

Peningkatan Efisiensi Penggunaan Bahan Bakar Kendaraan Bermotor Dengan Melakukan Pencarian Jarak Terdekat Menggunakan Jaringan Syaraf Tiruan Hopfield Di Wilayah Purwokerto

Fuel Use Efficiency on Motor Vehicle With Do Shortest Path Routing Using Hopfield Neural Networks at Purwokerto Area

Agung Mubyarto dan Hesti Susilawati

*Prodi Teknik Elektro Jurusan Teknik FST Unsoed
Jl. Kampus No.1, Grendeng, Purwokerto*

Abstract— *This proposed research concern with application of continuous Hopfield neural networks for shortest path routing in a given map, in this case Purwokerto city's map had choosen as an example. The method that was used in this research based on the ability of continuous Hopfield neural networks for representing their neuron as choosen vertices in the given map. The result shows that the networks can produce optimal route for such source and destination node pairs. As the product of this research, we have knowledge about the shortest way from one to another location at Purwokerto, so we can increase the efficiency on fuel use by using this route*

Keyword— *fuel efficiency, Hopfield neural networks, shortest path routing.*

PENDAHULUAN

Bahan bakar minyak (BBM) yang digunakan pada kendaraan bermotor termasuk jenis yang tidak dapat diperbaharui. Karena itu perlu cara yang efisien dalam penggunaannya mengingat jumlahnya yang terbatas. Pendekatan yang dilakukan dalam melakukan penghematan adalah dengan mencari rute terpendek yang mungkin ditempuh dari satu lokasi ke lokasi yang lain saat melakukan perjalanan. Semakin pendek rute yang ditempuh maka akan semakin hemat penggunaan BBM.

Pada penelitian ini dilakukan penelitian tentang bagaimana mencari rute terpendek yang dapat ditempuh menggunakan jaringan syaraf tiruan Hopfield kontinyu dengan lokasi pengujian dipilih kota Purwokerto.

Hopfield dan Tank (1982) pertama kali merumuskan suatu bentuk algoritma jaringan syaraf tiruan yang dapat digunakan untuk pengenalan pola dan dapat juga digunakan untuk memecahkan permasalahan optimisasi. Jaringan syaraf tiruan yang dirumuskan tersebut kemudian terkenal sebagai jaringan syaraf tiruan Hopfield (*Hopfield Neural Network*). Jaringan tersebut berbentuk recurrent neural network dengan setiap neuron terhubung pada semua neuron yang lain. Nilai bobot untuk dua neuron mempunyai besar yang sama ,

sedangkan nilai bobot untuk hubungan umpan balik pada dirinya sendiri besarnya nol.

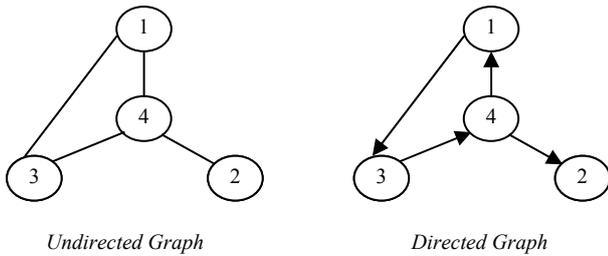
Ali dan Kamoun (1993) pertama kali menggunakan rumusan jaringan syaraf tiruan Hopfield untuk memecahkan permasalahan *shortest path routing*. Rumusan yang mereka buat menjadi dasar bagi penelitian – penelitian selanjutnya untuk riset tentang *shortest path routing* menggunakan jaringan syaraf tiruan Hopfield. Seperti yang dilakukan oleh Ahn (2001), dilakukan penelitian pemakaian jaringan syaraf tiruan Hopfield untuk *shortest path routing*. Pada risetnya mereka memodifikasi persoalan *shortest path routing* yang sebelumnya dibuat oleh Ali dan Kamoun dengan menambahkan dua parameter pada fungsi energi. Penambahan dua buah parameter dimaksudkan untuk mempercepat tercapainya konvergensi perubahan nilai internal neuron.

PERMASALAHAN SHORTEST PATH ROUTING

Shortest path routing berkaitan dengan pencarian jarak terdekat dari suatu titik asal menuju suatu titik tujuan pada suatu jaringan yang ditentukan. Persoalan ini tidak sekadar menyangkut jarak secara fisik antara titik asal dan titik tujuan tetapi juga menyangkut besarnya biaya minimum untuk menempuh jarak tersebut. *Shortest path routing* dapat diterapkan pada berbagai bidang seperti sistem transportasi, *traffic*

routing pada sistem komunikasi, dan perencanaan jalur pada sistem robot.

Permasalahan pencarian jarak terdekat dapat dirumuskan dengan suatu jaringan yang dinyatakan melalui suatu *directed graph* $G = (V, E)$ dimana G adalah himpunan dari sejumlah n titik yang disebut sebagai verteks dan E menyatakan himpunan sejumlah m arcs atau edges dengan $m \leq n^2$. Contoh *directed graph* dan *undirected graph* dapat dilihat pada Gambar 1.



Gambar 1 Directed dan undirected graph.

Urutan verteks $v_1, v_2, v_3, \dots, v_n$ membentuk suatu path sepanjang $n-1$, jika terdapat edge v_i ke v_{i+1} untuk $1 \leq i \leq n$. Simpul N menyatakan jumlah simpul yang ada pada jaringan, sedangkan edges A menyatakan hubungan antar simpul yang diwakili dengan pasangan simpul (i, j) sehingga $i, j \in V$.

Suatu edge (i, j) adalah edge dengan simpul awal i dan simpul akhir j . Setiap edges mempunyai bobot atau *cost coefficients* yang dinyatakan dengan matriks bobot C , sedangkan untuk dua simpul yang tidak mempunyai edges atau tidak terhubung diberikan nilai bobot nol. Secara umum bobot dapat bernilai negatif ataupun positif. Nilai negatif menyatakan perolehan (*gain*) sedangkan nilai positif menyatakan rugi - rugi (*loss*). Nilai - nilai bobot pada suatu jaringan G diwakili oleh sebuah *cost coefficient* matriks $C = [c_{ij}]$. Matriks C dapat berbentuk simetris untuk jaringan yang mempunyai bobot $c_{ij} = c_{ji}$, selain itu dapat juga berbentuk tak simetris jika nilai bobot untuk arah yang berlawanan mempunyai nilai yang tidak sama. Representasi dari edges menggunakan matriks biner $V = [v_{ij}]$ ordo $n \times n$.

Matriks tersebut hanya mengandung nilai 0 dan 1. Untuk dua simpul (i, j) yang saling terhubung diberikan nilai 1, sedangkan untuk dua simpul yang tak terhubung diberikan nilai 0. Diasumsikan tidak terjadi loop pada suatu simpul i terhadap dirinya sendiri sehingga edges simpul (i, i) tidak mempunyai nilai dan dianggap bernilai nol. Karena itu diagonal utama matriks V selalu bernilai nol. Atau dapat didefinisikan sebagai berikut :

$$v_{ij} = \begin{cases} 1 & \text{jika simpul } ij \text{ terhubung an} \\ 0 & \text{jika simpul } ij \text{ tak terhubung an} \end{cases} \quad (1)$$

Dengan menganggap elemen diagonal utama tidak ada, maka representasi matriks V yang disederhanakan hanya mengandung $n(n-1)$ elemen biner. Suatu path dari titik asal ke titik tujuan dapat didefinisikan sebagai $PSD = \{s, n1, n2, n3, \dots, d\}$ dengan s adalah asal (*source*), n_i adalah edges ke- i , dan d menyatakan tujuan (*destination*). Berdasar representasi edges, *shortest path routing* dapat dirumuskan sebagai permasalahan *linear programming* untuk meminimalisasi dari penjumlahan *cost coefficient* yang berada pada suatu path tertentu.

Rumusan untuk *shortest path routing* adalah: minimalisasi dari

$$\sum_{i=1}^n \sum_{j=1}^n C_{ij} v_{ij} \quad (2)$$

untuk

$$\sum_{j=1, j \neq i}^n v_{ij} - \sum_{j=1, j \neq i}^n v_{ji} = \phi_i \quad (3)$$

dimana

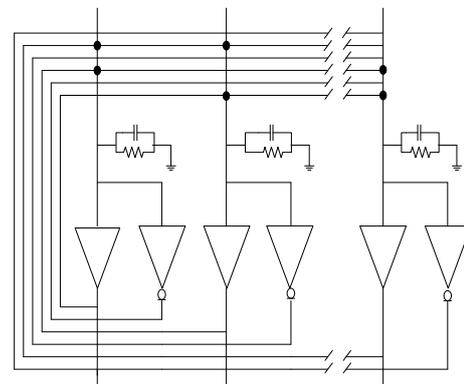
$$v_{ij} \in \{0, 1\}, i \neq j; i, j = 1, 2, \dots, n \quad (4)$$

$$\phi_i = \begin{cases} 1, & \text{jika } i = s \\ 0, & \text{jika } i \neq s \text{ \& } i \neq d \\ -1, & \text{jika } i = d \end{cases} \quad (5)$$

Dalam perumusan ini, jumlah variabel keputusan adalah $n(n-1)$, edges dari simpul i ke j adalah *shortest path* jika nilai $v_{ij} = 1$. Persamaan (3) dan (5) merupakan keangan dari persamaan (2) yang menyatakan bahwa path berasal dari satu simpul dan berakhir pada satu simpul yang lainnya.

JARINGAN SYARAF TIRUAN HOPFIELD

C. Arsitektur JST Hopfield



Gambar 2 Arsitektur JST Hopfield.

Jaringan ini tersusun atas satu lapis jaringan dengan umpan balik seperti dapat dilihat pada Gambar 2. Parameter W adalah bobot antar neuron, V adalah keluaran jaringan, U adalah aktivasi sel, I adalah nilai masukan dari luar, sedangkan R dan C adalah nilai untuk resistor dan kapasitor. Pada HNN setiap unit pemroses mempunyai suatu masukan dari luar dan bobot yang menghubungkan dengan neuron yang lainnya dalam jaringan tersebut. Hubungan antar neuron direpresentasikan dalam bentuk matriks bobot $W = [W_{ij}]$ dengan orde $n \times n$, dengan $w_{ij} = w_{ji}$ dan $w_{ij} = 1/R_{ij}$. Hubungan dari keluaran positif disebut sebagai *excitatory connections* dan hubungan dari keluaran negatif disebut sebagai *inhibitory connections*.

Jika dimisalkan U_i dan V_i menyatakan masukan dan keluaran terhadap neuron ke- i , maka masukan ke neuron ke- i adalah:

$$U_i = \sum_{j=1}^n W_{ij} V_j + I_i \quad (6)$$

dengan I_i adalah masukan luar. Keluaran dari neuron dinyatakan dengan masukan dari jaringan dan fungsi alih dari neuron.

Dari Gambar 2 tampak bahwa setiap neuron menerima masukan dari luar (*external input*) yang disebut sebagai bias yang merupakan data aktual yang diberikan oleh pengguna kepada jaringan. Dari Gambar 2 dapat dibuat suatu persamaan dinamika JST Hopfield sebagai berikut :

$$C_i \frac{dU_i}{dt} = \sum_{j=1}^n W_{ij} V_j - \frac{U_i}{R} + I_i \quad (7)$$

dengan membagi dengan C , dan mendefinisikan W_{ij}/C sebagai W_{ij} , serta I_i/C sebagai I_i maka diperoleh persamaan berikut:

$$\frac{dU_i}{dt} = -\frac{U_i}{\tau} + \sum_{j=1}^n W_{ij} V_j + I_i \quad (8)$$

dengan:

$\tau = RC$ = adalah konstanta waktu dari rangkaian
n = banyaknya neuron pada jaringan.

D. Fungsi Energi JST Hopfield

Hopfield membuktikan bahwa jaringan syaraf tiruan yang ditemukannya akan konvergen pada suatu titik stabil (pola aktivasi dari unit-unit) dengan menggunakan suatu fungsi energi pada sistem tersebut. Suatu fungsi energi adalah fungsi terkendala (*bounded*) dan adalah suatu fungsi yang tidak bertambah (*nonincreasing*) dari keadaan sistem.

Pada suatu jaringan syaraf tiruan keadaan sistem tersebut adalah vektor aktivasi dari tiap unit. Jika fungsi energi dapat diperoleh pada suatu iterasi jaringan syaraf tiruan, jaringan akan konvergen pada himpunan aktivasi yang stabil. Suatu fungsi energi untuk jaringan syaraf tiruan Hopfield diberikan sebagai berikut :

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n W_{ij} V_i V_j - \sum_{i=1}^n I_i V_i \quad (9)$$

dengan :

V_i = adalah aktivasi atau keluaran unit U_i
 I_i = adalah masukan luar unit U_i
 W_{ij} = adalah bobot antara unit U_i dan unit U_j

$$\Delta E = - \left[\sum_j V_j W_{ij} + I_i \right] \Delta V_i \quad (10)$$

Jaringan akan konvergen pada konfigurasi stabil, yaitu pada minimum dari fungsi energi selama $\Delta E \leq 0$, atau:

$$\frac{d}{dt} E \leq 0 \quad (11)$$

Dengan mengacu pada persamaan (8) maka diperoleh persamaan dinamika neuron sebagai fungsi turunan persamaan energi terhadap perubahan nilai V_{ij} . Persamaan tersebut adalah sebagai berikut :

$$\frac{dU_i}{dt} = -\frac{U_i}{\tau} - \frac{dE}{dV_i} \quad (12)$$

Fungsi energi tersebut nilainya tidak akan bertambah, sehingga akan konvergen terhadap suatu nilai yang ditentukan. Evolusi waktu sistem tersebut adalah gerakan state mencari dan menuju minimal E selanjutnya berhenti di titik tersebut. Konfigurasi stabil diperoleh dimana sinyal keluaran sel syaraf V_i tidak berubah lagi terhadap waktu dan bilamana nilai fungsi energi E telah konvergen pada proses iterasi jaringan syaraf tiruan Hopfield kontinyu.

E. Pemodelan Pencarian Jarak Terdekat menggunakan JST Hopfield

Untuk menyelesaikan permasalahan pencarian rute terpendek menggunakan JST Hopfield, dibuat suatu model dalam bentuk matriks orde $n \times n$ dengan elemen diagonal dihilangkan. Setiap elemen pada matriks dinyatakan dengan neuron yang dideskripsikan dengan indeks ganda (i,j), sehingga komputasi jaringan membutuhkan $n(n-1)$ neurons, dan neuron pada lokasi (i,j) mempunyai karakteristik keluaran V_{ij} sebagai berikut :

$$V_{ij} = \begin{cases} 1, & \text{jika jalur antara node } i \text{ dan } j \text{ ada di rute} \\ 0, & \text{jika bukan merupakan bagian rute} \end{cases} \quad (13)$$

kemudian matriks r_{ij} yang menyatakan *link connections* antar node i dan j

$$r_{ij} = \begin{cases} 1, & \text{jika node } i \text{ dan } j \text{ terhubung} \\ 0, & \text{jika node } i \text{ dan } j \text{ tak terhubung} \end{cases} \quad (14)$$

dan matriks C_{ij} yang menyatakan jarak antara node i dan j dimana untuk dua titik yang tidak terhubung diasumsikan mempunyai jarak nol.

F. Fungsi energi untuk pencarian jarak terdekat

Untuk melakukan proses optimisasi, perlu didefinisikan terlebih dahulu rumusan fungsi energi yang akan mengarahkan jaringan pada suatu keadaan stabil saat kondisi konvergensi diperoleh. Keadaan stabil tersebut adalah ketika hasil yang didapatkan dapat menghubungkan pasangan titik asal dan titik tujuan dengan biaya terkecil. Pada tulisan ini, digunakan rumusan fungsi energi seperti yang digunakan oleh Ali dan Kamoun (1993) yaitu :

5) *Fungsi energi* yang meminimalkan jumlah total biaya dengan menghitung biaya dari jalur yang dilewati saja.

$$E_1 = \frac{\mu_1}{2} \sum_{i=1}^n \sum_{j=1}^n C_{ij} V_{ij} \quad (15)$$

6) *Fungsi energi kedua* yang menjaga agar komputasi tidak menghitung jalur yang tidak muncul antara dua titik.

$$E_2 = \frac{\mu_2}{2} \sum_{i=1}^n \sum_{j=1}^n \rho_{ij} V_{ij} \quad (16)$$

7) *Fungsi energi ketiga* untuk menjaga agar jika suatu titik terpilih sebagai jalur masukan, maka jalur keluarnya juga harus melalui titik tersebut

$$E_3 = \frac{\mu_3}{2} \sum_{i=1}^n \left(\sum_{j=1, j \neq i}^n V_{ij} - \sum_{j=1, j \neq i}^n V_{ji} \right)^2 \quad (17)$$

Fungsi ini akan mempunyai nilai nol jika syaratnya terpenuhi.

8) *Fungsi energi keempat* berfungsi untuk memaksa jaringan agar konvergen pada salah satu titik dari hypercube 2^n .

$$E_4 = \frac{\mu_4}{2} \sum_{i=1}^n \sum_{j=1, j \neq i}^n V_{ij} (1 - V_{ij}) \quad (18)$$

Fungsi ini akan mempunyai nilai nol jika syaratnya terpenuhi.

9) *Fungsi energi kelima* berfungsi untuk membuat rute berawal dan berakhir pada suatu titik yang sama.

$$E_5 = \frac{\mu_5}{2} (1 - V_{ds}) \quad (19)$$

Fungsi ini akan bernilai nol jika neuron pada lokasi (d,s) bernilai satu.

Walaupun rute antara titik tujuan ke titik awal bukan merupakan bagian dari pemecahan optimisasi ini, tetapi hal ini dilakukan untuk menjaga keaslian dari jalur yang dipilih agar setelah titik tujuan diperoleh maka optimisasi tidak bergerak mencari titik yang lain, tetapi kembali pada titik asalnya. Akhirnya jumlah total keseluruhan fungsi energi adalah :

$$E = E_1 + E_2 + E_3 + E_4 + E_5 \quad (20)$$

G. Matriks Bobot dan Bias

Salah satu tugas yang harus dilakukan untuk dapat melakukan optimisasi ini adalah membentuk matriks bobot W_{ij} yang berfungsi untuk menjaga agar jalur yang tidak valid tidak muncul, dan memunculkan jalur yang valid dalam rute dari titik sumber sampai titik tujuan. Misalnya adalah untuk menjaga agar pada satu kolom atau baris hanya ada satu neuron saja yang melakukan aktivasi (*firing*). Untuk melaksanakan hal tersebut digunakan fungsi *delta Kronecker* δ_{ij} , dengan definisi nilai *delta Kronecker* akan sama dengan satu jika nilai $i=j$ dan nilai *delta Kronecker* akan sama dengan nol jika i tidak sama dengan j .

Neuron pada jaringan ini adalah mewakili jalur antara titik i dan titik j sehingga mempunyai dua indeks U_{ij} . Karena itu bobot yang menyatakan hubungan antara dua neuron U_{ij} dan neuron U_{kl} mempunyai empat indeks menjadi $W_{ij,kl}$.

Dengan mengacu pada persamaan fungsi energi dan persamaan dinamika JST Hopfield maka dapat ditulis persamaan dinamika neuron dengan indeks ganda sebagai berikut:

$$\frac{dU_{ij}}{dt} = -\frac{U_{ij}}{\tau} + \sum_{k=1}^n \sum_{l=1}^n W_{ij,kl} V_{ij} + I_{ij} \quad (21)$$

$$\frac{dU_{ij}}{dt} = -\frac{U_{ij}}{\tau} - \frac{dE}{dV_{ij}} \quad (22)$$

$$f(\lambda u_{ij}) = \frac{1}{2} (1 + \tanh(\lambda u_{ij})) \quad (23)$$

Dengan melakukan substitusi persamaan (20) ke persamaan (22) diperoleh persamaan untuk dinamika neuron sebagai berikut:

$$\begin{aligned} \frac{dU_{ij}}{dt} = & -\frac{U_{ij}}{\tau} - \frac{\mu_1}{2} C_{ij} (1 - \delta_{id} \delta_{js}) - \frac{\mu_2}{2} r_{ij} (1 - \delta_{id} \delta_{js}) \\ & - \mu_3 \sum_{k=1, k \neq i}^n (V_{ik} - V_{ki}) + \mu_3 \sum_{k=1, k \neq j}^n (V_{jk} - V_{kj}) - \frac{\mu_4}{2} (1 - 2V_{ij}) \\ & + \frac{\mu_5}{2} \delta_{id} \delta_{js} \end{aligned} \quad (24)$$

dengan membandingkan antara persamaan (24) dengan (21) maka dapat diperoleh rumusan untuk matriks bobot.

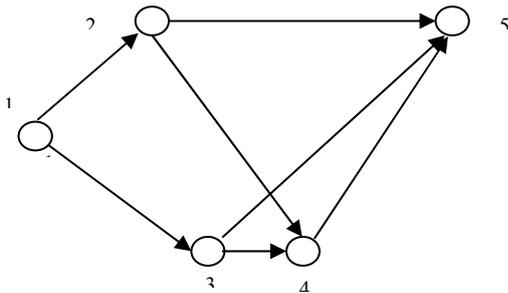
$$W_{ij,kl} = \mu_4 \delta_{ik} \delta_{jl} - \mu_3 \delta_{ik} \delta_{jl} + \mu_3 \delta_{il} \delta_{jk} \quad (25)$$

HASIL PERCOBAAN

Pada penelitian ini dilakukan dua kali percobaan. Pertama jaringan diuji untuk sembarang graf, selanjutnya jaringan diuji untuk melakukan pencarian jarak terdekat pada graf yang merupakan analogi peta kota Purwokerto.

H. Percobaan untuk sembarang graf

Dari Gambar 3 yang berupa *directed graph* dengan 5 simpul dapat dibuat 2 buah matriks yaitu matriks sinaptik *r* dan matriks jarak *C* yang dapat dilihat pada Tabel 1 dan Tabel 2. Matriks *r* menyatakan keadaan apakah terdapat hubungan antara simpul *i* dan simpul *j* atau tidak terdapat hubungan. Sedangkan matriks *C* menyatakan jarak antara simpul *i* dan simpul *j* dalam suatu satuan tertentu.



Gambar 3 Graf dengan lima simpul.

TABEL 1 MATRIKS SINAPTIK *r* UNTUK GRAF PADA GAMBAR 3

r	1	2	3	4	5
1	x	0	0	1	1
2	1	x	0	0	0
3	1	1	X	0	0
4	1	1	1	x	0
5	1	1	1	1	x

TABEL 2 MATRIKS JARAK *C* UNTUK GRAF PADA GAMBAR 3

C	1	2	3	4	5
1	x	0.66	0.53	-	-
2	-	X	0.44	1	1
3	-	-	x	0.66	1
4	-	-	-	x	0.44
5	-	-	-	-	x

Matriks jarak *C* masih mewakili nilai sebenarnya dari panjang antar node untuk graf pada Gambar 3. Untuk dapat digunakan dalam simulasi pencarian jarak terdekat maka perlu dilakukan normalisasi nilai – nilai elemennya. Sebagai acuan dipilih nilai elemen yang terbesar yaitu 4.5. Sehingga diperoleh matriks *C* yang dinormalisasi.

TABEL 3 MATRIKS JARAK *C* YANG DINORMALISASI

C	1	2	3	4	5
1	x	3	2.4	-	-
2	-	x	4	4.5	4.5
3	-	-	X	3	4.5
4	-	-	-	x	2
5	-	-	-	-	x

Dipilih pasangan simpul $P_{sd} = (1,5)$, dengan suatu asumsi awal bahwa simulasi akan memberikan hasil salah satu dari rute berikut ini :

- Rute 1 : 1 2 5
- Rute 2 : 1 2 3 5
- Rute 3 : 1 2 3 4 5
- Rute 4 : 1 4 5

Setelah dilakukan simulasi, maka diperoleh hasil yang salah satunya adalah seperti pada Tabel 4.

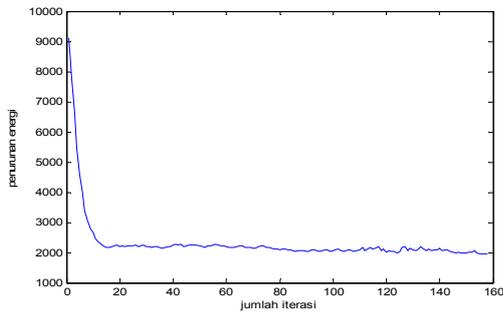
TABEL 4 MATRIKS *V* HASIL SIMULASI UNTUK PASANGAN SIMPUL $P_{sd}(1,5)$

V	1	2	3	4	5
1	x	0	0.9557	0	0
2	0	x	0	0	0
3	0	0	x	0	0.8772
4	0	0	0	x	0
5	1	0	0	0	x

Dari hasil simulasi, didapatkan beberapa hasil dengan klasifikasi mampu membentuk rute dan ada juga yang tidak berhasil membentuk rute pada saat konvergensi tercapai. Tabel 4 merupakan salah satu hasil simulasi yang berhasil membentuk suatu rute dari simpul asal 1 menuju simpul tujuan 5. Simulasi tersebut dilakukan dengan nilai parameter sebagai berikut:

- Faktor perolehan fungsi alih sigmoid biner (λ)=30
- Step size = 0.00001
- Jumlah iterasi saat konvergensi tercapai = 158

Proses selesai dengan hasil rute terpilih dari simpul 1 ke simpul 5 adalah rute yang melalui simpul 1–3–5 dengan panjang 6.9 satuan.



Gambar 4 Peluruhan fungsi energi untuk graf pada Gambar 3.

Peluruhan nilai fungsi energi dapat dilihat seperti pada Gambar 4.

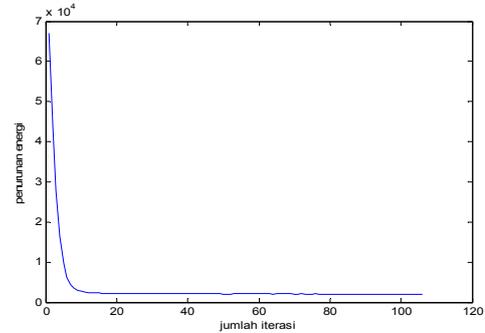
Proses *shortest path routing* dari node 1 menuju node 5 selesai dilakukan dalam 158 iterasi, dengan nilai energi awal sebesar 9126 satuan dan berakhir pada nilai energi sebesar 1964 satuan. Grafik peluruhan energi untuk P_{sd} 1-5 dapat dilihat seperti pada Gambar 4.

I. Percobaan pada graf peta Purwokerto

Setelah pengujian dengan suatu graf sederhana, akan diujikan suatu contoh kasus dengan mengambil lokasi di Purwokerto. Titik asal yang dipilih adalah pertigaan jalan Kampus dengan jalan Prof. Boenyamin. Titik tujuan adalah pertemuan antara jalan Prof Boenyamin, jalan Dr. Angka, jalan Ovis dan jalan Soeharso. Pengujian yang dilakukan dengan asumsi sudah diketahui arah yang akan ditempuh, sehingga pada awalnya akan ada beberapa jalur yang dapat ditempuh. Jaringan akan diuji untuk memilih jalur yang terpendek dari beberapa jalur yang disediakan.

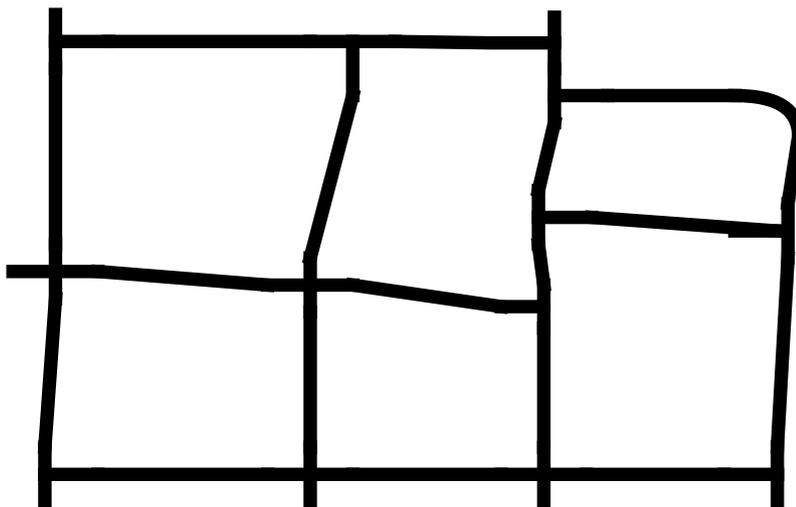
Pengujian memberikan hasil bahwa jaringan dapat menemukan jalur dari titik asal menuju titik tujuan, dan jalur tersebut merupakan jalur terpendek dibanding jalur lain yang dapat ditempuh.

Grafik peluruhan fungsi energi jaringan Hopfield seperti tampak pada Gambar 5 menunjukkan bahwa nilai fungsi energi bergerak dari nilai yang tinggi sekitar 7×10^4 menuju pada energi minimum dalam 106 iterasi.



Gambar 5 Grafik peluruhan fungsi energi untuk graf pada Gambar 6.

Matriks rute jalur yang ditempuh dapat dilihat pada Tabel 5. Pada elemen matriks tampak (yang dicetak tebal) pada elemen (1,2) (2,4) (4,5) (5,12) dan (12,1) mempunyai nilai yang lebih besar dari 0.5 yang menunjukkan bahwa jaringan memilih untuk menempuh rute dari titik 1 ke 2 dilanjutkan dari titik 2 ke 4 dilanjutkan dari 4 ke 5 dilanjutkan dari 5 ke 12 yang merupakan jalur terpendek dari beberapa rute yang mungkin. Pada elemen (12,1) bernilai satu bukan berarti jaringan kembali ke titik 1, tetapi dimaksudkan agar setelah titik tujuan tercapai maka jaringan mengunci untuk tidak melanjutkan ke titik yang lain.



Gambar 6 Peta Purwokerto untuk pengujian jaringan.

TABEL 5 MATRIKS V HASIL SIMULASI UNTUK GRAF PETA PURWOKERTO

V	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0.84609	2.40E-29	6.51E-30	2.88E-25	1.86E-28	6.80E-30	4.13E-29	1.80E-29	1.48E-29	1.04E-26	7.47E-21	5.22E-30
2	3.35E-35	0	2.43E-06	0.76214	3.14E-26	6.19E-27	7.44E-27	2.32E-29	4.33E-27	4.38E-29	1.07E-24	2.29E-23	4.06E-29
3	4.77E-40	8.23E-42	0	3.00E-35	1.91E-31	3.29E-34	5.78E-34	4.52E-34	7.27E-34	4.19E-34	1.11E-32	6.36E-28	1.10E-16
4	3.67E-39	1.99E-40	1.06E-09	0	0.80347	1.16E-33	5.31E-33	3.76E-32	2.69E-31	7.19E-32	2.52E-31	1.26E-23	9.15E-33
5	6.72E-43	6.32E-47	1.97E-38	1.23E-40	0	5.22E-37	2.46E-39	3.24E-37	3.93E-37	9.79E-37	8.60E-36	0.99336	2.84E-39
6	1.34E-14	2.26E-42	1.48E-35	5.54E-37	1.10E-31	0	8.95E-35	2.90E-14	7.12E-35	6.10E-35	2.29E-33	1.46E-26	1.60E-35
7	2.62E-41	3.35E-42	9.43E-36	2.04E-37	5.73E-32	5.10E-14	0	4.66E-35	8.72E-15	3.17E-35	1.01E-33	2.45E-25	9.68E-36
8	5.54E-42	1.29E-43	9.84E-36	1.05E-36	9.67E-07	5.84E-35	5.71E-35	0	4.96E-35	5.44E-35	3.18E-11	1.33E-25	1.25E-35
9	6.52E-39	2.28E-41	6.62E-36	1.59E-38	1.72E-31	7.03E-35	8.49E-35	8.53E-13	0	1.76E-13	1.74E-33	3.21E-25	1.63E-35
10	1.96E-41	1.54E-40	1.29E-35	6.61E-38	4.09E-32	6.85E-35	9.47E-35	1.14E-34	1.49E-34	0	3.07E-10	2.61E-28	1.89E-35
11	1.29E-41	2.27E-43	8.84E-37	1.41E-37	1.91E-31	2.71E-36	1.98E-36	2.74E-36	2.33E-36	3.08E-36	0	0.000266	8.43E-37
12	1	1.39E-52	3.49E-43	1.64E-44	4.23E-41	7.32E-43	5.49E-43	2.44E-44	3.11E-44	6.85E-44	2.68E-42	0	3.27E-23
13	1.50E-39	1.54E-42	5.29E-35	3.47E-37	1.79E-30	1.75E-34	3.73E-34	2.41E-34	3.83E-34	2.01E-34	6.50E-33	4.79E-26	0

KESIMPULAN

Dari penelitian yang sudah dilakukan dapat diperoleh beberapa kesimpulan yaitu.

- 1) Jaringan syaraf tiruan Hopfield dapat digunakan untuk melakukan optimisasi pencarian jarak terdekat.
- 2) Pengujian pada wilayah Purwokerto menunjukkan bahwa jaringan dapat diterapkan dengan baik untuk melakukan pencarian jarak terdekat.

Dengan memperoleh pengetahuan mengenai jarak terdekat dari satu pasangan lokasi asal dan lokasi tujuan maka dapat diketahui rute yang ditempuh. Selanjutnya pemilihan rute terdekat dapat meningkatkan efisiensi penggunaan kendaraan bermotor.

DAFTAR PUSTAKA

Fausset, L. 1994. *Fundamentals of Neural Networks Architectures, Algorithms and Application*. Prentice Hall, Englewood Cliffs, New Jersey

Wang, J. 1996. *A Recurrent Neural Network for solving the Shortest Path Problem*. IEEE Transactions on Circuit and Systems – 1: Fundamentals Theory and Applications, 43, 482- 486.

Wang, J. and Y. Xia. 2000. *A Discrete-Time Recurrent Neural Network for Shortest-Path Routing*. IEEE Transactions on Automatic Control, 45, 2129-2134.

Araujo, F., Ribeiro, B. and L. Rodrigues. 2000. *A Neural Network for Shortest Path Computation*. Departamento de Informatica Faculdade de Ciencias da Universidade. Lisboa.

Ahn, C.W., Ramakrishna, R.S., Choi, I.C. and C.G. Kang. 2001. *Shortest Path Routing Algorithm using Hopfield Neural Network*. Electronics Letter, 37, 1176-1178.

Kojic, N., Reljin, I. and B. Reljin. 2006. *Neural Network for Optimization of Routing in Communications Networks*. Facta Universitatis SER.Elec.Energ, 19, 317-329.

Rif'an, M. 2002. *Aplikasi Algoritma Genetik untuk Shortest Path Routing (Kasus di Kodya Yogyakarta)*. Tesis Program Studi Teknik Elektro Universitas Gadjah Mada.

Mubyarto, A. 2008. *Aplikasi Jaringan Syaraf Tiruan Untuk Pencarian Jarak Terdekat di Kodya Yogyakarta*. Tesis Program Studi S-2 Teknik Elektro Universitas Gadjah Mada