

# KONGRUENSI

## Pengertian

Kongruensi merupakan landasan umum sistem operasi, proses2 disebut kongruen jika proses2 (lebih dari satu proses) berada pada saat yang sama. Proses proses kongruen dapat sepenuhnya tak tergantung dengan proses lainnya tapi juga dapat saling berinteraksi.

## Masalah yang harus diselesaikan dalam kongruensi

- Mutual ekclusion
- Deadlock
- Starvation
- Sinkronisasi

## Mutual Exclusion

Deskripsi : Jaminan hanya satu proses yang mengakses sumber daya pada suatu interval waktu tertentu.

## Mutual Exclusion

Terdapat sumber daya yang tidak dapat dipakai bersama pada saat yang bersamaan, seperti printer. Sumber daya macam ini disebut sumber daya kritis. Hanya satu program pada satu saat yang diijinkan masuk critical region/section (bagian program yang yang menggunakan sumber daya kritis)

## Metode Penjaminan ME

- Metode variabel lock sederhana
- Metode bergantian secara ketat
- Metode penyelesaian Dekker
- Metode penyelesaian Peterson
- Metode hardware intruksi pematian interupsi
- Metode Hardware intruksi khusus

## Deadlock

Deskripsi: Proses disebut deadlock jika proses menunggu suatu kejadian tertentu yang tak pernah terjadi. Sekumpulan proses berkondisi deadlock bila setiap proses lain yang juga berada di kumpulan itu.

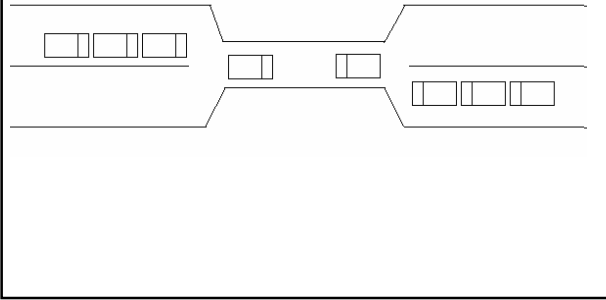
## Deadlock

### Ilustrasi Deadlock

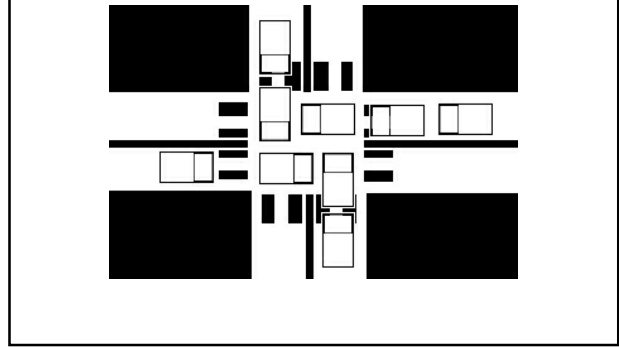
Terdapat dua proses P1 dan P2, dua sumber daya kritis R1 dan R2, Proses P1 dan P2 harus mengakses kedua sumber daya

Karena untuk melanjutkan eksekusi memerlukan sumber daya sekaligus maka kedua proses saling menunggu sumber daya lain selamanya. Tak ada proses yang dapat melepaskan sumber daya yang telah dipegangnya karena menunggu sumber daya yang lain yang tak pernah diperolehnya. Maka kedua proses dalam kondisi deadlock

## Deadlock



## Deadlock



## Metode Mengatasi Deadlock

- Metode pencegahan terjadinya deadlock
- Metode penghindaran terjadinya deadlock
- Metode Deteksi dan pemulihan dari deadlock

## Starvation

Deskripsi :Proses yang hak aksesnya digunakan oleh proses-proses yang lain secara terus menerus sehingga proses tersebut tidak akan dijalankan.

## Startvation

### Ilustrasi Startvation

Terdapat 3 proses P1, P2 dan P3 dan memerlukan sumber daya R secara periodik maka akan terjadi :

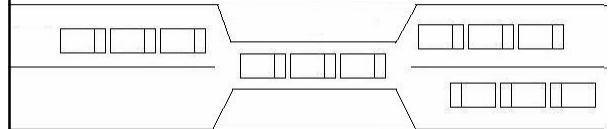
P1 sedang diberi sumber daya R, P2 dan P3 blocked menunggu sumber daya R

Ketika p1 keluar dari critical section, P2 dan p3 diijinkan mengakses R

Asumsi P3 diberi akses. Kemudian setelah selesai, hak akses dikembalikan ke P1 yang saat itu kembali membutuhkan sumber daya R

Jika hak akses digunakan oleh P1 dan P3 terus maka P2 tidak pernah mendapat hak akses sumber daya R pada situasi ini P2 mengalami startvation.

## Startvation



## Sinkronisasi

Pengendalian dari dari proses-proses yang saling berinteraksi. Penjaminan dari proses proses yang ada juga harus terpenuhi agar semua proses berjalan dengan baik.

## Prinsip-prinsip kongruensi

- Alokasi waktu pemroses untuk proses-proses
- Pemakaian bersama dan persaingan untuk mendapatkan sumber daya
- Komunikasi antar proses
- Sinkronisasi aktivitas banyak proses

### **Kesulitan yang ditimbulkan kongruensi**

#### **Pemakaian bersama sumber daya global**

Contoh : Jika dua proses menggunakan variabel global yang sama, serta keduanya membaca dan menulis variabel itu maka urutan terjadinya pembacaan dan penulisan terhadap variabel itu menjadi kritis

### **Kesulitan yang ditimbulkan kongruensi**

#### **Pengelolaan alokasi sumber daya optimal**

Contoh : Jika proses A meminta suatu kanal masukan/ inputan tertentu dan kemudian dapat terjadi proses A di suspend sebelum menggunakan kanal itu. Jika sistem operasi mengunci kanal untuk mencegah proses2 lain menggunakan kanal itu maka akan menghasilkan ketidak efisienan

### **Kesulitan yang ditimbulkan kongruensi**

#### **Pencarian kesalahan pemrograman**

Pencarian kesalahan pada pemrograman kongruen lebih sulit dibanding pencarian kesalahan pada program-program sekuen