PENYELARASAN ARAH VEKTOR GRADIEN UNTUK MENENTUKAN *STEP SIZE* METODE *STEEPEST DESCENT* PADA FUNGSI NONLINEAR KUADRATIK BANYAK VARIABEL

S. IDAMAN[1], B. P. SILALAHI[2], S. GURITMAN[3]

Abstrak

Masalah optimisasi banyak variabel dapat diselesaikan dengan berbagai metode untuk mendapatkan solusi yang optimal. Salah satu metode yang paling sederhana yaitu metode *steepest descent*. Metode *steepest descent* menggunakan vektor gradien untuk menentukan arah pencarian disetiap iterasi kemudian ditentukan *step size* sebagai jarak perubahan solusi yang dipengaruhi oleh vektor gradien. *Step size* (α_k) pada metode *steepest descent* sangat mempengaruhi kecepatan kekonvergenan metode ini. Sehingga diperlukan penentuan *step size* yang tepat untuk mempercepat kekonvergenan metode *steepest descent*. Penelitian ini akan memodifikasi *step size* pada metode *steepest descent* dengan menentukan *step size* yang dapat menghasilkan arah (vektor gradien) yang mendekati vektor eigen dari matriks Heisse suatu fungsi kuadratik definit positif banyak variabel. Hasil numerik menunjukkan bahwa *step size* yang diperoleh pada penelitian ini bisa mereduksi jumlah iterasi dan *running time* lebih baik dari pada metode *steepest descent* biasa terutama untuk kasus *ill-conditioned* yaitu kasus lamanya metode *steepest descent* mencapai kekonvergenan yang disebabkan oleh perbandingan (rasio) yang besar antara nilai eigen terbesar dan nilai eigen terkecil dari matriks Heisse.

Kata kunci: metode *steepest descent*, optimasi, *step size*, vektor gradien.

1 PENDAHULUAN

Metode *steepest descent* merupakan metode iteratif paling sederhana untuk menentukan nilai optimal dari suatu fungsi. Metode ini menggunakan vektor gradien untuk menentukan arah pencarian pada setiap iterasi. Metode ini pertama kali diperkenalkan oleh Cauchy [2] pada tahun 1847. Pada fungsi kuadratik tak linear dengan matiks Heisse-nya adalah matriks definit positif, kecepatan kekonvergenan metode *steepest descent* sangat bergantung pada perbandingan (rasio) nilai eigen terbesar dan nilai eigen terkecil dari matriks Heisse (Greenstadt [8]). Semakin besar perbandingan nilai eigen terbesar dan terkecil dari matriks Heisse akan mengakibatkan lamanya metode *steepest descent* untuk mencapai kekonvergenan solusi (*ill-conditioned problem*).

Dalam perkembangannya, metode *steepest descent* ini telah banyak dimodifikasi untuk mempercepat kekonvergenan solusi terutama dalam penentuan *step size* (ukuran langkah) pada setiap iterasi. Sebagai contoh, metode Barzilai dan

[1] Mahasiswa S2 Program Studi Matematika Terapan, Sekolah Pascasarjana IPB, Jalan Meranti Kampus IPB Dramaga Bogor, 16680. Email: syukrio.idaman@yahoo.com
[2] Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Jalan Meranti Kampus IPB Dramaga Bogor, 16680. E-mail : bibparuhumsilalahi@gmail.com
[3] Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Jalan Meranti Kampus IPB Dramaga Bogor, 16680. E-mail: guritman@yahoo.com

Borwein (BB) oleh Barzilai dan Borwein [1] yang memodifikasi *step size* berdasarkan pada penurunan pendekatan dua titik pada persamaan secant yang didasarkan pada metode quasi-Newton. *Step size* menggunakan metode ini mampu mereduksi jumlah iterasi yang dibutuhkan untuk mencapai kekonvergenan solusi, namun metode ini memiliki kelemahan yaitu berupa ketidakmonotonan dalam pencarian solusi pada setiap iterasi Metode *Alternating Minimization Gradient* (AM) oleh Dai YH dan Yuan Y [3] yang memodifikasi *step size* dengan meminimumkan norm gradient. *Step size* menggunakan metode AM mampu mereduksi jumlah iterasi yang dibutuhkan untuk mencapai kekonvergenan solusi pada tingkat keakuratan solusi yang kecil (kriteria penghentian iterasi cukup besar). Metode Yuan oleh Yuan Y [17] yang memodifikasi *step size* berdasarkan sifat kemonotonan dari metode *steepest descent* yaitu metode *steepest descent* selalu memiliki arah gradien yang saling tegak lurus dan mempunyai arah pencarian solusi dalam bentuk dua buah vektor gradien. *Step size* menggunakan metode Yuan mampu menghasilkan jumlah iterasi yang lebih sedikit terutama pada kasus berdimensi kecil namun pada kasus berdimensi besar tidak lebih baik dari metode BB. Selanjutnya penelitian-penelitian berikut mengkombinasikan beberapa metode dalam menyelesaikan masalah pengotimuman [14,16].

Menurut Frassoldati *et al*. [7], penyelarasan arah pencarian (vektor gradien) dengan suatu vektor eigen dapat mempercepat kekonvergenan metode *steepest descent*. Oleh karena itu perlu dilakukan penelitian untuk menentukan arah yang terbaik sehingga dapat mengarahkan pencarian solusi lebih cepat ke arah solusi optimal. Penelitian ini akan memodifikasi *step size* pada metode *steepest descent* dengan menentukan *step size* yang dapat menghasilkan arah (vektor gradien) yang mendekati vektor eigen dari matriks Heisse suatu fungsi kuadratik definit positif banyak variabel.

Pencarian nilai minimum dan maksimum ini dalam penerapannya dapat dilihat antara lain pada [5,6,9-13,15].

2 TINJAUAN PUSTAKA

2.1 Metode *Steepest Descent*

Metode *steepest descent* merupakan salah satu metode yang paling sederhana untuk mencari solusi masalah optimisasi tanpa kendala. Metode ini pertama kali dikemukakan oleh Cauchy (1847). Misalkan terdapat masalah pengoptimuman tanpa kendala

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \tag{1}$$

dimana $f(\mathbf{x})$ adalah fungsi yang terturunkan pada $\mathbb{R}^n$. Metode iteratif ini mengikuti bentuk

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\, \mathbf{g}_k \tag{2}$$

dimana $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k) = \nabla f(\mathbf{x}_k)$ adalah vektor gradient dari $f(\mathbf{x})$ di titik $\mathbf{x}_k$ dan $\alpha_k > 0$ adalah *step size*. *Step size* α_k dapat diperoleh dengan

$$\alpha_k = \min_{\alpha}\{f(\mathbf{x}_k - \alpha_k\, \mathbf{g}_k)\} \tag{3}$$

Algoritma *steepest descent*

Step 0 Tentukan nilai awal $\mathbf{x}_0 \in \mathbb{R}^n$ dan toleransi sebesar $0 < \varepsilon < 1$. Set $k = 0$

Step 1 Tentukan $\mathbf{g}_k$. Jika $\|\mathbf{g}_k\| \leq \varepsilon$, maka iterasi berhenti.
Step 2 Tentukan α_k sehingga meminimumkan $f(\mathbf{x}_k - \alpha_k\, \mathbf{g}_k)$
Step 3 Hitung $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\, \mathbf{g}_k$
Step 4 $k = k + 1$, dan kembali ke Step 1.

2.2 Metode Barzilai Borwein

Metode ini pertama kali dikemukakan oleh Barzilai dan Borwein pada tahun 1988. Ide utama dari Barzilai dan Borwein adalah menggunakan informasi dari iterasi sebelumnya untuk menentukan *step size* pada iterasi berikutnya. *Step size* yang digunakan pada metode ini didasarkan pada penurunan pendekatan dua titik pada persamaan secant yang didasarkan pada metode quasi-Newton. Bentuk metode Barzilai dan Borwein (BB):

$$\mathbf{x}_{k+1} = \mathbf{x}_k - B_k^{-1}\, \mathbf{g}_k \tag{4}$$

Dimana $B_k^{-1} = \alpha_k I$ dan I adalah matriks identitas dan B_k^{-1} adalah matriks Heisse dari $f(\mathbf{x})$ di titik x_k. Metode ini memilih *step size* α_k dengan bentuk

$$B_k = \arg\min_{B=\alpha I} \|B\mathbf{s}_{k-1} - \mathbf{y}_{k-1}\| \tag{5}$$

atau

$$B_k = \arg\min_{B^{-1}=\alpha I} \|\mathbf{s}_{k-1} - B^{-1}\mathbf{y}_{k-1}\| \tag{6}$$

dimana $\mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}$ dan $\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}$. Dari $B_k^{-1} = \alpha_k I$, diperoleh

$$\alpha_k^{BB1} = \frac{\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{s}_{k-1}}{\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} \tag{7}$$

atau

$$\alpha_k^{BB2} = \frac{\mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}} \tag{8}$$

2.3 Metode *Alternating Minimization Gradient*

Metode ini dikemukakan oleh Dai YH dan Yuan Y (2003). Ide utama pada metode ini yaitu memodifikasi *step size* dengan meminimumkan norm gradien ($\|g_k\|$). pada metode ini menggunakan pendekatan minimum norm gradien pada iterasi $2k-1$ dan menggunakan metode *steepest descent* pada iterasi $2k$. Pemilihan *step size* dilakukan dengan cara

$$\alpha_{2k-1} = \arg\min_{\alpha} \|\mathbf{g}(\mathbf{x}_{2k-1} - \alpha\, \mathbf{g}_{2k-1})\| \tag{9}$$

dan

$$\alpha_{2k} = \arg\min_{\alpha} \|f(\mathbf{x}_{2k} - \alpha\, \mathbf{g}_{2k})\|. \tag{10}$$

Sehingga diperoleh pemilihan *step size*

$$\alpha_k^{AM} = \begin{cases} \dfrac{\mathbf{g}_k^T A\mathbf{g}_k}{\mathbf{g}_k^T A^2\mathbf{g}_k}, & \text{jika } k \text{ ganjil} \\ \dfrac{\mathbf{g}_k^T\mathbf{g}_k}{\mathbf{g}_k^T A\mathbf{g}_k}, & \text{jika } k \text{ genap} \end{cases} \tag{11}$$

2.4 Metode Yuan

Metode ini dikemukakan oleh Yuan Y (2006). Ide utama dari metode ini berdasarkan sifat kemonotonan dari metode *steepest descent* yaitu metode *steepest*

descent selalu memiliki arah gradien yang saling tegak lurus dan mempunyai arah pencarian solusi dalam bentuk dua buah vektor gradien. Pada metode ini, *step size* dimodifikasi pada iterasi genap sehingga diperoleh

$$\alpha_{2k}^{Y} = \frac{2}{\sqrt{\left(\frac{1}{\alpha_{2k-1}} - \frac{1}{\alpha_{2k}}\right)^2 + \frac{4\|\mathbf{g}_{2k}\|^2}{\|\mathbf{s}_{2k-1}\|^2}} + \frac{1}{\alpha_{2k-1}} + \frac{1}{\alpha_{2k}}} \tag{12}$$

dimana $\mathbf{s}_{2k-1} = \mathbf{x}_{2k} - \mathbf{x}_{2k-1} = -\alpha_{2k-1}\mathbf{g}_{2k-1}$. Pada setiap iterasi, *step size* α_k metode Yuan dapat dituliskan sebagai berikut

$$\alpha_k = \begin{cases} \alpha_k^{SD}, & jika\ k\ ganjil \\ \alpha_k^{Y}, & jika\ k\ genap \end{cases} \tag{13}$$

dimana α_k^{SD} adalah *step size* menggunakan metode steepest descent biasa.

2.5 Metode *Steepest Descent* Untuk Fungsi Kuadratik

Misalkan terdapat masalah pengoptimuman tanpa kendala

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\,\mathbf{x} - \mathbf{b}^T\mathbf{x} \tag{14}$$

dimana $f(\mathrm{x})$ adalah fungsi yang terturunkan pada $\mathbb{R}^n$. Metode iteratif ini mengikuti bentuk

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\,\mathbf{g}_k \tag{15}$$

dimana $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k) = \nabla f(\mathbf{x}_k)$ adalah vektor gradient dari $f(\mathbf{x})$ di titik $\mathbf{x}_k$ dan $\alpha_k > 0$ adalah *step size*. *Step size* α_k dapat diperoleh dengan

$$\alpha_k^{SD} = \min_{\alpha}\{f(\mathbf{x}_k - \alpha_k\,\mathbf{g}_k)\} = \frac{\mathbf{g}_k^T\mathbf{g}_k}{\mathbf{g}_k^T A\mathbf{g}_k}. \tag{16}$$

3 METODE

3.1 Tahapan Penelitian

Penelitian ini disusun melalui tiga tahap, pertama dilakukan telaah pustaka mengenai metode *steepest descent* dan varian *step size*-nya. Pada tahap pertama ini akan menentukan *step size* metode *steepest descent* dengan melakukan penyelarasan arah pencarian (vektor gradien) dengan vektor eigen dari matriks Heisse suatu fungsi kuadratik banyak variabel. Kemudian membandingkan hasil uji komputasi *step size* yang baru dengan variasi *step size* metode *steepest descent* (metode *steepest descent*, metode BB, metode AM dan metode Yuan) dan mengimplementasikan metode ke dalam bahasa pemograman dengan menggunakan *software* Matlab. Setelah itu, dilakukan pengujian untuk beberapa fungsi kuadratik definit positif tak linier kemudian ditinjau dari segi iterasi dan *running time*.

3.2 Modifikasi *Step Size* Metode *Steepest Descent*

Tahap ini dilakukan penyelarasan arah pencarian (vektor gradien) dengan vektor eigen dari matriks Heisse suatu fungsi kuadratik tak linear banyak variable.

3.3 Pembuatan Algoritma

Pada tahap ini akan dilakukan pembuatan algoritma dengan modifikasi *step size* metode *steepest descent* yang dirancang untuk menentukan nilai optimum (minimum) persamaan kuadratik tak linear banyak variabel.

3.4 Pengujian Komputasi

Pada tahap ini akan dilakukan uji komputasi untuk membandingkan kemampuan *step size* yang diusulkan dengan metode *steepest descent*, metode BB, metode AM dan metode Yuan. Hasil uji komputasi ini digunakan untuk melihat jumlah iterasi dan waktu tempuh (*running time*) yang dibutuhkan metode baru untuk menyelesaikan beberapa fungsi yang telah diberikan. Perbandingan metode tersebut akan dilakukan dengan menggunakan fungsi-fungsi tak linear tak berkendala dengan bentuk kuadratik definit positif. Fungsi kuadratik yang akan digunakan pada penelitian ini mempunyai bentuk

$$f(x) = \frac{1}{2}(x - x^*)^T A\,(x - x^*) \tag{17}$$

$A = diag(\lambda_1, \lambda_2, \dots, \lambda_i, \dots, \lambda_n)$ dengan $n = 2, 3, 4, 5, 10, 20, 30, 40, 50, 100$.

Ditentukan $\lambda_1 = 1$, $\lambda_n = \{10, 100, 1000\}$, dan $\lambda_i (i = 2, \dots, n-1)$ adalah bilangan random diantara λ_1 dan λ_n. Lalu $x_i^* (i = 1, \dots, n)$ adalah bilangan random diantara selang $[-5.5]$. Pada penelitian ini digunakan titik awal $x_0 = \{0, 0, \dots, 0\}$ untuk semua dimensi , dan kriteria pemberhentian iterasi adalah $\|g_k\| < 10^{-8}$. Percobaan dilakukan dengan mengulang 3 kali untuk setiap dimensi dan setiap λ_n untuk setiap metode.

4 HASIL DAN PEMBAHASAN

4.1 Modifikasi *Step size*

Berdasarkan pada penelitian sebelumnya, penentuan *step size* sangat terkait dengan melakukan pendekatan *step size* berdasarkan invers dari matriks Heisse, minimum norm gradien dan sifat kemonotonan dari metode *steepest descent*. Namun, penyelarasan arah pencarian dengan suatu vektor eigen dapat mempercepat kekonvergenan metode *steepest descent* (Frassoldati *et al*. [7]). Penyelarasan arah pencarian ini juga telah diteliti oleh Asmundis *et al.* [4] yang memodifikasi metode *steepest descent* dengan membuat bentuk ke dalam ruang dimensi satu pada setiap iterasi. Pada penelitian ini penentuan *step size* dipilih berdasarkan pendekatan vektor eigen dari matriks Heisse. Sehingga diharapkan dapat mempercepat kekonvergenan solusi menuju titik optimal serta untuk menghindari bentuk zigzag metode *steepest descent*.

Diketahui bahwa hubungan antara suatu matriks, nilai eigen dan vektor eigen memiliki bentuk persamaan

$$A\mathbf{v} = \lambda\mathbf{v} \tag{18}$$

dimana A adalah suatu matriks segi, λ adalah suatu nilai eigen dari matriks A dan $\mathbf{v}$ adalah suatu vektor eigen yang bersesuaian dengan nilai eigen λ. Karena akan

dilakukan pemilihan *step size* yang memiliki arah yang sama (penyelarasan arah) dengan suatu vektor eigen dari matriks heisee, maka dapat dibuat suatu persamaan bahwa suatu vektor gradien (gradien ke-$k+1$) merupakan suatu vektor eigen dari matriks Heisse

$$A\,\mathbf{g}_{k+1} \approx \lambda \mathbf{g}_{k+1}. \tag{19}$$

Untuk fungsi kuadratik, vektor gradien pada iterasi ke- $k+1$ dapat menggunakan bentuk

$$\mathbf{g}_{k+1} = \mathbf{g}_k - \alpha_k A \mathbf{g}_k. \tag{20}$$

Sehingga persamaan $A\,\mathbf{g}_{k+1} \approx \lambda \mathbf{g}_{k+1}$ dapat dibentuk menjadi

$$A\,(\mathbf{g}_k - \alpha_k A \mathbf{g}_k) \approx \lambda(\mathbf{g}_k - \alpha_k A \mathbf{g}_k). \tag{21}$$

$$A\mathbf{g}_k - \alpha_k A A \mathbf{g}_k \approx \lambda \mathbf{g}_k - \alpha_k \lambda A \mathbf{g}_k \tag{22}$$

$$A\mathbf{g}_k - \lambda \mathbf{g}_k \approx \alpha_k A A \mathbf{g}_k - \alpha_k \lambda A \mathbf{g}_k \tag{23}$$

$$A\mathbf{g}_k - \lambda \mathbf{g}_k \approx \alpha_k A (A\mathbf{g}_k - \lambda \mathbf{g}_k) \tag{24}$$

$$\mathbf{y}_k \approx \alpha_k A \mathbf{y}_k \tag{25}$$

dengan $\mathbf{y}_k = A\mathbf{g}_k - \lambda \mathbf{g}_k$.
Akan ditentukan nilai $\alpha_k$sehingga

$$\mathbf{y}_k \approx \alpha_k A \mathbf{y}_k \tag{26}$$

***Step Size* Tipe 1**

Pemilihan *step size* akan menggunakan rumus

$$\alpha_k = \min_{\alpha_k} \|\mathbf{y}_k - \alpha_k A \mathbf{y}_k\|_2^2 \tag{27}$$

Dengan menggunakan uji turunan pertama, diperoleh

$$\alpha_k^{New1} = \frac{\mathbf{y}_k^T A \mathbf{y}_k}{\mathbf{y}_k^T A^2\, \mathbf{y}_k} \tag{28}$$

dengan $\mathbf{y}_k = A\,\mathbf{g}_k - \lambda\,\mathbf{g}_k$ dan λ adalah suatu nilai eigen positif dari matriks Heisse.

Penggunaan *step size* ini, akan menghasilkan vektor gradien yang searah dengan vektor eigen yang bersesuaian dengan nilai eigen λ, sehingga akan mengakibatkan $\mathbf{y}_k \approx 0$. Jika $\mathbf{y}_k \approx 0$, maka $A\,\mathbf{g}_k \approx \lambda\,\mathbf{g}_k$ Sehingga dengan menggunakan *step size* metode Cauchy pada iterasi selanjutnya (jika $\mathbf{y}_k \approx 0$) akan menghasilkan vektor gradien mendekati nol ($\mathbf{g}_{k+1} \approx \mathbf{0}$).

4.2 Pembuatan Algoritma

Misalkan untuk fungsi kuadratik

$$\min f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}} A\,\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x} \tag{29}$$

dengan $\mathbf{x} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^n$, dan $A \in \mathbb{R}^{n\times n}$ adalah sebuah matriks definit positif. Karena pemilihan *step size* bertujuan agar nilai fungsi objektif pada setiap iterasi semakin mengecil (kemonotonan), maka pemilihan *step size* harus memenuhi kondisi

$$f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k). \tag{30}$$

Diketahui bahwa jika fungsi $f(\mathbf{x})$ adalah fungsi kuadratik, agar menjamin nilai fungsi objektif pada setiap iterasi semakin mengecil maka pemilihan *step size* harus mengikuti aturan $\alpha_k \in (0\,, 2\alpha_k^{SD})$

Pemilihan *Step Size* Tipe 1a

Pemilihan *step size* tipe 1 dapat mengikuti aturan sebagai berikut

$$\alpha_k^{New1a} = \begin{cases} \dfrac{\mathbf{y}_k^T A \mathbf{y}_k}{\mathbf{y}_k^T A^2 \mathbf{y}_k}, & \text{jika } \mathbf{y}_k^T \mathbf{y}_k > \varepsilon \\ \dfrac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_k^T A \mathbf{g}_k}, & \text{jika } \mathbf{y}_k^T \mathbf{y}_k \leq \varepsilon \end{cases} \tag{31}$$

dengan $\mathbf{y}_k = A\,\mathbf{g}_k - \lambda\,\mathbf{g}_k$ dan λ adalah suatu nilai eigen positif dari matriks Heisse.

Proposisi 1

Jika $f(x) = \frac{1}{2}\mathbf{x}^{\mathbf{T}} A\,\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x}$ dengan $\mathbf{x} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^n$, dan $A \in \mathbb{R}^{n\times n}$ adalah sebuah matriks definit positif, agar menjamin sifat kemonotonan pada setiap iterasi terpenuhi maka pemilihan nilai eigen untuk *step size* baru agar $\alpha_k \in (0\,, 2\alpha_k^{SD})$ adalah nilai eigen terkecil dari matriks A.

Bukti.

Misalkan matriks $A = diag(\lambda_1, \lambda_2, \dots, \lambda_n)$ dengan $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_n$ dan $g_k = (g_k^{(1)},\ g_k^{(2)}, \dots, g_k^{(n)})$ dengan $g_k^{(i)}$ adalah vektor gradien ke$-i$ pada iterasi ke-k.

Akan dibuktikan $\alpha_k^{New1} < 2\alpha_k^{SD}$ dengan $\lambda = \lambda_1$

$$\alpha_k^{New1} = \frac{\mathbf{y}_k^T A \mathbf{y}_k}{\mathbf{y}_k^T A^2 \mathbf{y}_k} = \frac{\sum_{i=1}^n (\lambda_i - \lambda)^2 \lambda_i \left(g_k^{(i)}\right)^2}{\sum_{i=1}^n (\lambda_i - \lambda)^2 \lambda_i^2 \left(g_k^{(i)}\right)^2} \tag{32}$$

$$\alpha_k^{SD} = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_k^T A\,\mathbf{g}_k} = \frac{\sum_{j=1}^n \left(g_k^{(j)}\right)^2}{\sum_{j=1}^n \lambda_i \left(g_k^{(j)}\right)^2} \tag{33}$$

Jika $\lambda = \lambda_1$ maka

$$\alpha_k^{New1} = \frac{\sum_{i=2}^n (\lambda_i - \lambda_1)^2 \lambda_i \left(g_k^{(i)}\right)^2}{\sum_{i=2}^n (\lambda_i - \lambda_1)^2 \lambda_i^2 \left(g_k^{(i)}\right)^2} \tag{34}$$

$$\alpha_k^{New} < 2\alpha_k^{SD} \tag{35}$$

$$\alpha_k^{New} - 2\alpha_k^{SD} < 0 \tag{36}$$

$$\frac{\sum_{i=2}^n (\lambda_i - \lambda_1)^2 \lambda_i \left(g_k^{(i)}\right)^2}{\sum_{i=2}^n (\lambda_i - \lambda_1)^2 \lambda_i^2 \left(g_k^{(i)}\right)^2} - 2\frac{\sum_{j=1}^n \left(g_k^{(j)}\right)^2}{\sum_{j=1}^n \lambda_i \left(g_k^{(j)}\right)^2} < 0 \tag{37}$$

$$\frac{\sum_{j=1}^n \lambda_i \left(g_k^{(j)}\right)^2 \sum_{i=2}^n (\lambda_i - \lambda_1)^2 \lambda_i \left(g_k^{(i)}\right)^2}{\sum_{i=2}^n (\lambda_i - \lambda_1)^2 \lambda_i^2 \left(g_k^{(i)}\right)^2 \sum_{j=1}^n \lambda_i \left(g_k^{(j)}\right)^2} - \frac{2\sum_{j=1}^n \left(g_k^{(j)}\right)^2 \sum_{i=2}^n (\lambda_i - \lambda_1)^2 \lambda_i^2 \left(g_k^{(i)}\right)^2}{\sum_{i=2}^n (\lambda_i - \lambda_1)^2 \lambda_i^2 \left(g_k^{(i)}\right)^2 \sum_{j=1}^n \lambda_i \left(g_k^{(j)}\right)^2} < 0 \tag{38}$$

$$\frac{\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2\lambda_1\lambda_i\left(g_k^{(i)}\right)^2\left(g_k^{(1)}\right)^2+\sum_{\substack{i=2\\i=j}}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^4}{\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^2\sum_{j=1}^{n}\lambda_i\left(g_k^{(j)}\right)^2}+\frac{2\sum_{i=3}^{n}\sum_{\substack{j=2\\i\neq j}}^{i}(\lambda_i-\lambda_1)^2\lambda_i\lambda_j\left(g_k^{(i)}\right)^2\left(g_k^{(j)}\right)^2}{\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^2\sum_{j=1}^{n}\lambda_i\left(g_k^{(j)}\right)^2}-\frac{2\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^2\left(g_k^{(1)}\right)^2}{\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^2\sum_{j=1}^{n}\lambda_i\left(g_k^{(j)}\right)^2}-\frac{2\sum_{\substack{i=2\\i=j}}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^4}{\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^2\sum_{j=1}^{n}\lambda_i\left(g_k^{(j)}\right)^2}-\frac{4\sum_{i=3}^{n}\sum_{\substack{j=2\\i\neq j}}^{i}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^2\left(g_k^{(j)}\right)^2}{\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^2\sum_{j=1}^{n}\lambda_i\left(g_k^{(j)}\right)^2}<0 \tag{39}$$

$$\frac{\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2(\lambda_1-2\lambda_i)\lambda_i\left(g_k^{(i)}\right)^2\left(g_k^{(1)}\right)^2-\sum_{\substack{i=2\\i=j}}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^4}{\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^2\sum_{j=1}^{n}\lambda_i\left(g_k^{(j)}\right)^2}+\frac{\sum_{i=3}^{n}\sum_{\substack{j=2\\i\neq j}}^{i}(\lambda_i-\lambda_1)^2\lambda_i(2\lambda_j-4\lambda_i)\left(g_k^{(i)}\right)^2\left(g_k^{(j)}\right)^2}{\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^2\sum_{j=1}^{n}\lambda_i\left(g_k^{(j)}\right)^2}<0 \tag{40}$$

Karena

$$\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^2\sum_{j=1}^{n}\lambda_i\left(g_k^{(j)}\right)^2>0 \tag{41}$$

$$\sum_{i=2}^{n}(\lambda_i-\lambda_1)^2(\lambda_1-2\lambda_i)\lambda_i\left(g_k^{(i)}\right)^2\left(g_k^{(1)}\right)^2<0,\text{sebab }(\lambda_1-2\lambda_i)<0,\forall i \tag{42}$$

$$-\sum_{\substack{i=2\\i=j}}^{n}(\lambda_i-\lambda_1)^2\lambda_i^2\left(g_k^{(i)}\right)^4<0 \tag{43}$$

$$\sum_{i=3}^{n}\sum_{\substack{j=2\\i\neq j}}^{i}(\lambda_i-\lambda_1)^2\lambda_i\left(2\lambda_j-4\lambda_i\right)\left(g_k^{(i)}\right)^2\left(g_k^{(j)}\right)^2<0,\quad \text{sebab }\left(2\lambda_j-4\lambda_i\right)<0,\forall i>j \tag{44}$$

Akibatnya,

$$\alpha_k^{New1}-2\alpha_k^{SD}<0 \tag{45}$$

$$\alpha_k^{New1}<2\alpha_k^{SD}\blacksquare \tag{46}$$

Berdasarkan proposisi diatas, maka pemilihan nilai eigen adalah nilai eigen terkecil dari matriks Heisse. Sehingga algoritma *steepest descent*-nya menjadi

Step 0 Tentukan nilai awal $\mathbf{x}_0 \in \mathbb{R}^n$, λ adalah nilai eigen terkecil dari matriks A dan toleransi sebesar $0 < \varepsilon < 1$. Set $k = 0$

Step 1 Tentukan $\mathbf{g}_k$ dan $\mathbf{y}_k = A\,\mathbf{g}_k - \lambda\,\mathbf{g}_k$. Jika $\|\mathbf{g}_k\| \leq \varepsilon$, maka iterasi berhenti.

Step 2 Jika $\mathbf{y}_k^T\mathbf{y}_k > \varepsilon$, maka $\alpha_k = \alpha_k^{New1a} = \frac{\mathbf{y}_k^T A\mathbf{y}_k}{\mathbf{y}_k^T\, A^2\, \mathbf{y}_k}$.

Jika $\mathbf{y}_k^T\mathbf{y}_k \leq \varepsilon$, maka $\alpha_k = \alpha_k^{SD} = \frac{\mathbf{g}_k^T\mathbf{g}_k}{\mathbf{g}_k^T\, A\, \mathbf{g}_k}$

Step 3 Hitung $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\,\mathbf{g}_k$

Step 4 $k = k + 1$. Kembali ke Step 1.

Pemilihan *Step Size* Tipe 1b

Untuk kasus yang besar ($n > 5$) akan cukup sulit untuk menentukan nilai eigen. Oleh karena itu, penentuan nilai eigen dapat menggunakan Rayleigh quotient yaitu menentukan nilai eigen dengan hampiran numerik yaitu $\lambda_k = \frac{\mathbf{g}_k^T\, A\, \mathbf{g}_k}{\mathbf{g}_k^T\mathbf{g}_k} = \frac{1}{\alpha_k^{SD}}$.

Pemilihan *step size* lainnya dapat mengikuti aturan sebagai berikut

$$\alpha_k^{New1b} = \begin{cases} \dfrac{\mathbf{y}_k^T A\mathbf{y}_k}{\mathbf{y}_k^T\, A^2\, \mathbf{y}_k}, & \text{jika } \mathbf{y}_k^T\mathbf{y}_k > \varepsilon \\ \dfrac{\mathbf{g}_k^T\mathbf{g}_k}{\mathbf{g}_k^T\, A\, \mathbf{g}_k}, & \text{jika } \mathbf{y}_k^T\mathbf{y}_k \leq \varepsilon \end{cases} \tag{47}$$

dengan $\mathbf{y}_k = A\,\mathbf{g}_k - \frac{1}{\alpha_k^{SD}}\mathbf{g}_k$ dan α_k^{SD} adalah *step size* metode *steepest descent* cauchy.

Sehingga algoritma *steepest descent*-nya menjadi

Step 0 Tentukan nilai awal $\mathbf{x}_0 \in \mathbb{R}^n$, dan toleransi sebesar $0 < \varepsilon < 1$. Set $k = 0$

Step 1 Tentukan $\mathbf{g}_k$. Jika $\|\mathbf{g}_k\| \leq \varepsilon$, maka iterasi berhenti.

Step 2 Hitung $\lambda_k = \frac{\mathbf{g}_k^T\, A\, \mathbf{g}_k}{\mathbf{g}_k^T\mathbf{g}_k}$ dan $\mathbf{y}_k = A\,\mathbf{g}_k - \frac{1}{\alpha_k^{SD}}\mathbf{g}_k$

Step 3 Jika $\mathbf{y}_k^T\mathbf{y}_k > \varepsilon$, hitung $\alpha_k^{New} = \frac{\mathbf{y}_k^T A\mathbf{y}_k}{\mathbf{y}_k^T\, A^2\, \mathbf{y}_k}$.

Jika $\alpha_k^{New1b} < \frac{2}{\lambda_k}$, pilih $\alpha_k = \alpha_k^{New1b}$

Jika $\alpha_k^{New1b} > \frac{2}{\lambda_k}$, pilih $\alpha_k = \frac{1}{\lambda_k}$

Jika $\mathbf{y}_k^T\mathbf{y}_k \leq \varepsilon$, hitung $\alpha_k^{SD} = \frac{1}{\lambda_k}$

Step 4 Hitung $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\,\mathbf{g}_k$

Step 5 $k = k + 1$. Kembali ke Step 1.

4.3 Pengujian Komputasi

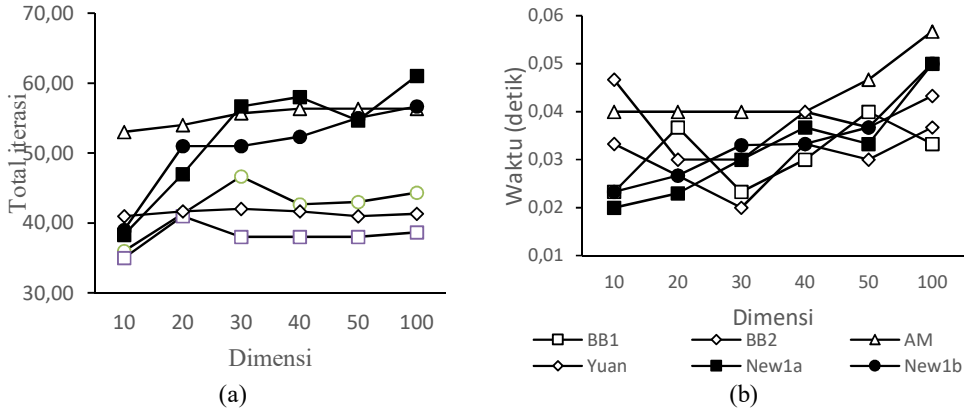
Pada pembahasan ini akan dilakukan perbandingan hasil numerik antara metode *steepest descent*, metode BB, metode AM, metode Yuan dan metode modifikasi baru yaitu penentuan *step size* tipe 1. Perbandingan metode tersebut akan dilakukan dengan menggunakan fungsi-fungsi tak linear tak berkendala dengan bentuk kuadratik.

Tabel 1
Perbandingan rata-rata hasil iterasi menggunakan *step size* tipe 1

n	λ_n	SD	BB1	BB2	AM	Yuan	New 1a	New 1b
2	10	16.67	8	6	5.67	3	2	4
	100	10.33	8	6	5	3	2	3
	1000	7	8	6	5	3	2	3
3	10	97	22.33	21.33	34.33	12.67	8.33	9.67
	100	1005	24	25.333	93.333	9	10.67	12
	1000	10003.7	38.33	26.33	420.33	7	12.67	15.67
4	10	100.33	32	30.33	52.33	45.67	20.33	15.67
	100	1001.67	77	90	150.33	107	27.33	20
	1000	9981	77.33	173.67	303.33	260.67	33.67	26.67
5	10	100.33	31.67	34.33	52.33	38.67	25.67	24.67
	100	995.67	77.67	55	148	135.33	35.67	41.33
	1000	9953	357.33	62.67	716.67	419.67	44.67	42.67
10	10	100.33	36	35	53	41	39	38.33
	100	997.67	92	134.67	143.67	152.22	76.67	74.33
	1000	9953	325.33	354	732.67	952.67	99.67	94.33
20	10	100.33	41.33	41	54	41.67	33	31.33
	100	997.67	142.33	143	139.67	264.33	133.33	132.33
	1000	9957.67	492.33	344.33	767	2244.67	178.67	177.67
30	10	100.67	46.67	38	55.67	42	37.33	35
	100	997.67	101.33	108.33	136	218.33	109.67	108.67
	1000	9959	433.33	245.67	762.67	2111.33	145.67	144
40	10	100.67	42.67	38	56.33	41.67	39.67	37.33
	100	998.33	150.67	155.67	163	243.33	139	138
	1000	9959,67	408.67	414.67	676.33	1952.67	187	186
50	10	101.33	43	38	56.33	41	41	38.67
	100	998.33	146	153	161.33	243	167	165.33
	1000	9960.33	344	372.33	743	1930	227.33	226.33
100	10	101.33	44.33	38.67	56.33	41.33	44	41
	100	998.33	153	151.33	166.67	251	285.33	283.67
	1000	9961	465.33	401.33	678.67	1988.67	416.33	415.33

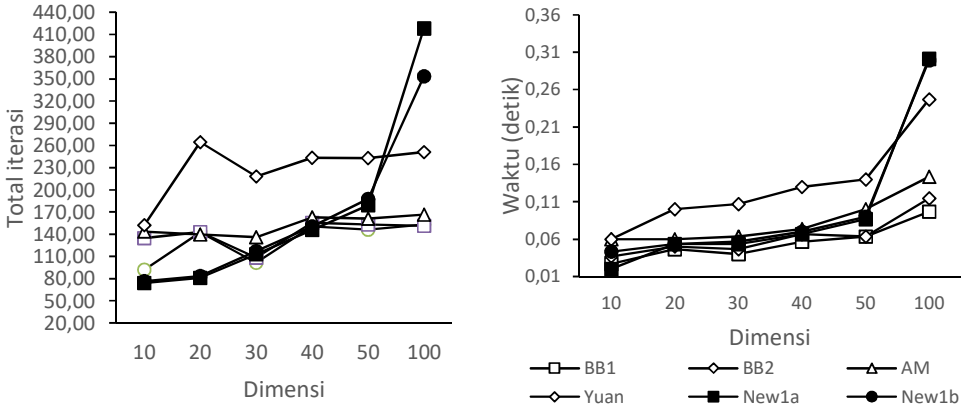
Tabel 2
Perbandingan rata-rata hasil *running time* menggunakan *step size* tipe 1

n	λ_n	SD	BB1	BB2	AM	Yuan	New 1a	New 1b
2	10	0.0567	0.02	0.02	0.0203	0.02	0.01	0.02
	100	0.038	0.02	0.0177	0.02	0.01	0.01	0.02
	1000	0.038	0.02	0.0177	0.02	0.01	0.01	0.02
3	10	0.038	0.025	0.023	0.027	0.02	0.02	0.02
	100	0.16	0.02	0.02	0.04	0.02	0.0233	0.02
	1000	8.6357	0.03	0.025	0.0967	0.02	0.0167	0.023
4	10	0.04	0.0233	0.0233	0.03	0.03	0.0233	0.0167
	100	0.1903	0.03	0.03	0.0467	0.05	0.0233	0.03
	1000	10.928	0.03	0.0433	0.0767	0.0733	0.0267	0.03
5	10	0.0467	0.03	0.03	0.04	0.03	0.02	0.03
	100	0.2257	0.04	0.0367	0.0567	0.0533	0.0333	0.0367
	1000	12.9807	0.0667	0.0367	0.1953	0.1033	0.0267	0.03
10	10	0.039	0.0233	0.0333	0.04	0.0467	0.02	0.0233
	100	0.3093	0.0267	0.0367	0.06	0.06	0.02	0.0433
	1000	24.3153	0.0667	0.0733	0.236	0.3873	0.0367	0.04
20	10	0.0433	0.0367	0.0267	0.04	0.03	0.023	0.0267
	100	0.5407	0.0467	0.05	0.06	0.1	0.0533	0.0533
	1000	49.2563	0.151	0.0933	0.41	2.723	0.0567	0.0733
30	10	0.0433	0.0233	0.02	0.04	0.03	0.03	0.033
	100	0.7643	0.04	0.0467	0.0633	0.1067	0.0533	0.0567
	1000	73.5493	0.1827	0.08	0.5313	3.4643	0.0633	0.0667
40	10	0.05	0.03	0.0333	0.04	0.04	0.0367	0.0333
	100	1.025	0.0567	0.0667	0.0733	0.13	0.0667	0.07
	1000	98.7803	0.2217	0.1973	0.544	3.8593	0.0767	0.0867
50	10	0.0567	0.04	0.03	0.0467	0.0367	0.0333	0.0367
	100	1.2673	0.0633	0.0633	0.1	0.14	0.0867	0.09
	1000	125.04	0.174	0.228	0.7833	4.784	0.12	0.1367
100	10	0.0733	0.0333	0.0367	0.0567	0.0433	0.05	0.05
	100	2.486	0.0967	0.1147	0.1433	0.2467	0.3013	0.299
	1000	244.933	0.5683	0.4357	1.2977	10.1923	0.5267	0.5407



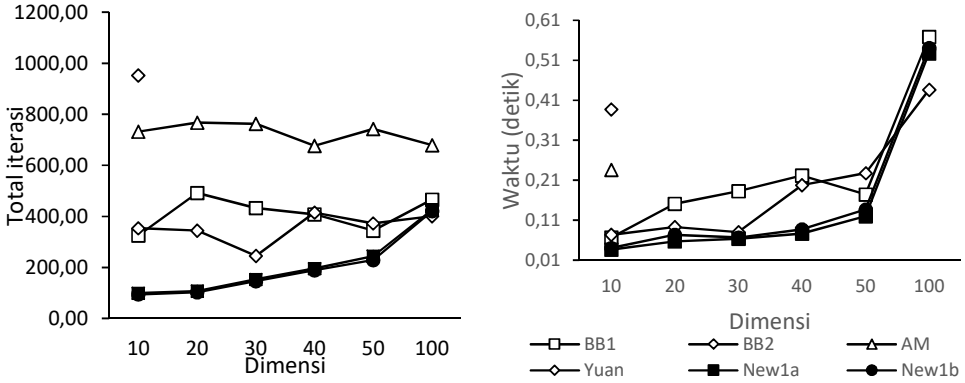
(a) (b)

Gambar 1 Perbandingan rata-rata banyaknya iterasi (a) dan lama *running time* (b) dari variasi *step size* metode *steepest descent* untuk $\lambda_n = 10$



(a) (b)

Gambar 2 Perbandingan rata-rata banyaknya iterasi (a) dan lama *running time* (b) dari variasi *step size* metode *steepest descent* untuk $\lambda_n = 100$



(a) (b)

Gambar 3 Perbandingan rata-rata banyaknya iterasi (a) dan lama *running time* (b) dari variasi *step size* metode *steepest descent* untuk $\lambda_n = 100$

Berdasarkan hasil iterasi pada Gambar 1, Gambar 2 dan Gambar 3, diketahui bahwa metode baru (*step size* tipe 1) ini membutuhkan jumlah iterasi yang relatif sedikit dari metode lainnya. Namun jumlah iterasi semakin meningkat seiring dengan bertambahnya variabel.

5 SIMPULAN DAN SARAN

Simpulan

Berdasarkan banyaknya iterasi yang diperlukan untuk menyelesaikan masalah fungsi kuadratik banyak variabel, pemilihan *step size* baru ini lebih cepat menuju kekonvergenan dari pada metode lainnya. Khususnya untuk fungsi yang memiliki rasio yang relatif besar antara nilai eigen terkecil dan terbesar dari

matriks Heisse. Namun jumlah iterasi semakin meningkat seiring dengan bertambahnya variabel.

Saran

Perlu penelitian lebih lanjut mengenai sifat dan karakteristik dari step size baru ini dan perlu dilakukan pengujian lebih lanjut untuk fungsi nonkuadratik.

DAFTAR PUSTAKA

[1] Barzilai J, Borwein JM. 1988. Two-Point Step Size Gradient Method. *IMA Journal of Numerical Analysis.* 8(1):141-148.

[2] Cauchy A. 1847. M´ethodes g´en´erales pour la r´esolution des syst`emes d´´equations simultan´ees. *CR Acad Sci Par.* 25(1):536–538.

[3] Dai YH, Yuan Y. 2003. Alternate Minimization Gradient Method. *IMA Journal of Numerical Analysis*. 23(3): 377-393.

[4] De Asmundis R., *et al*. 2012. On Spectral Properties of Steepest Descent Method. *IMA Journal Numerical Analysis*. 33(4):1416–1435.

[5] Estuningsih RD, Guritman S, Silalahi BP. 2014. Algorithm construction of HLI hash function. *Far East Journal of Mathematical Sciences*. 86(1) : 23-36.

[6] Firmansyah Z, Herdiyeni Y, Silalahi BP, Douady S. Landmark Analysis Of Leaf Shape Using Polygonal Approximation. *IOP Conference Series: Earth and Environmental Science 2016*. 31(1): 012018).

[7] Frassoldati G, Zanni L, Zanghirati G. 2008. New Adaptive Stepsize Selections in Gradient Methods. *JIMO.* 4(2):299–312.

[8] Greenstadt J. 1967. On The Relative Efficiencies of Gradient Method. *Math Comp.* 21(1):360-367.

[9] Kalengkongan WW, Silalahi BP, Herdiyeni Y, Douady S. 2015. Landmark analysis of leaf shape using dynamic threshold polygonal approximation. Di dalam: *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*; 2015 Okt 10. hlm: 287-292.

[10] Krisnawijaya NN, Herdiyeni Y, Silalahi BP. 2017. Parallel Technique for Medicinal Plant Identification System using Fuzzy Local Binary Pattern. *Journal of ICT Research and Applications*. 11(1):77-90.

[11] Saifudin MA, Silalahi BP, Sitanggang IS. 2015. Star Catalog Generation for Satellite Attitude Navigation Using Density Based Clustering. *Journal of Computer Science.* 11(12): 1082-1089.

[12] Silalahi BP. 2014. Sharper analysis of upper bound for the iteration complexity of an interior-point method using primal-dual full-Newton step algorithm. *Far East Journal of Mathematical Sciences*. 95(1):69-80.

[13] Silalahi BP, Dewi MS. 2014. Comparison Of Sensitivity Analysis On Linear Optimization Using Optimal Partition And Optimal Basis (In The Simplex Method) At Some Cases. *Indonesian Mathematical Society*. 1(1):82-90.

[14] Silalahi BP, Laila R, Sitanggang IS. 2017. A combination method for solving nonlinear equations. *IOP Conference Series: Materials Science and Engineering 2017*. 166(1): 012011.

[15] Silalahi BP, Wungguli D, Guritman S. 2015. Steepest Descent Method with New Step Sizes. World Academy of Science, *Engineering and Technology, International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*. 9(7): 378-384.

[16] Wihartiko FD, Buono A, Silalahi BP. 2017. Integer programming model for optimizing bus timetable using genetic algorithm. *IOP Conference Series: Materials Science and Engineering 2017*. 166(1): 012016.

[17] Yuan Y. 2006. A New Step Size for The Steepest Descent Method. *Journal of Computational Mathematics.* 24(2):149-156.