DEADLOCK

Haryono Setiadi, ST, M.Eng

OBJEK PEMBELAJARAN

- Overview Deadlock
- Ilustrasi Deadlock
- Syarat terjadinya deadlock
- Metode mengatasi deadlock
- Penghidaran deadlock

PENDAHULUAN

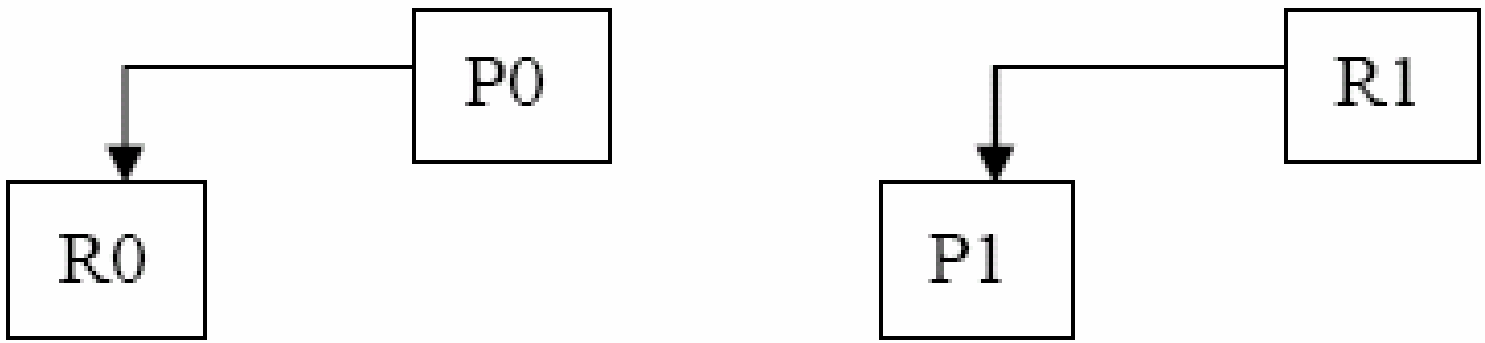
- Proses dikatan deadlock jika proses menunggu suatu kejadian tertentu yang tidak pernah terjadi
- Sekumpulan proses → deadlock jika setiap proses yang berada dikumpulan itu menunggu suatu kejadian yang dapat dilakukan proses lain yang juga berada dikumpulan itu
- Deadlock terjadi ketika proses-proses mengakses sumber daya secara ekslusif
- Semua deadlock yang terjadi melibatkan persaingan untuk memperoleh sumber daya eksklusif oleh 2 proses atau lebih

MODEL DEADLOCK

Model deadlock 2 proses dan 2 sumber daya

Deadlock dapat digambarkan sebagai graph

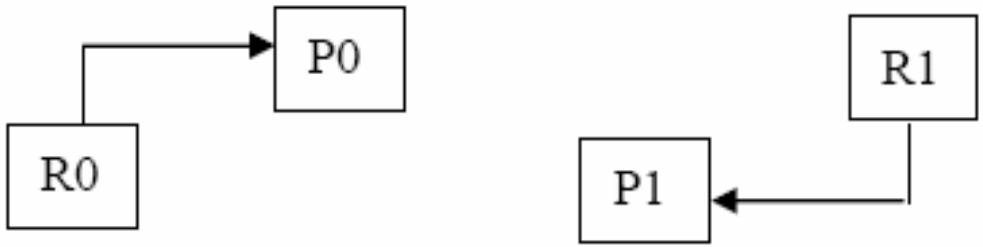
- Misal : 2 proses P0 dan P1
- 2 sumber daya R0 dan R1
- P0 meminta sumberdaya R0.
- Sumber daya R1 dialokasikan ke P1.



Skenario yang menimbulkan deadlock :

P0 dialokasikan R0

P1 dialokasikan R1

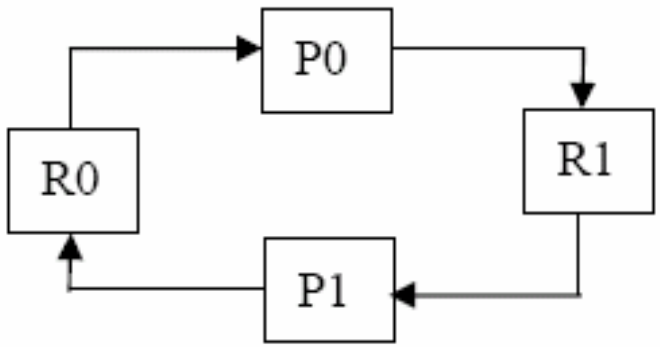


Kemudian

P0 sambil masih menggenggam R0, meminta R1

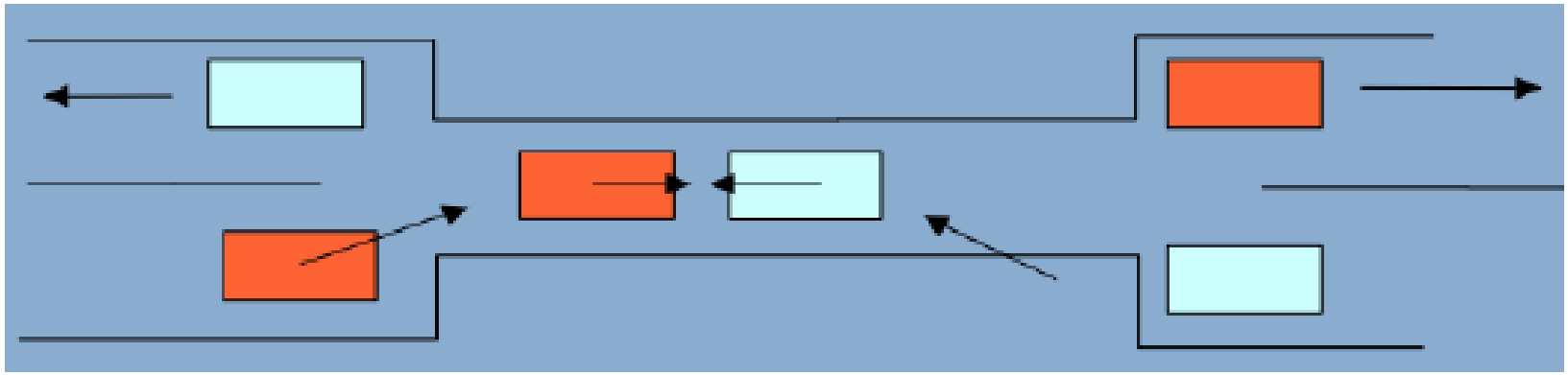
P1 sambil masih menggenggam R1, meminta R0

Terjadi deadlock karena sama-sama akan saling menunggu. Terjadinya deadlock ditandai munculnya graph melingkar.



Deadlock dapat terjadi dengan melibatkan lebih dari 2 proses dan 2 sumber daya.

ILUSTRASI DEADLOCK



- Pada satu jalan yang memungkinkan hanya satu arah yang berjalan
- Setiap jalan bisa dianggap sebagai sumber daya
- Saat deadlock terjadi hanya bisa diatasi jika salah satu mobil mundur, dalam hal ini butuh sumber daya yang direalokasikan
- Bahkan beberapa mobil harus mundur jika deadlock terjadi
- Pada kasus ini juga bisa terjadi “kelaparan”, yaitu ada proses yang tidak terlayani

Syarat terjadinya deadlock (1)

1. mutual exclusion (mutual exclusion conditional)

hanya satu proses pada satu waktu yang dpt menggunakan R

2. kondisi genggam dan tunggu (hold and wait)

proses-proses yang sedang menggenggam sumber daya, menunggu sumber daya-sumber daya baru.

3. kondisi non-preemption (non-preemption condition)

sumber daya-sumber daya yang sebelumnya diberikan tidak dapat diambil paksa dari proses itu.

Sumber daya harus secara explisit dilepaskan dari proses yang menggenggamnya.

4. kondisi menunggu secara sirkuler (circular wait condition)

Terdapat sekumpulan proses (P0, P1 … Pn) yang menunggu R, dimana Po menunggu R yang dibawa P1, P1 menunggu R yang dibawa P2 dst → Pn-1 menunggu R yang dibawa Pn

Syarat terjadinya deadlock (2)

- Ketiga syarat pertama merupakan syarat perlu (necessary condition) terjadinya deadlock, yaitu :
  1. **mutual exclusion (mutual exclusion conditional)**
  2. **kondisi genggam dan tunggu (hold and wait)**
  3. **kondisi non-preemption (non-preemption condition)**
- Terjadi deadlock bila terdapat 3 kondisi itu, tetapi adanya ketiga kondisi itu belum berarti terjadi deadlock.
- Deadlock benar-benar terjadi bila syarat keempat terpenuhi. Kondisi keempat merupakan keharusan bagi terjadinya deadlock. → **kondisi menunggu secara sirkuler**

Metode mengatasi deadlock (1)

1. Metode pencegahan deadlock
2. Penghindaran deadlock

Metode mengatasi deadlock (2)

1. Metode pencegahan deadlock

- Tiap proses harus meminta semua sumber daya yang diperlukan secara sekaligus dan tidak berlanjut sampai semuanya diberikan.
- Jika proses sedang memegang sumberdaya tertentu, untuk permintaan berikutnya proses harus melepas dulu sumberdaya yang dipegangnya.

Metode mengatasi deadlock (3)

2. Penghindaran deadlock

- Menghindari deadlock dengan cara hanya memberi akses ke permintaan sumber daya yang tidak mungkin menimbulkan deadlock.
  - Jika tidak aman (memungkinkan timbulnya deadlock), proses yang meminta di suspend sampai suatu waktu permintaannya aman diberikan.

Metode mengatasi deadlock (4)

2. Penghindaran deadlock

- Agar dapat mengevaluasi safe-nya state sistem, penghindaran deadlock mengharuskan semua proses menyatakan jumlah kebutuhan sumber daya maksimum sebelum eksekusi.
- Begitu eksekusi dimulai, tiap proses meminta sumber daya saat diperlukan sampai batas maksimum yang dinyatakan di awal.
- Proses-proses yang menyatakan kebutuhan sumber daya melebihi kapasitas total sistem tidak dapat dieksekusi.

Metode mengatasi deadlock (5)

2. Penghindaran deadlock

- Untuk penghidaran deadlock diperlukan pengertian mengenai :
  - state selamat (safe state)
  - state tak selamat (unsafe state)

Penghidaran Deadlock (1)

- **State selamat (safe state)**
- State selamat (safe state) → jika tidak deadlock, dan terdapat cara untuk memenuhi semua permintaan yang ditunda tanpa menghasilkan deadlock → dengan menjalankan proses secara hati-hati mengikuti suatu urutan tertentu.

Contoh : state selamat

Contoh : Sistem dengan 10 sumber daya setipe

- Proses A memerlukan sumber daya maksimum sebanyak 10, sedang saat ini menggenggam 2 sumber daya.
- Proses B memerlukan sumber daya maksimum sebanyak 3, sedang saat ini menggenggam 1 sumber daya.
- Proses C memerlukan sumber daya maksimum sebanyak 7, sedang saat ini menggenggam 3 sumber daya.
- Masih tersedia 4 sumber daya.

Proses	Jumlah sumberdaya digenggam	Maksimum sumberdaya dibutuhkan
A	2	10
B	1	3
C	3	7
Tersedia	4	

- State dinyatakan selamat (safe) karena terdapat barisan pengalokasian yang dapat memungkinkan semua proses selesai.
- Dengan penjadwalan secara hati-hati, sistem dapat terhindarkan dari deadlock. Barisan tersebut adalah :

Langkah 1 :

Alokasikan semua sumberdaya ke proses C, tunggu sampai proses C selesai.

Proses	Jumlah sumberdaya digenggam	Maksimum sumberdaya dibutuhkan
A	2	10
B	1	3
C	7	7
Tersedia	0	

Setelah proses C selesai menjadi :

Proses	Jumlah sumberdaya digenggam	Maksimum sumberdaya dibutuhkan
A	2	10
B	1	3
C	0	-
Tersedia	7	

Langkah 2 :

Alokasikan 2 sumberdaya ke proses B, tunggu sampai proses B selesai.

Proses	Jumlah sumberdaya digenggam	Maksimum sumberdaya dibutuhkan
A	2	10
B	3	3
C	0	-
Tersedia	5	

Setelah proses B selesai menjadi :

Proses	Jumlah sumberdaya digenggam	Maksimum sumberdaya dibutuhkan
A	2	10
B	0	-
C	0	-
Tersedia	8	

Langkah 3 :

Alokasikan 8 sumberdaya ke proses A, tunggu sampai proses A selesai.

Proses	Jumlah sumberdaya digenggam	Maksimum sumberdaya dibutuhkan
A	10	10
B	0	-
C	0	-
Tersedia	0	

Setelah proses A selesai menjadi :

Proses	Jumlah sumberdaya digenggam	Maksimum sumberdaya dibutuhkan
A	0	-
B	0	-
C	0	-
Tersedia	10	

Ketiga proses tersebut dengan penjadwalan yang hati-hati dapat menyelesaikan proses mereka dengan sempurna.

- **State tak selamat (unsafe state)**
- State tak selamat (unsafe state) : jika tidak terdapat cara untuk memenuhi semua permintaan yang saat ini ditunda dengan menjalankan proses-proses dengan suatu urutan.

Contoh :

State dibawah ini sama dengan state selamat sebelumnya, tapi dapat berubah menjadi state tak selamat bila alokasi sumber daya tak terkendali.

Proses	Jumlah sumberdaya digenggam	Maksimum sumberdaya dibutuhkan
A	2	10
B	1	3
C	3	7
Tersedia	4	

State berubah menjadi tak selamat bila 2 permintaan sumber daya oleh proses A dilayani kemudian permintaan 1 sumber daya oleh proses B dilayani. Maka state menjadi :

Proses	Jumlah sumberdaya digenggam	Maksimum sumberdaya dibutuhkan
A	4	10
B	2	3
C	3	7
Tersedia	1	

Alokasikan 1 sumberdaya ke proses B, tunggu sampai proses B selesai.

Proses	Jumlah sumberdaya Digenggam	Maksimum sumberdaya dibutuhkan
A	4	10
B	3	3
C	3	7
Tersedia	0	

Setelah proses B selesai menjadi :

Proses	Jumlah sumberdaya digenggam	Maksimum sumberdaya dibutuhkan
A	4	10
B	0	-
C	3	7
Tersedia	3	

Sumber daya yang tersedia tinggal 3 sedangkan 2 proses yang sedang aktif masing-masing membutuhkan 6 dan 4 sumber daya.

State tak selamat bukan berarti deadlock, tetapi state tersebut berkemungkinan menuju deadlock.

END CHAPTER

Chapter Selanjutnya :

Manajemen Memory