

Aplikasi Algoritma BFS pada Hantu dalam Permainan Pac-Man

Emeraldy Widiyadi - 13508067
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If18067@students.if.itb.ac.id

Abstrak— Pac-Man adalah sebuah permainan yang memiliki satu tantangan utama, yaitu menyelesaikan level dengan menggerakkan Pac-Man untuk memakan seluruh *pac-dots* yang tersebar di labirin sambil menghindari sergapan sejumlah hantu atau monster yang akan menghabisi Pac-Man begitu ia tertangkap. Pergerakan hantu dalam beberapa kondisi permainan ini dapat diatur oleh algoritma pencarian jalur terdekat, yaitu BFS (Breadth-First Search). Algoritma BFS adalah salah satu algoritma pencarian jalur yang dapat menganalisa lintasan dan mendeteksi jalur-jalur yang dapat dilalui. Penggunaan algoritma BFS disesuaikan dengan status hantu pada saat itu, yaitu “pengejaran”, dan “ketakutan”.

Kata Kunci— BFS, hantu, jalur, Pac-Man.

I. PENDAHULUAN

Pac-Man adalah sebuah permainan video *arcade* yang dikembangkan oleh Namco dan dipublikasikan oleh Midway. Game ini pertama kali dirilis di Jepang pada 22 Mei 1980. Pada awalnya, Pac-Man hanya dirilis untuk *arcade* saja. Namun dalam perkembangannya, Pac-Man yang masih populer hingga kini dirilis pula dalam *platform* lainnya seperti Game Boy, SNES, bahkan Playstation dan konsol game modern. Perancang permainan ini adalah Toru Iwatani, yang merupakan karyawan Namco.

Pemain harus mengontrol tokoh berwarna kuning bernama Pac-Man dan membawanya mengelilingi labirin sambil “memakan” *pac-dots* (titik-titik). Pada saat permainan dimulai, terdapat empat *hantu* yang akan keluar dari persembunyiannya, lalu berkeliling di labirin tersebut untuk menangkap Pac-Man. Sang pemain dapat menyelesaikan satu *level* (tingkat) jika berhasil memakan seluruh *pac-dots*, baik yang berjenis standar maupun spesial. Selain itu, di labirin pun terdapat item yang sesekali muncul untuk menambah skor jika dimakan.

Pada pergerakan otomatis *hantu* yang ingin menangkap Pac-Man, terdapat pengaplikasian algoritma untuk mencari jalur baik menuju daerah sekitar Pac-Man (pada status memburu) maupun menuju tempat persembunyiannya (pada status “ketakutan”). Salah satu dari algoritma yang dapat diaplikasikan tersebut adalah BFS (Breadth-First Search). Dengan menerapkan algoritma ini, game Pac-Man diharapkan akan lebih

menantang, karena *hantu* akan semakin “cerdas” dalam sejumlah kondisi permainan.



Gambar 1 Layar depan permainan Pac-Man

II. PERMAINAN PAC-MAN

Permainan Pac-Man memiliki suatu pola permainan yang repetitif pada tiap level. Seiring dengan meningkatnya level, permainan akan menjadi lebih sulit dengan berkembangnya labirin dan kemampuan hantu dalam mengejar Pac-Man (kecepatan dan kepekaan bertambah). Pengembangan game Pac-Man memunculkan banyak *side gameplay* baru yang diimplementasikan, namun pada umumnya *gameplay* dan tokoh utama Pac-Man tetap sama.

II.1 Tokoh

Pac-Man

Pac-Man adalah tokoh protagonis pada game. Tokoh yang dapat dikendalikan oleh seorang pemain (human) ini hanya memiliki satu tujuan utama, yaitu menghabiskan seluruh *pac-dots* yang ada pada labirin sambil menghindari kejaran *hantu*.

Hantu

Hantu adalah tokoh antagonis pada game. Tokoh yang pergerakannya dikendalikan oleh komputer (AI) ini bertugas untuk menangkap Pac-Man. Pada awal perilsan

Pac-Man, terdapat empat *hantu* yang memiliki warna, nama, dan sifat berbeda. Mereka adalah :



Gambar 2 Hantu-hantu dalam Pac-Man

- Blinky / Akabei (赤ベイ)

Sifat : Shadow / Oikake (追いかけ)

Karakter hantu merah yang bersifat seperti “bayangan” dikenal sebagai Blinky. Di Jepang, karakter yang diwakili oleh kata “Oikake” ini memiliki tugas "mengejar dan menangkap". Blinky tampaknya selalu menjadi hantu pertama yang melacak Pac-Man pada labirin. Selain itu, Blinky-lah yang paling agresif dari empat lainnya dan pasti akan mengejar Pac-Man secara efektif.

- Pinky / Pinky (ピンキー)

Sifat : Speedy / Machibuse (待ち伏せ)

Dijuluki Pinky, karakter hantu pink digambarkan sebagai hantu yang cepat. Di Jepang, ia dicirikan sebagai “Machibuse”, yang berarti "penyergapan". Pinky dapat tiba-tiba berada di depan Pac-Man dan menyergap Pac-Man tanpa dapat diduga sebelumnya.

- Inky / Aosuke (青助)

Sifat : Bashful / Kimagure (気まぐれ)

Hantu cyan (biru-hijau) pada game dijuluki Inky, dan karakter ini dianggap sebagai hantu yang malu-malu. Di Jepang, ia digambarkan sebagai “Kimagure”, yang berarti "temperamen berubah-ubah, murung, atau tidak merata". Kadang-kadang ia mengejar Pac-Man secara agresif seperti Blinky; kali lain ia mendahului Pac-Man seperti Pinky; ia bahkan dapat mengembara tak tentu arah seperti Clyde. Inky mungkin hantu paling berbahaya dari semua karena perilaku tak menentukannya.

- Clyde / Guzuta (愚図た)

Sifat : Pokey / Otoboke (お惚け)

Hantu jingga berjulukan Clyde dianggap sebagai hantu yang bermuslihat. Di Jepang, karakter ini digambarkan sebagai “Otoboke”, yang berarti "berpura-pura bodoh". Pada permainan, Clyde bergerak dengan kecepatan yang sama seperti Inky dan Pinky. Clyde adalah hantu terakhir yang meninggalkan tempat persembunyian dan cenderung untuk memisahkan diri dari hantu lain karena

gerakannya yang seperti berpatroli di area tertentu pada labirin.

II.2 Area Permainan

Contoh area permainan Pac-Man adalah sebagai berikut:



Gambar 3 Area permainan Pac-Man

Area permainan terdiri dari labirin yang dipenuhi oleh *pac-dots* dan beberapa *power pellet* atau *pac-dots* spesial. Terdapat pula tempat persembunyian hantu sebagai titik awal hantu saat permainan dimulai dan juga sebagai tempat hantu memulihkan diri apabila termakan oleh Pac-Man pada kondisi ketakutan.

II.1 Pola Permainan

- Permainan dimulai, labirin muncul, Pac-Man dapat digerakkan oleh pemain, dan hantu mulai keluar dari tempat persembunyiannya.
- Pac-Man memakan *pac-dots* sementara hantu menyebar dan mencari Pac-Man.
- Ketika Pac-Man memakan *power pellet* (*pac-dot* spesial), kemampuan hantu untuk memakan Pac-Man hilang, kecepatannya menurun, dan warna tubuhnya berubah menjadi biru tua. Keadaan ini biasa disebut *frightened* (ketakutan). Pada keadaan ini, hantu justru menghindari Pac-Man yang dapat memakannya dan berusaha untuk masuk ke tempat persembunyian. Di tempat inilah, hantu akan selamat dari kejaran Pac-Man. Keadaan ini akan berlangsung beberapa saat sebelum pola permainan kembali seperti semula (hantu mengejar Pac-Man). Jika dalam keadaan tadi hantu dimakan oleh Pac-Man, maka bagian yang tersisa dari hantu secepat kilat akan menuju tempat persembunyiannya dan berubah menjadi hantu yang normal.

- Jika Pac-Man tertangkap hantu, maka nyawa Pac-Man akan berkurang. Setelah itu, Pac-Man dan hantu akan muncul kembali pada tempat seperti sediakala, namun dengan kondisi jumlah tersisa *pac-dots* pada labirin tepat sebelum Pac-Man tertangkap.
- Pada waktu tertentu, akan muncul item yang berguna untuk menambah skor. Biasanya, setelah mencapai skor tertentu, nyawa Pac-Man akan bertambah.
- Pemain akan memasuki level yang lebih tinggi jika berhasil memakan seluruh *pac-dots* yang ada pada labirin. Permainan akan berakhir (game over) apabila nyawa Pac-Man habis, atau game telah tamat.

III. ALGORITMA BFS

Dalam kehidupan sehari-hari, banyak terdapat persoalan yang menuntut solusi berkaitan dengan pencarian jalur. Persoalan semacam ini membutuhkan algoritma yang mampu untuk mendeteksi sejumlah jalur yang mungkin dilalui dari titik awal menuju titik tujuan dan mengeluarkan hasil yang dianggap “pasti” (titik awal dan tujuan benar-benar terhubung) maupun hanya “mendekati”.

Algoritma BFS merupakan salah satu metode pencarian jalur yang cukup mampu untuk menghasilkan solusi masalah tersebut. Secara harfiah, BFS, atau Breadth-First Search dapat diartikan sebagai “pencarian yang melebar”. Maksud melebar di sini yaitu jika pencarian dianggap sebagai suatu pohon, di mana keadaan awal merupakan akar, BFS memiliki ciri khas di mana skema pencarian mendahulukan untuk memeriksa seluruh simpul hasil percabangan dalam suatu aras terlebih dahulu sebelum melanjutkan pencarian ke aras yang lebih dalam.

Algoritma BFS telah dimanfaatkan untuk memecahkan masalah dan membantu beberapa teori, contohnya :

1. Menemukan semua node dalam satu komponen terkoneksi.
2. Pengujian graf untuk bipartiteness.
3. Metode Ford-Fulkerson untuk menghitung *maximum flow* dalam *flow network*.
4. Pemrosesan citra.

III.1 Definisi Algoritma BFS

Dalam teori grafik, Breadth-First Search (BFS) adalah algoritma pencarian grafik yang dimulai pada node akar dan mengeksplorasi semua node tetangga. Lalu untuk masing-masing node terdekat, algoritma ini mengeksplorasi node tetangga mereka belum dijelajahi, dan seterusnya, sampai menemukan simpul tujuan.

III.2 Skema Pencarian Algoritma BFS

Secara umum, Algoritma BFS melakukan pencarian solusi dengan mengeksplorasi simpul-simpul tetangga lebih

dulu. Pencarian solusi dengan metode BFS pada pohon ruang status dijabarkan sebagai berikut :

1. Kunjungi simpul v ,
2. Kunjungi semua simpul yang bertetangga dengan simpul v terlebih dahulu.
3. Kunjungi simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya.

IV. PENGAPLIKASIAN ALGORITMA BFS

Penerapan algoritma BFS dapat dilakukan sebagai penentu langkah dari hantu, dan disesuaikan dengan karakteristik dari masing-masing hantu. Penerapan ini dibagi menjadi dua kondisi, yaitu kondisi pengejaran (*chase*) dan ketakutan (*frightened*).

IV.1 BFS pada Kondisi Pengejaran

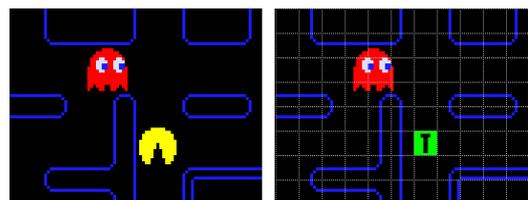
Kondisi pengejaran merupakan kondisi normal hantu yang ada dalam permainan Pac-Man. Pada kondisi ini, sesuai dengan sifat masing-masing, tiap hantu akan berusaha menangkap Pac-Man.

Terkait dengan penerapan BFS, kita dapat menganggap bahwa posisi hantu merupakan keadaan awal atau dapat dianggap sebagai akar dari pohon pencarian BFS. Simpul-simpul pada aras merupakan petak-petak dalam labirin, sedangkan arah jalur merupakan cabang-cabang pada pohon pencarian. Keadaan tujuan yang merupakan daun dari pohon pencarian disesuaikan dengan daerah cakupan tujuan langkah hantu, dapat berupa petak tempat Pac-Man berada, maupun petak-petak yang lain.

Algoritma BFS ini diaktifkan ketika hantu menemui persimpangan pada labirin. Hantu akan memilih jalur mana yang sesuai dengan objektif masing-masing, dan jalur yang dipilih tidaklah selalu jalur yang terdekat dari Pac-Man. Berikut objektif dan ilustrasi langkah tiap-tiap hantu yang terdapat pada permainan Pac-Man.

Algoritma BFS yang diterapkan selalu mencari jalur yang ada di depan hantu karena hantu tidak dapat berbalik secara tiba-tiba (kecuali pada awal mode ketakutan).

• Blinky



Gambar 4 Ilustrasi langkah Blinky

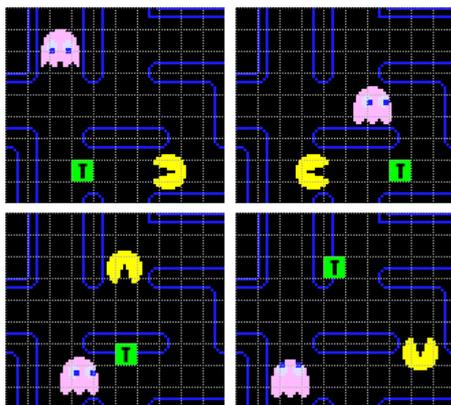
Sesuai dengan prinsip “mengejar dan menangkap”, Blinky akan menjadikan Pac-Man sebagai satu-satunya daerah tujuan solusi dari pencarian jalur yang diterapkan pada dirinya. Pada penerapan ini, Blinky akan secara otomatis mengikuti jalur menuju Pac-Man yang berada di

dekatnya.

Kita anggap Blinky memiliki suatu daerah batasan pencarian jalur seluas radius x petak dari tempatnya berada. Ketika menemukan suatu persimpangan, BFS yang terdapat pada Blinky akan mencari titik tujuan Blinky, yaitu Pac-Man. Apabila Pac-Man tidak terdapat dalam daerah radius tersebut, dengan kata lain, BFS tidak akan memberikan solusi dan Blinky akan memutuskan arah sendiri secara random.

Lain halnya ketika Pac-Man berhasil ditemukan dalam radius, Blinky akan mengikuti jalur yang diberikan oleh BFS, dan selanjutnya yang dilakukan Blinky yaitu mengikuti pergerakan Pac-Man sehingga jalur yang diikuti Blinky merupakan bayangan dari jalur yang ditempuh oleh Pac-Man.

• Pinky

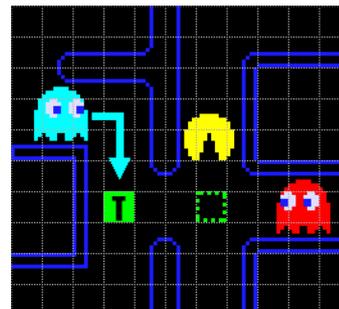


Gambar 5 Ilustrasi langkah Pinky

Sesuai dengan prinsip “penyergapan”, Pinky akan menjadikan beberapa petak di depan Pac-Man sebagai daerah tujuan solusi. Hal ini dilakukan agar Inky mampu memotong langkah Pac-Man dan menangkapnya

Seperti halnya Blinky, kita anggap Pinky memiliki suatu daerah batasan pencarian jalur seluas radius x petak dari tempatnya berada. Ketika menemukan suatu persimpangan, BFS yang terdapat pada Pinky akan mencari lokasi petak yang statusnya dicurigai akan dilangkahi oleh Pac-Man. Petak yang dicurigai ini biasanya merupakan petak yang berhadapan dengan mulut Pac-Man. Jika petak tidak terdapat dalam radius pencarian, maka Pinky akan memutuskan arah yang diambil secara random (diutamakan lurus).

• Inky

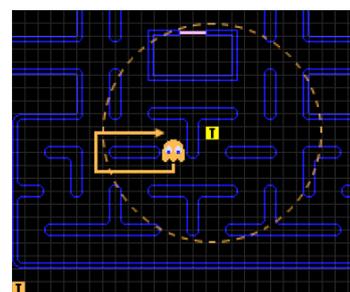


Gambar 6 Ilustrasi langkah Inky

Sesuai dengan prinsip ketidaktetapan *mood*-nya, Inky memiliki pola tujuan yang berubah-ubah. Inky tidak cukup berani untuk berhadapan langsung dengan Pac-Man sehingga Inky berpatokan awal pada dua petak di sekitar Pac-Man. Dalam *mood* menyerang, Inky akan memiliki tujuan dekat dua petak di depan Pac-Man, dan biasanya tujuan tersebut hanya berselang dua atau tiga petak. Jika Inky berada dalam *mood* menjauh, Inky justru akan menghindari Pac-Man dengan menghindari radius dua petak di sekitar Pac-Man tersebut.

Pada persoalan ini, BFS berperan ganda, karena selain BFS menyimpan jalur untuk dilalui hantu, BFS juga memiliki kemampuan menyimpan jalur untuk dihindari. Pada penyimpanan jalur untuk dilalui, biasanya hanya terdapat satu solusi, sedangkan pada penyimpanan jalur untuk dihindari, dapat terdapat beberapa solusi yang dapat ditempuh oleh Inky.

• Clyde



Gambar 7 Ilustrasi langkah Clyde

Penerapan BFS pada Clyde sedikit berbeda dengan ketiga hantu yang lain. Clyde memiliki jalur untuk patroli ketika ia berada pada radius kurang dari delapan petak dengan Pac-Man. Jalur patroli ini biasanya membentuk suatu *loop* atau lintasan yang tertutup. Simpul tujuan dalam BFS Clyde biasanya akan mengarah pada suatu persimpangan. Ketika sampai di tujuan tersebut, BFS akan diaktifkan dengan tujuan suatu persimpangan lagi, dan begitu seterusnya secara bergantian sehingga membentuk sirkuit. Bila sirkuit dipampakan sebagai persegi,

BFS pada Clyde akan dipanggil dengan modus pencarian simpul tujuan berupa titik sudut persegi.

Gerakan *loop* ini akan berhenti ketika Pac-Man keluar dari radius tersebut. Pada saat ini, Clyde akan menjadi seperti Blinky, yaitu mencari Pac-Man. Setelah Pac-Man masuk ke dalam radius, Clyde akan berpatroli kembali.

IV.2 BFS pada Kondisi Ketakutan



Gambar 8 Hantu pada saat ketakutan

Pada kondisi ketakutan, hantu akan berusaha menjauhi Pac-Man. BFS yang diterapkan pada saat ini berusaha mencari jalur pada petak apapun selain dari petak tempat Pac-Man berada.

Ketika hantu dimakan oleh Pac-Man, seperti terlihat pada gambar, hantu akan menyisakan kedua bola matanya saja. Pada saat ini BFS akan aktif dengan simpul tujuan yaitu tempat persembunyian hantu. Hantu akan berusaha mencari jalur untuk menuju ke tempat tersebut dan memulihkan diri untuk mengejar Pac-Man kembali.

V. KESIMPULAN

1. Algoritma pencarian solusi jalur, yaitu BFS dapat dipakai untuk mengatur pergerakan hantu dalam permainan Pac-Man.
2. Optimalisasi dilakukan dengan membatasi pencarian dalam batas tertentu, baik batas wilayah pencarian maupun waktu agar hantu tidak berpikir terlalu lama dalam permainan.
3. Penggunaan algoritma BFS dilakukan ketika hantu akan menemui persimpangan yang menuntut adanya keputusan pemilihan arah.
4. Solusi untuk algoritma BFS pada tiap hantu dalam permainan Pac-Man dapat berubah-ubah sesuai karakter dan kondisi.

VI. ACKNOWLEDGMENT

Penulis mengucapkan terima kasih terutama kepada Tuhan Yang Maha Esa karena berkat anugerah yang diberikan-Nya makalah ini dapat diselesaikan. Penulis juga mengucapkan terima kasih kepada Bapak Ir. Rinaldi Munir, M.T. selaku dosen pengajar kuliah IF3051 Strategi Algoritma karena berkat kuliah yang diberikan dan slide

kuliah yang ditulis oleh beliau makalah ini dapat disempurnakan.

DAFTAR PUSTAKA

- [1] http://en.wikipedia.org/wiki/Breadth-first_search
Waktu akses : 4 Desember 2010.
- [2] <http://en.wikipedia.org/wiki/Pac-Man>
Waktu akses : 2 Desember 2010.
- [3] <http://home.comcast.net/~jpittman2/pacman/pacmandossier.html#CH2%20-%20The%20Basics>
Waktu akses : 3 Desember 2010.
- [4] <http://id.wikipedia.org/wiki/Pac-Man>
Waktu akses : 2 Desember 2010.
- [5] <http://inst.eecs.berkeley.edu/~cs188/fa08/projects/search/searchproject.html>
Waktu akses : 2 Desember 2010.
- [6] <http://www.expert.tc/topic.php?id=54461#>
Waktu akses : 3 Desember 2010.
- [7] http://www.gunadarma.ac.id/library/articles/graduate/computer-science/2010/Artikel_21105199.pdf
Waktu akses : 3 Desember 2010.
- [8] <http://www.math.unl.edu/~s-dstolee1/Software/Pacman.pdf>
Waktu akses : 3 Desember 2010.
- [9] Slide kuliah IF2151 Strategi Algoritmik : BFS dan DFS

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, 6 Desember 2010

ttd



Emeraldy Widiyadi - 13508067