

Sistem Terdistribusi (ST)

Pertemuan 5 : Mutual Exclusion

Alauddin Maulana Hirzan

Fakultas Teknologi Informasi dan Komunikasi
Universitas Semarang

Outline

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

1 Mutual Exclusion

2 Algoritma Lamport

3 Algoritma Ricart-Agrawala

4 Lastly

Latar Belakang

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Sistem terdistribusi terdiri dari sekumpulan komputer otonom yang saling terhubung melalui jaringan dan bekerja sama untuk mencapai tujuan bersama. Setiap node menjalankan proses secara independen dan berkomunikasi menggunakan mekanisme message passing, bukan berbagi memori secara langsung.

Dalam kondisi nyata, beberapa proses sering kali membutuhkan akses terhadap sumber daya bersama seperti:

- 1 file sistem
- 2 database terdistribusi
- 3 perangkat keras seperti printer atau sensor

Latar Belakang

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Apabila tidak terdapat mekanisme pengaturan akses, dua atau lebih proses dapat memasuki critical section secara bersamaan. Kondisi ini berpotensi menimbulkan:

- 1 inkonsistensi data
- 2 konflik penulisan (write-write conflict)
- 3 kehilangan pembaruan (lost update)

Masalah tersebut dikenal sebagai Distributed Mutual Exclusion, yaitu permasalahan bagaimana menjamin hanya satu proses pada satu waktu yang dapat mengakses bagian kritis dari sistem dalam lingkungan yang tidak memiliki kontrol terpusat.

Tujuan Mutual Exclusion

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Mutual exclusion merupakan konsep fundamental dalam sistem operasi dan sistem terdistribusi yang bertujuan mengatur akses terhadap critical section agar eksekusi proses tetap konsisten dan aman.

Tujuan utama mutual exclusion adalah:

- 1 Menjamin hanya satu proses berada di critical section pada satu waktu
- 2 Mencegah race condition
- 3 Menjaga konsistensi data

Tujuan Mutual Exclusion

1. Menjamin hanya satu proses berada di critical section pada satu waktu

Hal ini penting untuk mencegah konflik operasi pada data bersama. Dalam sistem terdistribusi, hal ini harus dicapai tanpa bantuan shared memory atau scheduler pusat.

2. Mencegah race condition

Race condition terjadi ketika hasil akhir eksekusi bergantung pada urutan kedatangan pesan atau eksekusi proses. Pada sistem terdistribusi, kondisi ini lebih kompleks karena adanya:

- 1 latensi jaringan
- 2 reordering paket
- 3 kegagalan node

Tujuan Mutual Exclusion

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

3. Menjaga konsistensi data

Tanpa mutual exclusion, data pada replika atau penyimpanan terdistribusi dapat berada dalam keadaan tidak sinkron. Mutual exclusion membantu memastikan bahwa operasi write terjadi secara terurut dan deterministik.

Secara formal, algoritma mutual exclusion pada sistem terdistribusi harus memenuhi tiga properti:

- 1 Safety** : tidak lebih dari satu proses di critical section
- 2 Liveness** : setiap proses yang meminta akses akhirnya mendapat giliran
- 3 Fairness** : tidak terjadi starvation

Tantangan pada Sistem Terdistribusi

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Implementasi mutual exclusion pada sistem terdistribusi jauh lebih kompleks dibandingkan sistem terpusat karena beberapa karakteristik dasar berikut:

- 1 Tidak ada shared memory
- 2 Tidak ada global clock
- 3 Komunikasi melalui message passing
- 4 Delay jaringan tidak dapat diprediksi

Tantangan pada Sistem Terdistribusi

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

1. Tidak ada shared memory

Pada sistem terpusat, mutual exclusion dapat dicapai dengan mekanisme seperti:

- 1 mutex
- 2 semaphore
- 3 monitor

Namun dalam sistem terdistribusi, proses berada pada mesin berbeda sehingga tidak dapat mengakses variabel yang sama secara langsung. Semua koordinasi harus dilakukan melalui pesan jaringan.

Tantangan pada Sistem Terdistribusi

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

2. Tidak ada global clock

Setiap node memiliki jam lokal sendiri yang tidak selalu sinkron. Hal ini menyulitkan penentuan urutan kejadian secara global. Masalah ini mendorong pengembangan konsep seperti:

- 1 logical clock
- 2 Lamport timestamp

yang digunakan untuk mengurutkan permintaan akses critical section secara konsisten.

Tantangan pada Sistem Terdistribusi

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

3. Komunikasi melalui message passing

Semua koordinasi mutual exclusion dilakukan dengan:

- 1 request message
- 2 reply message
- 3 release message

Hal ini menambah overhead komunikasi dan memperbesar kompleksitas algoritma, karena setiap proses harus mempertimbangkan status proses lain.

Tantangan pada Sistem Terdistribusi

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

4. Delay jaringan tidak dapat diprediksi

Jaringan dapat mengalami:

- 1 latency
- 2 jitter
- 3 packet loss

Akibatnya, pesan permintaan critical section bisa tiba dalam urutan yang berbeda dari waktu pengirimannya, sehingga algoritma harus mampu menangani ketidakpastian tersebut tanpa melanggar safety dan fairness.

Definisi Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Algoritma mutual exclusion yang diperkenalkan oleh **Leslie Lamport** pada tahun 1978 merupakan salah satu solusi paling awal untuk mengatasi masalah distributed mutual exclusion tanpa shared memory.

Algoritma ini dibangun di atas tiga konsep utama:

- 1 Logical Clock
- 2 Total Ordering pada Pesan
- 3 Message Passing untuk Koordinasi

Definisi Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

1. Logical Clock

Karena tidak terdapat global clock pada sistem terdistribusi, Lamport memperkenalkan konsep **Logical Clock** yang memungkinkan proses mengurutkan kejadian berdasarkan hubungan sebab-akibat (happened-before).

Setiap proses:

- 1 memiliki counter lokal
- 2 menambah nilai clock setiap terjadi event
- 3 menyertakan timestamp pada setiap pesan

Definisi Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

2. Total Ordering pada Pesan

Lamport menggabungkan:

- 1 timestamp logical clock
- 2 ID proses

untuk menghasilkan total ordering. Jika dua timestamp sama, maka ID proses digunakan sebagai pembanding. Dengan demikian, semua node memiliki urutan permintaan yang identik.

Definisi Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

3. Message Passing untuk Koordinasi

Tidak seperti sistem terpusat yang menggunakan mutex atau semaphore, algoritma ini sepenuhnya mengandalkan:

- 1 pengiriman pesan
- 2 queue permintaan
- 3 persetujuan dari proses lain

Tujuan utama dari pendekatan ini adalah:

- 1 menjamin akses ke critical section secara terurut
- 2 menjaga keadilan (fairness) antar proses

Jenis Pesan Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

1. REQUEST

Pesan ini dikirim ketika suatu proses ingin memasuki critical section.

Isi pesan:

- 1 timestamp logical clock
- 2 ID proses pengirim

Semua proses lain akan menyimpan permintaan tersebut di dalam request queue lokal

Jenis Pesan Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

2. REPLY

Setiap proses yang menerima REQUEST akan mengirimkan REPLY sebagai tanda persetujuan bahwa proses peminta boleh memasuki critical section ketika gilirannya tiba.

REPLY bukan berarti izin langsung masuk, tetapi merupakan bukti bahwa proses lain mengetahui permintaan tersebut.

Jenis Pesan Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

3. RELEASE

Setelah proses keluar dari critical section, proses tersebut mengirimkan RELEASE ke semua node untuk:

- 1 menghapus request dari queue
- 2 memberi kesempatan kepada proses berikutnya

Skema tiga pesan ini membentuk protokol sinkronisasi yang sepenuhnya terdistribusi.

Cara Kerja Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Tahap 1 – Mengirim Permintaan

Ketika proses P_i ingin memasuki critical section:

- 1 menaikkan logical clock
- 2 membuat pesan REQUEST dengan timestamp
- 3 mengirim REQUEST ke semua proses lain

Secara bersamaan, proses juga menambahkan permintaannya sendiri ke queue lokal

Cara Kerja Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Tahap 2 – Mengelola Queue Permintaan

Setiap node memelihara priority queue berdasarkan timestamp

Queue ini memastikan bahwa semua proses memiliki urutan permintaan yang konsisten secara global.

Tahap 3 – Kondisi Masuk Critical Section

Suatu proses hanya dapat memasuki critical section jika dua syarat terpenuhi:

- 1 Telah menerima REPLY dari semua proses lain
- 2 Permintaan miliknya berada pada posisi teratas queue

Kompleksitas Pesan pada Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Algoritma Lamport dikenal memiliki biaya komunikasi tinggi karena setiap permintaan membutuhkan koordinasi dengan seluruh node.

Untuk sistem dengan N proses. Pesan yang dikirim:

- 1 Request $\rightarrow (N-1)$
- 2 Reply $\rightarrow (N-1)$
- 3 Release $\rightarrow (N-1)$

Total Pesan:

- $3(N-1)$

Kompleksitas Pesan pada Algoritma Lamport

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Contoh

Jika terdapat 5 node:

$$3(5 - 1) = 12 \text{ pesan}$$

Kompleksitas ini bersifat:

- 1 $O(N)$ untuk setiap fase
- 2 $O(N)$ latency karena harus menunggu semua REPLY

Kelebihan dan Kekurangan

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Kelebihan

- 1 Ordering Konsisten Secara Global.** Penggunaan logical clock menjamin bahwa semua proses memiliki:
 - urutan permintaan yang sama
 - keputusan yang deterministik
- 2 Fairness Tinggi.** Karena permintaan diproses berdasarkan timestamp:
 - tidak ada proses yang dapat memotong antrian
 - starvation hampir tidak mungkin terjadi

Kelebihan dan Kekurangan

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Kekurangan

- 1 Overhead Pesan Besar.** Setiap akses critical section memerlukan:
 - broadcast ke semua node
 - Respon dari semua node
- 2 Sensitif terhadap Kegagalan Node.** Jika satu node:
 - crash
 - atau tidak mengirim REPLY

Motivasi Algoritma

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Algoritma **Glenn Ricart** dan **Ashok K. Agrawala** dikembangkan pada tahun 1981 sebagai optimasi terhadap algoritma **Leslie Lamport**.

Masalah utama pada algoritma Lamport adalah:

- 1 jumlah pesan yang besar
- 2 kebutuhan pesan tambahan untuk RELEASE

Ricart–Agrawala mengusulkan pendekatan yang lebih efisien dengan:

- 1 menghapus pesan RELEASE
- 2 menggunakan mekanisme deferred reply untuk mengatur giliran akses

Jenis Pesan pada Ricart–Agrawala

Berbeda dengan Lamport yang menggunakan tiga jenis pesan, algoritma ini hanya menggunakan dua jenis pesan:

- 1 REQUEST** : Digunakan oleh proses untuk:
 - mengumumkan ke semua node bahwa critical section ingin diakses
 - menyertakan timestamp logical clock
- 2 REPLY** : Merupakan izin eksplisit dari node lain bahwa proses boleh masuk critical section.

Tidak terdapat pesan RELEASE karena:

- 1** pelepasan critical section dilakukan secara implisit
- 2** node yang menunda REPLY akan mengirimkannya setelah keluar dari critical section

Pendekatan ini mengurangi satu fase komunikasi penuh dalam setiap siklus mutual exclusion.

Cara Kerja Ricart-Agrawala

Ketika suatu proses P_i menerima REQUEST dari P_j , terdapat tiga kemungkinan:

- 1** P_i tidak membutuhkan critical section
 - Jika proses tidak sedang:
 - berada di critical section
 - maupun menunggu critical section
 - maka: langsung mengirim REPLY ke P_j
- 2** P_i sedang berada di critical section
 - Jika proses sedang menggunakan resource:
 - REPLY ditunda sampai P_i keluar dari critical section
- 3** P_i juga meminta critical section
 - Jika P_i juga memiliki permintaan aktif:
 - dilakukan perbandingan timestamp antara dua request

Dalam kondisi ini, salah satu proses akan menunda REPLY agar urutan akses tetap konsisten. Mekanisme ini dikenal sebagai: **Deferred Reply.**

Aturan Prioritas pada Ricart–Agrawala

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Untuk menentukan proses mana yang mendapat akses lebih dahulu, algoritma menggunakan aturan ordering yang sama dengan Lamport, yaitu:

1. Timestamp lebih kecil memiliki prioritas lebih tinggi

Timestamp berasal **Lamport Timestamp**

2. Jika timestamp sama, maka ID proses lebih kecil menang

Hal ini diperlukan untuk:

- 1 menghindari ambiguitas
- 2 memastikan total ordering tetap tercapai

Kompleksitas Pesan Ricart–Agrawala

Algoritma ini berhasil menurunkan kompleksitas komunikasi dibandingkan Lamport.

Untuk sistem dengan N proses:

- 1 Request $\rightarrow (N - 1)$
- 2 Reply $\rightarrow (N - 1)$

Sehingga total: $2(N-1)$

Perbandingan

Algoritma	Total Pesan
Lamport	$3(N - 1)$
Ricart-Agrawala	$2(N - 1)$

Kelebihan dan Kekurangan

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Kelebihan

- 1 Jumlah pesan lebih sedikit.** Penghapusan pesan RELEASE membuat algoritma ini lebih efisien dalam:
 - jaringan dengan bandwidth terbatas
 - sistem dengan frekuensi akses critical section tinggi
- 2 Protokol lebih sederhana.** Hanya dua jenis pesan yang perlu ditangani, sehingga:
 - implementasi lebih mudah
 - overhead manajemen queue lebih kecil dibanding Lamport

Kelebihan dan Kekurangan

Sistem
Terdistribusi
(ST)

Alauddin
Maulana
Hirzan

Mutual
Exclusion

Algoritma
Lamport

Algoritma
Ricart-
Agrawala

Lastly

Kekurangan

- 1** Sensitif terhadap kegagalan node. Jika satu node **crash** maka proses lain tidak akan pernah menerima REPLY dari node tersebut, sehingga: **mutual exclusion** menjadi **blocked indefinitely**
- 2** Tidak cocok untuk sistem berskala sangat besar. Walaupun lebih efisien dari Lamport, algoritma ini masih:
 - menggunakan broadcast REQUEST
 - menunggu REPLY dari semua node

Terima Kasih