

## PEMANFAATAN JARINGAN SARAF TIRUAN UNTUK PENYELESAIAN PERMASALAHAN OPTIMASI NONLINIER

Victor Hariadi<sup>1)</sup>, Rully Soelaiman<sup>2)</sup>

<sup>1,2)</sup> Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya  
Gedung Teknik Informatika, Kampus ITS, Jl. Raya ITS, Sukolilo, Surabaya - 60111  
email : [victor@its-sby.edu](mailto:victor@its-sby.edu), [rully@its-sby.edu](mailto:rully@its-sby.edu)

### ABSTRAK

Saat ini semakin banyak permasalahan pada kehidupan sehari-hari yang memerlukan pendekatan optimasi dalam penyelesaiannya. Pendekatan optimasi sendiri menyediakan banyak alternatif metode yang dapat dipilih sesuai dengan karakteristik permasalahan yang akan diselesaikan. Penyelesaian permasalahan riil menggunakan pendekatan optimasi akan melibatkan model matematis. Model yang dibuat/digunakan akan menentukan pada koridor teknik optimasi mana kita akan bekerja. Secara garis besar, permasalahan dalam teknik optimasi dapat berupa permasalahan (pemrograman) linier atau non linier. Sebenarnya kedua kelompok permasalahan ini masih memberikan ruang cukup luas bagi kegiatan riset yang bertujuan untuk merancang konsep atau metode penyelesaian yang lebih efisien. Namun pemrograman non linier menyisakan area yang lebih luas, mengingat model-model non linier seringkali memiliki bentuk yang lebih kompleks dan dinamis. Klas-klas pemrograman non linier dapat ditentukan dari bentuk D dan karakteristik fungsi tujuan/obyektif serta dari keberadaan dan bentuk fungsi pembatasnya. Salah satu subklas dalam permasalahan pemrograman nonlinier adalah masalah pemrograman kuadrat dengan fungsi obyektif berbentuk fungsi konveks. Penelitian ini membahas penggunaan recurrent neural network untuk menyelesaikan permasalahan minimisasi pemrograman kuadrat dengan batasan linier. Recurrent neural network digunakan karena mempunyai kelebihan pada strukturnya yang lebih sederhana dan kompleksitas yang lebih rendah untuk diimplementasikan daripada neural network yang digunakan sebelumnya untuk menyelesaikan permasalahan tersebut di atas. Ini menunjukkan bahwa recurrent neural network lebih stabil pada keadaan Lyapunov dan secara global mampu mencapai konvergensi dalam waktu singkat.

**Kata kunci:** teknik optimasi, pemrograman nonlinier, pemrograman kuadrat, recurrent neural network.

### 1. PENDAHULUAN

Banyak permasalahan perancangan dapat diselesaikan dengan mengubah permasalahan orisinal menjadi permasalahan optimasi konveks dengan batasan linear. Sebagai contohnya, permasalahan kuadrat terkecil dengan batasan persamaan linear dapat dilihat sebagai framework dasar yang digunakan secara luas untuk system modelling dan desain dalam aplikasi-aplikasi yang bervariasi seperti signal and image processing serta pengenalan pola.

Pada berbagai aplikasi, penyelesaian real-time biasanya penting. Salah satu contohnya pada pengolahan citra image fusion untuk transmisi citra wireless real-time. Dibandingkan dengan metode numerik tradisional untuk optimasi dengan batasan, pendekatan neural network memiliki beberapa kelebihan pada aplikasi-aplikasi real-time. Pertama, struktur dari neural network dapat diimplementasi secara efektif menggunakan VLSI dan teknologi optikal. Kedua, neural network dapat menyelesaikan banyak permasalahan optimasi dengan parameter yang bermacam-macam. Ketiga, teknik ordinary differential equation (ODE) numerik dapat diaplikasikan secara langsung pada neural network.

Dalam makalah ini dicoba penggunaan one-layer neural network untuk menyelesaikan permasalahan-permasalahan optimasi konveks.

### 2. PEMROGRAMAN NON LINIER DAN PEMROGRAMAN KUADRATIK

Permasalahan pemrograman non linier adalah proses untuk menyelesaikan sistem (persamaan atau pertidaksamaan), secara bersamaan dalam jangka waktu tertentu dengan menyertakan fungsi obyektif non linier untuk maksimasi atau minimisasi.

Jika fungsi obyektif yang dimaksud berbentuk fungsi kuadrat dengan fungsi-fungsi pembatas berbentuk linier, maka permasalahannya disebut Permasalahan Pemrograman Kuadrat (PPK) [12].

#### 2.1. Bentuk dasar pemrograman kuadrat

Bentuk dasar pemrograman kuadrat bentuk primal dan dual adalah [15]:

- **Bentuk Primal :**  
Minimasi  $f(x) = \frac{1}{2} x^T Q x + c^T x$   
Subject to  $g(x) = A x - b$  (2.1)

- **Bentuk Dual:**  
Maksimasi  $f(x) = -\frac{1}{2} x^T Q x + b^T y$   
Subject to  $g(x) = c + Q x - A^T y + z$  (2.2)

dengan :

- A : matriks berdimensi  $m \times n$  yang berisi koefisien variabel keputusan dalam fungsi pembatas;
- Q : matriks simetris berdimensi  $n \times n$  yang berisi koefisien dari suku kuadrat fungsi obyektif;
- c : vektor berdimensi  $n$  yang berisi koefisien dari suku linier fungsi obyektif;
- b : vektor berdimensi  $m$  untuk menampung koefisien yang ada di sisi kanan;
- x : vektor berdimensi  $n$  untuk menampung semua variabel keputusan;
- m : menunjukkan jumlah fungsi pembatas;
- n : menunjukkan jumlah variabel keputusan yang digunakan;
- y : vektor berdimensi  $m$  untuk menampung variabel bentuk yang digunakan dalam fungsi obyektif bentuk dual;
- z : vektor berdimensi  $n$  untuk menampung variabel bentuk yang digunakan dalam fungsi pembatas bentuk dual;

## 2.2. Konveksitas fungsi obyektif

Uji konveksitas dilakukan untuk memastikan bahwa permasalahan kita memiliki satu himpunan solusi yang unik. Selain pengujian analitis melalui fungsi obyektifnya, uji konveksitas ini dapat memanfaatkan matriks Q. Suatu fungsi obyektif (dalam hal ini adalah fungsi kuadrat) merupakan fungsi konveks jika matriks Q semidefinit positif. Pengujian matriks Q dapat dilakukan dengan 2 cara [2][4]:

1.  $x^T Q x > 0$
2. nilai eigen(Q)  $\geq 0$ , nilai eigen dapat diperoleh dari perhitungan  $\det(\lambda I - Q) = 0$

## 2.3. Kondisi optimal

Kondisi optimal dari penyelesaian permasalahan pemrograman kuadrat akan tercapai apabila memenuhi kondisi Karush-Kuhn-Tucker (KKT).

Kondisi KKT merupakan generalisasi dari metode *Lagrange multiplier* yang diaplikasikan untuk bentuk primal permasalahan pemrograman kuadrat [1]:

$$L(x, \mu) = c^T x + \frac{1}{2} x^T Q x - \mu(Ax - b) \quad (2.3)$$

Hanya saja penerapan KKT, yang identik dengan penerapan Lagrange multiplier ini, selain berfungsi untuk menjamin diperolehnya solusi yang optimal, dalam prosesnya juga me-linier-isasi bentuk nonlinier (kuadrat) dari fungsi obyektif menjadi bentuk linier serta membuat pertidaksamaan dalam fungsi pembatas menjadi bentuk persamaan. Hal ini dilakukan dengan tujuan untuk mempermudah proses komputasi dan penyelesaian secara umum permasalahan pemrograman kuadrat itu sendiri.

Formulasi umum Lagrange multiplier yang ditulis dengan mengkaitkan variabel dual, seperti berikut:

$$L(x, \lambda) = c^T x + \frac{1}{2} x^T Q x - y^T(Ax - b) \quad (2.4)$$

dimana  $y$  merupakan variabel dual dari persamaan primal pemrograman kuadrat.

Dikatakan bahwa sebuah sistem permasalahan nonlinier (khususnya permasalahan pemrograman kuadrat) akan memiliki solusi optimal, yaitu apabila memenuhi kondisi KKT. Kondisi KKT merupakan turunan pertama dari persamaan umum Lagrange (2.4).

Sehingga persamaan kondisi KKT yang digunakan menjadi :

$$\begin{aligned} c + Qx - A^T y + z &= 0 \\ Ax - b &= 0 \\ xz &= 0 \\ (x, z) &\geq 0 \end{aligned} \quad (2.5)$$

## 3. TINJAUAN TERHADAP RECURRENT NEURAL NETWORK

Lemma 1 :

$x^*$  adalah solusi untuk persamaan (2.1) jika dan hanya jika terdapat  $y^* \in R^m$ , yang memenuhi [1]

$$\begin{cases} (x - \alpha \nabla f(x) + \alpha A^T y)^+ - x = 0 \\ Ax - b = 0 \end{cases}$$

dimana  $\alpha$  konstanta positif  
 $(x)^+ = [(x_1)^+, \dots, (x_n)^+]^T$   
 $(x_i)^+ = \max\{0, x_i\}$

Berdasarkan formulasi pada Lemma 1, maka kita mendapatkan recurrent neural network dengan struktur satu-layer untuk menyelesaikan permasalahan (3.1), dengan rumus [13]

$$\frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} -x + (x - \alpha \nabla f(x) + \alpha A^T y)^+ \\ \alpha(-Ax + b) \end{pmatrix} \quad (3.1)$$

dimana  $\lambda >$  konstanta skala

Model jaringan saraf tiruan yang digunakan ini akan menunjukkan bahwa penyelesaian permasalahan pemrograman kuadrat akan konvergen pada suatu titik kesetimbangan, baik penyelesaian dengan nilai inisial berasal dari dalam daerah kelayakan maupun dari luar daerah kelayakan.

Ada 2 pengembangan jaringan saraf tiruan dari persamaan (3.1) yang digunakan pada penelitian ini, yaitu :

- permasalahan  $x$  di domain tertentu
- permasalahan inequality variational

### Permasalahan x di domain tertentu

Diketahui permasalahan pemrograman convex nonlinear dengan bentuk :

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{subject to } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \in \Omega_1 \end{aligned} \quad (3.2)$$

dimana  $f(x)$  continuously differentiable dan konveks dari  $\mathbb{R}^n$  ke  $\mathbb{R}$

$$\begin{aligned} & \mathbf{A} \in \mathbb{R}^{m \times n} \\ & \mathbf{b} \in \mathbb{R}^m \\ & \Omega_1 = \{x \in \mathbb{R}^n \mid \leq x \leq h\} \end{aligned}$$

Berdasarkan pada Lemma 1, maka :

$$\begin{cases} \mathbf{P}_{\Omega_1}(\mathbf{x} - \alpha \nabla f(\mathbf{x}) + \alpha \mathbf{A}^T \mathbf{y}) - \mathbf{x} = \mathbf{0} \\ \mathbf{Ax} - \mathbf{b} = \mathbf{0} \end{cases} \quad (3.3)$$

dimana  $\mathbf{P}_{\Omega_1} : \mathbb{R}^n \rightarrow \Omega_1$  adalah operator proyeksi yang didefinisikan dengan  $\mathbf{P}_{\Omega_1}(x) = [\mathbf{P}_{\Omega_1}(x_1), \dots, \mathbf{P}_{\Omega_1}(x_n)]^T$ , dan

$$\mathbf{P}_{\Omega_1}(x_i) = \begin{cases} 1_i \Rightarrow x_i < l_i \\ x_i \Rightarrow l_i \leq x_i \leq h_i \\ h_i \Rightarrow x_i > h_i \end{cases}$$

Sebagai salah satu perluasan dari jaringan saraf tiruan pada persamaan (3.1), model jaringan saraf tiruan untuk memecahkan persamaan (3.2) diberikan sebagai berikut :

$$\frac{d}{dx} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{P}_{\Omega_1}(\mathbf{x} - \alpha \nabla f(\mathbf{x}) + \alpha \mathbf{A}^T \mathbf{y}) - \mathbf{x} \\ \alpha(-\mathbf{Ax} + \mathbf{b}) \end{pmatrix} \quad (3.4)$$

Teorema 1 :

Asumsikan  $f(x)$  adalah strictly convex dan dua kali lebih differentiable untuk setiap  $x \geq 0$ . maka jaringan saraf tiruan pada persamaan (3.4) stabil pada tahap Lyapunov dan konvergen pada titik  $u^* = \{x^*, y^*\}$ , dimana  $x^*$  adalah solusi optimal untuk rumusan (3.3). Lebih jauh, kecepatan konvergen dari jaringan saraf tiruan pada rumusan (3.4) proporsional untuk parameter desain  $\lambda$ .

```

12 - Pomega = xinp - alpha * Df + alpha * A' * yinp ;
13 - if Pomega (1) < 0.1
14 -     Pomega (1) = 0.1 ;
15 - elseif Pomega (1) > 10
16 -     Pomega (1) = 10;
17 - end
18 - if Pomega (2) < 0
19 -     Pomega (2) = 0 ;
20 - elseif Pomega (2) > 10
21 -     Pomega (2) = 10;
22 - end
23 - if Pomega (3) < 0
24 -     Pomega (3) = 0 ;
25 - elseif Pomega (3) > 10
26 -     Pomega (3) = 10;
27 - end
28 - if Pomega (4) < 0.1
29 -     Pomega (4) = 0.1 ;
30 - elseif Pomega (4) > 10
31 -     Pomega (4) = 10;
32 - end
33 - dx = lambda * (Pomega - xinp);
34 - dy = lambda * (alpha * (-A * xinp + b));
35 - yp = [dx ; dy];
    
```

Gambar 3.1. Pseudocode untuk permasalahan x pada domain tertentu

### Permasalahan inequality variational

Diketahui permasalahan inequality variational dengan bentuk :

$$(\mathbf{x} - \mathbf{x}^*)^T \mathbf{F}(\mathbf{x}^*) \geq 0, \mathbf{x} \in \Omega_2 \quad (3.5)$$

dimana  $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  adalah continuously differentiable.

$$\Omega_2 = \{x \in \mathbb{R}^n \mid \mathbf{Ax} = \mathbf{b}, x \geq 0\}$$

Dengan kondisi KKT untuk (3.5) kita ketahui bahwa  $x^*$  adalah solusi untuk persamaan (3.5) jika dan hanya jika ada  $y^* \in \mathbb{R}^m$ , sedemikian rupa sehingga  $u^* = [x^*, y^*]^T$  untuk permasalahan :

$$(\mathbf{u} - \mathbf{u}^*)^T \mathbf{G}(\mathbf{u}^*) \geq 0, \mathbf{x} \in \Omega_0 \quad (3.6)$$

dimana  $\Omega_0 = \{\mathbf{u} = [\mathbf{x}, \mathbf{y}]^T \in \mathbb{R}^{n+m} \mid \mathbf{x} \geq 0\}$  dan

$$\mathbf{G}(\mathbf{u}) = \begin{pmatrix} \mathbf{F}(\mathbf{x}) - \mathbf{A}^T \mathbf{y} \\ \mathbf{Ax} + \mathbf{b} \end{pmatrix}$$

Sesuai dengan teorema proyeksi, persamaan (3.6) dapat diformulasikan sebagai

$$\begin{cases} (\mathbf{x} - \alpha \mathbf{F}(\mathbf{x}) + \alpha \mathbf{A}^T \mathbf{y})^* - \mathbf{x} = \mathbf{0} \\ \mathbf{Ax} - \mathbf{b} = \mathbf{0} \end{cases}$$

Sebagai perluasan lain dari jaringan saraf tiruan pada persamaan (3.2), kita mendapatkan model jaringan saraf tiruan untuk menyelesaikan permasalahan (3.5)

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \lambda \begin{pmatrix} (\mathbf{x} - \alpha \mathbf{F}(\mathbf{x}) + \alpha \mathbf{A}^T \mathbf{y})^* - \mathbf{x} \\ \alpha(-\mathbf{Ax} + \mathbf{b}) \end{pmatrix} \quad (3.7)$$

dimana  $\lambda > 0$  adalah konstanta penskalaan.

Teorema 2 :

Asumsikan  $\nabla F(x)$  positif definit pada  $R_+^n = \{x \in R^n \mid x \geq 0\}$ . Maka trayektori dari jaringan saraf tiruan pada rumusan (3.7) dengan titik awal  $(x(t_0), y(t_0)) \in R_+^n \times R^m$  konvergen terhadap titik kesetimbangan  $\{x^*, y^*\}$ , dimana  $x^*$  adalah solusi dari rumusan (3.5). Lebih jauh, waktu konvergen dari jaringan saraf tiruan pada rumusan (3.7) adalah finite.

```

15 - dua = alpha*A'*yinp;
16
17 - p = xinp - alpha * Fx + dua;
18 - p(find(p < 0)) = 0;
19
20 - dx = lambda * (p - xinp);
21 - dy = lambda * alpha * (-A * xinp + b);
22 - yp = [dx ; dy];
    
```

Gambar 3.2. Pseudocode untuk permasalahan inequality variational

## 4. HASIL UJI COBA

### 4.1. Uji Coba Pertama

Pada Uji coba pertama, diberikan contoh yang menampilkan keefektifan dalam memperoleh hasil

Bentuk permasalahan :

Minimize  $f(x)$

Subject to  $Ax = b, x \in \Omega_1$

Dimana

$$f(x) = \frac{3}{2}(x_1^2 + x_2^2) + 2(x_3^2 + x_4^2) - \ln(x_1 x_4) + 3x_1 x_2 + 4x_3 x_4 - 2x_1 - 3x_4$$

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$$b = [1, 1]^T$$

$$\Omega_1 = \{x \in R^4 \mid h \geq x \geq l\}$$

$$l = [0, 1, 0, 0, 0, 1]^T$$

$$h = [10, 10, 10, 10]^T$$

$$\lambda = 1$$

$$\alpha = 1$$

Penyelesaian Dengan Jaringan Saraf Tiruan:

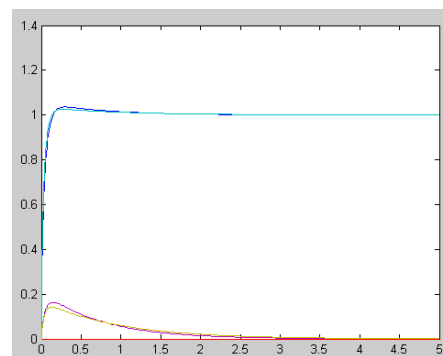
diberikan beberapa inisialisasi dan perkiraan waktu. Hal ini dimaksudkan untuk menunjukkan bahwa pada permasalahan pemrograman kuadratik dengan nilai inisial yang berbeda-beda dan perkiraan waktu yang tetap untuk semua inisialisasi akan menghasilkan grafik trayektori yang konvergen pada satu titik kesetimbangan. Untuk setiap nilai inisial dan perkiraan waktu yang berbeda, memiliki iterasi yang berbeda pula dalam mencapai titik kesetimbangan.

Data nilai inisial dan perkiraan waktu serta banyak iterasi ditunjukkan pada tabel 4.1 berikut:

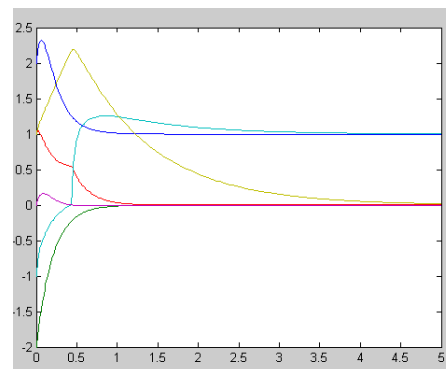
Tabel 4.1. Nilai, tspan, dan banyak iterasi

Uji Coba 1		
Nilai Inisial	tspan	Iterasi
[ 0; 0; 0; 0; 0; 0]	[0;5]	201
[2; -2; 1; -1; 0; 1]	[0;5]	280
[1.5; -1.5; 0.5; -1.5; 0.5; 1.5]	[0;5]	259
[2.25; -1.75; 0.75; -1.75; 0.75; 1.75]	[0;5]	243
[1.75; 2; -2; -1.5; -1.5; -1.75; 1]	[0;5]	271

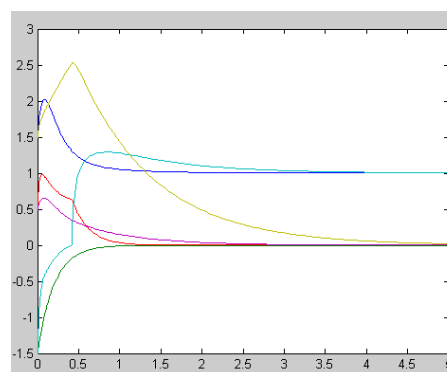
Perubahan nilai penyelesaian pada permasalahan uji coba 1 dapat dilihat pada grafik trayektori berikut :



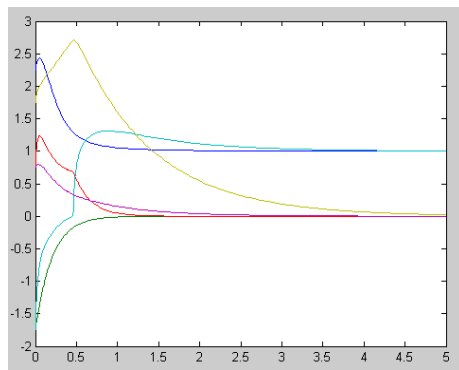
Gambar 4.1. Grafik trayektori  $x_1$



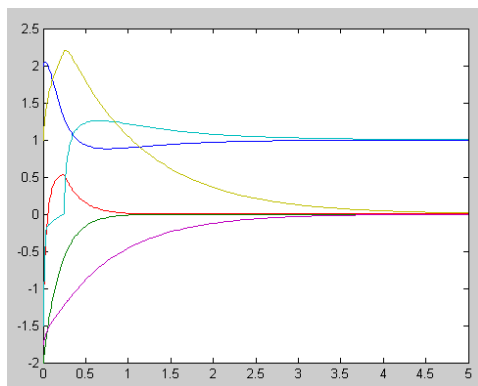
Gambar 4.2. Grafik trayektori  $x_2$



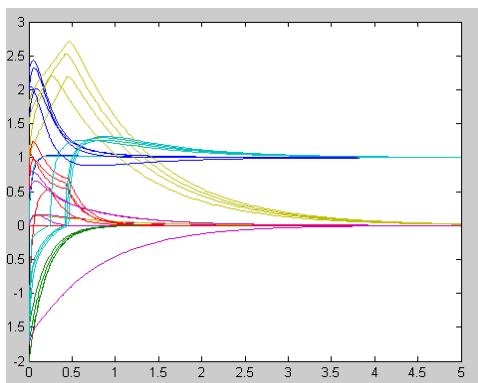
Gambar 4.3. Grafik trayektori  $x_3$



Gambar 4.4. Grafik trayektori  $x_4$



Gambar 4.5. Grafik trayektori  $x_5$



Gambar 4.6. Grafik trayektori semua x

$$F(x) = \begin{bmatrix} 5x_1 + x_1^2 + x_2 + x_3 \\ 5x_1 + 3x_2^2 + 10x_2 + 3x_3 \\ 10x_1^2 + 8x_2^2 + 4x_3 + 3x_3^2 \end{bmatrix} \quad A = (1,1,1)$$

$$b = 6$$

$$\lambda = 40, 20, 10$$

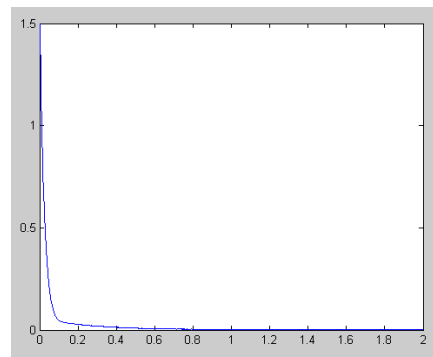
$$\alpha = 1$$

Penyelesaian Dengan Jaringan Saraf Tiruan: diberikan beberapa inialisasi perkiraan waktu dan nilai lamda yang berbeda. Data nilai inisial, nilai lamda dan perkiraan waktu serta banyak iterasi ditunjukkan pada tabel 4.2. berikut:

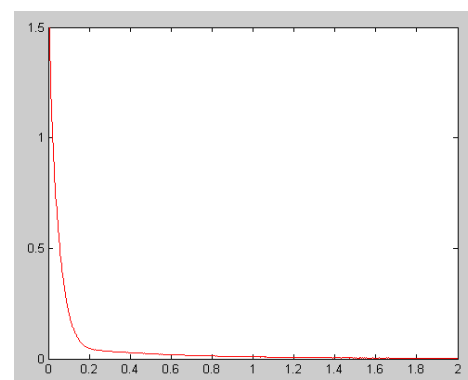
Tabel 4.2. Nilai, tspan, nilai lamda dan banyak iterasi

Uji Coba 2			
Nilai Inisial	lamda	tspan	Iterasi
[0;0;0.5;46.00]	40	[0;2]	1953
[0;0;0.5;46.00]	20	[0;2]	1001
[0;0;0.5;46.00]	10	[0;2]	525

Perubahan nilai penyelesaian pada permasalahan uji coba 2 dapat dilihat pada grafik trayektori berikut :



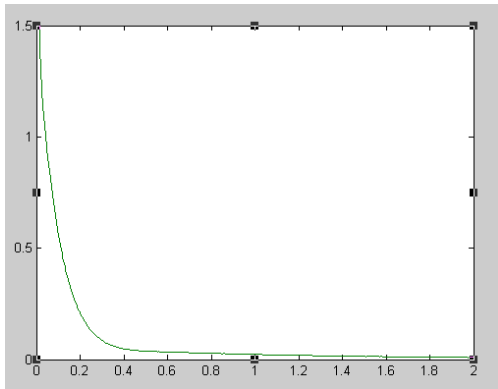
Gambar 4.7. Grafik 2 terhadap tspan dengan lamda = 40



Gambar 4.8. Grafik 2 terhadap tspan dengan lamda = 20

#### 4.2. Uji Coba Kedua

Pada Uji coba kedua, diberikan contoh permasalahan pertidaksamaan variatonal dengan bentuk permasalahan :



Gambar 4.9. Grafik 2 terhadap tspan dengan lamda = 10

## 5. KESIMPULAN

Beberapa hal yang dapat disimpulkan pada penelitian ini antara lain adalah:

1. Pendekatan jaringan saraf tiruan (neural network) sangat sesuai digunakan untuk menyelesaikan permasalahan pemrograman kuadrat, dan khususnya recurrent neural network sangat stabil pada keadaan Lyapunov dan mampu mencapai konvergensi secara lebih cepat untuk solusi optimal di bawah kondisi konveks dari fungsi obyektif;
2. Metode recurrent neural network memiliki struktur single-layer yang sederhana dan dapat diimplementasikan pada komputasi parallel;
3. Metode recurrent neural network tidak memerlukan kondisi kontinuitas Lipschitz dari fungsi obyektif, sehingga metode ini dapat digunakan untuk beragam bentuk permasalahan optimasi konveks dengan pembatas linier.

## DAFTAR PUSTAKA

- [1] Bazaraa, M.S., and Shetty, C.M., *Nonlinear Programming Theory and Algorithms*, John Willey and Sons, Inc., 1990
- [2] Ben-Tal, A., and Nemirovski, A., *Convex Optimization in Engineering: Modelling, Analysis and Algorithm*, Technion-Israel Institute of Technology, 1998
- [3] Bhatti, M.A., *Practical Optimization Methods*, Springer Verlag New York Inc., 2000
- [4] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2006
- [5] Bronson, R., *Operation Research*, McGraw-Hill, Inc., USA, 1982
- [6] Hariadi, V., *Penyelesaian Pemrograman Kuadrat Menggunakan Metode Interior-Point Dengan Fungsi Adaptive Barrier*, Jurusan Teknik Informatika, Fakultas Teknologi Informasi, ITS, Surabaya, 2007
- [7] Hariadi, V., *Aplikasi Variasi Fungsi Adaptif Barrier pada Metode Interior Point untuk*

Penyelesaian Permasalahan Pemrograman Kuadrat, *Proceeding of Conference of Applied Information Technology*, Bandung, 2007

- [8] Hariadi, V., *Otomasi Barrier Term dan Penerapannya pada Penyelesaian Permasalahan Optimasi*, *Prosiding Konferensi Nasional Sistem Informasi*, Jogjakarta, 2008
- [9] Lange, K., *An Adaptive Barrier Method for Convex Programming*, *Methods and Applications of Analysis*, International Press, 1994, pp. 392-402
- [10] Melman, A., and Polyak, R., *The Newton Modified Barrier Method for Quadratic Programming Problems*, *Ann. Oper. Res.*, vol. 62, pp. 465-519, 1996
- [11] Meszaros, C., *Stengths in Interior-Point Algorithms of Quadratic Programming*, *Computer and Automation Research Institut*, 1997
- [12] Rardin, R.L., *Optimization in Operation Research*, Prentice-Hall, Inc., New Jersey, 1998
- [13] Taha, H.A., *Operation Research : An Introduction*, 7<sup>th</sup> Edition, Pearson Education, Inc., 2003
- [14] Wright, M.H., *Primal-Dual Interior-Point Methods*, SIAM : Philadelphia, 1996