

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

BAB IV

ANALISA DAN PEBAHASAN

Pada tahap ini merupakan tahap dimana akan dijelaskan penerapan algoritme *harmony search* untuk menyelesaikan masalah jalur terpendek. Analisa merupakan tahap kegiatan dalam penentuan tindakan atau keputusan yang akan menentukan implementasi pada MATLAB.

4.1 Analisa Kebutuhan Data

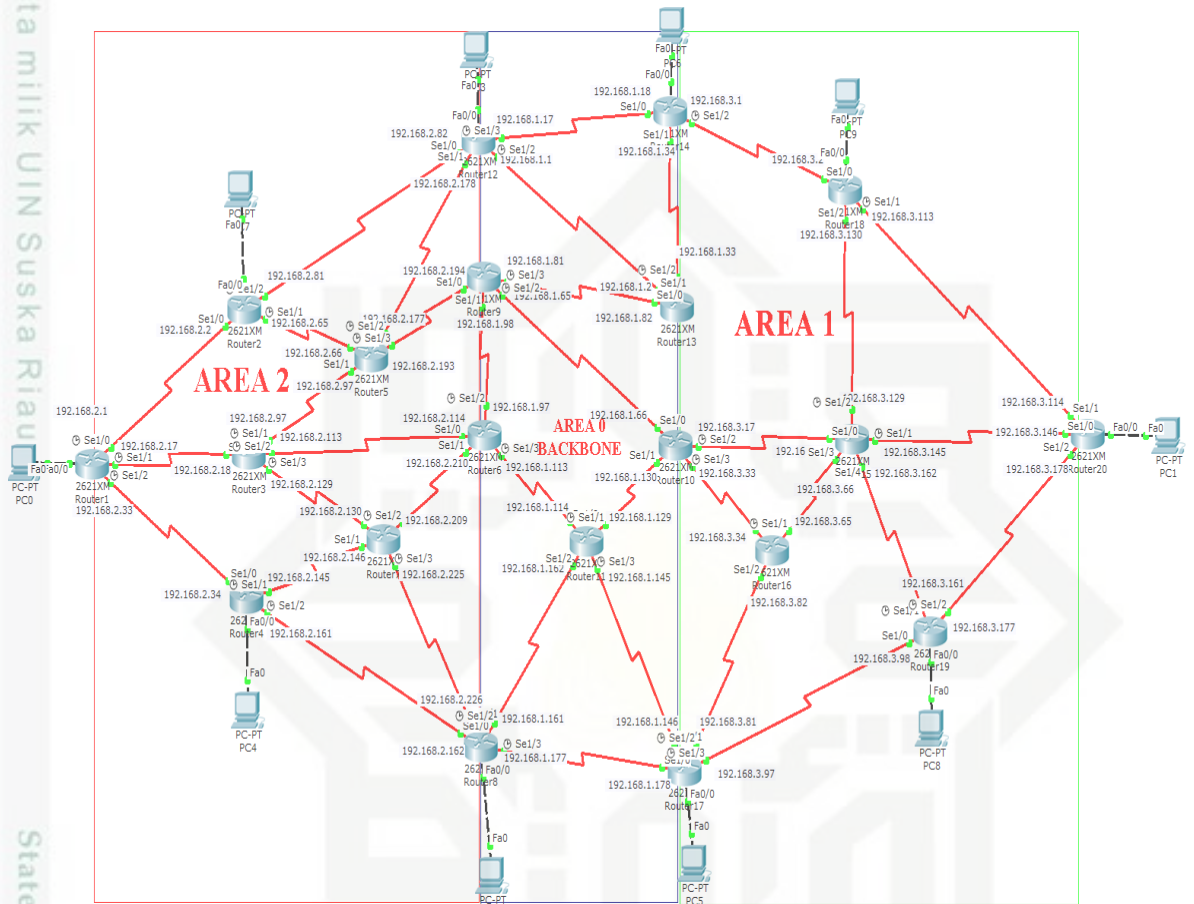
Pada penelitian ini menggunakan data yang diperoleh dari hasil simulasi aplikasi *Cisco Packet Tracer*, dengan menggunakan topologi *mesh* serta menerapkan dan mengkonfigurasi *protocol routing OSPF*. Adapun pada topologi ini menerapkan 20 *router* dan 36 *link* yang saling terhubung. Pemilihan jalur terpendek dengan meminimumkan *cost* dan *metric* pada setiap jalur antar *router*, pada penelitian ini bobot pada jalur routing adalah *delay* dan *bandwidth*.

Bobot *delay* dan *bandwidth* pada jalur didapat dari *link interface* pada setiap *router* yang terhubung dengan menggunakan kabel *serial*, bobot *delay* dan *bandwidth* yang dimaksud disini bukanlah bobot dalam bentuk fisik melainkan bobot dalam bentuk logic. Data dapat dilihat dengan membuka kolom CLI pada *router* dan melakukan perintah “show interface” seperti pada Gambar 4.1 berikut :

```
Router>show int
FastEthernet0/0 is up, line protocol is up (connected)
Hardware is Lance, address is 0001.4363.a001 (bia 0001.4363.a001)
Internet address is 192.168.3.193/28
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
ARP type: ARPA, ARP Timeout 04:00:00,
Last input 00:00:08, output 00:00:05, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: fifo
Output queue :0/40 (size/max)
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 51 bits/sec, 0 packets/sec
 0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
 0 input packets with dribble condition detected
96 packets output, 6144 bytes, 0 underruns
 0 output errors, 0 collisions, 1 interface resets
 0 babbles, 0 late collision, 0 deferred
 0 lost carrier, 0 no carrier
--More--
```

Gambar 4.1 Bandwidth dan Delay pada router

Adapun rancangan topologi yang akan digunakan pada penelitian ini dapat dilihat pada gambar 4.1 berikut ini :



Gambar 4.2 Topologi jaringan mesh

Pada penelitian ini data yang digunakan adalah bobot yang terdapat pada setiap *link interface* pada setiap *router* yang saling terhubung, beberapa data yang harus dikumpulkan dan digunakan pada penelitian ini antara lain *bandwidth* dan *delay*. Topologi yang digunakan adalah topologi *mesh* dengan menerapkan fungsi tiga area salah satunya adalah area 0 atau area *backbone* yang diterapkan pada simulasi *cisco packet tracer*. Data ini merupakan data yang digunakan oleh protokol OSPF dalam menentukan rute. Data dapat dilihat pada tabel 4.1 berikut ini :

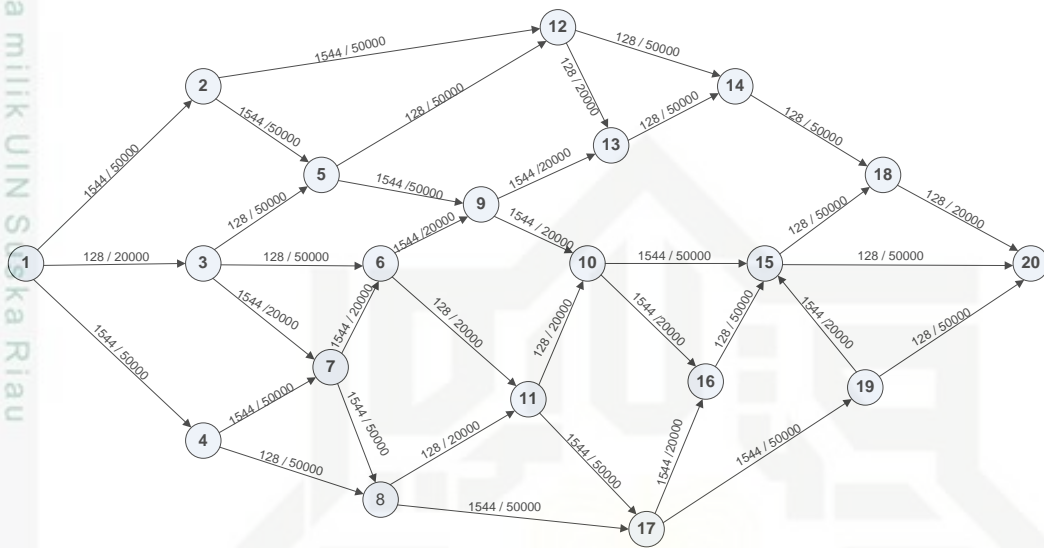
Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Tabel 4.1 Data bandwidth pada router

Simpul Awal (i)	Simpul Tujuan (j)	Bandwidth/ kbps	Delay/ms
1	2	1544	50000
	3	128	20000
	4	1544	50000
2	5	1544	50000
	12	1544	50000
3	5	128	50000
	6	128	50000
	7	1544	20000
4	7	1544	50000
	8	128	50000
5	9	1544	50000
	12	128	50000
6	9	1544	20000
	11	128	20000
7	6	1544	20000
	8	1544	50000
8	11	128	20000
	17	1544	50000
9	10	1544	20000
	13	1544	20000
10	15	1544	50000
	16	1544	20000
11	10	128	20000
	17	1544	50000
12	13	128	20000
	14	128	50000
13	14	128	50000
14	18	128	50000
15	18	128	50000
	19	1544	20000
	20	128	50000
16	15	128	50000
17	16	1544	20000
	19	1544	50000
18	20	128	20000
19	20	128	50000

Berikut adalah contoh data *bandwidth* dan *delay* pada jaringan komputer yang diterapkan pada graf berarah, dapat dilihat gambar 4.2 berikut ini :



Gambar 4.3 data pada graf

Pada gambar 4.2 dapat dilihat bobot pada setiap jalur untuk menghubungkan titik awal R1 menuju titik tujuan R20 dengan bobot berupa *bandwidth* dan *delay* dalam bentuk graf berarah.

4.2 Analisa Algoritme *Harmony Search*

Pengujian untuk menyelesaikan masalah jalur terpendek pada jaringan komputer yang tepat bisa diselesaikan menggunakan algoritme *harmony search*. Dibagian analisa ini akan dijelaskan bagaimana mengimplementasikan algoritme *harmony search* untuk mencari jalur terbaik pada jaringan antar *router*. Dalam penelitian ini algoritme HS digunakan untuk mencari *cost* dan *metric* minimum dari titik awal ke titik tujuan. Untuk masalah jalur terpendek menggunakan algoritme HS, ada beberapa langkah modifikasi salah satunya pada improvisasi rute baru (Santosa.2015).

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Berikut adalah contoh perhitungan manual algoritme HS dalam menentukan rute dengan meminimasi nilai *cost*, menggunakan data pada tabel 4.1. Terdapat lima tahap pada algoritme *harmony search* untuk meyelesaikan permasalahan, yaitu :

1. Inisialisasi masalah dan parameter

a. Fungsi objektif yaitu fungsi untuk meminimalisasi

Pada contoh perhitungan manual ini akan menentukan rute terbaik dari *Router1* ke *Router20* dengan meminimasi fungsi *cost*. Pencarian *cost* menggunakan persamaan (2.3)

b. HMS (*Harmony Memory Size*)

Pada tahap ini akan ditentukan kapasitas HMS yang nantinya berfungsi sebagai tempat dimana solusi sementara akan dibangkitkan secara *random* sebanyak HMS. Pada contoh ini akan menggunakan HMS=6

c. HMCR (*Harmony Memory Considering Rate*)

Parameter ini digunakan untuk mengambil sembarang titik sebagai titik setelah node awal pada saat improvisasi rute memori. Maka akan digunakan HMCR=0.9

d. PAR (*Pitch Adjusting Rate*)

Parameter ini digunakan untuk menentukan node tetangga berikutnya dari node yang sudah dipilih dengan menggunakan parameter HMCR. Maka akan digunakan PAR=0.3

e. NI (*Nilai Iterasi*)

Parameter ini menunjukkan nilai iterasi atau jumlah iterasi yang dilakukan. pada contoh ini akan menggunakan 1 iterasi.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Maka berdasarkan penjabaran diatas kita dapatkan parameter untuk contoh perhitungan manual sebagai berikut :

$$\begin{aligned} \text{HMS} &= 6 \\ \text{HMCR} &= 0.9 \\ \text{PAR} &= 0.3 \\ \text{NI (Nilai Iterasi)} &= 1 \end{aligned}$$

2. Inisialisasi memori rute

Dalam tahap ini penentuan memori rute pada HMS yang pada tahap sebelumnya telah kita tentukan kapasitasnya, yang dilakukan adalah menghasilkan rute yang diacak dari rute yang diberikan. Memori rute merupakan suatu matrik dengan baris menunjukkan himpunan solusi yang disebut vektor memori rute dan kolom menunjukkan variabel keputusan untuk tiap solusi. Berikut contoh *Harmony Memory* (HM) dengan kapasitas HMS=6.

$$\text{HM} = \begin{bmatrix} 1 & 2 & 12 & 13 & 14 & 18 & 20 & 0 \\ 1 & 2 & 5 & 9 & 13 & 14 & 18 & 20 \\ 1 & 3 & 5 & 9 & 10 & 15 & 20 & 0 \\ 1 & 3 & 6 & 11 & 10 & 15 & 18 & 20 \\ 1 & 4 & 7 & 8 & 11 & 17 & 19 & 20 \\ 1 & 4 & 8 & 11 & 17 & 19 & 20 & 0 \end{bmatrix}$$

Setelah mengisi memori rute sebanyak HMS, selanjutnya adalah menghitung fungsi objektif setiap vektor solusi memori rute untuk mendapatkan nilai *cost*. Sebagai contoh kita akan menghitung fungsi objektif pada vektor solusi pertama pada HM untuk meminimasi dengan menggunakan persamaan (2.1).

Diketahui vektor solusi pada baris pertama dalam HM adalah sebagai berikut [1 2 12 13 14 18 20]. Pada vektor solusi ini merupakan kumpulan node yang saling terhubung dan memiliki

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

bobot berdasarkan pada tabel 4.1, selanjutnya akan dihitung fungsi objektifnya menggunakan persamaan (2.1).

$$Cost = \frac{10^8}{Bandwidth\ 1} + \frac{10^8}{Bandwidth\ 2} + \frac{10^8}{Bandwidth\ 3} \dots + \frac{10^8}{Bandwidth\ n}$$

$$Cost = \frac{10^8}{1544} + \frac{10^8}{1544} + \frac{10^8}{128} + \frac{10^8}{128} + \frac{10^8}{128} + \frac{10^8}{128} = 3252$$

Jadi total *cost* pada vektor solusi baris pertama di HM adalah 3252, untuk vektor solusi lainnya dilakukan dengan menggunakan perhitungan yang sama.

$$f(x) = \begin{bmatrix} 3252 \\ 4899 \\ 2535 \\ 4750 \\ 1882 \\ 2595 \end{bmatrix}$$

Setelah proses pada HM telah selesai, langkah selanjutnya adalah improvisasi rute baru, langkah ini digunakan untuk mencari rute baru x_{new} yang nantinya akan dibandingkan dengan rute-rute yang ada pada HM.

3. Improvisasi rute baru

Dalam kasus menentukan jalur terpendek terdapat beberapa langkah modifikasi terhadap algoritme HS dibagian improvisasi rute pada HM. Pada tahap ini yang dilakukan adalah mengimprovisasi rute yang diberikan dalam HM, yakni dengan jalan membentuk vektor memori rute baru dengan menggunakan parameter HMCR dan PAR.

Dari persamaan (2.3), nilai x'_k akan dipilih secara random dari titik yang tersimpan pada HM ($\{x_k^1, x_k^2, \dots, x_k^{HMS}\}$) jika bilangan random yang pertama dibangkitkan adalah bernilai kurang dari atau

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

sama dengan HMCR. Sebaliknya jika bilangan random pertama yang dibangkitkan bernilai lebih dari HMCR maka nilai x'_k akan dipilih secara acak dari X_k .

Dari persamaan (2.4), nilai x'_k akan disesuaikan dengan menggunakan parameter PAR. Proses penyesuaian yang dilakukan adalah dengan penggantian nilai x'_k yang diperoleh pada proses dengan menggunakan parameter HMCR. Nilai x'_k akan digantikan dengan nilai dari tetangga terdekat pada X_k jika nilai bilangan random kedua yang dibangkitkan adalah bernilai kurang dari atau sama dengan $\text{HMCR} \times \text{PAR}$. Sebaliknya jika bilangan random kedua yang dibangkitkan bernilai lebih dari $\text{HMCR} \times \text{PAR}$ maka nilai x'_k tidak berubah. Contoh untuk mencari vektor solusi baru :

- a. Pilih bilangan *random* misal $r_1=0.85$, karena nilai *random* pertama yang dibangkitkan lebih kecil dari HMCR. $r_1 < \text{HMCR}$ berdasarkan aturan pada persamaan (2.3) maka x'_1 di ambil secara acak dari vektor keputusan $\in \{x_1^1, x_1^2, \dots, x_1^{\text{HMS}}\}$,

$$x'_1 = 1$$

Pada bilangan random misal $r_2 = 0.7$, karena nilai random kedua yang dibangkitkan lebih kecil dari $\text{HMCR} \times \text{PAR}$. $r_2 > \text{HMCR} \times \text{PAR}$, maka berdasarkan aturan (2.4) nilai x'_1 tetap sama dengan 1.

$$x'_1 = 1$$

Untuk perhitungan x'_2 sampai x'_n dilakukan perhitungan yang sama dengan menggunakan persamaan (2.3) dan persamaan (2.4) hingga Nilai Itersai terpenuhi. Seperti contoh lanjutan dibawah ini :

- b. Pilih bilangan *random* misal $r_1=0.99$, karena $r_1 > \text{HMCR}$ berdasarkan aturan pada persamaan (2.3) maka x'_2 dipilih secara *random* dari HM dengan syarat titik yang dipilih tidak boleh sama dengan x'_1 .

$$x'_2 = 2$$

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Pada bilangan random misal $r_2= 0.7$, karena $r_2>HMCR \times PAR$, maka berdasarkan aturan (2.4) nilai x'_2 tetap sama dengan 2.

$$x'_2= 2$$

- c. Pilih bilangan *random* misal $r_1=0.75$, karena $r_1<HMCR$ berdasarkan aturan pada persamaan (2.3) maka $x'_3 \in \{x_3^1, x_3^2, \dots, x_3^{HMS}\}$.

$$x'_3= 12$$

Pada bilangan random misal $r_2= 0.25$, karena $r_2<HMCR \times PAR$, maka berdasarkan aturan (2.4) nilai x'_3 digantikan dengan nilai tetangga terdekat pada X_i .

$$x'_3= 5$$

- d. Pilih bilangan *random* misal $r_1=0.91$, karena $r_1>HMCR$ berdasarkan aturan pada persamaan (2.3) maka x'_4 dipilih secara *random* dari HM dengan syarat titik yang dipilih tidak boleh sama dengan $x'_1 x'_2. x'_3$.

$$x'_4= 9$$

Pada bilangan *random* misal $r_2= 0.35$, karena $r_2>HMCR \times PAR$, maka berdasarkan aturan (2.4) nilai x'_4 tetap sama dengan 9.

$$x'_4= 9$$

- e. Pilih bilangan *random* misal $r_1=0.82$, karena $r_1<HMCR$ berdasarkan aturan pada persamaan (2.3) maka $x'_5 \in \{x_5^1, x_5^2, \dots, x_5^{HMS}\}$.

$$x'_5= 10$$

Pada bilangan *random* misal $r_2= 0.43$, karena $r_2>HMCR \times PAR$, maka berdasarkan aturan (2.4) nilai x'_5 tetap sama dengan 10.

$$x'_5= 10$$

- f. Pilih bilangan *random* misal $r_1=0.93$, karena $r_1>HMCR$ berdasarkan aturan pada persamaan (2.3) maka x'_6 dipilih secara *random* dari HM

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dengan syarat titik yang dipilih tidak boleh sama dengan x'_1 x'_2 . x'_3 x'_4 x'_5 .

$$x'_6 = 15$$

Pada bilangan *random* misal $r_2 = 0.52$, karena $r_2 > \text{HMCR} \times \text{PAR}$, maka berdasarkan aturan (2.4) nilai x'_6 tetap sama dengan 15.

$$x'_6 = 15$$

- g. Pilih bilangan *random* misal $r_1 = 0.77$, karena $r_1 < \text{HMCR}$ berdasarkan aturan pada persamaan (2.3) maka $x'_7 \in \{x_7^1, x_7^2, \dots, x_7^{\text{HMS}}\}$.

$$x'_7 = 20$$

Pada bilangan *random* misal $r_2 = 0.49$, karena $r_2 > \text{HMCR} \times \text{PAR}$, maka berdasarkan aturan (2.4) nilai x'_7 tetap sama dengan 20.

$$x'_7 = 20$$

Setelah proses iterasi selesai dilakukan maka didapat suatu vektor solusi baru $x_{\text{new}} = [1 \ 2 \ 5 \ 9 \ 10 \ 15 \ 20]$ vektor harmony rute baru selanjutnya dihitung fungsi objektif nilai *cost* menggunakan persamaan (2.3).

$$\text{Cost} = \frac{10^8}{1544} + \frac{10^8}{1544} + \frac{10^8}{1544} + \frac{10^8}{1544} + \frac{10^8}{1544} + \frac{10^8}{128} = 1101$$

Sehingga didapat nilai fungsi objektif pada vektor solusi yang baru x_{new} adalah $f(x) = [1101]$.

4. Update memori rute

Tahap ini dilakukan perhitungan nilai fungsi objektif pada vektor memori rute baru. Apabila vektor rute baru menghasilkan nilai objektif yang lebih baik dari vektor memori yang terburuk, maka vektor rute baru akan menggantikan di HM dan vektor terburuk akan dikeluarkan.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

$$HM = \begin{bmatrix} 1 & 2 & 12 & 13 & 14 & 18 & 20 & 0 \\ \mathbf{1} & \mathbf{2} & \mathbf{5} & \mathbf{9} & \mathbf{13} & \mathbf{14} & \mathbf{18} & \mathbf{20} \\ 1 & 3 & 5 & 9 & 10 & 15 & 20 & 0 \\ 1 & 3 & 6 & 11 & 10 & 15 & 18 & 20 \\ 1 & 4 & 7 & 8 & 11 & 17 & 19 & 20 \\ 1 & 4 & 8 & 11 & 17 & 19 & 20 & 0 \end{bmatrix} = \begin{bmatrix} 3252 \\ \mathbf{4899} \\ 2535 \\ 4750 \\ 1882 \\ 2595 \end{bmatrix} \rightarrow \text{Terburuk}$$

$$x_{new} = [1 \ 2 \ 5 \ 9 \ 10 \ 15 \ 20] = [1101].$$

Karena fungsi objektif pada kasus ini adalah meminimasi fungsi objektif maka dipilih hasil yang paling terkecil, sehingga nilai terbesar pada HM akan digantikan dengan nilai rute baru yang lebih kecil.

$$HM = \begin{bmatrix} 1 & 2 & 12 & 13 & 14 & 18 & 20 & 0 \\ \mathbf{1} & \mathbf{2} & \mathbf{5} & \mathbf{9} & \mathbf{10} & \mathbf{15} & \mathbf{20} & \mathbf{0} \\ 1 & 3 & 5 & 9 & 10 & 15 & 20 & 0 \\ 1 & 3 & 6 & 11 & 10 & 15 & 18 & 20 \\ 1 & 4 & 7 & 8 & 11 & 17 & 19 & 20 \\ 1 & 4 & 8 & 11 & 17 & 19 & 20 & 0 \end{bmatrix} = \begin{bmatrix} 3252 \\ \mathbf{1101} \\ 2535 \\ 4750 \\ 1882 \\ 2595 \end{bmatrix} \rightarrow x_{new}$$

5. Pemberhentian

Pemberhentian merupakan langkah apabila jumlah iterasi (NI) telah terpenuhi, namun apabila belum terpenuhi maka tetap dilanjutkan dengan mengulang langkah 3 hingga 4. Karena jumlah iterasi pada contoh perhitungan ini adalah 1 iterasi, maka jumlah iterasi maksimum telah terpenuhi dan iterasi berhenti.

Proses di atas menghasilkan urutan rute tempuh dari titik awal *router*1 hingga titik akhir *router* 20 dengan meminimasi fungsi nilai *Cost*, sehingga didapatkan rute terbaik dengan nilai *cost* yakni 1101 dan menghasilkan urutan rute [1 2 5 9 10 15 20] dengan total *bandwidth* sepanjang rute adalah 7848 kbps. Untuk perhitungan dengan nilai *metric* ialah dengan cara yang sama dengan pencarian menggunakan nilai *cost* namun perbedaannya terdapat pada perhitungan fungsi objektif yakni menggunakan persamaan (2.2).