Optimasi Masalah *Convex Quadratic Programming* dengan Metode *Primal Dual Interior Point*

Nurwahidah
Universitas Islam Negeri Alauddin Makassar, nurwahidah.abidin@uin-alauddin.ac.id

ABSTRAK, *Quadratic programming* adalah permasalahan optimasi untuk memaksimumkan atau meminimumkan fungsi tujuan kuadratik dan kendala berbentuk linear. *Quadratic programming* dapat diselesaikan menggunakan metode simpleks, hanya saja metode *interior point* lebih efisien dalam menemukan titik ekstrim yang optimal. Metode *primal dual* merupakan metode *interior point* untuk masalah *quadratic programming* yang lebih efektif dalam penyelesaian masalah skala besar atau lebih dari 2 variabel.

***Kata Kunci**: optimasi, quadratic programming, primal dual, interior pont*

1. PENDAHULUAN

Dalam kehidupan sehari-hari, manusia cenderung untuk ingin memperoleh keuntungan semaksimal mungkin dan meminimalisir kerugian. Hal ini berarti manusia melakukan optimasi dalam memenuhi keperluan hidupnya. Akan tetapi, masyarakat awam lebih banyak menggunakan intuisi dalam menerapkan konsep optimasi. Kasus optimasi dapat dirumuskan menggunakan model matematika. Optimasi diartikan sebagai sebuah usaha dalam mencari atau menentukan nilai maksimum atau minimum dari suatu fungsi. Dengan kata lain, optimasi merupakan penentuan nilai terbaik yang didasarkan pada fungsi tujuan dengan daerah asal yang terdefinisi.

Suatu masalah optimasi disebut non linear apabila salah satu atau keduanya dari fungsi tujuan dan fungsi kendala berbentuk non linear[1]. Optimasi non linear merupakan suatu konsep yang membutuhkan pengetahuan terkait kalkulus dan aljabar linear. Selain itu, penyelesaian masalah optimasi non linear juga membutuhkan algoritma numerik. Optimasi non linear memerlukan metode penentuan nilai optimal global agar tidak terjebak dalam pencapaian nilai optimal local [2].

Quadratic programming merupakan sebuah teknik dalam menyelesaikan masalah optimasi non linear dengan fungsi tujuan kuadratik dan kendala berbentuk linear. *Quadratic programming* dapat diterapkan pada berbagai masalah dalam bidang industri dan keuangan [5]. *Quadratic programming* dapat diselesaikan menggunakan metode simpleks, hanya saja metode *interior point* lebih efisien dalam menemukan titik ekstrim yang optimal. Metode *interior point* yang dapat menyelesaikan masalah *quadratic programming* terdiri atas metode *primal affine scaling* dan *primal dual interior point*. Metode *primal affine scaling* merupakan *metode interior point* ynag paling sederhana, sedangkan *primal dual* merupakan metode *interior point* yang lebih efektif dari metode simpleks untuk penyelesaian masalah skala besar.

Penelitian ini memaparkan penyelesaian masalah program non linear kuadratik untuk fungsi konveks dengan metode *primal dual interior point* Penelitian ini dibuat setelah dilakukan kajian literatur dari beberapa buku dan artikel ilmiah yang telah terbit. Tujuan dari penelitian ini adalah agar pembaca dapat membentuk model matematika dari masalah optimasi non linear. Penelitian ini diharapkan dapat menjadi salah satu alat bantu dalam studi terkait persoalan alokasi sumber-sumber secara optimal dan mencari keuntungan maksimum masalah optimasi non linear dengan metode *primal dual interior point*.

2. TINJAUAN PUSTAKA

Fungsi Konveks

Holder Stolz dan *Hadamard* memperkenalkan fungsi konveks pada tahun 1889-1893. Fungsi konveks diterapkan pada ketaksamaan Hadamard pada tahun 1893 [3]. Bentuk umum fungsi konveks adalah sebagai berikut:

$$f\big((1-t)x_1 + tx_2\big) \leq (1-t)f(x_1) + tf(x_2) \quad (2.1)$$

dimana $\forall\ 0 \leq t \leq 1$ dan x_1, x_2 merupakan sebarang anggota dari daerah asal f. Fungsi konveks merupakan fungsi dengan karakteristik garis yang menghubungkan titik $(x_1, f((x_1))$ dan $(x_2, f((x_2))$ selalu terletak di atas fungsi f.

Metode Pengali Lagrange

Metode Pengali Lagrange sering digunakan pada penyelesaian optimasi dengan kendala tertentu. Prinsip metode ini adalah mencari nilai optimal suatu fungsi tujuan f (x, y) dengan kendala tertentu, yaitu g (x, y) = 0. Fungsi Lagrange dapat dituliskan sebagai berikut:

$$F(x, y, \lambda) = f(x, y) + \lambda g(x, y) \qquad (2.2)$$

dengan syarat ekstrim:

$$\frac{\partial F}{\partial x} = 0, \frac{\partial F}{\partial y} = 0, dan\ \frac{\partial F}{\partial \lambda} = 0$$

Dimana λ adalah pengali Lagrange.

Metode pengali Lagrange dengan lebih dari satu fungsi kendala dapat ditambahkan parameter menjadi λ, μ atau parameter lain. Misal untuk memperoleh nilai ekstrim $f(x, y, z)$ dengan batasan $g(x, y, z) = 0$ dan $h(x, y, z) = 0$, maka fungsi Lagrange yang dibentuk adalah:

$$F(x, y, z, \lambda, \mu) = f(x, y, z) + \lambda g(x, y, z) + \mu h(x, y, z). \qquad (2.3)$$

Solusinya adalah dengan mencari penyelesaian dari

$$\frac{\partial F}{\partial x} = 0, \frac{\partial F}{\partial y} = 0, \frac{\partial F}{\partial z} = 0, \frac{\partial F}{\partial \lambda} = 0, \frac{\partial F}{\partial \mu} = 0.$$

Metode ini dapat diperluas untuk n variabel $f(x_1, x_2, \dots, x_n)$ dengan k kendala $\phi_1(x_1, x_2, \dots, x_n), \phi_2(x_1, x_2, \dots, x_n), \dots, \phi_k(x_1, x_2, \dots, x_n)$. Fungsi Lagrange yang dibentuk adalah:

$$F(x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_n) = f + \lambda_1\phi_1 + \lambda_2\phi_2 + \dots + \lambda_k\phi_k. \qquad (2.4)$$

Solusinya adalah dengan mencari penyelesaian dari:

$$\frac{\partial F}{\partial x_1} = 0, \frac{\partial F}{\partial x_2} = 0, \dots, \frac{\partial F}{\partial x_n} = 0, \frac{\partial F}{\partial \lambda_1} = 0, \frac{\partial F}{\partial \lambda_2} = 0, \dots, \frac{\partial F}{\partial \lambda_n} = 0$$

dimana $\lambda_1, \lambda_2, \dots, \lambda_n$ adalah pengali Lagrange.

Quadratic Programming

Quadratic Programming (QP) atau pemrograman kuadrat adalah permasalahan optimasi untuk memaksimumkan atau meminimumkan fungsi tujuan yang berbentuk kuadrat dengan fungsi kendala berbentuk persamaan atau pertidaksamaan linear. Masalah *quadratic programming* secara umum dapat dituliskan sebagai berikut:

$$(QP)\begin{cases} min_x & \frac{1}{2}x^T\Sigma x + c^T x \\ s.t. & Ax = b \\ & x \geq 0 \end{cases} \qquad (2.5)$$

Berdasarkan masalah (2.1), akan ditentukan nilai x yang dapat meminimalkan fungsi tujuan $\frac{1}{2}x^T\Sigma x + c^T x$ dengan kendala $Ax = b$ dan $x \geq 0$.

Keterangan:

x : Vektor variabel optimasi $(n \times 1)$

A : Matriks koefisien kendala $(n \times m)$

b : Vektor kendala ruas kanan $(m \times 1)$

c : Vektor yang memuat koefisien suku-suku linear pada fungsi kendala $(n \times 1)$

Σ : Matriks Hessian, matriks dengan koefisien dari suku berpangkat dua atau suku campuran dari fungsi tujuan $(n \times n)$.

Fungsi tujuan dari masalah *quadratic programming* merupakan fungsi konveks apabila Σ merupakan matriks *positive semidefinite* atau $x^T\Sigma x \geq 0$ untuk setiap x. Dengan demikian, ketika Σ merupakan matriks *positive semidefinite*, maka masalah (2.5) merupakan masalah optimasi konveks, dimana solusi optimal lokal juga merupakan solusi optimal global.

Bentuk Primal dan Dual

Masalah (2.5) dapat disebut berbentuk primal. Setiap bentuk primal akan memiliki "bayangan" yang disebut bentuk dual. Penyelesaian permasalahan optimasi dapat dilakukan tanpa melibatkan bentuk dual. Keberadaan dual dapat memudahkan penyelesaian masalah, karena bentuk dual biasanya lebih sederhana daripada primalnya.

Sistem (2.1) juga dapat diubah ke bentuk dual, sebagai berikut:

$$(QD)\begin{cases} max_{x,v,u} & -\frac{1}{2}x^T\Sigma x + v^T b \\ s.t. & \Sigma x + c - u - A^T v = 0 \\ & u_i \geq 0, i = 1,2, \dots n \end{cases} \qquad (2.6)$$

dimana v merupakan vektor pengali Lagrange yang bersesuaian dengan kendala persamaan dan $u \geq 0$ merupakan vektor pengali Lagrange dengan kendala positif.

Kondisi Karush-Kuhn-Tucker (KKT)

Kondisi optimal penyelesaian *quadratic programming* akan tercapai jika memenuhi kondisi KKT. Jika dihadapkan pada masalah optimasi dalam bentuk:

max/min $Z = f(x), X = \{x_1, x_2, \dots, x_n\}$

s.t $g_i(x) \leq 0$ atau

$g_i(x) \geq 0, i = 1,2, \dots, m$

$X \geq 0$

$m \leq n$

Langkah pertama adalah menuliskan kembali fungsi kendala yang memiliki tanda lebih kecil atau sama dengan, kemudian menambah variabel *slack* berturut-turut pada ruas kiri kendala sehingga setiap ketidaksamaan menjadi suatu kesamaan. Variabel surplus yang dikurangkan dari model matematik kendala untuk mengkonversi pertidaksamaan ≥ menjadi persamaan (=), sedangkan penambahan *slack* ke model matematik kendala dengan mengkonversikan pertidaksamaan ≤ menjadi persamaan (=).

Teorema 1 memberikan kondisi Kuhn-tucker untuk memecahkan persoalan linear berkendala.

Teorema 1, *Diasumsikan* $f(x)$, $g_1(x)$, $g_2(x)$, $\dots, g_n(x)$ *merupakan fungsi yang dapat diturunkan, maka* $x^* = (x_1', x_2', \dots, x_n')$ *menjadi solusi optimal untuk permasalahan pemrograman nonlinear masalah minimasi hanya jika terdapat sejumlah m bilangan* $v_1, v_2, \dots, v_m$ *sehingga semua syarat kondisi KKT berikut terpenuhi*:

(i) $\frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} v_i \frac{\partial g}{\partial x_j} \geq 0, x = x^*,$ $j = 1,2, \dots, n$

(ii) $x_j \left(\frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} v_i \frac{\partial g}{\partial x_j} \right) = 0, x = x^*$ $j = 1,2, \dots, n$

(iii) $\frac{\partial f}{\partial v_i} + \sum_{i=1}^{m} v_i \frac{\partial g}{\partial v_i} \leq 0, i = 1,2, \dots, m$

(iv) $v_i g_i(x) = 0, i = 1,2, \dots, m$

(v) $x_j \geq 0, j = 1,2, \dots, n$

(vi) $v_i \geq 0, i = 1,2, \dots, m$

Kondisi Optimal *Quadratic Programming*

Fungsi Lagrange untuk masalah *quadratic programming* (2.1) adalah sebagai berikut:

$$L(x, v) = cx + \frac{1}{2} x^T \Sigma x + v^T (b - Ax)$$

Dimana v adalah pengali Lagrange. Kondisi KKT *quadratic programming* untuk minimum lokal memenuhi:

$\frac{\partial L}{\partial x_j} \geq 0, j = 1, \dots, n$ maka

$c + x^T \Sigma - vA \geq 0$ (2.7)

$\frac{\partial L}{\partial v_j} \leq 0$, maka $b - Ax \leq 0$

$x_j \frac{\partial f}{\partial x_i} = 0, j = 1, \dots, n$ maka

$x^T (c^T + \Sigma x - A^T v) = 0$

$v_i g_i(x) = 0, i = 1, \dots, n$ maka $v(b - Ax) = 0$

$x_j \geq 0, j = 1, \dots,$ $x \geq 0$

$v_i \geq 0, i = 1, \dots, m, v \geq 0$

Persamaan (2.7) ditambahkan variabel q dan s untuk mencari penyelesaia optimal, sehingga diperoleh:

$c^T + \Sigma x - A^T v^T - q = 0$ dan $Ax - b + s = 0$

Dengan demikian, kondisi KKT menjadi:

$$-\Sigma x + A^T v^T + q = c^T$$
$$Ax - b = 0$$
$$q^T x = 0, vs = 0, x \geq 0, v \geq 0, q \geq 0, s \geq 0$$

Fungsi Barrier

Fungsi *barrier* berguna untuk menemukan solusi masalah optimasi nonlinear berkendala dengan mengubahnya menjadi barisan masalah tanpa kendala parameter. Bentuk umum fungsi *barrier* adalah sebagai berikut:

$$F(x, \gamma) = f(x) + \gamma \sum_j q_j(x) \quad (2.8)$$

dimana f(x) adalah fungsi tujuan, γ adalah pengali *barrier* positif dan $q_j(x) \to +\infty$ sebagai j kendala yang mendekati nol.

Jika meminimalkan fungsi f(x) dengan kendala $x \geq 0$, maka fungsi *barrier* yang dipakai untuk memberikan batas besar dari atau sama dengan nol terhadap suatu fungsi adalah fungsi *barrier* logaritma sebagai berikut:

$$\omega(x, \gamma) = f(x) - \gamma \log x \quad (2.9)$$

Nilai γ relatif besar mengakibatkan pencarian titik optimal berada jauh dari batas *feasible region*. Sebaliknya, nilai γ relatif kecil mengakibatkan pencarian titik optimal dilakukan di dekat batas *feasible region*.

Fungsi logaritmik *barrier* sesuai dengan fungsi Lagrange untuk masalah (2.5) adalah sebagai berikut:

$$L(x, v, \gamma) = \frac{1}{2} x^T \Sigma x + c^T x - \gamma \sum logx + v(b - Ax). \quad (2.10)$$

Masalah (2.5) menggunakan fungsi *barrier* logaritmik dalam hal ini $-\gamma \sum logx$ karena adanya kendala $x \geq 0$. Fungsi *barrier* yang terbentuk harus sesuai dengan kendala agar dapat diperoleh *feasible region* yang baik.

Metode Newton

Algoritma metode Newton untuk meminimumkan $f(x)$ tanpa kendala dengan $x \in \Re^n$, sebagai berikut:

a) Menentukan titik awal x^0, saat $k \leftarrow 0$.
b) Menentukan *Newton direction* $d^k = -\Sigma(x^k)^{-1}\nabla f(x^k)$. Jika $d^k = 0$, maka perhitungan berhenti.
c) Memilih *step size* $\alpha^k = 1$.
d) Mentukan $x^{k+1} \leftarrow x^k + \alpha^k d^k$, saat $k \leftarrow k+1$, kemudian kembali ke langkah b.

Metode Newton dapat dikembangkan dalam proses peminimalan fungsi berkendala khususnya dalam penyelesaian *quadratic programming* dengan metode titik interior. Dalam *quadratic programming*, *Newton direction* merupakan vektor yang mengarahkan ke nilai optimal fungsi tujuan.

3. HASIL DAN PEMBAHASAN

Penyelesaian Masalah Quadratic Programming dengan Metode Primal Dual Interior Point

Fungsi Lagrange untuk masalah (2.5) dapat dituliskan sebagai berikut:

$$L(x, v, \gamma) = \frac{1}{2}x^T\Sigma x + c^T x + v^T(b - Ax) - \gamma \sum_i log\ (x_i)$$

dimana v adalah pengali Lagrange dan γ adalah pengali dari *barrier term*.

Selanjutnya fungsi L diturunkan secara parsial terhadap x dan v, secara berturut-turut diperoleh kondisi *dual feasibility* dan *primal feasibily*.

$$\Sigma x + c - A^T v - \gamma X^{-1}$$
$$b - Ax = 0. \qquad (3.1)$$

X merupakan matriks diagonal $x_i, i = 1,2, \dots, n$. pada persamaan (3.2). Pembentukan *slackness condition* dapat dilakukan sebagai berikut:

$$u = \gamma X^{-1}$$
$$Xu = \gamma e \qquad (3.2)$$

dimana X adalah matriks diagonal x_i, U adalah matriks diagonal u_i, dan e adalah matriks yang semua entrinya bernilai 1.

Persamaan non-linear pada masalah *quadratic programming* menyebabkan tidak adanya teknik penyelesaian secara langsung, kecuali metode *trial and error* serta iterasi menggunakan metode Newton.Vektor x, u, v merupakan nilai awal yang pada iterasi selanjutnya akan menjadi $x + d_x, u + d_u, v + d_v$, dimana nilai d_x, d_u, d_v, perlu ditentukan.

$$-\Sigma(x + d_x) + A^T(v + d_v) + (u + d_u) = c$$
$$A(x + d_x) - b = 0 \qquad (3.3)$$
$$-XUe - Ud_x e - Xd_u e - d_x d_u e = -\gamma e.$$

$d_x d_u e$ pada sistem (3.3) memiliki nilai yang kecil, sehingga dapat diabaikan untuk melinearkan sistem (3.3) dengan (3.2) dan (3.1).

$$Ad_x = -Ax + b$$
$$-\Sigma d_x + d_u + A^T d_v = \Sigma x + c - A^T v - u \quad (3.4)$$
$$Ud_x + Xd_u = -XUe + \gamma e.$$

Berdasarkan (2.6), nilai $x, u,$ dan v yang baru dapat diperoleh dengan menyelesaikan persamaan berikut:

$$\begin{pmatrix} A & 0 & 0 \\ -\Sigma & I & A^T \\ U & X & 0 \end{pmatrix}\begin{pmatrix} d_x \\ d_u \\ d_v \end{pmatrix} = -\begin{pmatrix} Ax^k - b \\ -\Sigma x + A^T v^k + u^k - c \\ XUe - \gamma^k e \end{pmatrix} \qquad (3.5)$$

Matriks U dan X terdefinisi sebagai titik yang diketahui pada iterasi k. Perhitungan nilai d_x, d_u, d_v pada persamaan (3.5) dapat dilakukan sebagai berikut:

$$Ad_x = -Ax^k + b \equiv r_p \qquad (3.6)$$
$$-\Sigma d_x + d_u + A^T d_v = \Sigma x - A^T v^k - u^k + c \equiv r_d \qquad (3.7)$$
$$Ud_x + Xd_u = XUe - \mu^k e \equiv r_c. \qquad (3.8)$$

Dengan mengalikan persamaan (3.7) dengan X dan kemudian disubstitusi untuk Xd_u di persamaan (3.8) diperoleh:

$$-d_x + [X\Sigma + U]^{-1}XA^T d_v = [X\Sigma + U]^{-1}(Xr_d - r_c) \qquad (3.9)$$

Kalikan dengan matriks A, gunakan persamaan (3.9), diperoleh sistem persamaan berikut untuk menyelesaikan d_v:

$$A[X\Sigma + U]^{-1}XA^T d_v = r_p + A[X\Sigma + U]^{-1}(Xr_d - r_c)$$

d_x dapat dihitung dari (3.9) dan $d_u$dari (3.10).

$$d_x = [X\Sigma + \mathrm{U}]^{-1}(XA^T d_v - \mathrm{X}r_d + r_c)$$

dan

$$d_u = X^{-1}(r_c - Ud_x).$$

Diasumsikan nilai x, u, v yang baru dimulai dari nilai awal positif. Parameter α menyatakan *step length*. Nilai maksimum dari *step length* merupakan nilai yang membuat x_i atau γ_i menuju nol.

$$x_i + \alpha_p d_{xi} \geq 0 \; dan \; u^{k+1} = \gamma_i + \alpha_d d_{ui} \geq 0, i = 1, \dots, n$$

Untuk menjaga *feasibility*, *step length* bernilai lebih kecil dari nilai maksimum dan dirumuskan sebagai berikut:

$$\alpha_p = \beta \; Min \; [1, -\frac{x_i}{d_{xi}}, d_{xi} < 0]$$

$$\alpha_d = \beta \; Min \; [1, -\frac{u_i}{d_{ui}}, d_{ui} < 0]$$

β adalah skalar atau pengali yang bertujuan untuk menentukan banyaknya nilai yang ditambahkan dari arah sebelumnya pada proses penentuan nilai yang baru. Nilai parameter β berada diantara 0 dan 1. Nilai $\beta = 0.999$ sering digunakan. Nilai yang baru dari $x, u, dan\; v$ adalah sebagai berikut:

$$x^{k+1} = x^k + \alpha_p d_x$$
$$u^{k+1} = \mu^k + \alpha_d d_u$$
$$v^{k+1} = v^k + \alpha_d d_v.$$

Ada tiga syarat untuk mengetahui keoptimalan solusi, yakni memenuhi *konvergensi feasibility*, *konvergensi dual feasibility*, dan *complementary slackmess*. Apabila sudah memenuhi ketiga hal tersebut maka iterasi menggunakan metode Newton boleh dihentikan.

Fungsi kendala $Ax^k - b = 0$ harus terpenuhi oleh solusi optimal. Dengan demikian, kriteria konvergensi untuk *feasibility* dinyatakan sebagai berikut:

$$\sigma_p = \frac{\|Ax^k - b\|}{\|b\| + 1} \leq \epsilon_1$$

dimana ϵ_1 merupakan bilangan positif yang sangat kecil dan 1 ditambahkan pada penyebut untuk menghindari pembagian dengan bilangan yang bernilai kecil.

Selain itu dipunyai juga *dual feasibility* $-\Sigma x + u^k + A^T v^k = c$. Kriteria konvergensi untuk *dual feasibility* dinyatakan sebagai berikut:

$$\sigma_d = \frac{\|r_d\|}{\|\Sigma x + c\| + 1} \leq \epsilon_2$$

dimana ϵ_2 merupakan bilangan positif yang sangat kecil.

Nilai parameter μ menentukan seberapa baik *slackness condition* terpenuhi. Eksperimen numerik mendefinisikan sebuah nilai rata-rata dari μ sebagai berikut:

$$\gamma^k = \frac{(x^k)^T u}{n}$$

dimana n adalah banyaknya variabel optimasi. Parameter ini harus bernilai nol ketika kondisi optimum sudah tepenuhi. Sehingga, kriteria konvergensi untuk *slackness condition* adalah:

$$\gamma \leq \epsilon_3$$

dimana ϵ_3 merupakan bilangan positif yang sangat kecil.

Algoritma Penyelesaian Masalah Quadratic Programming dengan Metode Primal Dual Interior Point

Diberikan matriks koefisien kendala A, matriks Hessian Σ, vektor kendala ruas kanan b, vektor pengali Lagrange v, parameter perubahan titik β, dan parameter toleransi kekonvergenan.

Tahap-tahap penentuan nilai optimal adalah sebagai berikut:

a) Menentukan nilai awal saat k=0, dimana k menunjukkan iterasi yang dilakukan. Nilai awal yang diambil harus lebih besar atau sama dengan 0. Nilai awal yang diambil adalah:

$$x^0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \cdots \\ 1 \end{pmatrix}, u^0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \cdots \\ 1 \end{pmatrix}, v^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \cdots \\ 0 \end{pmatrix}.$$

b) Dilakukan perhitungan sebagai berikut:

$$r_p = -Ax^k + b$$
$$r_d = \Sigma^* x^k - A^T v^k - u^k + c$$
$$r_c = -XUe + u^k \mathrm{e}.$$

Dimana

$$\gamma^k = \left[\frac{(x^k)^T u}{n}\right] / (k+1).$$

c) Setelah vektor r_p, r_d, dan r_c ditentukan, dilakukan pengecekan konvergensi untuk melihat apakah nilai yang diambil sebelumnya sudah optimal atau belum. Kekonvergenan dipenuhi apabila:

$$\begin{bmatrix} \dfrac{\|Ax^k - b\|}{\|b\| + 1} \leq \epsilon_1, \\ \dfrac{\|r_d\|}{\|\Sigma^* x + c\| + 1} \leq \epsilon_2, \gamma \leq \epsilon_3 \end{bmatrix}$$

dimana $\epsilon_1, \epsilon_2, \epsilon_3$ merupakan bilangan yang bernilai kecil.

Jika hasil perhitungan sudah memenuhi syarat di atas, maka tidak usah dilakukan iterasi selanjutnya karena sudah ditemukan solusi optimal. Tetapi jika hasil perhitungan tidak memenuhi syarat di atas, maka harus dilakukan iterasi lagi.

d) Jika belum ditemukan solusi optimal, maka dilakukan perhitungan untuk mencari *Newton direction* d_x, d_v, d_u.

$$d_v = [[X\Sigma^* + \mathrm{U}]^{-1} XA^T]^{-1} [r_p + A[X\Sigma^* + \mathrm{U}]^{-1} (\mathrm{X}r_d - r_c)]$$

$$d_x = [X\Sigma^* + \mathrm{U}]^{-1} (XA^T d_v - \mathrm{X}r_d + r_c)$$

$$d_u = X^{-1}(r_c - Ud_x).$$

e) Penentuan *step length* dari nilai semula

$$\alpha_p = \beta\ Min\ [1, -\frac{x_i}{d_{xi}}, d_{xi} < 0]$$

$$\alpha_d = \beta\ Min\ [1, -\frac{u_i}{d_{ui}}, d_{ui} < 0].$$

f) Penentuan nilai $x^{k+1}, u^{k+1}, v^{k+1}$ yang baru:

$$x^{k+1} = x^k + \alpha_p d_x$$
$$u^{k+1} = \mu^k + \alpha_d d_u$$
$$v^{k+1} = v^k + \alpha_d d_v.$$

g) Langkah (b) dan (c) dilakukan kembali untuk mengecek konvergensi dari nilai yang baru. Jika syarat konvergensi terpenuhi, maka solusi optimal telah ditemukan. Apabila belum memenuhi syarat konvergensi, maka langkah (d), (e), dan (f) dilakukan kembali sampai menemukan solusi optimal.

Contoh Soal [4]

Tentukan solusi masalah *quadratic programming* berikut menggunakan metode primal-dual titik interior.

$$\min f = -2x_1 + \frac{x_1^2}{2} - 6x_2 - x_1x_2 + x_2^2$$

$$s.t. \begin{pmatrix} 3x_1 + x_2 \leq 25 \\ -x_1 + 2x_2 \leq 10 \\ x_1 + 2x_2 \leq 15 \\ x_i \geq 0, i = 1{,}2 \end{pmatrix}$$

Penyelesaian:

Langkah pertama adalah membawa sistem di atas ke bentuk standar *quadratic programming* sebagai berikut:

$$(QP) \begin{cases} min_x & \dfrac{1}{2} x^T \Sigma x + c^T x \\ s.t. & Ax = b \\ & x \geq 0 \end{cases}$$

Untuk memudahkan pembentukan matriks A, b, c dan Σ, maka masalah *quadratic programming* dapat dituliskan sebagai berikut:

$$\min f = -2x_1 + \frac{x_1^2}{2} - 6x_2 - x_1x_2 + x_2^2$$

$$s.t. \begin{pmatrix} 3x_1 + x_2 + x_3 = 25 \\ -x_1 + 2x_2 + x_4 = 10 \\ x_1 + 2x_2 + x_5 = 15 \\ x_i \geq 0, i = 1, \dots, 5 \end{pmatrix}$$

Dilakukan penambahan variabel slack x_3, x_4, x_5 untuk mengubah tanda (≤) menjadi (=).

Matriks koefisien kendala A, vektor kendala ruas kanan b, vektor koefisien suku-suku linear c, dan matriks hessian Σ dapat dituliskan sebagai berikut:

$$A = \begin{pmatrix} 3 & 1 & 1 & 0 & 0 \\ -1 & 2 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 25 \\ 10 \\ 15 \end{pmatrix}$$

$$c = \begin{pmatrix} -2 \\ -6 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Nilai awal untuk variabel $x, u, dan\ v$ dapat dituliskan dalam bentuk vektor sebagai berikut:

$$variabel\ primal (x) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

$$variabel\ dual\ (u) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$pengali\ (v) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Jika nilai $x_1 = 1, x_2 = 1$, maka $f = -7{,}5$.

Diambil $\in = 0{,}00001$, dari hasil pengecekan konvergensi diperoleh $\sigma_p = 0{,}760063$, $\sigma_d = 1{,}08505, \mu = 1$. Nilai $\sigma_p > \in$ $\sigma_d > \in, \mu > \in$

sehingga belum diperoleh solusi optimal. Dengan demikian, masih perlu dilakukan iterasi menggunakan metode Newton.
Pertama-tama, ditentukan r_p, r_d, r_c.

$$r_p = \begin{pmatrix} 20 \\ 8 \\ 11 \end{pmatrix}, r_d = \begin{pmatrix} -3 \\ -6 \\ -1 \\ -1 \\ -1 \end{pmatrix}, r_d = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Diketahui diagonal X dan U adalah sebagai berikut:

$$diagonal\ X = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$diagonal\ U = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Untuk memperoleh *newton direction* d_v, d_x, d_u atau vektor yang petunjuk menuju nilai minimal fungsi tujuan, maka dilakukan perhitungan sebagai berikut:

$$\mathrm{X\Sigma + U} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$(\mathrm{X\Sigma + U})^{-1} = \begin{pmatrix} 0,6 & 0,2 & 0 & 0 & 0 \\ 0,2 & 0,4 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$(\mathrm{X\Sigma + U})^{-1}\mathrm{XA^T} = \begin{pmatrix} 8 & 2,22045 \times 10^{-16} & 4 \\ 2,22045 \times 10^{-16} & 2,4 & 1 \\ 4 & 1 & 4 \end{pmatrix}$$

$$r_p + A(\mathrm{X\Sigma + U})^{-1}(\mathrm{X}r_d - r_c) = \begin{pmatrix} 7 \\ 4 \\ 1 \end{pmatrix}$$

Jadi,

$$d_v = \begin{pmatrix} 2,19079 \\ 2,76136 \\ -2,63158 \end{pmatrix} d_x = \begin{pmatrix} 4,19737 \\ 4,21711 \\ 3,19079 \\ 3,76316 \\ -1,63158 \end{pmatrix}, d_u = \begin{pmatrix} -4,19737 \\ -4,21711 \\ -3,19079 \\ -3,76316 \\ 1,63158 \end{pmatrix}$$

Setelah *Newton direction* ditentukan, maka dicari step length atau jarak antara nilai awal dengan nilai yang baru. Diperoleh nilai *step length* $\alpha_p = 0{,}606774$ dan $\alpha_d = 0{,}234758$.
Diperoleh nilai x, u, v yang baru atau $x', u', dan\ v'$ sebagai berikut:

$$x' = (3{,}54 \quad 3{,}55 \quad 2{,}93 \quad 3.28 \quad 0{,}01)$$
$$u' = (0{,}014 \quad 0{,}01 \quad 0{,}250 \quad 0{,}116 \quad 1{,}383)$$
$$v' = (0{,}514 \quad 0{,}648 \quad -0{,}617)$$
$$f = -22.114$$

$x_1 = 3{,}54685\ dan\ x_2 = 3{,}55883$. Setelah dilakukan pengecekan kriteria konvergensi, ternyata nilai $x_1\ dan\ x_2$ belum optimal sehingga dilakukan lagi iterasi sampai tujuh kali untuk memperoleh nilai optimal. Berdasarkan hasil iterasi ke-7 diperoleh $x_1 = 5{,}59981$, $x_2 = 4{,}69992$, $f = -27{,}9496$. Setelah dilakukan pengecekan konvergensi, dapat disimpulkan bahwa nilai optimal ditemukan pada iterasi ke-7.

4. KESIMPULAN

Berdasarkan pembahasan dari penelitian yang telah dilakukan, diperoleh kesimpulan bahwa penyelesaian masalah *quadratic programming* menggunakan metode primal dual titik interior dapat dilakukan untuk kasus yang lebih besar dari 2 variabel.

5. DAFTAR PUSTAKA

[1] Azis, A., Prihandono, B., & Ilhamsyah. (2016). Algoritma Genetika pada Pemrograman Linear dan Non-Linear. *Buletin Ilmiah Matematika, Statistika, Dan Terapannya (Bimaster)*, *5*(3), 265–274.

[2] Hariadi, V. (2009). Pencarian Solusi Pemrograman Non-Linear Menggunakan Algoritma Branch-and-Bound. *Section (SNATI) 2009: Bidang Informatika Teori*. Seminar

Nasional Aplikasi Teknologi Informasi (SNATI), Yogyakarta.
[3] Irawan, R., Eridani, & Jaelani, A. (2020). Ketaksamaan Hadamard pada Fungsi Konveks. *Contemporary of Mathematics and Applications*, *2*(1), 1–12.
[4] M Asghar Bhatti. (2000). *Practical optimization methods: with Mathematica applications*. Springer.
[5] Rianingsih, W., Hasan, M., & Pradjaningsih, A. (2017). Optimasi Portofolio dengan Menggunakan Pemprograman Kuadratik. *Majalah Ilmiah Matematika Dan Statistika*, *7*(2), 67–78.