STRATEGI PENANGANAN DEADLOCK YANG EFEKTIF DALAM SISTEM OPERASI BERBASIS WINDOWS: PENCEGAHAN DEADLOCK, MENGIDENTIFIKASI FAKTOR PENYEBAB, DAN DAMPAK DARI DEADLOCK

Ahmad Fajar[1)], Danendra Sajana Dhika[2)], Raynaldi Pratama Febriansyah[3)]
[1] 1519622018, Sistem dan Teknologi Informasi, Fakultas Teknik, Universitas Negeri Jakarta
[2] 1519622015, Sistem dan Teknologi Informasi, Fakultas Teknik, Universitas Negeri Jakarta
[3] 1519622010, Sistem dan Teknologi Informasi, Fakultas Teknik, Universitas Negeri Jakarta
email: ahmad_1519622018@mhs.unj.ac.id , danendra_1519622015@mhs.unj.ac.id , raynaldi_1519622010@mhs.unj.ac.id

Abstract

An operating system is software that manages computer resources and provides a place to execute programs. Within the scope of the operating system, deadlock is one of the problems that is often experienced by computer users. Deadlock is a condition where there are two or more processes waiting for each other to release the resources they use. This situation will make system resources unbalanced and inefficient, and can eliminate productivity. To overcome this problem, the author uses a research method, namely descriptive qualitative with literature study data collection techniques. Therefore, this research is carried out by collecting information from various reading sources such as books, notes, journals, or reports that have relevance to the problem under study. This research will focus on the aspects of preventing and handling deadlock, identifying the factors that cause deadlock, and the impact caused by deadlock in the Windows operating system. The author hopes that the results of this research will be able to help computer users, especially those with Windows operating systems, to know how to overcome deadlock.

Keywords: *deadlock, operating system, windows.*

Abstrak

Sistem operasi adalah perangkat lunak yang mengelola sumber daya komputer dan menyediakan suatu tempat untuk mengeksekusi program. Dalam lingkup sistem operasi, masalah deadlock atau kebuntuan merupakan salah satu permasalahan yang kerap kali dialami oleh pengguna komputer. Deadlock adalah sebuah kondisi dimana terdapat dua atau lebih proses yang saling menunggu satu sama lain untuk melepaskan resource yang mereka gunakan. Keadaan ini akan membuat sumber daya sistem tidak seimbang dan tidak efisien, serta dapat menghilangkan produktivitas. Untuk menanggulangi masalah itu, penulis menggunakan metode penelitian yaitu kualitatif deskriptif dengan teknik pengumpulan data studi literatur. Karena itu, penelitian ini dilakukan dengan cara mengumpulkan informasi dari berbagi sumber bacaan seperti buku, catatan, jurnal, atau laporan yang memiliki relevansi terhadap masalah yang diteliti. Penelitian ini nantinya akan memfokuskan pada aspek pencegahan dan penanganan deadlock, mengidentifikasi faktor penyebab terjadinya deadlock, serta dampak yang ditimbulkan oleh deadlock dalam sistem operasi Windows. Penulis berharap hasil penelitian ini nantinya akan dapat membantu para pengguna komputer khususnya yang bersistem operasi Windows agar dapat mengetahui bagaimana cara mengatasi *deadlock*.

Kata Kunci: *deadlock*, sistem operasi, windows.

1. PENDAHULUAN

Sistem Operasi adalah sebuah program perangkat lunak (*software*) yang memiliki peran sebagai perantara atau penghubung antara pengguna (*user*) dengan perangkat keras (*hardware*), agar perangkat keras tersebut dapat berjalan dengan baik dan normal. Sistem Operasi mengendalikan semua proses yang terjadi pada perangkat lunak maupun perangkat keras, setiap proses berada di bawah kendalinya.

Sistem Operasi memiliki peran yang mirip dengan pemerintah dalam sebuah negara, dalam hal ini yaitu menciptakan kondisi yang memungkinkan komputer untuk menjalankan program dengan tepat. Sistem Operasi mengatur akses pengguna terhadap sumber daya sebagai upaya untuk mencegah konflik atau *error* yang mungkin terjadi ketika pengguna mengakses sumber daya yang sama, hal ini yang menjadi alasan fungsi sistem operasi disebut sebagai *resource allocator*.

Sistem Operasi juga memiliki peran yang sangat penting dalam hal manajemen proses. Dalam sebuah sistem komputer yang sedang berjalan, pastinya akan ada banyak proses yang terjadi dalam suatu prossesor. Untuk itu manajemen proses menjadi bagian dari sistem operasi yang bertanggung jawab dalam mengatur, mengendalikan, dan menjalankan proses-proses tersebut agar tidak saling bertabrakan. Setiap proses-proses yang berjalan pada sistem komputer membutuhkan sinkronisasi dan komunikasi antarproses. Hal ini penting ketika dua atau lebih proses perlu berinteraksi atau sailng berbagi data. Sinkronisasi juga penting untuk menghindari sauatu kondisi yang dinamakan *deadlock*.

Deadlock adalah keadaan dalam sistem operasi dimana ada dua atau lebih proses yang saling memblokir dan tidak dapat melanjutkan proses eksekusi karena masing-masing menunggu sumber daya yang dipegang oleh proses lain. Dalam kondisi *deadlock*, tidak ada proses yang dapat menyelesaikan pekerjaannya dan sistem terjebak dalam kondisi *unproductive*.

Secara keseluruhan, manajemen proses memiliki peran vital dalam mengelola *deadlock*. Melalui deteksi, pencegahan, penghindaran, dan pemulihan, manajemen proses berupaya menjaga sistem operasi tetap berjalan tanpa terjebak dalam kondisi *deadlock* yang dapat menghambat kinerja dari sistem komputer.

2. METODE PENELITIAN

Pada paparan artikel ini, penulis menggunakan metode penelitian kualitatif deskriptif dengan pendekatan studi literatur. Studi literatur atau studi kepustakaan adalah sebuah metode pengumpulan data yang melibatkan penelaahan mendalam terhadap buku-buku, literatur-literatur, dan catatan, serta laporan yang masih relevan dengan masalah yang sedang diteliti (M. Nazir). Penelitian ini nantinya akan berfokus pada aspek pencegahan dan penangan deadlock, mengidentifikasi faktor penyebab terjadinya deadlock, serta dampak yang ditimbulkan oleh deadlock dalam sistem operasi Windows. Penelitian ini akan menyajikan tinjauan yang mendalam dan komprehensif tentang strategi penanganan deadlock yang efektif dalam sistem operasi berbasis Windows.

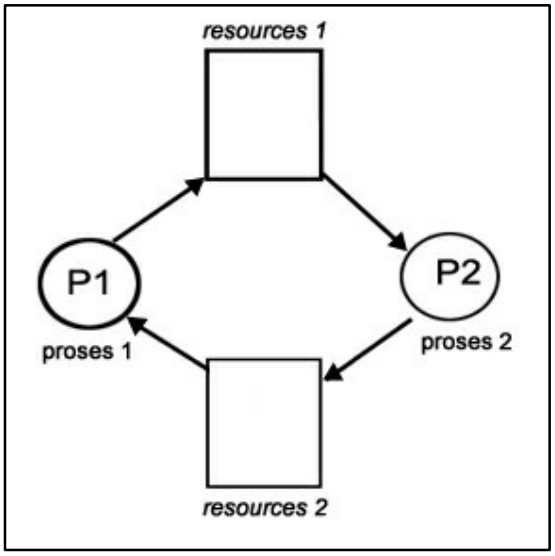
3. HASIL DAN PEMBAHASAN

Pengertian Windows

Windows adalah sistem operasi yang dirancang dan dipasarkan oleh perusahaan terkemuka yang bernama Microsoft Corporation. Sistem operasi ini adalah sistem operasi yang paling umum digunakan pada komputer pribadi (PC) dan laptop. Windows telah mengalami banyaknya perkembangan semenjak awal perilisannya. Hal tersebut dapat dibuktikan dengan banyaknya versi windows yang telah tersedia, mulai dari windows 1.0, 2.0, 3.0, 95, 98, 2000, XP, Vista, 7, 8, 8.1, 10, dan yang terbaru pada saat ini adalah windows 11. Salah satu keunggulan yang paling membedakan sistem operasi windows dengan yang lainnya adalah antarmuka penggunanya yang user-friendly, sehingga dapat memudahkan pengguna yang tidak terlalu berpengalaman dalam menggunakan komputer pribadi (PC) dan laptop.

Pengertian Deadlock

Deadlock adalah sebuah kondisi kebuntuan proses dimana terdapat dua atau lebih proses yang saling menunggu satu sama lain untuk melepaskan resource yang mereka gunakan. Pada keadaan deadlock, jika suatu proses tidak melepaskan resource yang digunakan, maka proses lainnya akan tetap berada dalam keadaan menunggu. Dikarenakan status proses tersebut adalah menunggu proses lain untuk melepaskan resourcenya, maka tidak akan terjadi kemajuan dalam proses tersebut sehingga proses tidak akan berjalan lagi serta komunikasi antar proses menjadi terhenti. Untuk lebih memahami penjelasan tersebut, berikut ini adalah ilustrasinya.



Gambar 1. Ilustrasi deadlock

Dalam situasi yang diilustrasikan dalam gambar, terdapat sebuah deadlock antara proses 1 dan proses 2. Proses 1 memiliki resource 2 dan membutuhkan resource 1 yang dimiliki oleh proses 2. Sementara itu, proses 2 memegang resource 1 dan membutuhkan resource 2 yang dimiliki oleh proses 1. Kedua proses saling memerlukan resource yang dikuasai oleh proses lain tanpa melepaskannya. Akibatnya, terjadi perulangan siklus yang tidak dapat terputus yang mengakibatkan kebuntuan atau deadlock.

Faktor Penyebab Terjadinya Deadlock

Deadlock atau kebuntuan dapat terjadi apabila terdapat beberapa faktor penyebab terjadinya deadlock sebagai berikut.

1. Mutual Exclusion

Mutual exclusion adalah sebuah konsep dimana suatu sumber daya tidak boleh dipakai dalam beberapa proses secara bersamaan. Sehingga jika proses sedang menggunakan sebuah resource, maka proses lain tidak diizinkan untuk mengaksesnya sampai proses tersebut melepaskan resourcenya. Tujuan utama dari konsep mutual exclusion adalah untuk menghindari adanya konflik yang kemungkinan timbul akibat dari akses yang diberikan kepada semua proses secara bersamaan. Dengan menerapkan mekanisme ini, sistem akan memastikan bahwa setiap proses yang membutuhkan akses ke resource harus melakukannya dengan terkontrol, teratur, dan terpisah. Akibat dari konsep ini adalah jika terdapat proses yang membutuhkan resource yang sedang digunakan pada saat itu juga, maka akan terjadi deadlock.

2. No Preemption

No preemption adalah sebuah konsep yang terdapat pada sistem operasi dimana konsep ini menyatakan bahwa proses yang sedang berjalan atau sedang mengakses resource tidak dapat dihentikan secara paksa oleh sistem operasi. Resource tersebut dapat diberikan kepada proses lain, apabila proses sebelumnya secara sukarela memberikan resourcenya yang digunakan. Karena konsep ini, suatu proses tidak dapat merebut resource yang sedang digunakan oleh proses lain dan akan mengakibatkan terjadinya deadlock.

3. Circular Wait

Circular wait adalah sebuah kondisi dimana terdapat sebuah rantai sirkuler antara proses satu dengan proses yang lainnya. Hal ini terjadi dikarenakan masing-masing proses memiliki ketergantungan resource yang dimiliki oleh proses lain tanpa melepaskan resource yang dimilikinya terlebih dahulu. Sebagai contoh terdapat proses A dan proses B. Proses A memiliki resource yang dibutuhkan oleh proses B dan proses A membutuhkan resource yang dimiliki oleh proses B. Sementara itu, proses B memiliki resource yang dibutuhkan oleh proses A dan proses B membutuhkan resource yang dimiliki oleh proses A. Kesalahannya disini adalah Proses A dan proses B tidak melepaskan resource yang mereka miliki, sehingga sistem mengalami kondisi deadlock dimana tidak ada proses yang dapat diselesaikan.

4. Hold and Wait

Hold and wait adalah konsep yang terjadi ketika suatu proses membawa resource yang sudah didapatkan dan menunggu untuk memperoleh resource lain yang dibutuhkan. Dalam konsep ini, proses yang memegang resource tidak akan melepaskan resource yang dimilikinya pada saat proses tersebut sedang menunggu resource lainnya. Hal ini jelas akan menyebabkan kondisi deadlock, dikarenakan setiap proses menahan resource yang seharusnya digunakan untuk proses lain. Oleh karena itu, konsep hold and wait dapat menyebabkan pemborosan terhadap resource dikarenakan beberapa proses mungkin menahan resource yang sebenarnya tidak digunakan pada saat itu.

Pencegahan Terjadinya Deadlock

Berikut ini adalah beberapa cara yang dapat digunakan untuk mencegah terjadinya deadlock.

1. Membuat Resource Menjadi Shareable

Kita dapat mengatasi deadlock dengan membuat resource yang ada menjadi shareable atau dapat digunakan secara bersamaan. Terdapat beberapa hal mengenai resource shareable yakni read only resource, resource copying dan resource partitioning berikut penjelasannya.

1) Read only resource dapat digunakan jika suatu proses hanya membutuhkan operasi read terhadap resourcenya. Dalam hal ini, proses dapat mengakses resource tersebut secara bersamaan dengan proses lain tanpa menyebabkan konflik pada operasinya.

2) Resource copying dapat digunakan dengan cara membuat salinan dari resource yang akan digunakan. Dalam hal ini, setiap proses memiliki salinan data pada setiap resource sehingga tidak ada ketergantungan sirkular pada prosesnya.

3) Resource partitioning dapat digunakan apabila resource memiliki bagian yang dapat digunakan secara individu, sehingga bagian dari resource tersebut dapat dipartisi atau dipisahkan agar dapat digunakan dalam proses secara bersamaan.

2. Melepas Resource Pada Saat Request

Melepaskan resource pada saat request merupakan strategi yang cukup baik untuk mencegah terjadinya kondisi deadlock. Cara kerja dari konsep ini adalah sistem akan meminta proses untuk melepaskan resource yang sudah dimilikinya apabila ingin membuat request atau permintaan untuk mendapatkan resource yang baru. Konsep ini bertujuan untuk mencegah sebuah proses memegang sejumlah resource ketika membuat sebuah permintaan untuk mendapatkan resource yang baru.

3. Melepas Resource Pada Saat Waiting

Melepaskan resource pada saat waiting atau menunggu merupakan strategi yang cukup baik untuk mencegah terjadinya kondisi deadlock. Cara kerja dari konsep ini adalah sistem akan meminta proses untuk melepaskan resource yang sudah dimilikinya apabila sedang menunggu untuk mendapatkan resource yang baru. Konsep ini bertujuan untuk mencegah sebuah proses memegang sejumlah resource ketika menunggu untuk mendapatkan resource yang baru.

4. Melakukan Request Secara Berurutan

Untuk mencegah adanya circular wait, kita dapat menggunakan strategi melakukan request secara berurutan. Proses dapat meminta resource berdasarkan urutan yang telah ditentukan. Hal ini harus membuat proses meminta resource dalam urutan yang konsisten. Berikut contoh dari melakukan request secara berurutan.

1) Proses A dimulai dan meminta resource R1. Apabila resource R1 tersedia, maka proses A akan mendapatkannya dan melanjutkan eksekusinya. Jika proses A sudah selesai menggunakan resource R1, maka Proses A harus melepaskannya untuk memberikan ke proses yang lain.

2) Proses B dimulai dan meminta resource R1. Dikarenakan resource R1 sudah dilepaskan oleh proses A, maka proses B dapat memiliki resource R1 dan melanjutkan eksekusinya. Apabila proses B sudah selesai menggunakan resource R1, maka proses B harus melepaskan resource R1 untuk memberikannya ke proses yang lain.

3) Proses A dimulai kembali dan meminta resource R2. Apabila resource R2 tersedia, maka proses A akan mendapatkannya dan melanjutkan eksekusinya. Jika proses A sudah selesai menggunakan resource R2, maka Proses A harus melepaskannya untuk memberikan ke proses yang lain

4) Proses B dimulai kembali dan meminta resource R2. Dikarenakan resource R2 sudah dilepaskan oleh proses A, maka proses B dapat memiliki resource R2 dan melanjutkan eksekusinya. Apabila proses B sudah selesai menggunakan resource R2, maka proses B harus melepaskan resource R2 untuk memberikannya ke proses yang lain.

5) Kemudian kembali ke proses A, setelah proses B melepaskan R1. proses A dapat meminta R1 kembali. Jika R1 tersedia, maka

proses A dapat memilikinya dan melanjutkan eksekusinya. Setelah menggunakan R1, proses A harus melepaskannya kembali.

Mengatasi Terjadinya Deadlock

Terdapat beberapa cara yang dapat dilakukan untuk mengatasi terjadinya kondisi deadlock. Berikut ini adalah empat cara untuk mengatasi deadlock yakni:

1. Menggunakan protokol dengan tujuan menjamin bahwa sistem tidak akan pernah memasuki kondisi deadlock.

2. Mengijinkan sistem untuk memasuki kondisi deadlock, kemudian memperbaiki sistem tersebut.

3. Menggunakan metode algoritma ostrich. Algoritma ostrich adalah sebuah metode dimana sistem berpura-pura tidak mengetahui masalah yang sedang terjadi. Hal ini seakan-akan seperti melakukan suatu hal yang fatal, namun sistem operasi Unix dapat menanggulangi permasalahan deadlock menggunakan metode ini.

4. Melakukan deteksi dan perbaikan terhadap proses yang terlihat dengan deadlock. Sehingga dengan adanya perbaikan pada proses yang berikatan, proses dapat melakukan eksekusi seperti biasanya. Adapun hal-hal yang terjadi dalam mendeteksi deadlock adalah permintaan resource diberikan selama kemungkinan, pemeriksaan kondisi circular wait oleh sistem operasi, pemeriksaan deadlock setiap pengambilan resource yang dilakukan oleh proses, pemeriksaan deadlock dengan algoritma tertentu.

Berdasarkan beberapa cara yang telah dijelaskan tersebut, terdapat beberapa jalan untuk kembali dari keadaan deadlock yakni:

1. Preemption

Preemption digunakan untuk menjauhkan resource dari proses yang sedang menggunakannya dan memberikan resource tersebut ke proses lain yang membutuhkan tanpa diketahui oleh proses yang memiliki resource tersebut. Perbaikan dengan cara preemption dapat dikatakan sangat sulit untuk dilakukan. Cara kerja dari preemption adalah dengan cara proses yang akan dikorbankan untuk diambil resourcenya untuk sementara waktu dengan perhitungan yang baik agar waktu yang dikorbankan dapat ditekan hingga seminimal mungkin. Setelah preemption dilakukan, kita harus melakukan pengkondisian proses tersebut dalam kondisi aman. Setelah itu, proses dapat dilakukan lagi pada kondisi aman tersebut.

2. Melacak Kembali

Setelah melakukan preemption, proses utama yang diambil resourcenya akan berhenti dan tidak akan bisa melakukan eksekusi seperti biasanya. Untuk mengatasi hal tersebut, dibutuhkan metode untuk kembali pada keadaan aman dimana proses masih berjalan dan kita dapat memulai proses lagi dari kondisi tersebut. Namun, pada beberapa keadaan sangat sulit untuk menentukan keberadaan kondisi aman tersebut. Oleh sebab itu, pada umumnya akan dilakukan dengan cara mematikan program tersebut, kemudian memulai kembali prosesnya. Untuk solusi lain, beberapa sistem biasanya mengadakan pengecekan secara periodik dan menandai tempat terakhir kali menulis ke disk (backup), sehingga pada saat terjadi kondisi deadlock, maka sistem dapat mulai kembali dari tempat terakhir penandanya berada.

3. Mematikan Proses Yang Menyebabkan Deadlock

Solusi selanjutnya untuk mengatasi deadlock yang terjadi adalah dengan mematikan seluruh proses yang sedang mengalami kondisi deadlock. Cara ini adalah hal yang paling umum dilakukan pada beberapa sistem operasi. Selain itu, kita juga dapat mematikan beberapa proses saja sehingga proses yang lainnya masih bisa berjalan.

4. Menghindari Deadlock

Pada beberapa sistem kebanyakan permintaan terhadap resource dilakukan sebanyak satu kali. Sistem sudah seharusnya mengenali apakah resource tersebut aman atau tidak dari kondisi deadlock. Terdapat dua cara untuk menghindari deadlock yakni:

1) Mencegah proses yang akan terkena kondisi deadlock dalam eksekusinya.

2) Mencegah proses dalam meminta resource apabila permintaan ini dapat membawa proses ke dalam kondisi deadlock.

Dampak Terjadinya Deadlock

Terjadi kondisi deadlock tentu saja membawa dampak yang sangat signifikan terhadap sistem operasi. Berikut ini adalah empat dampak yang yang disebabkan oleh kondisi deadlock.

1) Deadlock dapat menyebabkan adanya ketidakseimbangan dan ketidakefisienan sumber daya sistem.
2) Deadlock dapat menyebabkan hilangnya produktivitas dari suatu sistem dikarenakan berhenti proses yang berjalan.
3) Dikarenakan proses berhenti pada keadaan deadlock, maka keamanan dari suatu sistem akan menjadi rentan.

4. PENUTUP

Kesimpulan

Deadlock merupakan suatu kondisi dimana terdapat dua atau lebih proses yang menunggu untuk mendapatkan resource yang mereka butuhkan tanpa melepaskan resource yang mereka miliki sehingga terjadi kebuntuan proses. Kondisi deadlock memiliki empat faktor penyebab yakni mutual exclusion, no preemption, circular wait, serta hold and wait. Kita dapat melakukan pencegahan terhadap kondisi deadlock dengan cara membuat resource menjadi shareable, melepas resource pada saat request, melepas resource pada saat waiting, dan mengambil resource secara berurutan. Apabila deadlock sudah terjadi, kita dapat mengatasinya dengan melakukan preemption, melacak kembali, mematikan proses yang menyebabkan deadlock, dan menghindari deadlock. Deadlock mempunyai dampak yang cukup signifikan, dimana akan terjadi ketidakseimbangan sistem operasi, hilangnya produktivitas, keamanan tidak terjamin, serta sulit memecahkan masalah.

Saran

Berikut ini adalah beberapa saran yang dapat kami berikan:

1. Pengguna harus bisa memahami deadlock agar pengguna bisa menghindari terjadinya deadlock pada sistem operasi.
2. Ditingkatkan lagi untuk sistem operasi dalam menghindari terjadinya deadlock.
3. Pengguna memahami solusi mengatasi deadlock
4. Menerapkan mekanisme deteksi deadlock
5. Berikan prosedur jelas saat pengelolaan resource.

5. REFERENSI

[1] Yunianto, I., Krisna Adhiyarta. (2020). "Jurnal Review: Perbandingan Sistem Operasi Linux Dengan Sistem Operasi Windows." Journal of Computer, Information and Technology, Vol. 1, No. 1. Tersedia di: [https://jurnal.ibm.ac.id/index.php/jupiter/article/view/77]. (Akses: 15 Juni 2023).

[2] Kuswanto, J., Ferri Radiansah. (2018). "Media Pembelajaran Berbasis Android Pada Mata Pelajaran Sistem Operasi Jaringan Kelas XI". Jurnal Media Infotama, Vol. 14, No. 1. Tersedia di: [https://jurnal.unived.ac.id/index.php/jmi/article/view/467/424]. (Akses: 15 Juni 2023).

[3] Watrianthos, R., Iwan Purnama. (2018). "Buku Ajar Sistem Operasi." Ponorogo: Uwais Inspirasi Indonesia.

[4] Saifullah, Hani Atun M. (2017). "Implementasi Penanganan Deadlock Menggunakan Metode Taskkill." Jurnal Transformasi Informasi dan Pengembangan Iptek, Vol. 13, No. 2. Tersedia di: [https://ejournal.stmikbinapatria.ac.id/index.php/JT/article/view/144]. (Akses:15 Juni 2023).

[5] Bintara, W. S. (2023). "Pengertian Windows – Definisi, Fungsi, Sejarah, Kelebihan, Kekurangan." Tersedia di: [https://dianisa.com/pengertian-windows/]. (Akses: 15 Juni 2023).

[6] Setiawan, P. (2023). "Deadlock dan Starvation." Tersedia di: [https://www.gurupendidikan.co.id/deadlock-dan-starvation/]. (Akses: 16 Juni 2023).

[7] Muhajir, M. "Strategi Mengatasi Deadlock." Tersedia di: [https://sites.google.com/a/student.unsika.ac.id/karaos/deadlock-pada-sistem-operasi/strategi-mengatasi-deadlock]. (Akses: 16 Juni 2023).

[8] Saputri, A. "Mutual Exclusion." Tersedia di: [https://aristysaputri3.wordpress.com/sistem-operasi/mutual-exclusion/]. (Akses: 15 Juni 2023).

[9] Universitas Kristen Maranatha. "Sistem Operasi Komputer - Deadlock." Tersedia di: [https://repository.dinus.ac.id/docs/ajar/7-deadlock.pdf]. (Akses: 15 Juni 2023).