

# Studi Kasus: Topological Sort

IF2110/IF2111 – Algoritma dan Struktur Data  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung

# Deskripsi Persoalan

Diberikan urutan partial dari elemen suatu himpunan,

Dikehendaki agar elemen tersebut mempunyai keterurutan linier.

# Contoh Persoalan

- Dalam suatu kurikulum, suatu mata pelajaran mempunyai prerequisite mata pelajaran lain.
  - Urutan linier adalah urutan untuk seluruh mata pelajaran dalam kurikulum
- Dalam suatu proyek, suatu pekerjaan harus dikerjakan lebih dulu dari pekerjaan lain
  - misalnya membuat pondasi harus sebelum dinding, membuat dinding harus sebelum pintu. Namun pintu dapat dikerjakan bersamaan dengan jendela. Dan sebagainya.

## Contoh Persoalan (2)

- Dalam pembuatan tabel pada basis data, tabel yang di-refer oleh tabel lain harus dideklarasikan terlebih dulu. Jika suatu aplikasi terdiri dari banyak tabel, maka urutan pembuatan tabel harus sesuai dengan definisinya.
- Dalam sebuah program Pascal, pemanggilan prosedur harus sedemikian rupa, sehingga peletakan prosedur pada teks program harus sesuai dengan urutan (partial) pemanggilan.

# Definisi Permasalahan

Jika  $X < Y$  adalah simbol untuk  $X$  “sebelum”  $Y$ , dan merupakan keterurutan *partial*,

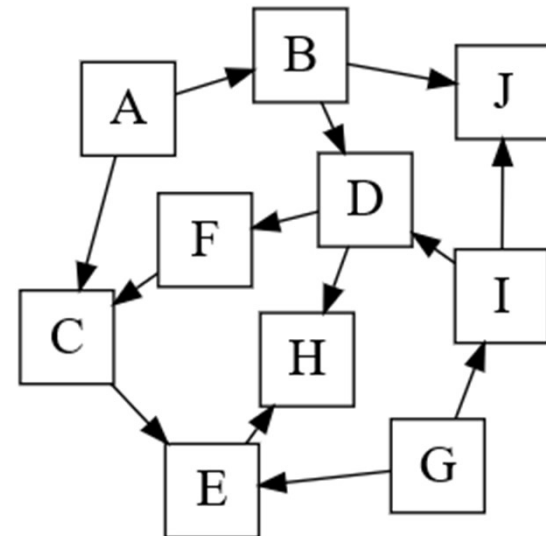
Contoh pada simpul-simpul dalam sebuah graf

$A < B$	$B < D$	$D < F$	$B < J$	$D < H$
$F < C$	$A < C$	$C < E$	$E < H$	$G < E$
$G < I$	$I < D$	$I < J$		

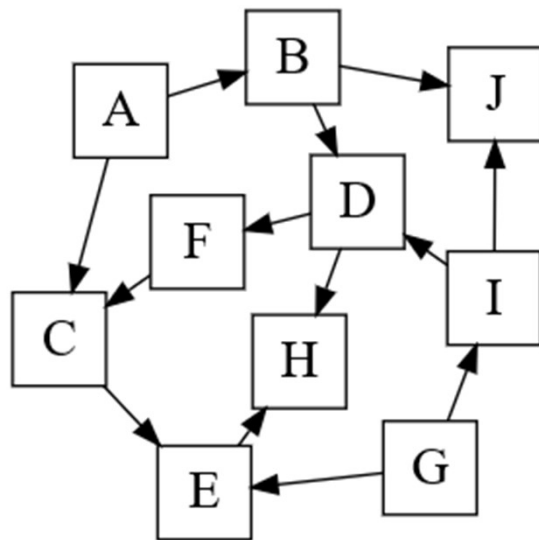
# Defini Persoalan

Representasi dalam Graf  
(DAG-Directed Acyclic Graph)

$A < B$	$B < D$	$D < F$	$B < J$	$D < H$
$F < C$	$A < C$	$C < E$	$E < H$	$G < E$
$G < I$	$I < D$	$I < J$		

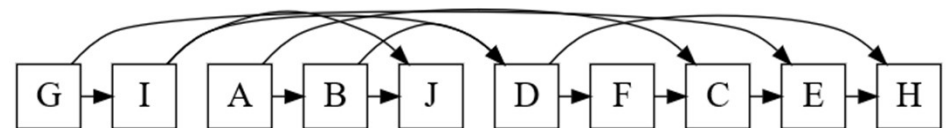


# Definisi Persoalan

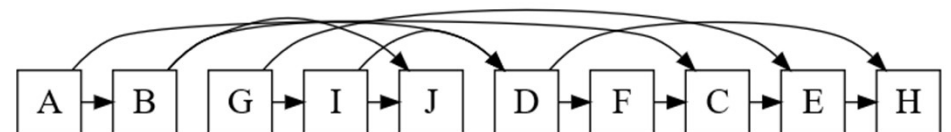


Bagaimana total/linear order? [beberapa solusi]

1.  $G \rightarrow I \rightarrow A \rightarrow B \rightarrow J \rightarrow D \rightarrow F \rightarrow C \rightarrow E \rightarrow H$



2.  $A \rightarrow B \rightarrow G \rightarrow I \rightarrow J \rightarrow D \rightarrow F \rightarrow C \rightarrow E \rightarrow H$



# Ide dari Topological Sort

*Andaikata item yang mempunyai keterurutan partial adalah anggota himpunan  $S$ .*

1. **Pilih salah satu item yang tidak mempunyai predesesor**, misalnya  $X$ . Minimal ada satu elemen semacam ini. Jika tidak, maka akan *looping*.
2. **Hapus  $X$  dari himpunan  $S$**  (dan keterhubungan dari  $X$  ke item lainnya), dan insert ke dalam list hasil sorting.
3. Sisa himpunan  $S$  masih merupakan himpunan terurut partial, maka **proses 1 dan 2** dapat dilakukan lagi terhadap sisa dari  $S$  **hingga kosong**.



# Solusi 1: MultiList

Struktur Data:

Jumlah elemen pada DAG *unbounded* → list berkait dengan representasi dinamis (pointer)

Field pada setiap elemen

- Identitas (id)

- Jumlah Predecessor (predCount)

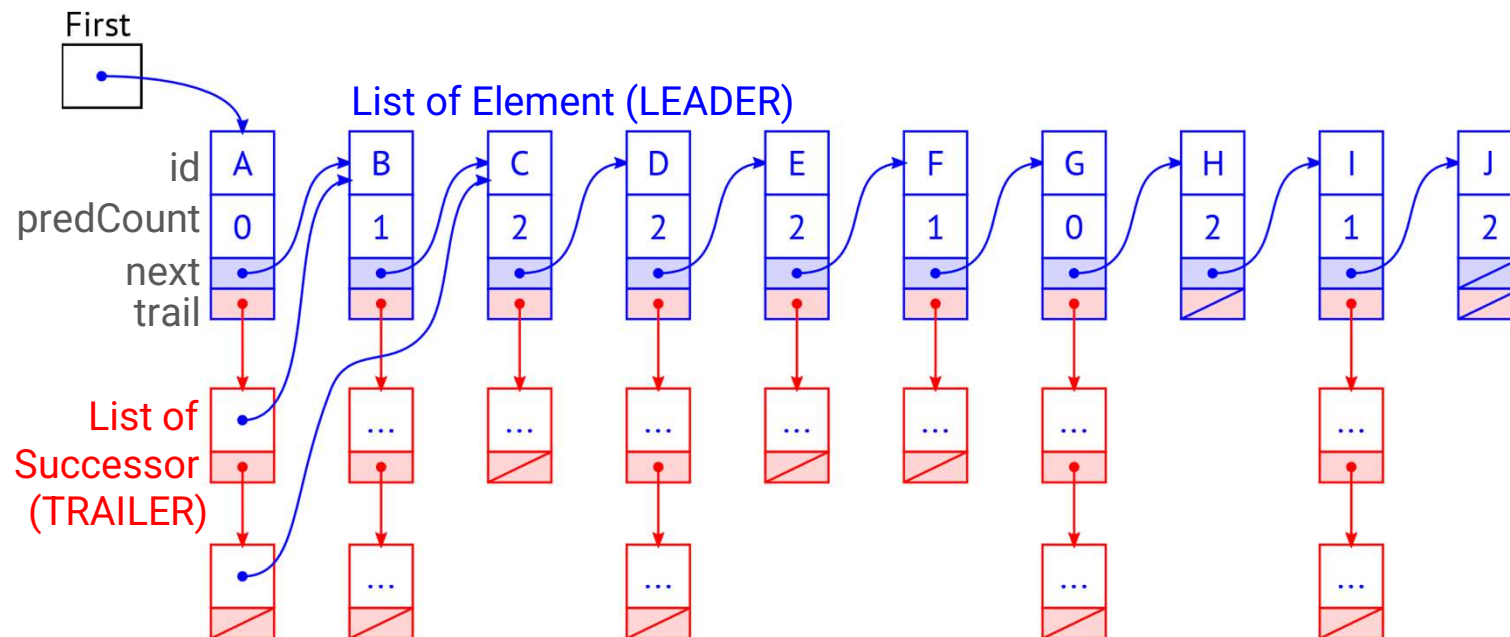
- List to Successor (trail)

MultiList

- List of Elemen (horizontal)

- List of Successor dari tiap Elemen (vertikal)

# Struktur Data



# Algoritma Pembentukan MultiList

Bentuk list Leader dan Trailer

baca pasangan nilai ( $X < Y$ ).

temukan alamat  $X$  dan  $Y$  pada Leader

jika belum ada sisipkan sebagai elemen baru

*InsertFirst* alamat  $Y$  sebagai Trailer  $X$ .

Increment nilai Count (jumlah predesessor) dari  $Y$

# Algoritma Topological Sort

*Search* elemen dari list Leader dengan jumlah predesesor = 0

Delete elemen tersebut dan decrement *Count* dari successor dari elemen tersebut

*InsertLast* sebagai elemen list linier hasil pengurutan

[Lakukan hingga List Leader kosong]

## Solusi 2: Pendekatan Fungsional dengan List Linear Biasa

### Struktur Data:

Graf partial dinyatakan sebagai list linier dengan representasi fisik First-Last dengan *dummy*.

Elemen list terdiri dari  $\langle \text{prec}, \text{succ} \rangle$ . Contoh: sebuah elemen bernilai  $\langle 1, 2 \rangle$  artinya 1 adalah predesesor dari 2.

```
type Node: < prec: char,  
              succ: char,  
              next: address >  
type ListTopo: < first: Address,  
                  last: Address >
```

# Algoritma Topological Sort

1. P adalah elemen pertama (First(L)), misalkan elemennya adalah  $\langle X, Y \rangle$
  2. *Search* pada sisa list (list tanpa elemen pertama), **apakah X mempunyai predesesor**.  
**Jika ya**, maka elemen ini harus dipertahankan sampai saatnya dapat dihapus dari list untuk dioutputkan:
    1. *Delete* P, tapi jangan didealokasi.
    2. *Insert* P sebagai Last(L) yang baru.**Jika tidak** mempunyai predesesor, maka X siap untuk dioutputkan, tetapi Y masih harus dipertanyakan. Maka langkah yang harus dilakukan:
    1. *InsertLast* X pada list hasil sorting, jika belum ada.
    2. *Search* **apakah Y masih ada pada sisa list**, baik sebagai Prec maupun sebagai Succ.  
**Jika ya**, maka Y akan dioutputkan nanti. Hapus elemen pertama yang sedang diproses dari list.  
**Jika tidak** muncul sama sekali, berarti Y tidak mempunyai predesesor, maka *InsertLast* Y pada list hasil sorting, baru hapus elemen pertama dari list.
- [Lakukan hingga list kosong]**

# Contoh proses topological sort (1)

List input: [ $\langle A, B \rangle$ ,  $\langle B, D \rangle$ ,  $\langle D, F \rangle$ ,  $\langle B, J \rangle$ ,  $\langle D, H \rangle$ ,  $\langle F, C \rangle$ ,  $\langle A, C \rangle$ ,  $\langle C, E \rangle$ ,  $\langle E, H \rangle$ ,  $\langle G, E \rangle$ ,  $\langle G, I \rangle$ ,  $\langle I, D \rangle$ ,  $\langle I, J \rangle$ ]

List output: [ ]

$P = \langle A, B \rangle$ ,  $X = A$  tidak memiliki predesesor,  $Y = B$  masih ada di sisa list

List output: [ $A$ ]

List input: [ $\langle B, D \rangle$ ,  $\langle D, F \rangle$ ,  $\langle B, J \rangle$ ,  $\langle D, H \rangle$ ,  $\langle F, C \rangle$ ,  $\langle A, C \rangle$ ,  $\langle C, E \rangle$ ,  $\langle E, H \rangle$ ,  $\langle G, E \rangle$ ,  $\langle G, I \rangle$ ,  $\langle I, D \rangle$ ,  $\langle I, J \rangle$ ]

$P = \langle B, D \rangle$ ,  $X = B$  tidak memiliki predesesor,  $Y = D$  masih ada di sisa list

List output: [ $A, B$ ]

List input: [ $\langle D, F \rangle$ ,  $\langle B, J \rangle$ ,  $\langle D, H \rangle$ ,  $\langle F, C \rangle$ ,  $\langle A, C \rangle$ ,  $\langle C, E \rangle$ ,  $\langle E, H \rangle$ ,  $\langle G, E \rangle$ ,  $\langle G, I \rangle$ ,  $\langle I, D \rangle$ ,  $\langle I, J \rangle$ ]

## Contoh proses topological sort (2)

$P=\langle D,F \rangle$ ,  $X=D$  memiliki predesesor

List output:  $[A,B]$

List input:  $[\langle B,J \rangle, \langle D,H \rangle, \langle F,C \rangle, \langle A,C \rangle, \langle C,E \rangle, \langle E,H \rangle, \langle G,E \rangle, \langle G,I \rangle, \langle I,D \rangle, \langle I,J \rangle, \langle D,F \rangle]$

$P=\langle B,J \rangle$ ,  $X=B$  tidak memiliki predesesor,  $Y=J$  masih ada di sisa list

List output:  $[A,B] \leftarrow$  tetap, karena  $B$  sudah ada

List input:  $[\langle D,H \rangle, \langle F,C \rangle, \langle A,C \rangle, \langle C,E \rangle, \langle E,H \rangle, \langle G,E \rangle, \langle G,I \rangle, \langle I,D \rangle, \langle I,J \rangle, \langle D,F \rangle]$

$P=\langle D,H \rangle$ ,  $X=D$  memiliki predesesor

List output:  $[A,B]$

List input:  $[\langle F,C \rangle, \langle A,C \rangle, \langle C,E \rangle, \langle E,H \rangle, \langle G,E \rangle, \langle G,I \rangle, \langle I,D \rangle, \langle I,J \rangle, \langle D,F \rangle, \langle D,H \rangle]$



## Contoh proses topological sort (3)

$P=\langle F,C\rangle$ ,  $X=F$  memiliki predesesor

List output:  $[A,B]$

List input:  $[\langle A,C\rangle, \langle C,E\rangle, \langle E,H\rangle, \langle G,E\rangle, \langle G,I\rangle, \langle I,D\rangle, \langle I,J\rangle, \langle D,F\rangle, \langle D,H\rangle, \langle F,C\rangle]$

$P=\langle A,C\rangle$ ,  $X=A$  tidak memiliki predesesor,  $Y=C$  masih ada di sisa list

List output:  $[A,B] \leftarrow$  tetap, karena  $A$  sudah ada

List input:  $[\langle C,E\rangle, \langle E,H\rangle, \langle G,E\rangle, \langle G,I\rangle, \langle I,D\rangle, \langle I,J\rangle, \langle D,F\rangle, \langle D,H\rangle, \langle F,C\rangle]$

$P=\langle C,E\rangle$ ,  $X=C$  memiliki predesesor

List output:  $[A,B]$

List input:  $[\langle E,H\rangle, \langle G,E\rangle, \langle G,I\rangle, \langle I,D\rangle, \langle I,J\rangle, \langle D,F\rangle, \langle D,H\rangle, \langle F,C\rangle, \langle C,E\rangle]$

## Contoh proses topological sort (4)

$P=\langle E, H \rangle$ ,  $X=E$  memiliki predesesor

List output:  $[A, B]$

List input:  $[\langle G, E \rangle, \langle G, I \rangle, \langle I, D \rangle, \langle I, J \rangle, \langle D, F \rangle, \langle D, H \rangle, \langle F, C \rangle, \langle C, E \rangle, \langle E, H \rangle]$

$P=\langle G, E \rangle$ ,  $X=G$  tidak memiliki predesesor,  $Y = E$  masih ada di sisa list

List output:  $[A, B, G]$

List input:  $[\langle G, I \rangle, \langle I, D \rangle, \langle I, J \rangle, \langle D, F \rangle, \langle D, H \rangle, \langle F, C \rangle, \langle C, E \rangle, \langle E, H \rangle]$

$P=\langle G, I \rangle$ ,  $X=G$  tidak memiliki predesesor,  $Y = I$  masih ada di sisa list

List output:  $[A, B, G] \leftarrow$  tetap, karena  $G$  sudah ada

List input:  $[\langle I, D \rangle, \langle I, J \rangle, \langle D, F \rangle, \langle D, H \rangle, \langle F, C \rangle, \langle C, E \rangle, \langle E, H \rangle]$

# Contoh proses topological sort (5)

$P=\langle I,D\rangle$ ,  $X=I$  tidak memiliki predesesor,  $Y = D$  masih ada di sisa list

List output:  $[A,B,G,I]$

List input:  $[\langle I,J\rangle, \langle D,F\rangle, \langle D,H\rangle, \langle F,C\rangle, \langle C,E\rangle, \langle E,H\rangle]$

$P=\langle I,J\rangle$ ,  $X=I$  tidak memiliki predesesor,  $Y = J$  tidak ada di sisa list

List output:  $[A,B,G,I,J]$

List input:  $[\langle D,F\rangle, \langle D,H\rangle, \langle F,C\rangle, \langle C,E\rangle, \langle E,H\rangle]$

$P=\langle D,F\rangle$ ,  $X=D$  tidak memiliki predesesor,  $Y = F$  masih ada di sisa list

List output:  $[A,B,G,I,J,D]$

List input:  $[\langle D,H\rangle, \langle F,C\rangle, \langle C,E\rangle, \langle E,H\rangle]$

## Contoh proses topological sort (6)

$P=\langle D,H\rangle$ ,  $X=D$  tidak memiliki predesesor,  $Y = H$  masih ada di sisa list

List output:  $[A,B,G,I,J,D]$  ← tetap, karena  $D$  sudah ada

List input:  $[\langle F,C\rangle, \langle C,E\rangle, \langle E,H\rangle]$

$P=\langle F,C\rangle$ ,  $X=F$  tidak memiliki predesesor,  $Y = C$  masih ada di sisa list

List output:  $[A,B,G,I,J,D,F]$

List input:  $[\langle C,E\rangle, \langle E,H\rangle]$

$P=\langle C,E\rangle$ ,  $X=C$  tidak memiliki predesesor,  $Y = E$  masih ada di sisa list

List output:  $[A,B,G,I,J,D,F,C]$

List input:  $[\langle E,H\rangle]$

# Contoh proses topological sort (7)

$P=\langle E, H \rangle$ ,  $X=E$  tidak memiliki predesesor,  $Y = H$  tidak ada di sisa list

List output: [A,B,G,I,J,D,F,C,E,H]

List input: [ ]

# Fungsi Search

Untuk solusi ini diperlukan primitif *search* apakah sebuah nilai X muncul sebagai Prec(P) atau Succ(P) pada “sisal list” (list tanpa elemen pertama). Untuk efisiensi proses, kedua macam *search* digabungkan dengan spesifikasi:

```
function searchTopo (l: ListTopo, x: char) → <boolean, boolean>
{ boolean pertama akan bernilai true jika x muncul sebagai
  prec pada salah satu elemen sisal list l
  boolean kedua akan bernilai true jika x muncul sebagai
  succ pada salah satu elemen sisal list l }
```