Pengambilan Keputusan Bisnis dengan Stochastic Dynamic Programming

Hafizh Budiman 13516137
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516137@std.stei.itb.ac.id

***Abstract*—Pengambilan keputusan yang tepat adalah kunci kesuksesan untuk setiap investor dalam usaha untuk mengikuti arus dari lingkungan bisnis yang kompetitif. Mitigasi dari paparan resiko memainkan peran yang vital, karena sekarang investor secara langsung terkena lingkungan keputusan yang tidak pasti. Ketidakpastian dan resiko dari suatu investasi akan meningkat seiring dengan peningkatan jumlah investor pesaing yang masuk ke pasar. Akibatnya, *return* dari investasi pada bisnis juga membawa ketidakpastian yang tinggi juga. Di makalah ini akan dirumuskan model pemrograman dinamis untuk merumuskan keputusan investasi bisnis dengan menggunakan cara probabilistik, dan mengadopsi kebijakan policy dari dynamic programming untuk menyelesaikan model ini. Hasil dari simulasi ini menunjukkan bahwa algoritma ini dapat membantu dalam mengambil keputusan investasi bisnis yang optimal.**

***Keywords*—keputusan bisnis, stochastic, dynamic programming.**

I. PENDAHULUAN

Prinsip pemrograman dinamis, pertama kali dikenal setelah Perang Dunia Kedua. Profesor Richard Bellman dianggap sebagai pencipta metode ini. Pada tahun 1957, dalam bukunya ‘Pemrograman Dinamis’ (Princeton Universitas Press, Princeton), Bellman menetapkan prinsip-prinsip dasar dari metode ini. Apa yang beliau lakukan adalah sebagai berikut: beliau melihat masalah tertentu dengan membuat hierarki sub-masalah dan kemudian melanjutkan untuk menyelesaikan yang paling mudah. Begitulah cara prinsip tentang optimalitas muncul, yang merupakan esensi dari dynamic programming. Gagasan utama dalam aplikasi metode ini adalah pembagian sebuah proses manajemen menjadi beberapa tahap dan dipilih model manajemen mana yang paling optimal untuk setiap tahap.

Model manajemen yang dipilih adalah model yang mengarah ke fungsi proses yang optimal. Pada saat yang sama kita harus ingat bahwa manajemen optimal dicirikan oleh hal-hal berikut: keputusan masa depan harus mengarah ke manajemen yang optimal dengan mempertimbangkan keadaan saat ini dan terlepas dari state sebelumnya atau manajemen sebelumnya. Saat ini, pengambilan keputusan investasi adalah tugas penting karena setiap investasi menunjukkan setidaknya beberapa jumlah risiko dan ketidakpastian. Risiko dan ketidakpastian ini adalah hasil dari persaingan bisnis yang sangat besar dan pasar ekonomi yang dinamis. Akibatnya, penelitian terbaru dalam pengambilan keputusan investasi mengalami pergeseran paradigma dengan banyak integrasi teknik baru dengan metode yang ada untuk mengembangkan proses pengambilan keputusan yang kuat.

Net present value (NPV) adalah metode paling umum dalam evaluasi investasi. Alkaraan dan Northcott telah menganalisis penggunaan teknik penilaian investasi konvensional seperti pengembalian aset (ROA), laba atas investasi (ROI), tingkat pengembalian internal (IRR), NPV dan pendekatan analisis risiko seperti analisis sensitivitas, penyesuaian periode pengembalian, atau tingkat diskonto. Penanganan risiko dan ketidakpastian dalam proyek saat ini adalah salah satu topik utama yang menarik bagi para peneliti dan praktisi yang bekerja di bidang proyek pengambilan keputusan investasi. Topaloglou dan Rockafellar memecahkan masalah optimasi portofolio dengan program linier dan mereka menggunakan simulasi Monte Carlo untuk menangkap risiko setiap investasi terkait.

Sebagian besar penelitian sebelumnya tidak mempertimbangkan dinamika yang saling terkait dari sistem yang dapat dihadapi oleh model investasi dinamis stochastic. Penelitian ini bertujuan untuk menangani hal ketidakpastian dan dinamika yang saling terkait menggunakan pemrograman dinamis stochastic tak terbatas dan pada saat yang sama membuat keputusan investasi optimal yang dapat menerima total *return* maksimal yang diharapkan..

II. DASAR TEORI

A. Program Dinamis

Program Dinamis (*dynamic programming*) adalah metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. Pada penyelesaian persoalan dengan metode ini:

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

Pada program dinamis, rangkaian keputusan yang optimal dibuat dengan menggunakan Prinsip Optimalitas. Prinsip ini berbunyi : *jika solusi total optimal, maka bagian solusi sampai tahap ke-k juga optimal*. Prinsip optimalitas berarti bahwa jika kita bekerja dari tahap k ke tahap $k + 1$, kita dapat menggunakan hasil optimal dari tahap k tanpa harus kembali ke tahap awal. Jika pada setiap tahap kita menghitung ongkos (*cost*), maka dapat dirumuskan bahwa ongkos pada tahap $k + 1$

= (ongkos yang dihasilkan pada

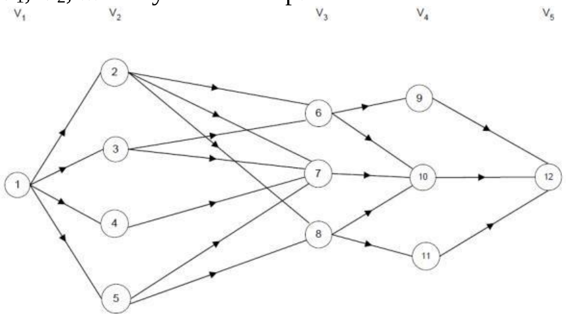
tahap k) + (ongkos dari tahap k ke tahap $k + 1$)

Dengan prinsip optimalitas ini dijamin bahwa pengambilan keputusan pada suatu tahap adalah keputusan yang benar untuk tahap-tahap selanjutnya.

Program dinamis diterapkan pada persoalan yang memiliki karakteristik sebagai berikut:

1. Persoalan dapat dibagi menjadi beberapa tahap (*stage*), yang pada setiap tahap hanya diambil satu keputusan.
2. Masing-masing tahap terdiri dari sejumlah status (*state*) yang berhubungan dengan tahap tersebut. Secara umum, status merupakan bermacam kemungkinan masukan yang ada pada tahap tersebut. Jumlah status bisa berhingga (*finite*) atau tidak berhingga (*infinite*).

Gambar 2 memperlihatkan perbedaan antara tahap dan status diberikan pada graf multitahap (*multistage graph*). Tiap simpul di dalam graf tersebut menyatakan status, sedangkan V_1, V_2, ... menyatakan tahap.



Gambar 1: Graf yang menyatakan tahap (stage) dan status (state)

3. Hasil dari keputusan yang diambil pada setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. Ongkos (*cost*) pada suatu tahap meningkat secara teratur (*steadily*) dengan bertambahnya jumlah tahapan.
5. Ongkos pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah berjalan dan ongkos pada tahap tersebut.
6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya.
7. Adanya hubungan rekursif yang mengidentifikasikan keputusan terbaik untuk setiap status pada tahap k memberikan keputusan terbaik untuk setiap status pada tahap $k + 1$.
8. Prinsip optimalitas berlaku pada persoalan tersebut.

Dalam menyelesaikan persoalan dengan program dinamis, orang dapat menggunakan dua ancangan (*approach*) berbeda: maju (*forward* atau *up-down*) dan mundur (*backward* atau *bottom-up*). Misalkan x_1, x_2, ..., x_n menyatakan peubah (*variable*) keputusan yang harus dibuat masing-masing untuk tahap 1, 2, ..., n. Maka,

a. Program dinamis maju. Program dinamis bergerak mulai dari tahap 1, terus maju ke tahap 2, 3, dan seterusnya sampai tahap n. Runtunan peubah keputusan adalah x_1, x_2, ..., x_n.
   Prinsip optimalitas pada program dinamis maju:
   ongkos pada tahap $k + 1$ = (ongkos yang dihasilkan pada tahap ke k) + (ongkos dari tahap k ke tahap $k + 1$) dengan $k = 1, 2, ..., n - 1$.
b. Program dinamis mundur. Program dinamis bergerak mulai dari tahap n, terus mundur ke tahap $n - 1$, $n - 2$, dan seterusnya sampai tahap 1. Runtunan peubah keputusan adalah x_n, x_{n-1}, ..., x_1.
   Prinsip optimalitas pada program dinamis mundur:
   ongkos pada tahap k = (ongkos yang dihasilkan pada tahap $k + 1$) + (ongkos dari tahap $k + 1$ ke tahap k) dengan $k = n, n - 1, ..., 1$.

Penyelesaian secara maju atau mundur keduanya ekivalen dan menghasilkan solusi optimum yang sama. Pengalaman menunjukkan bahwa penyelesaian dengan program dinamis mundur umunya lebih mangkus. Kebanyakan literatur perihal program dinamis menyajikan penyelesaian persoalan dengan ancangan mundur. Karena itu, sebagian contoh-contoh yang diberikan di sini akan diselesaikan dengan program dinamis mundur, disamping itu diberikan pula penyelesaian secara maju.

Secara umum, ada empat langkah yang dilakukan dalam mengembangkan algoritma program dinamis:

1. Karakteristikkan struktur solusi optimal.
2. Definisikan secara rekursif nilai solusi optimal.
3. Hitung nilai solusi optimal secara maju atau mundur.
4. Konstruksi solusi optimal.

Perlu dicatat bahwa solusi optimal yang dihasilkan oleh program dinamis dapat lebih dari satu buah.

B. Stochastic Dynamic Programming

Pemrograman dinamis adalah alat sistematis berdasarkan ide sederhana dari prinsip optimalitas. Jika masalah keputusan dapat dilihat sebagai beberapa tahap dengan beberapa state dan transisi state yang dikenal terkait dengan setiap tindakan tertentu, maka masalahnya dapat diselesaikan secara sistematis dengan deterministic dynamic programming dan dengan bantuan persamaan Bellman. Tetapi jika transisi state bersifat probabilistik, maka kita dapat menerapkan pemrograman dinamis stokastik. Terlepas dari ini, jika jumlah tahapannya tidak terbatas (yaitu, jika kita tidak ingin memaksakan batasan pada jumlah tahap), maka masalahnya menjadi tak terbatas untuk masalah pemrograman dinamis stokastik.

Dalam pengambilan keputusan investasi ini, akan dipertimbangkan waktu sebagai berbagai tahap dan jumlah akumulasi dana sebagai state. Faktor kunci untuk transisi state adalah ROI dan ada ketidakpastian yang terkait dengannya yang dapat digambarkan dengan distribusi probabilitas, yang membentuk probabilitas *state transition*. Seperti yang di rencanakan untuk membuat keputusan tentang berapa banyak yang harus diinvestasikan kembali dalam setiap proyek setiap tahun sehingga memaksimalkan yang diharapkan mengumpulkan dana selama waktu yang tak terbatas, kita harus menggunakan IHSDP untuk membuat keputusan berdasarkan jangka panjang manfaat potensial dan menghindari *blind reaction* terhadap fluktuasi pasar jangka pendek yang dapat membawa kita kepada pembuatan keputusan yang berisiko.

Di antara berbagai pendekatan untuk IHSDP, algoritma yang paling banyak digunakan adalah *policy iteration*, nilai iterasi dan pemrograman linier. Dalam tulisan ini kita akan menggunakan teknik iterasi kebijakan untuk menyelesaikan investasi masalah keputusan. Notasi berikut akan digunakan untuk menggambarkan algoritma ini:

S = himpunan semua states

V (s) i = nilai dari fungsi nilai untuk state dan iterasi i

R (s, a) = reward pada state dan tindakan α

p [y | s, a] = probabilitas transisi dari state ke y, jika kita mengambil tindakan α

α = faktor diskon

ε = parameter toleransi

μ^i = kebijakan stasioner pada iterasi ke i

C. *Policy Iteration*

Algoritma iterasi kebijakan menghitung kebijakan optimal secara langsung tanpa bantuan fungsi nilai yang optimal. Iterasi kebijakan menyatu lebih cepat dalam hal jumlah iterasi (i), tetapi setiap iterasi akan dikomputasi secara boros jika terdapat terlalu banyak states. Ini disebabkan proses inversi matriks untuk ukuran besar matriks probabilitas transisional. Langkah-langkah berikut ini terlibat dalam Algoritma Iterasi Kebijakan:

Langkah 1: Mulai dengan tindakan atau kebijakan stasioner apa pun yang dimiliki. Kebijakan-kebijakan ini hanya khusus untuk beberapa state dan state yang tergantung pada state tersebut.

Langkah 2: Untuk suatu aksi random, lakukan evaluasi kebijakan. Dan menghitung perkiraan jumlah reward yang diharapkan. Dengan mengikuti evaluasi kebijakan, persamaan akan memeriksa, apakah kebijakan saat ini optimal atau tidak.

$$V^i(s) = R(s, \mu^i(s)) + \alpha \sum_{y \in S} p[y|s,a] V^i(y)$$

Langkah 3: Jika pada langkah ke-2 kebijakan ditemukan tidak optimal, maka lakukan perbaikan kebijakan untuk yang optimal atau tindakan dengan operasi lanjutan.

Langkah 4: Untuk memeriksa optimalitas tindakan optimal, periksa kondisi berikut untuk melihat peningkatannya fungsi nilai:

$$\left\| V^i(s) - V^{i-1}(s) \right\|_\infty < \frac{1-\alpha}{\alpha}\varepsilon$$

Jika kondisi di atas terpenuhi maka berhentilah, jika tidak, kembali ke langkah 2.

III. IMPLEMENTASI DAN EKSPERIMEN

1. Kasus 1

Diketahui 2 tabel sebagai berikut:

Kuantitas	T1	T2	T3
Medium	48.16	37.14	30.42
Besar	40.00	29.52	23.18
Kecil	64.62	52.37	44.91

Tabel 1. Persentase Penjualan

Investasi	T1	T2	T3
20000	9632	7428	6084
40000	16000	11800	9272
60000	38772	31422	26946
80000	41600	43536	39040

Tabel 2. Ekspektasi Keuntungan

Sebuah Perusahaan memiliki jumlah uang sebesar EUR 80000 dan akan menginvestasikan uang ini dalam produksi tiga transformer dengan tujuan menghasilkan keuntungan maksimal. Untuk memecahkan masalah investasi ini, model prinsip Bellman akan digunakan. Maka bentuk persamaan matematikanya sebagai berikut:

$$\max F(x_1, x_2, x_3) = f_1(x) + f_2(x_2) + f_3(x_3)$$

Dengan batasan:

$$x_1 + x_2 + x_3 = 80000\ ,\ x_1, x_2, x_3 \geq 0\,.$$

Seperti telah disebutkan, masalahnya adalah diselesaikan secara bertahap tetapi hasil dari satu tahap tergantung pada hasil yang diperoleh dalam tahap yang sebelumnya. Fungsi Fk (bk) ditentukan berturut-turut untuk k = 1,2,3. Berikut ini juga harus diperhitungkan:

$$b_k \in \{0,20000,40000,60000,80000\},\ \text{for}\ k = 1,2,3\,.$$

Nilai dari Fi(bi) yang akan didapatkan pada tahap pertama dengan cara berikut:

$$F_1(b_1) = \max f_1(x_1),$$

Lebih jauh dijabarkan dengan fungsi:

$$F_1(0) = 0, F_1(20000) = 9632, F_1(40000) = 16000,$$
$$F_1(60000) = 38772, F_1(80000) = 41600$$

Tahap berikutnya menentukan alokasi optimal dari investasi untuk produk T1 dan T2 maka fungsi objektif F2(b2) akan memiliki bentuk sebagai berikut:

$$F_2(b_2) = \max_{x_2 \le b_2} \{ f_2(b_2) + F_1(b_2 - x_2) \}$$

Dengan menggunakan nilai yang didapat untuk fungsi F1(b1) dan data efisiensi investasi dalam produk T2, dan menggunakan formula yang ada diatas, maka nilai dari fungsi dapat ditentukan sebagai berikut:

Untuk b2 = 0, maka

$$F_2(0) = \max_{x_2 \le 0} \{ f_2(0) + f_1(0 - x_2) \} = 0,\ x_2 = 0$$

Sedangkan untuk b2 = 20000

$$F_2(20000) = \max_{x_2 \le 20000}\{f_2(x_2) + F_1(20000 - x_2)\}$$

$$= \max\begin{Bmatrix} f_2(0)+F_1(20000) \\ f_2(20000)+F_1(0)\end{Bmatrix} = \max\begin{Bmatrix}0+9632\\7428+0\end{Bmatrix} = 9632 \quad (x_2 = 0)$$

Dan untuk b2 = 40000

For $b_2 = 40000$, we have the following:

$$F_2(400000) = \max_{x_2\le 40000}\{f_2(x_2)+F_1(40000-x_2)\} = \max\begin{Bmatrix} f_2(0)+F_1(40000)\\ f_2(20000)+F_1(20000)\\ f_2(40000)+F_1(0)\end{Bmatrix}$$

$$=\max\begin{Bmatrix}0+16000\\7428+9632\\11808+0\end{Bmatrix} = \max\begin{Bmatrix}16000\\17060\\11808\end{Bmatrix} = 17060 \text{ whereby } x_1=x_2=20000.$$

Dan untuk b2 = 60000

$$F_2(60000) = \max_{x_2\le 60000}\{f_2(x_2)+F_1(60000-x_2)\}$$

$$=\max\begin{Bmatrix}0+38772\\7428+16000\\11808+9632\\31422+0\end{Bmatrix} = \max\begin{Bmatrix}38772\\23428\\21440\\31422\end{Bmatrix} = 38772$$

Dimana x2 = 0.

Pada akhir tahap 1 untuk b2 = 80000 akan didapatkan

$$F_2(80000) = \max_{x_2\le 80000}\{f_2(x_2)+F_1(80000-x_2)\}$$

$$=\max\begin{Bmatrix} f_2(0)+F_1(80000)\\ f_2(20000)+F_1(60000)\\ f_2(40000)+F_1(40000)\\ f_2(60000)+F_1(20000)\\ f_2(80000)+F_1(0)\end{Bmatrix} = \max\begin{Bmatrix} f_2(0)+F_1(80000)\\ f_2(20000)+F_1(60000)\\ f_2(40000)+F_1(40000)\\ f_2(60000)+F_1(20000)\\ f_2(80000)+F_1(0)\end{Bmatrix}$$

$$=\max\begin{Bmatrix}0+41600\\7428+38772\\11808+16000\\31422+9632\\43536+0\end{Bmatrix} = \max\begin{Bmatrix}41600\\46200\\27808\\41054\\43536\end{Bmatrix} = 46200$$

Dengan x1 = 60000 dan x2 = 20000.

Pada basis dari hasil yang didapat, dapat disimpulkan bahwa diperlukan investasi sebesar 60000 pada T1, sedangkan uang yang dibutuhkan untuk produksi T2 adalah 20000. Kemudian pada tahap 3, alokasi dari dana diantara 3 produk akan seperti berikut:

$$F_3(b_3) = \max_{x_3\le b_3}\{f_3(x_3)+F_2(b_3-x_3)\} \quad (8)$$

Jika b3 = 2000 maka

$$F_3(20000) = \max_{x_3\le 20000}\{f_3(x_3)+F_2(20000-x_3)\}$$

$$= \max\begin{Bmatrix} f_3(0)+F_2(20000) \\ f_3(20000)+F_2(0)\end{Bmatrix} = \max\begin{Bmatrix}0+9632\\6084+0\end{Bmatrix} = 9632 \quad (x_3 = 0)$$

Jika nilai parameter ib3 = 40000

$$F_3(40000) = \max_{x_3\le 40000}\{f_3(x_3)+F_2(40000-x_3)\}$$

$$=\max\begin{Bmatrix} f_3(0)+F_2(40000)\\ f_3(20000)+F_2(20000)\\ f_3(40000)+F_2(0)\end{Bmatrix} = \begin{Bmatrix}0+17060\\6084+9632\\9272+0\end{Bmatrix} = \max\begin{Bmatrix}17060\\15716\\9272\end{Bmatrix} = 17060$$

Sedangkan jika b3 = 60000 maka

if $b_3 = 60000$, the following is true:

$$F_3(60000) = \max_{x_3\le 60000}\{f_3(x_3)+F_2(60000-x_3)\} = \max\begin{Bmatrix} f_3(0)+F_2(60000)\\ f_3(20000)+F_2(40000)\\ f_3(40000)+F_2(20000)\\ f_3(60000)+F_2(0)\end{Bmatrix}$$

$$=\max\begin{Bmatrix}0+38772\\6084+17060\\9272+9632\\26946+0\end{Bmatrix} = \max\begin{Bmatrix}38772\\23144\\18904\\26946\end{Bmatrix} = 38772 \text{ whereby } x_3 = 0$$

Dan jika uang yang diinvestasikan sejumlah 80000, akan didapatkan nilai F3 sebagai berikut:

$$F_3(80000) = \max_{x_3\le 80000}\{f_3(x_3)+F_2(80000-x_3)\}$$

$$=\max\begin{Bmatrix} f_3(0)+F_2(80000)\\ f_3(20000)+F_2(60000)\\ f_3(40000)+F_2(40000)\\ f_3(60000)+F_2(20000)\\ f_3(80000)+F_2(0)\end{Bmatrix} = \begin{Bmatrix}0+46200\\6084+38772\\9272+17060\\26946+9632\\39040+0\end{Bmatrix} = \begin{Bmatrix}46200\\44856\\26332\\36578\\39040\end{Bmatrix} = 46200$$

Kita telah mendapatkan nilai untuk x3 = 0, x2 = 20000, dan x1 = 60000, yang hasilnya uang 80000 EUR tersebut akan diinvestasikan dengan cara berikut: 60000 pada produksi 1, 20000 pada produksi 3, dan tidak ada uang yang diinvestasikan pada produksi T3.dengan cara ini perusahaan akan mendapat profit maksimal sejumlah 46200. Hasilnya disajikan pada table dibawah ini:

Investasi	X1	F1(x1)	X2	F2(x2)	X3	F3(x3)
0	0	0	0	0	0	0
20000	20000	9632	0	9632	0	9632
40000	40000	16000	20000	17060	0	17060
60000	60000	38772	0	35772	0	38772
80000	80000	41600	20000	46200	0	46200

Tabel 3. Hasil dari analisis dengan pemrograman dinamis.

2. Kasus 2

Kami mengembangkan studi kasus dengan mengasumsikan nilai uji kasus yang wajar untuk semua variabel menggunakan algoritma iterasi kebijakan untuk pengambilan keputusan investasi: Sebuah perusahaan memiliki dua proyek yang sedang berjalan. Pada setiap akhir tahun, setiap proyek memiliki sejumlah dana yang tersedia untuk diinvestasikan kembali ke proyeknya sendiri atau proyek lainnya. Dalam masalah ini jumlah dana yang tersedia untuk setiap proyek mewakili keadaan masalah.

Untuk penyederhanaan, anggap saja perusahaan hanya memiliki kemampuan untuk menginvestasikan kembali dalam sebuah proyek tunggal pada suatu waktu. Investasi ulang dapat dilakukan untuk proyek tertentu dengan memanfaatkan seluruh dana yang tersedia proyek itu atau dengan memanfaatkan seluruh dana yang tersedia dari proyek itu ditambah beberapa bagian dari proyek lain. Karena, diasumsikan bahwa jika tidak ada reinvestasi dalam sebuah proyek, maka setidaknya modal kerja $ 10.000 (atau $ 10K) diperlukan untuk proyek itu untuk menjalankan bisnisnya sendiri yang sedang berjalan. Juga diasumsikan bahwa tidak ada kemungkinan untuk mengubah tingkat dana yang tersedia jika tidak ada reinvestasi dalam suatu proyek.

A. States / Keadaan

Jumlah dana yang tersedia untuk proyek-1 dan proyek-2 mewakili keadaan / state dari masalah. Diasumsikan bahwa untuk setiap proyek memiliki jumlah dana yang tersedia, bisa 10K, 11K, ...50K. Pada titik tertentu, misalnya, *state* bisa berupa (13K, 20K), yang berarti bahwa dana yang tersedia dalam proyek-1 adalah 13K dan dalam proyek-2 adalah 20K.

B. Tindakan

Menurut uraian masalah di atas ada empat jenis tindakan reinvestasi yang dapat kita pertimbangkan untuk dua proyek ini. Misalnya, tindakan-1 adalah menginvestasikan kembali jumlah total dana yang tersedia dalam proyek-1 ke proyek-1 saja, yaitu, jika *state* (13K, 22K) kemudian menginvestasikan kembali seluruh 13K ke proyek-1 saja. Tindakan-2 adalah menginvestasikan kembali jumlah total yang tersedia dana dalam proyek-1 dan kelebihan jumlah di atas modal kerja (diasumsikan sebelumnya 10K) dalam proyek-2 ke proyek-1 saja yaitu, jika state (13K, 22K) kemudian menginvestasikan 13K dari proyek-1 dan (22K-10K) = 12k dari proyek-2 ke proyek-1. Tindakan-3 dan tindakan-4 didefinisikan sebagai tindakan lain yang sesuai.

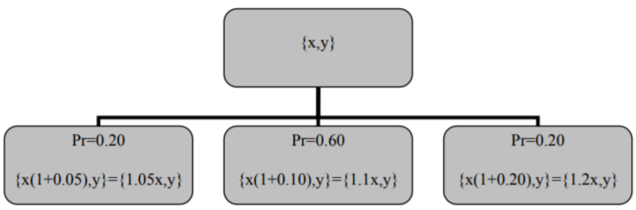
C. Transisi Keadaan / State

Seperti disebutkan sebelumnya ada berbagai tingkat ketidakpastian untuk ROI untuk setiap proyek individu. Untuk kasus studi ini diasumsikan bahwa untuk proyek-1, ada probabilitas 20% bahwa ROI adalah 5%, probabilitas 60% bahwa ROI adalah 10% dan 20% kemungkinan bahwa ROI adalah 20%. Di sisi lain, untuk proyek-2, ada Probabilitas 20% bahwa ROI 2,5%, Probabilitas 50% bahwa ROI adalah 10% dan 30% Probabilitas bahwa ROI 15%.

Anggaplah keadaan proyek saat ini adalah (X, Y), yang berarti dana yang tersedia dalam proyek-1 adalah X dan proyek-2 adalah Y. Untuk dapat memahami lebih lanjut, transisi keadaan yang terkait dengan tindakan-1 untuk probabilitas yang berbeda (Pr) scenario yang ditunjukkan oleh persamaan berikut:

$X_{t+1}=X_t(1+i/100)$
$Y_{t+1}=Y_t$

dengan *i* sebagai ROI, *t* sebagai tahun.



Gambar 2: Diagram transisi keadaan untuk tindakan-1

D. Perhitungan Keuntungan

Kita dapat memperoleh untung untuk setiap keputusan reinvestasi terkait tergantung pada state dan tindakan yang kita ambil. Persamaan berikut digunakan untuk menghitung *gain* yang diharapkan.

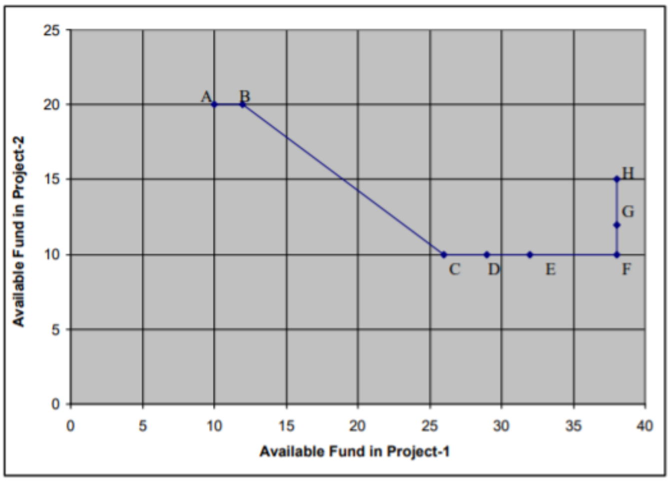
Gain yang diharapkan = ∑(Peluang ROI x ROI x Jumlah yang diinvestasikan ulang)

E. Sasaran

Tujuan yang diinginkan adalah untuk memilih tindakan yang optimal (yaitu, keputusan reinvestasi) di antara keempat opsi yang ada untuk setiap *state* dalam memaksimalkan perolehan atau *gain* total.

F. Analisa Hasil

Dalam pengujian menggunakan MATLAB, didapatkan Matriks transisi probabilitas dan matriks reward untuk setiap tindakan terkait yang dikembangkan sesuai dengan deskripsi masalah dan aturan transisi. Dengan matriks probabilitas transisi dan matriks reward, masalah diselesaikan untuk faktor diskon $\alpha = 0{,}9$ dan parameter toleransi $\varepsilon = 0{,}000001$, menggunakan *Policy Iteration Algorithm*. Untuk membantu menentukan apakah kebijakan optimal berfungsi dengan baik atau tidak, kita bisa mulai dari *state* dengan ROI yang dipilih secara acak dan bergerak dengan kebijakan optimal yang telah dipilih sebelumnya. Gambar berikut menunjukkan sebuah contoh transisi keadaan dengan aksi optimal.



Gambar 2. Realisasi sampel di dalam MATLAB untuk tindakan optimum (state-A dimulai dari 10K, 10K)

Pada Gambar 2, sumbu horizontal mewakili jumlah total dana yang tersedia dalam proyek-1 dan sumbu vertikal mewakili jumlah total dana yang tersedia dalam proyek-2. A, B, C, D, E, F, G, H mewakili pernyataan yang berbeda. Di antaranya A dan H adalah state awa; dan state akhir, dan sisanya adalah state pertengahan. Untuk menunjukkan efek dari tindakan optimal kita mulai dari state-A dengan tindakan optimum yang bergantung pada state dan mencapai ke state-B. Sekali lagi di state-B, telah diambil tindakan optimal dan mencapai ke state-C, sampai mencapai state terakhir yakni state-H. Jelas terlihat bahwa setiap kali kita pindah ke state yang lebih baik (yaitu, jumlah total dana yang tersedia di dua proyek lebih tinggi dari tahun sebelumnya), bahwa kebijakan optimal adalah optimal untuk semua state.

Tujuannya adalah untuk memaksimalkan *reward* dan *gain* total, yang berbanding lurus dengan nilai variabel state. Sebagai tambahannya, dapat dilihat dari gambaran di atas bahwa kebijakan optimal kita berbeda untuk bagian state yang berbeda saat kita bergerak maju; ini disebabkan dalam penerapan luring kami (yaitu, memilih kebijakan optimal dengan *policy iteration algorithm*), Kebijakan optimal dipilih mengingat semua ketidakpastian yang terkait dengan ROI dan transisi state.

V. KESIMPULAN

Dalam metode tradisional, kita dapat memilih proyek hanya dengan ROI yang diharapkan lebih tinggi, tetapi hal ini sering kali malah membawa kita ke keputusan suboptimal karena pengembalian investasi yang diharapkan (ROI) dari suatu keputusan sering membawa ketidakpastian yang tinggi pula dengan dinamika yang saling terkait. Makalah menyelidiki usaha untuk menangani dinamika ketidakpastian dan interelasi ini menggunakan pemrograman dinamis stochastic dan pada saat yang sama membuat keputusan investasi optimal untuk mencapai maksimum *reward* yang diharapka. Untuk mengingat hal ini, kami telah memformulasikan keputusan investasi masalah dalam kerangka kerja pemrograman dinamis stokastik dan menggunakan teknik iterasi kebijakan untuk mendapatkan keputusan investasi yang optimal.

Untuk menguji kelayakan dari metode yang kami usulkan kami telah menjalankan simulasi dengan studi kasus oleh dengan asumsi skenario kasus yang masuk akal. Hasil simulasi kami menunjukkan bahwa algoritma ini dapat membantu kami dalam mengambil keputusan investasi optimal. Dalam metode ini, kita dapat berpindah dari satu state ke state lain melalui serangkaian kebijakan state optimal yang diperoleh dari algoritma iterasi kebijakan, yang akan mengarah ke peningkatan status pada setiap langkah. Metode analisis keputusan investasi ini membuat keputusan investasi dengan memperhatikan ketidakpastian dalam ROI dan dinamika transisi state secara bersamaan. Oleh karena itu, kami melihat bahwa kebijakan optimal kami berbeda untuk state yang berbeda saat kami bergerak dan dijalankan. Penelitian lebih lanjut dapat dilakukan untuk memeriksa efisiensi teknik iterasi kebijakan dalam statistic run time dan jumlah iterasi yang diperlukan dengan algoritma pemrograman dinamis stokastik lainnya seperti nilai iterasi dan pemrograman linier.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bu Dr. Nur Ulfa Maulidevi, S.T., M.Sc., Bu Dr. Masayu Leylia Khodra, S.T., M.T., dan Bapak Dr. Ir. Rinaldi Munir, M.T. sebagai dosen mata kuliah Strategi Algoritma. Penulis juga mengucapkan terima kasih kepada seluruh penulis yang tulisannya dijadikan referensi untuk makalah ini. Terakhir, penulis berterima kasih kepada orang tua dan teman-teman atas dukungannya dalam menyelesaikan makalah ini.

REFERENCES

[1] Munir, Rinaldi. 2009. Diktat Kuliah Strategi Algoritma. Bandung: Program Studi Teknik Informatika Institut Teknologi Bandung.
[2] Ma, Shiqian. 2013. Handout 8: Introduction to Stochastic Dynamic Programing. SEEM 3470 Second Term.
[3] Xu-song, X., and Jian-mou, W., (2002), "A Dynamic Programming Algorithm on Project-Gang Investment Decision-Making", Wuhan University Journal of Natural Sciences vol. 7, no. 4, pp. 103-107
[4] Bertsekas, D. P., (2007), "Dynamic Programming and optimal Control", Third Edition, Volume II., Athena Scientific, Belmont, Massachusetts

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, 13 Mei 2017

Hafizh Budiman
13516137