# Pertemuan 11

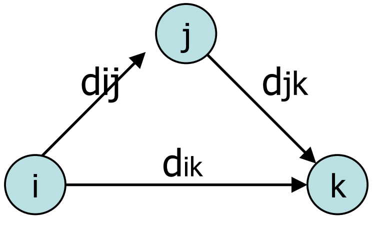
## Algoritma rute terpendek

# 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

# Floyd's Algorithm

- Lebih general di banding dijkstra, karena menghitung rute terpendek diantara semua node yang ada
- Dasar pemikiran:
  – Semua node direpresentasikan dalam matrik D ➔ n x n (dengan n adalah jumlah node yang ada)
  – Entry (i,j) untuk matrik adalah jarak antara node i dan node j.
  – Triple Operation
    - Misal ada 3 node i,j,k dengan jarak dij,djk,dik seperti pada gambar:



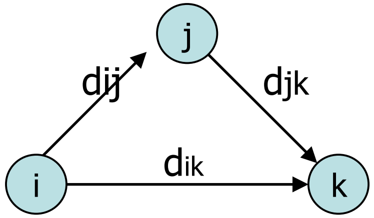

    - Jika dij+djk < dik, maka bisa dikatakan jarak jarak tempuh (i→j→k) < (i→k)

# Floyd's Algorithm

- Dengan demikian, path i→k bisa ditempuh melalui i→j→k untuk mendapatkan shortest pathnya.
- Perubahan triple operation ini bisa dilakukan secara sistematis pada semua jaringan dengan melalui step2 berikut:
  - Step 0
    - Buat matrik node distance D0 dan matrik sequence node S0. Tandai element diagonal dengan (-) yang menandakan element itu di blok. Kita sebut sebagai set k = 1.
  - Step k
    - Tentukan baris dan kolom k sebagai pivot row dan pivot column. Jalankan triple operation untuk tiap elemen dij pada Dk-1 untuk semua i dan j.
    - Jika dik+dkj < dij (i≠k,j≠k dan i≠j) maka
      - » Buat matrik Dk dengan mengganti dij di Dk-1 dengan dik+dkj
      - » Buat matrik Sk dengan mengganti Sij di Sk-1 dengan k.
      - » Set k = k+1
      - » Ulangi step k.

# Floyd’s Algorithm

1. Buat matrix distance D0
2. Buat matrix sequence S0
3. K = 1
4. K sebagai pusat
5. Tripel operation
   - Misal ada 3 node i,j,k dengan jarak dij,djk,dik seperti pada gambar:




   - Jika dij+djk < dik, maka bisa dikatakan jarak jarak tempuh (i→j→k) < (i→k)
   - Dengan demikian, path i→k bisa ditempuh melalui i→j→k untuk mendapatkan shortest pathnya.

6. Buat matrix distance Dk (ganti jarak dengan tripel operation)
7. Buat matrix sequence Sk (ganti dengan k)
8. K = k +1
9. Kembali ke langkah 4

# Contoh6.3-5

- For the network in figure 6.21 find the shortest route between every two nodes. The distances (in miles) are given on the arcs. Arc (3,5) is directional so that no trafic is alowed from node 5 to node 3. All the other arcs allow trafic in both directions.

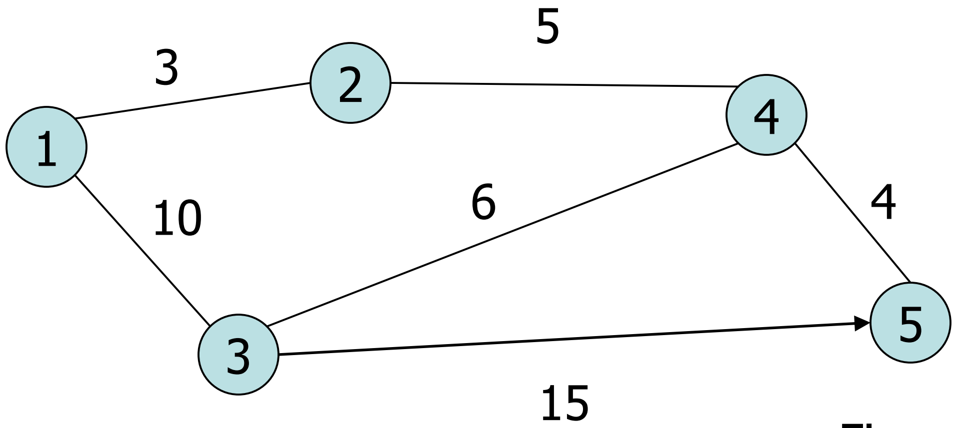



Figure 6.21

# Contoh6.3-5

☑ Check here if network is symmetrical

| | | N1 | N2 | N3 | N4 | N5 |
|---|---|---|---|---|---|---|
| | Node Name | | | | | |
| N1 | | | 3,00 | 10,00 | infinity | infinity |
| N2 | | 3,00 | | infinity | 5,00 | infinity |
| N3 | | 10,00 | infinity | | 6,00 | 15,00 |
| N4 | | infinity | 5,00 | 6,00 | | 5,00 |
| N5 | | infinity | infinity | infinity | 4,00 | |

# Contoh 6.3-5

- Iteration 0

Membentuk matrik $D_0$ dan $S_0$ sebagai kondisi awal iterasi.

| Iter 0 | D 0 | | | | | | S 0 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | infinity | infinity | N1 | | 2 | 3 | 4 | 5 |
| N2 | 3.00 | | infinity | 5.00 | infinity | N2 | 1 | | 3 | 4 | 5 |
| N3 | 10.00 | infinity | | 6.00 | 15.00 | N3 | 1 | 2 | | 4 | 5 |
| N4 | infinity | 5.00 | 6.00 | | 4.00 | N4 | 1 | 2 | 3 | | 5 |
| N5 | infinity | infinity | infinity | 4.00 | | N5 | 1 | 2 | 3 | 4 | |

| Iter 0 | D 0 | | | | | | S 0 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | infinity | infinity | N1 | | 2 | 3 | 4 | 5 |
| N2 | 3.00 | | infinity | 5.00 | infinity | N2 | 1 | | 3 | 4 | 5 |
| N3 | 10.00 | infinity | | 6.00 | 15.00 | N3 | 1 | 2 | | 4 | 5 |
| N4 | infinity | 5.00 | 6.00 | | 4.00 | N4 | 1 | 2 | 3 | | 5 |
| N5 | infinity | infinity | infinity | 4.00 | | N5 | 1 | 2 | 3 | 4 | |

- Iteration 1
  - k=1; maka pivot row dan column yang dipake adalah row dan kolom 1 (1 sbg pusat)
  - element yang mungkin untuk dirubah dengan triple operation dalam D0 adalah

    d23 dan d32 (1 sebagai pusat)
    - Replace d23 dengan d21 + d13 = 3 + 10 = 13 dan Set s23 = 1
    - Replace d23 dengan d32 + d12 = 10 + 3 = 13 dan Set s32 = 1
  - Bearti sudah ada garis dari 3 ke 2 dan ini bisa kita manfaatkan

| Iter 1 | D 1 | | | | | | S 1 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | infinity | infinity | N1 | | 2 | 3 | 4 | 5 |
| N2 | 3.00 | | 13.00 | 5.00 | infinity | N2 | 1 | | 1 | 4 | 5 |
| N3 | 10.00 | 13.00 | | 6.00 | 15.00 | N3 | 1 | 1 | | 4 | 5 |
| N4 | infinity | 5.00 | 6.00 | | 4.00 | N4 | 1 | 2 | 3 | | 5 |
| N5 | infinity | infinity | infinity | 4.00 | | N5 | 1 | 2 | 3 | 4 | |

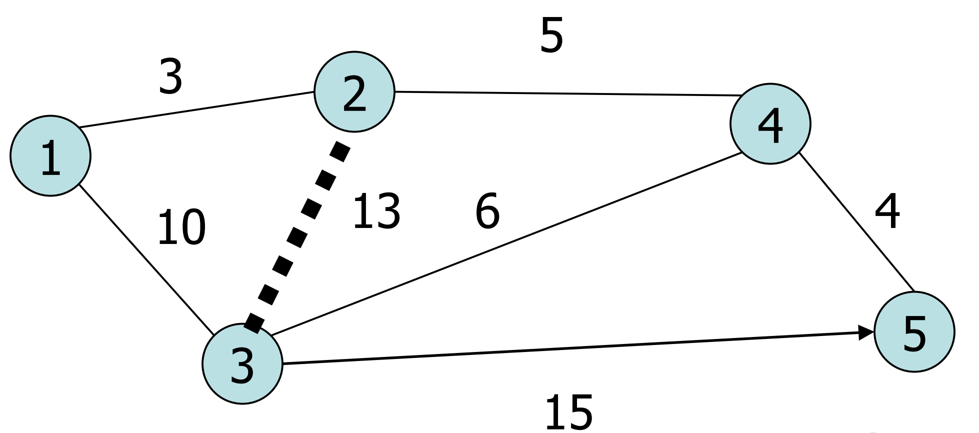


Figure 6.21

| Iter 1 | D 1 | | | | | | S 1 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | infinity | infinity | N1 | | 2 | 3 | 4 | 5 |
| N2 | 3.00 | | 13.00 | 5.00 | infinity | N2 | 1 | | 1 | 4 | 5 |
| N3 | 10.00 | 13.00 | | 6.00 | 15.00 | N3 | 1 | 1 | | 4 | 5 |
| N4 | infinity | 5.00 | 6.00 | | 4.00 | N4 | 1 | 2 | 3 | | 5 |
| N5 | infinity | infinity | infinity | 4.00 | | N5 | 1 | 2 | 3 | 4 | |

- Iteration 2
  - K=2; pivot = kolom dan baris 2 (2 sbg pusat)
  - Yang memungkinkan untuk dirubah pada $d_{14}$ dan $d_{41}$
  - Berarti sudah ada garis dari 1 ke 4

| Iter 2 | D 2 | | | | | | S 2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | 8.00 | infinity | N1 | | 2 | 3 | 2 | 5 |
| N2 | 3.00 | | 13.00 | 5.00 | infinity | N2 | 1 | | 1 | 4 | 5 |
| N3 | 10.00 | 13.00 | | 6.00 | 15.00 | N3 | 1 | 1 | | 4 | 5 |
| N4 | 8.00 | 5.00 | 6.00 | | 4.00 | N4 | 2 | 2 | 3 | | 5 |
| N5 | infinity | infinity | infinity | 4.00 | | N5 | 1 | 2 | 3 | 4 | |

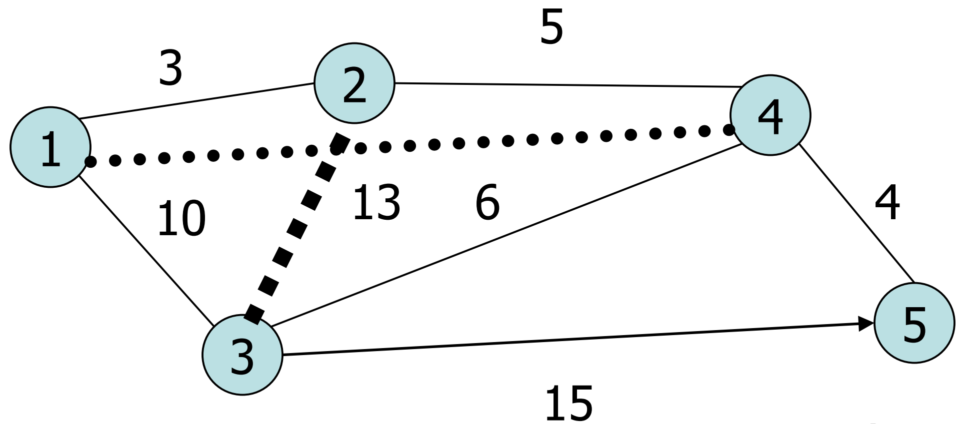



Figure 6.21

| Iter 2 | D 2 | | | | | | S 2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | 8.00 | infinity | N1 | | 2 | 3 | 2 | 5 |
| N2 | 3.00 | | 13.00 | 5.00 | infinity | N2 | 1 | | 1 | 4 | 5 |
| N3 | 10.00 | 13.00 | | 6.00 | 15.00 | N3 | 1 | 1 | | 4 | 5 |
| N4 | 8.00 | 5.00 | 6.00 | | 4.00 | N4 | 2 | 2 | 3 | | 5 |
| N5 | infinity | infinity | infinity | 4.00 | | N5 | 1 | 2 | 3 | 4 | |

- Iteration 3
  - k=3; pivot = kolom dan baris 3 ( 3 sbg pusat)
  - Yang memungkinkan untuk dirubah pada d15 dan d25
  - Berarti sudah ada garis dari 1 ke 5 dan dari 2 ke 5

| Iter 3 | D 3 | | | | | | S 3 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | 8.00 | 25.00 | N1 | | 2 | 3 | 2 | 3 |
| N2 | 3.00 | | 13.00 | 5.00 | 28.00 | N2 | 1 | | 1 | 4 | 3 |
| N3 | 10.00 | 13.00 | | 6.00 | 15.00 | N3 | 1 | 1 | | 4 | 5 |
| N4 | 8.00 | 5.00 | 6.00 | | 4.00 | N4 | 2 | 2 | 3 | | 5 |
| N5 | infinity | infinity | infinity | 4.00 | | N5 | 1 | 2 | 3 | 4 | |

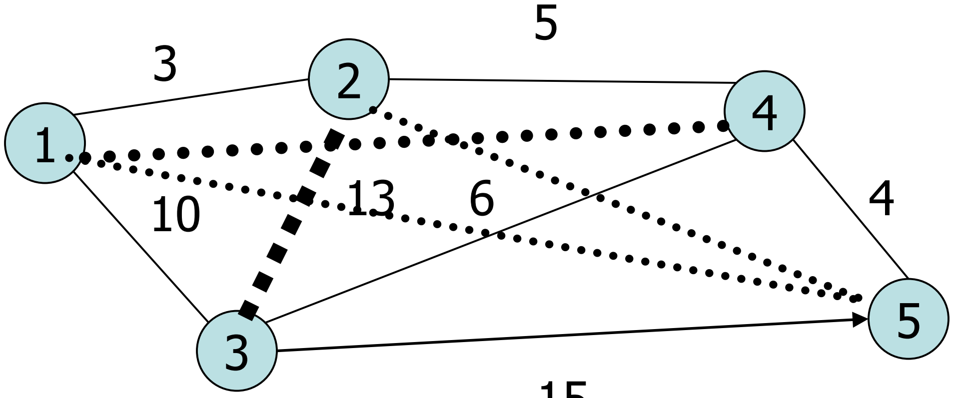



Figure 6.21

| Iter 3 | D 3 | | | | | | S 3 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | 8.00 | 25.00 | N1 | | 2 | 3 | 2 | 3 |
| N2 | 3.00 | | 13.00 | 5.00 | 28.00 | N2 | 1 | | 1 | 4 | 3 |
| N3 | 10.00 | 13.00 | | 6.00 | 15.00 | N3 | 1 | 1 | | 4 | 5 |
| N4 | 8.00 | 5.00 | 6.00 | | 4.00 | N4 | 2 | 2 | 3 | | 5 |
| N5 | infinity | infinity | infinity | 4.00 | | N5 | 1 | 2 | 3 | 4 | |

- Iteration 4
  - k=4; pivot = kolom dan baris 4
  - Yang memungkinkan untuk dirubah pada d23, d32, d51, d15, d52, d25, d35 dan d53

| Iter 4 | D 4 | | | | | | S 4 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | 8.00 | 12.00 | N1 | | 2 | 3 | 2 | 4 |
| N2 | 3.00 | | 11.00 | 5.00 | 9.00 | N2 | 1 | | 4 | 4 | 4 |
| N3 | 10.00 | 11.00 | | 6.00 | 10.00 | N3 | 1 | 4 | | 4 | 4 |
| N4 | 8.00 | 5.00 | 6.00 | | 4.00 | N4 | 2 | 2 | 3 | | 5 |
| N5 | 12.00 | 9.00 | 10.00 | 4.00 | | N5 | 4 | 4 | 4 | 4 | |

| Iter 4 | D 4 | | | | | | S 4 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | 8.00 | 12.00 | N1 | | 2 | 3 | 2 | 4 |
| N2 | 3.00 | | 11.00 | 5.00 | 9.00 | N2 | 1 | | 4 | 4 | 4 |
| N3 | 10.00 | 11.00 | | 6.00 | 10.00 | N3 | 1 | 4 | | 4 | 4 |
| N4 | 8.00 | 5.00 | 6.00 | | 4.00 | N4 | 2 | 2 | 3 | | 5 |
| N5 | 12.00 | 9.00 | 10.00 | 4.00 | | N5 | 4 | 4 | 4 | 4 | |

- Iteration 5
  - k=5, melihat sudah tidak mungkin terjadi perubahan lagi pada tabel D4 maka bisa diasumsikan D5 dan S5 sama dengan D4 dan S4.

| Iter 5 | D 5 | | | | | | S 5 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | N 1 | N 2 | N 3 | N 4 | N 5 | | N 1 | N 2 | N 3 | N 4 | N 5 |
| N1 | | 3.00 | 10.00 | 8.00 | 12.00 | N1 | | 2 | 3 | 2 | 4 |
| N2 | 3.00 | | 11.00 | 5.00 | 9.00 | N2 | 1 | | 4 | 4 | 4 |
| N3 | 10.00 | 11.00 | | 6.00 | 10.00 | N3 | 1 | 4 | | 4 | 4 |
| N4 | 8.00 | 5.00 | 6.00 | | 4.00 | N4 | 2 | 2 | 3 | | 5 |
| N5 | 12.00 | 9.00 | 10.00 | 4.00 | | N5 | 4 | 4 | 4 | 4 | |

# Contoh 6.3-5

- Bukti..
  - Matrik D5 dan S5 sudah memuat semua informasi yang kita butuhkan untuk menentukan shortest path antara dua node. Isi element pada matrik D menunjukkan jarak sedangkan isi pada element matrik S adalah node yang harus dilintasi. Jika Sij=j maka ini menunjukkan direct link. Sebaliknya jika tidak maka Sij adalah node yang harus ditempuh sebelum menuju node j.
  - Misal untuk node 1 dan 5. tertulis jarak antara node 1 dan 5 (d15) adalah 12 mil.
  - Dengan path sebagai berikut:
    - 1→5
    - 1→4→5
    - 1→2→4→5
  - jadi path terpendek dari node 1 ke 5 adalah melalui 1→2→4→5 dengan jarak 12 mile.

# PR ➔ Cari jarak dengan Floyd