

# Penerapan Algoritma Program Dinamis dalam Diversifikasi Investasi Saham

Roselina - 13515034

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13515034@std.stei.itb.ac.id

**Abstraksi**—Saham adalah salah satu instrumen finansial yang banyak dipilih masyarakat untuk menginvestasikan uangnya. Saham sendiri adalah tanda bukti kepemilikan seseorang atas suatu perusahaan. Menginvestasikan sejumlah uang ke suatu perusahaan pun tentu saja mempunyai resiko, karena perusahaan tersebut bisa saja suatu saat merugi atau tidak bisa beroperasi lagi. Untuk meminimalkan resiko kehilangan uang karena suatu perusahaan, seorang investor pun harus melakukan diversifikasi, yaitu menginvestasikan uangnya ke banyak perusahaan. Dalam hal memilih saham mana saja yang akan dibeli tentu saja harus mempunyai strategi. Makalah ini akan dibahas tentang strategi membeli saham dengan algoritma program dinamis sehingga investor diharapkan dapat memperoleh keuntungan yang maksimal.

**Kata Kunci**—saham, indeks LQ45, diversifikasi, program dinamis.

## I. PENDAHULUAN

Investasi merupakan hal yang sangat penting untuk masa depan yang aman secara finansial. Saat ini, sudah banyak instrumen finansial yang dapat dipilih oleh seseorang untuk menjamin masa tuanya, seperti deposito, reksa dana, saham, dan obligasi. Instrumen-instrumen tersebut pun mempunyai profil resiko yang berbeda-beda dan tentu saja dengan prospek keuntungan yang berbeda-beda pula.

Instrumen seperti deposito dan obligasi mempunyai resiko yang rendah, yang berarti modal yang diinvestasikan disana cukup aman dari resiko kehilangan dalam jumlah besar. Karena itu, deposito dan obligasi cocok untuk orang-orang yang mempunyai tanggungan besar atau harus mempunyai kondisi keuangan yang stabil seperti orang yang sudah berkeluarga atau lansia.

Sebaliknya, instrumen saham lebih cocok untuk orang-orang yang mempunyai profil resiko tinggi, seperti mahasiswa atau profesional muda yang dapat menanggung fluktuasi dari modal yang diinvestasikan. Resiko yang tinggi pun membuat saham mempunyai prospek keuntungan yang tinggi, sesuai salah satu prinsip investasi yaitu *high risk high return*. Karena itulah saham banyak dipilih oleh anak muda yang sadar akan pentingnya investasi.

Walaupun demikian, bukan berarti investor dengan profil resiko rendah tidak bisa berinvestasi saham. Ada satu cara bagi investor berprofil rendah untuk meminimalkan resiko dari

investasi saham, yaitu dengan melakukan diversifikasi. Diversifikasi inilah yang sebenarnya dilakukan oleh manajer pengelola reksa dana, sehingga akibatnya tingkat resiko pada reksa dana menjadi lebih kecil daripada resiko investasi saham.

Tingkat resiko yang tinggi dari saham muncul karena investor menaruh semua telurnya didalam satu keranjang, yang artinya menginvestasikan uangnya hanya pada satu atau dua perusahaan. Sehingga, kalau terjadi apa-apa pada perusahaan tersebut maka modal yang diinvestasikan pun akan berkurang dengan signifikan.

Tetapi, hal ini umum dilakukan bila sang investor adalah seorang investor berpengalaman yang sangat mengerti perusahaan yang diinvestasikannya, sedangkan bagi investor awam yang belum punya banyak pengalaman, ada baiknya melakukan diversifikasi sehingga resiko yang ditanggung pun bisa berkurang.

Dalam makalah ini akan dibahas salah satu strategi diversifikasi yang akan diaplikasikan ke indeks saham Indonesia yaitu LQ45 dengan menggunakan algoritma optimasi program dinamis. Ide dari strategi diversifikasi yang akan dibahas adalah menggunakan algoritma program dinamis untuk memaksimalkan potensi keuntungan dengan melihat histori pertumbuhan suatu saham dari beberapa tahun lalu serta dari pembagian dividen.



Gambar 1: Bursa Efek Indonesia

Sumber: <http://www.businesstimes.com.sg/stocks/indonesia-stock-exchange-caps-daily-share-price-declines-at-10>

## II. LANDASAN 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

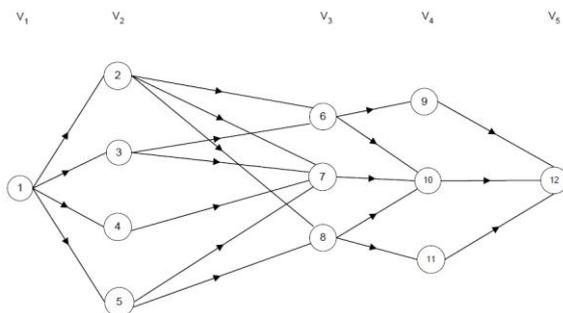
ongkos pada tahap  $k + 1 =$  (ongkos yang dihasilkan pada tahap  $k$ ) + (ongkos dari tahap  $k$  ke tahap  $k + 1$ )

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 multistage (*multistage graph*). Tiap simpul di dalam graf tersebut menyatakan status, sedangkan  $V_1, V_2, \dots$  menyatakan tahap.



Gambar 2: 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 mengidentifikasi 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 pendekatan (*approach*) berbeda: maju (*forward* atau *up-down*) dan mundur (*backward* atau *bottom-up*). Misalkan  $x_1, x_2, \dots, x_n$  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  $x_1, x_2, \dots, x_n$ . 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  $k = 1, 2, \dots, n - 1$ .
- 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  $x_n, x_{n-1}, \dots, x_1$ . 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  $k = n, n - 1, \dots, 1$ .

Penyelesaian secara maju atau mundur keduanya ekuivalen dan menghasilkan solusi optimum yang sama. Pengalaman menunjukkan bahwa penyelesaian dengan program dinamis mundur umumnya lebih mangkus. Kebanyakan literatur perihal program dinamis menyajikan penyelesaian persoalan dengan pendekatan 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. Karakteristik 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. Saham

### B.1 Definisi dan Jenis-Jenis Saham

Saham adalah sebuah sekuritas yang menandakan kepemilikan atas suatu perusahaan dan merepresentasikan klaim atas sebagian aset dan pendapatan perusahaan.

Ada dua jenis saham, yaitu saham biasa dan saham preferen, dengan ciri-ciri sebagai berikut:

1. Saham biasa
  - Pemegang saham memiliki hak suara untuk memilih dewan komisaris.
  - Bila organisasi penerbit menerbitkan saham baru, maka hak pemegang saham didahulukan.
  - Tanggung jawab terbatas pada tingkat kepemilikan pemegang saham.
2. Saham preferen
  - Dapat diterbitkan dalam beberapa tingkat dengan masing-masing memiliki ciri-ciri yang berbeda.
  - Memiliki prioritas lebih tinggi dari saham biasa dalam hal pembagian dividen.
  - Dividen bersifat kumulatif, sehingga bila ada dividen yang belum dibayarkan sebelumnya, maka dapat dibayarkan pada periode berjalan dan lebih dahulu dari saham biasa.
  - Saham dapat ditukar menjadi saham biasa dengan kesepakatan antara pemegang saham dan organisasi penerbit.

### B.2 Terminologi

Berikut ini adalah beberapa terminologi penting:

1. Ticker: Kode perdagangan saham yang terdiri dari 4 karakter berupa alfabet yang merepresentasikan suatu perusahaan publik di Bursa Efek Indonesia.
2. Emiten: Pihak yang melakukan kegiatan penawaran efek (saham atau obligasi) untuk menjual efek kepada masyarakat.
3. Indeks: Harga atau nilai dari sekelompok saham yang dikumpulkan berdasarkan kategori tertentu dan merupakan indikator pergerakan harga dari seluruh saham yang diwakilinya.
4. Pertumbuhan modal (*capital gain*): Kenaikan dari nilai aset yang membuat aset tersebut memiliki nilai lebih tinggi dari harga beli.
5. Penurunan modal (*capital loss*): Perbedaan diantara harga beli dan harga jual, dimana harga jual lebih rendah daripada harga beli.
6. Dividen: Pembagian sebagian keuntungan dari sebuah perusahaan ke pemegang sahamnya.
7. Rasio dividen (*dividend yield*): Rasio dari nilai dividen per saham dibanding harga sahamnya.
8. Saham blue chip: Saham dari perusahaan besar yang mempunyai pendapatan stabil dan sudah beroperasi dalam waktu yang lama.
9. Portfolio: Kumpulan investasi yang dimiliki oleh perusahaan investasi, hedge fund, lembaga keuangan, atau perorangan.

10. Diversifikasi: Teknik manajemen resiko yang menggabungkan beragam investasi dalam satu portfolio untuk meminimalkan dampak yang dimiliki setiap sekuritas terhadap keseluruhan kinerja portfolio.
11. Bursa Efek Indonesia (BEI): Bursa hasil penggabungan dari Bursa Efek Surabaya (BES) sebagai pasar obligasi dan derivatif dengan Bursa Efek Jakarta (BEJ) sebagai pasar saham demi efektivitas operasional dan transaksi.

### B.3 Indeks Saham di Bursa Efek Indonesia

Berikut ini adalah beberapa jenis indeks saham di Bursa Efek Indonesia:

1. Indeks Harga Saham Gabungan (IHSG) adalah indeks yang mencakup pergerakan harga seluruh saham biasa dan saham preferen yang tercatat di Bursa Efek Indonesia. Indeks ini diperkenalkan pertama kali pada tanggal 1 Agustus 1983 sebagai indikator pergerakan saham di Bursa Efek Jakarta. Hari Dasar untuk perhitungan IHSG adalah tanggal 10 Agustus 1982. Pada tanggal tersebut, Indeks ditetapkan dengan Nilai Dasar 100 dan saham tercatat pada saat itu berjumlah 13 saham.
2. Indeks Kompas 100 adalah suatu indeks saham dari 100 saham perusahaan publik yang diperdagangkan di Bursa Efek Indonesia. Indeks Kompas 100 secara resmi diterbitkan oleh Bursa Efek Indonesia (BEI) bekerjasama dengan koran Kompas pada hari Jumat tanggal 10 Agustus 2007. Saham-saham yang terpilih untuk dimasukkan dalam indeks Kompas 100 ini selain memiliki likuiditas yang tinggi, serta nilai kapitalisasi pasar yang besar, juga merupakan saham-saham yang memiliki fundamental dan kinerja yang baik.
3. Jakarta Islamic Index (JII) adalah indeks saham dari 30 emiten yang masuk dalam kriteria syariah berdasarkan Daftar Efek Syariah yang diterbitkan OJK dengan mempertimbangkan kapitalisasi pasar dan likuiditas.
4. Indeks LQ45 adalah indeks saham dari 45 emiten yang dipilih berdasarkan tingkat likuiditas dan kapitalisasi pasar, yang diseleksi melalui beberapa kriteria pemilihan.

### B.4 Daftar Saham LQ45

Berikut ini adalah daftar saham yang termasuk dalam indeks LQ45 periode Agustus 2016 - Januari 2017:

No.	Kode	Nama Emiten
1	AALI	Astra Agro Lestari Tbk.
2	ADHI	Adhi Karya (Persero) Tbk.
3	ADRO	Adaro Energy Tbk.
4	AKRA	AKR Corpindo Tbk.
5	ANTM	Aneka Tambang (Persero) Tbk.
6	ASII	Astra International Tbk.
7	ASRI	Alam Sutera Realty Tbk.
8	BBCA	Bank Central Asia Tbk.

9	BBNI	Bank Negara Indonesia (Persero) Tbk.
10	BBRI	Bank Rakyat Indonesia (Persero) Tbk.
11	BBTN	Bank Tabungan Negara (Persero) Tbk.
12	BMRI	Bank Mandiri (Persero) Tbk.
13	BMTR	Global Mediacom Tbk.
14	BSDE	Bumi Serpong Damai Tbk.
15	CPIN	Charoen Pokphand Indonesia Tbk.
16	ELSA	Elnusa Tbk.
17	GGRM	Gudang Garam Tbk.
18	HMSP	H.M. Sampoerna Tbk.
19	ICBP	Indofood CBP Sukses Makmur Tbk.
20	INCO	Vale Indonesia Tbk.
21	INDF	Indofood Sukses Makmur Tbk.
22	INTP	Indocement Tunggak Prakarsa Tbk.
23	JSMR	Jasa Marga (Persero) Tbk.
24	KLBF	Kalbe Farma Tbk.
25	LPKR	Lippo Karawaci Tbk.
26	LPPF	Matahari Department Store Tbk.
27	LSIP	PP London Sumatra Indonesia Tbk.
28	MNCN	Media Nusantara Citra Tbk.
29	MPPA	Matahari Putra Prima Tbk.
30	MYRX	Hanson International Tbk.
31	PGAS	Perusahaan Gas Negara (Persero) Tbk.
32	PTBA	Tambang Batubara Bukit Asam (Persero) Tbk.
33	PTPP	PP (Persero) Tbk.
34	PWON	Pakuwon Jati Tbk.
35	SCMA	Surya Citra Media Tbk.
36	SILO	Siloam International Hospitals Tbk.
37	SMGR	Semen Indonesia (Persero) Tbk.
38	SMRA	Summarecon Agung Tbk.
39	SRIL	Sri Rejeki Isman Tbk.
40	SSMS	Sawit Sumbermas Sarana Tbk.
41	TLKM	Telekomunikasi Indonesia (Persero) Tbk.
42	UNTR	United Tractors Tbk.
43	UNVR	Unilever Indonesia Tbk.
44	WIKA	Wijaya Karya (Persero) Tbk.
45	WSKT	Waskita Karya (Persero) Tbk.

### III. RUMUSAN MASALAH

Permasalahan yang harus dipecahkan adalah persoalan optimasi prospek keuntungan dari saham-saham yang dipilih dengan modal pembelian saham tidak melebihi yang sudah ditetapkan. Pemecahan persoalan disini juga menggunakan teknik diversifikasi, yang artinya modal yang sudah ditetapkan akan diinvestasikan ke beberapa saham sekaligus sehingga resiko yang ditanggung dapat diminimalkan.

Persoalan akan dipecahkan menggunakan algoritma program dinamis yang sudah diimplementasikan menggunakan bahasa pemrograman Java dengan data bersumber dari 45 saham yang termasuk dalam indeks LQ45. Berikut ini akan dilakukan contoh langkah-langkah pemecahan masalah.

Tabel berikut adalah tabel harga saham dari sampel 5 saham dalam indeks LQ45 selama 6 tahun (2012 - 2017) beserta jumlah dividen tahun 2017 per lembar saham.

Kode	Harga Saham (Per 1 Januari)						Dividen (2017)
	2017	2016	2015	2014	2013	2012	
BBCA	15300	13100	13375	9925	9650	8000	200
BBNI	5700	4910	6350	4360	3925	3625	213
BBRI	11725	11225	11675	8325	7950	6850	429
BMRI	10900	9600	11000	8700	9050	6700	266
ICBP	8400	7225	7250	5500	4000	2550	256

Data diatas akan diolah lebih lanjut dengan mencari pertumbuhan (atau pengurangan) harga saham per tahun dari satu tahun sebelumnya. Setelah itu persentase pertumbuhan tersebut akan dirata-rata dan dikali harga saham tahun 2017 untuk mendapatkan proyeksi keuntungan tahun 2018. Terakhir, semua harga akan dikali 100 karena BEI menetapkan minimum pembelian saham adalah 1 lot per kode saham, yaitu 100 lembar. Jadi, tabel berikut adalah kode, harga tahun 2017, proyeksi keuntungan 2018 dan dividen per 1 lot saham.

Kode	Harga Saham	Proyeksi Pertumbuhan	Dividen
BBCA	1530000	223298	20000
BBNI	570000	66591	21300
BBRI	1172500	144489	42900
BMRI	1090000	127439	26600
ICBP	840000	238726	25600

Selanjutnya, persoalan akan diselesaikan dengan program dinamis yang mempunyai relasi rekurens sebagai berikut:

$$f_0(y) = 0, \quad y = 0, 1, 2, \dots, C$$

$$f_k(y) = -\infty, \quad y < 0$$

$$f_k(y) = \max\{f_{k-1}(y), p_k + f_{k-1}(y - c_k)\}, \quad k = 1, 2, \dots, n$$

Keterangan:

1. Tahap (k) adalah proses memasukkan saham ke dalam portfolio.
2. Status (y) menyatakan modal yang tersisa setelah memasukkan saham pada tahap sebelumnya.

#### IV. PEMECAHAN MASALAH

Untuk memudahkan penyelesaian, harga saham akan dijadikan dalam jutaan rupiah dan dibulatkan ke lima ratus ribuan terdekat. Selain itu, proyeksi pertumbuhan harga saham dan dividen saham akan ditambahkan menjadi proyeksi keuntungan. Sehingga tabelnya akan menjadi seperti berikut:

Saham ke- <i>i</i>	Kode	Harga Saham (juta) ( $c_i$ )	Proyeksi Keuntungan ( $p_i$ )
1	BBCA	1.5	243298
2	BBNI	0.5	87891
3	BBRI	1	187389
4	BMRI	1	154039
5	ICBP	1	264326

Dalam studi kasus ini modal ( $C$ ) yang ditetapkan adalah 3 juta rupiah. Penyelesaian dengan program dinamis maju adalah sebagai berikut:

Tahap 1:  $f_1(y) = \max\{f_0(y), 243298 + f_0(y - 1.5)\}$

y	Solusi Optimum			
	$f_0(y)$	$243298 + f_0(y - 1.5)$	$f_1(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$
0	0	$-\infty$	0	(0, 0, 0, 0, 0)
0.5	0	$-\infty$	0	(0, 0, 0, 0, 0)
1	0	$-\infty$	0	(0, 0, 0, 0, 0)
1.5	0	243298	243298	(1, 0, 0, 0, 0)
2	0	243298	243298	(1, 0, 0, 0, 0)
2.5	0	243298	243298	(1, 0, 0, 0, 0)
3	0	243298	243298	(1, 0, 0, 0, 0)

Tahap 2:  $f_2(y) = \max\{f_1(y), 87891 + f_1(y - 0.5)\}$

y	Solusi Optimum			
	$f_1(y)$	$87891 + f_1(y - 0.5)$	$f_2(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$
0	0	$-\infty$	0	(0, 0, 0, 0, 0)
0.5	0	87891	87891	(0, 1, 0, 0, 0)
1	0	87891	87891	(0, 1, 0, 0, 0)
1.5	243298	87891	243298	(1, 0, 0, 0, 0)
2	243298	331189	331189	(1, 1, 0, 0, 0)
2.5	243298	331189	331189	(1, 1, 0, 0, 0)
3	243298	331189	331189	(1, 1, 0, 0, 0)

Tahap 3:  $f_3(y) = \max\{f_2(y), 187389 + f_2(y - 1)\}$

y	Solusi Optimal			
	$f_2(y)$	$187389 + f_2(y - 1)$	$f_3(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$
0	0	$-\infty$	0	(0, 0, 0, 0, 0)
0.5	87891	$-\infty$	87891	(0, 1, 0, 0, 0)
1	87891	187389	187389	(0, 0, 1, 0, 0)
1.5	243298	275280	275280	(0, 1, 1, 0, 0)
2	331189	275280	331189	(1, 1, 0, 0, 0)
2.5	331189	430687	430687	(1, 0, 1, 0, 0)
3	331189	518578	518578	(1, 1, 1, 0, 0)

Tahap 4:  $f_4(y) = \max\{f_3(y), 154039 + f_3(y - 1)\}$

y	Solusi Optimal			
	$f_3(y)$	$154039 + f_3(y - 1)$	$f_4(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$
0	0	$-\infty$	0	(0, 0, 0, 0, 0)
0.5	87891	$-\infty$	87891	(0, 1, 0, 0, 0)
1	187389	154039	187389	(0, 0, 1, 0, 0)
1.5	275280	241930	275280	(0, 1, 1, 0, 0)
2	331189	341428	341428	(0, 0, 1, 1, 0)
2.5	430687	429319	430687	(1, 0, 1, 0, 0)
3	518578	485228	518578	(1, 1, 1, 0, 0)

Tahap 5:  $f_5(y) = \max\{f_4(y), 264326 + f_4(y - 1)\}$

y	Solusi Optimal			
	$f_4(y)$	$264326 + f_4(y - 1)$	$f_5(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$
0	0	$-\infty$	0	(0, 0, 0, 0, 0)
0.5	87891	$-\infty$	87891	(0, 1, 0, 0, 0)
1	187389	264326	264326	(0, 0, 0, 0, 1)
1.5	275280	352217	352217	(0, 1, 0, 0, 1)
2	341428	451715	451715	(0, 0, 1, 0, 1)
2.5	430687	539606	539606	(0, 1, 1, 0, 1)
3	518578	605754	605754	(0, 0, 1, 1, 1)

Dari penyelesaian dengan program dinamis diatas, dapat dilihat kalau prospek keuntungan terbesar dari modal 3 juta yang diinvestasikan selama setahun adalah 605754 rupiah. Prospek keuntungan tersebut dapat didapat dengan menginvestasikan ke saham berkode BBRI, BMRI, dan ICBP.

Berikutnya, akan dicari saham-saham yang mempunyai prospek keuntungan terbesar dari seluruh saham LQ45 menggunakan program dinamis yang diimplementasi menggunakan bahasa pemrograman Java. Untuk data masukkan, semua data dari saham LQ45 dimasukkan ke dalam suatu file eksternal dengan semantik: kode, harga saham dari tahun 2012 sampai dengan 2017, dan dividen. Data masukkan dan kode program tersebut dapat dilihat di lampiran 1 dan 2.

Modal = Rp 5.000.000

```
D:\Roselina_2\Java Program>java -cp . Diversification
Masukkan Modal: 5000000
Saham Yang Dipilih:
Kode | Harga
-----|-----
ADHI | 210000
ADRO | 169500
ASRI | 38200
ELSA | 43200
HMSP | 385000
ICBP | 840000
KLB | 145000
LPPF | 1477500
MPPA | 122500
MYRX | 15200
PTPP | 359000
SMRA | 131000
SRIL | 23000
SSMS | 140000
TLKM | 387000
WKA | 257000
WKT | 255000
Total Harga: 4998100
Prospek Keuntungan Maksimum: 2480500
Sisa Modal: 1900
D:\Roselina_2\Java Program>
```

Modal = Rp 10.000.000

```
D:\Roselina_2\Java Program>java -cp . Diversification
Masukkan Modal: 10000000
Saham Yang Dipilih:
Kode | Harga
-----|-----
ADHI | 210000
ADRO | 169500
ASRI | 38200
BBTN | 190500
BSDE | 183000
ELSA | 43200
HMSP | 385000
ICBP | 840000
KLB | 145000
LPPF | 1477500
MNCN | 169500
MPPA | 122500
MYRX | 15200
PTPP | 359000
PWON | 56000
SCMA | 282000
SMRA | 131000
SRI | 23000
SSMS | 140000
TLKM | 387000
UNVR | 412000
WKA | 257000
WKT | 255000
Total Harga: 9999100
Prospek Keuntungan Maksimum: 3341100
Sisa Modal: 900
D:\Roselina_2\Java Program>
```

### V. ANALISIS

Dari hasil program tersebut ternyata banyak saham yang dapat dibeli hanya dengan modal 5 juta atau 10 juta rupiah. Walaupun investasi tersebut didiversifikasi karena untuk masing-masing kode saham hanya dibeli satu lot (100 lembar), tetapi kita masih bisa mendapat prospek keuntungan yang tidak sedikit.

Pada kenyataannya, proyeksi harga saham dipengaruhi oleh banyak faktor lainnya selain harga saham historis sehingga

prospek keuntungan yang didapat bisa menjadi lebih rendah ataupun lebih tinggi. Selain itu juga ada beberapa perusahaan yang tidak setiap tahun memberi dividen karena mungkin perusahaan tersebut merugi pada tahun itu atau karena ingin menggunakan keuntungan untuk ekspansi perusahaan.

Teknik diversifikasi yang dilakukan oleh program dinamis diatas memang cocok untuk investor yang ingin berinvestasi di pasar modal tetapi mempunyai profil resiko yang rendah. Namun, untuk investor yang mempunyai profil resiko yang tinggi, idealnya adalah untuk tidak melakukan diversifikasi dan menginvestasikan uangnya hanya ke beberapa perusahaan yang memang kondisi keuangannya stabil dan harga sahamnya cenderung selalu meningkat.

### VI. KESIMPULAN

Algoritma program dinamis mempunyai banyak aplikasi dalam berbagai industri. Salah satunya adalah industri finansial seperti pasar modal yang dibahas di makalah ini. Algoritma ini selalu mendapatkan nilai optimal karena tidak ada fungsi heuristik yang digunakan dan semuanya bergantung pada data yang tersedia.

### UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada Ibu Masayu Leylia Khodra, Ibu Nur Ulfa Maulidevi, dan Bapak Rinaldi Munir selaku dosen mata kuliah Strategi Algoritma program studi Informatika yang telah membimbing peserta kuliah ini dengan baik selama satu semester. Selain itu, penulis juga ingin mengucapkan terima kasih kepada orang tua dan teman-teman penulis yang sudah membantu dan memberi semangat kepada penulis dalam menyelesaikan makalah ini.

### REFERENSI

- [1] Munir, Rinaldi. 2007. Diktat Kuliah IF2211 Strategi Algoritma. Bandung.
- [2] May, Ellen. Smart Trader Rich Investor. 2013. Jakarta: Gramedia Pustaka Utama.
- [3] <http://www.idx.co.id/id-id/beranda/perusahaantercatat/profilperusahaantercatat.aspx>. Diakses pada 17 Mei 2017.
- [4] <https://finance.yahoo.com>. Diakses pada 17 Mei 2017
- [5] <http://britama.com>. Diakses pada 17 Mei 2017.

### 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, 17 Mei 2017



Roselina - 13515034

## Lampiran 1

stocklist.txt								
1	AALI	15775	16270	22154	20463	17961	19629	370
2	ADHI	2100	2550	3283	1581	1777	613	26
3	ADRO	1695	525	1000	950	1650	1830	17
4	AKRA	6675	7100	4695	4400	3875	3650	50
5	ANTM	800	329	895	865	1142	1361	0
6	ASII	7950	6450	7850	6425	7350	7890	113
7	ASRI	382	321	595	510	770	485	133
8	BBCA	15300	13100	13375	9925	9650	8000	200
9	BBNI	5700	4910	6350	4360	3925	3625	213
10	BBRI	11725	11225	11675	8325	7950	6850	429
11	BBTN	1905	1365	995	900	1620	1154	49
12	BMRI	10900	9600	11000	8700	9050	6700	266
13	BMTR	600	800	1855	1850	2175	1110	5
14	BSDE	1830	1730	2020	1440	1400	1050	5
15	CPIN	3100	3345	3955	4135	3875	2500	29
16	ELSA	432	230	570	406	175	245	10
17	GGRM	61750	58350	57800	41900	51850	57000	2600
18	HMSP	3850	4140	2532	2526	2385	1601	108
19	ICBP	8400	7225	7250	5500	4000	2550	256
20	INCO	2370	1445	3450	2305	2750	4000	0
21	INDF	7925	6200	7550	6975	6050	4800	168
22	INTP	15025	19700	23000	22400	21750	16950	415
23	JSMR	4220	5737	7183	5163	5487	4365	78

24	KLBF	1450	1335	1865	1250	1090	705	19
25	LPKR	735	1055	1135	950	1030	670	2
26	LPPF	14775	16000	15525	11625	2700	2300	485
27	LSIP	1550	1425	1840	1655	2200	2425	37
28	MNCN	1695	1190	2860	2235	2375	1350	42
29	MPPA	1225	1670	3800	2005	1220	920	26
30	MYRX	152	134	142	119	63	58	0
31	PGAS	2880	2405	5050	4770	4675	3375	75
32	PTBA	11600	4450	11375	9250	15500	20150	286
33	PTPP	3590	3707	3721	1283	827	542	50
34	PWON	560	448	499	307	275	198	5
35	SCMA	2820	2700	3415	2650	2375	1860	83
36	SILO	10900	9571	13556	9349	0	0	0
37	SMGR	9025	11050	14575	14200	15750	11300	305
38	SMRA	1310	1445	1650	955	945	600	5
39	SRIL	230	404	162	235	0	0	3
40	SSMS	1400	1800	1690	880	0	0	18
41	TLKM	3870	3340	2830	2275	1940	1370	137
42	UNTR	21850	17400	17900	19300	19750	28350	536
43	UNVR	41200	36700	35825	28550	22050	19600	375
44	WIKA	2570	2593	3468	1806	1528	658	34
45	WSKT	2550	1620	1431	410	420	0	38

## Lampiran 2

```
Diversification.java x
1  import java.io.File;
2  import java.io.FileNotFoundException;
3  import java.util.Scanner;
4
5  class Diversification {
6
7      private int[][] K;
8      private int[][] P;
9      private Stock[] stockList;
10     private int capital;
11     private int numberOfStocks;
12     private int remainingCapital;
13
14     public Diversification(String filePath, int capital) {
15         this.capital = capital;
16         Scanner input = null;
17         try {
18             input = new Scanner(new File(filePath));
19         } catch (FileNotFoundException e) {
20             e.printStackTrace();
21         }
22         int rows = 0;
23         while (input.hasNextLine()) {
24             rows++;
25             input.nextLine();
26         }
27         initializeStocks(filePath, rows);
28     }
29
30     private void initializeStocks(String filePath, int rows) {
31         stockList = new Stock[rows];
32         numberOfStocks = rows;
33         Scanner input = null;
34         try {
35             input = new Scanner(new File(filePath));
36         } catch (FileNotFoundException e) {
37             e.printStackTrace();
38         }
39         for (int i = 0; i < rows; i++) {
40             String ticker = input.next();
41             int[] priceList = new int[6];
42             for (int j = 0; j < priceList.length; j++) {
43                 priceList[j] = input.nextInt();
44             }
45             int dividend = input.nextInt();
46             stockList[i] = new Stock(ticker, priceList, dividend);
47         }
48     }
49
50     public void printStockList() {
51         for (int i = 0; i < numberOfStocks; i++) {
52             System.out.println(stockList[i].getTicker() + " " + stockList[i].getPrice() + " " + stockList[i].getProfit());
53         }
54     }
55
56     public int getCapital() {
57         return capital;
58     }
59
60     public void setCapital(int capital) {
61         this.capital = capital;
62     }
63 }
```

```

63
64 public int getNumberOfStocks() {
65     return numberOfStocks;
66 }
67
68 public int getMaxProfit() {
69     int capital = this.capital / 100;
70     K = new int[numberOfStocks+1][capital+1];
71     P = new int[numberOfStocks+1][capital+1];
72     for (int i = 0; i <= numberOfStocks; i++) {
73         for (int w = 0; w <= capital; w++) {
74             if (i == 0 || w == 0) {
75                 K[i][w] = 0;
76             }
77             else if (stockList[i-1].getPrice() <= w) {
78                 if (stockList[i-1].getProfit() + K[i-1][w-stockList[i-1].getPrice()] > K[i-1][w]) {
79                     K[i][w] = stockList[i-1].getProfit() + K[i-1][w-stockList[i-1].getPrice()];
80                     P[i][w] = 1;
81                 } else {
82                     K[i][w] = K[i-1][w];
83                     P[i][w] = -1;
84                 }
85             }
86             else {
87                 K[i][w] = K[i-1][w];
88                 P[i][w] = -1;
89             }
90         }
91     }
92     return K[numberOfStocks][capital] * 100;
93 }
94

```

```

95 public void printChosenStocks() {
96     String[] chosenStocksTicker = new String[45];
97     int[] chosenStocksPrice = new int[45];
98     int chosen = 0;
99     int W = capital / 100;
100    for (int n = numberOfStocks; n > 0; n--) {
101        if (P[n][W] == 1) {
102            chosenStocksTicker[chosen] = stockList[n - 1].getTicker();
103            chosenStocksPrice[chosen] = stockList[n - 1].getPrice() * 100;
104            chosen++;
105            W -= stockList[n - 1].getPrice();
106        }
107    }
108    System.out.println("Kode | Harga");
109    System.out.println("-----");
110    for (int i = chosen - 1; i >= 0; i--) {
111        System.out.println(chosenStocksTicker[i] + " | " + chosenStocksPrice[i]);
112    }
113    remainingCapital = W * 100;
114    System.out.println("Total Harga: " + (capital - remainingCapital));
115 }
116
117 public int getRemainingCapital() {
118     return remainingCapital;
119 }
120

```

```

121 public static void main(String[] args) {
122     Scanner input = new Scanner(System.in);
123     System.out.print("Masukkan Modal: ");
124     int capital = input.nextInt();
125     Diversification D = new Diversification("D:\\Roselina_2\\Java Program\\stocklist.txt", capital);
126     int maxProfit = D.getMaxProfit();
127     System.out.println("Saham Yang Dipilih:");
128     D.printChosenStocks();
129     System.out.println("Prospek Keuntungan Maksimum: " + maxProfit);
130     System.out.println("Sisa Modal: " + D.getRemainingCapital());
131 }
132
133

```

```
Stock.java
1 class Stock {
2
3     private String ticker;
4     private int price;
5     private int profit;
6
7     public Stock(String ticker, int[] priceList, int dividend) {
8         this.ticker = ticker;
9         this.price = priceList[0];
10        float projectedGrowth = 0;
11        int countYear = 0;
12        for (int i = 0; i < priceList.length - 1; i++) {
13            if (priceList[i + 1] != 0) {
14                float growth = priceList[i] - priceList[i + 1];
15                growth /= priceList[i + 1];
16                projectedGrowth += growth;
17                countYear++;
18            }
19        }
20        projectedGrowth = (projectedGrowth / countYear) * priceList[0];
21        this.profit = (int)projectedGrowth + dividend;
22    }
23
24    public String getTicker() {
25        return ticker;
26    }
27
28    public int getPrice() {
29        return price;
30    }
31
32    public int getProfit() {
33        return profit;
34    }
35
36 }
```