

Penggunaan *Dynamic Programming* dalam Transaksi Saham

Jonathan Benedict / 13513003
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia
13513003@std.stei.itb.ac.id

Abstract—*Dynamic Programming* dikenal sebagai salah satu algoritma yang dapat dengan fleksibel diterapkan sebagai metode pemecahan untuk berbagai masalah. Hal ini dikarenakan prinsip utama dari *Dynamic Programming* adalah memecah masalah menjadi beberapa submasalah yang akhirnya dapat diselesaikan satu per satu oleh program. Oleh karena itu, penggunaan *Dynamic Programming* dapat juga diterapkan di dalam dunia transaksi saham dimana setiap pemegang saham mencari cara terbaik untuk mendapatkan keuntungan yang sebesar-besarnya. Untuk masalah ini, *Dynamic Programming* akan memberikan bantuan kepada pemegang saham mengenai kapan saat terbaik untuk membeli dan menjual sahamnya.

Index Terms—*Dynamic Programming*, Saham, Selisih Indeks Harga, Keuntungan.

I. PENDAHULUAN

Di zaman modern ini, banyak sekali ide baru yang muncul berdasarkan hasil pemikiran sejumlah ahli di masa lampau. Salah satunya adalah sistem ekonomi yang dianut oleh sebagian besar negara maju di dunia yaitu sistem ekonomi pasar bebas yang juga dikenal dengan nama lain sistem ekonomi liberal. Sistem ekonomi ini memungkinkan terjadinya berbagai hal seperti meningkatnya inovasi, munculnya banyak bidang baru, dan mendorong persaingan. Karena persaingan yang ketat di antara perusahaan besar maka para pemilik perusahaan mencari cara agar dapat memperoleh modal lebih supaya tidak kalah daya saingnya. Maka muncullah ide untuk membagi kepemilikan perusahaannya dalam bentuk saham dan menjualnya kepada pebisnis lainnya yang ingin berinvestasi. Dengan sistem ini sekarang dimungkinkan bagi seseorang untuk menginvestasikan uangnya ke suatu perusahaan milik orang lain dan ia bisa mendapatkan keuntungan yang dihasilkan oleh perusahaan itu tanpa harus bekerja di dalamnya. Peluang baru ini menyebabkan banyak orang yang beralih profesi menjadi pemegang

saham. Tugasnya setiap hari adalah memperhatikan secara seksama harga-harga saham yang ada di pasar dan melakukan transaksi baik pembelian maupun penjualan di saat yang tepat. Selama ini, para pemegang saham mengandalkan instingnya untuk melakukan transaksi yang dirasanya tepat. Namun, di era kehebatan informatika seperti sekarang tidak ada salahnya agar pemegang saham dapat diberi bantuan oleh sebuah program.

Algoritma yang dirasa paling cocok untuk menyelesaikan permasalahan ini adalah algoritma *Dynamic Programming*. Algoritma ini melakukan pemecahan masalah menjadi beberapa submasalah lalu menyelesaikannya satu per satu. Algoritma ini dikenal karena dapat memecahkan permasalahan seperti *DNA Sequencing*, Teka-teki Hanoi, dan Teka-teki Penjatuhan Telur. Pada permasalahan ini, *Dynamic Programming* akan memperhatikan harga saham yang ada dan melakukan kalkulasi untuk memberikan saran kepada pemegang saham untuk melakukan transaksi. Diharapkan saran yang diberikan dapat memberikan keuntungan sebesar-besarnya kepada pemegang saham.

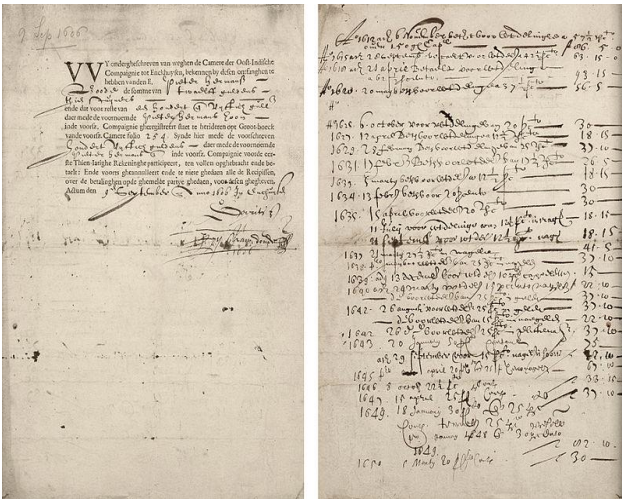
II. DASAR TEORI

2.1 Saham

Perdagangan saham sudah terjadi sejak abad ke-12 di Perancis. Saat itu, saham dikenal dengan nama *courretiers de change*. Pada abad ke-13, Bankir di Republik Venesia menjual sekuritas kepada penduduk kota itu. Hal ini menyebabkan banker di kota-kota di Italia lainnya seperti Pisa, Genoa, Verona, dan Firenze untuk melakukan hal yang sama. Selain itu, banyak perusahaan Italia yang mulai menjual saham kepemilikan kepada masyarakat. Setelah sistem ini terbukti di Italia, perusahaan-perusahaan di Inggris dan Belanda mulai menirunya pada abad ke-16.

Perusahaan bertipe ini dikenal juga dengan nama *joint-stock company* yang terjemahannya ke dalam bahasa Indonesia adalah perseroan terbatas. Salah satu perusahaan pertama di dunia yang sepenuhnya mengandalkan penjualan saham kepemilikannya untuk mendapat modal adalah *Dutch East Indies Company* yang

di Indonesia lebih dikenal dengan sebutan VOC atau kumpeni. Perusahaan ini melakukan penjualan saham karena terlibat persaingan dengan *British East Indies Company* dalam perdagangan rempah-rempah dan akhirnya butuh modal lebih untuk meningkatkan daya saingnya. Selain itu, resiko yang ditanggung juga besar sebab VOC harus terlibat dengan kolonisasi dan juga perang yang tentunya memakan biaya yang besar.



Gambar 1. Saham VOC. Sumber :

http://upload.wikimedia.org/wikipedia/commons/e/ed/VOC_aandeel_9_september_1606.jpg



Gambar 2. Saham di Indonesia. Sumber : http://upload.wikimedia.org/wikipedia/id/a/ac/Surat_Saham_CV.Sejahtera.jpg

Penjualan saham pun terus berlangsung seiring berkembangnya pengaruh kapitalisme dalam ekonomi Inggris dan Amerika Serikat yang akhirnya menyebarkan pengaruhnya ke seluruh dunia. Di dunia modern ini, pasar saham sudah sangat berkembang dan berbagai celah dalam sistem ini sudah diperbaiki sehingga tidak dimungkinkan terjadinya eksploitasi celah tersebut seperti yang pernah terjadi di masa lampau.

Pada perkembangannya, perusahaan yang menjual sahamnya bertanggung jawab secara langsung kepada

para pemegang saham. Oleh karena pertanggungjawaban ini, saham dibagi menjadi dua jenis :

- Saham Biasa
- Saham Preferen

Saham biasa memiliki karakteristik sebagai berikut:

- Hak suara pemegang saham, dapat memilih dewan komisaris
- Hak didahulukan, bila organisasi penerbit menerbitkan saham baru
- Tanggung jawab terbatas, pada jumlah yang diberikan saja

Sementara itu, saham preferen memiliki karakteristik sebagai berikut :

- Memiliki berbagai tingkat, dapat diterbitkan dengan karakteristik yang berbeda
- Tagihan terhadap aktiva dan pendapatan, memiliki prioritas lebih tinggi dari saham biasa dalam hal pembagian dividen
- dividen kumulatif, bila belum dibayarkan dari periode sebelumnya maka dapat dibayarkan pada periode berjalan dan lebih dahulu dari saham biasa
- Konvertibilitas, dapat ditukar menjadi saham biasa, bila kesepakatan antara pemegang saham dan organisasi penerbit terbentuk

Ada beberapa jenis saham preferen, antara lain:

- Saham preferen partisipasi; saham preferen yang membagikan dividen kepada pemegangnya; pemilik saham ini setelah menerima dividen tetap mempunyai hak untuk membagi keuntungan yang dinyatakan sebagai dividen kepada pemegang saham biasa.
- Saham preferen nonkumulatif; saham preferen yang tidak mempunyai hak untuk mendapatkan dividen yang belum dibayarkan pada tahun-tahun yang lalu secara kumulatif .

Dilihat dari kinerja perdagangan, saham dapat dikelompokkan menjadi :

1. *Blue chip stocks*, saham biasa yang memiliki reputasi tinggi, sebagai pemimpin dalam industrinya, memiliki pendapatan yang stabil dan konsisten dalam membayar dividen
2. *Income stocks*, saham suatu emiten dengan kemampuan membayar dividen lebih tinggi dari rata-rata dividen yang dibayarkan pada tahun sebelumnya

3. *Growth stocks*, terdiri dari *well-known* dan *lesser-known*
4. *Speculative stocks*, saham secara konsisten memperoleh penghasilan dari tahun ke tahun, mempunyai kemungkinan penghasilan yang tinggi di masa mendatang, namun belum pasti
5. *Cyclical stocks*, saham yang tidak terpengaruh oleh kondisi ekonomi makro maupun situasi bisnis secara umum
6. *Emerging Growth Stocks*, saham yang dikeluarkan oleh emiten yang relatif kecil dan stabil meskipun dalam kondisi ekonomi yang kurang mendukung
7. *Defensive Stocks*, saham yang tetap stabil dari suatu periode atau kondisi yang tidak menentu dan resesi.

Komponen penting lainnya yang penting dalam perdagangan saham adalah bursa saham. Pialang saham yang ingin membeli atau menjual sahamnya harus menjadi nasabah dari satu bursa tertentu dengan membayar deposit uang dalam jumlah tertentu. Setelah itu barulah mereka dapat melakukan transaksi atas saham perusahaan yang terdaftar juga di bursa efek tersebut. Pembelian saham tidak dilakukan secara satuan namun *sautna lot* yang terdiri dari 500 lembar diterapkan. Penjualan satuan dapat dilakukan di luar bursa. Beberapa perusahaan melakukan praktek yang disebut *dual-listing* yaitu memperjualbelikan sahamnya di lebih dari satu pasar saham. Beberapa bursa saham terkenal di dunia memiliki banyak nasabah seperti Nasdaq / *Nasdaq Stock Market* (Amerika Serikat), NYSE / *New York Stock Exchange* (New York), SEAQ / *Stock Exchange Automated Quotations* (London), Euronext (merger pasar saham antara negara Paris, Amsterdam, dan Brussels), TSE / *Tokyo Stock Exchange* (Tokyo), dan SGX / *Singapore Exchange* (Singapura). Harga saham dapat dipengaruhi oleh permintaan pasar dengan setiap Negara memiliki barometer sendiri seperti Indonesia yang memiliki IHSG (Indeks Harga Saham Gabungan) . Seringkali kondisi perekonomian suatu Negara dapat dilihat dari IHSG Negara tersebut.

2.2 Algoritma Dynamic Programming

Dynamic Programming atau dikenal juga sebagai Program Dinamis merupakan metode pemecahan masalah dengan menguraikan permasalahan menjadi sekumpulan tahapan yang dapat diselesaikan satu per satu. Algoritma ini menyusun tahapan-tahapan yang ada menjadi sebuah rangkaian keputusan yang saling berkaitan. Kebanyakan penyelesaian dalam Program Dinamis menggunakan tabel. Jika dilihat dengan kasat mata, Program Dinamis sangat mirip dengan Algoritma *Greedy* namun letak perbedaannya adalah bahwa *Greedy* hanya menghasilkan satu rangkaian keputusan sementara Program Dinamis

menghasilkan lebih dari satu. Pada Program Dinamis, berlaku prinsip optimalitas yang berbunyi demikian :

“JIKA SOLUSI TOTAL OPTIMAL, MAKA BAGIAN SOLUSI SAMPAI TAHAP KE-K JUGA OPTIMAL.”

Hal ini berarti bahwa jika program bekerja dari tahap k ke $k+1$ maka hasil optimal dari tahap k dapat digunakan tanpa harus kembali ke tahap awal. Misal untuk mendapat hasil $k+1$ dapat dilakukan penambahan hasil dari tahap k dengan hasil dari tahap k ke $k+1$.

Berikut adalah karakteristik dari Program Dinamis :

1. Persoalan dapat dibagi menjadi beberapa tahapan yang hanya diambil satu keputusan pada setiap tahapannya.
2. Masing-masing tahap terdiri dari sejumlah status yang berhubungan dengan tahapan tersebut. Secara umum, status merupakan bermacam kemungkinan masukan yang ada pada tahap tersebut.
3. Hasil dari keputusan yang diambil pada setiap tahapan ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. Hasil pada suatu tahap meningkat secara teratur dengan bertambahnya jumlah tahapan.
5. Hasil pada suatu tahap bergantung pada hasil tahap-tahap yang sudah berjalan dan hasil pada tahap tersebut.
6. Keputusan terbaik pada suatu tahapan bersifat independen terhadap keputusan yang dilakukan dalam tahapan sebelumnya.
7. Adanya hubungan rekursif yang mengidentifikasi keputusan terbaik untuk setiap status pada tahap k memberikan keputusan terbaik untuk setiap status pada tahap $k + 1$. Hubungan inilah yang menyebabkan prinsip optimalitas berlaku.

Terdapat 2 buah pendekatan yang digunakan dalam Program Dinamis yaitu :

- Program Dinamis maju (*forward* atau *up-down*)
- Program Dinamis mundur (*backward* atau *bottom-up*).

Program Dinamis maju melakukan penelusuran dari tahap pertama dan maju sampai tahap ke- n . Sementara Program Dinamis mundur melakukan penelusuran dari tahap ke- n dan mundur ke tahap pertama. Contoh masalah yang dapat diselesaikan oleh Program Dinamis mundur adalah mencari lintasan terpendek. Contoh masalah yang dapat diselesaikan oleh Program Dinamis maju adalah penganggaran modal.

III. PEMECAHAN MASALAH

Harga saham yang dinamis dan tidak menentu menyebabkan sulitnya usaha untuk memprediksi saat yang tepat untuk menjual dan membeli.



Gambar 3. Ilustrasi Harga Saham yang dinamis.
Sumber: <http://hargasaham.org/wpcontent/uploads/2012/11/harga-saham-dild-20-november-2012.png>

Misal harga saham dapat direpresentasikan menjadi sebuah *array of integer* seperti ini:

`harga[] = { 10, 22, 5, 75, 65, 80 }`

Diberikan batasan untuk pialang saham bahwa dalam satu hari ia hanya dapat melakukan dua kali transaksi dimana satu transaksi terdiri dari satu kali membeli dan satu kali menjual. Bagaimanakah caranya untuk memperoleh keuntungan terbesar dalam 2 kali transaksi?

Berikut adalah langkah penyelesaiannya :

- 1) Buat tabel bernama `profit` sebesar *array* harga dan isi nilai setiap selnya dengan 0.
- 2) Secara iterasi dari kanan ke kiri lakukan pencarian dalam *array* harga dan ubah nilai di `profit[i]` menjadi nilai maksimum keuntungan yang didapat dari selisih `harga[i]` dengan anggota *subarray* `harga[i..n-1]`.
- 3) Lakukan iterasi lagi dengan arah sebaliknya dan ubah nilai `profit[i]` menjadi nilai minimum dari selisih `harga[i]` dengan anggota *subarray* `harga[0..i]` sehingga nilai `profit[i]` merupakan nilai maksimum dari kedua buah transaksi.
- 4) Mengembalikan nilai `profit[n-1]`

Saat melakukan tahap ke 2, iterasi dari kanan ke kiri untuk mendapat nilai maksimum. Lalu saat dilakukan tahap ke 3, nilai minimum didapatkan. Iterasi bolak balik dilakukan

untuk menghemat memori karena di tahap 3 *array* profit yang digunakan akan mengandung nilai dari tahap sebelumnya sehingga setelah iterasi ini didapatkan nilai maksimum keuntungan dari kedua transaksi yang dilakukan dengan `profit[0..i]` mengandung nilai maksimum keuntungan dan `profit[i+1..n-1]` mengandung sisanya.

Berikut adalah *pseudo-codenya* :

```
maxProfit(harga, n)
```

```
Input: harga (string of integer),  
n (size of harga).
```

```
Output: maxprofit (integer).
```

```
{Mengisi nilai array profit dengan nilai  
0}
```

```
for (int i=0; i<n; i++) do
```

```
profit[i] ← 0
```

```
{Mendapat nilai maks profit dalam satu  
kali transaksi dari harga[i..n-1]}
```

```
int max_price ← harga[n-1]
```

```
for (int i=n-2; i>=0; i--)
```

```
{max_price berisi nilai max dari  
harga[i..n-1]}
```

```
if (harga[i] > max_price) then
```

```
max_price ← harga[i]
```

```
{Nilai profit[i] didapat dari nilai max  
dari nilai maks sebelumnya (profit[i+1])  
atau dari membeli saat harga[i] dan  
menjual saat max_price}
```

```
profit[i] ← max(profit[i+1], max_price-  
harga[i])
```

```
endfor
```

```
{Mendapat nilai maks profit dari dua transaksi, profit[n-1] akan berisi keuntungan maks}
```

```
int min_price ← harga[0]
```

```
for (int i=1; i<n; i++)
```

```
{min_price adalah nilai min dari harga[0..i]}
```

```
if (harga[i] < min_price) then
```

```
min_price ← harga[i]
```

```
{Nilai maks profit didapat dari nilai max dari nilai maks sebelumnya (profit[i-1]) atau dari membeli saat min_price dan menjual saat harga[i] ditambah dengan profit[i]}
```

```
profit[i] ← max(profit[i-1], profit[i] + (harga[i]-min_price) )
```

```
endfor
```

```
int maxprofit ← profit[n-1]
```

```
return maxprofit
```

IV. ANALISIS HASIL

Untuk harga berikut :

harga[] = { 10, 22, 5, 75, 65, 80 }

Dilakukan pembelian saat nilai 10 dan dijual saat nilai 22 menghasilkan keuntungan 12 dan pembelian saat nilai 5 penjualan saat nilai 80 menghasilkan keuntungan 75. Nilai total keuntungan sebesar $12 + 75 = 87$.

harga[] = { 2, 30, 15, 10, 8, 25, 80 }

Dilakukan pembelian saat nilai 2 dan dijual saat nilai 30 menghasilkan keuntungan 28 dan pembelian saat nilai 8 penjualan saat nilai 80 menghasilkan keuntungan 72. Nilai total keuntungan sebesar $28 + 72 = 100$.

harga[] = { 100, 30, 15, 10, 8, 25, 80 }

Dilakukan pembelian saat nilai 8 dan dijual saat nilai 80 menghasilkan keuntungan 72. Pada kasus ini, program memutuskan bahwa transaksi hanya akan dilakukan sekali agar terjadi keuntungan.

harga[] = { 90, 80, 70, 60, 50 }

Di kasus seperti ini tidak dimungkinkan untuk mendapat keuntungan sehingga program tidak melakukan transaksi sama sekali sebab bila dilakukan pembelian maka pasti akan terjadi kerugian .

Kondisi di atas adalah kondisi yang terjadi saat sebuah nilai saham menurun secara terus menerus dan akhirnya nilainya merosot tajam sehingga menyebabkan sahamnya menjadi tak bernilai lagi. Hal ini yang akan dihindari oleh program dalam keputusannya saat transaksi saham.

Kompleksitas waktu yang dihasilkan oleh program adalah $O(n)$ jika dibandingkan dengan algoritma *brute force* yang menghasilkan kompleksitas waktu $O(n^2)$ dengan *pseudocode* berikut.

```
maxProfit(harga,n)
```

```
Input: harga (string of integer),  
n (size of harga).
```

```
Output: maxprofit (integer).
```

```
int maxprofit ← harga[1] - harga[0]
```

```
int i, j
```

```
for(i = 0; i < n; i++)
```

```
for(j = i+1; j < n; j++)
```

```
if(harga[j] - harga[i] > maxprofit)
```

```
maxprofit ← harga [j] - harga[i]
```

```
endfor
```

```
endfor
```

```
return maxprofit
```

V. KESIMPULAN

Pemrograman Dinamis memiliki banyak kelebihan yaitu fleksibilitasnya dan kompleksitas waktunya yang tidak besar. Oleh karena itu, Program Dinamis dapat dimanfaatkan untuk menyelesaikan masalah penentuan saat yang tepat untuk melakukan pembelian dan penjualan saham. Sebenarnya algoritma lain seperti *Brute Force* dapat juga dipakai untuk menyelesaikan permasalahan ini namun memiliki kompleksitas waktu yang lebih besar dibanding algoritma Program Dinamis. Oleh karena itu, dapat dikatakan bahwa Program Dinamis adalah algoritma paling cocok untuk menyelesaikan masalah ini.

VI. UCAPAN TERIMA KASIH

Penulis ingin mengucapkan kepada Tuhan YME karena telah memampukan penulis untuk menyelesaikan makalah ini dengan baik dan tepat waktu, juga kepada Bapak Rinaldi dan Ibu Ulfa yang telah mengajarkan dasar-dasar teori yang penulis butuhkan untuk menyelesaikan makalah ini. Tidak hanya bahan untuk makalah ini, penulis juga ingin mengucapkan terima kasih kepada kedua dosen yang telah membimbing dan menemani kami selama satu semester ini mempelajari semua pelajaran Strategi Algoritma yang menarik dan telah memberikan pengetahuan baru yang berharga bagi penulis. Terakhir, penulis juga ingin mengucapkan terima kasih kepada keluarga serta teman yang telah mendukung penulis selama proses pembuatan makalah ini.

VII. DAFTAR REFERENSI

- [1] Munir, Rinaldi; Strategi Algoritma. Informatika. Bandung. 2009
- [2] Darmadji, Tjiptono; Hendy, M, Fakhruddin. Pasar Modal di Indonesia. 2001. Indonesia. Salemba Empat. hal 8. Diakses pada 3 Mei 2015 Pk. 09.00
- [3] "Stock Definition". Investopedia. Diakses pada 3 Mei 2015 Pk 09.10.
- [4] "Common Stock vs. Preferred Stock, and Stock Classes". *InvestorGuide.com*. Diakses pada 3 Mei 2015 Pk 09.10.
- [5] Bellman, Richard (1957), *Dynamic Programming*, Princeton University Press. Dover paperback edition (2003), ISBN 0-486-42809-5. Diakses pada 3 Mei 2015 Pk 10.00.
- [6] <http://www.investopedia.com/articles/07/stock-exchange-history.asp>. Diakses pada 3 Mei 2015 Pk 09.00.
- [7] <https://www.topcoder.com/community/data-science/data-science-tutorials/dynamic-programming-from-novice-to-advanced/>. Diakses pada 3 Mei 2015 Pk 09.00.

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, 3 Mei 2015



Jonathan Benedict-13513003