



Sistem Operasi Waktu Nyata [Part 2]

Sistem Waktu Nyata – 10th Meeting

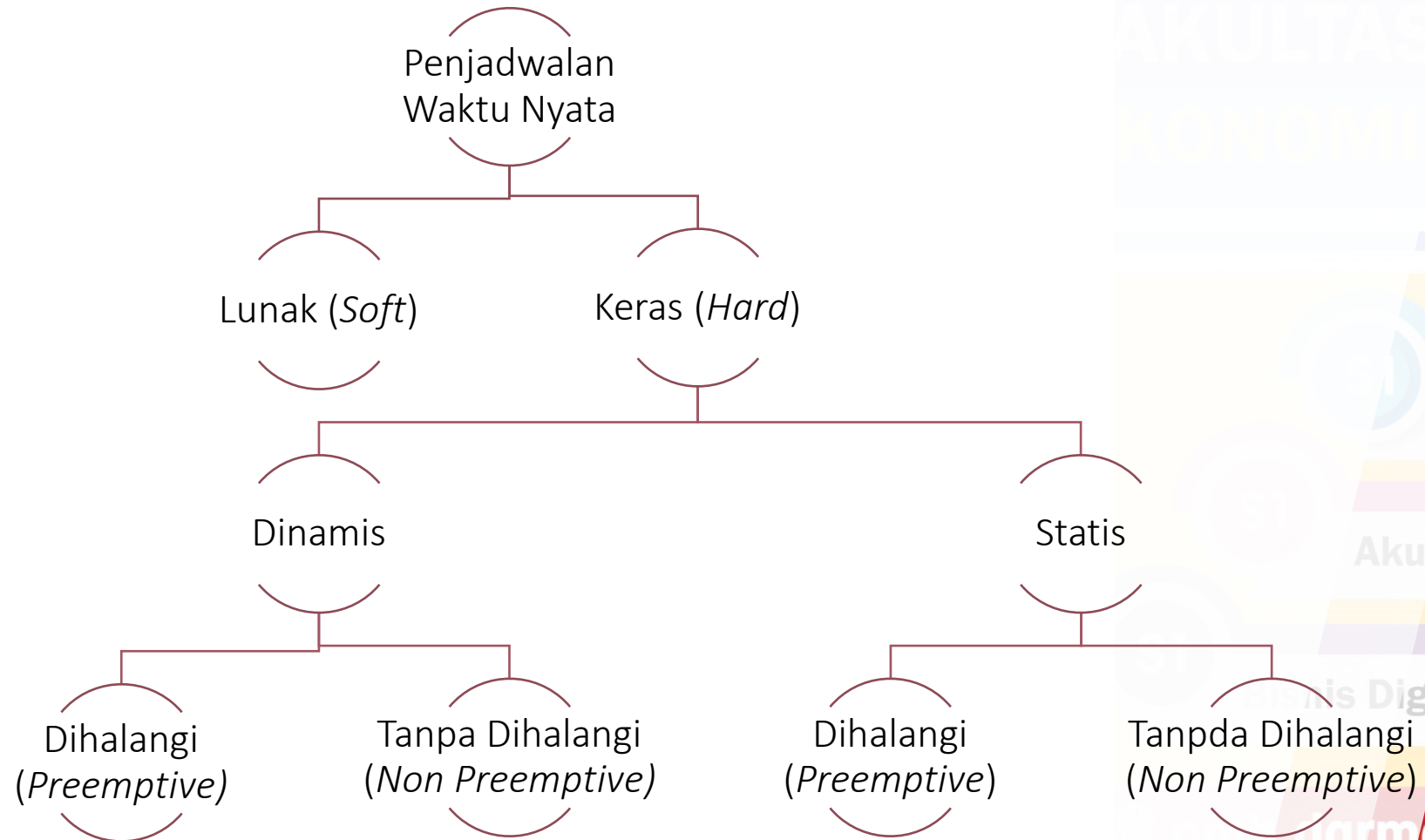
Penjadwalan Waktu-Nyata

- Penjadwalan (scheduling) merupakan komponen sistem operasi waktu-nyata paling penting.
- Penjadwalan terjadi apabila jumlah proses (task) yang aktif lebih banyak dibanding jumlah prosesor (CPU).
- Persoalan alokasi/penjadwalan berbunyi :
“Diberikan sekumpulan task-task, task mempunyai kendala mendahulukan (precedence constraint), kebutuhan sumber-daya (resources), karakteristik task, dan batas-waktu (deadline) tertentu. Diminta untuk mengusulkan suatu alokasi/jadwal yang cocok pada suatu sistem komputer yang diberikan”

Penjadwalan Waktu-Nyata

- Kebanyakan sistem waktu-nyata dirancang berdasarkan sasaran antara lain :
 - agar dapat memulai task waktu-nyata secepat mungkin;
 - menekankan pada penanganan selaan dan;
 - mengirim (*dispatch*) task ke prosesor dengan cepat.
- Pada sistem waktu-nyata keras (*hard real-time*), batas waktu (*deadline*) semua proses kritis harus dijamin pada semua situasi dan harus diantisipasi.
- Penjadwalan dinamis akan membuat keputusan pada waktu run (*run-time*) berbasis permintaan layanan.
- Sedangkan penjadwal static akan membuat keputusan-keputusan jadwal secara offline dibuat sebelum *run*.

Klasifikasi Penjadwalan Waktu-Nyata



ARMAMAJAYA
The Best

FAKULTAS
MANAJEMEN & BISNIS

Manajemen
Akuntansi
Sistem Informasi Digital
darmajaya.ac.id



Klasifikasi Penjadwalan Waktu-Nyata

- Penjadwalan dapat bersifat tanpa-dihalangi (*non preemptive*) atau dihalangi (*preemptive*).
- Pada penjadwalan bersifat tanpa-dihalangi (*non preemptive*), sekali suatu *task* memulai, *task* tersebut harus dieksekusi (*run*) sampai selesai atau sampai *task* diblokir karena menunggu/memerlukan sumber daya (*resource*).
- Pada penjadwalan dihalangi (*preemptive*), suatu *task* dapat disela oleh *task* lain yang lebih penting dan kemudian lain waktu dapat menggunakan prosesor lagi (*resumed*).
- Pada umumnya jika memungkinkan, suatu *task* yang kritis harus diizinkan untuk menyela *task* lain yang kurang kritis agar dapat memenuhi batas-waktu (*deadline*).



Penjadwalan Waktu-Nyata

- Pada jadwal tanpa dihalangi (*non preemptive*), dalam kebanyakan kasus, waktu kedatangan, waktu pemrosesan dan batas waktu (*deadline*) adalah sembarang serta tidak diketahui oleh penjadwal sampai task tiba.
- Tujuan algoritma tanpa-dihalangi (*non-preemptive*) adalah memaksimalkan jumlah *task* yang selesai; disini tidak dimungkinkan adanya jaminan kepastian pemrosesan; bisa saja tidak ditemukan jadwal yang layak (*feasible*).
- Pada jadwal dihalangi (*preemptive*) akan dijamin pemrosesan task-task secara periodik dan diperoleh jadwal yang layak; serta *priority inversion* dapat diminimalkan.
- *Priority inversion* adalah situasi dimana eksekusi task yang berprioritas tinggi ditunda oleh task berprioritas lebih rendah.

Penjadwalan Waktu-Nyata

- Kebanyakan sistem operasi yang bukan waktu-nyata (*nonreal-time*) mengimplementasikan penjadwalan dengan metode *round-robin*, dimana suatu *task* waktu-nyata ditambahkan ke antrian *run* untuk menunggu jatah pemakaian prosesor berikutnya (jatah waktu sama untuk semua *task*).
- Penjadwalan tanpa-dihalangi (*non-preemptive*) dikemudikan-prioritas (*priority-driven*) [hal 167].
- Penjadwalan dihalangi (*preemptive*) berdasarkan pada titik halangan (*preemption*) [hal 168].
- Penjadwalan dihalangi (*preemptive*) penuh [hal 169]
-

ARJANA

S2

S2

id.darmajaya.ac.id

Penjadwalan Dinamis

- Tiga Langkah dasar yang diperhatikan dalam penjadwalan dinamis adalah :
 1. Kebijakan penjadwalan
 2. Penyusunan jadwal
 3. Pengiriman (*dispatching*)



Model Kendala Waktu

- Kendala waktu (time constraint) suatu task periodik T dicirikan dengan parameter (s, e, d, p) , dimana s adalah titik mulai pemrosesan, e adalah waktu pemrosesan task T , d adalah deadline dari T , p adalah periode dari T , dimana berlaku $0 \leq e \leq d \leq p$.
- Utilisasi prosesor U adalah penjumlahan dari rasio waktu pemrosesan yang digunakan task-task yang dijamin dibanding dengan total waktu pemrosesan.

$$U = \frac{e_1}{p_1} + \frac{e_2}{p_2} + \dots + \frac{e_n}{p_n}$$

Algoritma Penjadwalan Waktu-Nyata

- Algoritma *Rate Monotonic Scheduling* (RMS)
- Algoritma *Earliest – Deadline – First* (EDF)
- Algoritma *Least – Laxity First* (LLF)



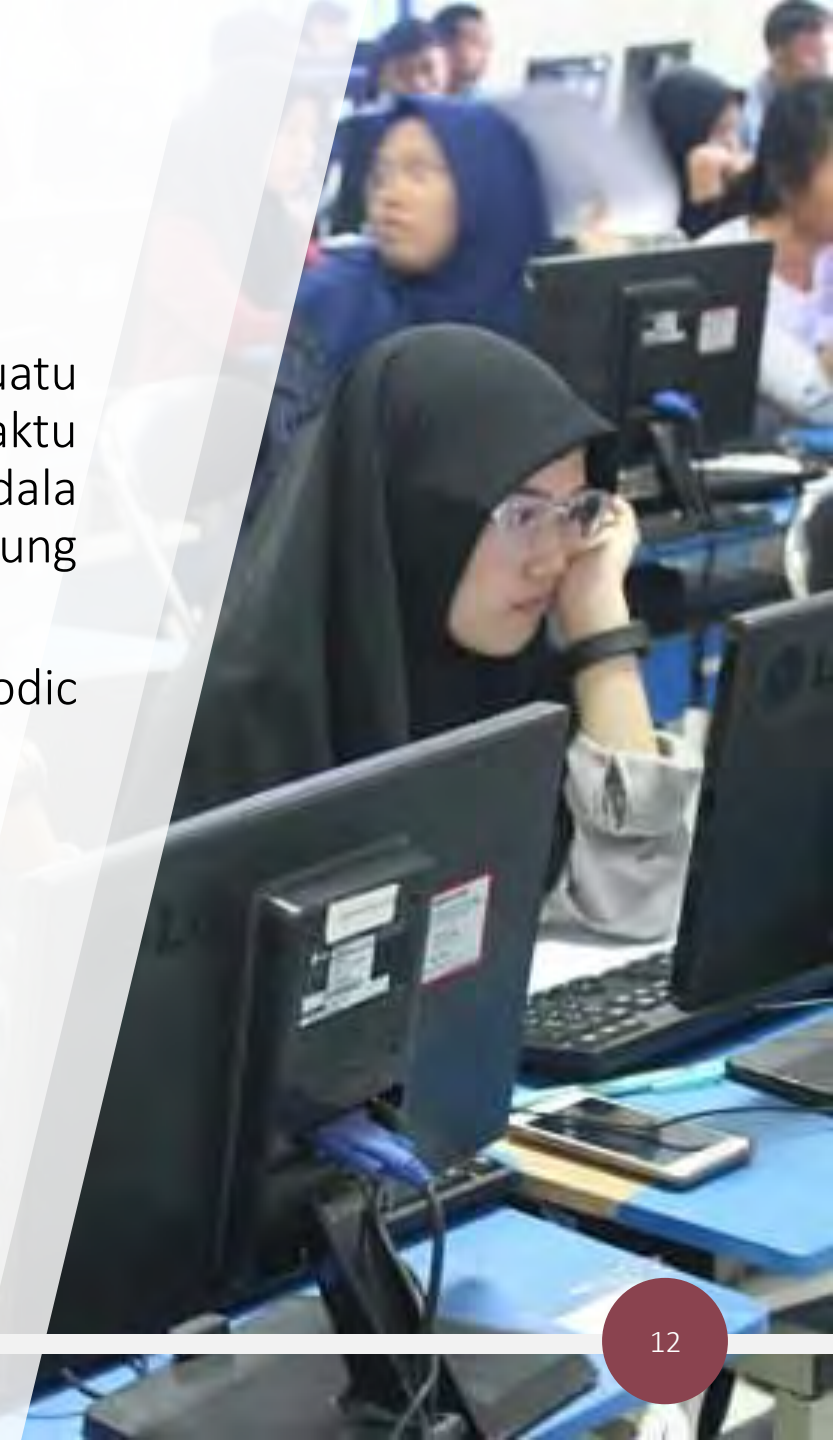
Penjadwalan Waktu-Nyata

- Penjadwalan Kernelized Monitor
- Penjadwalan Priority Ceiling Protocol

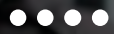


Penjadwalan Statis

- Pada penjadwalan statis atau sebelum waktu eksekusi (*pre run time*), suatu jadwal yang layak dari sekumpulan *task* yang menjamin semua batas-waktu (*deadline*), dengan mempertimbangkan sumber daya, kendala mendahulukan (*precedence*) dan kebutuhan sinkronisasi, akan dihitung secara *offline*.
- Penjadwalan statis umumnya merupakan jadwal yang dipicu waktu periodic (*periodic time triggered schedule*).



THANK YOU



Nurfiana, S.Kom., M.Kom



(0721) 787214



nurfiana@darmajaya.ac.id



<https://www.darmajaya.ac.id/>