Jurnal Sistem Informasi dan Ilmu Komputer
Volume. 3 Nomor. 1 Tahun 2025
e-ISSN: 2986-4976; dan p-ISSN: 2986-5158, Hal 115-128
DOI: https://doi.org/10.59581/jusiik-widyakarya.v3i1.4471
Available online: https://ifrelresearch.org/index.php/jusiik-widyakarya

Perbandingan Manajemen Memori pada Sistem Operasi *Windows* dan *Linux*

Rini Ariza[1*], Muhammad Reza Maulana[2], Elkin Rilvani[3]
[1,2,3]Universitas Pelita Bangsa, Indonesia
Email: [1*]riniarizaaa@gmail.com, [2]rezamaulana9569@gmail.com, [3]elkin.rilvani@pelitabangsa.ac.id

Alamat: Jl. Inspeksi Kalimalang No.9, Cibatu, Cikarang Sel., Kabupaten Bekasi, Jawa Barat 17530
Korespondensi penulis: riniarizaaa@gmail.com

Abstract*. This research compares memory management between Windows 10 and Linux Ubuntu operating systems, focusing on the issues of memory fragmentation and memory leaks. Windows 10 applies paging and segmentation methods, which provide a high degree of flexibility, but on the other hand carry considerable system overhead. On the other hand, Linux Ubuntu uses the Buddies system which is more efficient in dealing with fragmentation issues. In handling memory leaks, Windows 10 relies on Garbage Collection, while Linux Ubuntu utilizes the Permchecker tool. This research provides knowledge related to the differences between Windows 10 and Linux Ubuntu operating systems in the context of handling memory management problems, especially on memory fragmentation and memory leak problems.*

Keywords*: fragmentation, linux ubuntu, memory, leak, windows 10*

Abstrak. Penelitian ini membandingkan manajemen memori antara sistem operasi Windows 10 dan Linux Ubuntu, dengan fokus pada masalah fragmentasi memori serta kebocoran memori (memory leak). Windows 10 menerapkan metode paging dan segmentation, yang memberikan tingkat fleksibilitas tinggi, tetapi di sisi lain membawa overhead sistem yang cukup besar. Di sisi lain, Linux Ubuntu menggunakan sistem Buddies yang lebih efisien dalam mengatasi masalah fragmentasi. Dalam menangani kebocoran memori, Windows 10 mengandalkan Garbage Collection, sementara Linux Ubuntu memanfaatkan tools Permchecker. Penelitian ini memberikan pengetahuan terkait perbedaan sistem operasi windows 10 dan linux ubuntu dalam konteks penanganan masalah manajemen memori khusunya pada masalah fragmentasi memori dan memory leak.

Kata kunci: fragmentasi, linux ubuntu, , memory, leak, windows 10

1. LATAR BELAKANG

Sistem operasi adalah bagian penting dari pengolah peranti lunak dasar yang berfungsi sebagai pengelola sumber daya perangkat keras komputer (hardware) dan menyediakan layanan umum untuk aplikasi perangkat lunak (Wahid, 2019). Sistem operasi dalam sebuah hardware memiliki peranan yang sangat penting, jika tidak ada maka perangkat komputer tidak dapat menjalankan fungsi-fungsinya dengan baik. Banyak sekali jenis-jenis sistem operasi yang biasa digunakan dalam perangkat komputer, beberapa diantaranya yaitu windows, linux dan mac os.

Dalam sistem operasi terdapat beberapa komponen yang mendukung jalannya sebuah sistem operasi, salah satunya adalah manajamen memori. Manajemen memori ini mencakup penempatan, penggunaan, dan pembebasan memori untuk melakukan berbagai tugas

Received: Oktober 15, 2024; Revised: November 17, 2024 Accepted: Desember 20, 2024;
Online Available: Desember 23, 2024

komputasi (Angella Harsono et al., 2022). Dalam penelitian ini windows dan linux merupakan sistem operasi yang paling banyak digunakan dari berbagai faktor seperti kemudahan akses, kepraktisan, kestabilan, dan keamanan (Maolia & Aprilliana, 2024). Maka dari itu dalam penelitian ini akan melakukan komparasi pada sistem operasi windwos dan linux. Dalam cakupan seperti itu, sangat penting untuk memeriksa cara sistem operasi Windows dan Linux mengelola memori.

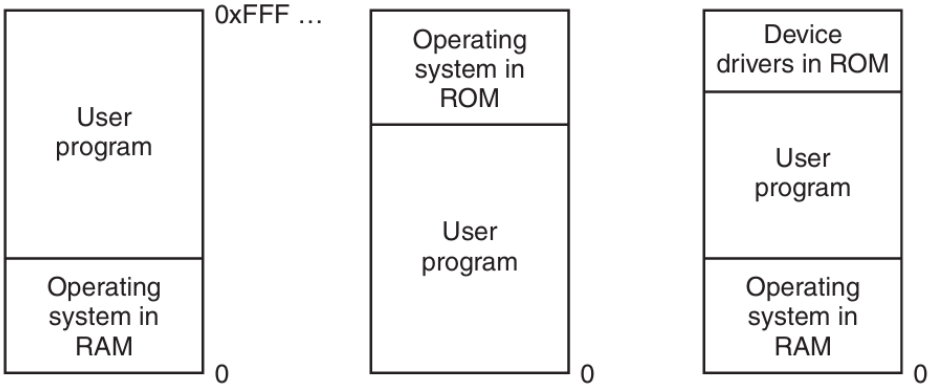
Perbedaan teknik manajemen memori antara Windows dan Linux dapat menyebabkan masalah yang signifikan. Fragmentasi memori eksternal dan internal adalah salah satu masalah utama yang dihadapi. Fragmentasi eksternal terjadi ketika tidak ada cukup memori bebas yang berdekatan untuk memenuhi permintaan. Ketika alokasi yang berdekatan dengan ukuran yang berbeda memiliki masa pakai yang berbeda, memori bebas akan bercampur dengan memori yang dialokasikan (Mansi & Swift, 2024). Sedangkan fragmentasi internal adalah sebaliknya yaitu ketika blok memori yang dialokasikan lebih besar dari yang diperlukan, akan mengakibatkan terbuangnya ruang dalam blok tersebut. Maka dari itu Fragmentasi memori internal akan terjadi ketika aplikasi memerlukan memori yang bervariasi . Selain itu memory leak atau kebocoran memori juga dapat menimbulkan penurunan performa sistem. Kebocoran memori terjadi saat aplikasi mengalokasikan memori tetapi gagal melepaskannya setelah tidak lagi digunakan, yang menyebabkan peningkatan berkelanjutan dalam konsumsi memori dari waktu ke waktu. Maka pertanyaan penelitian kali ini yang dilakukan untuk menyelesaikan masalah diatas, Bagaimana mengetahui kinerja memori dari sistem operasi windows dan linux dalam menangani fragmentasi memori dan memory leak?. Pada penelitian kali ini dibatasi pada sistem operasi windows 10 dan linux ubuntu.

Tujuan dari penelitian ini adalah untuk mengetahui berbagai aspek yg harus diperbaiki dan membandingkan metode manajemen memori di Windows dan Linux. Terlebih lagi, studi ini juga dimaksudkan untuk memberikan saran berdasarkan data kepada pengguna mengenai sistem operasi yang sesuai dengan kebutuhan mereka.

Manfaat yang didapat dari penelitian ini adalah memberikan pedoman yang jelas bagi pengguna dalam memilih sistem operasi yang sesuai, berdasarkan analisis menyeluruh tentang manajemen memori. Melalui hasil penelitian ini, diharapkan pengguna mampu memahami kelebihan dan kekurangan sistem operasi dalam manajemen memori, sehingga dapat mengambil keputusan yang lebih efektif sesuai kebutuhan mereka.

2. KAJIAN TEORITIS

Teori dasar dari manajemen memori yaitu proses penting yang terdapat dalam sistem operasi untuk mengalokasikan dan mengelola ruang memori secara efisien pada proses yang berjalan (Tanenbaum & Bos, n.d.). Berikut gambaran umum terkait manajemen memori dapat dilihat pada Gambar 1:



Gambar 1. Gambaran umum manajemen memori

Maka dari itu sangat penting untuk mengetahui dan membandingkan bagaimana proses tersebut berjalan dalam sistem operasi windows dan linux. Dalam penelitian ini berfokus pada masalah fragmentasi memori dan memory leak atau kebocoran memori. Fragmentasi memori mengacu pada penggunaan ruang memori yang tidak efisien yang terjadi ketika memori bebas dibagi menjadi blok-blok kecil yang tidak bersebelahan, sehingga sulit untuk mengalokasikan segmen memori bersebelahan yang lebih besar, sedangkan kebocoran memori (memory leak) adalah satu isu besar dalam pengelolaan memori, terutama pada bahasa pemrograman yang memerlukan pengaturan alokasi dan pembebasan memori secara manual, seperti C/C++. Kebocoran memori bisa berdampak besar pada kinerja sistem, terutama dalam kondisi terbatasnya memori seperti pada sistem tertanam dan server, yang dapat mengakibatkan penolakan layanan (Lamprakos et al., 2023).

3. METODE PENELITIAN

Pada penelitian ini menggunakan metode komparatif dan studi literatur untuk mengetahui teknik manajemen memori antara sistem operasi Windows dan Linux, dengan fokus utama pada masalah fragmentasi memori (internal dan eksternal) dan kebocoran memori. Penelitian ini tidak melakukan pengujian eksperimental secara langsung namun membandingkan hasil dari berbagai sumber teoritis dan penelitian terkait untuk mendapatkan pemahaman lebih dalam mengenai perbedaan kedua sistem operasi dalam manajemen memori.

Langkah pertama dari metode ini adalah studi literatur, dimana informasi dan data terkait manajemen memori, khususnya masalah fragmentasi memori dan kebocoran memori, dikumpulkan dari banyak sumber penelitian teoritis dan penelitian sebelumnya. Studi literatur ini akan mencakup informasi teknis tentang bagaimana kedua sistem operasi menangani alokasi dan penggunaan memori serta bagaimana mereka menangani tantangan seperti fragmentasi internal, fragmentasi eksternal, dan kebocoran memori.

Tahap kedua penelitian ini adalah perbandingan berdasarkan hasil studi literatur. Perbandingan ini akan dilakukan dengan menganalisis perbedaan utama antara teknik manajemen memori Windows dan Linux yang mencakup penanganan fragmentasi memori, penanganan kebocoran memori (memory leak).

Setelah melakukan perbandingan, data dari literatur dan penelitian sebelumnya akan dianalisis secara mendalam untuk mengidentifikasi perbedaan yang signifikan antara kedua sistem operasi untuk mengatasi masalah fragmentasi memori dan kebocoran memori.

4. HASIL DAN PEMBAHASAN

Pada bab ini akan disajikan hasil dari penelitian terkait perbandingan teknik manajemen memori pada sistem operasi Windows 10 dan Linux Ubuntu, dengan fokus pada dua aspek utama yaitu fragmentasi memori (internal dan eksternal) serta kebocoran memori (memory leak) beserta dengan pembahasan dari hasil tersebut.

4.1. Fragmentasi Memori

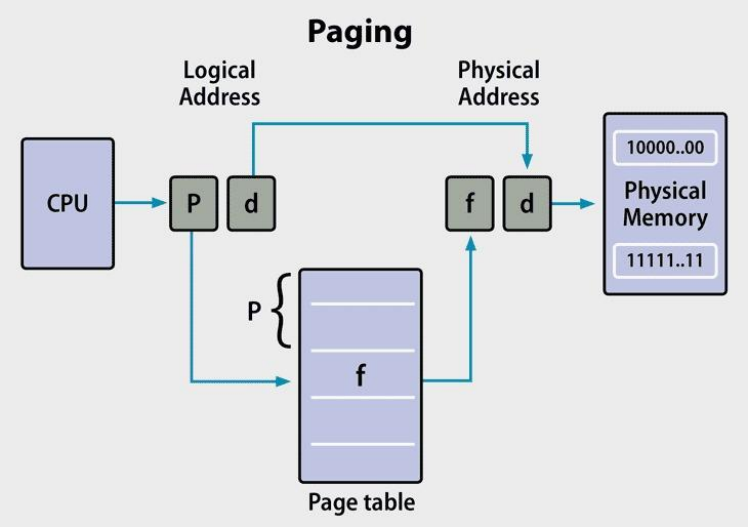
4.1.1. Fragmentasi Memori Pada Windows

Fragmentasi memori terjadi pada windows 10 akibat fragmentasi eksternal, dimana memori bebas dibagi menjadi blok-blok kecil yang tidak bersebelahan karena ukuran alokasi yang berbeda. Windows 10 menggunakan paging dan segmentation untuk mengelola alokasi memori. Teknik ini memungkinkan sistem operasi menggunakan memori yang tersedia secara efisien, sehingga mengurangi dampak fragmentasi (Mansi & Swift, 2024).

1. Paging

Paging di Windows 10 mengacu pada skema manajemen memori yang memungkinkan sistem operasi mengambil data dari memori sekunder dalam bentuk blok-blok atau halaman daripada memuat seluruh program ke dalam RAM. Metode ini membantu mengurangi fragmentasi eksternal dengan memungkinkan sistem untuk mengalokasikan memori secara lebih fleksibel, hal ini terjadi karena halaman dapat didistribusikan di berbagai lokasi memori fisik tanpa memperhatikan kedekatan (Mansi & Swift, 2024). Hal ini memungkinkan alokasi memori yang tidak berdekatan, sehingga mengurangi fragmentasi dan mengoptimalkan penggunaan memori (Bender et al., 2021; Silberschatz et al., n.d.). Berikut contoh proses paging pada windows 10:

Dapat dilihat pada Gambar 2 bahwa paging melakukan proses di mana CPU menggunakan alamat logis untuk mengakses data di memori fisik. Alamat logis terdiri dari page number (P) dan offset (d), yang harus diterjemahkan ke dalam alamat fisik menggunakan page table. Page table berperan sebagai peta untuk menerjemahkan alamat logis ke alamat fisik di memori. Dengan paging, penggunaan memori menjadi lebih efisien karena data dapat disimpan di lokasi yang tidak berurutan secara fisik. Proses paging dilakukan secara cepat dan transparan, memastikan program berjalan tanpa gangguan. CPU menggunakan alamat fisik tersebut untuk mengakses data yang diperlukan dan melanjutkan pemrosesan dengan lancar.



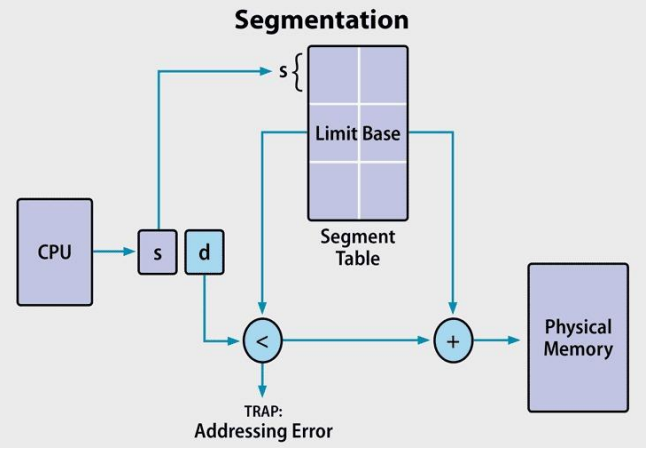
Gambar 2. Paging

2. Segmentation

Segmentasi dalam Windows 10 adalah pembagian data atau proses ke dalam segmen-segmen yang berbeda untuk peningkatan manajemen, keamanan, dan analisis. Konsep ini dapat digunakan dalam berbagai konteks, termasuk visualisasi data, arsitektur keamanan, dan manajemen kerentanan. Segmentasi juga melibatkan pembagian memori menjadi segmen-segmen berukuran variabel berdasarkan kriteria logis. Pendekatan ini dapat mengalokasikan memori dengan lebih fleksibel, di mana segmen dapat berkembang atau menyusut sesuai kebutuhan, sehingga dapat mengurangi fragmentasi internal (Baharom et al., 2021; Jalaman & Teleron, 2024). Berikut contoh proses segmentation pada windows 10:

Dapat dilihat pada Gambar 3 proses segmentation dimulai ketika CPU diminta mengakses memori menggunakan alamat logis yang diterima, terbagi menjadi nomor segmen dan offset. Langkah pertama adalah mencari informasi segmen dalam Segment Table, yang berisi data penting terkait setiap segmen seperti Base (lokasi awal segmen) dan Limit (ukuran maksimum segmen). Sistem kemudian memverifikasi apakah offset berada dalam batas yang valid dengan membandingkannya dengan Limit. Jika tidak valid, sistem akan memicu TRAP untuk menghentikan proses. Jika valid, sistem akan menghitung alamat fisik dengan menambahkan Base dan offset, untuk mengakses data dalam memori fisik dengan aman dan sesuai batas.

4.1.2. Fragmentasi Memori Pada Linux



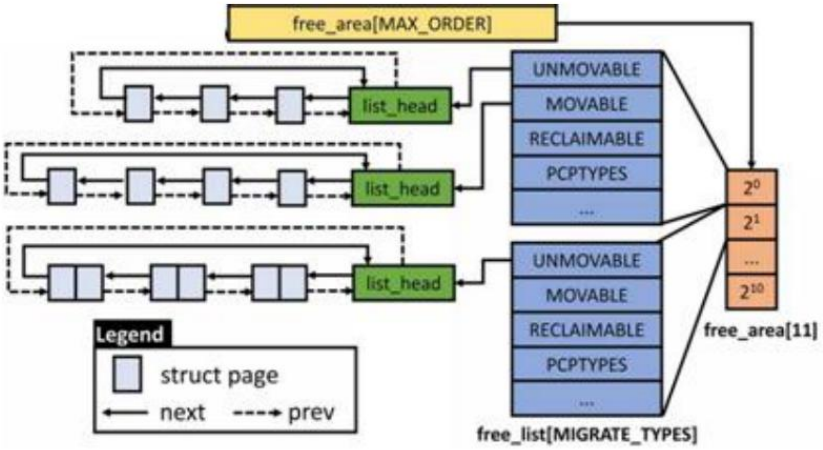
Gambar 3. Segmentation

Fragmentasi memori pada Linux, khususnya Ubuntu, terjadi akibat alokasi dan dealokasi memori secara dinamis. Hal ini menciptakan wilayah memori bebas yang tidak berdekatan, berdampak pada kinerja sistem. Penyelesaiannya menggunakan metode Buddy System.

Metode ini secara dinamis menyesuaikan organisasi memori sesuai dengan pola permintaan yang ada dan mengurangi masalah fragmentasi yang sering terjadi (Cronburg & Guyer, 2021).

- Buddy System

  Buddy System merupakan suatu teknik manajemen memori yang diterapkan di Linux, bertujuan untuk mengalokasikan dan melepaskan blok memori, sambil meminimalkan terjadinya fragmentasi. Sistem ini bekerja dengan cara membagi memori menjadi partisi-partisi, sehingga dapat membuat blok dengan berbagai ukuran yang memungkinkan proses alokasi dan deallokasi terjadi dengan cepat. Adaptasi ini meningkatkan konkurensi dalam proses alokasi dan dealokasi memori, serta secara signifikan mengurangi latensi (71,47% untuk alokasi dan 93,20% untuk dealokasi) dengan menyesuaikan organisasi memori secara dinamis sesuai dengan pola penggunaan (Lu et al., 2021). Berikut contoh proses Buddy System pada linux ubuntu:



Gambar 3. Buddy System

Dapat dilihat pada Gambar 4 bahwa teknik buddy system mengatur memori dalam bentuk blok-blok dengan ukuran yang berlipat ganda, dikelola melalui array free_area. Setiap blok disimpan dalam rak yang sesuai ukurannya, diatur melalui linked list dalam struct page. Blok memori dikelompokkan berdasarkan jenis penggunaan, seperti UNMOVABLE, MOVABLE, RECLAIMABLE, dan PCPTYPES. Saat ada permintaan memori, sistem mencari blok paling cocok, memecah blok besar menjadi buddy jika perlu. Ketika blok tidak terpakai, sistem akan memeriksa buddy, jika tersedia, keduanya digabungkan kembali. Proses ini menjaga agar memori tidak terfragmentasi dan selalu optimal. Approach Buddy System memastikan pengelolaan memori yang efisien, dengan

menjaga keseimbangan alokasi dan penghematan ruang serta memudahkan penggabungan blok yang terfragmentasi.

4.1.3. Analisis Perbandingan

Pada bagian ini akan dijelaskan perbandingan teknik manajemen memori antara sistem operasi windows 10 dan linux ubuntu dalam mengatasi permasalahan fragmentasi memori. Untuk menganalisis teknik penanganan fragmentasi memori di Windows 10 dan Linux Ubuntu, peneliti dapat mengeksplorasi pendekatan yang diambil oleh masing-masing sistem operasi dalam upaya mengurangi dampak fragmentasi serta meningkatkan efisiensi manajemen memori. Berikut ini adalah analisis perbedaan antara keduanya dalam bentuk tabel perbandingan:

Tabel 1. Perbandingan Metode Penanganan Fragmentasi Memori

OS	**Windows 10**	**Linux Ubuntu**
Metode	Paging dan Segmentation	Buddy System
Cara Kerja	- Paging: Membagi memori menjadi halaman kecil untuk fleksibilitas alokasi. - Segmentation: Membagi proses/data ke dalam segmen logis untuk efisiensi.	- Membagi memori menjadi blok dengan ukuran berlipat ganda (2^0, 2^1, dll.). - Menggabungkan blok (buddy) saat tidak digunakan untuk mengurangi fragmentasi.
Fleksibilitas Alokasi	Tinggi: Halaman atau segmen dapat dialokasikan di lokasi fisik terpisah.	Sedang: Blok memori dialokasikan dalam ukuran tertentu (berlipat ganda).
Efisiensi Fragmentasi	- Paging: Mengurangi fragmentasi eksternal. - Segmentation: Mengurangi fragmentasi internal dengan alokasi fleksibel.	Sangat efektif mengurangi fragmentasi eksternal dan internal.
Overhead Sistem	Tinggi: Memerlukan pengelolaan tabel seperti page table dan segment table.	Rendah: Algoritma Buddy System lebih sederhana dan cepat.
Kecepatan Alokasi	Sedang: Bergantung pada ukuran tabel dan proses pencarian.	Tinggi: Blok memori dikelola dalam array dengan latensi rendah.
Keamanan Data	Tinggi: Segmentation memastikan akses memori berada dalam batas yang sah.	Sedang: Fokus pada pengelolaan efisiensi memori, bukan keamanan data.
Kompleksitas Implementasi	Tinggi: Kombinasi paging dan segmentation membutuhkan arsitektur yang kompleks.	Rendah: Algoritma lebih mudah diterapkan dan dikelola.

Dapat dilihat pada Tabel 1 analisis perbandingan teknik penanganan fragmentasi memori antara Windows 10 dan Linux Ubuntu menunjukkan bahwa kedua sistem operasi memiliki kelebihan dan kekurangan yang sesuai dengan kebutuhan pengelolaan memori. Windows 10 memberikan tingkat fleksibilitas tinggi melalui teknik paging dan segmentation, namun juga memiliki overhead sistem yang tinggi. Di sisi lain, Linux Ubuntu menggunakan Buddy System yang efisien namun memiliki fleksibilitas yang lebih terbatas dibanding Windows. Meskipun Buddy System menawarkan efisiensi dalam alokasi dan dealokasi memori, serta overhead sistem yang lebih rendah, namun kontrol keamanan data tidak sekuat yang ditawarkan oleh segmentation di Windows.

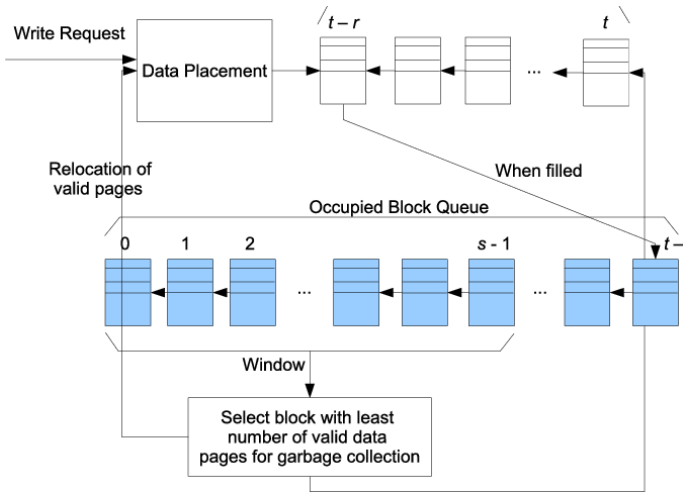
4.2. Memory Leak

4.2.1. Memory Leak Pada Windows

Kebocoran memori (memory leak) di sistem operasi Windows khususnya pada windows 10 terutama disebabkan oleh pengelolaan sumber daya memori yang kurang optimal sehingga dapat mengakibatkan penurunan efisiensi sumber daya dan performa. Objek memori juga sering tidak dialokasikan, terutama dalam aplikasi yang kompleks di mana perlu dilakukan manajemen secara manual (Gangwar & Katal, 2021). Biasanya pada windows 10 untuk mengatasi permasalahan memory leak menggunakan metode Garbage Collection. Teknik tersebut berfungsi untuk mengidentifikasi objek mana dalam memori yang masih sedang digunakan serta objek mana yang dapat direklamasi (Murali et al., 2024).

- Garbage Collection

  Garbage Collection dalam konteks sistem operasi Windows 10 merujuk pada proses otomatis yang bertujuan untuk mengambil kembali memori yang tidak lagi digunakan, hal ini bertujuan untuk mengoptimalkan kinerja sistem. Proses ini memiliki peranan yang sangat penting dalam pengelolaan memori yang efisien serta memastikan agar aplikasi dapat berjalan dengan lancar. Metode ini melibatkan proses penyalinan objek hidup ke dalam ruang memori baru yang berfungsi untuk membantu dalam memadatkan memori (Murali et al., 2024). Berikut ini contoh proses garbage collection pada windows 10:



Gambar 4. Garbage Collection

Dapat dilihat pada Gambar 5 Proses Garbage Collection dimulai dengan Write Request untuk menulis data baru ke dalam blok yang kosong. Data kemudian ditempatkan secara berurutan dalam blok-blok tersebut hingga terisi penuh. Blok yang penuh akan menjalani Relocation of Valid Pages untuk memindahkan data valid ke blok lain, sementara data tidak valid akan ditandai untuk dihapus. Blok-blok terisi masuk ke dalam Occupied Block Queue, kemudian dipilih berdasarkan jumlah halaman valid terendah melalui mekanisme window. Garbage Collection akan mengosongkan blok terpilih dengan menghapus data tidak valid, sehingga blok tersebut dapat digunakan kembali. Proses ini membantu mengoptimalkan ruang penyimpanan, menjaga performa dan mencegah kebocoran memori.

4.2.2. Memory Leak Pada Linux

Kebocoran memori dalam sistem operasi Linux khususnya pada linux ubuntu dapat muncul akibat berbagai teknik pemrograman dan batasan sistem yang ada. Kejadian ini terjadi ketika memori yang telah dialokasikan tidak dikelola dengan baik, sehingga menyebabkan pengurangan sumber daya seiring berjalannya waktu. Biasanya kode yang salah mengenai alokasi memori dinamis bisa menimbulkan kebocoran memori, terutama dalam aplikasi yang kompleks dan memiliki banyak fungsi (Sharan, 2014). Memahami faktor-faktor yang menyebabkan kebocoran memori sangat penting untuk memastikan kinerja dan stabilitas sistem tetap terjaga. Biasanya pada linux ubuntu untuk mengatasi permasalahan memory leak menggunakan tools dari linux itu sendiri yaitu permchecker. Alat ini menyediakan informasi yang mendetail mengenai karakteristik kesalahan yang teridentifikasi, sehingga memudahkan proses debugging dan pemecahan masalah (Fan, 2023).

- Permchecker

Permchecker adalah alat khusus yang diciptakan untuk mengatasi masalah kebocoran memori di Linux dengan memanfaatkan typestates. Pendekatan proaktif ini memungkinkan deteksi inkonsistensi memori selama waktu berjalan, mencegah kerusakan aplikasi. Permchecker menghubungkan typestate dengan alokasi memori, menjamin bahwa akses memori selaras dengan status yang diharapkan (Fan, 2023). Berikut ini contoh proses tools permchecker pada linux ubuntu:

Gambar 5. Proses Eksekusi Permchecker

Dapat dilihat pada Gambar 6 proses awal dimulai dengan Layout Spec, suatu spesifikasi yang menentukan tata letak sistem. Codegen tool kemudian menghasilkan Macros yang dapat memperluas kemampuan bahasa pemrograman seperti C dengan fitur tambahan preprocessing. Macros ini terhubung dengan Proxy Objects, Java FFI, dan Annotations.

- Proxy Objects sebagai jembatan antarbagian dalam sistem.
- Java FFI memungkinkan komunikasi antara kode C dan Java.
- Annotations memberi info metadata saat kompilasi.

Tracking Interface mengawasi jalannya program dengan masukan dari elemen sebelumnya. JIT Assembly Instrumentation melacak dinamis program dan optimalkan kode. DBI Engine sebagai mesin utama untuk menyadap fungsi-fungsi selama runtime dengan bantuan Shadow Memory dan Typestate Checker. Didukung oleh Library dan Pin-tool API untuk memperluas kemampuan. Sistem integrasi proses kompilasi dan runtime, dari membangun struktur program hingga pelacakan dan debugging real-time.

4.2.3. Analisis Perbandingan

Perbandingan teknik manajemen memori antara Windows 10 dan Linux Ubuntu dalam penanganan kebocoran memori, sistem operasi dapat dijelajahi untuk meningkatkan efisiensi

manajemen memori dan mengurangi dampak kebocoran memori. Berikut ini adalah analisis perbedaan antara keduanya dalam bentuk tabel perbandingan:

Tabel 2. Perbandingan Metode Penanganan Memory Leak

OS	**Windows 10**	**Linux Ubuntu**
Metode Deteksi	Otomatis melalui *Garbage Collection*, mendeteksi objek yang tidak digunakan untuk dibebaskan.	Proaktif melalui *Permchecker* dengan pelacakan mendalam berbasis *typestate*.
Pendekatan	Otomatis, minimal intervensi pengembang.	Manual dengan alat bantu untuk pengembang berpengalaman.
Proses Penanganan	Relokasi data valid ke blok baru, penghapusan data tidak valid untuk mengoptimalkan ruang memori.	Deteksi *runtime* menggunakan *Shadow Memory* dan validasi status dengan *Typestate Checker*.
Fleksibilitas	Kurang fleksibel untuk aplikasi non-.NET atau bahasa yang tidak mendukung *Garbage Collection*.	Sangat fleksibel, mendukung debugging di berbagai bahasa seperti C/C++ dan interoperabilitas.
Efisiensi Sistem	Efisiensi bergantung pada algoritma *Garbage Collection*; berpotensi mengalami fragmentasi memori.	Efisiensi tinggi dengan deteksi dini, minim fragmentasi memori melalui optimasi langsung.
Kegunaan	Cocok untuk aplikasi umum dan pengembang yang ingin solusi sederhana tanpa banyak konfigurasi.	Ideal untuk sistem kompleks, seperti server atau sistem tertanam yang memerlukan stabilitas.
Lingkungan Dukungan	Terintegrasi dalam ekosistem Windows dengan dukungan untuk aplikasi .NET.	Dapat digunakan pada aplikasi berbasis Linux dengan alat bantu debugging tingkat lanjut.

Dapa dilihat pada Tabel 2 bahwa windows 10 menggunakan Garbage Collection untuk mengelola memori yang tidak terpakai secara otomatis, terutama cocok untuk aplikasi. NET namun dapat menyebabkan fragmentasi memori. Di sisi lain, Linux Ubuntu menggunakan Permchecker untuk mendeteksi dan mengatasi kebocoran memori dengan lebih mendalam, cocok untuk aplikasi kritis dan sistem tertanam. Meskipun lebih teknis, Permchecker dapat mengurangi memory leak dan menjaga stabilitas dalam jangka panjang. Secara keseluruhan,

Windows 10 mudah digunakan untuk aplikasi umum, sementara Linux Ubuntu lebih fleksibel untuk debugging dan pengelolaan memori yang kompleks.

5. KESIMPULAN DAN SARAN

Penelitian ini membandingkan teknik manajemen memori antara Windows 10 dan Linux Ubuntu. Windows 10 menggunakan paging dan segmentation untuk mengatasi fragmentasi memori, namun memiliki overhead yang tinggi. Linux Ubuntu menggunakan Buddy System untuk alokasi memori yang efisien. Dalam mengelola memory leak, Windows 10 menggunakan Garbage Collection untuk mendeteksi memori yang tidak terpakai secara otomatis, sementara Linux Ubuntu menggunakan Permchecker yang proaktif namun memerlukan keahlian teknis lebih tinggi. Penelitian merekomendasikan pengujian eksperimental langsung pada kedua sistem operasi untuk perbandingan maksimal, serta mendorong pengembang untuk meningkatkan fleksibilitas dan efisiensi teknik manajemen memori mereka agar dapat lebih adaptif terhadap kebutuhan pengguna di masa depan.

DAFTAR REFERENSI

Angella Harsono, C., Faiza Shalekha, N., & Yudo Berliana, R. (2022). STRATEGI EFISIEN MANAJEMEN MEMORI UNTUK MENINGKATKAN KINERJA SISTEM OPERASI. SINTESIA: Jurnal Sistem Dan Teknologi Informasi Indonesia, 02(1), 12.

Baharom, S. H., Setapa, S., Ong, H. H., & Luke, J. Y. (2021). Segmentation system and method for virtualized environment.

Bender, M. A., Bhattacharjee, A., Conway, A., Farach-Colton, M., Johnson, R., Kannan, S., Kuszmaul, W., Mukherjee, N., Porter, D., Tagliavini, G., Vorobyeva, J., & West, E. (2021). Paging and the address-translation problem. Annual ACM Symposium on Parallelism in Algorithms and Architectures, 105–117. https://doi.org/10.1145/3409964.3461814

Cronburg, K., & Guyer, S. Z. (2021). Permchecker: A toolchain for debugging memory managers with typestate. Proceedings of the ACM on Programming Languages, 5(OOPSLA). https://doi.org/10.1145/3485526

Fan, Z. (2023). Research on using kernel dynamic tracking to locate system bug. https://doi.org/10.1117/12.2675203

Gangwar, Di. K., & Katal, A. (2021). Memory Leak Detection Tools: A Comparative Analysis. 2021 6th International Conference on Recent Trends on Electronics, Information, Communication and Technology, RTEICT 2021, 315–320. https://doi.org/10.1109/RTEICT52294.2021.9574012

Jalaman, J. R. C., & Teleron, J. I. (2024). Optimizing Operating System Performance through Advanced Memory Management Techniques: A Comprehensive Study and Implementation. Engineering and Technology Journal, 09(05). https://doi.org/10.47191/etj/v9i05.33

Lamprakos, C. P., Xydis, S., Catthoor, F., & Soudris, D. (2023). Viewing Allocators as Bin Packing Solvers Demystifies Fragmentation. http://arxiv.org/abs/2304.10862

Lu, Y., Liu, W., Wu, C., Wang, J., Gao, X., Li, J., & Guo, M. (2021). Spring Buddy: A Self-Adaptive Elastic Memory Management Scheme for Efficient Concurrent Allocation/Deallocation in Cloud Computing Systems. Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS, 2021-December. https://doi.org/10.1109/ICPADS53394.2021.00056

Mansi, M., & Swift, M. M. (2024). Characterizing Physical Memory Fragmentation. http://arxiv.org/abs/2401.03523

Maolia, S., & Aprilliana, R. E. (2024). Literature Review Jurnal Memahami Faktor-Faktor Yang Mempengaruhi Popularitas Windows Dibandingkan Linux. In JAMASTIKA (Vol. 3).

Murali, A., Alfadel, M., Nagappan, M., Xu, M., & Sun, C. (2024). AddressWatcher: Sanitizer-Based Localization of Memory Leak Fixes. IEEE Transactions on Software Engineering, 50(9), 2398–2411. https://doi.org/10.1109/TSE.2024.3438119

Sharan, K. (2014). Garbage Collection. In Beginning Java 8 Language Features: Lambda Expressions, Inner Classes, Threads, I/O, Collections, and Streams (pp. 485–517). Apress. https://doi.org/10.1007/978-1-4302-6659-4_11

Silberschatz, A., Galvin, P. B., & Gagne, G. (n.d.). OPERATING SYSTEM CONCEPTS NINTH EDITION (9th ed.).

Tanenbaum, A. S., & Bos, H. (n.d.). MODERN OPERATING SYSTEMS FOURTH EDITION (4th ed.).

Wahid, A. A. (2019). ANALISIS SISTEM KEAMANAN PADA SISTEM OPERASI MICROSOFT WINDOWS, LINUX DAN MACINTOSH. Jurnal Teknik Informatika, 1(3).