Aplikasi Dynamic Programming dalam Decision Making pada Reinvestment Problem

Faisal Ibrahim Hadiputra (13509048)[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*faisal.hadiputra@students.itb.ac.id*

***Abstrak*—Pengambilan keputusan untuk memberikan investasi yang tepat merupakan kunci kesuksesan seorang manajer atau investor. Untuk mengambil keputusan yang benar, diperlukan peramalan ilmiah dengan pemodelan matematis. Pada makalah ini, dibahas mengenai penggunaan *dynamic programming* dan dikombinasikan dengan konsep *Time Value of Money* untuk meramalkan Net Present Value dari sebuah investasi. Selain itu, pada makalah ini juga akan merancang sebuah studi kasus sederhana sebagai instansiasi persoalan agar lebih mudah memahami metode penyelesaian masalah ini. Pada studi kasus ini didapatkan bahwa *dynamic programming* dapat memudahkan kita untuk bekerja lebih terstruktur dalam menyelesaikan permasalahan investasi, serta dapat diintegrasikan dengan konsep *Time Value of Money*.**

***Kata Kunci*—Dynamic Programming, Decision Analysis, Decision Tree, Time Value of Money, Discounting Factor, Investmen Problem, Net Present Value.**

I. PENDAHULUAN

Saat ini, persoalan mengenai pengambilan keputusan telah menjadi sesuatu yang umum dibahas dalam dunia bisnis dan manajemen. Dalam hal ini, apabila seorang manajer dituntut untuk melakukan peramalan terhadap hasil yang akan dicapai sebelum mengambil keputusan dan peramalan ini harus tetap bersifat rasional. Pada manajemen klasik, persoalan pengambilan keputusan tidak difasilitasi oleh model matematis, sehingga lebih mengandalkan intuisi dan pengalaman untuk mengambil keputusan. Namun, pada konsep manajemen modern, seorang manajer diharuskan berpikir rasional untuk mengambil keputusan. Untuk itu, diperlukan alat pemodelan matematis yang dapat merepresentasikan persoalan dalam mengambil keputusan. Beberapa model matematis yang sering digunakan adalah *Linear Programming* dan *Decision Tree Analysis*.

Permasalahan pengambilan keputusan yang umum dihadapi oleh seorang manajer adalah memilih satu atau lebih proyek untuk menginvestasikan modal perusahaan yang sedang dimiliki agar mendapatkan *Return of Investment* (ROI) yang optimal.

Dalam teori ekonomi, *Net Present Value* (NPV) merupakan konsep yang paling umum untuk mengevaluasi investasi. Pada konsep ini, dijelaskan perubahan nilai uang dari waktu ke waktu atau disebut *Time Value of Money* (TVOM).

Permasalahan yang dibahas kali ini adalah persoalan mengenai pemilihan proyek secara repetitive yang akan diinvestasikan dari tahun ke tahun dalam jangka waktu beberapa tahun. Persoalan ini dapat diformulasikan dalam bentuk tahap per tahap. Salah satu algoritma yang dapat memfasilitasi persoalan ini adalah *Dynamic Programming* (Pemrograman Dinamis).

Oleh karena itu, akan dibahas penyelesaian permasalahan mengenai pengambilan keputusan investasi berulang dengan *Dynamic Programming*.

II. DASAR TEORI

A. Dynamic Programming

Program Dinamis (*Dynamic Programming*) adalah metode pemecahan masalah dengan cara menguraikan solusi dari persoalan menjadi sekumpulan langkah/tahap sedemikian sehingga solusi dari persoalan dapat dipandang dari sekumpulan keputusan yang saling berkaitan. Pada penyelesaian persoalan dengan metode ini terdapat beberapa kondisi:

1. Terdapat sejumlah berhingga pilihan yang mungkin.
2. Solusi pada setiap tahap yang dibangun dari hasil solusi tahap sebelumnya.
3. Kita menggunakan persyaratan optimasi dan kendala untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap.

Program dinamis harus memenuhi prinsip optimalitas, yaitu: jika solusi total optimal, maka bagian solusi sampai tahap ke-k jugaoptimal.

Beberapa elemen dari sebuah program dinamis adalah sebagai berikut:

1. Tahapan-tahapan (k) dengan $k \in \{1,2,...,n\}$.
2. Variable keputusan $x_1,x_2,\ldots,x_n$.
3. Fungsi ongkos untuk setiap tahap dengan definisi rekursif $f_n = c_{n,n-1} + f_{n-1}$. Dengan f_i adalah ongkos dari pada tahap i dan $c_{a,b}$ adalah dari tahap a ke tahap b.

Dalam penyelesaian persoalan, program dinamis memiliki dua jenis pendekatan:

1. Program dinamis maju, yaitu bergerak maju mulai dari tahap 1, terus maju ke tahap 2, 3, dan

seterusnya sampai tahap n. Runtunan peubah keputusan adalah $x_1, x_2, \ldots, x_n$.

2. Program dinamis mundur, yaitu bergerak mundur mulai tahap n, terus mundur ke tahap n-1, n-2, dan seterusnyya sampai tahap 1. Runtunan peubah keputusan adalah $x_n, x_{n-1}, \ldots, x_1$.

Pada praktiknya, terdapat dua jenis program dinamis ditinjau dari data pada persoalannya, yaitu:

1. *Deterministic Dynamic Programming*, yaitu program dinamis untuk menyelesaikan persoalan dengan fungsi ongkos yang terdefinisi dengan jelas dan pasti untuk setiap aksi yang dipilih untuk suatu variabel keputusan.
2. *Stochastic Dynamic Programming*, yaitu program dinamis untuk menyelesaikan persoalan dengan fungsi ongkos yang memiliki ketidakpastian nilai untuk setiap aksi yang dipilih untuk suatu variable keputusan. Dalam hal ini, terdapat daftar kemungkinan nilai dan probabilitas untuk nilai tersebut.

B. Decision Analysis

Salah satu metode yang paling umum dan mudah digunakan dalam persoalan pengambilan keputusan adalah *Decision Analysis*. Berikut ini adalah elemen yang perlu ditentukan pada metode *Decision Analysis*:

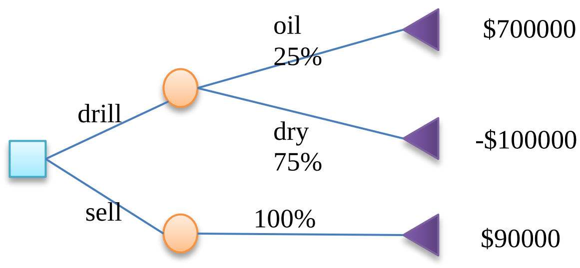
1. *Decision Maker*, orang yang mengambil keputusan.
2. *Alternatives*, daftar alternatif keputusan yang harus diambil oleh *Decision Maker*.
3. *State of Nature*, seluruh kondisi yang mungkin diluar kuasa *Decision Maker*.
4. *Prior Probability*, probabilitas kemungkinan terjadinya setiap *State of Nature*.
5. *Payoff*, *cost* atau *reward* yang dihasilkan untuk setiap kombinasi alternatif dan stata alam.

Pada dasarnya, metode ini adalah mendaftarkan semua kemungkinan kombinasi pengambilan keputusan dan memilih kombinasi yang memberikan *payoff* terbaik.

Salah satu alat yang dapat digunakan untuk melakukan analisis keputusan adalah *Decision Tree*. Elemen-elemen yang terdapat pada pohon keputusan adalah sebagai berikut :

1. *Decision nodes*, merupakan simpul yang mewakili keadaan untuk memilih alternatif yang ada.
2. *Chance nodes*, merupakan simpul yang mewakili percabangan terjadinya setiap *state of nature*.
3. *End nodes*, merupakan simpul yang mewakili *payoff* hasil kombinasi alternatif dan stata alam.

Berikut ini adalah contoh gambar Decision Tree untuk permasalahan apakah lebih baik mengebor atau menjual pengilangan minyak :



Dalam hal ini, *decision nodes* dilambangkan sebagai persegi, *chance nodes* dilambangkan sebagai lingkaran, dan *end nodes* dilambangkan sebagai segitiga. Permasalahan pada decision tree di atas adalah memilih apakah sebaiknya kita memilih untuk mengebor atau menjual sebuah kilang minyak. Penentuan ini dilakukan dengan cara menghitung keuntungan yang diperoleh pada setiap alternatif yang diambil.

Perhitungan alternatif yang diambil ini memenuhi rumus berikut:

$$R_i = \sum(\text{probability} \times \text{payoff})$$

Untuk kasus ini berikut ini adalah nilai R untuk setiap alternatif:

- R_1 = 0.75 * \$700000 + 0.25 * (-\$100000) = \$500000
- R_2 = 1.00 * \$90000 = \$90000

Dengan demikian, karena $R_1 > R_2$, alternatif 1 yang sebaiknya dipilih oleh *Decision Maker*.

C. Time Value of Money

Pada teori ekonomi, nilai uang akan terus berubah dari waktu ke waktu. Perubahan nilai uang ini dipengaruhi oleh *discounting factor* atau *compounding factor*. Dalam hal ini kedua faktor tersebut bergantung pada tingkat pengembalian yang berkorelasi dengan suku bunga bank.

Pada konsep teori *Time Value of Money*, terdapat beberapa istilah/besaran sebagai berikut:

1. Present Value (PV) : Nilai uang saat ini.
2. Future Value (FV) : Nilai uang di masa depan
3. Rate of Return (r) : Tingkat pengembalian suku bunga
4. Tahun ke- (n) : tahun di mana nilai future value dipetakan.

Keempat besaran tersebut memiliki hubungan sebagai berikut:

$$PV = \frac{FV}{(1+r)^n}$$

$$FV = PV(1+r)^n$$

Dalam hal ini, untuk tingkat suku bunga positif, apabila nilai uang masa akan datang dipetakan ke masa kini, nilai uang tersebut akan lebih kecil.

Pada praktiknya, teori *Time Value of Money* ini digunakan untuk menentukan apakah sebuah investasi itu layak dilakukan atau tidak. Hal ini dilakukan dengan menghitung *Net Present Value* (NPV). Rumus dari NPV adalah sebagai berikut:

$$NPV = \sum_{i=0}^{n} PV_i$$

$$NPV = \sum_{i=0}^{n} \frac{FV_i}{(1+r)^i}$$

Berikut ini adalah contoh kasus investasi sebuah mesin dengan awal investasi adalah \$100000 dan diramalkan akan beroperasi selama 4 tahun dan akan menghasilkan keuntungan sebagai berikut :

Tahun	Keuntungan
1	-\$50000
2	\$150000
3	\$100000
4	\$50000

Apabila nilai tingkat pengembaliannya adalah 10%, maka perhitungan NPVnya adalah sebagai berikut:

Tahun	Future Value	Present Value (r = 0.1)
0	-\$100000	-\$100000
1	-\$50000	-\$45454.5
2	\$150000	\$123966.9
3	\$100000	\$75131.48
4	\$50000	\$34150.67
NPV		\$87794.55

Pada kasus ini karena NPV > 0, maka investasi untuk mesin ini adalah layak. Pada praktiknya, nilai NPV ini juga dapat digunakan untuk membandingkan dua investasi atau lebih dan investasi yang NPVnya lebih besar adalah investasi yang sebaiknya dipilih.

III. Definisi Permasalahan

Permasalahan yang diangkat pada makalah ini adalah persoalan seorang manajer yang memiliki sejumlah n proyek. Tiap proyek memiliki dana awal p_1, p_2, p_3,…,p_n. Dalam kasus ini, manajer harus memilih proyek yang dijalankan untuk satu tahun dengan sejumlah m aksi a_1, a_2, a_3, …, a_m. Dalam hal ini, setiap aksi a_i merepresentasikan pendistribusian dana ke proyek yang akan dijalankan.

Untuk setiap aksi yang dilakukan, terdapat daftar nilai pengembalian *Return of Investment* (ROI) untuk setiap proyek yang dipilih oleh aksi dengan probabilitas tertentu. Misalnya untuk aksi a_1 daftar nilai ROInya adalah sesuai tabel berikut:

ROI	Probabilitas
b_{11}%	c_{11}%
b_{12}%	c_{12}%
…	…
b_{1k}%	c_{1k}%

Dalam hal ini, nilai ROI setiap proyek untuk setiap tahun akan berbeda-beda. Selain itu, diketahui pula tingkat pengembalian sebesar r. Tujuan yang dicari dari urutan aksi yang diperlukan untuk setiap tahun untuk jangka waktu m tahun. Selain itu, aksi yang diambil tidak boleh diulang pada tahun berikutnya.

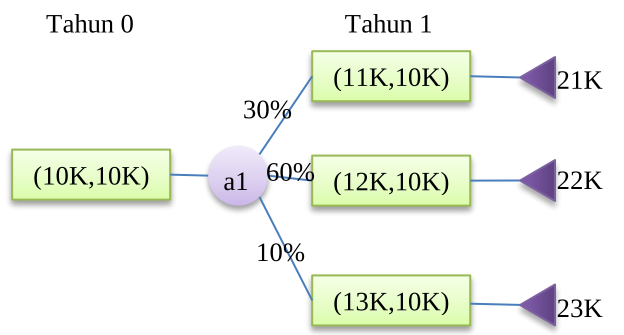
IV. Metodologi Pemecahan Masalah

Pada makalah ini, metodologi yang digunakan dalam pemecahan permasalahan investasi berulang yang dibahas pada bab sebelumnya adalah dengan mengkombinasikan *dynamic programming* dengan fungsi biaya yang mengikuti aturan *Time Value of Money*.

Sebelum menggunakan dynamic programming, kita dapat mendefinisikan *decision tree* untuk memodelkan permasalahan di atas dengan elemen sebagai berikut:

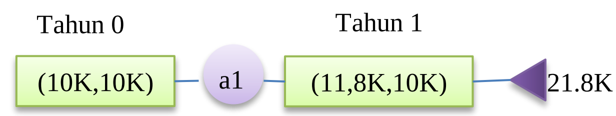
1. *Decision nodes*, mewakili *state* pengalokasian proyek, yaitu tuple (p_1,p_2,p_3,…,p_n).
2. *Chance nodes*, mewakili aksi yang diambil oleh manajer a_i.
3. *End nodes*, mewakili nilai akhir P yang kita dapatkan setelah mengambil runtunan aksi tertentu.

Contoh sederhana dari *Decision Tree* yang dihasilkan adalah sebagai berikut:



Pada gambar tersebut, direpresentasikan bahwa seorang manajer memiliki dua jenis proyek dan masing-masing proyek memiliki nilai awal masing-masing 10K. Aksi a1 adalah mengalokasikan seluruh nilai awal proyek 1 untuk menjalankan proyek 1 untuk satu tahun ke depan. Nilai *Return of Investment* untuk aksi a1 adalah 10% dengan probabilitas 30%, 20% dengan probabilitas 60%, dan 30% dengan probabilitas 10%.

Apabila kita menggunakan pohon jenis ini, kita berarti menggunakan metode *stochastic dynamic programming*. Hal ini, memerlukan perhitungan yang rumit. Oleh karena itu, untuk menyederhanakannya, kita dapat memodelkannya menjadi berikut:



Proses penyederhanaan yang dilakukan adalah dengan rumus:

$$S_{ai}' = \sum_{j=0}^{k} (Prob_j \times S_{aij})$$

Keterangan :

S_{ai}' : state yang diakibatkan oleh aksi a1 setelah disederhanakan

S_{aij} : setiap state oleh aksi a1 dengan setiap ROI.

$Prob_j$: probabilitas untuk setiap ROI.

Dengan demikian, permasalahan *stochastic* ini dapat didekati dengan metode *deterministic dynamic programming*. Strategi yang digunakan adalah sebagai berikut:

1. Tahap (k) adalah proses memilih aksi untuk investasi proyek.
2. Status x = ($p_1, p_2, \ldots, p_n$) menyatakan tuple dana yang dialokasikan untuk sejumlah n proyek.
3. Himpunan alternatif (A) adalah himpunan aksi yang dipilih oleh manajer untuk menginvestasikan dana pada proyek tertentu.

Penyelesaian dilakukan dengan Program Dinamis Maju, misalkan:

R_{ijk} = nilai ekspektasi keuntungan yang didapatkan jika mengambil aksi a_i pada tahun k setelah sebelumnya mengambil aksi a_j

$f_k(a_i,S)$ = nilai ekspektasi keuntungan yang didapatkan jika mengambil aksi a_i pada tahun k setelah mengambil aksi pada himpunan S.

Dalam hal ini, perumusan R menggunakan konsep NPV adalah sebagai berikut:

$$R_{ijk} = \frac{selectproja_i(x_k) \times AccROI_k}{(1+r)^k}$$

$$x_k = a_j(x_{k-1})$$

Rumus ini menyatakan proyek terpilih oleh aksi a_i dikalikan dengan nilai akumulasi ROI pada tahun k untuk proyek tersebut dibagi *discounting factor*-nya. Perlu diperhatikan bahwa fungsi $a_j(x_{k-1})$ menyatakan fungsi akibat pemilihan aksi a_j pada tahun k-1 menghasilkan nilai x_k.

Relasi rekurens yang menyatakan keuntungan optimal dari persoalan ini adalah:

$f_1(a_i, \emptyset) = R_{i01}, 1 \leq i \leq n$ (basis)

$f_k(a_i, S) = \max_{j \in S}\left\{R_{ijk} + \frac{f_{k-1}(a_j, S-\{a_j\})}{(1+r)}\right\}$

$k = 2,3,\ldots,n$ (rekurens)

Penyelesaian persoalan ini diselesaikan dari tahap per tahap sesuai dengan definisi *dynamic programming*.

V. STUDI KASUS

Untuk menambah pemahaman terhadap metode ini, kita dapat membuat instansiasi permasalahan dengan melakukan studi kasus.

Misalkan seorang manajer mengelola sebuah perusahaan yang mengelola dua proyek. Tiap akhir tahun manajer tersebut dapat menginvestasikan dana pada proyek itu kembali atau pada proyek lain. Pada kasus ini, pada akhir tahun, manajer dapat memilih menginvestasikan dana proyek ke proyek itu sendiri, atau menyisakan dana sebesar 5K di proyek itu dan sisanya dialokasikan untuk proyek lain. Diketahui nilai tingkat pengembaliannya adalah r = 10%. Yang perlu dilakukan oleh manajer adalah menentukan investasi yang dilakukan selama 4 tahun untuk mendapatkan nilai ROI yang maksimal jika dana awalnya adalah 10K untuk proyek 1 dan 10K untuk proyek 2.

Catatan : setiap aksi hanya dapat diambil satu kali.

Nilai *Return of Investment* (ROI) untuk tiap proyek adalah sebagai berikut :

Proyek 1

Probabilistik	Thn 1	Thn 2	Thn 3	Thn 4
20%	30%	-10%	-20%	30%
60%	30%	10%	10%	-20%
20%	10%	0%	-40%	0%

Proyek 2

Probabilistik	Thn 1	Thn 2	Thn 3	Thn 4
10%	15%	0%	50%	30%
50%	40%	-30%	10%	20%
40%	10%	10%	-40%	0%

A. States

Kita mendefinisikan status dalam tuple (x,y) dalam hal ini, status awal permasalahan ini adalah (10K,10K).

B. Action

Pada persoalan ini, terdapat empat jenis aksi, yaitu :

a_1 : investasi seluruh dana proyek 1 untuk proyek 1

a_2 : investasi seluruh dana proyek 1 dan menyisakan dana proyek 2 sebesar 5K untuk selebihnya ke proyek 1

a_3 : investasi seluruh dana proyek 2 untuk proyek 2

a_4 : investasi seluruh dana proyek 2 dan menyisakan dana proyek 1 sebesar 5K untuk selebihnya ke proyek 1

Contohnya: (10K,10K) untuk aksi a_2 menjadi (15K,5K) sebelum penghitungan ROI. Dalam hal ini, jika aksi a1 telah diambil pada tahun sebelumnya, maka aksi a1 tidak dapat diambil pada tahun ini.

C. Accumulative ROI

Didapatkan dengan mengalikan setiap ROI dan Probabilitasnya, kemudian dijumlahkan.

Proyek	Thn 1	Thn 2	Thn 3	Thn 4
1	20%	16%	-6%	-6%
2	25.5%	-11%	-6%	13%

D. Penyelesaian

Tahap 1

x_1 = (10K, 10K)

$f_1(a_i, \emptyset) = R_{i01}$, $1 \leq i \leq n$

Diperoleh :

$f_1(a_1, \emptyset) = R_{101}$= 1,818K,
$x_2 = a_1$(10K,10K) = (12K, 10K).
$f_1(a_2, \emptyset) = R_{201}$= 2,727K
$x_2 = a_2$(10K,10K) = (18K, 5K).
$f_1(a_3, \emptyset) = R_{301}$= 2,318K
$x_2 = a_2$(10K,10K) = (10K, 12,55K).
$f_1(a_4, \emptyset) = R_{401}$= 3,077K
$x_2 = a_2$(10K,10K) = (5K, 18,385K).

Tahap 2

S	$f_2(a_i,S) = \max_{j \in S}\{R_{ij2} + \frac{f_1(a_j, S-\{a_j\})}{(1+r)}\}$			
	$a_i = a_1$	$a_i = a_2$	$a_i = a_3$	$a_i = a_4$
$\{a_1\}$	-	3,901K	0,744K	0,107K
$\{a_2\}$	4,063K	-	2,025K	0,843K
$\{a_3\}$	2,987K	4,428K	-	0,512K
$\{a_4\}$	3,237K	5,228K	1,126K	-

x_2	x_3			
	$a_i = a_1$	$a_i = a_2$	$a_i = a_3$	$a_i = a_4$
(12K, 10K)	-	(14,72K, 5K)	(12K, 8,9K)	(5K, 15,13K)
(18K, 5K)	(20,88K, 5K)	-	(18K, 4,45K)	(5K, 16,02K)
(10K, 12,55 K)	(11,6K, 12,55K)	(20,358K, 5K)	-	(5K, 15,620K)
(5K, 18,38 5K)	(5,8K, 18,385K)	(21,3266K, 5K)	(5K, 16,363K)	-

Tahap 3

Untuk $f_3(a_1,S)$ diperoleh

S	$f_3(a_1,S) = \max_{j\in S}\{R_{1j3}+\frac{f_2(a_j,S-\{a_j\})}{(1+r)}\}$			Solusi Optimal	
	a_j=a_2	a_j=a_3	a_j=a_4	$f_3(a_1,S)$	a_j
$\{a_2,a_3\}$	3,108 K	1,029 K	-	3,108 K	a_2
$\{a_2,a_4\}$	3,792 K	-	0,541 K	3,792 K	a_2
$\{a_3,a_4\}$	-	0,798 K	0,240 K	0, 798 K	a_3

Solusi Optimal
x_3
(19,137 K, 5 K)
(20,047 K, 5 K)
(4,7 K, 16,363 K)

Untuk $f_3(a_2,S)$ diperoleh

S	$f_3(a_2,S) = \max_{j\in S}\{R_{2j3}+\frac{f_2(a_j,S-\{a_j\})}{(1+r)}\}$			Solusi Optimal	
	a_j=a_1	a_j=a_3	a_j=a_4	$f_3(a_2,S)$	a_j
$\{a_1,a_3\}$	1,852 K	0,135 K	-	1,852 K	a_1
$\{a_1,a_4\}$	2,078 K	-	-0,585 K	2,078 K	a_1
$\{a_3,a_4\}$	-	0,286 K	-0,239 K	0,286 K	a_3

Solusi Optimal
x_3
(18,001 K, 5 K)
(18,034 K, 5 K)
(15,381 K, 5 K)

Untuk $f_3(a_3,S)$ diperoleh

S	$f_3(a_3,S) = \max_{j\in S}\{R_{3j3}+\frac{f_2(a_j,S-\{a_j\})}{(1+r)}\}$			Solusi Optimal	
	a_j=a_1	a_j=a_2	a_j=a_4	$f_3(a_3,S)$	a_j
$\{a_1,a_2\}$	3,468 K	3,321 K	-	3,468 K	a_1
$\{a_1,a_4\}$	2,114 K	-	-0,585 K	2,114 K	a_1
$\{a_2,a_4\}$	-	4,528 K	0,044 K	4,528 K	a_2

Solusi Optimal
x_3
(20,88 K, 4,7 K)
(5,8 K, 17,282 K)
(21,327 K, 4,7 K)

Untuk $f_3(a_4,S)$ diperoleh

S	$f_3(a_4,S) = \max_{j\in S}\{R_{4j3}+\frac{f_2(a_j,S-\{a_j\})}{(1+r)}\}$			Solusi Optimal	
	a_j=a_1	a_j=a_2	a_j=a_3	$f_3(a_4,S)$	a_j
$\{a_1,a_2\}$	2,752 K	2,657 K	-	2,752 K	a_1
$\{a_1,a_3\}$	1,852 K	-	-0,041 K	1,852 K	a_1
$\{a_2,a_3\}$	-	3,108 K	1,054 K	3,108 K	a_2

Solusi Optimal
x_3
(5 K, 19,627K)
(5 K, 18,001 K)
(5 K, 19,137 K)

Tahap 4

Untuk $f_4(a_1,S)$ diperoleh

S	$f_4(a_1,S) = \max_{j\in S}\{R_{1j4}+\frac{f_3(a_j,S-\{a_j\})}{(1+r)}\}$			Solusi Optimal	
	a_j=a_2	a_j=a_3	a_j=a_4	$f_4(a_1,S)$	a_j
$\{a_2,a_3,a_4\}$	-0,370 K	3,242 K	2,620 K	3,242 K	a_3

$x_4 = a_1$(21,327 K, 4,7 K) = (20,047K, 4,7 K)

Untuk $f_4(a_2,S)$ diperoleh

S	$f_4(a_2,S) = \max_{j\in S}\{R_{2j4}+\frac{f_3(a_j,S-\{a_j\})}{(1+r)}\}$			Solusi Optimal	
	a_j=a_1	a_j=a_3	a_j=a_4	$f_4(a_2,S)$	a_j
$\{a_1,a_3,a_4\}$	0,067 K	1,181 K	0,946 K	1,181 K	a_3

$x_4 = a_2$(5,8K,17,282 K) = (21,697 K, 5 K)

Untuk $f_4(a_3,S)$ diperoleh

S	$f_4(a_3,S) = \max_{j\in S}\{R_{3j4}+\frac{f_3(a_j,S-\{a_j\})}{(1+r)}\}$			Solusi Optimal	
	a_j=a_1	a_j=a_2	a_j=a_4	$f_4(a_3,S)$	a_j
$\{a_1,a_2,a_4\}$	3,891 K	2,333 K	4,245 K	4,245 K	a_4

$x_4 = a_3$(5K,19,627 K) = (5 K, 22,179 K)

Untuk $f_4(a_4,S)$ diperoleh

S	$f_4(a_4,S) = \max_{j \in S}\{R_{4j4}+\frac{f_3(a_j,S-\{a_j\})}{(1+r)}\}$			Solusi Optimal	
	$a_j=a_1$	$a_j=a_2$	$a_j=a_3$	$f_4(a_4,S)$	a_j
$\{a_1,a_2,a_3\}$	4,524 K	3,282 K	4,980 K	4,980 K	a_3

$x_4 = a_4(20{,}88K, 4{,}7\ K) = (5\ K, 23{,}255\ K)$

Setelah tahap 4 kita dapat menyimpulkan bahwa nilai NPV optimal untuk persoalan investasi tersebut adalah 4,980 K dengan urutan aksi $a_2 \Rightarrow a_1 \Rightarrow a_3 \Rightarrow a_4$. FutureValue untuk masing-masing proyek diperkirakan pada akhir Tahun ke empat untuk urutan aksi tersebut adalah 5K untuk proyek 1 dan 23,255 K untuk proyek 2.

VI. KESIMPULAN

Berdasarkan penjelasan mengenai analisis permasalahan, metodologi pemecahan masalah, dan studi kasus pada bab-bab sebelumnya, dapat disimpulkan beberapa hal:

1. Persoalan investasi proyek dapat diselesaikan dengan menggunakan *dynamic programming* dan menghasilkan solusi optimal.
2. Penyederhanaan permasalahan stochastic menjadi permasalahan deterministik memudahkan kita untuk menyelesaikan persoalan tersebut.
3. Metode *dynamic programming* untuk persoalan investasi ini belum menjadi algoritma yang mangkus karena secara tidak langsung kita mendaftarkan seluruh *state* yang mungkin, tetapi dilakukan secara terstruktur tahap demi tahap. Hal ini setidaknya memudahkan kita untuk mengintegrasikannya dengan konsep *Time Value of Money*.
4. Penyelesaian persoalan investasi dengan metode *dynamic programming* pada dasarnya mirip dengan penyelesaian TSP dengan *dynamic programming*.

VII. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa karena berkat rahmat-Nya makalah ini dapat terselesaikan dengan baik. Tidak lupa, penulis juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. selaku dosen pengajar kuliah IF3051 Strategi Algoritma karena ilmu yang diberikannya sangat bermanfaat dan menjadi dasar pembuatan makalah ini.

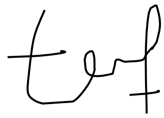
REFERENCES

[1] http://en.wikipedia.org/wiki/Time_value_of_money [8 December 2011]
[2] http://en.wikipedia.org/wiki/Decision_tree [8 December 2011]
[3] http://en.wikipedia.org/wiki/Decision_analysis [8 December 2011]
[4] http://en.wikipedia.org/wiki/Dynamic_programming [8 December 2011]
[5] Alam-Doucette, "Stochastic Investment Decision Making with Dynamic Programming"Dhaka, Bangladesh, 2010
[6] Rinaldi Munir, "Diktat Kuliah Strategi Algoritmik", Program Studi Teknik Informatika ITB, 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, 9 November 2011



Faisal Ibrahim Hadiputra/ 13509048