

Optimasi Portofolio Saham dengan Menggunakan Algoritma Greedy

Raynard Tanadi - 13521143
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail : 13521143@std.stei.itb.ac.id

Abstract—Portofolio saham adalah sekumpulan saham yang dimiliki oleh seorang investor. Dalam membangun suatu portofolio saham, terkadang investor masih membuat kesalahan sehingga portofolio menyimpang dari tujuan investasinya. Maka dari itu, diperlukan suatu optimasi portofolio yang dapat memperbaiki suatu portofolio supaya sejalan dengan suatu tujuan investasi. Salah satu optimasi portofolio yang dapat dilakukan adalah dengan menerapkan persoalan *fractional knapsack* yang diselesaikan dengan algoritma greedy.

Keywords—Optimasi; Portofolio; Algoritma Greedy; Knapsack;

I. PENDAHULUAN

Investasi merupakan aktivitas penanaman modal dengan tujuan memperoleh keuntungan. Terdapat berbagai macam instrument investasi, salah satunya saham.

Dalam melakukan investasi saham, setiap investor tentunya memiliki tujuan investasinya masing – masing. Investor juga akan memiliki *expected return* dan *risk tolerance* yang berbeda – beda. Maka dari itu, investor seharusnya membuat portofolio saham yang sesuai dengan tujuan investasi yang dimiliki, *expected return* yang sesuai, dan *risk* yang sanggup dihadapi. Akan tetapi, sering sekali investor membuat kesalahan dalam membangun portofolionya sehingga tidak sejalan dengan ketiga hal tersebut, seperti *fear of missing out*, *overconfident*, mengikuti keputusan investasi orang lain tanpa analisa sendiri, dan tidak mau mengakui dirinya salah.

Meskipun terkadang investor sudah mengetahui bahwa portofolionya sudah tidak sejalan dengan tujuan investasinya, investor merasa ragu untuk memperbaikinya dan merasa sudah terlanjut. Padahal, langkah terbaik yang dapat dilakukan oleh investor dalam kondisi tersebut adalah memperbaikinya. Oleh karena itu, dibutuhkan optimasi portofolio sehingga investor dapat mengevaluasi portofolio saham yang dimilikinya dan dapat dengan yakin memperbaiki portofolio tersebut.

Optimasi portofolio saham yang dilakukan tentu tidak akan sempurna karena hanya dalam lingkup penggunaan persoalan *fractional knapsack* dengan algoritma greedy yang tidak begitu memasukkan beberapa faktor penting lainnya, seperti pemilihan sektor, likuiditas, kondisi pasar, dan lain – lain. Akan tetapi, optimasi portofolio saham sederhana ini dapat membantu investor untuk tidak mengambil keputusan yang

beresiko dan dapat mengevaluasi, serta memperbaiki portofolio yang dimiliki.

II. TEORIDASAR

A. Greedy Algorithm

Algoritma *greedy* merupakan metode yang paling populer dan sederhana untuk memecahkan persoalan optimasi sehingga dapat mencari solusi optimal. Terdapat dua macam persoalan optimasi, yaitu makimasi dan minimasi.

Dalam algoritma *greedy*, algoritma akan memecahkan persoalan secara langkah per langkah sedemikian sehingga pada setiap langkah algoritma akan mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan, serta berharap bahwa dengan memilih optimum local pada setiap langkah akan berakhir dengan optimum global.

Terdapat 6 elemen algoritma *greedy* :

1. Himpunan kandidat, C: berisi kandidat yang akan dipilih pada setiap langkah (contoh: *task*, koin, benda, dan lain – lain).
2. Himpunan solusi, S: berisi kandidat yang sudah dipilih.
3. Fungsi solusi: menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi.
4. Fungsi seleksi (*selection function*): memilih kandidat berdasarkan strategi *greedy* tertentu. Strategi *greedy* ini bersifat heuristik.
5. Fungsi kelayakan (*feasible*): memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi (layak atau tidak).
6. Fungsi obyektif: memaksimumkan atau meminimumkan.

Algoritma *greedy* melibatkan pencarian sebuah himpunan bagian, S dari himpunan kandidat, C; yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu S menyatakan suatu solusi dan S dioptimasi oleh fungsi obyektif.

Algoritma *greedy* juga lebih cocok digunakan untuk persoalan yang tidak diperlukan solusi terbaik yang mutlak

karena algoritma ini bisa digunakan untuk menghasilkan solusi hampiran.

B. Fractional Knapsack Problem

Fractional Knapsack Problem adalah varian persoalan *knapsack* yang solusinya boleh dalam bentuk pecahan atau fraksi.



Objek yang dimasukkan ke dalam *knapsack* dapat berupa sebagian saja ($0 \leq x_i \leq 1$).

Maksimasi $F = \sum_{i=1}^n p_i x_i$

dengan kendala (*constraint*)

$$\sum_{i=1}^n w_i x_i \leq K$$

yang dalam hal ini, $0 \leq x_i \leq 1, i = 1, 2, \dots, n$

Ketiga strategi *greedy* dapat digunakan pada persoalan ini, yaitu *greedy by profit*, *greedy by weight*, dan *greedy by density*.

Contoh *fractional knapsack problem* yang diselesaikan dengan algoritma *greedy*.

Contoh 9. Diberikan 3 buah objek sbb:
 $(w_1, p_1) = (18, 25); (w_2, p_2) = (15, 24); (w_3, p_3) = (10, 15);$
 dan sebuah *knapsack* dengan kapasitas $K = 20$.
 Solusi dengan algoritma *greedy*:

Properti objek				<i>Greedy by</i>		
<i>i</i>	w_i	p_i	p_i/w_i	<i>profit</i>	<i>weight</i>	<i>density</i>
1	18	25	1,4	1	0	0
2	15	24	1,6	2/15	2/3	1
3	10	15	1,5	0	1	1/2
Total bobot				20	20	20
Total keuntungan				28,2	31,0	31,5

- Solusi optimal: $X = (0, 1, 1/2)$
- yang memberikan keuntungan maksimum = 31,5.

Dalam persoalan ini, terdapat teorema yang menyebutkan bahwa strategi pemilihan objek berdasarkan densitas p_i/w_i terbesar selalu memberikan solusi optimal.

Maka dari itu, algoritma dapat diawali dengan menghitung densitas p_i/w_i , lalu mengurutkan objek berdasarkan nilai p_i/w_i dari besar ke kecil, dan memasukkan objek sampai kapasitas penuh.

C. Portofolio Saham

Portofolio investasi adalah kumpulan aset investasi milik individu, lembaga keuangan, perusahaan, maupun manajer investasi yang berisi dari berbagai instrument investasi, seperti saham, obligasi, reksa dana, komoditas, atau uang tunai milik investor.

Portofolio saham adalah kumpulan saham yang dimiliki oleh investor. Sebuah portofolio saham dapat menunjukkan profil risiko dari investor rendah atau tinggi.

Secara umum, terdapat beberapa jenis portofolio saham:

1. *Growth Portfolio*:

Portofolio ini dibuat untuk mendorong perkembangan keuntungan dari portofolio tersebut. Portofolio ini lebih berfokus kepada perkembangan aset yang umumnya bersifat *high risk, high return*.

Prinsip ini biasa diadopsi oleh investor yang memiliki profil risiko yang tinggi sehingga berani mengambil risiko lebih tinggi untuk meraih hasil yang lebih besar. Umumnya, portofolio ini lebih berfokus kepada industri yang sedang berkembang dan likuiditas yang menengah atau saham lapis kedua.

2. *Income Portfolio*:

Portofolio ini lebih berfokus pada pengamanan pendapatan regular dari investasi daripada *capital gain*.

Pendapatan regular yang dimaksud biasa disebut dalam bentuk dividen yang umumnya akan dibagikan oleh perusahaan dalam periode waktu tertentu. Akan tetapi, perlu diingat bahwa tidak semua perusahaan rutin membagikan dividen.

3. *Value Portfolio*:

Portofolio ini lebih berfokus kepada saham yang memiliki valuasi yang lebih murah daripada valuasi aslinya atau biasa disebut dengan *undervalued stock*

Untuk membangun suatu portofolio, terdapat tiga tahap umum yang dapat dilakukan, yaitu:

1. Tentukan Tujuan Investasi:

Hal pertama yang harus dilakukan oleh investor adalah menentukan tujuan investasi. Pertama – tama, tujuan investasi dipisah menjadi dua, yaitu jangka panjang dan pendek. Lalu, tujuan investasi dapat lebih diperinci lagi, seperti dana pension, membeli rumah, uang kuliah, atau berbagai tujuan lainnya.

Hal ini penting karena tidak semua instrumen saham cocok untuk investasi jangka pendek. Apalagi jika

tujuan investasinya adalah untuk kehidupan sehari – hari, tentu harus dicari instrumen saham yang dapat memberikan *cashflow* secara rutin dan tidak terlalu berfluktuatif.

2. Kenali Profil Risiko:

Tahap selanjutnya, setiap investor harus mengenal profil risikonya sendiri. Profil risiko dapat dibagi menjadi tiga, yaitu moderat, agresif, dan konservatif.

Dengan mengetahui profil risiko, investor dapat memilih instrumen saham yang lebih cocok sesuai dengan risiko yang sanggup dihadapi sehingga investor juga tidak kaget ketika mengalami kerugian dan tidak terlalu percaya diri ketika mengalami keuntungan besar.

3. Diversifikasi Portofolio:

Terdapat juga diversifikasi portofolio. Diversifikasi ini dilakukan supaya investor dapat menghindari risiko terburuk bila terdapat salah analisa yang mengharuskan investor mengalami kerugian di salah satu instrumen saham.

Akan tetapi, tentunya hal ini bukan merupakan keharusan dan dapat disesuaikan dengan profil risiko dari masing – masing investor. Terdapat metode lain yang bertolak belakang dari diversifikasi, yaitu konsentrasi di mana investor meletakkan modalnya hanya ke dalam sedikit instrumen saham.

III. IMPLEMENTASI

A. Implementasi Fractional Knapsack & Algoritma Greedy

Pada implementasi ini, algoritma *greedy* akan mencari solusi menggunakan *greedy by density*. Profit di sini akan berupa *expected profit* dari suatu instrumen saham dan *weight* di sini akan berupa suatu *expected risk* dari suatu instrumen saham. Lalu, kapasitas di sini akan berupa *risk tolerance* dari investor. Berikut contohnya:

- 1. BBKA = (wi, pi) = (0.1, 0.4)
- 2. BBRI = (wi, pi) = (0.05, 0.35)
- 3. TLKM = (wi, pi) = (0.02, 0.5)

Risk Tolerance = 0.1

Properti Objek				Greedy by		
I	wi	pi	Pi/wi	profit	weight	density
1	0.1	0.4	4	1	3/10	4/5
2	0.05	0.175	3.5	0	1	0
3	0.02	0.1	5	0	1	1
Total bobot				0.1	0.1	0.1
Total Keuntungan				40%	39.5%	42%

Hasil yang didapatkan adalah saham TLKM dengan bobot sebesar 20% dan saham BBKA dengan bobot 80% dari portofolio dengan ekspektasi keuntungan dari portofolio sebesar 42.5%.

Pada contoh, dapat dilihat bahwa *expected return* yang diketahui diubah terlebih dahulu dengan dikali dengan *weight* dibagi dengan *risk tolerance* nya supaya dapat mendapatkan profit yang sesungguhnya.

Dapat terlihat juga bahwa dengan menggunakan *greedy by density*, akan didapatkan hasil yang paling optimal dalam persoalan *fractional knapsack*. Maka dari itu, pada implementasi, program hanya akan mencari *greedy by density*.

B. Implementasi Program

1. Penjelasan Program

Pada program ini, akan terdapat 3 mode, yaitu evaluasi saham, menambah saham, dan mengurangi saham.

Pada mode evaluasi saham, investor dapat memasukkan portofolio saham yang dimilikinya, beserta *expected return*, *expected risk*, dan *risk tolerance*. Lalu, program akan melakukan algoritma *greedy by density* dan akan mengembalikan portofolio saham yang telah dioptimasi, yaitu bobot terbaru dari setiap saham di portofolio.

Lalu, terdapat mode menambah saham, di mana investor juga akan memasukkan hal – hal seperti sebelumnya, lalu investor juga harus memasukkan satu saham yang ingin dimasukkan ke dalam portofolio. Setelah itu, program akan melakukan optimasi terlebih dahulu ke portofolio awal dan program juga akan melakukan optimasi ke portofolio yang ditambah dengan saham baru, jika hasil kedua lebih menguntungkan, maka program akan merekomendasikan untuk menambah saham tersebut, begitu pula sebaliknya.

Terakhir, terdapat mode mengurangi saham di mana investor juga akan memasukkan hal yang sama seperti pada mode evaluasi, tetapi investor juga harus memasukkan satu saham yang ingin dihapus dari portofolio. Lalu, program juga akan melakukan optimasi terlebih dahulu ke portofolio awal dan portofolio yang sudah dikurangi saham tersebut, jika lebih menguntungkan, maka program akan merekomendasikan untuk mengurangi saham tersebut, begitu pula sebaliknya.

Program ini tentunya tidak mencakup berbagai faktor penting, seperti likuiditas fluktuasi, kondisi pasar, diversifikasi, dan lain – lain. Akan tetapi, program ini dapat menjadi optimasi portofolio sederhana yang dapat membantu investor untuk merapikan portofolionya kembali dan menyadarkan investor bahwa terdapat kesalahan dalam portofolionya.

2. Algoritma Greedy & Penyelesaian Fractional Knapsack

```

import numpy as np

def fractional_greedy(stocks, stocks2, mode,
riskTolerance) :

    n = len(stocks)
    result = np.empty((n,2), dtype=object)
    portfolioReturn = 0
    flag = False
    i = 0
    riskToleranceFix = riskTolerance

    #Sort by density
    sortedStocks = stocks[stocks[:, 3].argsort()][::-1]

    if (mode == 2 or mode == 3):
        sortedStocks2 = stocks2[stocks2[:,
3].argsort()][::-1]

    for stock in sortedStocks :
        stockWeight = min(riskTolerance, stock[2])
        result[i] = stock[0],
"{:.2f}".format(stockWeight/riskToleranceFix*100)
        portfolioReturn = portfolioReturn +
stockWeight/riskToleranceFix*stock[1]*100
        riskTolerance -= stockWeight
        i+=1
        if (riskTolerance == 0) :
            break

    if (mode == 2 or mode == 3) :
        i = 0
        n = len(stocks2)
        result2 = np.empty((n,2), dtype=object)
        portfolioReturn2 = 0
        for stock in sortedStocks2 :
            stockWeight = min(riskTolerance, stock[2])
            result2[i] = stock[0],
"{:.2f}".format(stockWeight/riskToleranceFix*100)
            portfolioReturn2 = portfolioReturn2 +
stockWeight/riskToleranceFix*stock[1]*100

```

```

        riskTolerance -= stockWeight
        i+=1

        if (riskTolerance == 0) :
            break

        if (portfolioReturn2 > portfolioReturn) :
            result = result2
            portfolioReturn = portfolioReturn2
            flag = True

    return result, flag,
"{:.2f}".format(portfolioReturn)

```

3. Input Controller

```

def input_controller(mode) :
    n = int(input("Jumlah Saham di Portofolio: "))
    stocks = np.empty((n,4), dtype = object)
    stocks2 = None
    riskTolerance = float(input("Berapa risk
maksimal kamu? (desimal):"))

    print("Input saham - saham yang ada di
portofolio Anda")
    for i in range (n) :
        stock_name = input (f"Nama saham {i+1}: ")
        stock_return = float(input (f"Expected return
saham {stock_name} (dalam desimal berapa
persen): "))
        stock_risk = float(input (f"Expected risk
saham {stock_name} (dalam desimal berapa
persen): "))
        stock_return =
stock_risk/riskTolerance*stock_return

        stocks[i] = [stock_name, stock_return,
stock_risk, stock_return/stock_risk]

    if (mode == 2) :
        print("Input saham yang ingin anda
tambahkan")

```

```

stocks2 = stocks.copy()
stock_name = input("Nama saham: ")
stock_return = float(input("Expected return
saham {stock_name} (dalam desimal berapa
persen): "))
stock_risk = float(input("Expected risk saham
{stock_name} (dalam desimal berapa persen): "))
stock_return =
stock_risk/riskTolerance*stock_return

stocks2.append(stock_name, stock_return,
stock_risk, stock_return/stock_risk)
elif (mode == 3):
print("Input saham yang ingin anda
keluarkan")
stocks2 = stocks.copy()
stock_name = input("Nama saham: ")
stock_return = float(input("Expected return
saham {stock_name} (dalam desimal berapa
persen): "))
stock_risk = float(input("Expected risk saham
{stock_name} (dalam desimal berapa persen): "))
stock_return =
stock_risk/riskTolerance*stock_return

idx = np.where(stocks2[:,0] ==
stock_name)[0]
stocks2.remove(stocks2, idx, axis = 0)

return stocks, stocks2, riskTolerance

```

4. Main

```

def main():
print("Selamat datang di optimasi portofolio
saham\nTerdapat 3 pilihan dalam program ini.
Berikut adalah pilihannya: \n1. Evaluasi Portofolio
Saham \n2. Tambah Saham \n3. Kurangi Saham")
mode = int(input("Apa yang ingin Anda lakukan
(1/2/3): "))

stocks, stocks2, riskTolerance =
input_controller(mode)
result, flag, portfolioReturn =
fractional_greedy(stocks, stocks2, mode,
riskTolerance)

```

```

print("Optimized Portfolio: ")
print(result)
if (not flag):
for stock, weight in result:
print("Stock: " + str(stock) + ", Weight: " +
str(weight) + "%")
print(f"Portfolio Return: {portfolioReturn}%")
if (mode == 2 or mode == 3):
if (flag):
for stock, weight in result:
print("Stock: " + str(stock) + ", Weight: "
+ str(weight) + "%")
print(f"Portfolio Return: {portfolioReturn}%")
else:
if (mode == 2):
print("Don't add stock")
else:
print("Don't delete stock")

main()

```

C. Test Case Program

1. Tes Evaluasi Saham

a. Input

```

Selamat datang di optimasi portofolio saham
Terdapat 3 pilihan dalam program ini. Berikut adalah pilihannya:
1. Evaluasi Portofolio Saham
2. Tambah Saham
3. Kurangi Saham
Apa yang ingin Anda lakukan (1/2/3): 1
Jumlah Saham di Portofolio: 3
Berapa risk maksimal kamu? (desimal):0.1
Input saham - saham yang ada di portofolio Anda
Nama saham 1: BBKA
Expected return saham BBKA (dalam desimal berapa persen): 0.4
Expected risk saham BBKA (dalam desimal berapa persen): 0.1
Nama saham 2: BBRI
Expected return saham BBRI (dalam desimal berapa persen): 0.35
Expected risk saham BBRI (dalam desimal berapa persen): 0.05
Nama saham 3: TLKM
Expected return saham TLKM (dalam desimal berapa persen): 0.5
Expected risk saham TLKM (dalam desimal berapa persen): 0.02

```

b. Output

```

Optimized Portfolio:
Stock: TLKM, Weight: 20.00%
Stock: BBKA, Weight: 80.00%
Stock: None, Weight: None%
Portfolio Return: 42.00%

```

2. Tes Menambah Saham

a. Input

```
Selamat datang di optimasi portofolio saham
Terdapat 3 pilihan dalam program ini. Berikut adalah pilihannya:
1. Evaluasi Portofolio Saham
2. Tambah Saham
3. Kurangi Saham
Apa yang ingin Anda lakukan (1/2/3): 2
Jumlah Saham di Portofolio: 1
Berapa risk maksimal kamu? (desimal):0.2
Input saham - saham yang ada di portofolio Anda
Nama saham 1: bbca
Expected return saham bbca (dalam desimal berapa persen): 0.4
Expected risk saham bbca (dalam desimal berapa persen): 0.2
Input saham yang ingin anda tambahkan
Nama saham: mtdl
Expected return saham mtdl (dalam desimal berapa persen): 0.8
Expected risk saham mtdl (dalam desimal berapa persen): 0.2
```

```
Optimized Portfolio:
Stock: MTDL, Weight: 100.00%
Stock: None, Weight: None%
Portfolio Return: 80.00%
Don't delete stock
```

IV. KESIMPULAN

Berdasarkan penelitian ini, dapat disimpulkan bahwa algoritma greedy juga dapat membantu melakukan optimasi sederhana pada portofolio saham sehingga dapat mendapatkan return yang optimal.

b. Output

```
[[ 'mtdl' 0.8 0.2 '4.00' ]
 [ 'bbca' 0.4 0.2 '2.00' ]
80.0
Optimized Portfolio:
Stock: mtdl, Weight: 100.00%
Stock: None, Weight: None%
Portfolio Return: 80.00%
add stock
```

REFERENCES

- [1] <https://accurate.id/ekonomi-keuangan/portofolio-saham/> diakses pada 22 Mei 2023
- [2] <https://www.ocbcnisp.com/id/article/2021/09/27/portofolio-investasi> diakses pada 22 mei 2023
- [3] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf) diakses pada 22 mei 2023

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, 22 Mei 2023

3. Tes Mengurangi Saham

a. Input

```
Selamat datang di optimasi portofolio saham
Terdapat 3 pilihan dalam program ini. Berikut adalah pilihannya:
1. Evaluasi Portofolio Saham
2. Tambah Saham
3. Kurangi Saham
Apa yang ingin Anda lakukan (1/2/3): 3
Jumlah Saham di Portofolio: 2
Berapa risk maksimal kamu? (desimal):0.2
Input saham - saham yang ada di portofolio Anda
Nama saham 1: BBKA
Expected return saham BBKA (dalam desimal berapa persen): 0.4
Expected risk saham BBKA (dalam desimal berapa persen): 0.2
Nama saham 2: MTDL
Expected return saham MTDL (dalam desimal berapa persen): 0.8
Expected risk saham MTDL (dalam desimal berapa persen): 0.2
Input saham yang ingin anda keluarkan
Nama saham: MTDL
Expected return saham MTDL (dalam desimal berapa persen): 0.8
Expected risk saham MTDL (dalam desimal berapa persen): 0.2
```



Raynard Tanadi dan 13521143

b. Output

□