**MAKALAH**

**PENJADWALAN PROSES**

**Disusun guna memenuhi tugas mata kuliah**

**Sistem Operasi**

Dosen Pembimbing :

Risda Mairatih, S. Kom



Kelas 3B

Disusun Oleh :

**Kelompok 1**

Muhammad Syafiq Yusuf : 2210010514

Ahmad Syairozi : 2210010066

Muhammad Adam Alghifari : 2210010314

Muhammad Abdul Jabbar : 2210010343

Nur Yoga Andika : 2210010652

Winda Dwi Ningsih : 2210010530

Wahyu Afriliyani : 2210010127

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INGORMASI**

**UNIVERSITAS ISLAM KALIMANTAN MUHAMMAD ARSYAD AL BANJARI**

**TAHUN 2023**

# KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas limpahan rahmatnya penyusun dapat menyelesaikan makalah ini tepat waktu tanpa ada halangan yang berarti dan sesuai dengan harapan.

Sistem Operasi merupakan mata pelajaran yang sangat penting mengingat kita harus mengetahui beberapa komponen komponen yang berada didalam sistem operasi. Melalui mata pelajaran ini, kita diajarkan mengenai penjelasan lebih mendalam terkait materi Sistem Operasi.

Makalah ini bertujuan untuk memberikan pemahaman yang lebih mendalam tentang pentingnya penjelasan mata kuliah terkait Sistem Operasi dalam membentuk jiwa kita yang terbuka terhadap hal-hal baru yang ingin kita pelajari. Kami berharap makalah ini dapat bermanfaat bagi pembaca dalam memahami dan mengapresiasi nilai-nilai kewarganegaraan.

Ucapan terima kasih kami sampaikan kepada Ibu Asisten Dosen Risda Mairatih, S. Kom, sebagai dosen pengampu mata kuliah Sistem Operasi yang telah membantu memberikan arahan dan pemahaman dalam penyusunan makalah ini.

Kami menyadari bahwa dalam penyusunan makalah ini masih banyak kekurangan karena keterbatasan kami. Maka dari itu penyusun sangat mengharapkan kritik dan saran untuk menyempurnakan makalah ini. Semoga apa yang ditulis dapat bermanfaat bagi semua pihak yang membutuhkan.

|  |
| --- |
| Banjarbaru, 16 November 2023 |
|  |
|  |
|  |
| Kelompok 1 |

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc151491362)

[DAFTAR ISI ii](#_Toc151491363)

[BAB 1 PENDAHULUAN 1](#_Toc151491364)

[A. Latar Belakang 1](#_Toc151491365)

[B. Rumusan Masalah 1](#_Toc151491366)

[C. Metode Penulisan 1](#_Toc151491367)

[D. Tujuan Penulisan 2](#_Toc151491368)

[BAB 2 PENJADWALAN PROSES 3](#_Toc151491369)

[A. Pengertian Penjadwalan Proses 3](#_Toc151491370)

[B. Sasaran Penjadwalan 4](#_Toc151491371)

[C. Antrian Penjadwalan Proses 5](#_Toc151491372)

[D. Five State Model Process 6](#_Toc151491373)

[E. Two State Model Process 7](#_Toc151491374)

[F. Perbedaan Two State Process Model dengan Five States Process Model 8](#_Toc151491375)

[G. Tipe-Tipe Penjadwalan 9](#_Toc151491376)

[H. Strategi Penjadwalan 10](#_Toc151491377)

[I. Sakelar Konteks 11](#_Toc151491378)

[BAB 3 PENUTUP 13](#_Toc151491379)

[A. Kesimpulan 13](#_Toc151491380)

[B. Saran 14](#_Toc151491381)

[DAFTAR PUSTAKA 15](#_Toc151491382)

# BAB 1 PENDAHULUAN

## A. Latar Belakang

Penjadwalan proses dalam sistem operasi merupakan aspek krusial untuk meningkatkan kinerja dan responsivitas sistem. Dalam perkembangan teknologi informasi yang pesat, tuntutan terhadap efisiensi sistem semakin meningkat. Oleh karena itu, penelitian mengenai penjadwalan proses menjadi penting untuk memahami dan meningkatkan performa sistem operasi.

## B. Rumusan Masalah

Rumusan masalah mencakup pertanyaan-pertanyaan yang ingin dijawab melalui penulisan. Beberapa contoh rumusan masalah untuk penjadwalan proses sistem operasi mungkin mencakup:

1. Bagaimana algoritma penjadwalan proses dapat memengaruhi kinerja sistem operasi?
2. Apa keuntungan dan kerugian dari berbagai metode penjadwalan proses yang ada?
3. Bagaimana penjadwalan proses dapat dioptimalkan untuk memenuhi kebutuhan spesifik suatu aplikasi atau lingkungan sistem?

## C. Metode Penulisan

Metode penulisan mencakup pendekatan atau strategi yang akan digunakan dalam penyusunan penulisan. Untuk topik penjadwalan proses sistem operasi, metode penulisan mungkin melibatkan:

1. Analisis literatur tentang berbagai algoritma penjadwalan proses.
2. Implementasi dan uji coba algoritma-algoritma tersebut dalam suatu lingkungan simulasi.
3. Perbandingan hasil kinerja dari berbagai metode penjadwalan proses.
4. Studi kasus pada implementasi nyata di berbagai sistem operasi.

## D. Tujuan Penulisan

Tujuan penulisan adalah hasil yang ingin dicapai melalui penulisan tersebut. Beberapa tujuan yang mungkin untuk penulisan tentang penjadwalan proses sistem operasi termasuk:

1. Memberikan pemahaman yang mendalam tentang konsep penjadwalan proses.
2. Menilai kinerja berbagai algoritma penjadwalan proses.
3. Menyajikan rekomendasi atau pedoman untuk memilih algoritma penjadwalan proses yang sesuai dengan kebutuhan tertentu.
4. Menyumbangkan pengetahuan baru atau solusi inovatif dalam bidang penjadwalan proses.

# BAB 2 PENJADWALAN PROSES

## Pengertian Penjadwalan Proses

Penjadwalan proses adalah aktivitas manajer proses yang menangani penghapusan proses yang berjalan dari CPU dan pemilihan proses lain berdasarkan strategi tertentu. Penjadwalan proses adalah bagian penting dari sistem operasi Multiprogramming. Sistem operasi semacam itu memungkinkan lebih dari satu proses untuk dimuat ke dalam memori yang dapat dieksekusi pada suatu waktu dan proses yang dimuat berbagi CPU menggunakan penggandaan waktu.

Adapun kegunaan dari penjadwalan proses ini adalah berguna untuk menjalankan proses yang harus berjalan. Serta, kegunaan lainnya adalah untuk mengetahu kapan dan berapa lama proses itu berjalan.

Dalam penjadwalan proses terdapat dua kategori, kategori itu adalah sebagai berkut.

1. Non-preemptive

Di sini sumber daya tidak dapat diambil dari suatu proses sampai proses tersebut menyelesaikan eksekusi. Peralihan sumber daya terjadi ketika proses yang sedang berjalan berhenti dan berpindah ke keadaan menunggu.

2. Preemptive

Di sini OS mengalokasikan sumber daya ke suatu proses untuk jangka waktu tertentu. Selama alokasi sumber daya, proses berpindah dari status berjalan ke status siap atau dari status menunggu ke status siap. Peralihan ini terjadi karena CPU mungkin memberikan prioritas pada proses lain dan mengganti proses dengan prioritas lebih tinggi dengan proses yang sedang berjalan.

## Sasaran Penjadwalan

Sasaran utama penjadwalan proses adalah optimalisasi kinerja sstem komputer menurut kriteria tertentu. kriteria-kriteria tersebut yang digunakan sebagai sasaran penjadwalan adalah sebagai berikut.

1. Adil (Fairness)

Adil atau Fairness ini adalah Proses-proses diperlakukan sama yaitu mendapat jatah waktu layanan pemroses yang sama dan tidak ada proses yang tidak kebagian layanan pemroses sehingga mengalami startvation. Startvation adalah kondisi bahwa proses tidak pernah berjalan karena tidak dijadwalkan untuk berjalan. Sasaran penjadwalan seharusnya menjamin setiap proses mendapat pelayanan dari pemroses secara adil.

2. Efisiensi (Eficiency)

Efisiensi atau utilisasi pemroses dihitung dengan perbandingan (rasio) waktu sibuk pemroses dengan total waktu operasi sistem komputer secara keseluruhan. Sasaran penjadwalannya adalah menjaga agar pemroses tetap dalam keadaan sibuk sehingga efisiensi sistem komputer mencapai nilai maksimum. Keadaan sibuk berarti pemroses tidak menganggur. Layanan pemroses termasuk waktu yang dihabiskan untuk mengeksekusi program pemakai dan layanan sistem operasi secara efektif, bukan untuk melakukan penjadwalan itu sendiri.

3. Waktu tanggapan (Response time)

waktu tanggapan atau response time terbagi menjadi dua yaitu sebagai berikut :

1. Waktu tanggapan dalam sistem interaktif

Waktu yang dihabiskan dari saat karakter terakhir dari perintah dimasukkan oleh program atau transaksi sampai hasil pertama muncul di perangkat masukan keluaran seperti layar (terminal).Waktu tanggap untuk sistem interaktif biasa disebut terminal responce time.

1. Waktu tanggapan dalam sistem waktu nyata (Real time)

Pada sistem waktu nyata, waktu tanggap didefinisikan sebagai waktu dari saat kemunculan suatu kejadian (internal/eksternal) sampai instruksi pertama rutin layanan terhadap kejadian dieksekusi. Waktu untuk sistem waktu nyata biasa disebut event response time.

Sasaran penjadwalan dari waktu tanggap adalah meminimalkan waktu tanggap sehingga menghasilkan sistem yang responsif.

4. Turn around time

Waktu yang dihabiskan dari saat proses atau job mulai masuk ke sistem sampai proses itu diselesaikan sistem.Waktu yang dimaksud adalah waktu yang dihabiskan proses berada di sistem, diekspresikan sebagai penjumlahan waktu eksekusi (waktu layanan proses/job) dan waktu menunggu dari proses itu, yaitu sebagai berikut.

Turn around time = Waktu eksekusi + Waktu menunggu

Sasaran penjadwalan dari turn around time ini adalah meminimalkan turn arround time.

5. Troughput

Troughput adalah jumlah kerja yang dapat diselesesaikan selama satu selang/unit waktu. Cara untuk mengekspresikan throughput adalah dengan jumlah proses/job pemakai yang dapat dieksekusi dalam satu unit/ interval waktu tertentu. Sasaran penjadwalannya adalah memaksimalkan jumlah job/ proses yang dilayani per satu interval waktu. Lebih tinggi angka througput maka lebih banya kerja yang dilakukan sistem. Kriteria tersebut saling bergantung dan dapat saling bertentangan sehingga tidak dimungkinkan optimasi semua kriteria secara simultan.

## Antrian Penjadwalan Proses

Sistem operasi memelihara semua PCB dalam Antrian Penjadwalan Proses. Sistem operasi memelihara antrian yang terpisah untuk masing-masing status proses dan PCB dari semua proses dalam keadaan eksekusi yang sama ditempatkan dalam antrian yang sama. Ketika status suatu proses diubah, PCB-nya tidak terhubung dari antrian saat ini dan dipindahkan ke antrian status baru.

Sitem operasi mempertahankan antrian penjadwalan proses penting sebagai berikut :

a. Job Queue => Antrian ini menyimpan semua proses dalam sistem.

b. Ready anrian => Antrian ini menyimpan satu set semua proses yang berada di memoriutama, siap dan menunggu untuk dieksekusi. Proses baru selalu dimasukkan dalam antrian ini.

c. Antrian perangkat => Proses yang diblokir karena tidak tersedianya perangkat I / O merupakan antrian ini.

Sistem Operasi dapat menggunakan kebijakan yang berbeda untuk mengelola setiap antrian (FIFO, Round Robin, Priority, dll.). Penjadwal sistem operasi menentukan cara memindahkan proses antara antrian siap dan jalankan yang hanya dapat memiliki satu entri per inti prosesor pada sistem; pada diagram di atas, telah digabungkan dengan CPU.

## D. Five State Model Process

Ide dari model 5-state model adalah, bahwa process dalam suatu waktu dapat berada dalam kondisi new, ready, running, waiting, exit/terminated.

Secara garis besar Model ini memiliki 5 tahapan yaitu :

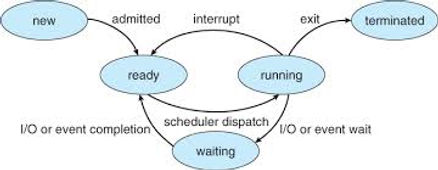
1. New adalah Proses sedang dikerjakan/dibuat.

2. Running adalah Instruksi sednag dikerjakan.

3. Waiting adalah Proses sedang menunggu sejumlah kejadian untuk terjadi (seperti sebuah penyelesaian I/O atau penerimaan sebuah tanda/signal).

4. Ready adalah Proses sedang menunggu untuk ditugaskan pada sebuah prosesor.

5.Terminated/Exit adalah Proses telah selsesai melaksanakan tugasnya/mengeksekusi.



Penjelasan :

1. Semua proses dimulai ketika sebuah proses dibuat yaitu dalam NEW

2. kemudian proses yang sudah dibuat itu akan ADMIT atau di ijinkan untuk masuk maka statusnya menjadi READY.

3. Setelah itu untuk menuju ke proses RUNNING setelah di kirim atau DISPATCH.

4. Di dalam proses running, setiap proses ini akan dijalankan, disini ada 3 kemungkinan yang terjadi yaitu :

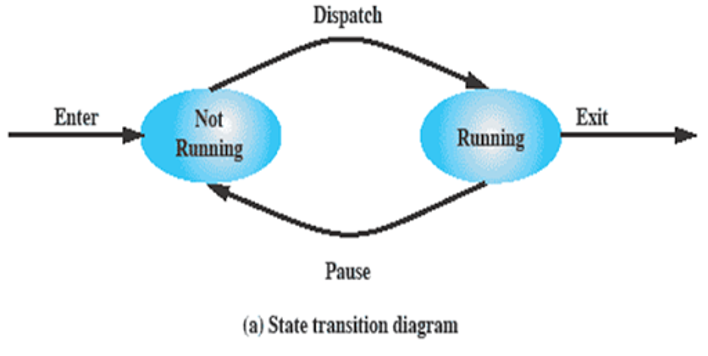
a. Timeout : jika proses tersebut mengalami timeout, maka akan dikembalikan lagi ke status ready untuk di running lagi.

b. Event wait : jika proses itu dalam keadaan event wait, maka proses itu akan masuk ke dalam status Blocked, dimana jika sudah tidak ada proses yang menunggu untuk dijalankan lagi maka, proses dalam status Blocked ini akan kembali ke status ready dan siap untuk di running lagi.

c.Release : Jika suatu proses sudah di release, maka proses itu sudah selesai dan masuk ke status EXIT.

## E. Two State Model Process

Ide dari model 2-state model adalah, bahwa process dalam suatu waktu dapat sedang dieksekusi oleh processor (Running), dan tidak ridak dieksekusi (Not Running).



Proses dapat berada pada salah satu dari dua status :

1. Berjalan (running) berarti program sedang melaksanakan perintah dan sedang dieksekusi serta proses exit – Not-running berarti program tidak dalam proses eksekusi melainkan persiapan akan dieksekusi
2. Not-running berarti program tidak dalam proses eksekusi melainkan persiapan akan dieksekusi

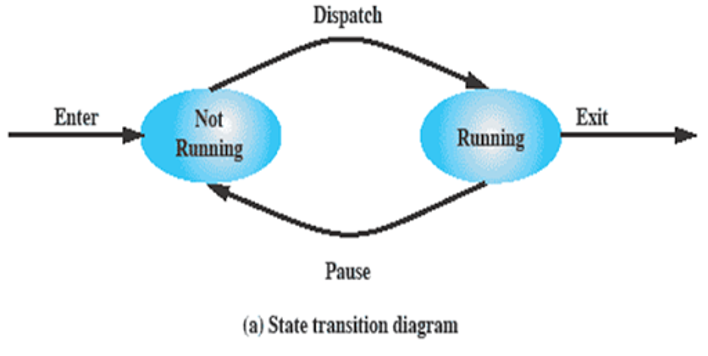
## F. Perbedaan Two State Process Model dengan Five States Process Model

1. Jumlah proses dalam two state ada dua dan five state ada 5

2. dalam five state proses berjalan lebih detail

3. five state membutuhkan waktu yang lebih lama dibandingkan dengan two state

4. five state digunakan untuk hal yang kompleksitasnya tinggi



Proses dapat berada pada salah satu dari dua status :

1. Berjalan (running) berarti program sedang melaksanakan perintah dan sedang dieksekusi serta proses exit
2. Not-running berarti program tidak dalam proses eksekusi melainkan persiapan akan dieksekusi

Perbedaan Antara Two State Process Model dengan Five States Process Model :

1. Jumlah proses dalam two state ada dua dan five state ada 5
2. dalam five state proses berjalan lebih detail
3. five state membutuhkan waktu yang lebih lama dibandingkan dengan two state
4. five state digunakan untuk hal yang kompleksitasnya tinggi

G. Tipe-Tipe Penjadwalan  
Dapat terdapat 3 tipe penjadwal berada secara bersama-sama pada sistem operasi yang kompleks, yaitu :

1. Penjadwal jangka pendek (*short-term scheduller*)

Penjadwalan jangka pendek bertugas menjadwalkan alokasi pemroses di antara proses-proses Ready yang berada di memori utama. sasaran utama penjadwal jangka pendek adalah memaksimumkan kinerja sistem untuk memenuhi satu kumpulan kriteria yang diharapkan. Penjadwal ini dijalankan setiap terjadi pengalihan proses untuk memilih proses berikutnya yang harus dijalankan.

1. Penjadwal jangka menengah (*medium-term scheduller*)

Setelah eksekusi selama suatu waktu, proses mungkin ditunda karena permintaan layanan masukan/keluaran atau memanggil suatu system call. Proses-proses yang tertunda tidak dapat membuat suatu kemajuan untuk menuju selesai sampai ondisi yang menyebabkannya hilang. Agar ruang memori dapat bermanfaat maka proses dipindah dari memori utama ke memori sekunder sehingga tersedia ruang yang lebih besar untuk proses yang lain. Kapasitas memori utama terbatas untuk sejumlah proses yang aktif. Aktivitas pemindahan proses yang tertunda dari memori utama ke memori sekunder disebut *swapping*. Penjadwal jangka menengah bertugas menangani proses swapping . Proses yang mempunyai kepentingan kecil saat itu adalah proses yang tertunda. Tetapi begitu kondii yang membuat proses tertunda hilang dan proses dimasukkan kembali ke memori utama dan Ready. Penjadwal jangka menengah mengendalikan transisi dari suspended ke ready (dari state suspend ke Ready dari proses yang mengalami swapping).

1. Penjadwal jangka panjang (*long-term scheduller*)

Penjadwal jangka panjang bekerja terhadap antrian*batch* dan memilih batch berikutnya yang harus dieksekusi sistem. Batch biasanya berupa proses-proses dengan penggunaan sumber daya yang intensif (yaitu waktu pemroses, memori, perangkat masukan/keluaran), program ini mempunyai prioritas yang rendah, dan biasa digunakan sebagai pengisi (agar pemroses sibu) selama periode aktivitas proses-proses interaktif rendah. Sasaran utama penjadwal jangka panjang adalah memberi keseimbangan proses-proses campuran. Tipe-tipe penjadwal dapat dikaitkan dengan state proses. Kaitan antara tipe-tipe penjadwalan dengan state proses digambarkan pada gambar berikut :

H. Strategi Penjadwalan  
Terdapat 2 strategi penjadwalan, yaitu :

1. Penjadwalan *nonpreemptive* (*run-to-completion*). Begitu proses diberi jatah layanan pemroses aka pemroses tidak dapat diambil alih oleh proses lain sampai proses itu selesai. *Non-preemptive* juga disebut run-to-completion karena proses yang telah dijadwalkan akan dijalankan sampai selesainya atau proses tersebut meminta layanan masukan/keluaran.
2. Penjadwalan *preemptive.*Saat proses diberi jatah layanan pemroses maka pemroses dapat diambil alih proses lain yang mempunyai prioritas lebih tinggi berdasarkan kriteria sistem itu. Pada penjadwalan preemptive, proses dapat disela oleh proses lain sebelumnya selesainya dan harus dilanjutkan menunggu jatah waktu layanan pemroses tiba kembali pada proses itu. Proses yang disela berubah menjadi state *Ready*.

Penjadwalan *preemptive* berguna pada sistem yakni proses-proses yang perlu mendapat perhatian/ tanggapan pemroses secara cepat. Misalnya :

* 1. Pada sistem-sistem waktu nyata, kehilangan interupsi (yaitu interupsi tidak segera dilayani) dapat berakibat fatal
  2. Pada sistem-sistem interatif timesharing, penjadwalan preemptive penting agar dapat menjamin waktu tanggap yang memadai.

Peralihan proses (yaitu layanan pemroses dari satu proses beralih ke proses lain) memerlukan *overhead* (karena banya tabel yang dikelola). Agar penjadwalan *preemptive* menjadi efektif, banyak proses harus berada di memori utama sehingga proses-proses tersebut dapat segera Running begitu diperlukan. Menyimpan banyak proses yang tidak *Running* di memori utama merupakan suatu overhead tersendiri.

## Sakelar Konteks

Dalam komputasi , sakelar konteks adalah proses menyimpan status proses atau utas , sehingga dapat dipulihkan dan melanjutkan eksekusi di lain waktu. Hal ini memungkinkan beberapa proses untuk berbagi satu unit pemrosesan pusat (CPU), dan merupakan fitur penting dari sistem operasi multitasking .

Arti yang tepat dari frasa "beralih konteks" bervariasi. Dalam konteks multitasking, ini mengacu pada proses penyimpanan status sistem untuk satu tugas, sehingga tugas dapat dijeda dan tugas lain dilanjutkan. Peralihan konteks juga dapat terjadi sebagai akibat dari interupsi , seperti saat tugas perlu mengakses penyimpanan disk , membebaskan waktu CPU untuk tugas lain. Beberapa sistem operasi juga memerlukan sakelar konteks untuk berpindah antara mode pengguna dan tugas mode kernel . Proses pengalihan konteks dapat berdampak negatif pada kinerja sistem.

Sakelar konteks biasanya intensif secara komputasi, dan sebagian besar desain sistem operasi adalah untuk mengoptimalkan penggunaan sakelar konteks. Berpindah dari satu proses ke proses lainnya membutuhkan waktu tertentu untuk melakukan administrasi – menyimpan dan memuat register dan peta memori, memperbarui berbagai tabel dan daftar, dll. Apa yang sebenarnya terlibat dalam sakelar konteks tergantung pada arsitektur, sistem operasi, dan jumlah sumber daya yang dibagikan (utas yang termasuk dalam proses yang sama berbagi banyak sumber daya dibandingkan dengan proses non-kerja sama yang tidak terkait). Misalnya, di kernel Linux , pengalihan konteks melibatkan register switching, penunjuk tumpukan (biasanya register penunjuk tumpukan ), penghitung program , pembilasanterjemahan lookaside buffer (TLB) dan memuat tabel halaman dari proses berikutnya untuk dijalankan (kecuali proses lama berbagi memori dengan yang baru). Selanjutnya, pengalihan konteks analog terjadi di antara utas pengguna , terutama utas hijau , dan seringkali sangat ringan, menyimpan dan memulihkan konteks minimal. Dalam kasus ekstrim, seperti beralih antara goroutine di Go , sebuah context switch setara dengan hasil coroutine , yang hanya sedikit lebih mahal daripada panggilan subrutin .

Paling umum, dalam beberapa skema penjadwalan , satu proses harus dimatikan dari CPU sehingga proses lain dapat berjalan. Sakelar konteks ini dapat dipicu oleh proses yang membuatnya tidak dapat dijalankan, seperti menunggu operasi I/O atau sinkronisasi selesai. Pada sistem multitasking pre-emptive , penjadwal juga dapat mengganti proses yang masih dapat dijalankan. Untuk mencegah proses lain dari kekurangan waktu CPU, penjadwal pre-emptive sering mengonfigurasi interupsi pengatur waktu untuk diaktifkan ketika suatu proses melebihi irisan waktunya . Interupsi ini memastikan bahwa penjadwal akan mendapatkan kontrol untuk melakukan sakelar konteks.

Arsitektur modern didorong oleh interupsi . Ini berarti bahwa jika CPU meminta data dari disk, misalnya, tidak perlu sibuk-menunggu sampai pembacaan selesai; itu dapat mengeluarkan permintaan (ke perangkat I/O) dan melanjutkan dengan beberapa tugas lainnya. Ketika pembacaan selesai, CPU dapat diinterupsi (oleh perangkat keras dalam hal ini, yang mengirimkan permintaan interupsi ke PIC ) dan disajikan dengan pembacaan. Untuk interupsi, sebuah program yang disebut penangan interupsi diinstal, dan penangan interupsi yang menangani interupsi dari disk.

# BAB 3 PENUTUP

## A. Kesimpulan

Penjadwalan proses dalam sistem operasi adalah elemen kunci yang memengaruhi kinerja dan responsivitas sistem secara keseluruhan. Dalam menjalankan tugasnya, beberapa kesimpulan dapat diambil:

1. Diversitas Algoritma:

Dalam menyeleksi algoritma penjadwalan, penting untuk mempertimbangkan karakteristik spesifik dari aplikasi atau lingkungan sistem. Setiap algoritma memiliki kelebihan dan kekurangan, dan pilihan yang tepat sangat tergantung pada kebutuhan sistem.

1. Optimalitas dan Keadilan:

Algoritma penjadwalan yang optimal untuk satu keadaan mungkin tidak optimal untuk yang lain. Oleh karena itu, penyeimbangan antara optimalitas dan keadilan perlu diperhatikan untuk memastikan setiap proses mendapatkan akses yang adil dan efisien terhadap sumber daya sistem.

1. Responsivitas Pengguna:

Algoritma penjadwalan harus dirancang untuk memberikan responsivitas yang tinggi terhadap input pengguna. Penundaan dan waktu tanggapan yang singkat adalah faktor kunci untuk meningkatkan pengalaman pengguna.

1. Evaluasi Kinerja:

Evaluasi kinerja sistem harus dilakukan secara terus-menerus untuk memastikan bahwa algoritma penjadwalan yang digunakan masih relevan dan efektif. Dalam kasus perubahan beban kerja atau karakteristik sistem, penyesuaian mungkin diperlukan.

## B. Saran

1. Sebaiknya lebih dijelaskan dengan detail perihal materi penjadwalan proses dan dan dijabarkan lebih

2. Pada bagian penjelasan seharunya dijelaskan secara singkat, jelas dan padat agar pembaca dapat memahami isi dari makalah ini

3. Terlalu banyak penjelasan yang kurang tepat dan seharusnya bisa lebih diperbaiki lagi agar tidak melesset dari topik

# DAFTAR PUSTAKA

2021. Aristy. Penjadwalan Proses. STMIK Dharma Wacana. <https://aristi.dharmawacana.ac.id>

2015. Rudy. Operating System. Rudy’s Site.

http://sistemoperasi.weebly.com/process/operating-system

2023. Samwer. Five State Model Process dan Two State Model Process. Vaberal.

<https://emaemolcemol.wixsite.com/emafadlila/single-post/2015/09/08/five-state-model-process-dan-two-state-model-process>