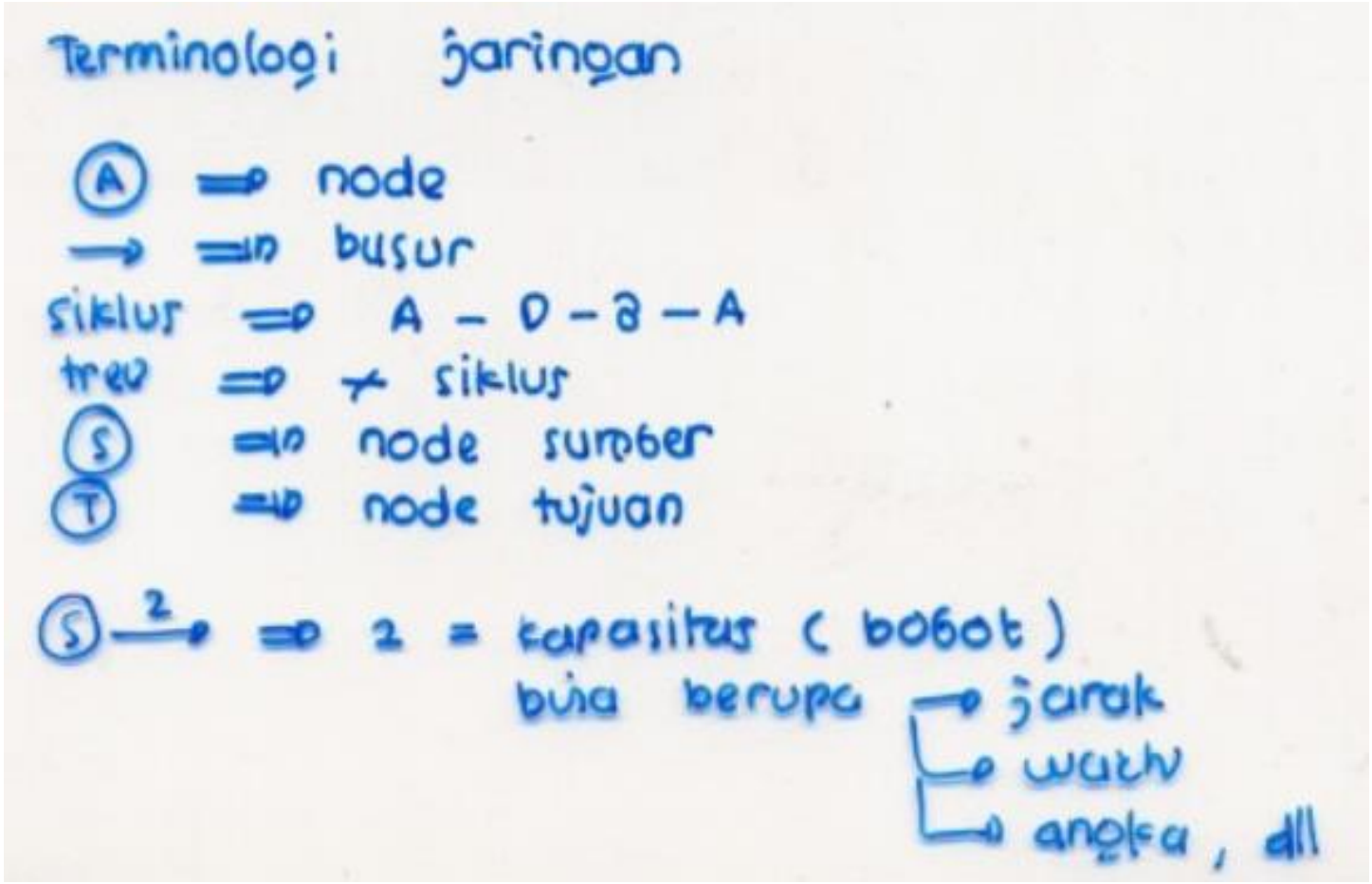
Pertemuan 10

Analisis Jaringan



permasalahan

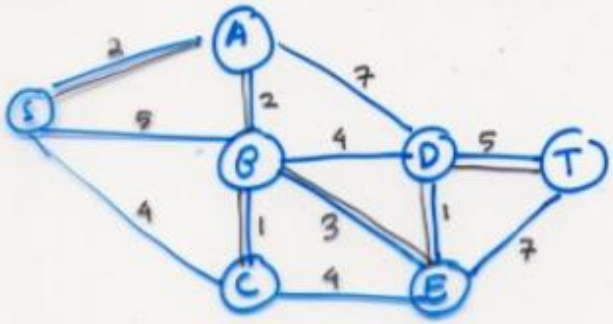
akan diperang kelpon pd tiap kempat,
jalur manakah yg hrs dipakai agar
panjang total kabel min ?
→ Persoalan minimasi jaringan yg
rentang pohon minimal yg
minimal spanning tree

Minimal Spanning Tree

Cara :

1. Dari node (S) hubungkan dgn node lain yg terdekat (ex : A)
2. Pada node yg sudah terhubung
   - pd gambar → garis tebal
   - pd tabel → lingkaran
3. dari beberapa node yang telah terhubung, tent. node berikutnya :
   - cari jarak terpendek antar node
   - ≠ boleh membentuk siklis
   - ≠ kembali ke node sebelumnya
4. apakah semua node sudah terhubung ?
   if yes then stop
   else back to no : (2)

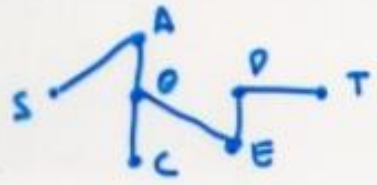
contoh :



tabel :

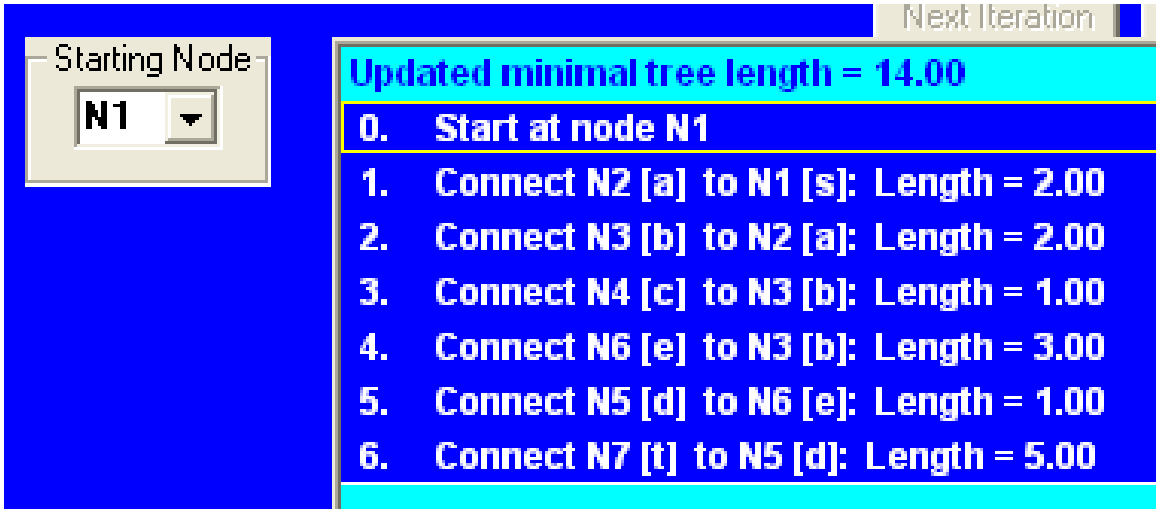
iterasi	node	A	B	C	D	E	T
1	S	(2)	5	4	-	-	-
2	S	-	5	4	-	-	-
	A	-	(2)	-	7	-	-
3	S	-	-	4	-	-	-
	A	-	-	-	7	-	-
	B	-	-	(1)	4	3	-
4	A	-	-	-	7	-	-
	B	-	-	-	4	(3)	-
	C	-	-	-	-	4	-
5	A	-	-	-	7	-	-
	B	-	-	-	4	-	-
	E	-	-	-	(1)	-	7
6	D	-	-	-	-	-	(5)
	E	-	-	-	-	-	7

∴ spanning tree :



∴ total jarak = 2 + 2 + 1 + 3 + 1 + 5 = 14 //

		N1	N2	N3	N4	N5	N6	N7
	Node Name	s	a	b	c	d	e	t
N1	s		2.00	5.00	4.00	infinity	infinity	infinity
N2	a	2.00		2.00	infinity	7.00	infinity	infinity
N3	b	5.00	2.00		1.00	4.00	3.00	infinity
N4	c	4.00	infinity	1.00		infinity	4.00	infinity
N5	d	infinity	7.00	4.00	infinity		1.00	5.00
N6	e	infinity	infinity	3.00	4.00	1.00		7.00
N7	t	infinity	infinity	infinity	infinity	5.00	7.00	



Algoritma rute terpendek

- Ada dua algoritma yang dapat digunakan untuk menyelesaikan persoalan mencari rute terpendek
  - Dijkstra's Algorithm
    - Digunakan untuk mencari rute terpendek dari suatu node dengan semua node lain dalam suatu network
  - Floyd's Algorithm
    - Digunakan untuk mencari rute terpendek antara 2 node dalam suatu network

Dijkstra's Algorithm

- Misalkan u_i adalah rute terpendek dari node 1 ke node i, dan d_{ij} adalah panjang dari arcs(i,j) maka
  - $[u_j,i] = [u_i + d_{ij},i]$, $d_{ij} >= 0$
- Label untuk node awal adalah [0,-] menandakan bahwa node tersebut tidak mempunyai predecessor
- Label suatu node dalam algoritma dijkstra dibedakan menjadi 2
  - Temporary
    - Diubah nilainya jika rute yang lebih pendek bisa ditemukan
  - Permanent
    - Ditentukan jika tidak ada rute lain yang lebih pendek yang dapat ditemukan

Langkah2 Algoritma Dijkstra

- Tandai label awal dengan label permanent [0,-], set i=1
- Hitung label temporary [ui+dij,i] untuk tiap node j yang dapat dicapai dari node i, beri tanda temporari
  - Jika node j sudah punya label [uj,k] melalui node lain k dan jika ui+dij<uj, ganti [uj,k] dengan [uj+dij,i]
- Jika semua node telah mempunyai label permanen, stop. Jika tidak, pilih label [ur,s] yang mempunyai jarak terpendek(ur) dari semua label temporary. Set i=r dan ulaingi step 1

Algoritma Dijkstra

1. Tandai/label node awal [0,-] dan status permanent
2. Beri label untuk node yang dapat berhubungan dengan node permanen dengan [a,b] dan status temporary

   dimana :

   a = jarak terpendek ke node awal

   b = node sebelumnya / yang mendahului
3. Cari a terkecil dan status temporary berubah menjadi permanen
4. Apakah status sudah permanen semua ?

   if yes then stop

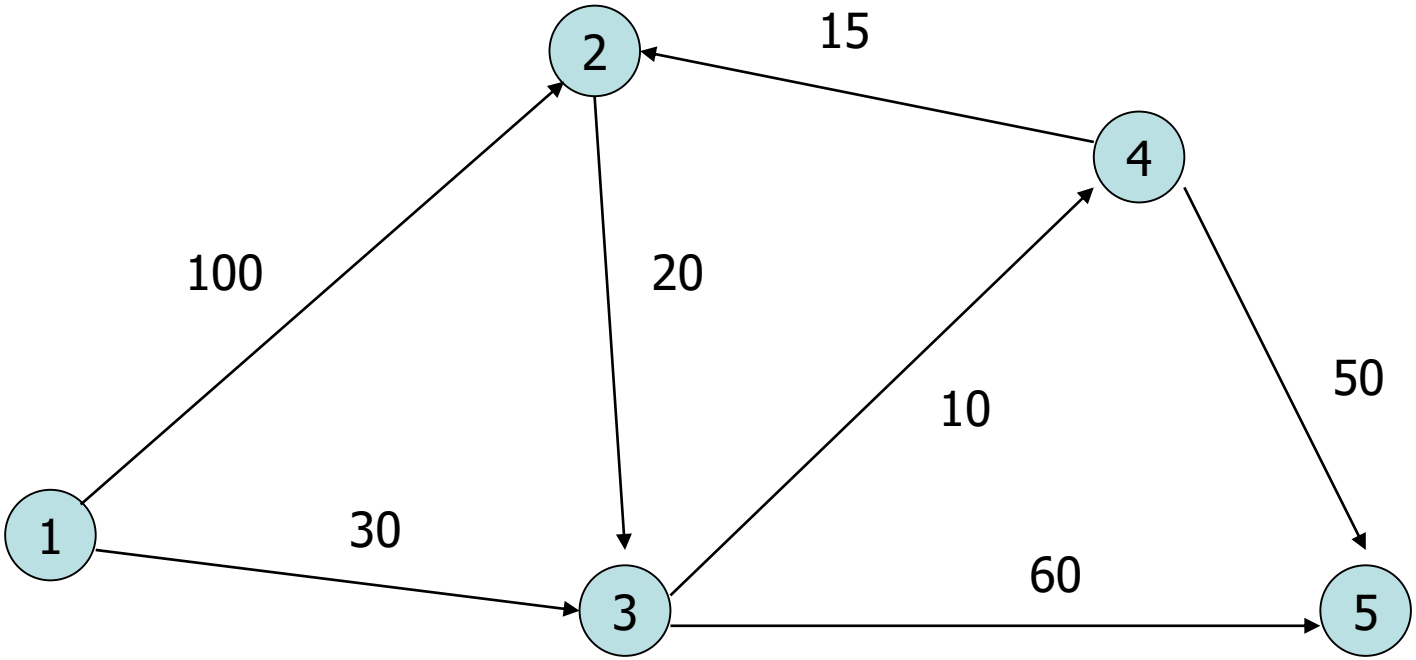
   else back to 2

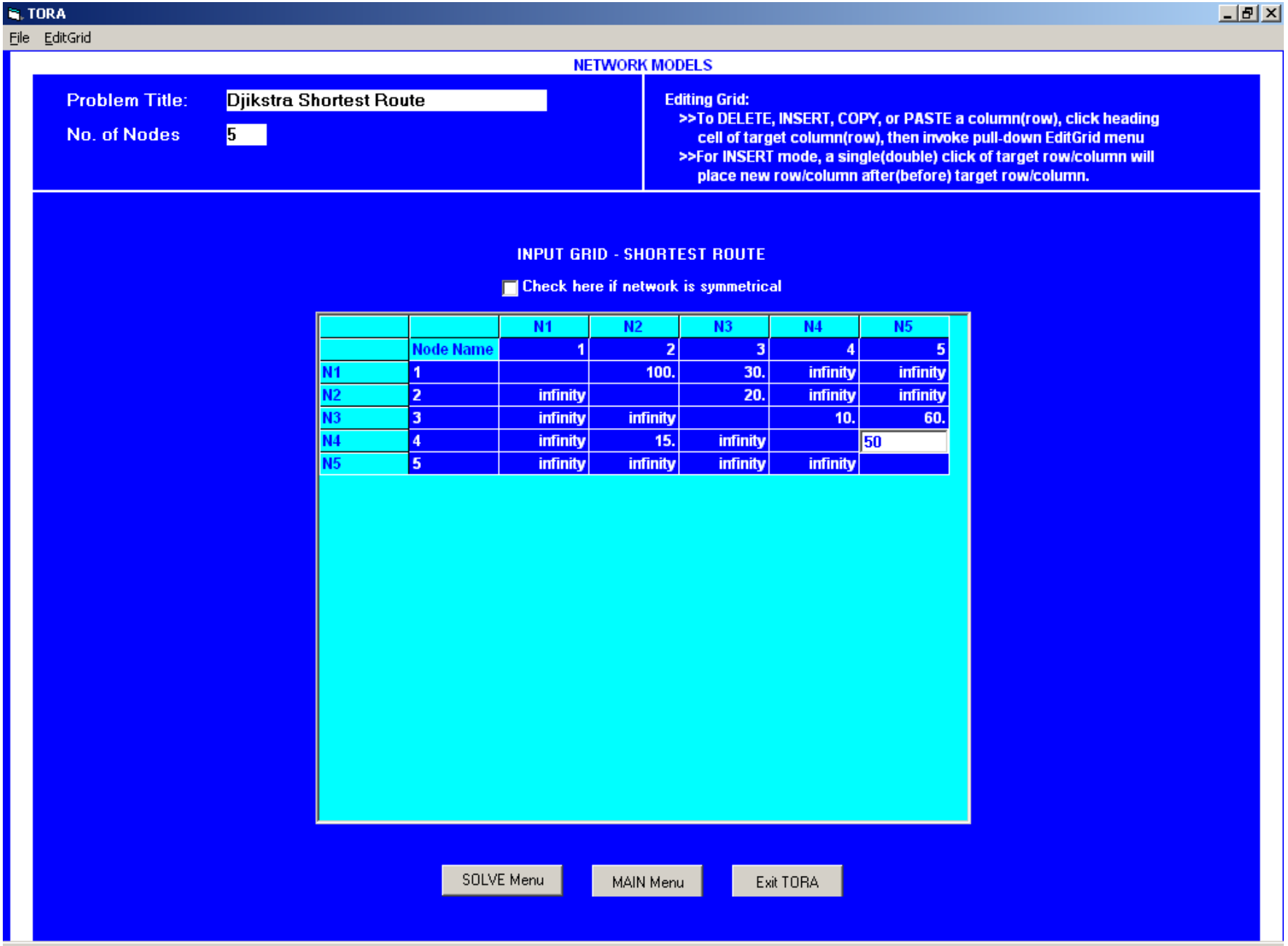
Contoh di gambar 6.3-4

- Iterasi 0
  - Tandai node 1 dengan label permanen [0,-]
- Iterasi 1
  - Node 2 dan 3 dapat dicapai dari node 1 dan dari 2 temporary label itu node 3 mempunyai jarak yg lebih pendek, maka node 3 menjadi label permanen
- Iterasi 2
  - Node 4 dan 5 dapat dicapai dari node 3 dan karena node 4 mempunyai jarak yg lebih pendek, node 4 menjadi node permanen
- Iterasi 3
  - Node 2 dan 5 dapat dicapai dari node 4. temporary label pada node 2 diganti karena rute yg lebih pendek didapat dari node 4. Node 5 mempunyai 2 label yg sama jaraknya maka nilainya tidak diganti
- Iterasi 4
  - Node 2 hanya dapat menuju ke node 3 yang sudah mempunyai label permanen, maka node 2 diberi label permanen
  - Node 5 tidak dapat menuju ke node lain sehingga node 5 juga diberi label permanen

Fig 6.3-4

Contoh jaringan:





		N1	N2	N3	N4	N5
	Node Name	1	2	3	4	5
N1	1		100.	30.	infinity	infinity
N2	2	infinity		20.	infinity	infinity
N3	3	infinity	infinity		10.	60.
N4	4	infinity	15.	infinity		50
N5	5	infinity	infinity	infinity	infinity	

TORA

NETWORK MODELS

DIJKSTRA'S SHORTEST ROUTE ALGORITHM

Select Output Option: Iterations

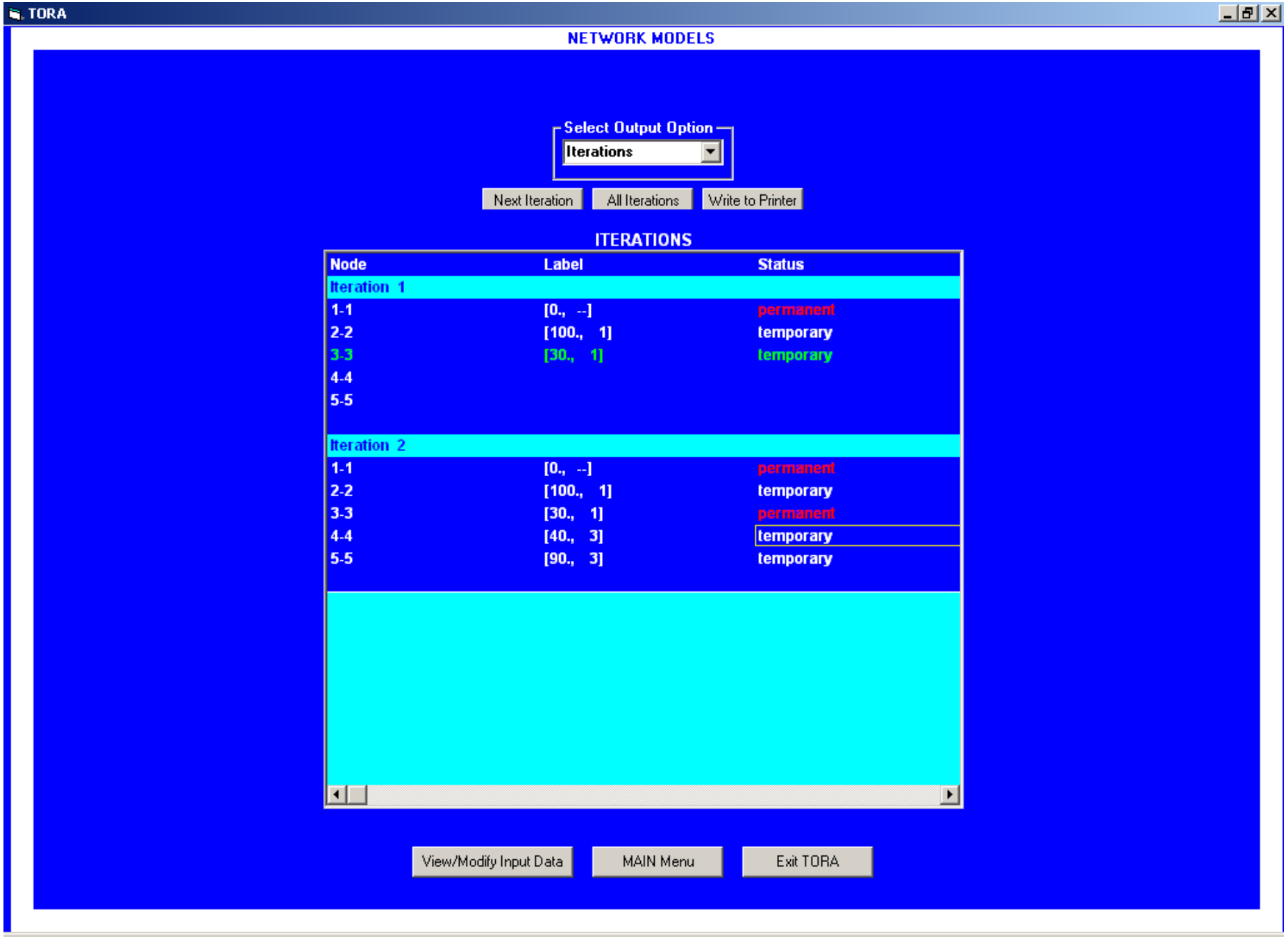
Next Iteration | All Iterations | Write to Printer

Title: Djikstra Shortest Route

ITERATIONS

Node	Label	Status
Iteration 1		
1-1	[0., --]	permanent
2-2	[100., 1]	temporary
3-3	[30., 1]	temporary
4-4		
5-5		

View/Modify Input Data | MAIN Menu | Exit TORA


TORA
NETWORK MODELS
Select Output Option
Iterations
Next Iteration
All Iterations
Write to Printer
ITERATIONS
Node
Label
Status
Iteration 1
1-1
[0., --]
permanent
2-2
[100., 1]
temporary
3-3
[30., 1]
temporary
4-4
5-5
Iteration 2
1-1
[0., --]
permanent
2-2
[100., 1]
temporary
3-3
[30., 1]
permanent
4-4
[40., 3]
temporary
5-5
[90., 3]
temporary
View/Modify Input Data
MAIN Menu
Exit TORA

TORA

NETWORK MODELS

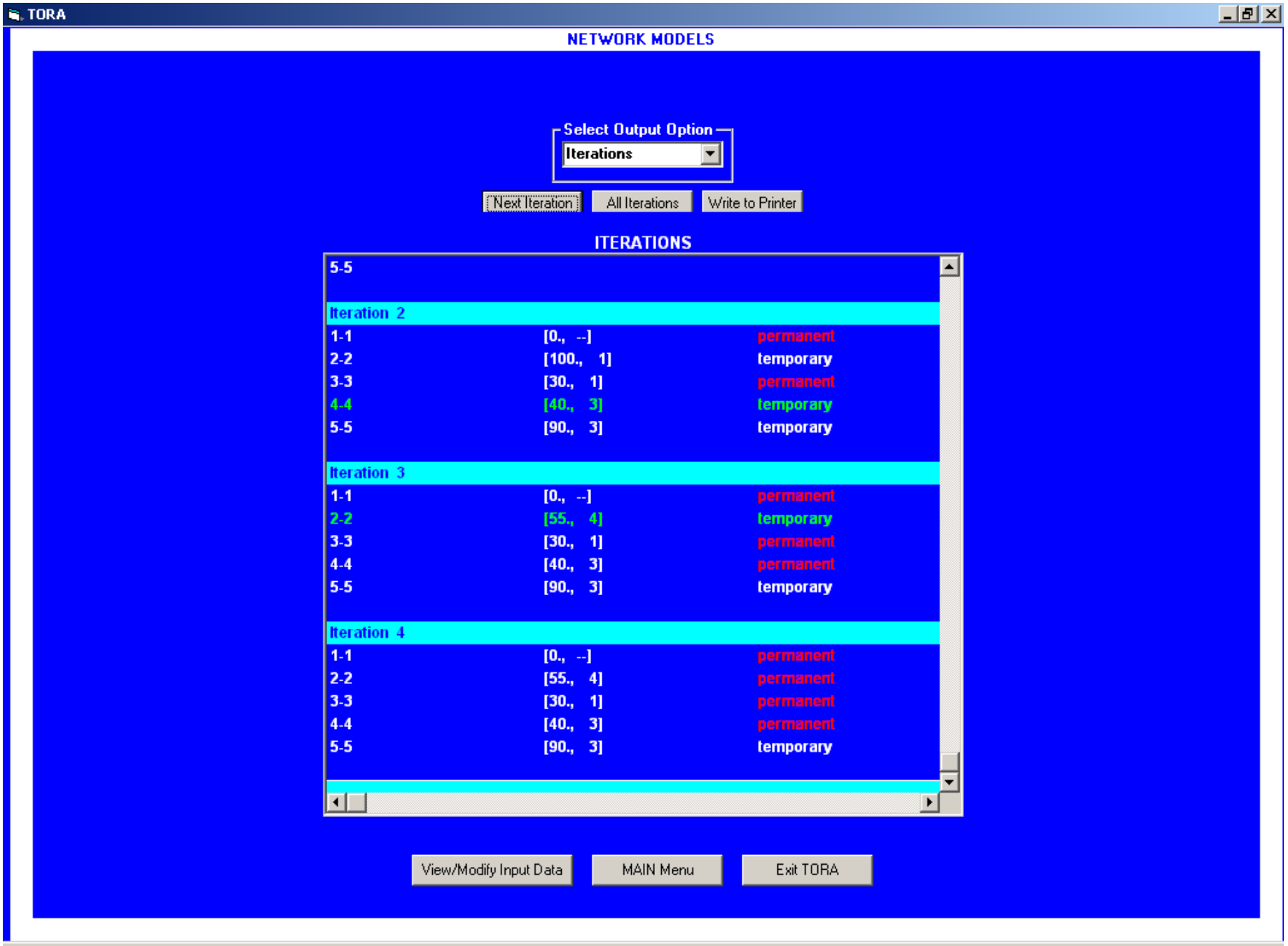
Select Output Option: Iterations

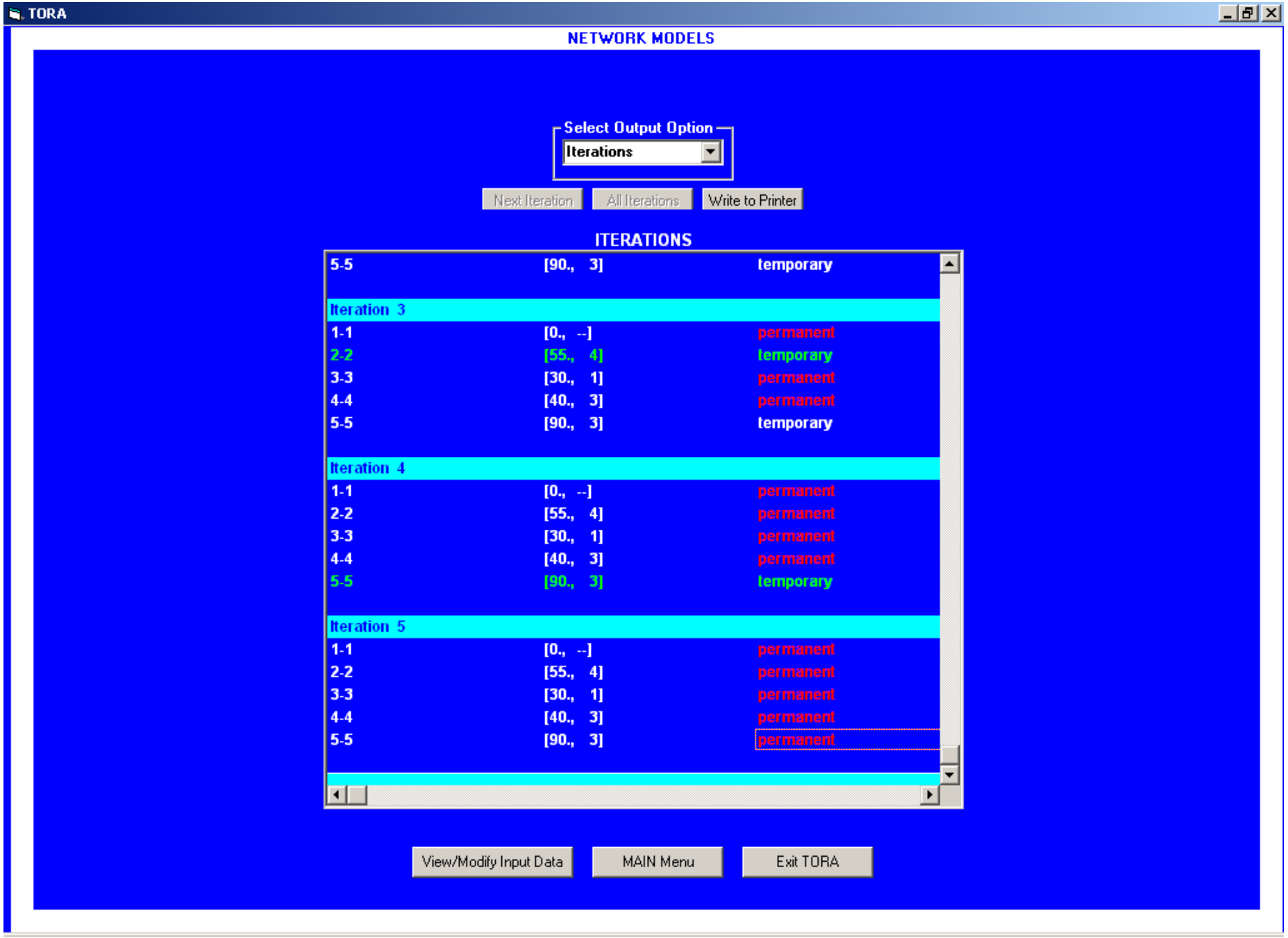
Next Iteration | All Iterations | Write to Printer

ITERATIONS

Node	Label	Status
Iteration 1		
1-1	[0., --]	permanent
2-2	[100., 1]	temporary
3-3	[30., 1]	temporary
4-4		
5-5		
Iteration 2		
1-1	[0., --]	permanent
2-2	[100., 1]	temporary
3-3	[30., 1]	permanent
4-4	[40., 3]	temporary
5-5	[90., 3]	temporary
Iteration 3		
1-1	[0., --]	permanent
2-2	[55., 4]	temporary
3-3	[30., 1]	permanent
4-4	[40., 3]	permanent
5-5	[90., 3]	temporary

View/Modify Input Data | MAIN Menu | Exit TORA


TORA
NETWORK MODELS
Select Output Option
Iterations
Next Iteration
All Iterations
Write to Printer
ITERATIONS
5-5
Iteration 2
1-1 [0., --] permanent
2-2 [100., 1] temporary
3-3 [30., 1] permanent
4-4 [40., 3] temporary
5-5 [90., 3] temporary
Iteration 3
1-1 [0., --] permanent
2-2 [55., 4] temporary
3-3 [30., 1] permanent
4-4 [40., 3] permanent
5-5 [90., 3] temporary
Iteration 4
1-1 [0., --] permanent
2-2 [55., 4] permanent
3-3 [30., 1] permanent
4-4 [40., 3] permanent
5-5 [90., 3] temporary
View/Modify Input Data
MAIN Menu
Exit TORA


TORA
NETWORK MODELS
Select Output Option
Iterations
Next Iteration
All Iterations
Write to Printer
ITERATIONS
5-5 [90., 3] temporary
Iteration 3
1-1 [0., --] permanent
2-2 [55., 4] temporary
3-3 [30., 1] permanent
4-4 [40., 3] permanent
5-5 [90., 3] temporary
Iteration 4
1-1 [0., --] permanent
2-2 [55., 4] permanent
3-3 [30., 1] permanent
4-4 [40., 3] permanent
5-5 [90., 3] temporary
Iteration 5
1-1 [0., --] permanent
2-2 [55., 4] permanent
3-3 [30., 1] permanent
4-4 [40., 3] permanent
5-5 [90., 3] permanent
View/Modify Input Data
MAIN Menu
Exit TORA

TORA

NETWORK MODELS

Select Output Option

Shortest routes

Next Iteration | All Iterations | Write to Printer

Title: Djikstra Shortest Route

SHORTEST ROUTES

Node	Distance	Route
1	0.	--
2	55.	1 - 3 - 4 - 2
3	30.	1 - 3
4	40.	1 - 3 - 4
5	90.	1 - 3 - 5

View/Modify Input Data | MAIN Menu | Exit TORA

Persoalan Rute Terpendek

- Rute terpendek dari node 1 ke node2 yang lain telah dapat ditentukan
- Misal untuk mengetahui rute terpendek dari node 1 ke node 2

(2)->[55,4]->(4)->[40,3]->(3)->[30,1]->(1)

Rute terpendeknya 1->3->4->2

Total jaraknya 55 mil

PR
Minimal Spanning Tree dan rute terpendek