

Diskusi.4

Lakukan: Kirim balasan: 1

Jatuh tempo: Minggu, 2 November 2025, 23:59

menampilkan balasan dalam bentuk bertingkat

Setelan ▾

Diskusi.4

Rabu, 28 Mei 2025, 10:28

Jawab dan diskusikanlah soal dibawah ini:

Jelaskan permasalahan critical section!

Tautan permanen Balas

Re: Diskusi.4

oleh [SYAHRİ WAHYUNI ABDUL RAZAK 056995024](#) - Senin, 27 Oktober 2025, 05:37

Izin menjawab hasil Diskusi 4 🙏

Permasalahan critical section muncul ketika beberapa proses dalam sistem komputer berusaha mengakses dan menggunakan sumber daya yang sama pada waktu bersamaan, misalnya variabel, file, atau perangkat keras tertentu. Karena akses dilakukan secara bersamaan, hal ini bisa menyebabkan konflik atau kesalahan data, terutama jika satu proses sedang mengubah data sementara proses lain juga mencoba mengaksesnya.

Untuk menghindari hal tersebut, sistem operasi harus mengatur agar hanya satu proses yang boleh masuk ke bagian critical section pada satu waktu. Inilah yang disebut dengan konsep mutual exclusion.

Selain itu, penyelesaian masalah critical section juga harus memenuhi dua syarat penting lainnya:

1. Progress (Kemajuan): Jika tidak ada proses di critical section, maka proses lain yang ingin masuk tidak boleh menunggu terlalu lama.

2. Bounded Waiting (Waktu Tunggu Terbatas): Setiap proses yang menunggu giliran masuk ke critical section harus mendapat kesempatan dalam batas waktu yang wajar.

Sebagai contoh, jika dua program mencoba memperbarui data stok barang di sistem penjualan secara bersamaan tanpa pengaturan yang tepat, hasil akhirnya bisa keliru — misalnya jumlah stok menjadi salah. Karena itu, diperlukan mekanisme seperti lock, mutex, atau semaphore untuk memastikan proses berjalan bergantian dan data tetap konsisten.

Singkatnya, permasalahan critical section adalah tantangan dalam mengatur proses agar tidak saling mengganggu saat mengakses sumber daya bersama, sehingga hasil tetap akurat dan sistem berjalan stabil.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Hide sidebar

Re: Diskusi.4

oleh [052869663 ANTENG FARIZATUL AWALLIA](#) - Selasa, 28 Oktober 2025, 20:53

Nama : Anteng Farizatul Awallia

NIM : 052869663

Permasalahan critical section muncul dalam lingkungan dimana beberapa proses atau thread berjalan secara bersamaan(konkuren) dan mereka harus mengakses sumber daya bersama (shered resources) yang sama, seperti variabel global,berkas, atau hardware.

Permasalahan pada Anatomi proses. Untuk memecahkan masalah ini, setiap proses yang ingin mengakses sumber daya bersama harus mengikuti struktur protokol tertentu,yang terdiri dari emapt bagian :

- Entry section : kode digunakan untuk meminta izin masuk ke critical section(misal ,mengunci sumber daya)
- Critical section : Bagian kode aktual dimana sumber daya bersama diakses dan dimodifikasi.
- Exit section : Kode yang digunakan untuk melepaskan izin setelah selesai (misal,membuka kunci sumber daya) memungkinkan proses lain masuk.
- Remainder section : Kode lainnya untuk tidak melibatkan akses ke sumber daya bersama.

Referensi :

<https://share.google/tLw4ZDXqdmEV3bsZC>

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)


Re: Diskusi.4

oleh [INDRI SUHERDI 056356352](#) - Senin, 27 Oktober 2025, 07:53

Izin menjawab hasil diskusi dan untuk jawabannya sebagai berikut :

[Diskusi 4.pdf](#)

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)


Re: Diskusi.4

oleh [RIVALINO SHEVA ANDREAN 052541771](#) - Senin, 27 Oktober 2025, 11:02

Selamat Pagi Bapak/ Ibu Dosen,

Berikut diskusi untuk topik Permasalahan Critical Section.

Apa itu Critical Section?

Critical section adalah **bagian dari program** (biasanya beberapa baris kode) yang mengakses **sumber daya bersama** seperti:

- Variabel global
- File
- Memori bersama

Masalah muncul ketika **dua atau lebih proses/thread** mencoba mengakses bagian ini **secara bersamaan**. Jika tidak ada pengaturan, hasilnya bisa salah karena terjadi **race condition** (proses saling berebut mengubah data).

Mengapa Ini Jadi Masalah?

Bayangkan ada dua orang yang menulis di papan tulis yang sama **pada waktu bersamaan**:

- Orang A menulis angka 5, lalu menambah 1 → seharusnya jadi 6.
- Orang B menulis angka 5, lalu menambah 1 → seharusnya jadi 6 juga. Kalau dilakukan bersamaan tanpa aturan, hasil akhirnya bisa tetap 5 atau angka acak, karena operasi saling tumpang tindih.

Permasalahan Utama dalam Critical Section

1. Mutual Exclusion

Hanya **satu proses** yang boleh masuk ke critical section pada satu waktu.

2. Race Condition

Jika dua proses mengakses data bersama tanpa pengaturan, hasilnya tidak terduga.

3. Deadlock

Dua proses saling menunggu kunci (lock) yang tidak pernah dilepas.

4. Starvation

Ada proses yang tidak pernah mendapat giliran karena selalu didahului proses lain.

5. Fairness

Semua proses harus mendapat kesempatan yang adil untuk mengakses critical section.

Solusi Umum

- **Locking:** Menggunakan **mutex** atau **semaphore** agar hanya satu proses yang bisa masuk.
- **Atomic Operation:** Operasi dilakukan secara utuh tanpa bisa diinterupsi.
- **Monitor:** Struktur yang mengatur akses ke critical section.
- **Algoritma Mutual Exclusion:** Contoh: Peterson's Algorithm, Bakery Algorithm.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh [ZIKRA RACHMAT SAPUTRA 052513267](#) - Senin, 27 Oktober 2025, 15:45

Izin menjawab diskusi 4 ini bapak/ibu dosen

Menurut saya, Permasalahan critical section ini ada karena beberapa proses yang berjalan bersamaan dan mengakses data atau resource yang secara bersamaan. Critical section itu adalah merupakan bagian dari proses untuk mengakses data seperti mengubah file, variabel umum, mengubah tabel nah ini disebut critical section. Makanya untuk mengatasi permasalahan critical section perlu dilakukan sinkronisasasi agar, yang boleh masuk ke critical section dalam satu waktu yaitu hanya satu proses, lalu jika ada yang mau proses lain mau masuk harus menunggu giliran agar bisa dijalankan. Kesimpulannya jika tidak ingin mengalami permasalahan critical section maka pastikan hanya satu proses saja yang berjalan dan tidak ada proses yang jalan secara bersamaan untuk mengalami permasalahan critical section, karena akan menyebabkan data menjadi tidak konsisten. Oleh karena itu, sistem operasi harus menyediakan mekanisme sinkronisasi agar hanya satu proses yang bisa berada di critical section pada satu waktu, sedangkan proses lain harus menunggu sampai giliran mereka tiba.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh [A. GANI AJI SAPUTRA 056165787](#) - Selasa, 28 Oktober 2025, 13:28

Izin Menjawab Diskusi 4

Permasalahan critical section terjadi ketika beberapa proses atau thread dalam sistem paralel bersaing untuk mengakses sumber daya bersama, seperti variabel global, file, atau memori, yang hanya boleh diakses oleh satu

proses pada satu waktu. Jika tidak diatur dengan benar, hal ini dapat menyebabkan kondisi yang tidak diinginkan seperti race condition, deadlock, dan starvation.

Pengertian Critical Section

Critical section adalah bagian dari kode program yang mengakses atau memodifikasi sumber daya bersama. Dalam critical section, hanya satu proses yang boleh dieksekusi pada satu waktu untuk mencegah konflik akses.

Permasalahan yang Muncul :

- Race Condition: Terjadi ketika dua atau lebih proses mengakses sumber daya bersama secara bersamaan, sehingga hasil akhir program bergantung pada urutan eksekusi yang tidak dapat diprediksi, menyebabkan data menjadi tidak konsisten.
- Deadlock: Terjadi ketika dua atau lebih proses saling menunggu untuk mengakses sumber daya yang sedang digunakan oleh proses lain, sehingga program terjebak dan tidak dapat berjalan.
- Starvation: Terjadi ketika satu atau lebih proses tidak pernah mendapatkan akses ke critical section karena selalu diabaikan oleh proses lain, menyebabkan kinerja sistem menjadi buruk.

Syarat Solusi Critical Section :

Untuk mengatasi permasalahan critical section, solusi harus memenuhi tiga syarat utama:

- Mutual Exclusion: Hanya satu proses yang boleh berada di critical section pada satu waktu.
- Progress: Jika tidak ada proses yang sedang berada di critical section dan ada proses lain yang ingin masuk, maka pemilihan proses berikutnya tidak boleh ditunda tanpa batas.
- Bounded Waiting: Ada batasan jumlah proses yang diizinkan masuk ke critical section setelah permintaan masuk, sehingga tidak ada proses yang menunggu selamanya.

Dengan memahami permasalahan critical section dan menerapkan solusi yang sesuai, sistem paralel dapat berjalan dengan aman dan konsisten.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh [RAFAEL GUEVARA TARIGAN 053739643](#) - Rabu, 29 Oktober 2025, 08:06

Bayangan aja ada 2 orang mau ngerjain file Excel yang sama di waktu yang sama satu netik di sheet1, satu lagi netik di sheet1 juga. tidak pakai aturan. Hasilnya? Berantakan, datanya tabrakan, kadang corrupt. Nah, itu inti dari critical section problem: dua proses (atau lebih) menyentuh data yang sama di saat bersamaan tanpa koordinasi.

<https://repository.unimal.ac.id/4073/1/BUKU%20sistem%20operasi%20PDF.pdf>

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh [BAYU DWI SAPUTRA 054794681](#) - Rabu, 29 Oktober 2025, 09:50

Critical Section adalah segmen kode dalam program di mana proses atau thread mengakses dan memanipulasi sumber daya yang dibagi (*shared resource*), seperti variabel global, struktur data, file, atau perangkat keras.

Inti Permasalahan: Masalah utama dalam critical section adalah menjamin bahwa hanya satu proses atau thread yang dapat mengeksekusi critical section-nya pada suatu waktu. Jika dua proses atau lebih masuk ke critical section mereka secara bersamaan (*concurrently*), dapat terjadi race condition yang mengakibatkan ketidakkonsistensi data (*data inconsistency*).

Ilustrasi Contoh:

Misalkan dua thread, T1 dan T2, ingin menambah nilai variabel global **saldo** yang awalnya 100.

- Thread T1: Membaca **saldo = 100**, menambah 50, sehingga **saldo = 150**.
- Thread T2: Membaca **saldo = 100**, menambah 25, sehingga **saldo = 125**.

Jika eksekusinya bersamaan (interleaved), bisa terjadi:

1. T1 baca **saldo = 100**.
2. T2 baca **saldo = 100**.
3. T1 tulis **saldo = 100 + 50 = 150**.
4. T2 tulis **saldo = 100 + 25 = 125** (nilai tambah 50 dari T1 hilang).

Daerah kode dimana **saldo** dibaca, diubah, dan ditulis adalah Critical Section.

Tiga Kondisi yang Harus Dipenuhi untuk Menyelesaikan Masalah Critical Section

Untuk memastikan bahwa *race condition* tidak terjadi, solusi untuk masalah critical section harus memenuhi tiga kondisi berikut:

1. Mutual Exclusion (Eksklusi Mutual):

Jika suatu proses sedang mengeksekusi critical section-nya, maka proses lain tidak boleh masuk ke dalam critical section mereka. Hanya satu proses yang diizinkan masuk.

2. Progress (Kemajuan):

Jika tidak ada proses yang berada di dalam critical section dan ada beberapa proses yang ingin memasukinya, maka pemilihan proses mana yang akan masuk ke critical section berikutnya tidak boleh ditunda tanpa batas. Pemilihan ini harus dilakukan oleh proses-proses yang sedang bersaing (bukan oleh proses di luar mereka) dan harus diputuskan dalam waktu terbatas.

3. Bounded Waiting (Penantian Terbatas):

Terdapat batas berapa kali proses lain diizinkan masuk ke critical section setelah suatu proses menyatakan keinginannya untuk masuk. Dengan kata lain, proses yang ingin masuk ke critical section tidak boleh ditunggu secara tak terbatas. Ini mencegah *starvation* (proses tidak pernah mendapat akses).

Dampak jika Masalah Critical Section Tidak Ditangani

Kegagalan mengelola critical section akan mengakibatkan:

1. Race Condition: Output sistem menjadi bergantung pada urutan eksekusi proses yang tidak ditentukan, sehingga hasilnya tidak dapat diprediksi dan seringkali salah.
2. Data Corruption: Data yang dibagi (*shared data*) menjadi rusak atau tidak konsisten karena manipulasi yang bersamaan.
3. System Instability: Program atau sistem dapat berperilaku tidak normal, mengalami *deadlock*, atau menghasilkan output yang merugikan, terutama dalam sistem yang membutuhkan presisi tinggi seperti sistem perbankan, kontrol lalu lintas udara, atau sistem embedded.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh [AHMAD HASBY MAULANA 057088008](#) - Kamis, 30 Oktober 2025, 10:26

Izin menanggapi diskusi 4 terkait permasalahan critical section

Critical Section itu bagian dari proses yang mana memungkinkan untuk mengubah variabel umum, mengubah file, tabel dan lain sebagainya[1]. Nah permasalahan critical section itu terjadi ketika beberapa proses menjalankan critical section pada suatu area yang sama secara bersamaan, masalah ini juga bisa disebut race condition. Apasih akibatnya permasalahan ini? Akibatnya bisa berupa ketidak konsistensi data, sederhananya datanya jadi tidak sinkron. Terus solusinya gimana? Ada 3 syarat yang harus dipenuhi untuk solusi permasalahan critical section ini, berikut 3 syaratnya[1]:

- Mutual Exclusion, hanya boleh ada 1 proses yang terdapat pada critical section dalam satu waktu.

- Progress, tidak boleh ada antrian yang macet tanpa alasan
- Bounded Waiting, semua proses yang antre harus dijalankan

Ada juga beberapa metode yang dapat dilakukan untuk mengatasi masalah ini, diantaranya[1]:

- Solusi Peterson
- Penggunaan Kunci (Lock)
- Sinkronisasi Perangkat Keras
- Solusi dengan intruksi TestAndSet

Hide sidebars

Contohnya begini, semisal ada 1 tiket bioskop yang tersisa, dan ada 2 orang yang ingin membeli tiket, 1 di loket A dan 1 di loket B. Loket A melihat masih ada sisa 1 tiket, dan disaat bersamaan loket B juga melihat masih ada sisa 1 tiket. Nah apa yang terjadi? secara logika pasti ada 1 orang yang dapat tiket dan 1 nya tidak dapat tiket, tapi ini bukan tentang itu melainkan siapa yang mendapatkan tiket terlebih dahulu. Petugas loket A memberikan tiket ke pelanggan A dan secara bersamaan petugas di loket B juga memberikan tiket yang sama ke pelanggan B, apa yang terjadi? yang terjadi akan ada 2 tiket yang sama, padahal hanya tinggal 1 kursi saja. Inilah yang dimaksud dari permasalahan critical section.

Demikian tanggapan dari saya. Mohon koreksi jika terdapat kekurangan. Terima kasih.

Referensi:

[1] S. Rinusantoro, Sistem Operasi (MSIM4201), Edisi 2. Tangerang Selatan: Universitas Terbuka, 2023.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh [UMARETA RIZKY SHABRINA 056505332](#) - Kamis, 30 Oktober 2025, 12:58

Salam Sejahtera Bapak/Ibu Tutor. Izin menjawab soal diskusi sesi 4 ini.

Permasalahan critical section terjadi ketika dua atau lebih proses mencoba mengakses sumber daya atau data yang sama secara bersamaan. Bagian program yang mengakses data bersama itulah yang disebut *critical section*. Masalah muncul karena jika dua proses mengubah data secara bersamaan tanpa pengaturan, maka hasil akhirnya bisa menjadi tidak benar atau bahkan menyebabkan sistem error.

Untuk mencegah hal itu, diperlukan beberapa kondisi penting. Pertama adalah mutual exclusion, yaitu hanya satu proses yang boleh berada di *critical section* pada satu waktu. Jika ada satu proses yang sedang menggunakan sumber daya bersama, maka proses lain harus menunggu sampai yang pertama selesai. Kedua adalah progress, artinya proses yang tidak sedang berada di *critical section* tidak boleh menghalangi proses lain untuk masuk ke dalamnya; keputusan siapa yang masuk berikutnya harus ditentukan secara adil. Ketiga adalah bounded waiting, yaitu setiap proses yang menunggu harus diberi jaminan akan mendapat giliran dalam waktu terbatas, sehingga tidak ada proses yang menunggu selamanya.

Untuk mengatasi permasalahan tersebut, ada beberapa solusi yang bisa digunakan.

1. Solusi Peterson, yang merupakan algoritma perangkat lunak untuk dua proses. Dalam solusi ini, setiap proses memiliki *flag* dan *turn* yang digunakan untuk menentukan siapa yang boleh masuk ke *critical section*. Jika satu proses ingin masuk, dia akan memberi tahu melalui *flag* bahwa dirinya siap, lalu menunggu sampai *turn*-nya tiba. Dengan cara ini, tidak ada dua proses yang masuk ke *critical section* bersamaan.

2. Solusi Kunci (lock), yaitu mekanisme untuk mengunci bagian *critical section* saat sedang digunakan oleh satu proses. Proses lain harus menunggu sampai kunci dilepaskan. Mekanisme kunci ini umum digunakan dalam pemrograman modern karena mudah diterapkan.
3. Sinkronisasi perangkat keras (hardware synchronization) dapat dilakukan dengan cara menonaktifkan interupsi sementara agar tidak ada proses lain yang mengganggu proses yang sedang berjalan di *critical section*.
4. Instruksi TestAndSet, yaitu perintah khusus pada perangkat keras yang digunakan untuk membuat *lock*. Instruksi ini bekerja sangat cepat dan atomik (tidak bisa dipecah), sehingga bisa menjamin hanya satu proses yang memegang kunci dalam satu waktu. Dengan cara ini, kondisi *mutual exclusion* dapat dipertahankan dan masalah *critical section* bisa dihindari. Namun, penggunaan TestAndSet juga harus hati-hati karena bisa menyebabkan penundaan jika banyak proses berebut masuk secara bersamaan.

Dari semua penjelasan tersebut, saya bisa menyimpulkan bahwa *critical section* adalah masalah penting dalam sistem operasi yang harus diatasi dengan pengendalian akses yang baik. Konsep seperti *mutual exclusion*, *progress*, dan *bounded waiting* digunakan agar proses-proses dalam sistem bisa berjalan secara adil dan teratur. Solusi seperti Peterson, penggunaan kunci, sinkronisasi perangkat keras, dan TestAndSet merupakan cara yang efektif untuk menjaga agar sistem tetap stabil dan tidak terjadi konflik antar proses.

Referensi

Silalahi, R. (2022). *Konsep Dasar Critical Section dalam Sistem Operasi*. Universitas Sumatera Utara.
<https://repository.usu.ac.id>

Rosyidah, N. (2021). *Mutual Exclusion, Progress, dan Bounded Waiting pada Sistem Operasi*. Universitas Negeri Yogyakarta. <https://eprints.uny.ac.id>

Yuliana, D. (2020). *Solusi Peterson untuk Permasalahan Critical Section*. Universitas Gadjah Mada.
<https://repository.ugm.ac.id>

Herlambang, A. (2021). *Sinkronisasi Proses dan Penggunaan Lock dalam Sistem Operasi*. Universitas Bina Sarana Informatika. <https://repository.bsi.ac.id>

Kurniawan, T. (2023). *Penerapan Instruksi TestAndSet dalam Pengendalian Akses Proses*. Universitas Diponegoro.
<https://eprints.undip.ac.id>

Kementerian Komunikasi dan Informatika RI. (2022). *Pengantar Sistem Operasi dan Sinkronisasi Proses Komputer*.
<https://digitalent.kominfo.go.id>

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.4

oleh [DIMAS BIMA S 053790635](#) - Kamis, 30 Oktober 2025, 19:49

Assalamualaikum Wr. Wb.

Permasalahan critical section terjadi ketika dua atau lebih proses mencoba mengakses sumber daya yang sama secara bersamaan, seperti memori bersama (shared memory), file, atau variabel global. Jika tidak diatur dengan baik, hal ini dapat menyebabkan data menjadi tidak konsisten atau sistem mengalami gangguan (race condition).

Dalam sistem operasi, critical section harus memenuhi tiga kondisi penting agar berjalan dengan aman, yaitu:

Mutual Exclusion → hanya satu proses yang boleh berada di critical section pada satu waktu.

Progress → jika tidak ada proses yang sedang berada di critical section, maka proses lain yang ingin masuk tidak boleh ditunda tanpa batas.

Bounded Waiting → ada batas waktu seberapa lama suatu proses harus menunggu giliran untuk masuk ke critical section, agar tidak terjadi kelaparan (starvation).

Untuk mengatasi masalah ini, dapat digunakan berbagai solusi sinkronisasi seperti Peterson Solution, penggunaan Lock, Semaphore, TestAndSet Instruction, atau mekanisme sinkronisasi perangkat keras. Teknik-teknik tersebut membantu memastikan bahwa proses dapat berjalan secara bergantian dengan aman tanpa mengganggu satu sama lain.

Sumber: Materi Pengayaan MSIM4201 – Sistem Operasi: [Manajemen Proses \(Synchronization dan Scheduling\)](#), Universitas Terbuka

Wassalamualaikum Wr. Wb.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.4

oleh [NABIL HANIF ATHALLAH 055592508](#) - Kamis, 30 Oktober 2025, 21:45

Assalamualaikum Wr.Wb

Nama : Nabil Hanif Athallah

NIM : 055592508

Selamat malam Pak Ade Chandra Saputra S.Kom.,M.Cs dan teman-teman semuanya.

Saya akan menjawab pertanyaan dari diskusi 4, sebagai berikut :

Yang dimaksud Critical Section adalah segmen kode dalam suatu proses di mana proses tersebut mengakses Shared Resources, seperti variable global, berkas, database atau hardware. Permasalahannya muncul Ketika dua atau lebih proses mencoba masuk ke critical section secara bersamaan. Jika ini terjadi, hasil akhir dari eksekusi kode akan bergantung pada urutan proses dieksekusi, yang tidak dapat diprediksi. Tujuan Utama dari solusi critical section adalah memastikan Mutual Exclusion, yaitu hanya satu proses yang diizinkan berada di dalam critical section pada satu Waktu.

Ada beberapa solusi untuk permasalahan critical section :

- Mutual Exclusion

Jika satu proses sedang mengeksekusi critical sectionnya, maka tidak ada proses lain yang diizinkan masuk ke critical section yang sama.

- Progress

Jika tidak ada proses yang berada di critical section dan ada beberapa proses yang ingin masuk, maka hanya proses-proses yang tidak berada di remainder section yang dapat berpartisipasi dalam pengambilan keputusan untuk memilih proses mana yang kan masuk selanjutnya.

- Bounded Waiting

Harus ada Batasan jumlah Waktu atau jumlah proses lain yang diizinkan untuk masuk ke critical section setelah suatu proses meminta akses dan sebelum permintaan proses tersebut dikabulkan

Sekian dari saya, Mohon maaf bila mana ada kesalahan.

terima kasih.

Wassalamualaikum Wr.Wb

SUMBER : - Sistem Operasi (MSIM4201)

- Andrew S. Tanenbaum, Herbert Bos. Modern Operating Systems.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

**Re: Diskusi.4**

oleh NAJWA SALSABILA AZZAHRA 057245181 - Jumat, 31 Oktober 2025, 15:49

Permasalahan critial selection merupakan suatu sistem yang dimana nantinya akan memiliki banyak proses yang akan berkompetisi ketika menggunakan shared-data atau datanya dingunakan secara bersama-sama. Critial selection ini merupakan bagian bagian dari proses yang memungkinkan untuk mengubah bagian dari proses variabel umum, mengubah tabel, menulis atau mengubah file dan lain sebagainya. Permasalahan critial selection ini terbagi menjadi beberapa bagian yaitu

* Mutual Exclusion

Yang dimana jika proses Pi sedang dieksekusi pada bagian critial selection (dari proses Pi) maka tidak ada proses lain yang dapat dieksekusi pada bagian critial selection.

* Proses

Yaitu jika tidak ada proses yang sedang eksekusi pada bagian critial selectionnya dan terdapat beberapa proses lain yang ingin masuk ke critial selection, maka pemilihannya siapa yang masuk ke critial selection tidak dapat ditunda tanpa batas.

*Bounded Waiting

Yaitu terdapat batasan beberapa lama suatu proses harus menunggu giliran untuk mengakses critial selection seandainya proses lain yang diberikan hak akses ke critial selection. Hal ini dikarnakan untuk menjamin proses dapat mengakses critial selection pada bagian yang tidak ada asumsi mengenai kecepatan eksekusi proses-proses tersebut.

sekian hasil diskusi saya bapak/ibuk jika ada salah dalam pemahaman saya pada diskusi kali ini saya mohon bantuan koreksinya

saya mendapat jawaban ini dari materi yang diberikan dan buku bahan ajar atau modul yang diberikan sekian terimakasih.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)
**Re: Diskusi.4**

oleh ZIBRAN DIRZA ARGADEWA 057030395 - Jumat, 31 Oktober 2025, 19:30

Nama : Zibran Dirza Argadewa

NIM : 0570303995

Critical section adalah bagian proses yang dapat mengubah variabel bersama, tabel, atau file. Jika beberapa proses menjalankannya secara bersamaan, akan terjadi *race condition* yang menyebabkan ketidakstabilan data. Karena itu, diperlukan sinkronisasi agar proses dapat bekerja sama dengan aman.

Setiap proses harus meminta izin sebelum memasuki critical section-nya. Solusi untuk masalah ini harus memenuhi tiga syarat utama:

1. Mutual exclusion

Jika proses sedang dieksekusi pada bagian critical section maka tidak ada proses lain yang dapat dieksekusi pada bagian critical sectionnya.

2. Progress

Jika tidak ada proses yang sedang dieksekusi pada critical section. Maka ada pemilihan siapa yang masuk ke critical section tidak dapat ditunda tanpa terbatas.

3. Bounded Waiting

Ada batasan waktu tunggu bagi setiap proses yang ingin memasuki critical section agar tidak mengalami starvation.

Adapun permasalahan ketika syarat-syarat tidak terpenuhi ini meliputi:

- **Race condition** hasil program bergantung pada urutan proses eksekusi.
- **Deadlock** proses saling menunggu tanpa ada yang bisa melanjutkan.
- **Starvation** Beberapa proses terus-menerus tidak mendapat kesempatan untuk masuk ke bagian kritis.

SUMBER REFERENSI

Sonasa Risunantoro DKK. (2025). Sistem Operasi MSIM4201 Modul 4 4.6. Universitas Terbuka.

[https://herovired-com.translate.goog/old/Learning-hub/topics/critical-section-problem-in-os/?
_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc&_x_tr_hist=true](https://herovired-com.translate.goog/old/Learning-hub/topics/critical-section-problem-in-os/?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc&_x_tr_hist=true)

Tautan permanen Tampilkan induk Balas

Re: Diskusi.4

oleh [054625452 SIWI PUSPITASARI](#) - Jumat, 31 Oktober 2025, 20:17

izin menjawab diskusi 4 ini

Critical section adalah bagian penting dari program dimana dua atau lebih proses yang berjalan bersamaan mengakses data yang sama. Masalah muncul jika proses itu mengubah data secara bersamaan. Jika hal ini tidak dikendalikan dengan baik, maka data bisa menjadi salah, tidak sinkron atau bahkan menyebabkan program tidak berjalan semestinya.

Contoh sederhananya yaitu, ketika ada dua kasir yang sama-sama mengubah jumlah stok barang di komputer. Kasir pertama mengurangi 1 barang, sementara kasir kedua juga melakukannya di waktu yang sama. Akibatnya, stok bisa berkurang dua kali atau malah tidak berubah sama sekali. Inilah contoh sederhana dari masalah critical section.

Masalah yang bisa terjadi adalah :

1. Race condition, yaitu hasil akhir tergantung siapa yang menulis data lebih dulu.
2. Data tidak konsisten, yaitu isi data bisa salah karena diubah oleh banyak proses sekaligus.
3. Deadlock, yaitu proses saling menunggu giliran sehingga tidak ada yang jalan.

Solusi untuk mengatasi masalah tersebut :

1. Mutual Exclusion (Mutex): Memberikan kunci agar satu proses saja yang bisa mengakses data bersama, proses lain harus menunggu.
2. Semaphore: Memberikan sinyal atau izin bagi proses untuk masuk ke bagian critical section secara bergantian.
3. Monitor: Struktur pengendali yang mengatur akses proses ke sumber daya secara otomatis dan aman.

Kesimpulan :

Permasalahan critical section muncul karena tidak adanya pengaturan akses data bersama, sehingga menyebabkan hasil yang tidak konsisten. Solusinya adalah mengatur agar proses berjalan bergantian agar data tetap aman dan akurat.

- 1). Sumber Referensi :

**Re: Diskusi.4**oleh [MUHAMMAD AQSHAL ZAKKI 055457214](#) - Jumat, 31 Oktober 2025, 22:00

Critical section adalah bagian dari kode program di mana proses mengakses dan memanipulasi data bersama yang tidak boleh diakses lebih dari satu proses secara bersamaan untuk menghindari inkonsistensi data.

Masalah utama pada critical section adalah bagaimana mengatur agar saat satu proses sedang menjalankan critical section-nya, proses lain tidak dapat masuk ke critical section tersebut. Konsep utama di sini adalah "mutual exclusion" atau pengecualian bersama, yang memastikan bahwa hanya satu proses yang boleh mengakses critical section pada waktu tertentu. Jika mutual exclusion tidak diterapkan dengan benar, dapat terjadi kondisi balapan (race condition) di mana hasil akhir bergantung pada urutan eksekusi proses yang tidak terduga.

Selain mutual exclusion, solusi masalah critical section juga harus memenuhi:

- Progress (kemajuan): Jika tidak ada proses yang sedang dalam critical section dan ada beberapa proses yang ingin masuk, maka pemilihan proses berikutnya untuk masuk critical section tidak boleh ditunda secara tidak perlu.
- Bounded waiting (pembatasan penantian): Setiap proses yang ingin mengakses critical section akan mendapatkan giliran dalam batas waktu tertentu, sehingga tidak terjadi starvation (process starvation) di mana suatu proses terus menerus ditolak aksesnya.
- Mekanisme pengaturan critical section biasanya digunakan dalam program multithreading untuk melindungi variabel atau sumber daya bersama dari akses bersamaan yang menyebabkan data menjadi tidak konsisten.

Sumber Referensi:

- "Sinkronisasi Proses," arna.lecturer.pens.ac.id.
- "Analisis Efektivitas Semaphore, Mutex, dan Monitor," penerbitadm.pubmedia.id.

**Re: Diskusi.4**oleh [AKNES MULIANA SIREGAR 056288131](#) - Jumat, 31 Oktober 2025, 22:44

izin menjawab bapak ibu tutor

jelaskan permasalahan critical section?

Permasalahan critical section adalah masalah yang terjadi dalam pemrograman paralel ketika beberapa proses atau thread bersaing untuk mengakses sumber daya bersama. Sumber daya bersama ini dapat berupa variabel global, file, atau perangkat keras. Sehingga hal ini menyebabkan suatu kondisi yang dianggap sebagai kerentanan "Race Condition" dalam kamus IT Security.

Critical section adalah bagian dari kode yang mengakses atau memodifikasi sumber daya bersama. Ketika sebuah proses atau thread berada di dalam critical section, proses atau thread lain tidak boleh mengakses sumber daya tersebut. Jika tidak, dapat terjadi race condition, di mana hasil akhir dari program bergantung pada urutan eksekusi proses atau thread yang tidak dapat diprediksi. Hal ini dapat mengakibatkan hasil yang tidak konsisten atau tidak diinginkan.

Misalnya, pertimbangkan sebuah program yang memiliki dua proses, P1 dan P2. Masing-masing proses memiliki variabel global bernama jumlah. Proses P1 menambah nilai jumlah sebesar 1, sedangkan proses P2 mengurangi nilai jumlah sebesar 1. Jika kedua proses ini mengakses variabel jumlah secara bersamaan, dapat terjadi RACE CONDITION. Misalnya, jika proses P1 sedang menambah nilai jumlah, sedangkan proses P2 sedang mengurangi nilai jumlah, maka nilai jumlah akan menjadi tidak konsisten

sumber referensi:<https://kamsib.id/critical-section-mari-mengenal-lebih-jelas/1585/>

Tautan permanen Tampilkan induk Balas

Hide sidebar



Re: Diskusi.4

oleh [TRYNUR ALHAFIIDZ 055334472](#) - Sabtu, 1 November 2025, 03:00

Critical Section merupakan bagian dari suatu program di mana beberapa proses atau thread dapat mengakses sumber daya bersama (shared resource) seperti variabel global, berkas (file), atau perangkat keras pada waktu yang bersamaan.

Permasalahan muncul ketika dua atau lebih proses mengakses dan memodifikasi data yang sama secara bersamaan, sehingga dapat menimbulkan tidak konsisten data atau hasil eksekusi yang tidak diharapkan.

Permasalahan utama dalam critical section adalah bagaimana cara mengatur agar hanya satu proses yang dapat masuk ke dalam critical section pada satu waktu, sementara proses lain harus menunggu giliran.

Masalah ini dikenal sebagai Critical Section Problem, dan menjadi fokus utama dalam sistem operasi untuk memastikan bahwa proses berjalan secara sinkron tanpa konflik terhadap sumber daya bersama.

- **Mutual Exclusion :**

Hanya satu proses yang diizinkan berada di dalam critical section pada suatu waktu. Proses lain harus menunggu hingga proses tersebut selesai.

- **Progress :**

Jika tidak ada proses di dalam critical section dan ada beberapa proses yang ingin masuk, maka sistem harus memastikan salah satu dari mereka dapat segera dieksekusi tanpa penundaan yang tidak perlu.

- **Bounded Waiting :**

Setiap proses yang menunggu untuk masuk ke critical section harus memperoleh gilirannya dalam waktu yang wajar dan tidak menunggu tanpa batas.

Contoh Kasus Permasalahan Critical Section

P1 menulis: "User A menyetor Rp1.000.000"

P2 menulis: "User B menarik Rp500.000"

Kalau input pada waktu yang bersamaan hasil log akan menjadi

User A menyetor Rp1User B menarik Rp500.000

data menjadi tidak konsisten dan sulit dibaca sistem

Tautan permanen Tampilkan induk Balas



Re: Diskusi.4

oleh [053960862 MUHAMMAD DIMAS LUTFI ABDILLAH](#) - Sabtu, 1 November 2025, 07:55

Ijin menjawab diskusi 4

NAMA : Muhammad Dimas Lutfi Abdillah

NIM : 053960862

MATKUL : Sistem Operasi

Critical section merupakan konsep fundamental dalam sistem operasi dan pemrograman paralel yang menggambarkan segmen kode dimana proses atau thread mengakses sumber daya yang dibagi. Masalah utama yang timbul adalah ketika multiple proses mencoba mengakses critical section secara bersamaan, yang dapat

menyebabkan inkonsistensi data dan hasil yang tidak terprediksi. Inti permasalahan critical section terletak pada kebutuhan untuk memastikan hanya satu proses yang dapat mengakses shared resource pada suatu waktu. Bayangkan beberapa orang yang ingin menggunakan printer yang sama secara bersamaan - jika tidak diatur, hasil cetakan akan kacau balau. Demikian pula dalam pemrograman, ketika dua proses mencoba mengubah nilai variabel yang sama secara bersamaan, nilai akhir menjadi tidak dapat diprediksi. Masalah critical section yang tidak terkelola dengan baik memunculkan race condition, dimana hasil eksekusi program bergantung pada urutan eksekusi proses yang tidak terkontrol. Dalam konteks nyata, ini dapat berarti saldo bank yang tidak akurat, data transaksi yang hilang, atau sistem yang mengalami crash. Yang lebih berbahaya, kesalahan ini seringkali sulit dideteksi karena tidak selalu konsisten dalam reproduksinya.

Menyelesaikan permasalahan critical section tidak semudah mengunci akses ke resource. Solusi yang efektif harus memenuhi tiga kriteria utama: mutual exclusion untuk memastikan eksklusivitas akses, progress untuk menjamin kelangsungan eksekusi program, dan bounded waiting untuk mencegah starvation proses tertentu.

Berbagai teknik telah dikembangkan, mulai dari solusi perangkat lunak seperti algoritma Peterson, hingga mekanisme perangkat keras seperti test-and-set instruction. Sistem operasi modern menyediakan primitive seperti semaphore, mutex, dan monitor yang menyederhanakan implementasi pengelolaan critical section. Namun, pemilihan teknik yang tepat tetap memerlukan pertimbangan mendalam mengenai karakteristik aplikasi dan lingkungan eksekusinya.

Sumber Referensi : BMPMSIM4201 / Modul 4 / Hal 4.2-4.19

<https://ocw.ui.ac.id/course/sistem-operasi/>

<https://www.codepolitan.com/course/sistem-operasi>

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh [SARAH HANIFAH 053132687](#) - Sabtu, 1 November 2025, 13:21

Selamat siang pak Ade Chandra Saputra,

Saya izin menjawab diskusi terlampir.

Permasalahan critical section adalah isu utama dalam sistem komputer yang berkaitan dengan pengelolaan akses ke sumber daya bersama secara simultan oleh berbagai proses atau thread. Tujuan utama dari pengelolaan critical section adalah untuk memastikan bahwa hanya satu proses yang dapat mengakses sumber daya tertentu pada satu waktu guna mencegah kondisi balapan (race condition), inkonsistensi data, dan kerusakan data (Rinusantoro dkk., 2025).

Dalam konteks sistem terdistribusi, permasalahan critical section menjadi semakin kompleks karena proses tidak berbagi memori secara langsung dan harus berkomunikasi melalui pesan untuk mengkoordinasikan akses ke sumber daya bersama. Atreya, Mittal, dan Peri (2007) mengidentifikasi pentingnya algoritma mutual exclusion berbasis quorum untuk mengatur akses secara efisien dalam sistem terdistribusi yang dinamis, di mana grup proses dapat berubah-ubah. Mereka menyoroti tantangan dalam memastikan bahwa semua proses dapat mencapai konsensus tentang siapa yang berhak masuk ke critical section tanpa menyebabkan deadlock atau starvation.

Selain itu, Bierkowski dkk. (2016) menambahkan bahwa dalam jaringan multiple access channel, algoritma mutual exclusion harus mampu mengatasi kondisi kontensi dan memastikan keadilan serta efisiensi dalam pengaturan akses. Mereka memperkenalkan pendekatan acak untuk mengurangi kemungkinan konflik dan meningkatkan skalabilitas.

Secara umum, permasalahan critical section meliputi syarat-syarat berikut (Rinusantoro dkk., 2025):

1. Mutual Exclusion: Hanya satu proses yang boleh mengakses sumber daya pada satu waktu.
2. Progress: Jika tidak ada proses yang sedang dalam critical section, proses lain harus dapat masuk ke critical section tanpa penundaan yang tidak perlu.
3. Bounded Waiting: Tidak boleh ada proses yang harus menunggu waktu yang tidak terbatas sebelum dapat mengakses critical section.

Dalam sistem terdistribusi dan jaringan, algoritma seperti yang dibahas oleh Joung (1999, 2001) dan Gokhale dan Mittal (2018) berusaha mengatasi permasalahan ini melalui pendekatan berbasis quorum, token, dan metode acak untuk memastikan bahwa proses dapat mengakses critical section secara efisien dan adil.

Secara keseluruhan, permasalahan critical section merupakan tantangan utama dalam desain algoritma sinkronisasi dan koordinasi proses yang harus mampu menjamin konsistensi data, keadilan, dan efisiensi dalam pengelolaan sumber daya bersama.

Referensi:

- Atreya, R., Mittal, N., & Peri, S. (2007). A quorum-based group mutual exclusion algorithm for a distributed system with dynamic group set. *IEEE Transactions on Parallel and Distributed Systems*, 18(10), 1345-1360. <https://doi.org/10.1109/tpds.2007.1072>
- Bieńkowski, M., Klonowski, M., Korzeniowski, M., & Kowalski, D. (2016). Randomized mutual exclusion on a multiple access channel. *Distributed Computing*, 29(5), 341-359. <https://doi.org/10.1007/s00446-016-0265-z>
- Gokhale, S., & Mittal, N. (2018). Fast and scalable group mutual exclusion. <https://doi.org/10.48550/arxiv.1805.04819>
- Joung, Y. (1999). The congenial talking philosophers problem in computer networks. In Proceedings of the International Conference on Computer Communications and Networks (pp. 195-211). https://doi.org/10.1007/3-540-48169-9_14
- Joung, Y. (2001). Quorum-based algorithms for group mutual exclusion. In Proceedings of the International Conference on Distributed Computing Systems (pp. 16-32). https://doi.org/10.1007/3-540-45414-4_2
- Rinusantoro, dkk. (2025). Sistem Operasi. Universitas Terbuka: Tangerang Selatan.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)
**Re: Diskusi.4**oleh [ZIDAN MUHAMAD DAFFA 056388169](#) - Sabtu, 1 November 2025, 13:41

Izin menanggapi topik diskusi sesi 4 sebagai berikut:

Critical section adalah bagian dari suatu proses yang melakukan akses dan manipulasi terhadap sumber daya bersama (shared resources) yang dapat menyebabkan konflik jika dijalankan secara bersamaan oleh beberapa proses . Ketika sebuah proses sedang dijalankan dalam critical section-nya, tidak boleh ada proses lain yang dijalankan dalam critical section tersebut pada waktu yang sama, karena hal ini akan menciptakan kondisi yang tidak aman dan hasil yang tidak dapat diprediksi .

Tiga Kondisi Utama dalam Critical Section

Permasalahan critical section mencakup tiga kondisi utama yang harus dipenuhi untuk memastikan keamanan akses terhadap sumber daya bersama :

1. Mutual Exclusion

Berarti jika proses Pi sedang dieksekusi pada bagian critical section dari proses Pi, maka tidak ada proses lain yang dapat dieksekusi pada bagian critical section-nya . Dengan kata lain, hanya satu proses yang dapat mengakses sumber daya bersama pada satu waktu tertentu .

2. Progress

Mengatur bahwa jika tidak ada proses yang sedang eksekusi pada critical section-nya dan terdapat beberapa proses lain yang ingin masuk ke critical section, maka pemilihan proses yang akan masuk ke critical section tidak dapat ditunda tanpa batas . Kondisi ini memastikan bahwa jika ada proses yang menunggu untuk masuk ke critical section sementara tidak ada proses lain yang menggunakaninya, maka salah satu dari proses yang menunggu tersebut harus segera diberikan akses .

3. Bounded Waiting

Menetapkan bahwa terdapat batasan berapa lama suatu proses harus menunggu giliran untuk mengakses critical section, seandainya proses lain yang diberikan hak akses ke critical section . Hal ini digunakan untuk menjamin bahwa setiap proses dapat mengakses critical section dalam waktu yang terbatas dan mencegah terjadinya starvation, yaitu situasi di mana suatu proses terus-menerus diabaikan dan tidak pernah mendapatkan kesempatan untuk mengakses sumber daya .

Metode Penyelesaian

Untuk mengatasi masalah critical section, modul menyebutkan beberapa metode antara lain Solusi Peterson, penggunaan Kunci (Lock), Sinkronisasi Perangkat Keras, dan Solusi dengan instruksi TestAndSet . Salah satu pendekatan yang lebih efisien adalah penggunaan Semaphore, yang merupakan suatu cara sinkronisasi tanpa

memerlukan penungguan busy waiting dengan prinsip kerja menggunakan suatu penanda sebagai media kerja sama dari beberapa proses .

Konsekuensi Jika Tidak Ditangani

Apabila permasalahan critical section tidak ditangani dengan baik, dapat terjadi beberapa masalah serius seperti race condition (kondisi di mana hasil akhir tidak dapat diprediksi), deadlock (dua atau lebih proses saling menunggu tanpa batas waktu), dan starvation (proses tidak mendapatkan resource yang diminta dalam jangka waktu yang sangat lama) . Oleh karena itu, penanganan critical section menjadi aspek fundamental dalam manajemen proses pada sistem operasi modern.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh YUSUF FAJRI KURNIANTO 055178877 - Sabtu, 1 November 2025, 20:28

Mohon izin untuk menjawab diskusi 4.

Permasalahan critical section yaitu Bagaimana cara mengatur agar beberapa proses yang berjalan bersamaan tidak masuk ke dalam critical section secara bersamaan, sehingga tidak terjadi konflik atau inkonsistensi data. Jika dua proses masuk ke critical section bersamaan, bisa terjadi masalah seperti:

- Nilai variabel bersama menjadi salah (data race),
- File yang diakses rusak,
- Hasil perhitungan tidak sesuai harapan.

Tujuan atau syarat solusi Critical Section.

Solusi yang baik untuk masalah critical section harus memenuhi tiga kondisi utama (menurut Dijkstra):

- Hanya satu proses yang boleh berada di critical section pada suatu waktu.
- Jika tidak ada proses di critical section, maka proses lain yang ingin masuk tidak boleh ditunda tanpa alasan.
- Setiap proses harus mendapatkan kesempatan masuk ke critical section dalam waktu terbatas, tidak boleh menunggu selamanya (no starvation).

Terima kasih

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh 054160782 RIZA HARDIAN - Sabtu, 1 November 2025, 22:43

Izin menanggapi diskusi sebagai berikut,

Permasalahan di critical section muncul ketika dua proses atau lebih dalam sebuah thread mengakses sumber daya bersama secara simultan, misalnya variabel global atau berkas. Jika tidak ditangani dengan baik, ini dapat menyebabkan kondisi balapan, ketidakselarasan data, dan bahkan deadlock. Penyelesaian untuk masalah ini adalah dengan menggunakan mekanisme sinkronisasi, seperti mutex, semaphore, atau monitor, sehingga hanya satu proses yang diperbolehkan masuk ke bagian critical section pada satu waktu (exclusion mutual). Fadhila et al. (2025) dalam Jurnal Komputer, Informasi dan Teknologi menyatakan bahwa masalah di bagian critical section dapat dihindari dengan mempertahankan mutual exclusion, progress, dan bounded waiting, sehingga setiap proses bisa mengakses daerah kritis tanpa mengalami konflik.

Sumber referensi :

Fadhila, A. R., Nugraha, A. H., Mazaya, C. H., Ramadhan, M. S., & Kusnendar, J. (2025). Optimalisasi sinkronisasi proses: Analisis efektivitas semaphore, mutex, dan monitor. Jurnal Komputer, Informasi dan Teknologi, 5(1).

<https://penerbitadm.pubmedia.id/index.php/KOMITEK/article/view/2595>

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh RAFA NURHUDAN WIGUNA 055747162 - Minggu, 2 November 2025, 00:39

izin menjawab diskusi 4

Permasalahan Critical Section adalah tantangan utama dalam sinkronisasi proses dalam sistem komputasi paralel. Inti masalahnya terletak pada sumber daya bersama (shared resources). Untuk mencegah kondisi balapan (race condition) yang mengakibatkan inkonsistensi data, kita harus merancang protokol ketat. Protokol ini harus menjamin Mutual Exclusion—bahwa hanya satu proses yang boleh berada di segmen kode kritisnya pada satu waktu. Solusi harus juga memastikan kemajuan (progress) dan menunggu terbatas (bounded waiting) agar sistem tidak mengalami kebuntuan (deadlock) atau kelaparan (starvation).

Isu Critical Section merujuk pada kebutuhan mendesak untuk menegakkan kontrol akses ke bagian kode yang mengakses data atau sumber daya yang dibagi oleh banyak thread atau proses. Tujuan utamanya adalah mencegah manipulasi data yang tumpang tindih (interleaved) yang berpotensi menghasilkan output tidak deterministik atau salah. Solusi yang efektif harus mencakup tiga jaminan atomik:

1. Saling Pengecualian : Memastikan akses tunggal ke sumber daya.
2. Kemajuan : Mencegah penundaan tak terbatas dalam pemilihan proses yang siap masuk.
3. Menunggu Terbatas: Memberikan batasan waktu tunggu yang wajar bagi setiap proses. Mekanisme seperti Mutex, Semaphore, atau instruksi hardware digunakan untuk memenuhi persyaratan protokol ini.

Critical Section adalah masalah desain protokol yang bertujuan menjaga integritas dan keandalan data dalam lingkungan multitasking. Jika sistem membiarkan beberapa entitas mengubah data bersama secara simultan, hasil akhirnya tidak dapat diandalkan. Oleh karena itu, critical section membutuhkan "gerbang" sinkronisasi. Gerbang ini harus berfungsi dengan adil dan efisien sambil memastikan keamanan mutlak. Kegagalan pada protokol ini secara langsung mengancam stabilitas dan keakuratan operasi sistem.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh [RADEN AGUNG PRAKOSO WIBOWO 052866944](#) - Minggu, 2 November 2025, 02:27

Selamat pagi

izin untuk memberikan pendapat terkait masalah critical section

file terlampir



[diskusi 4.pdf](#)

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)



Re: Diskusi.4

oleh [TATA PUTRIA SALASABILA 057204874](#) - Minggu, 2 November 2025, 06:00

Assalamualaikum Wr.Wb

Nama: Tata Putria Salsabila

NIM: 057204874

Prodi Sistem Informasi

UPBJJ-UT Surakarta

Izin menjawab Diskusi 4

Suatu sistem nantinya akan memiliki banyak proses yang akan berkompetisi dalam menggunakan shared-data atau data yang digunakan bersama-sama, dari setiap proses tersebut akan memiliki segmen atau bagian yang disebut bagian kritis (critical section). Critical section merupakan bagian dari proses yang memungkinkan untuk mengubah variabel umum, mengubah tabel, menulis atau mengubah file, dll.

permasalahan critical section digunakan untuk mendesain sebuah protokol dimana proses-proses dapat bekerja sama. Dan masing-masing proses harus memiliki ijin untuk memasuki critical section:

- Mutual Exclusion Jika proses Pi sedang dieksekusi pada bagian critical section (dari proses Pi) maka tidak ada proses

lain yang dapat dieksekusi pada bagian critical sectionnya.

-Progress Jika tidak ada proses yang sedang eksekusi pada critical sectionnya dan terdapat beberapa proses lain yang ingin masuk ke critical section, maka pemilihan siapa yang masuk ke critical section tidak dapat ditunda tanpa terbatas.

-Bounded Waiting Terdapat batasan berapa lama suatu proses harus menunggu giliran untuk mengakses critical section seandainya proses lain yang diberikan hak akses ke critical section. Hal ini digunakan untuk menjamin proses dapat mengakses critical section.

Untuk mengatasi permasalahan dalam critical section, berbagai teknik sinkronisasi digunakan, seperti mutex (mutual exclusion), semafor, monitor, dan lainnya. Tujuan utamanya adalah untuk memastikan bahwa hanya satu proses yang memiliki akses ke critical section pada satu waktu dan untuk mencegah konflik dan deadlock. Pemahaman yang baik tentang masalah-masalah ini dan penggunaan alat-alat sinkronisasi yang tepat adalah kunci untuk menghindari masalah dalam pengembangan perangkat lunak paralel dan sistem yang aman.

Referensi: Modul MSIM4321

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

Re: Diskusi.4

oleh [053681662 LAURA ANGELINA PAKPAHAN](#) - Minggu, 2 November 2025, 08:55

Izin Menanggapi soal diskusi 4

critical section hanya timbul bila dua atau beberapa proses atau threads coba memperoleh beberapa sumber daya yang sama pada waktu yang bersamaan, seperti variabel global, file, atau perangkat I/O. Dalam hal ini, akan terjadi gangguan jika ia tahu bahwa sumber daya yang bersamaan tidak ada yang benar-benar mengurus pelebaran kritis, sebagai contoh: data menjadi tak konsisten atau hasil perhitungan menjadi salah. Bagian-bagian yang mengakses sumber daya bersama ini dinamakan critical section, dan hanya satu proses yang boleh menjalankannya pada satu waktu.

Masalah utama adalah bagaimana memastikan bahwa proses-proses ini saling bergilir dengan selamat. Untuk itu, ada tiga syarat yang harus dipenuhi, yaitu:

Mutual Exclusion – hanya satu proses yang boleh masuk critical section pada satu waktu.

Progress – tetapi tidak ada proses yang dalam critical section, maka proses yang lain langsung memasukinya dengan cara yang sesuai keinginan dirinya, dengan halilintar suatu penundaan. Tanpa dipihak entah sekadar pada waktu yang sudah ditetapkan sendiri atau sapan pendapat orang lain, karena di pintu masuk tersebut orang menyerobot

Bounded Waiting – setiap proses yang menunggu untuk masuk ke critical section pasti akan mendapat kesempatan dalam waktu yang tidak terlalu lama.

Dengan ketiga syarat ini itu dibuktikan, akan menghindari konflik data dan memastikan proses berjalan dengan cara yang konsisten efisien.

Referensi :

Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Sistem Operasi: Konsep Dasar (Edisi 9). Yogyakarta: Andi.Sutanta, E. (2011). Pengantar Sistem Operasi. Yogyakarta: Andi.

[Tautan permanen](#) [Tampilkan induk](#) [Balas](#)

◀ Manajemen Proses

Lompat ke...

 Navigasi

Hide sidebars

✓ Dasbor**⌂ Beranda situs**

> Laman situs

✓ Kelasku> [STSI4203.108](#)> [STSI4202.42](#)> [STSI4103.119](#)> [Peserta](#)**☒ Nilai**> [Pendahuluan](#)> [Sesi 1](#)> [Sesi 2](#)> [Sesi 3](#)> [Sesi 4](#)**☒ Kehadiran Sesi ke-4** [Manajemen Proses \(Synchronization dan Scheduling\).](#) [Manajemen Proses](#)**✉ Diskusi.4**> [MKKI4201.278](#)> [STSI4201.161](#)> [STSI4205.331](#)> [STSI4104.284](#)> [MKDI4202.1514](#)> [Kelas](#)**⚙ Administrasi****✓ Forum administrasi**

Berlangganan dinonaktifkan

Follow Us:

UNIVERSITAS TERBUKA ©2025

Anda masuk sebagai [INDRAWAN LISANTO 053724113](#) (Keluar)[Dapatkan aplikasi seluler](#)