

Penerapan Program Dinamis dalam Pembuktian Pencarian Nilai Optimum Permainan Ishitori

Charles Hariyadi(13505105)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung

Jl.Ganesha 10, Bandung

E-mail : if15105@students.if.itb.ac.id

ABSTRAK

Saat ini semakin banyak bermunculan program – program permainan komputer yang ditujukan untuk kesenangan pemainnya. Permainan- permainan itu biasanya banyak dibuat berdasarkan permainan yang dimainkan oleh masyarakat. Salah satu permainan itu adalah “ishitori”, ishitori merupakan permainan yang cukup digemari oleh masyarakat Jepang. Permainan yang dulunya sangat digemari itu sekarang telah merambah ke dunia komputer dan dapat dimainkan secara web-base. Permainan ini juga dikenal dengan permainan “get last, lost”. Permainan ini biasanya dimainkan oleh 2 orang dimana salah satu pemain berperan sebagai penentu jumlah angka(kita sebut saja JML) yang akan dimainkan, biasanya berkisar antara 10- ~(tak terhingga). Dalam permainan ini, pemain secara bergantian mengurangi JML dengan 1, 2 atau 3. Pemain yang mendapat giliran terakhir untuk mengurangi JML menjadi 0 merupakan pemain yang kalah, demikian sebaliknya pemain yang berhasil membuat pemain lainnya mendapat giliran terakhir untuk mengurangi JML adalah pemenangnya. Dalam hal ini, karena permainan ini diimplementasikan ke dalam komputer(AI atau Artificial Intelligence), maka salah satu pemain adalah komputer. AI dalam hal ini melakukan proses pengurangan dengan menerapkan Algoritma Program Dinamis agar mendapatkan nilai optimum yang mampu membuat dirinya menang. Makalah ini akan membahas bagaimana algoritma Program Dinamis mampu menghasilkan nilai optimum dan menang dari pemain.

Kata kunci : *Ishitori, Artificial intelligence, Program Dinamis*

1.PENDAHULUAN

Game Ishitori adalah permainan yang biasanya dimainkan oleh masyarakat Jepang untuk pengisi waktu luang, sekaligus untuk melakukan sedikit pengujian pada kemampuan berpikir anak(bilamana dimainkan bersama anak-anak). Di Jepang, game ini bahkan bisa dimainkan dimana saja tanpa memerlukan komputer, hanya dengan bermodalkan ingatan ataupun kertas dan pena. Dengan perkembangan teknologi yang pesat, kita juga dapat menikmatinya di komputer dengan melawan AI. Tetapi pada game-game ishitori yang ada saat ini terdapat sedikit kekurangan, dalam permainan ini biasanya AI selalu melakukan pilihan langkah yang terkesan tergesa-gesa, sehingga kadang kala, bila terjadi sedikit ” kebocoran” pilihan, maka pemain akan menang. Dalam makalah ini, penulis ingin membuktikan bahwa fungsi ($x \bmod 4 = 1$) merupakan fungsi yang sangat ideal untuk AI ataupun pemain dalam memenangkan permainan ishitori ini(x dalam hal ini merepresentasikan jumlah nilai pertamayang dibangkitkan). Penulis menggunakan algoritma Program dinamis untuk membuktikan hal tersebut, Karena Program dinamis merupakan salah satu algoritma yang cocok untuk melakukan solusi pencarian nilai optimum selain *exhaustive search*. nilai optimal dalam pengurangan akan dimasukkan ke dalam F_i (dalam hal ini F_i adalah jumlah nilai yang akan dikurangi sampai 0). Dalam makalah ini, penulis merepresentasikan solusi optimum sebagai F_i .

2.TERMINOLOGI

- JML merupakan representasi dari nilai optimum setelah pengurangan yang dilakukan oleh pemain sejumlah 1.
- Tahap(k) merupakan proses-proses pengurangan nilai dari f_i .
- X_i merupakan status optimum yang dipilih. ($i=1,2,3,\dots,n$)

Peraturan ISHITORI adalah sebagai berikut :

Salah satu pemain menetapkan nilai awal yang akan dikurangkan sampai mencapai nol, setelah itu ditentukan siapa yang akan melakukan pengurangan dahulu, apakah pemain ataupun komputer. Angka pengurang berkisar antara 1-3, pemain ataupun AI boleh memilih salah satu dari ke 3 angka tersebut(Dalam permainan saat ini ditambahkan fungsi waktu yang menyebabkan permainan menjadi lebih mendebarkan dan menarik). Permainan akan berakhir pada saat JML mencapai nilai 0, pemain dinyatakan kalah bila ia merupakan pengurang terakhir dan menang apabila ia mampu menyebabkan pemain lainnya menjadi pengurang terakhir.

3.PEMBUKTIAN OPTIMASI FUNGSI(X MOD 4==1) DENGAN MENGGUNAKAN ALGORITMA PROGRAM DINAMIS

3.1 Model Pembuktian

Pembuktian solusi ini akan menggunakan Program dinamis mundur.

Dalam permainan ini, penulis mengambil beberapa asumsi :

- Angka yang dibangkitkan adalah 24
- Dalam pembuktian ini, penulis menuliskan tindakan dari AI(komputer)
- Pemain diasumsikan selalu mengambil/mengurangi sebanyak 1 angka
- Komputer merupakan pemain yang pertama kali melakukan pengurangan.
- Permainan berhenti sampai JML berjumlah 0.
- **Perpindahan setiap status, selalu ada 3 pilihan, yaitu pengurangan terhadap 1,2 dan 3**

Dari tahap ke-1 komputer akan melakukan pengurangan terhadap nilai yang dibangkitkan. Tetapi dalam usahanya melakukan pengurangan itu, komputer akan melakukan pertimbangan apakah hasil pengurangannya bila dibagi dengan 4 akan menghasilkan sisa 1 ($x \text{ mod } 4 = 1$). Komputer akan memilih nilai pengurang yang memenuhi hal tersebut. Untuk tahap ke-2 sampai tahap ke-n komputer akan terus melakukan pertimbangan tersebut. Relasi Rekurens untuk persoalan ini adalah :

$$f_0(y) = 0, y = 0,1,2, \dots n \text{ (basis)}$$

$$f_k(y) = \infty \text{ (basis)}$$

$$f_k(y) = \text{optimum}\{f_{k-1}(y), pk + f_{k-1}(y - w_k)\}, \\ k = 1,2, \dots, n \text{ (rekurens)}$$

Program akan memilih antara pilihan pengurang (1 sampai 3) dan akan melakukan rekurens bila salah satu kondisi yaitu ($x \text{ mod } 4=1$) tidak terpenuhi, dan program akan terus berulang sampai nilai JML menjadi 0.

Tahap 1 :

Tabel Tahap 1

Y	Solusi optimum			
	0	F1(s)	X1	JML
1	23	23	0	22
2	22	22	0	21
3	21	21	0	20

Dalam tahap ini, terdapat 3 status (1,2,3) dari status 0 hanya terdapat 3 pilihan, jadi otomatis status berpindah.

Tahap 2 :

Tabel Tahap 2

y	Solusi optimum					
	1	2	3	F2(s)	X2	JML
4	21	20	19	21	1	20
5	20	19	18	18	3	17
6	19	18	17	17	3	16

Dalam tahap ini terdapat 3 status(2,3,4) status 4 melakukan pilihan sesuai dengan ($x \text{ mod } 4 ==1$) yaitu 21, begitu juga dengan status 6.

Tahap 3 :

Tabel Tahap 3

y	Solusi optimum					
	4	5	6	F2(s)	X2	JML
7	19	18	17	17	6	16
8	16	15	14	14	6	13
9	15	14	13	13	6	12

Dalam tahap 4 status 7 dan 9 merupakan status yang sesuai dengan syarat ($x \text{ mod } 4 ==1$) yaitu 17 dan 13. Akan tetapi status 8 yang tidak memenuhi salah satu dari criteria itu, mengambil nilai optimum dengan memilih nilai solusi yang paling kecil.

Tahap 4 :

Tabel Tahap 4

y				Solusi optimum		
	7	8	9	F2(s)	X2	JML
10	15	14	13	13	9	12
11	12	11	10	10	9	9
12	11	10	9	9	9	8

Dalam tahap ini, status 10 dan 12 merupakan status yang sesuai dengan prasyarat ($x \bmod 4 == 1$), karena itu dipilihlah pengurangan terhadap 3 dari solusi status sebelumnya, untuk kedua status itu.

Tahap 5:

Tabel Tahap 5

y				Solusi optimum		
	10	11	12	F2(s)	X2	JML
13	11	10	9	9	12	8
14	8	7	6	6	12	5
15	7	6	5	5	12	4

Dalam tahap 5 terjadi pengulangan kasus seperti tahap-tahap sebelumnya. Tapi dari sini kita dapat melihat, bahwa status 14 merupakan status yang kalau dipandang dari sudut pemain, merupakan status yang memenuhi syarat ($x \bmod 4 == 1$). Karena itu, mari kita lihat kelanjutan dari kasus ini.

Tahap 6 :

Tabel Tahap 6

y				Solusi optimum		
	13	14	15	F2(s)	X2	JML
16	7	6	5	5	15	4
17	4	3	2	2	15	1
18	3	2	1	1	15	0

Dalam tahap 6, pada saat status 18 tersisa satu angka, yang mau tidak mau harus diambil oleh pemain, satu solusi telah ditemukan oleh komputer kalau mengikuti langkah tersebut. Satu kemenangan untuk komputer telah tercipta, sekarang kita lanjutkan ke tahap berikutnya.

Tahap 7 :

Tabel Tahap 7

y				Solusi optimum		
	16	17	18	F2(s)	X2	JML
19	3	2	1	1	18	1
20	0	-	-	0	16	-
21	-	-	-	-	-	-

Dalam tahap ke 7 kembali ditemukan 1 status kemenangan untuk komputer yaitu status 19(status yang selalu mengikuti syarat ($x \bmod 4 == 1$)). Dalam tahap ini juga terlihat satu kenyataan bahwa pemain juga mampu memenangkan 1 status dengan mengikuti syarat ($x \bmod 4 == 1$). Terlihat pada status 20 bahwa mau tidak mau, komputer dipaksa untuk menghabiskan 1 nilai terakhir yang ditinggalkan oleh pemain.

Dari tahap-tahap diatas, ditemukan solusi dengan melalui urutan sebagai berikut :

$$x1 = 0 \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow 10 \rightarrow 13 \rightarrow 16 \rightarrow 19$$

$$x2 = 0 \rightarrow 3 \rightarrow 6 \rightarrow 9 \rightarrow 12 \rightarrow 15 \rightarrow 18$$

Solusi-solusi ini ditemukan dengan menerapkan prasyarat ($x \bmod 4 == 1$). Terbukti kalau fungsi ($x \bmod 4 == 1$) dan Program Dinamis merupakan salah satu cara yang mangkus dan manjur untuk menyelesaikan persoalan ishitori.

3.2 Potongan Algoritma

```

Function Pil_computer(int n) → boolean
{melakukan pencocokan pengurangan nilai apakah
sesuai dengan prasarat
Masukan : JML masukan
Keluaran : nilai hasil pengurangan}
Deklarasi:
Stop, kedudukan : Boolean
Algoritma
Stop ← false
While(!stop) do
    If_(n mod 4 == 1 ) then

```

```

        kedudukan← true

    Else

        kedudukan← false

    Endif

Endwhile

```

```

        {yang diambil merupakan nilai yang terkecil
        yang merupakan hasil dari pengurangan}

    endif

end procedure

```

```

Procedure Ishitori(I/O int hsl)

{ melakukan pengurangan terhadap angka yang
dibangkitkan

Masukan : angka yang akan dikurangkan

Keluaran : dihasilkan suatu angka hasil
pengurangan}

Deklarasi:

Stop : boolean

Test : boolean

I : integer

Algoritma:

I=1

While (i<>3) do

    Test ←Pil_computer(hsl)

    If (Test = false) then

        Ishitori(hsl-i+1)

    Else

        {cetak nilai yang benar ke main program}

    endwhile

{penanganan pada saat tidak ditemukannya kecocokan
dengan prasyarat x mod 4 == 1, maka program akan
secara langsung melakukan tindakan untuk mengambil
nilai yang terkecil }

If (Test=false) then

    Hsl=hsl-3

```

4.KESIMPULAN

Setelah melakukan pembuktian dengan menggunakan algoritma Program dinamis, terbukti bahwa prasarat ($x \bmod 4 == 1$) merupakan salah satu solusi yang sangat cocok untuk menyelesaikan program ishitori dengan kemenangan. Dapat dilihat dari table pada Bab 3 bahwa semua keadaan menang yang dicapai baik oleh komputer maupun oleh pemain merupakan suatu keadaan yang dicapai melalui kecocokan terhadap syarat ($x \bmod 4 == 1$).

Dari pembuktian ini, penulis menemukan bahwa pada kenyataannya algoritma Program Dinamis merupakan sebuah algoritma yang sangat cocok untuk melakukan penyelesaian terhadap program ishitori. Tetapi kemangkusan untuk masalah yang memerlukan tahap lebih dari n mungkin belum begitu memuaskan. Karena itu, sampai saat ini penulis masih terus berusaha menemukan algoritma yang mangkus untuk penyelesaian masalah ishitori untuk nilai pembangkitan yang lebih besar dari 1000. Penemuan terhadap algoritma mangkus untuk menyelesaikan persoalan ishitori ini merupakan sebuah cabaran baru.

5.REFERENSI

Munir Rinaldi.2006. Strategi Algoritmik IF2251, Bandung : Penerbit ITB.