

PROGRAM KOMPUTER PENENTUAN PROPORSI SAHAM PORTFOLIO MENGGUNAKAN QUADRATIC PROGRAMMING WOLFE

computer program for determining portfolio stock proportions using quadratic programming wolfe

Ngarap Imanuel Manik^{1*}, Elvia Yoe²

Computer Science–Applied Mathematics,
Bina Nusantara University
Jln. Kebon Jeruk Raya No.27, Jakarta 11480, Indonesia

Corresponding author : Tel: 0812-1104453

Corresponding author e-mail: manik@binus.ac.id

Received: April 17, 2026. **Revised:** June 29, 2026. **Accepted:** July 15, 2026. **Issue Period:** Vol.10 No.3 (2026), Pp. 760-771

Abstrak: Para penanam modal umumnya menginginkan keuntungan yang besar dari investasi yang mereka lakukan dengan risiko kerugian yang kecil dan menanam modal mereka ke beberapa jenis saham (diversifikasi saham). Masalah yang kemudian muncul adalah bagaimana mereka menentukan besarnya proporsi saham dalam portfolio yang akan mereka tanami modal sehingga memberikan return (keuntungan) yang besar dengan risiko yang kecil. Untuk menyelesaikan masalah tersebut, penulis akan mengimplementasikan Quadratic Programming menggunakan metode Wolfe untuk menentukan proporsi portfolio yang akan ditanami modal optimal. Beberapa contoh perhitungan Quadratic Programming Metode Wolfe yang dikerjakan dengan bantuan program komputer ditampilkan dalam makalah ini. Hasil yang didapat menunjukkan tidak ada perbedaan yang signifikan dengan hasil perhitungan secara manual.

Kata kunci: Saham portfolio, Quadratic Programming, Metode Wolfe.

Abstract: Investors generally desire substantial returns from their investments with minimal risk of loss, and they invest their capital in several types of stocks (stock diversification). The problem that then arises is how they determine the proportion of stocks in their portfolio that will provide a large return (profit) with minimal risk. To solve this problem, the author will implement Quadratic Programming using the Wolfe method to determine the optimal portfolio proportion for investment. Several examples of Quadratic Programming calculations using the Wolfe method performed with the help of a computer program are presented in this paper. The results obtained show no significant difference from the results of manual calculations.



DOI: 10.52362/jisamar.v10i3.2498

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).

Keywords: *Portfolio stocks, Quadratic Programming, Wolfe Method.*

I. PENDAHULUAN

Pada umumnya para penanam modal merupakan orang yang tidak suka risiko (*risk averse*) namun biasanya saham yang memberikan keuntungan yang lebih tinggi dari saham lainnya, risikonya juga akan lebih besar. Para penanam modal hanya akan menginvestasikan modalnya dalam saham yang berisiko besar hanya jika penanam modal tersebut yakin saham tersebut akan memberikan keuntungan yang lebih tinggi dari saham lainnya.

Penanam modal seringkali menanam modal mereka ke beberapa jenis *asset*/saham (diversifikasi saham). Hal ini dilakukan untuk menghindari kerugian yang besar karena bertumpu hanya pada satu kondisi. Semua saham-saham yang diperdagangkan dalam pasar saham/bursa saham tidak bergerak bersama. Secara umum pasar bergerak naik, tapi pada saat yang sama ada saham-saham yang mengalami penurunan harga. Ada beberapa saham yang mempunyai kecenderungan untuk bergerak bersama, ada beberapa saham yang bergerak berlawanan, dan ada beberapa saham yang kelihatannya tidak mempunyai hubungan dengan saham yang lain.[1]

Oleh sebab itu diversifikasi saham diperlukan untuk mengurangi risiko dalam sebuah *portfolio*. Jadi apabila nilai saham yang satu mengalami penurunan, maka nilai saham jenis lain bisa saja mengalami peningkatan sehingga *investor* bisa saja tetap mendapatkan keuntungan. Sekumpulan dari saham-saham yang berisiko disebut *portfolio*.

Masalah yang kemudian muncul adalah bagaimana mereka menentukan besarnya proporsi saham-saham dalam *portfolio* yang akan mereka tanami modal sehingga dapat memberikan *return* (keuntungan) yang besar dengan risiko yang kecil. Untuk menyelesaikan masalah tersebut, penulis mengimplementasikan *Quadratic Programming* yang penyelesaiannya menggunakan metode Wolfe untuk menentukan proporsi *portfolio* yang akan ditanami modal. Penulis juga menggunakan Markowitz *mean variance* model yang merupakan dasar dari modern *portfolio theory* untuk menghitung besarnya *expected return* dan standart deviasi untuk mengetahui besar kecilnya risiko dari sebuah *portfolio*, serta kovarians untuk mengetahui hubungan antara saham yang satu dengan yang lainnya.[2]

Dalam contoh perhitungan, penulis mengambil tiga saham yang ada di masyarakat. Saham tersebut antara lain : Dari sektor pertambangan : Tambang Batubara Bukit Asam Tbk (PTBA), dari sektor aneka industri : Indofood Sukses Makmur Tbk (INDF) dan dari sektor keuangan : Bank Central Asia Tbk (BBCA) dan untuk mempermudah dan mempercepat perhitungannya dilakukan dengan bantuan program komputer.

II. METODE DAN MATERI

2.1. Expected Rate of Return

Setiap pengembalian dari investasi yang diharapkan diberi istilah *return*, menurut Penman “*Returns are uncertain –risky- and normal returns are compensation for bearing the risk and for the time value of money*”.[3] Bisa diartikan pengembalian adalah risiko yang bersifat tidak pasti dan pengembalian normal adalah kompensasi dari menerima risiko tersebut dan sebagai nilai waktu dari uang.

Dapat disimpulkan, *return* merupakan keuntungan yang didapat dari investasi yang dilakukan oleh investor. *Return* dapat berupa dividen atau *capital gain*. *Rate of return* adalah skala yang menunjukkan hasil dari investasi atau *return* dari investasi. *Rate of return* mempunyai rumus sebagai berikut [4].

$$\text{Rate of return} = \frac{\text{amount received} - \text{amount invested}}{\text{amount invested}}$$

Expected rate of return adalah jumlah hasil kali probabilitas kejadian dengan *rate of return*, ditulis :

$$r = \sum_{i=1}^n r_i P_i = P_1(r_1) + P_2(r_2) + \dots + P_n(r_n)$$

r_i = *rate of return stock-i*

P_i = probabilitas terjadinya *rate of return stock-i*

Jika data ditampilkan dalam bentuk *array of return*, dimana setiap komponen i dari array ini adalah *expected return* dari *asset i* maka dapat ditulis sebagai berikut: $r = (r_1, r_2, \dots, r_n)$. Dalam menentukan *expected rate of*



DOI: 10.52362/jisamar.v10i3.2498

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).

return bisa juga mengambil data historis. Jadi dengan melihat data-data dari masa yang lalu dapat diharapkan nilai di masa yang akan datang.

2.2. Portfolio Returns

Portfolio Returns merupakan *return* dari sekumpulan *asset* (portfolio). Jika investor mempunyai dana sebesar x yang akan dialokasikan ke sejumlah *asset*, maka dapat dinyatakan : $x = (x_1, x_2, \dots, x_n)$

dimana x_i adalah jumlah dana yang dialokasikan ke *asset* i dan $\sum_{i=1}^n x_i = 1$, $x_i \geq 0$; n = banyak *asset*.

$$\text{Portfolio return} = \sum_{i=1}^n x_i r_i = r_p$$

dimana r_i = *expected return* dari *asset* ke – i

n = jumlah saham dalam portfolio

x_i = proporsi pengalokasian dana pada *asset* ke – i .

2.3. Risiko (Risk)

Untuk menentukan besar kecilnya risiko, kita dapat menggunakan standar deviasi. Standar deviasi (σ) merupakan akar dari varians. Varians adalah ukuran sebaran data.

$$\text{Varians (untuk 1 asset)} = \sigma^2 = \sum_{i=1}^n \frac{(r_i - \bar{r})^2}{n - 1}$$

Dimana i = *asset* ke- i

n = banyak *asset* ; r_i = *expected return asset* i

$\bar{r}$ = rata-rata *expected return* keseluruhan saham/*asset*.

$$\text{Varians (portfolio)} = \sigma_p^2 = \sum_{i=1}^n \sum_{j=1}^n x_i x_j \sigma_{ij}$$

Dimana x_i = proporsi saham i dalam *portfolio*

x_j = proporsi saham j dalam *portfolio*

n = jumlah saham dalam *portfolio*

σ_{ij} = kovarians antara saham i dan j

Dua saham yang *return*nya bergerak berlawanan, dalam kombinasi, akan menghasilkan nilai varians *portfolio* yang lebih rendah daripada varians dari saham individual. Jika *portfolio* mempunyai standar deviasi yang tinggi menandakan variasi dari *return* yang besar maka peluang *return* yang sebenarnya mungkin lebih kecil *expected return* menjadi lebih besar, jadi risiko dari *portfolio* juga tinggi.[5]

2.4. Quadratic Programming

Model *Quadratic Programming* ditujukan sebagai berikut [6].

Maximize $z = CX + X^TDX$

Dengan batasan

$$AX \leq b, X \geq 0$$

dimana

$$X = (x_1, x_2, \dots, x_n)^T$$

$$C = (c_1, c_2, \dots, c_n)$$

$$b = (b_1, b_2, \dots, b_n)^T$$



DOI: 10.52362/jisamar.v10i3.2498

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).

$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \quad ; \quad \mathbf{D} = \begin{pmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & & \vdots \\ d_{m1} & \cdots & d_{mn} \end{pmatrix}$$

Fungsi $\mathbf{X}^T \mathbf{D} \mathbf{X}$ menunjukkan bentuk kuadratik dimana $\mathbf{D}$ adalah simetris. Matriks $\mathbf{D}$ diasumsikan definit negatif jika masalahnya adalah maksimasi, dan definit positif jika masalahnya adalah minimasi. Hal ini berarti z konveks dalam $\mathbf{X}$ untuk minimasi dan konkaf untuk maksimasi. Batasan-batasan diasumsikan linear dalam kasus ini, yang menjamin tempat solusi konveks. Solusi untuk masalah ini berdasarkan KKT *necessary conditions*. Masalah *quadratic programming* akan diberlakukan untuk kasus maksimasi, mengubah formulasi untuk minimasi merupakan hal yang mudah. Masalahnya dapat ditulis sebagai berikut:[7].

$$\text{Maximize } z = \mathbf{C} \mathbf{X} + \mathbf{X}^T \mathbf{D} \mathbf{X}$$

Dengan batasan

$$\mathbf{G}(\mathbf{X}) = \begin{pmatrix} \mathbf{A} \\ -\mathbf{I} \end{pmatrix} \mathbf{X} - \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix} \leq 0$$

Dan $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)^T$
 $\mathbf{U} = (\mu_1, \mu_2, \dots, \mu_n)^T$

merupakan Lagrange Multipliers yang sesuai dengan dua set batasan $\mathbf{A} \mathbf{X} - \mathbf{b} \leq 0$ dan $-\mathbf{X} \leq 0$. Aplikasi dari KKT menghasilkan :

$$\begin{aligned} \lambda &\geq 0, \mathbf{U} \geq 0 \\ \nabla z - (\boldsymbol{\lambda}^T, \mathbf{U}^T) \nabla \mathbf{G}(\mathbf{X}) &= 0 \\ \lambda_i \left(b_i - \sum_{j=1}^n a_{ij} x_j \right) &= 0, \quad i = 1, 2, \dots, m \\ \mu_j x_j &= 0, \quad j = 1, 2, \dots, n \\ \mathbf{A} \mathbf{X} &\leq \mathbf{b} \\ -\mathbf{X} &\leq 0 \end{aligned}$$

sekarang

$$\begin{aligned} \nabla z &= \mathbf{C} + 2\mathbf{X}^T \mathbf{D} \\ \nabla \mathbf{G}(\mathbf{X}) &= \begin{pmatrix} \mathbf{A} \\ -\mathbf{I} \end{pmatrix} \end{aligned}$$

Biarkan $\mathbf{S} = \mathbf{b} - \mathbf{A} \mathbf{X} \geq 0$ menjadi *slack variables* atas batasan-batasan. Kondisi-kondisi dikurangi menjadi:

$$\begin{aligned} -2\mathbf{X}^T \mathbf{D} + \boldsymbol{\lambda}^T \mathbf{A} - \mathbf{U}^T &= \mathbf{C} \\ \mathbf{A} \mathbf{X} + \mathbf{S} &= \mathbf{b} \\ \mu_j x_j = 0 = \lambda_i S_i &\text{ semua } i \text{ dan } j \\ \boldsymbol{\lambda}, \mathbf{U}, \mathbf{X}, \mathbf{S} &\geq 0 \end{aligned}$$

Karena $\mathbf{D}^T = \mathbf{D}$, transpose dari set pertama atas persamaan-persamaan dapat ditulis sebagai

$$-2\mathbf{D} \mathbf{X} + \mathbf{A}^T \boldsymbol{\lambda} - \mathbf{U} = \mathbf{C}^T$$

Maka, kondisi-kondisi yang diperlukan dapat dikombinasikan sebagai

$$\begin{aligned} \begin{pmatrix} -2\mathbf{D} & \mathbf{A}^T & -\mathbf{I} & 0 \\ \mathbf{A} & 0 & 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ \boldsymbol{\lambda} \\ \mathbf{U} \\ \mathbf{S} \end{pmatrix} &= \begin{pmatrix} \mathbf{C}^T \\ \mathbf{b} \end{pmatrix} \\ \mu_j x_j = 0 = \lambda_i S_i &\text{ semua } i \text{ dan } j \\ \boldsymbol{\lambda}, \mathbf{U}, \mathbf{X}, \mathbf{S} &\geq 0 \end{aligned}$$



Kecuali untuk kondisi-kondisi $\mu_j x_j = 0 = \lambda_i S_i$ semua i dan j , persamaan-persamaan yang tersisa adalah persamaan linear dalam λ , U , X , S . Karena z konfak dan tempat solusinya adalah konveks, solusi *feasible* yang memenuhi semua kondisi harus unik dan optimum. [8]

2.5. Metode Wolfe

Untuk bentuk dari *Quadratic Programming* yang umum

$$\text{Minimize } z = \frac{1}{2} x^T C x + p^T x$$

Dengan $Ax = b$ dan $x \geq 0$

Dengan $A = (a_{ij})$ merupakan matrix $m \times n$, $b = (b_i)$ merupakan vektor $m \times 1$, $x = (x_{ij})$ dan $p = (p_j)$ keduanya merupakan vektor $n \times 1$. Matrix C merupakan matrix $n \times n$.

Diasumsikan matriks C simetris ($C^T = C$) merupakan positif definit, yaitu $x^T C x \geq 0$ untuk setiap $n \times 1$ vektor x .

Metode Wolfe untuk menyelesaikan *Quadratic Programming* merupakan variasi dari metode simplex untuk *linear programming*. Teorema : x^* merupakan solusi optimal untuk *Quadratic Programming* jika dan hanya jika terdapat $m \times 1$ vektor u^* dan $n \times 1$ vektor v^* sehingga (x^*, u^*, v^*) yang menyelesaikan sistem persamaan :

$$\begin{aligned} Ax &= b \\ Cx + A^T u - v &= -p \\ x^T v &= 0 \\ x &\geq 0, v \geq 0 \end{aligned}$$

Metode Wolfe menyatakan untuk menyelesaikan masalah *Quadratic Programming* sama halnya dengan menyelesaikan system persamaan diatas. Persamaan-persamaan diatas merupakan system persamaan linear dan untuk menyelesaikan persamaan-persamaan tersebut dapat digunakan algoritma simplex.

Persamaan $Ax = b$ dapat ditulis sebagai berikut [9].

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &= b_1 \\ a_{m1}x_1 + \dots + a_{mn}x_n &= b_m \end{aligned}$$

Dapat ditambahkan variabel buatan $d = (d_1, d_2, \dots, d_m)$ sehingga persamaan linearnya menjadi :

$$\begin{aligned} \text{Minimize } Z &= d_1 + d_2 + \dots + d_m \\ \text{Dan} \\ a_{11}x_1 + \dots + a_{1n}x_n \pm d_1 &= b_1 \\ a_{m1}x_1 + \dots + a_{mn}x_n \pm d_m &= b_m \\ x &\geq 0, d \geq 0 \end{aligned}$$

Yang dimaksud dengan $\pm d_i$ adalah $+d_i$ jika $b_i \geq 0$, dan $-d_i$ jika $b_i < 0$. *Initial Basic Feasible Solution* nya adalah $x = 0$ dan $d_i = |b_i|$. Jika sistem persamaan $Ax = b$, $x \geq 0$ mempunyai solusi, nilai optimal Z^* akan sama dengan 0, sebaliknya, nilai optimal akan menjadi positif. Algoritma Simplex akan menyelesaikan persamaan-persamaan diatas. Tetapi karena ada persamaan $x^T v = 0$, maka persamaan tersebut akan digunakan sebagai aturan pengecualian, yaitu x_j dan v_j keduanya tidak boleh positif pada saat yang sama untuk setiap $j = 1, \dots, n$. Dengan kata lain, pada setiap tahap dari algoritma simplex untuk menyelesaikan persamaan-persamaan diatas, jika x_j berada dalam basis lama ($x_j > 0$) maka v_j tidak boleh masuk ke dalam basis. Jika v_j berada dalam basis lama ($v_j > 0$), maka x_j tidak boleh masuk ke dalam basis.

III. PEMBAHASAN DAN HASIL

3.1 Pembahasan

Untuk membantu proses perhitungannya telah dilakukan perancangan program komputer yang dapat dilihat pada gambar 1. Dari bentuk *flowchart* dapat dibuat sebuah program komputer



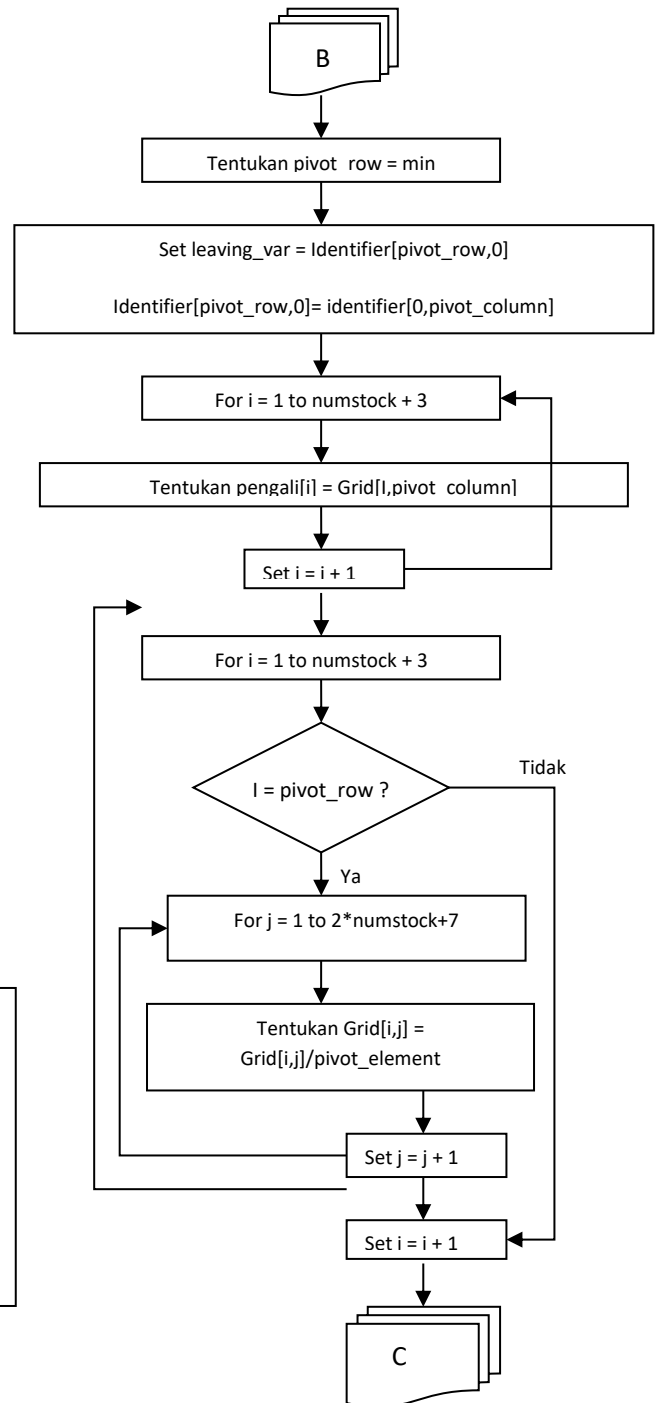
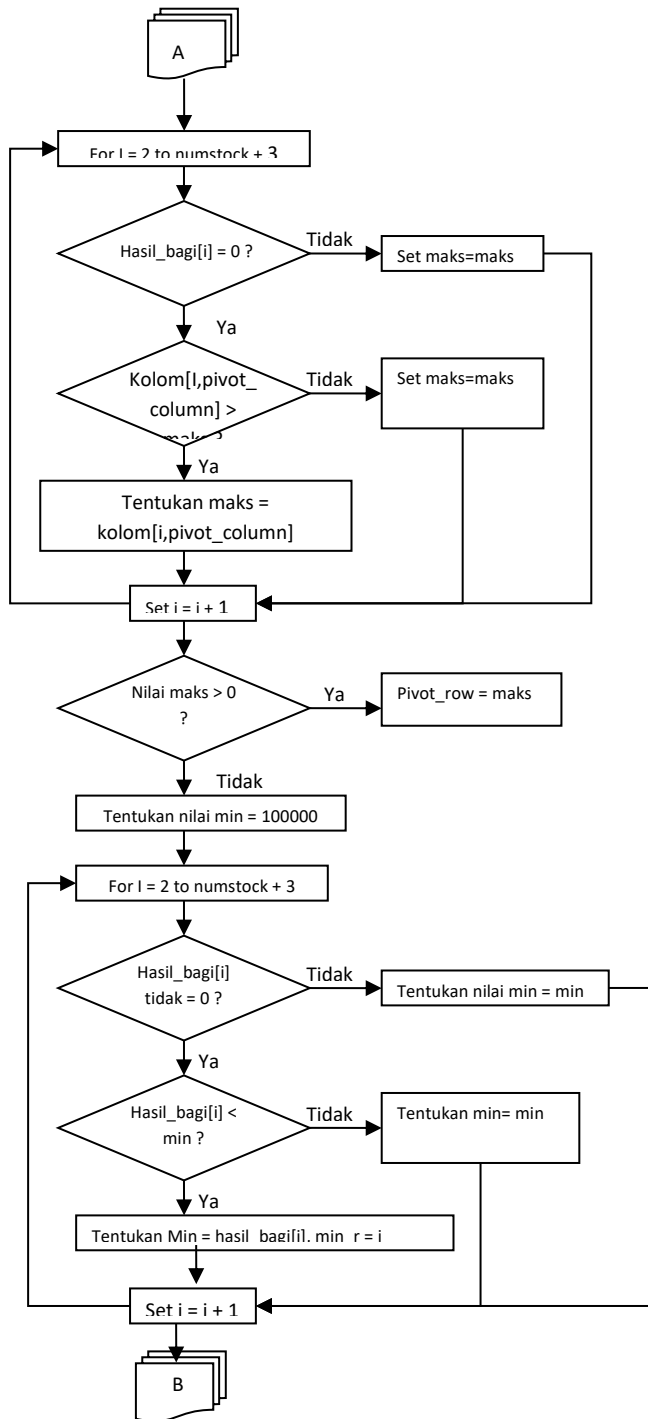
DOI: 10.52362/jisamar.v10i3.2498

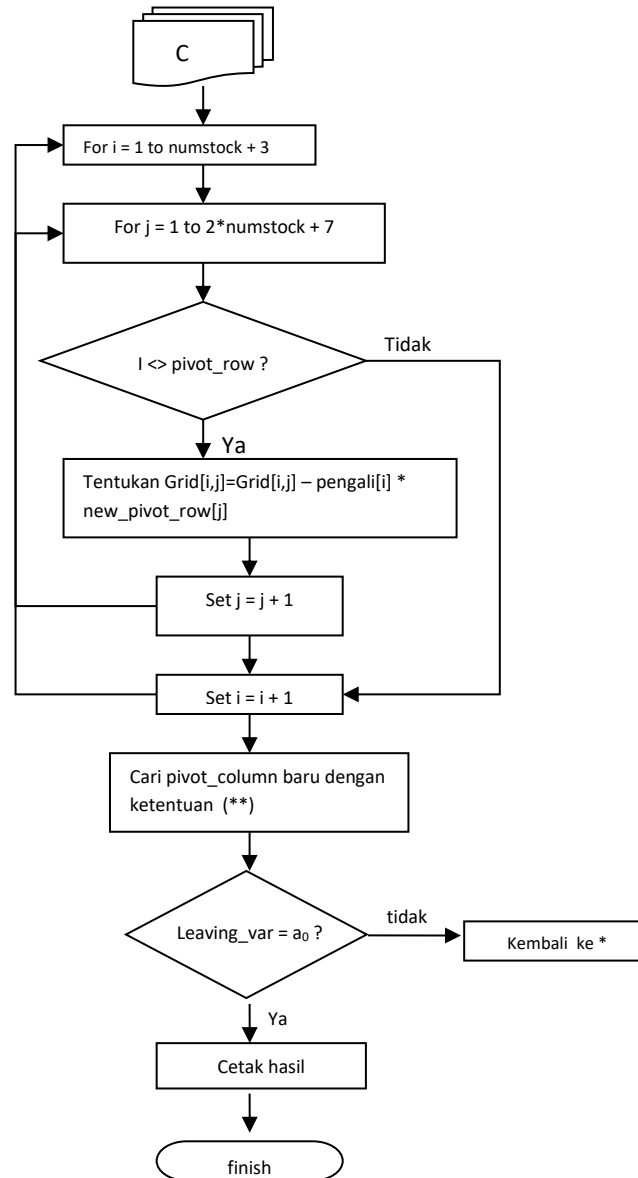
Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



DOI: 10.52362/jisamar.v10i3.2498

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).





Gambar 1. Flowchart Program Penentuan Proporsi Saham Portfolio

3.2 Hasil

2.1 Layar Menu

Bila aplikasi dijalankan maka pertama kali ditampilkan Layar *Menu* (gambar 2). ini berfungsi sebagai layar pembuka dan pengantar menuju ke layar-layar selanjutnya.



DOI: 10.52362/jisamar.v10i3.2498

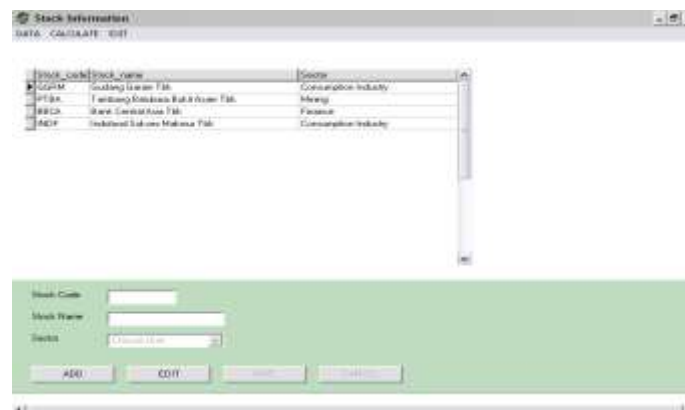
Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



Gambar 2 Layar Menu

2.2 Layar Stock Information

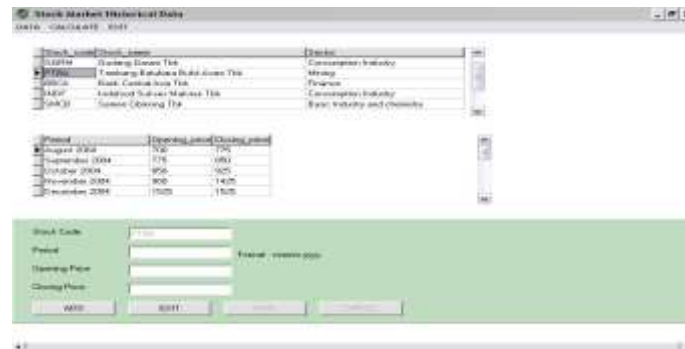
Layar *Stock Information* (gambar 3) akan ditampilkan setelah user memilih menu *Stock Information* pada layar Menu. Layar ini menampilkan nama-nama saham yang sudah tersimpan di dalam database beserta kode saham dan sektor saham masing-masing. Tombol 'save' dan 'cancel' tidak diaktifkan, sedangkan tombol 'add' dan 'edit' aktif. Tombol 'save' dan 'cancel' hanya akan aktif jika user telah menekan tombol 'add' atau 'edit' untuk melakukan penambahan data baru atau perubahan data lama.



Gambar 3. Layar Stock Information

2.3 Layar Stock Market Historical Data

Layar *Stock Market Historical Data* (gambar4) akan ditampilkan jika user memilih menu *Stock Market Historical Data* dari layar Menu atau dari layar Stock Information. Layar ini menampilkan informasi harga masing-masing saham sesuai dengan periode waktu. Pada saat layar pertama kali dibuka, tombol 'add', 'edit', 'save', dan 'cancel' tidak ada yang aktif. Pada saat user meng-click isi pada DBGrid1, yaitu tabel yang berisi nama-nama saham, maka pada DBGrid2 akan muncul data harga untuk masing-masing periode sesuai dengan nama saham yang dipilih user. Pada saat itu, tombol 'save' dan 'cancel' tidak diaktifkan, sedangkan tombol 'add' dan 'edit' aktif. Sama seperti layar *Stock Information*, tombol 'save' dan 'cancel' akan aktif jika tombol 'add' atau 'edit' ditekan.



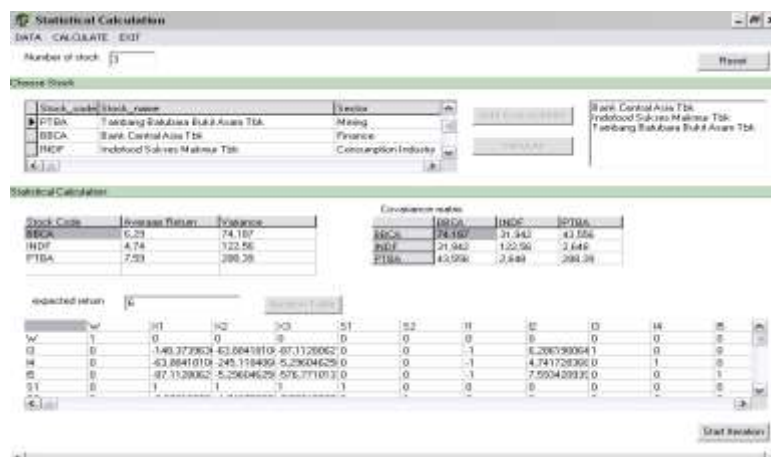
Gambar 4. Layar Stock Market Historical Data

2.4 Layar Statistical Calculation

Layar ini (gambar 5) akan ditampilkan jika menu *Statistical Calculation* dipilih dari menu bar, baik menu bar yang ada di layar menu, layar *Stock Information*, atau layar *Stock Market Historical Data*. Layar ini menampilkan hasil perhitungan statistik terhadap data harga saham-saham seperti *mean*, *varians*, *kovarians*, serta menampilkan tabel iterasi awal.

Mula-mula *user* diminta untuk menentukan berapa banyak saham yang hendak dimasukkan dalam *portfolio*. Setelah itu *user* bisa memilih nama-nama saham yang ingin dimasukkan ke dalam *portfolio*. Data nama-nama saham beserta sektornya disajikan dalam bentuk tabel. Bila *user* ingin mengambil salah satu nama saham, *user* hanya perlu meng-click nama harga saham yang dimaksud di dalam tabel, lalu menekan tombol 'add to my portfolio'.

User bisa melihat saham-saham apa saja yang sudah dipilih pada kotak Memo. Setiap saham yang dipilih akan langsung dihitung *average return* dan *variance* nya dan hasilnya akan ditampilkan pada StringGrid1. Jika *user* telah selesai memilih dan siap melakukan perhitungan *kovarians*, maka *user* hanya perlu menekan tombol 'calculate'. Tombol 'Iteration Table' berfungsi untuk menampilkan tabel iterasi pertama kali. Tombol 'Start Iteration' berfungsi untuk melanjutkan iterasi atau perhitungan dengan Metode Wolfe dan mengantar *user* ke layar *Final Calculation*. Pada awalnya semua tombol ini tidak aktif, tombol 'add to my portfolio' akan aktif setelah *user* meng-click nama saham yang hendak dimasukkan ke dalam perhitungan. Setelah selesai memilih, tombol 'add to my portfolio' menjadi tidak aktif, dan tombol 'calculate' menjadi aktif. Setelah tombol 'calculate' ditekan, tombol 'iteration table' menjadi aktif dan setelah tombol 'iteration table' dijalankan tombol 'start iteration' menjadi aktif.

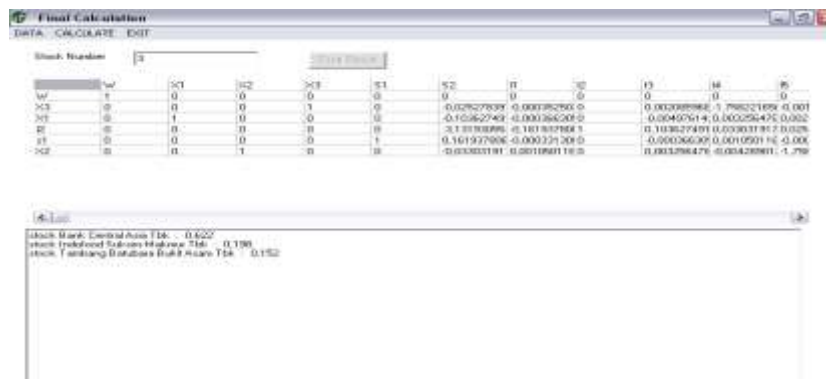


Gambar 5. Layar Statistical Calculation

2.5 Layar Final Calculation

Layar ini (gambar 6) ditampilkan bila tombol 'Start Iteration' pada layar *Statistical Calculation* ditekan. Layar ini menampilkan hasil perhitungan dengan menggunakan Metode Wolfe dengan melakukan iterasi sampai didapatkan nilai proporsi untuk setiap saham.





The screenshot shows a window titled "Final Calculation" with a menu bar (DATA, CALCULATE, EXIT). It contains a table with columns labeled X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15. The table contains numerical data. Below the table, there is a summary section with the following text:

```

Akhir Mark Central Area Tdk : 0.622
Akhir Mark Central Area Tdk : 0.198
Akhir Mark Central Area Tdk : 0.152
  
```

Gambar 6. Layar *Final Calculation*

3.3 Analisis

Seperti pada perhitungan yang telah dilakukan secara manual, program juga menghitung besarnya proporsi investasi untuk saham BBKA, INDF, dan PTBA dengan *expected return* 6%. Hasil yang didapat dari perhitungan secara manual adalah proporsi investasi pada saham BBKA sebesar 0,631 dari 1 juta rupiah, pada saham INDF sebesar 0,182 dari 1 juta rupiah dan pada saham PTBA sebesar 0,154 dari 1 juta rupiah. Sedangkan dengan melakukan perhitungan menggunakan program didapat hasil proporsi investasi pada saham BBKA sebesar 0,622 dari 1 juta rupiah, pada saham INDF sebesar 0,198 dari 1 juta rupiah, dan pada saham PTBA sebesar 0,152 dari 1 juta rupiah.

Pada kedua hasil tersebut dapat dilihat perbedaan angka yang didapat dari hasil perhitungan secara manual dan perhitungan dengan menggunakan program. Pada saham BBKA terdapat perbedaan hasil sebesar 0,009. Pada saham INDF terdapat perbedaan hasil sebesar 0,016. Sedangkan pada saham PTBA terdapat perbedaan hasil sebesar 0,002. Perbedaan hasil tersebut disebabkan karena pada perhitungan manual, banyak dilakukan pembulatan dan dalam menghitung *return*, *average return*, *varians*, dan *kovarians*, hanya dua angka dibelakang koma yang diambil untuk diiterasikan menurut Metode Wolfe. Sedangkan pada program ini tidak dilakukan pembulatan dan semua angka dibelakang koma diambil sampai hasil akhir, baru dilakukan pembulatan. Dapat disimpulkan juga bahwa perbedaan angka yang terjadi tidak signifikan, hal ini memberikan keuntungan bagi *user*, yaitu efisiensi dalam segi ketelitian dan kecepatan perhitungan.

Pada program juga diujicobakan perhitungan saham BBKA, INDF, dan PTBA dengan *expected return* sebesar 7%. Tetapi ternyata iterasi yang dilakukan oleh komputer tidak pernah berhenti sehingga menimbulkan *floating point*. Untuk mengatasi hal ini, pada program ditambahkan *counter* yang berfungsi untuk menghitung berapa kali iterasi yang sudah dilakukan oleh komputer. Bila *counter* telah mencapai angka 1000 yang berarti komputer telah melakukan 1000 kali iterasi, maka perhitungan akan dihentikan dan dianggap hasil perhitungan tidak ditemukan.

Penyebab iterasi yang tidak pernah berhenti dapat disebabkan oleh nilai *expected return* yang dimasukkan oleh *user* terlalu besar. Misalnya *return* dari saham BBKA sebesar 6,29 %, *return* dari saham INDF sebesar 4,06 %, *return* dari saham PTBA sebesar 7,59 % maka nilai *return* paling sedikit yang diharapkan oleh *user* harus berada dibawah 7, karena nilai *return* yang dimasukkan adalah nilai yang diharapkan paling sedikit tercapai, bukan nilai *return* terbesar yang diharapkan *user*. Maka *user* juga harus rasional dalam memasukkan nilai *expected return* yang diharapkan.

IV. KESIMPULAN

Setelah melakukan analisis pada program aplikasi pembentukan *portfolio* dengan *Quadratic Programming* metode Wolfe ini, penulis mengambil beberapa kesimpulan bahwa dengan program ini dapat diketahui hubungan antar saham serta *return* tiap saham dengan waktu proses yang efisien. Kemudian dengan *Quadratic Programming* metode Wolfe ini *investor* dapat menentukan proporsi investasi saham-saham dalam *portfolio* yang dapat mengoptimalkan *return* dan meminimalkan risiko. Demikian juga dengan program ini, *investor* dapat dengan mudah melakukan diversifikasi saham dan menentukan proporsi investasi untuk masing-masing saham yang telah dipilih, namun dalam program ini tidak menghitung *return* yang akan dicapai oleh *user* jika menginvestasikan dana sesuai dengan proporsi hasil perhitungan.



REFERENSI

- [1] Tandelilin, E., *Analisis Investasi dan Manajemen Portofolio*. ,Yogyakarta, BPPEE, 2020.
- [2] Markowitz. H., “Portfolio Selection”, *The Journal of Finance*, vol 7(1) 1982,, pp77–91.
- [3] Frederick S. Hillier, G J. Lieberman, *Introduction to Operations Research*, USA, McGraw-Hill, 2022.
- [4] Afrilia Putri Wulandari dkk, “Penentuan Solusi Optimal Pemrograman Kuadratik Metode Wolfe”, *Buletin Ilmiah Math. Stat. dan Terapannya (Bimaster)*, Volume 14, No. 2 (2025), hal 245–252.
- [5] Dias, Fabio Silva., “Mathematical Programming In Portfolio Optimization: A Three-Part Review Of Classical, Advanced, And Emerging Techniques”, *IOSR Journal Of Economics and Finance (IOSR-JEF)* e-ISSN: 2321-5933, p-ISSN: 2321-5925. Volume 16, Issue 5 Ser. 5 (Sept. – Oct. 2025), pp 19-32
- [6] Natalia, E., Darmanto, E., & MG, W. “Penentuan Portofolio Saham Yang Optimal Dengan Model Markowitz Sebagai Dasar Penetapan Investasi Saham”. *Jurnal Administrasi Bisnis (JAB)*. vol. 9, no. 1, 2014, pp. 1–9.
- [7] Putri, R. H., Sudarwanto, S., & Wiraningsih, W, Analisis Penentuan Proporsi Portofolio Optimal pada Program Kuadratik dengan Menggunakan Metode Wolfe. In *PRISMA, Prosiding Seminar Nasional Matematika*, Vol. 5, 2021, pp. 657-666.
- [8] Render, Barry.,Ralph M. Stair Jr., Michael E.Hanna, *Quantitative Analysis for Management*, Eight edition, New Jersey, Prentice Hall, Upper Saddle River, 2023.
- [9] Taylor III, Bernard W, *Introduction to Management Science*, Seventh Edition,New Jersey, Prentice Hall, Upper Saddle River, 2022.

