CPU Scheduling

Week 4: Contoh Soal dan Penyelesaian MLFQ

lectura.id/course/os

Program Studi Teknik Informatika
Institut Teknologi Sumatera

2026

OUTLINE

Peta materi pertemuan ini

1. **Review Aturan MLFQ**
2. **Contoh Soal 1 (Dasar)**
3. **Penyelesaian Soal 1**
4. **Contoh Soal 2 (Lanjut)**
5. **Penyelesaian Soal 2**
6. **Soal Tantangan 1**
7. **Soal Tantangan 2**
8. **Kesimpulan**
9. **Latihan Mandiri**

Capaian Pembelajaran

Kemampuan yang ditargetkan setelah kuliah

1. Mengingat kembali aturan dasar Multi-Level Feedback Queue (MLFQ).
2. Memahami cara kerja MLFQ melalui simulasi contoh soal.
3. Mampu menelusuri (tracing) eksekusi proses langkah demi langkah.
4. Memahami bagaimana prioritas proses berubah seiring waktu.

Target Akhir

Setelah pertemuan ini, mahasiswa tidak hanya tahu teori MLFQ, tetapi **bisa menyelesaikan** soal penjadwalan MLFQ dengan benar dan sistematis.

Mengingat Kembali MLFQ

5 Aturan Emas MLFQ

Sebelum masuk ke contoh soal, mari kita ingat kembali aturan main MLFQ dengan bahasa yang sederhana:

1. **Aturan 1:** Jika Prioritas A > Prioritas B, maka A berjalan (B menunggu).
2. **Aturan 2:** Jika Prioritas A = Prioritas B, A dan B berjalan bergantian (Round-Robin).
3. **Aturan 3:** Pekerjaan **baru** selalu masuk ke antrean paling atas (prioritas tertinggi).
4. **Aturan 4:** Jika pekerjaan menghabiskan jatah waktunya di suatu antrean, ia akan **"turun kelas"** ke antrean di bawahnya.
5. **Aturan 5 (Boost):** Setelah periode waktu tertentu (S), **semua** pekerjaan dikembalikan ke antrean paling atas agar tidak ada yang kelaparan (starvation).

Apakah Aturan MLFQ Selalu Sama?

Variasi Implementasi di Dunia Nyata

Tidak selalu! Konsep dasarnya sama, tetapi **parameter dan detail aturannya** bisa berbeda-beda tergantung sistem operasi (Windows, Solaris, dll).

- **Jumlah Antrean:** Bisa 3, 4, atau bahkan 60 tingkat antrean.
- **Jatah Waktu (Time Quantum):** Biasanya semakin rendah prioritasnya, jatah waktunya semakin **panjang** (contoh: Q3=10ms, Q2=20ms, Q1=40ms).
- **Interval Boost (*S*):** Kapan semua pekerjaan diangkat kembali ke atas? Jika terlalu cepat, pekerjaan interaktif terganggu. Jika terlalu lambat, terjadi *starvation*.
- **Aturan I/O (Accounting):** Di MLFQ modern, sistem **mengingat** total waktu CPU yang dipakai di suatu tingkat, meskipun pekerjaan tersebut sering berhenti untuk I/O (mencegah *gaming the system*).

Skenario Sistem (Soal 1)

Spesifikasi Antrean MLFQ

Bayangkan kita memiliki sistem MLFQ dengan **3 tingkat antrean**:

- **Q3 (Prioritas Tertinggi):** Jatah waktu (Time Quantum) = **10 ms**
- **Q2 (Prioritas Menengah):** Jatah waktu (Time Quantum) = **20 ms**
- **Q1 (Prioritas Terendah):** Jatah waktu (Time Quantum) = **40 ms**

Aturan Tambahan

Setiap **50 ms**, sistem melakukan **Priority Boost** (Aturan 5). Semua pekerjaan yang belum selesai akan diangkat kembali ke Q3.

Daftar Pekerjaan (Jobs)

Siapa saja yang akan dieksekusi?

Terdapat 3 pekerjaan yang datang ke sistem kita:

Nama Job	Waktu Datang (Arrival)	Kebutuhan CPU (Burst)
Job A	0 ms	45 ms
Job B	15 ms	15 ms
Job C	22 ms	5 ms

Mari kita simulasikan perjalanannya **langkah demi langkah**!

Langkah 1: Waktu = 0 ms

Kedatangan Job A

- **Kejadian:** Job A datang.
- **Aksi:** Sesuai Aturan 3, Job A masuk ke antrean paling atas (Q3).
- **Status Antrean:**
  - Q3: [**A**]
  - Q2: [ ]
  - Q1: [ ]
- **Eksekusi:** CPU mulai menjalankan Job A. Jatah waktu Q3 adalah 10 ms.

Catatan

Sisa waktu Job A = 45 ms.

Langkah 2: Waktu = 10 ms

Job A Turun Kelas

- **Kejadian:** Job A sudah berjalan selama 10 ms.
- **Aksi:** Job A menghabiskan jatah waktu Q3 (10 ms). Sesuai Aturan 4, Job A **"turun kelas"** ke Q2.
- **Status Antrean:**
  - Q3: [ ]
  - Q2: [**A**]
  - Q1: [ ]
- **Eksekusi:** Karena Q3 kosong, CPU lanjut menjalankan Q2. Job A kembali berjalan. Jatah waktu Q2 adalah 20 ms.

Catatan

Sisa waktu Job A = 45 - 10 = **35 ms**.

Langkah 3: Waktu = 15 ms

Kedatangan Job B (Interupsi!)

- **Kejadian:** Job B datang!
- **Aksi:** Job B masuk ke Q3. Saat ini Job A sedang berjalan di Q2 (baru jalan 5 ms). Karena ada pekerjaan di Q3, CPU **menghentikan** Job A sementara (Aturan 1).
- **Status Antrean:**
  - Q3: [**B**]
  - Q2: [**A**] (A baru pakai 5 ms dari jatah 20 ms-nya)
  - Q1: [ ]
- **Eksekusi:** CPU menjalankan Job B dari Q3.

Catatan

Sisa waktu Job A = 35 - 5 = **30 ms**. Sisa jatah A di Q2 = **15 ms**.
Sisa waktu Job B = **15 ms**.

Langkah 4: Waktu = 22 ms

Kedatangan Job C

- **Kejadian:** Job C datang!
- **Aksi:** Job C masuk ke Q3. Saat ini Job B sedang berjalan di Q3 (sudah jalan 7 ms). Job C antre di belakang Job B.
- **Status Antrean:**
  - Q3: [**B**, **C**] (B sedang jalan)
  - Q2: [**A**]
  - Q1: [ ]
- **Eksekusi:** CPU lanjut menjalankan Job B sampai jatah waktu Q3-nya (10 ms) habis.

Catatan

Sisa waktu Job C = **5 ms**.

Langkah 5: Waktu = 25 ms

Job B Turun Kelas

- **Kejadian:** Job B sudah berjalan 10 ms (dari waktu 15 ke 25).
- **Aksi:** Job B menghabiskan jatah waktu Q3. Job B **"turun kelas"** ke Q2, antre di belakang A.
- **Status Antrean:**
  - Q3: [**C**]
  - Q2: [**A**, **B**]
  - Q1: [ ]
- **Eksekusi:** CPU melihat Q3 ada isinya (Job C). CPU mulai menjalankan Job C.

Catatan

Sisa waktu Job B = 15 - 10 = **5 ms**.

Langkah 6: Waktu = 30 ms

Job C Selesai!

- **Kejadian:** Job C berjalan selama 5 ms (dari 25 ke 30).
- **Aksi:** Kebutuhan Job C hanya 5 ms. Jadi, Job C **SELESAI** dan keluar dari sistem!
- **Status Antrean:**
  - Q3: [ ]
  - Q2: [**A**, **B**]
  - Q1: [ ]
- **Eksekusi:** Q3 kosong. CPU turun ke Q2. Di Q2 ada A dan B. Giliran Job A yang jalan (karena A masuk Q2 lebih dulu).

Catatan

Job C selesai pada waktu **30 ms**.

Langkah 7: Waktu = 45 ms

Job A Turun Kelas (Lagi)

- **Kejadian:** Job A melanjutkan sisa jatah waktunya di Q2 (15 ms). Ia berjalan dari waktu 30 ke 45.
- **Aksi:** Job A menghabiskan total jatah waktu Q2 (20 ms). Job A **"turun kelas"** ke Q1.
- **Status Antrean:**
  - Q3: [ ]
  - Q2: [**B**]
  - Q1: [**A**]
- **Eksekusi:** Q3 kosong, Q2 ada isinya. CPU menjalankan Job B di Q2.

Catatan

Sisa waktu Job A = 30 - 15 = **15 ms**.

Langkah 8: Waktu = 50 ms

Job B Selesai & PRIORITY BOOST!

- **Kejadian 1:** Job B berjalan 5 ms (dari 45 ke 50). Sisa waktu B habis. Job B **SELESAI**!
- **Kejadian 2:** Waktu menunjukkan **50 ms**. Waktunya **Priority Boost** (Aturan 5)!
- **Aksi:** Semua pekerjaan yang tersisa (hanya Job A) diangkat kembali ke Q3.
- **Status Antrean:**
  - Q3: [**A**] (Naik dari Q1)
  - Q2: [ ]
  - Q1: [ ]
- **Eksekusi:** CPU menjalankan Job A di Q3.

Catatan

Job B selesai pada waktu **50 ms**. Sisa waktu Job A = **15 ms**.

Langkah 9: Waktu = 60 ms

Job A Turun Kelas (Setelah Boost)

- **Kejadian:** Job A berjalan 10 ms di Q3 (dari 50 ke 60).
- **Aksi:** Job A menghabiskan jatah waktu Q3. Job A **"turun kelas"** ke Q2.
- **Status Antrean:**
  - Q3: [ ]
  - Q2: [**A**]
  - Q1: [ ]
- **Eksekusi:** CPU lanjut menjalankan Job A di Q2.

Catatan

Sisa waktu Job A = 15 - 10 = **5 ms**.

Langkah 10: Waktu = 65 ms

Semua Selesai!

- **Kejadian:** Job A berjalan 5 ms lagi (dari 60 ke 65).
- **Aksi:** Sisa waktu Job A habis. Job A **SELESAI**!
- **Status Antrean:**
  - Q3: [ ]
  - Q2: [ ]
  - Q1: [ ]
- **Eksekusi:** Sistem kosong. CPU menganggur (idle).

Catatan

Job A selesai pada waktu **65 ms**. Simulasi berakhir!

Skenario Sistem (Soal 2)

Tantangan Lebih Berat: 5 Pekerjaan

Aturan sistem masih sama:

- **Q3:** Jatah waktu = **10 ms**
- **Q2:** Jatah waktu = **20 ms**
- **Q1:** Jatah waktu = **40 ms**
- **Priority Boost:** Setiap **50 ms**

Nama Job	Waktu Datang (Arrival)	Kebutuhan CPU (Burst)
Job A	0 ms	30 ms
Job B	15 ms	20 ms
Job C	25 ms	15 ms
Job D	35 ms	5 ms
Job E	42 ms	10 ms

Langkah 1: Waktu 0 - 15 ms

Awal Mula

- **0 ms:** Job A datang ke Q3. CPU menjalankan A.
- **10 ms:** Job A menghabiskan jatah Q3 (10 ms). A **turun kelas** ke Q2. CPU lanjut menjalankan A di Q2.
- **15 ms:** Job B datang ke Q3! Karena Q3 lebih tinggi dari Q2, Job B **menginterupsi** Job A.

Status di 15 ms

Q3: [**B**] (sedang jalan)
Q2: [**A**] (A sudah jalan 10ms di Q3 + 5ms di Q2. Sisa burst A = 15ms)
Q1: [ ]

Langkah 2: Waktu 15 - 35 ms

Kedatangan Beruntun

- **25 ms:** Job B menghabiskan jatah Q3 (10 ms). B **turun kelas** ke Q2. Bersamaan dengan itu, **Job C datang** ke Q3! CPU menjalankan C.
- **35 ms:** Job C menghabiskan jatah Q3 (10 ms). C **turun kelas** ke Q2. Bersamaan dengan itu, **Job D datang** ke Q3! CPU menjalankan D.

Status di 35 ms

Q3: [**D**] (sedang jalan)
Q2: [**A**, **B**, **C**] (Antre sesuai urutan masuk Q2)
Q1: [ ]

Langkah 3: Waktu 35 - 42 ms

Pekerjaan Pendek Selesai

- **40 ms:** Job D hanya butuh 5 ms. Job D **SELESAI**! Q3 kosong. CPU turun ke Q2 dan melanjutkan Job A (karena A paling depan di Q2).
- **42 ms:** Job E datang ke Q3! Job E **menginterupsi** Job A lagi.

Status di 42 ms

Q3: [**E**] (sedang jalan)
Q2: [**A**, **B**, **C**] (A baru jalan 2ms lagi di Q2. Total A jalan = 17ms. Sisa burst A = 13ms)
Q1: [ ]

Langkah 4: Waktu 50 ms
PRIORITY BOOST!

- **50 ms:** Waktunya **Priority Boost**! Semua pekerjaan yang ada di Q2 dan Q1 diangkat kembali ke Q3.
- Saat ini Job E sedang berjalan (sudah jalan 8 ms, sisa 2 ms). Job A, B, dan C naik ke Q3 di belakang E.

Status di 50 ms

Q3: [**E**, **A**, **B**, **C**] (E lanjut jalan)
Q2: [ ]
Q1: [ ]

Langkah 5: Waktu 50 - 80 ms

Penyelesaian Akhir

- **52 ms:** Job E selesai (butuh 2 ms lagi). CPU lanjut ke Job A.
- **62 ms:** Job A menghabiskan jatah Q3 (10 ms). A turun ke Q2. CPU lanjut ke Job B. (Sisa burst A = 3ms).
- **72 ms:** Job B menghabiskan jatah Q3 (10 ms). Karena sisa burst B memang 10 ms, Job B **SELESAI**! CPU lanjut ke Job C.
- **77 ms:** Job C selesai (sisa burst C hanya 5 ms). Q3 kosong.
- **80 ms:** CPU turun ke Q2, menjalankan sisa Job A (3 ms). Job A **SELESAI**!

Ringkasan Waktu Penyelesaian (Soal 2)

Kapan setiap pekerjaan selesai?

Nama Job	Waktu Datang	Burst Time	Waktu Selesai
Job D	35 ms	5 ms	**40 ms**
Job E	42 ms	10 ms	**52 ms**
Job B	15 ms	20 ms	**72 ms**
Job C	25 ms	15 ms	**77 ms**
Job A	0 ms	30 ms	**80 ms**

Pelajaran Penting

Meskipun Job A datang pertama, ia selesai paling akhir karena butuh waktu lama. Job D dan E yang datang belakangan tapi butuh waktu singkat bisa selesai dengan cepat berkat interupsi dan *Priority Boost*.

Soal Tantangan 1: Perbandingan 5 Algoritme

FIFO, SJF, STCF, RR, dan MLFQ

Diberikan 5 pekerjaan dengan spesifikasi berikut:

Job	Arrival Time	Burst Time
P1	0	10
P2	2	5
P3	4	2
P4	6	8
P5	8	4

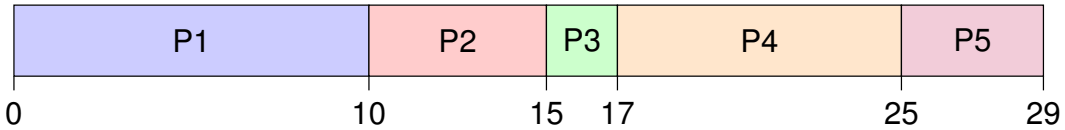
Aturan:

- **RR:** Quantum = 2
- **MLFQ:** Q3 (q=2), Q2 (q=4), Q1 (q=8). Boost setiap 20 ms.

Penyelesaian Soal 1: FIFO & SJF

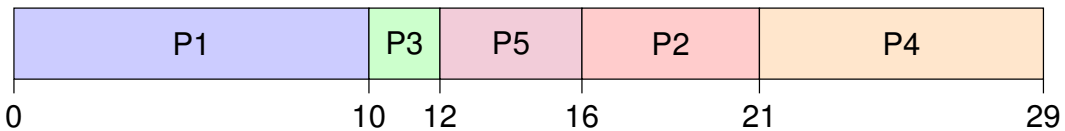
1. FIFO (First In First Out)

Pekerjaan dieksekusi murni berdasarkan urutan kedatangan.



2. SJF (Shortest Job First - Non Preemptive)

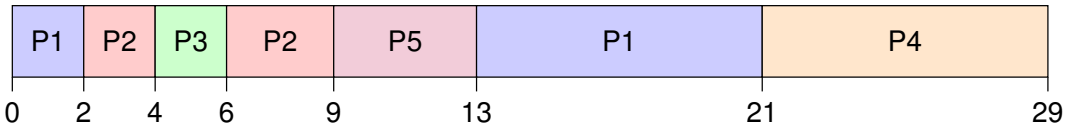
P1 berjalan sampai selesai. Setelah itu, pilih yang terpendek dari yang sudah datang.



Penyelesaian Soal 1: STCF & RR

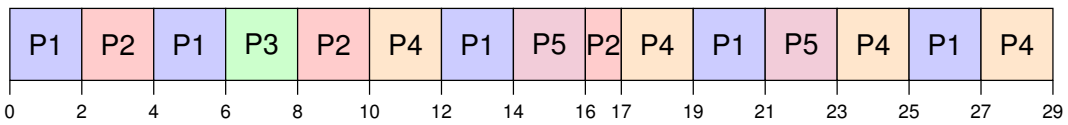
3. STCF (Shortest Time-to-Completion First)

Pekerjaan baru bisa menginterupsi jika sisa waktunya lebih pendek.



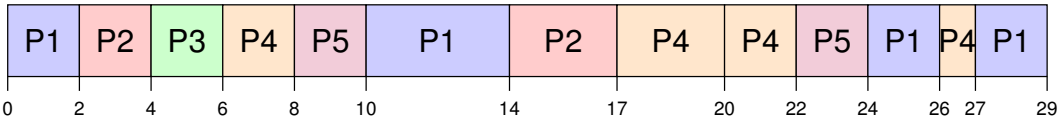
4. Round Robin (q=2)

Bergantian setiap 2 ms. Adil, tapi banyak *context switch*.



Penyelesaian Soal 1: MLFQ

5. MLFQ (Q3=2, Q2=4, Q1=8, Boost=20)
Pekerjaan baru masuk Q3. Jika jatah habis, turun kelas. Boost di ms ke-20.



Pembahasan

Pada MLFQ, P3 (pekerjaan terpendek) selesai sangat cepat (di ms ke-6) tanpa perlu tahu durasinya di awal. P4 sempat terpotong di ms ke-20 karena ada *Priority Boost*, sehingga ia kembali ke Q3 dan langsung dieksekusi lagi.

Analisis Metrik Soal Tantangan 1

Perbandingan Rata-rata RT, TAT, dan WT

Berdasarkan Gantt Chart sebelumnya, kita hitung rata-rata Response Time (RT), Turnaround Time (TAT), dan Waiting Time (WT) untuk 5 pekerjaan:

Algoritme	Avg Response Time	Avg Turnaround Time	Avg Waiting Time
FIFO	9.4 ms	15.2 ms	9.4 ms
SJF	7.8 ms	13.6 ms	7.8 ms
STCF	3.2 ms	**11.6 ms**	**5.8 ms**
RR (q=2)	2.4 ms	16.8 ms	11.0 ms
MLFQ	**0.0 ms**	16.6 ms	10.8 ms

Kesimpulan Soal 1

- **Terbaik untuk Turnaround Time: STCF**. Karena STCF selalu mendahulukan sisa pekerjaan terpendek, rata-rata waktu selesainya paling optimal.
- **Terbaik untuk Response Time: MLFQ**. Karena setiap pekerjaan baru langsung diletakkan di antrean tertinggi (Q3), semua pekerjaan langsung direspons seketika (RT = 0).

Soal Tantangan 2: Perbandingan 5 Algoritme

Kasus dengan Jeda Kedatangan

Diberikan 5 pekerjaan dengan spesifikasi berikut:

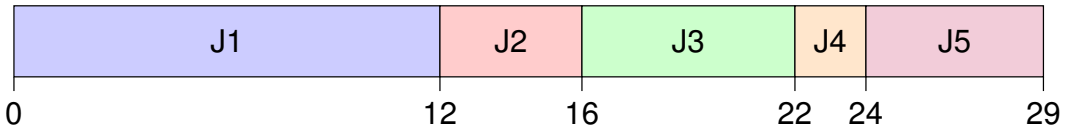
Job	Arrival Time	Burst Time
J1	0	12
J2	3	4
J3	5	6
J4	9	2
J5	12	5

Aturan:

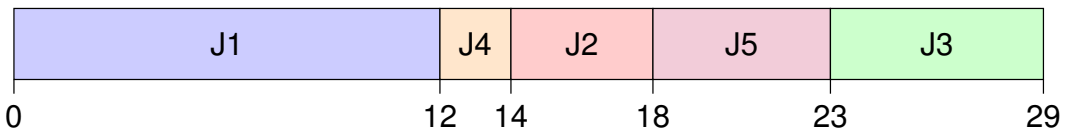
- **RR:** Quantum = 3
- **MLFQ:** Q3 (q=3), Q2 (q=5), Q1 (q=10). Boost setiap 30 ms.

Penyelesaian Soal 2: FIFO & SJF

1. FIFO (First In First Out)

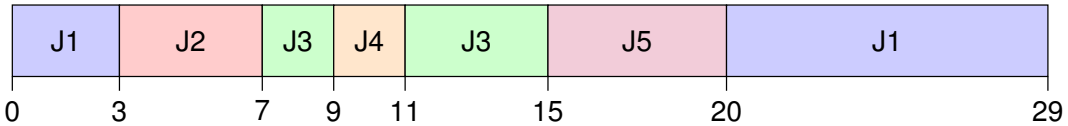


2. SJF (Shortest Job First - Non Preemptive)

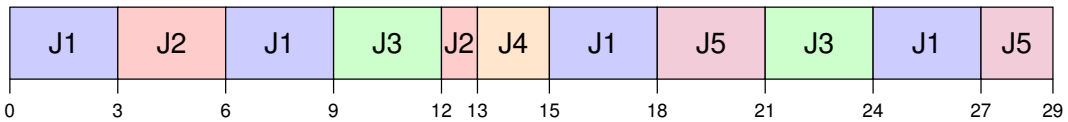


Penyelesaian Soal 2: STCF & RR

3. STCF (Shortest Time-to-Completion First)

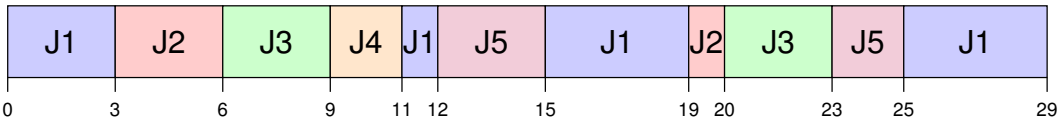


4. Round Robin (q=3)



Penyelesaian Soal 2: MLFQ

5. MLFQ (Q3=3, Q2=5, Q1=10, Boost=30)



Pembahasan

Pada MLFQ, J1 yang panjang terus turun kelas. Saat J5 datang di ms ke-12, J5 masuk ke Q3 dan langsung menginterupsi J1 yang saat itu sedang berjalan di Q2. Ini membuktikan MLFQ sangat responsif terhadap pekerjaan baru yang interaktif.

Analisis Metrik Soal Tantangan 2

Perbandingan Rata-rata RT, TAT, dan WT

Perhitungan rata-rata metrik untuk Soal 2 (kasus dengan jeda kedatangan):

Algoritme	Avg Response Time	Avg Turnaround Time	Avg Waiting Time
FIFO	9.0 ms	14.8 ms	9.0 ms
SJF	7.6 ms	13.4 ms	7.6 ms
STCF	1.0 ms	**10.6 ms**	**4.8 ms**
RR (q=3)	2.8 ms	15.8 ms	10.0 ms
MLFQ	**0.2 ms**	15.8 ms	10.0 ms

Kesimpulan Soal 2

- **STCF** kembali menjadi juara mutlak untuk **Turnaround Time** dan **Waiting Time**.
- **MLFQ** sangat unggul di **Response Time** (0.2 ms), hampir menyamai STCF, namun tanpa perlu mengetahui durasi pekerjaan (burst time) di awal! Inilah alasan MLFQ dipakai di OS modern.

Ringkasan Waktu Penyelesaian (Contoh Dasar)

Kapan setiap pekerjaan selesai?

Dari simulasi Contoh Dasar tadi, kita mendapatkan waktu selesai (Completion Time) untuk masing-masing pekerjaan:

Nama Job	Waktu Datang	Burst Time	Waktu Selesai
Job C	22 ms	5 ms	**30 ms**
Job B	15 ms	15 ms	**50 ms**
Job A	0 ms	45 ms	**65 ms**

Pelajaran Penting

Perhatikan bagaimana Job C (pekerjaan pendek) bisa selesai dengan cepat meskipun datang belakangan. Inilah kehebatan MLFQ: **memprioritaskan pekerjaan pendek/interaktif** tanpa perlu tahu durasinya di awal!

Latihan Mandiri

Uji Pemahaman Anda!

Selesaikan soal berikut menggunakan 5 algoritme (FIFO, SJF, STCF, RR, MLFQ) dan tentukan algoritme mana yang memberikan **Turnaround Time** terbaik!

Job	Arrival Time	Burst Time
T1	0	8
T2	2	6
T3	4	2
T4	5	4
T5	8	10

Aturan:

- **RR:** Quantum = 4
- **MLFQ:** Q3 (q=2), Q2 (q=4), Q1 (q=8). Boost setiap 25 ms.

Ada Pertanyaan?

Jangan ragu untuk mengulang simulasi ini di atas kertas agar lebih paham!