

# IMPLEMENTASI ALGORITMA MULTILEVEL FEEDBACK QUEUE DALAM MEMENTUKAN WAKTU TUNGGU DAN WAKTU KESELURUHAN PROSES

**Winda Sulastri**

Dosen Tetap AMIK ROYAL Kisaran  
Jl. Tuanku Imam Bonjol No. 179 Kisaran  
[www.Royal.ac.id/](http://www.Royal.ac.id/) Email: [stmik.Royal@yahoo.co.id](mailto:stmik.Royal@yahoo.co.id)

## ABSTRAK

Penjadwalan merupakan protokol atau ketentuan untuk menyelesaikan process berdasarkan kinerja CPU. Sistem Operasi melakukan polling kepada setiap Program counter Process yang berada dalam antrian Ready (Ready Queue). Jika diketahui terdapat nilai alamat dalam program counter process maka process tersebut layak untuk dijalankan (Running). Running adalah pengerjaan instruksi process berdasarkan waktu pelayanan (bursttime) pada CPUtime atau I/Otime. Salah satu penjadwalan yang digunakan untuk menyelesaikan process-process adalah algoritma penjadwalan Multilevel Feedback Queue (MLFQ). MLFQ menerapkan beberapa algoritma-algoritma yaitu algoritma First Come First Serve (FCFS), Round Robin (RR), Priority Scheduler (PS) dan Multilevel Queue (MLQ) dan aturan perpindahan ke bawah level antrian persatu antrian. Pada tulisan ini akan membahas penyelesaian process-process dengan algoritma multilevel Feedback Queue (MFLQ) dan memberlakukan Running process per-quantum time antrian process sehingga dapat menentukan waktu tunggu (WaitingTime) dan waktu keseluruhan (TurnAroundTime) yang dimiliki process.

**Kata Kunci** : Multilevel Feedback Queue (MLFQ), Queue, Quantum Time, Sistem Operasi

## PENDAHULUAN

Penjadwalan merupakan mekanisme dari mesin abstrak dalam menentukan process yang akan dikerjakan di antara process yang ada antrian ready (ready queue).

Pengerjaan Process berdasarkan yang dapat digantikan process lain

disebut tipe penjadwalan Preemptive sedangkan tipe penjadwalan Non-Preemptive yaitu pengerjaan Process yang tidak dapat digantikan process lain. Dari tipe penjadwalan tersebut maka diciptakan algoritma penjadwalan seperti First Come First Served, Shortest Job Remains, Round Robin, Priority Scheduler, Multilevel Queue, Quarantee Scheduler dan Multilevel Feedback Queue.

Algoritma Penjadwalan Multilevel Feedback Queue (MLFQ) adalah algoritma yang memiliki urutan level antrian dan proses dapat berpindah dari level antrian awal ke level-level antrian berikutnya. Algoritma MLFQ memiliki beberapa antrian secara urutan jumlah nilai waktu jatah (QuantumTime) berbeda. Secara urutan waktu antrian level terkecil memiliki waktu jatah lebih kecil dari waktu jatah antrian lainnya.

Penyelesaian secara multi level antrian membuat proses yang mempunyai waktu pengerjaan (BurstTime / Service Time) besar akan memiliki pecahan-pecahan BurstTime dari QuantumTime di setiap level antrian. Sehingga mencari waktu tunggu dan waktu keseluruhan sulit diterapkan pada algoritma MLFQ. Maka perlu pendefinisian pecahan BurstTime dalam bentuk variabel-variabel Pengerjaan (Running Process).

## LANDASAN TEORI

### Sistem Operasi

Pengertian sistem operasi secara umum ialah pengelola seluruh sumberdaya yang terdapat pada sistem komputer dan menyediakan sekumpulan layanan (*system calls*) yang sering disebut "tools atau utility" berupa aplikasi ke pemakai sehingga memudahkan dan menyamankan penggunaan ketika Memanfaatkan

sumber-daya sistem komputer tersebut (Turban, 1995).

Sistem Operasi merupakan program utama yang menghubungkan Software Aplikasi yang digunakan oleh user dengan hardware (Turban, 1995).

Sistem Operasi yang sangat populer pada sistem komputer pada milenium sekarang ini yaitu Windows, Linux, dan Mac OS. Sedangkan untuk komputer mobile yaitu Symbian, Android, RIM Black Berry, dan Bada.

### Penjadwalan

Penjadwalan proses merupakan kumpulan kebijaksanaan dan mekanisme di sistem operasi yang berkaitan dengan urutan kerja yang dilakukan sistem komputer.

Terdapat dua tipe penjadwalan (Scheduler) pada sistem operasi, yaitu:

1. Preemptive, proses yang dapat digantikan atau bersifat kooperatif.
2. Non-Preemptive, proses yang tidak dapat digantikan disebut juga independent.

### Algoritma Penjadwalan

Terdapat berbagai jenis penjadwalan antara lain:

1. First Come First Serve (FCFS)  
FCFS merupakan algoritma penjadwalan yang sederhana yang digunakan CPU. Dengan menggunakan algoritma FCFS setiap proses yang berada pada status *ready* dimasukkan ke dalam antrian. FIFO sesuai dengan waktu kedatangannya. Proses yang tiba terlebih dahulu yang akan dieksekusi terlebih dahulu (Martin and Steven, 1988).
2. Shortest Job First (SJF)  
Dengan algoritma ini maka setiap proses yang ada di antrian *ready* akan dieksekusi berdasarkan *burst time* terkecil. Hal ini mengakibatkan *waiting time* yang pendek untuk setiap proses dan karena hal tersebut maka *waiting time* rata-ratanya juga menjadi pendek, sehingga dapat dikatakan bahwa algoritma ini adalah algoritma yang optimal (Turban, 1995).

Ada beberapa kekurangan dari algoritma ini yaitu:

1. Kesulitan untuk memprediksi *burst time* proses yang akan dieksekusi selanjutnya.
2. Proses yang mempunyai *burst time* yang besar akan memiliki *waiting time* yang besar pula karena yang dieksekusi terlebih dahulu adalah proses dengan *burst time* yang lebih kecil (Martin and Steven, 1988).

### 3. Round Robin

Algoritma ini didesain untuk sistem *time-sharing*. Proses akan mendapat jatah sebesar *time quantum* dengan nilai *quantum* umumnya sebesar 10-100 ms. Jika *quantum*nya habis atau proses sudah selesai CPU akan dialokasikan ke proses berikutnya. Tentu proses ini cukup adil karena tak ada proses yang di-prioritaskan, semua proses mendapat jatah waktu yang sama dari CPU ( $1/n$ ), dan tak akan menunggu lebih lama dari  $(n-1)/q$ .

Algoritma ini sepenuhnya bergantung besarnya *time quantum*. Jika terlalu besar, algoritma ini akan sama saja dengan algoritma *first-come-first-served*. Jika terlalu kecil, akan semakin banyak peralihan proses sehingga banyak waktu terbuang (Martin and Steven, 1988).

### 4. Priority Scheduling (PS)

Priority Scheduling merupakan algoritma penjadwalan yang mendahulukan proses dengan nilai prioritas tertinggi. Setiap proses memiliki prioritasnya masing-masing. Prioritas suatu proses dapat ditentukan melalui beberapa karakteristik antara lain:

- a. Batas waktu
- b. Kebutuhan Memori
- c. Akses file
- d. Perbandingan antara I/O *Burst* dengan CPU *Burst*.
- e. Tingkat kepentingan proses

Penjadwalan dengan prioritas juga dapat dijalankan secara preemptive maupun non-preemptive (Martin and Steven, 1988).

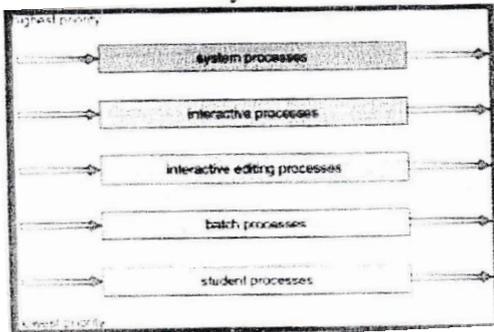
Pada preemptive, jika ada suatu proses yang baru datang memiliki prioritas yang lebih tinggi daripada proses yang sedang dijalankan, maka

proses yang sedang berjalan tersebut dihentikan, lalu CPU dialihkan untuk proses yang baru datang tersebut. Sementara itu, pada non-preemptive, proses yang baru datang tidak dapat mengganggu proses yang sedang berjalan, tetapi hanya diletakkan di depan antrian.

Kelemahan pada pwnjadwalan prioritas adalah dapat terjadi *indefinite blocking* (starvation) yaitu suatu proses dengan prioritas yang rendah memiliki kemungkinan untuk tidak dieksekusi jika terdapat proses lain yang memiliki prioritas lebih tinggi darinya. Solusi dari permasalahan ini adalah *Paging*, yaitu meningkatkan prioritas dari setiap yang menunggu dalam antrian secara bertahap (Martin and Steven, 1988).

#### 5. Multilevel Queue (MLQ)

Ide dasar dari algoritma ini adalah berdasarkan pada sistem prioritas proses. Prinsipnya adalah, jika setiap proses dapat dikelompokkan berdasarkan prioritasnya.



Gambar 1. Penjadwalan MLQ

#### 6. Multilevel Feedback Queue (MLFQ)

Algoritma ini mirip sekali dengan algoritma Multilevel Queue. Perbedaannya ialah algoritma ini mengizinkan proses untuk pindah antrian.

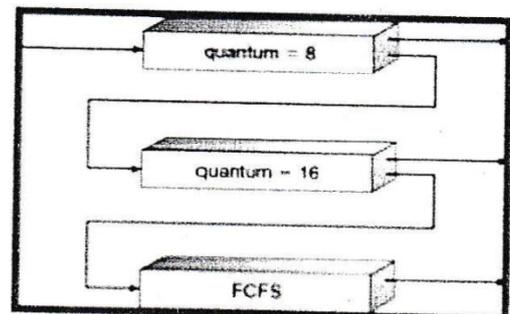
Jika suatu proses menyita CPU terlalu lama, maka proses itu akan dipindahkan ke antrian yang lebih rendah. Ini menguntungkan proses interaksi, karena proses ini hanya memakai waktu CPU yang sedikit. Demikian pula dengan proses menunggu terlalu lama. Proses ini akan dinaikkan tingkatannya.

Biasanya prioritas tertinggi diberikan kepada proses proses dengan CPU burst terkecil, dengan begitu CPU

akan dimanfaatkan penuh dan I/O dapat terus sibuk. Semakin rendah tingkatannya, panjang CPU burst proses juga semakin besar.

Algoritma ini didefinisikan melalui beberapa parameter, antara lain:

1. Jumlah antrian
2. Algoritma penjadwalan tiap antrian
3. Kapan menaikkan proses ke antrian yang lebih tinggi
4. Kapan menurunkan proses ke antrian yang lebih rendah
5. Antrian mana yang akan dimasuki proses yang membutuhkan



Gambar 2. Penjadwalan MLFQ

Dengan pendefinisian seperti tadi membuat algoritma ini sering dipakai. Karena algoritma ini mudah dikonfigurasi ulang supaya cocok dengan sistem. Tapi untuk mengetahui mana penjadwal terbaik, diharuskan mengetahui nilai parameter tersebut. Multilevel feedback queue adalah salah satu algoritma yang berdasar pada algoritma multilevel queue. Perbedaan mendasar yang membedakan multilevel feedback queue biasa adalah terletak pada adanya kemungkinan suatu proses berpindah dari satu antrian ke antrian lainnya, entah dengan prioritas yang lebih rendah ataupun lebih tinggi, misalnya pada contoh berikut.

1. Semua proses yang baru datang akan diletakkan pada antrian 0 (quantum = 8 ms)
2. Jika suatu proses tidak dapat diselesaikan dalam 8 ms, maka proses terdapat akan dihentikan dan dipindahkan ke antrian pertama (quantum = 16 ms)
3. Antrian pertama hanya akan dikerjakan jika tidak ada lagi proses di antrian 0, dan jika suatu proses di antrian pertama 1 tidak selesai

- dalam 16 ms, maka proses tersebut akan dipindahkan ke antrian kedua
4. Antrian kedua akan dikerjakan bila antrian 0 dan 1 kosong, dan akan berjalan dengan algoritma Round Robin.

## ANALISA DAN PEMBAHASAN

### Analisa

Langkah-langkah yang dilakukan dalam analisa menentukan waktu tunggu dan waktu keseluruhan dalam penjadwalan MLFQ yaitu:

1. Mendefinisikan rumus running process untuk ketentuan dalam perhitungan waktu tunggu.
2. Waktu tunggu dijumlahkan dengan waktu layanan untuk mendapatkan waktu keseluruhan.

#### 1. Analisa Waktu Tunggu (WaitingTime)

Setiap proses akan memiliki waktu tunggu yang didapatkan dari waktu layanan proses yang mempengaruhi ditambah dengan waktu kedatangan proses yang dicari.

Ketentuan proses yang mempengaruhi yaitu:

- a. Proses yang dikerjakan waktu ayanannya digantikan oleh proses lain. Maka proses yang menggantikan disebut sebagai proses yang mempengaruhi atau menunggu karena proses lain maka terdapat lebih dari satu proses yang mempengaruhi proses yang berjalan tersebut.
- b. Proses belum dikerjakan pada waktu kedatangannya.

Contoh problema dalam waktu tunggu

1. P0 dikerjakan pertama kali dan tidak ada proses yang mempengaruhi, maka kondisi proses yang mempengaruhi Null (0).
2. P1 dikerjakan dan digantikan P2 dan uga digantikan P3, maka P1 dipenharuhi P2 dan P3.
3. P1 dikerjakan dan digantikan P3, maka P1 dipengaruhi P2 dan P3.
4. P2 belum dikerjakan pada waktu kedatangannya dan pada waktu itu p1 sedang dikerjakan, maka P2 dipengaruhi P1.
5. P2 belum dikerjakan pada waktu kedatangannya dan pada saat itu P1 juga sedang dikerjakan,

sementara sebelumnya P1 juga dipengaruhi P0, maka P2 dipengaruhi P1 dan P0.

6. P1 digantikan P2. Setelah P2 selesai P1 dikerjakan kembali dan pada waktu pengerjaan selanjutnya berjalan P1 digantikan P3, maka P1 dipengaruhi P2 dan P3.

Penyederhanaan problema waktu tunggu. Dalam menghitung waktu tunggu proses diketahui dari proses-proses diketahui dari proses-proses yang mempengaruhinya.

1. Dari problema no. 1  
 $WTP0 = 0$   
 Karena tidak ada yang mempengaruhi P0
2. Dari problema no. 2  
 $WTP1 = BTP2 - ATP1$
3. Dari problema no. 3  
 $WTP1 = (BTP2 + BTP3) - ATP1$
4. Dari problema no. 4  
 $WTP2 = BTP1 - ATP2$
5. Dari problema no. 5  
 $WTP2 = (BTP0 + BTP1) - ATP2$
6. Dari problema no. 6  
 $WTP1 = (BTP2 + BTP3) - ATP1$

#### 2. Running Process (RP)

Running Process mempermudah perhitungan waktu tunggu karena pada MLFQ diketahui process dikerjakan berdasarkan QuantumTime (QT).

Ketentuan pengerjaan Running Process yaitu dengan pembagian Burst Time dibagi Quantum Time sedangkan Sisa Bagi untuk pengerjaan selanjutnya yang kurang dari Quantum Time.

Adapun Running Process sebagai berikut:

1. Hasil bagi → perulangan Quantum Time
2. Sisa bagi → Running process selanjutnya
3. Jika BurstTime < Quantum Time maka Running Process adalah BurstTime process tersebut.

#### 3. Analisa Running Process Algoritma MLFQ

Algoritma MLFQ memiliki berapa antrian. Setiap process akan dikerjakan pada antrian pertama jika BurstTime proses lebih besar dari QuantumTime pertama maka akan dikerjakan pada antrian selanjutnya. Jika terdapat tiga antrian maka ketentuan tersebut berlaku

pada antrian pertama dan kedua. Pada antrian ketiga yang merupakan antrian terakhir berlaku ketentuan Running Process Pembagian.

**PEMBAHASAN**

Penyelesaian proses-proses dalam penjadwalan MLFQ.

1. Process

Tabel 1. Urutan Proses-Proses

Process	ArrivalTime	BurstTime
P1	0	6
P2	2	3
P3	5	4
P4	7	11

2. Level Antrian dan Quantum Time

Tabel 2. Nilai Quantum Time setiap Queue

Queue	QuantumTime
Q0	2
Q1	3
Q2	5

3. Running Process

P1 di Q1 :

RP1Q1 → BTP1 > QT1  
 → 6 > 2  
 → ya, kerjakan sebanyak 2

RP1Q1 = 2  
 Sisa = 4, dipindahkan ke Q2

P1 di Q2 :

RP1Q1 → SBTP1 > QT2  
 → 4 > 3  
 → ya, kerjakan sebanyak 3

RP1Q2 = 3  
 Sisa = 1, dipindahkan ke QT3

P1 di Q3 :

RP1Q3 → SBTP1 > Q3  
 → 1 < 5  
 → ya

RP1Q3(1) = 1

Jadi : RP1Q1 = 2, RP1Q2 = 3, RP1Q3(1) = 1

P2 di Q1 :

RP2Q1 → BTP2 > QT1  
 → 3 > 2  
 → ya, kerjakan sebanyak 2

RP2Q1 = 2  
 Sisa = 1, dipindahkan ke Q2

P2 di Q2:

RP1Q1 → SBTP2 > QT2  
 → 1 > 3  
 → tidak, kerjakan sebanyak 1

RP1Q2 = 1

Jadi : RP2Q1 = 2, RP2Q2 = 2

P3 di Q1:

RP3Q1 → SBTP3 > QT2  
 → 2 > 3  
 → tidak, kerjakan sebanyak 2

RP1Q1 = 2  
 Sisa = 2, dipindahkan ke Q2

P1 di Q2 :

RP3Q1 → SBTP1 > QT2  
 → 2 > 3  
 → tidak, kerjakan sebanyak 2

RP3Q2 = 2

Jadi : RP3Q1 = 2, RP3Q2 = 2

P4 di Q1 :

RP4Q1 → BTP4 > QT1  
 → 11 > 2  
 → ya, kerjakan sebanyak 2

RP4Q1 = 2  
 Sisa = 9, dipindahkan ke Q2

P4 di Q2 :

RP4Q1 → SBTP4 > QT2  
 → 9 > 3  
 → ya, kerjakan sebanyak 3

RP4Q2 = 3  
 Sisa = 6, dipindahkan ke QT3

P1 di Q3 :

RP4Q3 → SBTP4 > Q3  
 → 1 < 5  
 → tidak

Maka

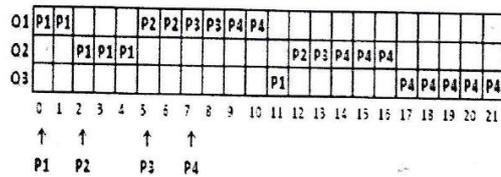
RP4Q3 = SBTP4/QT3  
 = 6 / 5  
 = 1, sisa 1

Hasil 1 perulangan sebanyak QT3  
 → RP4Q3(1) = 5  
 → RP4Q3(2) = 1

Jadi :

RP4Q1 = 2, RP4Q2 = 3, RP4Q3(1) = 5, RP4Q3(2) = 1

4. Diagram Gantt



5. Indeks proses Sesuai ArrivalTime

Tabel 3. Indeks Process

No	Process yang dilayani			
1	P1			
2	P1	P2	P3	P4
3	P1	P4		

6. Waktu Tunggu

$$\begin{aligned}
 WTP1 &= (RP2Q1+RP3Q1+RP4Q1) - ATP1 \\
 &= (2+2+2) - 0 \\
 &= 6 - 0 \\
 &= 6 \text{ ms} \\
 WTP2 &= (RP1Q1+RP1Q2+RP3Q1+RP1Q3_{(1)}) - ATP1 \\
 &= (2+3+2+2+1) - 2 \\
 &= 10 - 2 \\
 &= 8 \text{ ms} \\
 WTP3 &= (RP1Q1+RP1Q2+RP2Q1+RP4Q1+RP1Q3_{(1)}+RP2Q2) - ATP1 \\
 &= (2+3+2+2+1+1) - 5 \\
 &= 11 - 5 \\
 &= 6 \text{ ms} \\
 WTP4 &= (RP1Q1+RP1Q2+RP2Q1+RP3Q1+RP1Q3_{(1)}+RP2Q2)+RP3Q2+ RP1Q3_{(1)}-ATP1 \\
 &= (2+3+2+2+1+1+1) - 7 \\
 &= 12 - 7 \\
 &= 5 \text{ ms}
 \end{aligned}$$

7. Waktu Keseluruhan

$$\begin{aligned}
 TAP1 &= BTP1 + WTP1 \\
 &= 6 + 6 \\
 &= 12 \text{ ms} \\
 TAP2 &= BTP2 + WTP2 \\
 &= 4 + 8 \\
 &= 11 \text{ ms} \\
 TAP3 &= BTP3 + WTP3 \\
 &= 4 + 6 \\
 &= 10 \text{ ms} \\
 TAP4 &= BTP4 + WTP4 \\
 &= 11 + 5 \\
 &= 15 \text{ ms}
 \end{aligned}$$

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan studi dan penelitian yang dilakukan maka dapat disimpulkan yaitu:

1. Algoritma Multi Level Feedback Queue (MLFQ) merupakan algoritma penjadwalan yang baik dalam mengoptimalkan proses-proses yang memiliki prioritas, karena dapat memindahkan proses yang memiliki waktu layanan (BurstTime) yang besar.
2. Running Process yaitu waktu layanan dibagi waktu jatah pengerjaan atau QuantumTime (QT) sehingga hasil bagi digunakan untuk banyak perulangan QT dan sisa bagi untuk banyak pengerjaan berikutnya tapi kurang dari QT.
3. Penerapan Running Proses (RP) memudahkan penentuan waktu tunggu, karena dapat menggunakan RP dalam mengetahui proses-proses yang mempengaruhi dalam diagram gantt penjadwalan MLFQ.

Saran

Beberapa saran yang dapat dipertimbangkan untuk yaitu:

1. Mengimplementasikan analisa algoritma MLFQ ke dalam algoritma pembuatan program pembelajaran dan penyelesaian problema yang menyelesaikan antrian.
2. Analisa algoritma MLFQ ini dapat dikembangkan lagi dalam penentuan waktu respon tiap proses.

DAFTAR PUSTAKA

Azis, M. F. 1994. Belajar Sendiri Pemrograman System Pakar, Jakarta, Elex Media Kaputindo.

Kusumadewi, S. 2003. Artificial Intelegence (Teknik dan Aplikasinya), Yogyakarta, Graha Ilmu.

Martin, J. and O. Steven. 1988. building Expert System, New Jersey, Penerbit Hall.

Turban, E, 1995. Decision Support System and Expert System, USA, Penerbit Hall International Inc.

*Winda Sulastri : Implementasi Algoritma Multilevel Feedback Queue dalam .....*

Turban, E, 2005. Decision Support System and Inteligent System, Yogyakarta, Penerbit Andi Yogyakarta.