

• Kondisi untuk mencapai deadlock

1.) Mutual Exclusion (mutual exclusion conditional)

↳ Jika suatu proses menggunakan suatu resource, tidak ada proses lain yang boleh menggunakan resource.

2.) Kondisi genggam & tunggu (hold and wait)

↳ pada saat suatu proses mengakses suatu resource, proses tersebut dapat meminta ijin untuk mengakses resource lain.

3.) non-preemption

↳ Jika suatu proses meminta ijin 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 ijin untuk mengakses R_j dan pada saat bersamaan proses P_j sedang mengakses R_j dan minta ijin untuk mengakses resource R_i

• Penanganan Deadlock

1.) Mengabaikan permasalahan (The Ostrich Algorithm)

↳ mengabaikan semua permasalahan sama sekali dan berpura-pura bahwa deadlock tidak pernah terjadi pada sistem

2.) Deteksi dan pemulihan (recovery)

↳ Deteksi digunakan pada sistem yang memungkinkan terjadinya deadlock, dengan memeriksa apakah terjadi deadlock dan menentukan proses dan sumber daya yang terlibat deadlock secara presisi. Begitu selesai, sistem dipulihkan sehingga beroperasi kembali.

3.) Pencegahan, dengan meniadakan salah satu dari empat kondisi deadlock

↳ pengondisian sistem agar menghilangkan kemungkinan terjadinya deadlock. Pencegahan merupakan solusi yang bersih dipandang dari sudut tercegahnya deadlock

1.) pengalokasian sumber daya yang efisien

↳ sumber daya yang dapat digunakan dengan aman oleh satu proses pada suatu saat