



# **MODUL SISTEM OPERASI DEADLOCK**

# Definisi

- *Deadlock* dalam arti sebenarnya adalah kebuntuan.
- Kebuntuan yang dimaksud dalam sistem operasi adalah kebuntuan proses.
- **Jadi *Deadlock* ialah suatu kondisi dimana proses tidak berjalan lagi atau pun tidak ada komunikasi lagi antar proses.**

# Definisi

- **Deadlock** disebabkan karena proses yang satu **menunggu** sumber daya yang sedang dipegang oleh proses lain yang sedang menunggu sumber daya yang dipegang oleh proses tersebut.
- Dengan kata lain, **Deadlock** terjadi ketika proses menunggu sumber daya untuk melakukan suatu kejadian tertentu yang **tidak akan pernah terjadi**.

# Syarat Terjadinya Deadlock

## ***1. Mutual Exclusion Condition***

Sumber daya yang harus diberikan hanya pada satu proses.

## ***2. Hold and Wait Condition***

Proses yang sedang memegang sumberdaya, menunggu sumber daya yang baru.

# Syarat Terjadinya Deadlock

## ***3. Non-Preemption Condition***

Sumber daya yang diberikan sebelumnya tidak dapat diambil paksa dari proses yang sedang menggenggamnya. Harus terlebih dahulu dilepaskan oleh proses yang menggenggamnya.

# Syarat Terjadinya Deadlock

## ***4. Circular Wait Condition***

Harus terdapat rantai sirkuler dari dua proses atau lebih, dan masing-masing proses menunggu sumber daya yang digenggam oleh proses berikutnya.

# Metode Mengatasi Deadlock

- 1. Metode Pencegahan Deadlock  
(Deadlock Prevention)**
- 2. Metode Penghindaran Deadlock  
(Deadlock Avoidance)**
- 3. Metode Deteksi (Deadlock Detection)**

# Metode Mengatasi Deadlock (1)

## 1. Metode Pencegahan Deadlock (Deadlock Prevention)

### a. Meniadakan Mutual Exclusion

Melakukan spooling perangkat-perangkat yang harus didedikasikan ke suatu proses. Dengan spooling, permintaan-permintaan diantrikan di harddisk. Setiap job di antrian spooler akan dilayani satu per satu.



# Metode Mengatasi Deadlock (1)

## 1. Metode Pencegahan Deadlock (Deadlock Prevention)

### b. Meniadakan Hold and Wait

- Mengalokasikan semua sumber daya atau tidak sama sekali
- Hold and release

# Metode Mengatasi Deadlock (1)

**c. Meniadakan Non-preemption**

**d. Meniadakan Menunggu Sirkular**

- Proses hanya diperbolehkan menggenggam satu sumber daya.
- Penomoran global semua sumber daya.

# Metode Mengatasi Deadlock (2)

## 2. Metode Penghindaran Deadlock

### (Deadlock Avoidance)

- Proses harus menyatakan seluruh sumber daya maksimum yang dibutuhkan sebelum eksekusi.
- Ketika eksekusi berlangsung, proses meminta sumber daya yang diperlukan hingga batas maksimum yang dinyatakan di awal.

# Metode Mengatasi Deadlock (2)

- Proses yang menyatakan kebutuhan melewati kapasitas sistem, tidak akan dieksekusi.

## *Safe State*

- State dinyatakan safe state jika tidak deadlock dan terdapat cara untuk memenuhi seluruh permintaan tanpa menghasilkan deadlock.

# Metode Mengatasi Deadlock (2)

## *Unsafe State*

- State dinyatakan unsafe state jika tidak terdapat cara untuk memenuhi semua permintaan yang tertunda dengan menjalankan proses-proses sesuai suatu urutan.

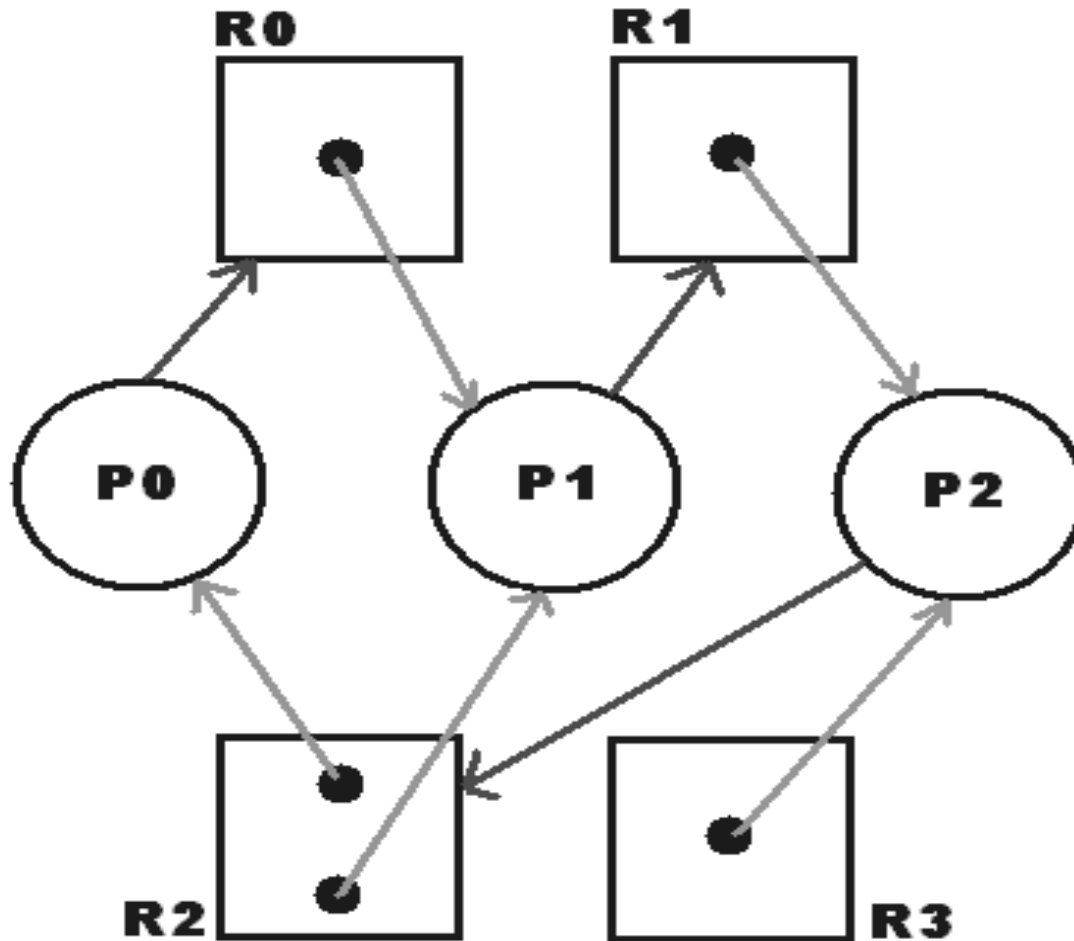
# Metode Mengatasi Deadlock (3)

## 3. Metode Deteksi Deadlock (Deadlock Detection)

Untuk mengetahui ada atau tidaknya deadlock dalam suatu graf dapat dilihat dari **perputaran** dan **resource** yang dimilikinya, yaitu:

1. Jika tidak ada perputaran berarti tidak deadlock.
2. Jika ada perputaran, ada potensi terjadi deadlock.
3. Resource dengan instan tunggal dan perputaran mengakibatkan deadlock.

# Metode Mengatasi Deadlock (3)



# Metode Mengatasi Deadlock (3)

Gambar di atas menunjukkan beberapa hal sebagai berikut:

1. P0 meminta sumber daya R0.
2. R0 mengalokasikan sumber dayanya pada P1.
3. P1 meminta sumber daya R1.
4. R1 mengalokasikan sumber dayanya pada P2.
5. P2 meminta sumber daya R2.
6. R2 mengalokasikan sumber dayanya pada P0 dan P1.
7. R3 mengalokasikan sumber dayanya pada P2.



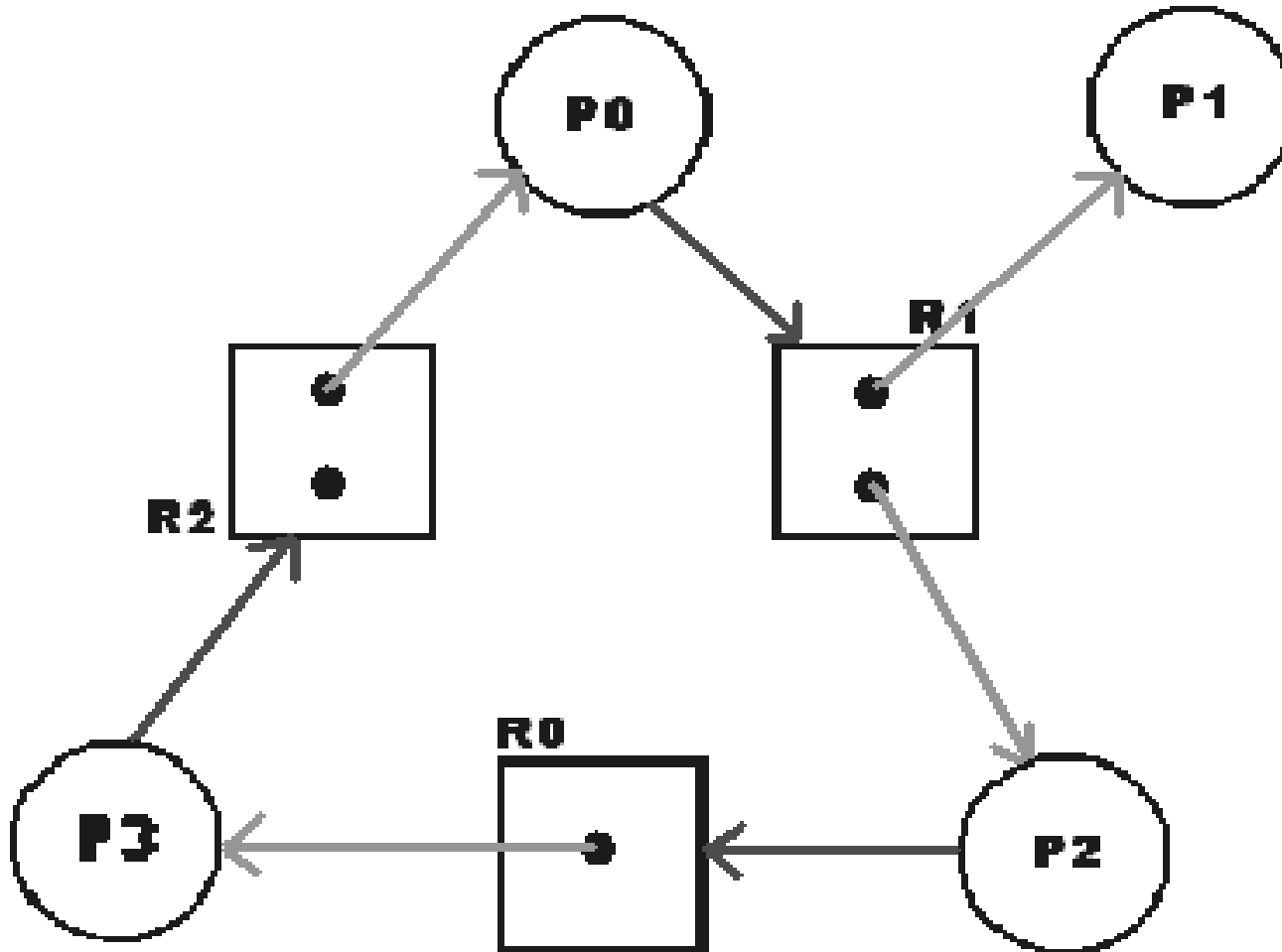
# Metode Mengatasi Deadlock (3)

Hal-hal tersebut dapat mengakibatkan *deadlock* sebab P0 memerlukan sumber daya R0 untuk menyelesaikan prosesnya, sedangkan R0 dialokasikan untuk P1.

Di lain pihak P1 memerlukan sumber daya R1 sedangkan R1 dialokasikan untuk P2.

P2 memerlukan sumber daya R2 akan tetapi R2 mengalokasikan sumber dayanya pada R3.

# Metode Mengatasi Deadlock (3)



# Metode Mengatasi Deadlock (4)

## 4. Metode Pemulihan Deadlock (Deadlock Recovery)

### *Preemption*

Untuk sementara waktu menjauhkan sumber daya dari pemakainya, dan memberikannya pada proses yang lain.

### **Melacak Kembali**

Dibutuhkan langkah untuk kembali pada keadaan aman dimana proses masih berjalan dan memulai proses lagi.

# Metode Mengatasi Deadlock (4)

## **Membunuh proses yang menyebabkan *Deadlock***

1. Yang paling jarang memakai prosesor
2. Yang paling sedikit hasil programnya
3. Yang paling banyak memakai sumber daya sampai saat ini
4. Yang alokasi sumber daya totalnya tersedikit
5. Yang memiliki prioritas terkecil

# KUIS

1. Apa yang dimaksud dengan sumber daya?
2. Sebutkan 4 kondisi yang menyebabkan Deadlock?
3. State akan dinyatakan safe state apabila dapat memenuhi kebutuhan seluruh permintaan tanpa menghasilkan Deadlock, bagaimanakah pengaturan yang tepat dari setiap proses yang masuk unuk mencapai kondisi safe state?