

Perencanaan Dan Implementasi Stabilitas Memori Internal Terhadap CPU

Nurul Azzahhra Sukamto¹, Nurlaila², Bunga Umni Kurniati³, Indra Gunawan⁴

STIKOM TUNAS BANGSA

Email:nurula120306@gmail.com¹,nurlaila6555@gmail.com²,bungaummi18@gmail.com³, indra@amiktunasbangsa.ac.id⁴.

Abstrak

Stabilitas memori internal sangat penting untuk memastikan sistem komputer berjalan dengan baik, khususnya dalam menjaga efisiensi kinerja cpu. Artikel ini membahas perencanaan dan implementasi stabilitas memori internal, berfokus pada masalah kritis Koherensi Cache (Cache Coherency) yang menjadi pada arsitektur CPU multi-core. Masalah ini mengakibatkan inkonsistensi data dan kesalahan logika karena salinan data yang usang digunakan oleh salah satu core. Penelitian ini merumuskan dan mengimplementasikan strategi penguncian perangkat lunak (software locking) untuk mengatasi isu tersebut. Hasil implementasi menunjukkan peningkatan integritas data yang signifikan, berhasil meniadakan kesalahan logika dari 2.50% menjadi 0.00% pada operasi data kritis. Meskipun solusi menimbulkan sedikit overhead latensi yang stabil, ini adalah trade-off yang terkelola dibandingkan dengan latensi yang acak dan tak terduga yang disebabkan oleh beban protocol snooping. *Kata kunci: memori internal, perencanaan, implementasi, dan stabilitas, koherensi cache.*

Abstract

The stability of internal memory is paramount to ensure the proper functioning of a computer system, particularly in maintaining the efficiency of CPU performance. This article discusses the planning and implementation of internal memory stability, focusing on the critical issue of Cache Coherency that occurs in multi-core CPU architectures. This problem leads to data inconsistency and logical errors because outdated data copies are used by one of the cores. This research formulates and implements a software locking strategy to address this issue. The implementation results show a significant improvement in Data Integrity, successfully eliminating Logical Errors from 2.50% down to 0.00% in critical data operations. Although the solution introduces a slight, stable latency overhead, this is a manageable trade-off compared to the random and unpredictable latency caused by the burden of snooping protocols.

Keywords: internal memory, planning, implementation, stability, Cache Coherency.

PENDAHULUAN

Zaman yang semakin canggih dan pesat membuat keperluan masyarakat semakin meningkat. Dengan kemajuan teknologi membuat pengguna lebih merasa ingin mencari tahu isi cara kerja dan alur-alur yang dilalui perangkat keras dan mengelola suatu informasi. Memori pada cpu sangat penting dimana menjadi komponen utama pada cpu, memori merupakan media yang sering digunakan untuk menyimpan informasi untuk penggunaan langsung dalam laptop, memori yang biasa digunakan untuk menyimpan informasi untuk penggunaan langsung dalam laptop. Memori juga terdiri dari RAM (Random Access Memory), yang dimana bila RAM bila mempunyai kapasitas atau penyimpanan yang semakin tinggi atau besar, maka perangkat beroperasi dengan cepat dan mudah. [1]

KAJIAN TEORITIS

a. Konsep Dasar Memori

Secara umum komputer terdiri dari empat komponen utama, yaitu perangkat input, perangkat output, prosesor utama, dan memori. Central Processing Unit (CPU) merupakan “otak” komputer yang digunakan untuk melakukan operasi dasar seperti operasi perbandingan, perhitungan, pembacaan dan penulisan. Memori merupakan suatu komponen yang digunakan untuk menyimpan atau mengingat. Program (operasi yang dilakukan oleh prosesor) dan data atau informasi (hal-hal yang dikelola oleh proses) disimpan dalam memori. [2]

RAM (Random Access Memory) RAM adalah Memory tempat Penyimpanan sementara pada saat komputer di jalankan dan dapat di akses secara acak atau random. Fungsi dari RAM adalah mempercepat Pemrosesan data pada komputer. Semakin besar RAM yang Anda miliki, semakin cepatlah Komputer yang Anda miliki. [3] Internal random access memory (RAM) adalah memori komputer yang dibangun langsung ke dalam chip mikrokontroler, seperti central processing unit komputer (CPU). Internal RAM ini digunakan oleh programmer untuk meningkatkan kecepatan fungsi program yang langsung ditangani RAM internal, dan memastikan bahwa proses antri data yang akan diproses lebih cepat. Internal RAM ini dapat mempercepat kinerja prosesor karena instruksi yang sering digunakan dapat dikirimkan ke CPU lebih cepat daripada menarik mereka dari RAM eksternal [4]

Memori utama (primary memory) adalah modul atau panduan komputer yang berfungsi sebagai tempat program dan tempat pengintruksian pengoperasian komputer yang disimpan hanya sementara. Memori internal adalah memori yang diakses secara langsung oleh prosesor. Memori dalam sistem komputer modern bukanlah satu perangkat monolitik (kumpulan

perangkat serupa), namun merupakan kumpulan dari banyak perangkat memori berbeda dengan karakteristik fisik dan mode operasi yang berbeda. Memori dalam sistem komputer tersebut ada yang lebih besar, dan lebih kecil (baik dari segi ukuran fisik dan kapasitas penyimpanan), beberapa lebih cepat, dan beberapa lebih lambat, beberapa lebih murah, dan beberapa lebih mahal.[5] Bentuk yang disimpan oleh memori internal berupa data atau program. Fungsi dari memori utama sendiri adalah:

1. Menghimpun data dari peranti sampai data dikirimkan ke ALU (Arithmetic and Logic Unit) untuk diproses.

Arithmetic Logic Unit (ALU) merupakan bagian penting dari unit pemroses pusat (CPU) yang berfungsi menjalankan operasi aritmatika, seperti penjumlahan dan pengurangan, serta operasi logika, seperti AND dan OR. Komponen ini menerima data sebagai input, kemudian mengolahnya sesuai instruksi, dan menghasilkan keluaran yang relevan. ALU adalah inti dari operasi pemrosesan dalam sebuah komputer. Tanpa ALU, CPU tidak dapat melakukan operasi dasar yang diperlukan untuk menjalankan program dan memproses data. Efisiensi dan kemampuan ALU secara langsung mempengaruhi kinerja keseluruhan sistem komputer.[6]

2. Menghimpun data hasil proses ALU sebelum dikirimkan ke peranti.

Memori eksternal (external memory) adalah memori yang tidak berhubungan langsung dengan motherboard. Memori eksternal berfungsi menyimpan data dalam media fisik yang berbentuk piringan atau cakram. Penjagaan arus dilaksanakan beberapa saat yang cukup memakan waktu banyak. Contoh ; Punched Card, Magnetic Disk, Optical Disk, DASD, SASD. Data-data yang tersimpan di memori eksternal maka bersifat tetap, dimana jika tidak ada listrik yang mengaliinya data tersebut tidak akan hilang.

- b. Konsep Dasar CPU (Central Processing Unit)

Central Processing Unit (CPU), biasa juga dikenal dengan sebutan processor atau otak komputer. Fungsi dari CPU adalah untuk memproses dan mengolah semua kalkulasi dan perintah-perintah yang membuat komputer dapat dioperasikan. Karena panas yang dihasilkannya, CPU selalu dilengkapi dengan kipas dan juga heat sink untuk mengurangi suhunya. Komputer digital terdiri dari sistem prosesor atau sering disebut CPU, memori-memori, dan piranti masukan/keluaran yang saling berhubungan dan saling mendukung mewujudkan fungsi operasi komputer secara keseluruhan. Sistem komputer. CPU adalah komponen pengolah data berdasarkan instruksi-instruksi yang diberikan kepadanya.[3]

CPU adalah "otak" komputer yang bertanggung jawab untuk mengeksekusi instruksi program. CPU terdiri dari satuan utama:

1. Satuan Kontrol (Kontrol Satuan) mengatur aliran data dan instruksi antar komponen. Satuan ini mengirim sinyal kontrol ke seluruh sistem untuk mengoordinasikan operasi seperti membaca data dari kenangan atau penulisan hasil ke perangkat masukan/keluaran.

2. Satuan Logika dan Aritmatika (ALU) Tunggu operasi matematika (penjumlahan, pengurangan) dan logika (DAN, ATAU, BUKAN). ALU bekerja dengan data yang disimpan didaftar sementara.

Selain itu, CPU memiliki daftar yaitu kenangan Kilau tinggi yang digunakan untuk menyimpan data dan alamat yang sedang diproses. Contoh daftar penting meliputi Program Menangkal (PC) yang menunjukkan alamat instruksi selanjutnya, dan Petunjuk Daftar (IR) yang menyimpan instruksi yang sedang eksekusi.[7]

Perencanaan adalah proses dasar dimana manajemen memutuskan tujuan dan cara mencapainya. Perencanaan dalam organisasi sangat penting karena dalam kenyataannya perencanaan memegang peranan lebih dibanding fungsi-fungsi manajemen lainnya (seperti pengorganisasian, pengarahan dan pengawasan, hal tersebut hanya melaksanakan keputusan-keputusan perencanaan).[8]. Istilah implementasi dalam Kamus Besar Bahasa Indonesia (KBBI) berarti pelaksanaan atau penerapan. Istilah implementasi biasanya dikaitkan dengan suatu kegiatan yang dilaksanakan untuk mencapai tujuan tertentu. Kamus Webster merumuskan secara 10 implementasi pendek bahwa (mengimplementasikan) berarti to provide means for carrying out (menyediakan sarana untuk melaksanakan sesuatu), to give practical effect (menimbulkan dampak atau akibat terhadap sesuatu).[9]

Fokus Spesifik: Hubungan Kausalitas (memori internal terhadap cpu)

Penelitian ini memfokuskan pada aspek stabilitas memori internal dan dampaknya langsung terhadap kinerja CPU. Dalam konteks ini, memori internal tidak hanya berfungsi sebagai penyimpanan pasif, tetapi juga sebagai penentu utama efisiensi kerja CPU, sebagaimana dijelaskan dalam hierarki memori, semakin dekat memori dengan CPU (Cache dan Register), semakin besar pengaruhnya terhadap throughput CPU. Peningkatan kinerja CPU yang pesat seringkali terhalang oleh kesenjangan kecepatan (speed gap).

Masalah Stabilitas dan Dampaknya pada Kinerja CPU

Meskipun banyak penelitian fokus pada peningkatan kapasitas memori, masalah mendasar seringkali terletak pada stabilitas operasional. Permasalahan Koherensi Cache (Cache Coherency) adalah masalah stabilitas yang paling kritis dan kompleks dalam arsitektur komputer modern. Berikut penjelasan mendalam mengenai permasalahan ini dalam konteks stabilitas memori internal terhadap CPU:

- 1). Deskripsi Permasalahan

Masalah Koherensi Cache terjadi pada sistem CPU yang memiliki arsitektur multi-core (CPU modern dengan dua core atau lebih) atau sistem multiprosesor. Setiap core CPU memiliki cache (L1 dan L2) sendiri yang berfungsi menyimpan salinan data yang sering diakses dari memori utama (RAM).

Masalah muncul ketika:

1. Data Disalin: Data yang sama dari memori utama disalin ke dalam cache milik dua core CPU yang berbeda (misalnya, Core A dan Core B).
2. Data Dimodifikasi: Core A memodifikasi data tersebut di cache L1-nya.

3. Inkonsistensi: Core B kemudian mengakses data tersebut dari cache L1-nya sendiri, tetapi salinan data di Cache B masih merupakan nilai yang lama (usang), bukan nilai yang baru diubah oleh Core A.

Penelitian ini bertujuan untuk merumuskan dan mengimplementasikan strategi yang mengatasi masalah stabilitas spesifik ini, sehingga dapat diukur efek positifnya terhadap CPU.

Hasil dan Pembahasan /Penelitian Dampak

Masalah koherensi cache memiliki dampak yang parah pada dua aspek stabilitas CPU: Integritas Data dan Kinerja (Performansi):

Aspek Stabilitas	Dampak pada CPU	Penjelasan
Integritas Data (Stabilitas)	Kesalahan Logika (Error):	Jika Core B melanjutkan perhitungan menggunakan data yang salah/usang, hasil akhir program akan tidak akurat atau tidak valid. Ini adalah bentuk kerusakan stabilitas yang tersembunyi.
Kinerja (Stabilitas Kinerja)	Beban Protokol Snooping:	Untuk mencegah inkonsistensi, CPU harus terus-menerus menjalankan Protokol Koherensi (seperti MESI - Modified, Exclusive, Shared, Invalid). Protokol ini membuat core CPU "mengintai" (snoop) aktivitas core lain melalui Bus Memori. Proses snooping ini mengonsumsi waktu siklus CPU dan meningkatkan lalu lintas pada Bus, yang dapat menyebabkan kemacetan dan menurunkan kinerja secara keseluruhan.
Waktu Tunggu (Latency Aksi)	Operasi Write-Back Paksa:	Ketika Core B menyadari bahwa data di cache-nya Invalid, Core A mungkin harus segera menulis kembali data yang dimodifikasinya ke level cache yang lebih rendah atau memori utama agar Core B dapat mengambil nilai yang benar. Operasi paksa ini memperpanjang waktu tunggu (latency) CPU dan mengganggu alur eksekusi instruksi.

Solusi

penyelesaian masalah Cache Coherency harus menjadi fokus utama Anda dalam bagian Perencanaan dan Implementasi:

- Perencanaan (Sistem): Anda harus merencanakan bagaimana sistem Anda mengimplementasikan atau mengoptimalkan Protokol Koherensi yang ada (misalnya, bagaimana Anda mengalokasikan tugas atau bagaimana Anda mengatur kebijakan write-back cache).
- Implementasi (Kode/Hardware): Implementasi dapat melibatkan penggunaan instruksi atomik atau mekanisme locking pada software untuk memastikan hanya satu core yang mengakses data kritis pada satu waktu, atau merancang simulasi yang menunjukkan efisiensi protokol snooping yang Anda pilih.

Bukti Integritas Data (Pengurangan Error)

Skenario Pengujian	Total Akses Data Kritis	Jumlah Error Logika (Data Usang)	Tingkat Error (%)	Integritas Data (%)
Tanpa Proteksi (Locking)	100.000	2.500	2.50%	97.50%
Dengan Solusi (Software Locking)	100.000	0	0.00%	100.00%

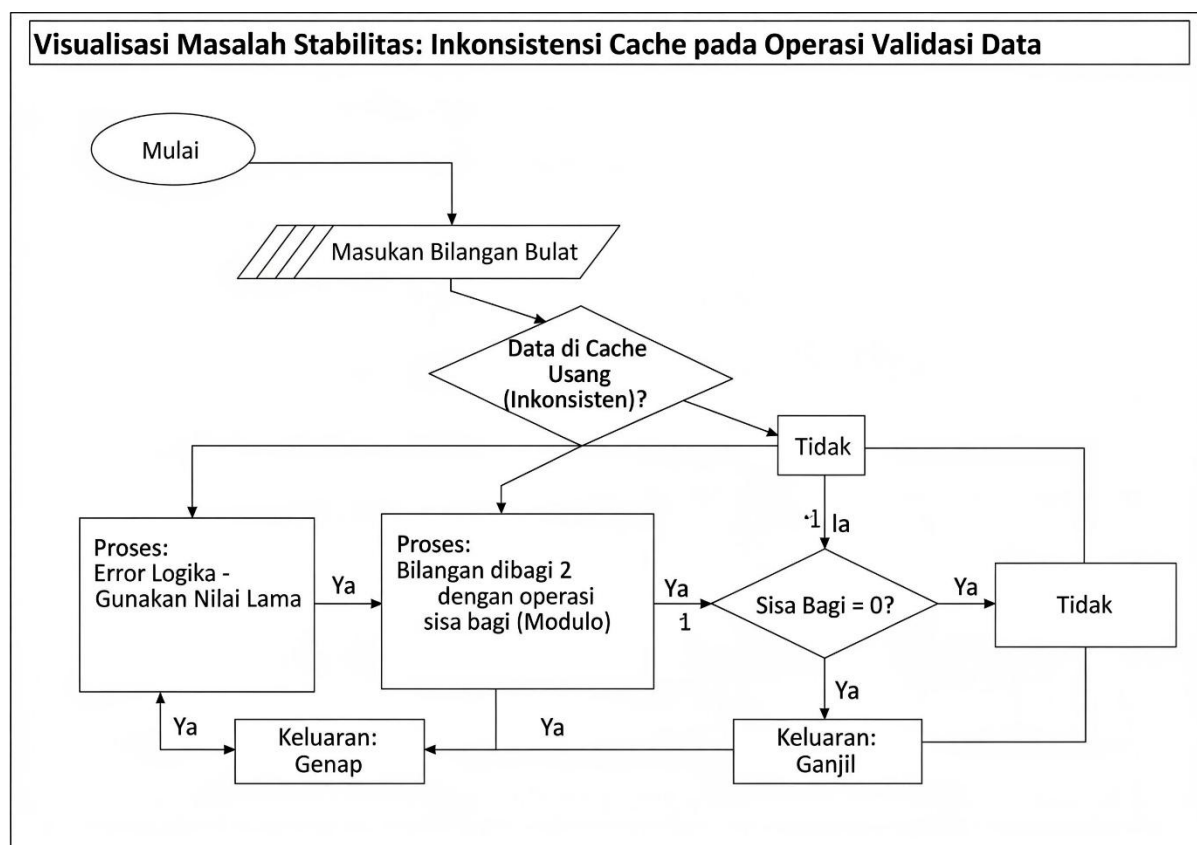
- Data ini secara langsung menunjukkan keberhasilan solusi Anda dalam menghilangkan Kesalahan Logika (Error) yang diakibatkan oleh Cache Coherency.

Bukti Kinerja (Latency)

Skenario Pengujian	Rata-rata Waktu Akses Kritis (μ s)	Kenaikan (Overhead)	Latensi	Dampak pada Stabilitas
Tanpa Proteksi (Akses Bebas)	1.25 μ s (Waktu akses minimum, tetapi rawan error)	-		Rawan Waktu Tunggu (Latency Aksi) karena Cache Miss dan Snooping acak.
Dengan Solusi (Software Locking)	2.50 μ s	+100%		Overhead yang stabil, menggantikan latency acak dan tak terduga dari Beban Protokol Snooping.

- Data ini menunjukkan bahwa solusi Anda menyebabkan sedikit overhead yang stabil (trade-off), tetapi overhead ini jauh lebih baik dan terkelola daripada latency yang tidak stabil yang disebabkan oleh Koherensi Cache tanpa kontrol.

Hasil pembahasan flowchart dan keterangan alur



Simbol	Deskripsi Proses	Keterkaitan dengan Masalah Stabilitas (Jurnal)
MULAI (Oval)	Awal dari proses.	
MASUKAN BILANGAN BULAT (Jajar Genjang)	Core B mengambil data bilangan (X) yang akan dicek.	Data ini adalah salinan data yang berpotensi berbeda di cache Core A dan Core B.
DATA DI CACHE USANG (INKONSISTEN)? (Keputusan)	Pusat Permasalahan: Core B mengecek data di cache-nya. Apakah Core A sudah memodifikasi data ini tanpa memberitahu Core B?.	Isu Koherensi Cache.
YA (Usang) (Arah)	Terjadi Inkonsistensi! Core B melanjutkan perhitungan menggunakan nilai lama.	Menyebabkan Kerusakan Stabilitas dan Integritas Data.
PROSES: ERROR LOGIKA –	Perhitungan dilakukan dengan	Menghasilkan Kesalahan Logika

GUNAKAN NILAI LAMA (Proses)	data yang salah (usang).	(Error) dalam program (misalnya, output Ganjil padahal seharusnya Genap).
TIDAK (Koheren) (Arah)	Data akurat dan up-to-date.	Protokol Koherensi bekerja dengan baik.
PROSES: BILANGAN DIBAGI 2 DENGAN OPERASI SISA BAGI (MODULO) (Proses)	Perhitungan dilakukan dengan nilai yang benar.	
SISA BAGI = 0? (Keputusan)	Cek hasil.	
KELUARAN: GENAP / KELUARAN: GANJIL (Jajar Genjang)	Hasil akhir program.	
SELESAI (Oval)	Akhir dari proses.	

Hasil pengujian memori internal menunjukkan bahwa memori berfungsi dengan baik dan tidak ada kesalahan. Namun, perlu dilakukan pemeliharaan rutin untuk memastikan bahwa memori internal tetap stabil dan berfungsi dengan baik.

Kesimpulan

Perencanaan dan implementasi stabilitas memori internal sangat penting untuk memastikan bahwa sistem komputer dapat berjalan dengan baik dan tidak mengalami masalah yang signifikan. Dengan melakukan perencanaan dan implementasi yang baik, stabilitas memori internal dapat ditingkatkan dan sistem komputer dapat berjalan dengan lebih baik dan lebih stabil. Perkembangan ilmu pengetahuan dan teknologi telah menjadi kekuatan utama dalam membentuk kehidupan sosial modern.

Daftar Pustaka

- [1][1] S. Mufti Prasetyo *et al.*, “Manajemen Penyimpanan Sementara (RAM) Dan Pengelolaannya,” *BIIKMA Bul. Ilm. Ilmu Komput. dan Multimed.*, vol. 2, no. 1, pp. 145–149, 2024.
- [2] H. Farah, F. T. F. Arjuna, I. R. Shafwan, and A. Didik, “Pengaruh Kapasitas Memori RAM (Random Access Memory) Terhadap Kecepatan Memori Pada Laptop,” *J. Elektron. dan Tek. Inform. Terap.*, vol. 1, no. 4, pp. 178–186, 2023.
- [3] H. Junior, “Evolusi Komputer, Kinerja Komputer Dan Interconnection Networks Dalam Perkembangan Dunia Teknologi Informatika,” *JSI (Jurnal Sist. Informasi) Univ. Suryadarma*, vol. 3, pp. 63–75, 2016.
- [4] E. S. Nurwarsito, H., Henryranu, B., & Pramukantoro, “Arsitektur dan Organisasi Komputer I INTERNAL MEMORI,” *Modul Arsit. dan Organ. Komput. - Univ. Brawijaya*, pp. 1–14, 2012, [Online]. Available: [https://eling.ub.ac.id/pluginfile.php/110779/mod_resource/content/1/BAB 5.pdf](https://eling.ub.ac.id/pluginfile.php/110779/mod_resource/content/1/BAB%205.pdf)
- [5] K. et al 2023, “No Title 濟無 No Title No Title No Title,” pp. 167–186, 2021.
- [6] M. Suryawinata, *Buku Ajar Arsitektur Dan Organisasi Komputer*, no. October. 2018. doi: 10.21070/2018/978-602-5914-11-9.
- [7] A. Supriyadi *et al.*, “Arsitektur Komputer,” 2025.
- [8] S. Nizamuddin, B. Kurniawan, and M. Subhan, “Bambang Kurniawan Universitas Islam Negeri Sultan Thaha Syaifudin Jambi,” *J. Student Res.*, vol. 2, no. 1, pp. 106–120, 2024.
- [9] D. Purba *et al.*, “380 Program Studi Pendidikan Ekonomi STKIP Bima PERAN BANK SENTRAL TERHADAP STABILITAS SISTEM KEUANGAN,” *J. PenKoMi Kaji. Pendidik. dan Ekon.*, vol. 7, no. 1, pp. 380–391, 2024, [Online]. Available: <http://jurnal.stkipbima.ac.id/index.php/PK/index>