

# Penerapan Algoritma Dijkstra untuk Pemilihan Rute Jalan Terpendek pada Permainan Harvest Moon: Back to Nature

Ryan Yonata (13513074)  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
ryanyonata@students.itb.ac.id

**Abstrak** — Permainan *Harvest Moon: Back to Nature* memerlukan manajemen waktu dan tenaga yang baik agar pemain bisa menjalani tugas-tugas untuk membuat perkebunan sukses menjadi lebih efisien. Manajemen waktu dalam permainan *Harvest Moon: Back to Nature* dapat dilakukan dengan cara mencari rute berjalan terpendek agar waktu yang dihabiskan tidak terbuang percuma hanya untuk berjalan. Pencarian jalur terpendek ini dapat dilakukan dengan menggunakan algoritma pencarian jalur terpendek (shortest path), yaitu algoritma Dijkstra. Dengan algoritma Dijkstra, kita dapat mengetahui jalur terpendek dari suatu tempat ke seluruh tempat lain agar penggunaan waktu efisien.

**Kata kunci** — *Harvest Moon: Back to Nature*, Dijkstra, path, jalur, terpendek, graf

## I. PENDAHULUAN

Video game atau permainan video merupakan permainan elektronik yang melibatkan interaksi antara pengguna dengan mesin. Video game menggunakan joystick, keyboard, atau alat input lain sebagai perangkat input dan layar atau monitor sebagai perangkat output.

Tujuan utama video game tentu saja adalah sebagai hiburan bagi orang yang memainkannya. Selain untuk hiburan, video game juga dapat memberikan manfaat-manfaat lain yang bisa anda dapat seperti memberikan inspirasi, pendidikan, mengontrol emosi, melatih kemampuan pengambilan keputusan, sampai melatih logika berpikir. Semua manfaat itu dapat diambil sesuai dengan jenis video game yang dimainkan.

Seiring dengan berkembangnya zaman, perkembangan video game pun semakin pesat. Video game dikemas dengan teknologi yang semakin canggih, yang tadinya hanya merupakan permainan dua dimensi dengan layar monokrom, sampai sekarang dikemas dalam animasi tiga dimensi berwarna.

Salah satu video game yang sempat populer sekitar tahun 1995 – 2005 adalah permainan *Harvest Moon: Back to Nature*. *Harvest Moon: Back to Nature* adalah permainan yang cukup sederhana bergenre simulation/role playing yang mencerminkan simulasi perkebunan di

pedesaan.

Tujuan dari game ini adalah untuk mengatur perkebunan dan juga menjalin hubungan sosial dengan warga desa dan menyelesaikan tugas-tugas yang diberikan. Salah satu kesulitan dalam permainan ini adalah bagaimana pemain mengatur waktu untuk melakukan semua tugas karena hanya ada waktu yang sedikit (terbatas) setiap harinya. Pemanfaatan waktu yang baik bisa dilakukan dengan memilih rute berjalan yang tepat sehingga penggunaan waktu menjadi lebih sedikit.

Masalah pemanfaatan waktu dengan mengatur rute berjalan terdekat dan tercepat ini dapat diselesaikan dengan algoritma Dijkstra.

## II. DASAR TEORI

### A. Algoritma Dijkstra

Algoritma Dijkstra adalah algoritma pencarian lintasan terpendek yang paling terkenal. Algoritma Dijkstra dinamai sesuai dengan nama penemunya, Edsger Wybe Dijkstra. Dari naskah aslinya, algoritma Dijkstra diterapkan untuk mencari lintasan terpendek pada graf berarah. Namun, algoritma ini juga dapat tetap diterapkan untuk graf tak berarah.

Algoritma Dijkstra mencari lintasan terpendek dengan menggunakan sejumlah langkah. Algoritma ini menggunakan strategi *Greedy* sebagai berikut:

Pada setiap langkah, ambil sisi yang berbobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Lintasan dari simpul asal ke simpul yang baru haruslah merupakan lintasan yang terpendek diantara semua lintasannya ke simpul-simpul yang belum terpilih.

Ada beberapa versi algoritma Dijkstra yang ditulis di berbagai pustaka. Berikut merupakan kode algoritma Dijkstra.

(a) Deklarasi Struktur Data

```
const n = . . . {jumlah simpul dalam graf}
type matriks = array[1..n, 1..n] of integer
type tabel = array[1..n] of integer
```

(b) Algoritma Dijkstra

```
function Dijkstra(input M : matriks, a : integer) → tabel
{ Mencari lintasan terpendek dari simpul awal a ke semua simpul lainnya.
Masukan: matriks ketetangaan (M) dari graf berbobot G dan simpul awal a.
Keluaran: tabel D yang berisi panjang lintasan terpendek dari a ke semua simpul lainnya }

Deklarasi
D,S : array[1..n] of integer
i,j,k,min : integer

Algoritma
{ Langkah 0 inisialisasi }
for i ← 1 to n do
  S[i] ← 0
  D[i] ← M[a,i]
endfor

{ Langkah 1 }
S[a] ← 1 {masukkan simpul awal ke dalam S}
D[a] ← ∞ {jarak diberi nilai tak hingga}

{ Langkah 2, 3, . . . , n-1 }
for k ← 2 to n - 1 do
  {Cari simpul j sedemikian sehingga S[j] = 0 dan D[j] = Minimum{D[1],D[2],...,D[n]} }
  min ← D[1]
  j ← 1
  for k ← 2 to n- 1 do
    if (S[i] = 0) and (D[i] < min) then
      min ← D[i]
      j ← i
    endif
  endfor
  S[j]←i{Simpul j sudah terpilih dlm lintasan terpendek}

  {Hitung D[i] yang baru dari a ke simpul i !e S}
  for i ← 1 to n do
    if S[i] = 0 then
      if D[i] > (D[j] + jarak[j,i]) then
        D[i] ← D[j] + jarak[j,i]
      endif
    endif
  endfor
endfor

return D
```

Kompleksitas waktu algoritma Dijkstra adalah  $O(n^2)$ . Hal ini dijelaskan sebagai berikut:

- a. Instruksi di dalam kalang for pertama:
 

```
for i ← 1 to n do
  . . .
endfor
```

 dikerjakan sebanyak n kali, sehingga waktu komputasinya adalah  $O(n)$

- b. Kalang bersarang
  - i) Kalang for terluar
 

```
for i ← 2 to n-1 do
  . . .
endfor
```

 dilakukan sebanyak n - 2 kali
  - ii) Kalang for terdalam untuk menentukan D[i] minimum
 

```
for i ← 2 to n do
  . . .
endfor
```

 membutuhkan waktu dalam  $O(n)$
  - iii) Kalang for berikutnya untuk memperbarui nilai D[1..n]
 

```
for i ← 2 to n do
  . . .
endfor
```

 membutuhkan waktu dalam  $O(n)$

Dengan demikian, total waktu yang dibutuhkan adalah:

$$\begin{aligned}
 T(n) &= O(n) + (n - 2)[O(n) + O(n)] \\
 &= O(n) + O((n - 2)(n)) \\
 &= O(n) + O(n^2) \\
 &= O(n^2)
 \end{aligned}$$

B. Harvest Moon: Back to Nature

Permainan Harvest Moon adalah sebuah permainan yang dirilis oleh perusahaan asal Jepang bernama NATSUME. Pada dasarnya permainan Harvest Moon memiliki fitur-fitur yang hampir sama pada setiap serinya, diantaranya adalah menanam, beternak, berkencan dengan gadis di desa, memancing mengikuti festival, dan lain-lain. Akan tetapi, yang akan dibahas dalam makalah ini adalah *Harvest Moon: Back to Nature* yang dirilis pada tanggal 16 Desember 1996 pada platform PlayStation.



Gambar 1. Tampilan Awal Game Harvest Moon: Back to Nature

Seperti yang sudah dibahas, *Harvest Moon: Back to Nature* menyediakan beragam fitur permainan yang ada di dalamnya, mulai dari beternak, ayam,sapi,domba, dan ikan, memelihara anjing dan kuda, mengikuti berbagai macam festival dan kontes sepanjang tahun, bercengkrama dengan penduduk sekitar, mencari asangan, memasak, dan lain-lain.

Di dalam game, Jack (bisa dinamai bebas) memulai permainan dengan memiliki sebuah perkebunan yang kacau-balau dan sedikit uang untuk memulai perkebunan. Akhir dari game ini adalah pemain harus membuat sebuah perkebunan yang terawat dan sukses, serta menjalin persahabatan dengan penduduk desa lainnya di Mineral Town, desa dimana pemain tinggal. 3 faktor yang penting untuk diperhatikan adalah ketersediaan waktu, uang, dan tenaga. Tanaman yang penyiramannya perlu dilakukan setiap hari membutuhkan waktu dan tenaga yang tidak sedikit. Pemain harus meng-upgrade peralatan yang dimilikinya sehingga lebih efisien dalam penggunaan waktu dan tenaga. Untuk meng-upgrade peralatan juga membutuhkan waktu dan tenaga.



Gambar 2. Gameplay Harvest Moon: Back to Nature

Sumber penghasilan utama dalam game ini adalah dengan bertani dan beternak, tetapi bisa juga dengan melakukan penambangan logam di tempat yang bernama Mine.



Gambar 3. Pemain sedang menyiram tanaman



Gambar 4. Pemain beternak ayam

### III. PENERAPAN ALGORITMA DJIKSTRA PADA PENENTUAN RUTE TERPENDEK (SHORTEST PATH) DALAM PERMAINAN HARVEST MOON: BACK TO NATURE

Seperti yang sudah dijelaskan, pengelolaan waktu merupakan salah satu aspek terpenting dalam keberhasilan memainkan permainan Harvest Moon:Back to Nature ini. Oleh karena itu, pemain harus memiliki manajemen waktu yang baik dalam bermain permainan ini. Proses manajemen waktu dalam permainan ini dapat dilakukan dengan berbagai cara. Salah satunya adalah dengan memilih rute berjalan yang paling dekat agar waktu tidak terbuang sia-sia karena salah memilih jalan atau melewati jalan yang cukup jauh.

Persoalan ini disebut persoalan shortest path problem yang dapat diselesaikan dengan algoritma Dijkstra. Untuk melakukan pemecahan masalah. Harus dilakukan hal-hal sebagai berikut:

#### A. Menentukan elemen-elemen permasalahan

Elemen-elemen permasalahan dalam kasus ini adalah sebagai berikut:

##### 1. Jarak antara suatu tempat dengan tempat lain

Jarak antara suatu tempat dengan tempat lain dalam permainan ini dapat dihitung dengan jumlah langkah kaki (footsteps) yang dilakukan oleh pemain untuk mengunjungi suatu tempat. Jarak antar tempat dalam permainan ini dapat direpresentasikan sebagai matriks ketetanggaan sebagai berikut:

Pada suatu universitas, terdapat 13 fakultas yang membentuk tim untuk bertanding dalam liga olahraga mahasiswa tingkat universitas tersebut. Liga mahasiswa ini dilaksanakan dengan sistem setengah kompetisi. Liga mahasiswa dibagi menjadi dua divisi, yaitu divisi utama dan divisi 1. Divisi utama terdiri atas 7 tim dan divisi 1 terdiri dari 6 tim. Universitas hanya memiliki 3 buah lapangan dan setiap harinya pertandingan dilakukan dalam 2 kloter, yaitu pagi dan sore. Hari pertandingan dalam satu divisi harus berselang satu hari dan pertandingan divisi satu harus lebih dulu selesai dibandingkan dengan divisi utama. Keterbatasan ini membuat penjadwalan pertandingan menjadi sangat penting karena efektivitas penjadwalan memengaruhi lamanya kompetisi ini berlangsung.

Banyaknya tim yang bertanding dan pembagian menjadi dua divisi memungkinkan liga mahasiswa ini berlangsung lama. Semakin lama kompetisi berlangsung maka semakin banyak dana yang dibutuhkan. Dengan aturan aturan dan keterbatasan diatas, ketua pelaksana ingin mengetahui jumlah hari minimum yang mungkin untuk melaksanakan liga mahasiswa ini. Semakin sedikit hari pelaksanaan, maka semakin sedikit pula jumlah pengeluaran.

Permasalahan tersebut dapat diselesaikan dengan teori

graf, khususnya pewarnaan graf. Hal pertama yang harus dilakukan adalah merepresentasikan sistem pertandingan setengah kompetisi dengan menggambar graf. Kita misalkan peserta divisi utama adalah fakultas A, B, C, D, E, F, dan G. Sedangkan peserta divisi 1 adalah P, Q, R, S, T, dan U. Pertandingan sistem setengah kompetisi kedua divisi tersebut dapat direpresentasikan sebagai berikut.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A	-1	35	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
B	35	-1	38	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
C	-1	38	-1	57	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
D	-1	-1	57	-1	37	29	-1	-1	-1	-1	-1	-1	-1	-1	-1
E	-1	-1	-1	37	-1	-1	20	16	-1	-1	-1	-1	-1	-1	-1
F	-1	-1	-1	29	-1	-1	43	-1	-1	-1	-1	-1	-1	-1	-1
G	-1	-1	-1	-1	20	43	-1	-1	-1	-1	-1	24	-1	-1	-1
H	-1	-1	-1	-1	16	-1	-1	-1	22	-1	-1	-1	-1	-1	-1
I	-1	-1	-1	-1	-1	-1	-1	22	-1	20	-1	-1	-1	-1	-1
J	-1	-1	-1	-1	-1	-1	-1	-1	20	-1	21	-1	26	-1	-1
K	-1	-1	-1	-1	-1	-1	-1	-1	-1	21	-1	-1	-1	21	-1
L	-1	-1	-1	-1	-1	-1	24	-1	-1	-1	-1	-1	-1	21	-1
M	-1	-1	-1	-1	-1	-1	-1	-1	-1	26	-1	-1	-1	18	-1
N	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	21	21	18	-1	28
O	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	28	-1

Gambar 5 Matriks Ketetangaan Tempat

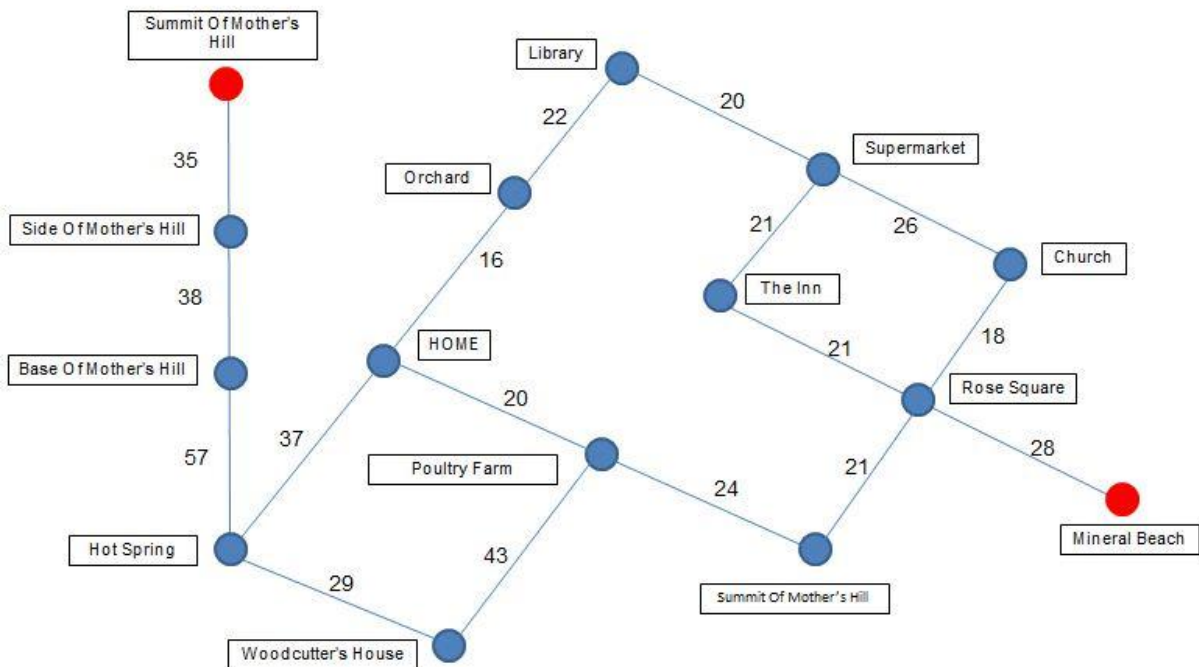
B. Representasi dengan Graf

Setelah mendapatkan matriks ketetangaan, lalu representasikan matriks tersebut dengan graf. Graf yang dibuat menyesuaikan peta permainan berikut ini:



Gambar 6 Peta Permainan

Representasi peta dan matriks ketetangaan tersebut dengan graf adalah dengan gambar di bawah ini:



Gambar 7. Representasi dengan Graf

C. Penyelesaian dengan Algoritma Dijkstra

Dengan menggunakan algoritma Dijkstra dan menentukan titik awal adalah titik A (Summit of Mother's Hill), maka kita bisa mendapatkan jarak terdekat dari titik tersebut ke semua titik (tempat).

Hasil algoritma Dijkstra dapat dilihat dalam tabel-tabel di bawah ini:

Lelaran	Simpul yang Dipilih	Lintasan	S															
			A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
Inisial	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	A	A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	B	AB	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	C	ABC	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
4	D	ABCD	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	
5	F	ABCDF	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	
6	E	ABCDE	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
7	H	ABCDEH	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
8	G	ABCDEG	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	
9	I	ABCDEHI	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	
10	J	ABCDEHIJ	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	
11	K	ABCDEHIJK	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	
12	L	ABCDEGL	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
13	N	ABCDEGLN	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0
14	M	ABCDEGLNM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Tabel 1 Proses pemilihan simpul dengan algoritma djikstra

Lelaran	Simpul yang Dipilih	Lintasan	S															
			A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
Inisial	-	-	∞	35	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	A	A	∞	35	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	B	AB	∞	35	73	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
3	C	ABC	∞	35	73	130	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
4	D	ABCD	∞	35	73	130	167	159	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
5	F	ABCDF	∞	35	73	130	167	159	202	∞	∞	∞	∞	∞	∞	∞	∞	∞
6	E	ABCDE	∞	35	73	130	167	159	187	183	∞	∞	∞	∞	∞	∞	∞	∞
7	H	ABCDEH	∞	35	73	130	167	159	187	183	205	∞	∞	∞	∞	∞	∞	∞
8	G	ABCDEG	∞	35	73	130	167	159	187	183	205	∞	∞	211	∞	∞	∞	∞
9	I	ABCDEHI	∞	35	73	130	167	159	187	183	205	225	∞	211	∞	∞	∞	∞
10	J	ABCDEHIJ	∞	35	73	130	167	159	187	183	205	225	246	211	251	∞	∞	∞
11	K	ABCDEHIJK	∞	35	73	130	167	159	187	183	205	225	246	211	251	267	∞	∞
12	L	ABCDEGL	∞	35	73	130	167	159	187	183	205	225	246	211	251	232	∞	∞
13	N	ABCDEGLN	∞	35	73	130	167	159	187	183	205	225	246	211	250	232	260	∞
14	M	ABCDEGLNM	∞	35	73	130	167	159	187	183	205	225	246	211	250	232	260	∞

Tabel 2. Nilai setiap proses pemilihan simpul

Dari hasil tabel tersebut dapat dilihat bahwa hanya terjadi 14 lelaran karena proses terakhir sudah pasti diisi oleh simpul O. Dari tabel tersebut dapat diketahui bahwa:

Jarak terdekat dari simpul A ke simpul:

- B = AB, 35 langkah
- C = ABC, 73 langkah
- D = ABCD, 130 langkah
- F = ABCDF, 159 langkah
- E = ABCDE, 167 langkah
- H = ABCDEH, 183 langkah
- G = ABCDEG, 187 langkah
- I = ABCDEHI, 205 langkah
- L = ABCDEGL, 211 langkah
- J = ABCDEHIJ, 225 langkah
- N = ABCDEGLN, 232 langkah
- K = ABCDEHIJK, 246 langkah
- M = ABCDEGLNM, 250 langkah
- O = ABCDEGLNO, 260 langkah

Dengan representasi huruf-huruf tersebut mewakili tempat-tempat berikut ini:

- A = Summit of Mother's Hill
- B = Side of Mother's Hill
- C = Base of Mother's Hill
- D = Hot Spring
- E = Home
- F = Woodcutter's House
- G = Poultry Farm
- H = Orchard
- I = Library
- J = Supermarket
- L = The Inn
- L = Ranch
- M = Church
- N = Rose Square
- O = Mineral Beach

#### IV. KESIMPULAN

Penerapan algoritma Dijkstra sangat banyak dalam kehidupan sehari-hari sesuai dengan tujuannya yaitu mencari jarak terdekat ke suatu titik. Penerapan algoritma Dijkstra ini dapat kita temui pada permainan video (video game), salah satunya adalah pada permainan Harvest Moon: Back to Nature. Dengan menggunakan algoritma Dijkstra kita dapat mengetahui jarak terdekat dari suatu tempat ke tempat lain. Dari hasil pembahasan yang dijadikan tempat acuan adalah titik A (Summit of Mother's Hill). Sehingga kita bisa mengetahui rute berjalan dari titik tersebut ke titik atau tempat lainnya agar tidak membuang waktu dan memanfaatkannya secara efisien. Pemanfaatan waktu yang efisien pada permainan ini akan sangat mendukung kita untuk memenangkan permainan ini.

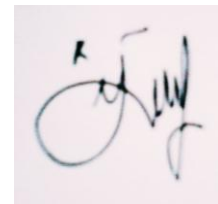
#### REFERENSI

- [1] Munir, Rinaldi, *Diktat Kuliah IF2211, Strategi Algoritma*, Program Studi Teknik Informatika, STEI, ITB, 2009
- [2] <http://komunikasi.us/index.php/course/3276-perkembangan-video-game> diakses pada 4 Mei 2015 pukul 15.10
- [3] <http://rinalhehe.blogspot.com/2012/04/harvestmoon-back-to-nature.html> diakses pada 4 Mei 2015 pukul 21.02

#### 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, 4 Mei 2015



Ryan Yonata (13513074)