3.2, pemain akan perlahan-lahan menelusuri dengan algoritma DFS. Awal mula dari labirin tersebut direpresentasikan ke dalam gambar berikut.



Gambar 3.3 representasi denah awal

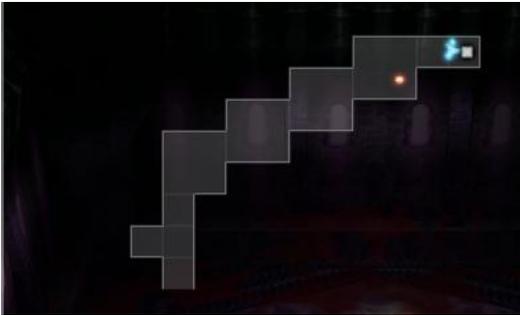
Tanda bintang dalam gambar tersebut menunjukkan posisi awal pemain. Saat ini pemain dapat memutuskan untuk berbelok ke kanan atau ke kiri. asumsikan bahwa pemain mendahulukan arah kiri terlebih dahulu.

Dalam keadaan denah labirin belum penuh, pemain dapat melihat satu langkah (petak) kedepan dari posisinya saat ini. seperti pada gambar 3.3 bahwa pemain dapat melihat bahwa pada arah ke kanan terdapat daerah, dan

pemain dapat mengetahui ia bisa belok ke kiri dan ke kanan dari daerah tersebut, sedangkan dari petak pemain sendiri, ia tidak bisa kemana-mana karena adanya tembok.

Apabila pemain berbelok kearah kiri, menurut denah yang telah lengkap, tidak ada lagi persimpangan setelahnya sampai di sebuah titik berwarna putih di ujung tersebut.

Labirin yang telah ditelusuri sampai saat ini dapat dilihat sebagai berikut,



Gambar 3.4 langkah awal ketika berbelok ke arah kiri

Karena tidak ada lagi jalan lain, pemain kembali ke titik awal, lalu mengambil jalan berbelok ke arah kanan, sampai bertemu dengan persimpangan selanjutnya.

Ketika pemain mengambil jalan yang lain tersebut dan meneruskan berjalan sampai bertemu ke persimpangan selanjutnya, denah yang ditunjukkan pada layar permainan sampai persimpangan kedua adalah sebagai berikut,



Gambar 3.5 Persimpangan kedua

Pada gambar di atas, setelah pemain pergi ke arah kanan dari kedudukan awal (gambar 3.1), pemain menemukan persimpangan selanjutnya dan memutuskan ke arah kiri. pemain menemukan akses tangga untuk ke lantai berikutnya (lantai 27). Namun, tujuan pemain saat ini adalah mengeksplorasi denah labirin di lantai tersebut seutuhnya (seperti gambar 3.2 sebelumnya). Jadi, pemain akan kembali ke persimpangan kedua dan mencoba jalan lain yang belum dicari tahu.

Jadi, pemain akan bergerak ke bawah, menurut denah

di atas. dan pemain akan menemukan persimpangan ke tiga ketika berjalan terus. Yang dapat dilihat pada gambar ini,



Gambar 3.6 Persimpangan ke 3

Pada persimpangan ketiga ini, pemain akan langsung mencoba jalur ke kiri dan berjalan terus. dan informasi yang didapatkan oleh pemain setelah memilih jalur kiri adalah sebuah item (tanda bintang di gambar 3.2) dan ruang kosong lainnya.



Gambar 3.7 Pemain menemukan item

Lalu, karena tidak ada jalan lain di situ, pemain kembali ke persimpangan ke tiga karena masih ada jalur yang belum dilewati.

Ketika pemain melewati jalur tersebut, pemain kembali menemukan persimpangan yang tidak terlalu banyak. Karena seperti penjelasan sebelumnya, pemain dapat mengetahui informasi mengenai satu petak disekitar petak yang saat ini ditempati oleh pemain.



Gambar 3.8 persimpangan terakhir

Dari gambar tersebut, pemain telah mendapatkan informasi bahwa arah kiri dari persimpangan adalah jalan buntu, arah kanan dari persimpangan adalah sebuah ruang kosong, dan tinggal arah depan saja yang masih bisa dicoba lagi.

Setelah maju lurus, pemain akan mendapatkan jalan buntu. Akhirnya, denah labirin pada lantai saat ini telah lengkap sesuai dengan yang digambarkan pada gambar 3.2 di atas.

Akhirnya, penelusuran untuk lantai 26 saat ini telah berakhir. Dengan informasi yang bisa didapat dari denah yang lengkap (informasi item, *access point*, tangga ke lantai atas, dan musuh disekitarnya).

Pemain dapat melakukan apa yang ia inginkan dengan informasi lengkap mengenai denah tersebut. Namun, pemain tidak dapat mengetahui informasi tentang musuh yang ada diseluruh lantai, melainkan hanya mengetahui musuh-musuh yang ada disekitarnya saja. Keberadaan musuh dapat dilihat dengan titik berwarna merah pada denah.

Langkah dengan algoritma DFS seperti ini dapat dipakai untuk lantai-lantai berikutnya di dalam Tartarus. Karena susunan tartarus yang berubah-ubah setiap kali pemain masuk, mengakibatkan tidak ada solusi yang tetap untuk langsung menuju arah yang diinginkan pada setiap lantai.

Misalnya, penulis mencoba untuk memasuki lantai 26 untuk kedua kalinya pada hari yang sama (hari dalam sistem permainan, bukan waktu asli). Penulis kembali mengimplementasikan algoritma *Depth First Search* untuk mengeksplorasi labirin dan denah dibawah adalah denah lengkap dari lantai 26 yang didapat oleh penulis pada kunjungan kedua.



Gambar 3.9 Kunjungan kedua lantai 26

Gambar denah di atas membuktikan bahwa susunan tartarus selalu berubah bahkan hanya dengan ditinggal sesaat.

IV. FUNGSI KOMANDO “SPLIT UP”

Pada Persona 3, atau Persona 3 Portable yang penulis mainkan, terdapat komando bernama “*Split up*” ketika sedang bermain dalam mode penjelajahan di dalam Tartarus. Fungsi ini memberi perintah kepada anggota team (*party*) untuk menyebar menelusuri tartarus dengan arah masing-masing, menemukan item di dalamnya, serta melawan musuh apabila diperintahkan oleh karakter utama.

Menurut pengamatan penulis, setiap anggota *party* di Persona 3, menelusuri labirin Tartarus menggunakan algoritma DFS. Dengan cara kerja dalam pseudocode sebagai berikut:

```

prosedure Split_Up(){
  while (command_regroup = false){
    repeat{
      //algoritma DFS sistem
      // sistem memilih secara random terhadap
      // prioritas anggota player (kiri atau
      // kanan)
      ...
      if (command_regroup = true)
        break;
    until((something_found = true)
           OR (battle = true));
    something_found = false;
    battle = false;
  }
}

```

Ketika pemain mengaktifkan komando untuk menyebar (*split up*), seluruh anggota *party* akan melakukan pergerakan otomatis. Sistem akan menentukan apakah anggota *party* mengutamakan belok kiri atau kanan, lalu ia akan melakukan penelusuran dengan DFS sampai ia menemukan sesuatu (item atau melakukan battle dengan musuh). Ketika ini terjadi, program DFS akan terhenti, dan akan diaktifkan lagi setelah item terambil atau musuh dikalahkan (jika anggota yagn sedang bertarung tersebut berhasil menang).



Gambar 4.1 Contoh tampilan denah ketika melakukan *split up*

Seluruh proses tersebut akan terus berjalan selamanya sampai *player* utama mengeluarkan komando *regroup*. *Ragroup* adalah fungsi komando untuk menyuruh seluruh anggota *party* untuk kembali berkumpul pada pemain utama.

Dengan fungsi ini, pemain tidak harus mengeksplorasi labirin sendiri secara total. Melainkan menyerahkan sebagian tugas eksplorasi kepada anggota *party*. Seluruh anggota *party* juga dapat mengambil item yang mereka dapatkan selama menelusuri labirin, dan akan diberikan pada pemain utama ketika anggota tersebut kembali pada karakter utama saat ia memanggil komando *regroup*. Namun, apabila pemain utama, atau anggota *party* memutuskan untuk melakukan *battle* dengan musuh yang ditemui ketika dalam keadaan *split up*, anggota, atau pemain, juga hanya dapat melakukan *battle* tersebut sendiri. Kecuali apabila terdapat anggota lain berada disekitarnya.

V. KESIMPULAN

Algoritma penelusuran *Deth First Search* dapat digunakan untuk menelusuri labirin yang tidak diketahui struktur labirinnya. Sebagai contoh, adalah labirin menara tartarus dalam permainan Persona 3 (atau labirin dalam seri Persona lainnya) yang struktur nya selalu berubah-ubah setiap kali pemain memasukinya. Sehingga tidak ada cara lain selain harus menggunakan algoritma pencarian dengan *uninformed search*, dan salah satunya adalah dengan metode DFS. Dalam penelusuran Tartarus dengan DFS, apapun yang dicari (tangga, *access point*, atau item) pasti akan ketemu karena labirin Tartarus merupakan labirin dengan jalur yang relatif kecil apabila dibandingkan dengan medan tempur pada permainan lainnya.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Allah Subhanahu Wa Ta'ala karena dengan izin Nya penulis dapat menyelesaikan makalah ini pada tepat waktu. Penulis juga berterima kasih kepada Ibu Nur Ulfa

Maulidevi selaku pembimbing penulis pada mata kuliah Strategi Algoritma selama satu semester ini, juga kepada Bapak Rinaldi Munir dan Ibu Masayu Leylia Khodra sebagai dosen mata kuliah tersebut. Tak lupa, kepada teman-teman dari informatika yang selama ini turut membantu penulis semasa perkuliahan. Serta beberapa pihak lain yang belum disebutkan di atas yang turut membantu atas pembuatan makalah dan perjuangan penulis selama satu semester di mata kuliah ini.

REFERENSI

- [1] Munir, Rinaldi. 2009. *Diktat Kuliah IF2211 Strategi Algoritma*. Bandung : Penerbit Informatika.
- [2] http://megamitensei.wikia.com/wiki/Persona_3 diakses pada 18 Mei 2017.
- [3] <http://megamitensei.wikia.com/wiki/Tartarus> diakses pada 18 Mei 2017.
- [4] http://megamitensei.wikia.com/wiki/Abyss_of_Time diakses pada 18 Mei 2017.
- [5] www.mobygames.com/game/shin-megami-tensei-persona-3-portable/cover-art/gameCoverId.216587/ diakses pada 17 Mei 2017.
- [6] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/BFS%20dan%20DFS%20\(2015\).pptx](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/BFS%20dan%20DFS%20(2015).pptx) diakses pada 17 Mei 2017.
- [7] <http://dedy-froze.blogspot.co.id/> diakses pada 17 Mei 2017.

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 Mei 2016

Mohammad Dicky Andika Putra
13515044