

### ❑ Kondisi untuk mencapai Deadlock:

- 1/ mutual exclusion : jika suatu proses menggunakan suatu resource, tidak ada proses lain yang boleh menggunakan resource tersebut
- 2/ Hold and wait : pada saat suatu proses mengakses suatu resource, proses tersebut dapat meminta izin untuk mengakses resource lain
- 3/ non-preemption : jika suatu proses meminta izin untuk mengakses resource, sementara resource tidak tersedia, maka permintaan tidak dapat dibatalkan
- 4/ circular wait : jika proses  $P_i$  sedang mengakses resource  $R_i$ , dan meminta izin untuk mengakses resource  $R_j$ , dan pada saat bersamaan proses  $P_j$  sedang mengakses  $R_j$  dan meminta izin untuk mengakses resource  $R_i$ .

### ❑ Pendanganan Deadlock:

#### 1/ mengabaikan permasalahan (The ostrich algorithm)

yaitu sistem mengabaikan atau pura-pura tidak tahu kalau terjadi deadlock. Pendanganan ini mengasumsikan bahwa permasalahan akan sangat jarang terjadi sehingga kita dapat mengabaikannya jika muncul.

#### 2/ deteksi & pemulihan (recovery)

digunakan pada sistem yang mengizinkan terjadinya deadlock.

terbentuk ruf mendeteksi deadlock lalu memulihkan kembali.

Proses pendeteksian menghasilkan informasi apakah sistem sedang deadlock atau tidak serta proses mana yang mengalami deadlock setelah itu dilakukan metode pemulihan

#### 3/ Pencegahan dengan meniadakan salah satu dari empat kondisi deadlock.

Pengondisian sistem agar menghilangkan kemungkinan terjadinya

deadlock, pencegahan merupakan solusi yang bersih dari suatu terdapatnya deadlock

4/ pengalokasian sumber daya yg efisien

situasi ketika sumber daya dialokasikan pada pengguna nilai tertinggi mereka tidak ada alternatif untuk menggunakan sumber daya lebih lanjut tanpa membuat yang lain lebih buruk