

JURNAL ILMIAH TEKNOLOGI INFORMASI ASIA



INSTITUT
ASIA

Volume 15, Nomor 2, Tahun 2021

ISSN 2580-8397



JURNAL ILMIAH TEKNOLOGI INFORMASI ASIA

LP2M INSTITUT TEKNOLOGI DAN BISNIS ASIA MALANG

[Home \(https://jurnal.stmikasia.ac.id/index.php/jitika/index\)](https://jurnal.stmikasia.ac.id/index.php/jitika/index) / Editorial Team

Editorial Team

The editors shall be entitled to abbreviate and refine the submitted scientific works insofar as they do not alter the purpose of the contents. The manuscripts in published articles do not always reflect editorial views.

PEMIMPIN REDAKSI / Editor-in-Chief

Abd Hadi, S.Kom, M.Kom. (<https://scholar.google.co.id/citations?user=ZY0vJuEAAAAJ&hl=en&oi=ao>) (Teknik Informatika, STMIK Asia Malang)

PENYUNTING AHLI / Associate Editor

Adriani Kala'lembang, S.Kom, M.M. (<http://sinta2.ristekdikti.go.id/authors/detail?id=6149039&view=overview>) (Teknik Informatika, STMIK Asia Malang)

PENYUNTING NASKAH / Manuscript Editor

Siti Nurul Afiyah, S.Si, M.Si (<https://scholar.google.co.id/citations?user=Q3SexakAAAAJ&hl=en&oi=ao>) (Teknik Informatika, STMIK Asia Malang)

Lukman Hakim, S.Si, M.Si (<https://scholar.google.co.id/citations?user=jvx-Qn0AAAAJ&hl=en>) (Teknik Informatika, STMIK Asia Malang)

PENYUNTING PERENCANA / Planning Editor

Erma Suryani, S.T, M.T, Ph.D. (<https://scholar.google.co.id/citations?user=fMx19EkAAAAJ&hl=en>) (Sistem Informasi, Institut Teknologi Sepuluh Nopember)

Gayatri Dwi Santika, S.Si., M.Kom (<https://scholar.google.co.id/citations?user=V16cpBcAAAAJ&hl=en>) (Teknologi Informasi, Universitas Jember)

Yuita Arum Sari, S.Kom., M.Kom (<https://scholar.google.co.id/citations?user=J5gQoNcAAAAJ&hl=en>) (Teknik Informatika, Universitas Brawijaya)

PEMERIKSA AKSARA / Proofreading

Nur Lailatul Aqromi, S.S., M.A (<https://scholar.google.co.id/citations?user=O35F8U4AAAAJ&hl=en>) (Teknik Informatika, STMIK Asia Malang)

PENYUNTING KELOLA / Managing Editor

Azwar Riza Habibi, S.Si, M.Si (<https://scholar.google.co.id/citations?user=pCcxzLoAAAAJ&hl=en>) (Teknik Informatika, STMIK Asia Malang)

Lia Farokhah, S.Kom, M.Eng (<https://scholar.google.co.id/citations?user=pFQaOPQAAAAJ&hl=en>) (Teknik Informatika, STMIK Asia Malang)

Mufidatul Islamiyah, S.T., M.T (<https://scholar.google.co.id/citations?user=SxacVdAAAAJ&hl=en>) (Teknik Informatika, STMIK Asia Malang)

[DOWNLOAD ARTICLE TEMPLATE \(HTTPS://DRIVE.GOOGLE.COM/FILE/D/0B8L5LEP6P2I\)](https://drive.google.com/file/d/0B8L5LEP6P2I)

Information

For

Readers

(<https://jurnal.stmikasia.ac.id/index.php/jitika/information/readers>)

For

Authors

(<https://jurnal.stmikasia.ac.id/index.php/jitika/information/authors>)

For

Librarians

(<https://jurnal.stmikasia.ac.id/index.php/jitika/information/librarians>)

Open Journal

Systems

(<http://pkp.sfu.ca/ojs/>)

Publication Ethics

(<https://jurnal.stmikasia.ac.id/index.php/jitika/pubetic>)

Journal Description

(<https://jurnal.stmikasia.ac.id/index.php/jitika/descjournal>)

Journal History

(<https://jurnal.stmikasia.ac.id/index.php/jitika/hysjournal>)

Scientific Publication

(<https://jurnal.stmikasia.ac.id/index.php/jitika/pubsci>)

Statistics Detail (<http://statcounter.com/p11573448/?guest=1>)



(<https://wa.me/6285730349141>)

Language

PENELAAH SEJAWAT / Peer-Reviewer

Aeri Rachmad, ST., M.T (<https://scholar.google.co.id/citations?user=7F6bwkQAAAAJ&hl=en>) (Fakultas Teknik Multimedia and Jaringan, Universitas Trunojoyo Madura)

Akhlis Munazilin, S.Kom., M.T. (<https://scholar.google.co.id/citations?user=J6pdLdcAAAAJ&hl=en>) (Manajemen Informatika, AMIK Ibrahimy)

Bilal Luqman Bayasut, B.CS, M.CS. (<https://scholar.google.co.id/citations?user=5PVcBFsAAAAJ&hl=en>) (Fakultas Teknik Informatika, Universitas Wijaya Putra)

Philip Faster Eka Adipraja, S.Pd., M.Kom. (<https://scholar.google.co.id/citations?hl=en&user=ObDUmdAAAAJ>) (Teknik Informatika, STMIK Asia Malang)

Philip Faster Eka Adipraja, S.Pd., M.Kom. (<https://scholar.google.co.id/citations?hl=en&user=ObDUmdAAAAJ>) (Teknik Informatika, STMIK Asia Malang)

Philip Faster Eka Adipraja, S.Pd., M.Kom. (<https://scholar.google.co.id/citations?hl=en&user=ObDUmdAAAAJ>) (Teknik Informatika, STMIK Asia Malang)

Philip Faster Eka Adipraja, S.Pd., M.Kom. (<https://scholar.google.co.id/citations?hl=en&user=ObDUmdAAAAJ>) (Teknik Informatika, STMIK Asia Malang)

Denny Trias Utomo, S.Si, MT (<https://dennytrias.wordpress.com/tentang-saya-2/>) (Teknologi Informasi, Politeknik Negeri Jember)

Dwija Wisnu Brata, S.ST, M.T (sinta2.ristekdikti.go.id/authors/detail?id=5980202&view=overview) (Teknik Informatika, STMIK Asia Malang)

Dwija Wisnu Brata, S.ST, M.T (sinta2.ristekdikti.go.id/authors/detail?id=5980202&view=overview) (Teknik Informatika, STMIK Asia Malang)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

I Komang R. Yasa Negara, ST, MT (<https://scholar.google.co.id/citations?user=SsKWr3wAAAAJ&hl=en>) (Sistem Komputer, STMIK STIKOM Bali)

Suryo Atmojo, S.Kom, M.Kom. (<https://scholar.google.co.id/citations?user=0HuXzXUAAAAJ&hl=en>) (Fakultas Teknik Informatika, Universitas Wijaya Putra)

Suryo Atmojo, S.Kom, M.Kom. (<https://scholar.google.co.id/citations?user=0HuXzXUAAAAJ&hl=en>) (Fakultas Teknik Informatika, Universitas Wijaya Putra)

Titania Dwi Andini, S.Kom, M.Kom. (<https://scholar.google.co.id/citations?user=MnF3ZR4AAAAJ&hl=en>) (Teknik Informatika, STMIK Asia Malang)

Titania Dwi Andini, S.Kom, M.Kom. (<https://scholar.google.co.id/citations?user=MnF3ZR4AAAAJ&hl=en>) (Teknik Informatika, STMIK Asia Malang)

Titus Kristanto, S.Kom, M.Kom. (<https://scholar.google.co.id/citations?user=Q5pO42EAAAAJ&hl=en>) (Fakultas Teknik Informatika, Institut Teknologi Adhi Tama Surabaya)

Titus Kristanto, S.Kom, M.Kom. (<https://scholar.google.co.id/citations?user=Q5pO42EAAAAJ&hl=en>) (Fakultas Teknik Informatika, Institut Teknologi Adhi Tama Surabaya)

Fransiska Sisilia Mukti, S.T., M.T. (<https://scholar.google.co.id/citations?user=MnF3ZR4AAAAJ&hl=en>) (Teknik Informatika, STMIK Asia Malang)

Fransiska Sisilia Mukti, S.T., M.T. (<https://scholar.google.co.id/citations?user=MnF3ZR4AAAAJ&hl=en>) (Teknik Informatika, STMIK Asia Malang)

Achmad Noercholis, S.T., M.T. (Teknik Informatika, STMIK Asia Malang)

Lia Farokhah, S.Kom., M.Eng. (Teknik Informatika, STMIK Asia Malang)

English

(https://jurnal.stmikasia.ac.id/index.php/jitika/user/setLocale/en_US?source=%2Findex.php%2Fjitika%2Fabout%2FeditorialTeam)

Bahasa

Indonesia

(https://jurnal.stmikasia.ac.id/index.php/jitika/user/setLocale/id_ID?source=%2Findex.php%2Fjitika%2Fabout%2FeditorialTeam)

MAKE SUBMIT ([HTTPS://JURNAL.STMIKASIA.AC.ID/INDEX.PHP/JITIKA/ABOUT/SUBMI](https://jurnal.stmikasia.ac.id/index.php/jitika/about/submi))

Current

Issue

 1.0

(<https://jurnal.stmikasia.ac.id/index.php/jitika/gateway/plugin/WebFeedGatewayPlugin/>)

 2.0

(<https://jurnal.stmikasia.ac.id/index.php/jitika/gateway/plugin/WebFeedGatewayPlugin/>)

 1.0

(<https://jurnal.stmikasia.ac.id/index.php/jitika/gateway/plugin/WebFeedGatewayPlugin/>)

Fadhli Almu'iini Ahda, S.Kom., M.Kom. (Teknik Informatika, STMIK Asia Malang)

LP3M Decree # 294 / E.1 / LP2M / STMIK-ASIA / X / 2018

Published By:

Lembaga Penelitian, Pengembangan, dan Pengabdian Masyarakat (LP2M)
INSTITUT TEKNOLOGI DAN BISNIS ASIA MALANG

ISSN 2580-8397 (online) - 0852-730X (Print)

Statistic (<http://statcounter.com/p11573448/?guest=1>)**Details** (<http://statcounter.com/p11573448/?guest=1>)



(<http://creativecommons.org/licenses/by/4.0/>)

This work is licensed under a Creative Commons Attribution 4.0 International License
(<http://creativecommons.org/licenses/by/4.0/>)

powered by **OJS** | Open Journal Systems

(<https://jurnal.stmikasia.ac.id/index.php/jitika/about/aboutThisPublishingSystem>)

PKP | PUBLIC KNOWLEDGE PROJECT (<http://pkp.sfu.ca/ojs>)

© 2020 Copyright: **StackIt Mod**

Home (<https://jurnal.stmikasia.ac.id/index.php/jitika/index>)

/ Archives (<https://jurnal.stmikasia.ac.id/index.php/jitika/issue/archive>)

/ Vol 15 No 2 (2021): Volume 15 Nomor 2 (8)

DOI: <https://doi.org/10.32815/jitika.v15i2> (<https://doi.org/10.32815/jitika.v15i2>)

Published: 2021-08-20

Articles

Deteksi Dini Masalah dalam Proses Belajar Mengajar Secara Daring Menggunakan Sistem AT-OLP (<https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/485>)

Karfindo Karfindo, Rifa Turaina

PDF (BAHASA INDONESIA) (<HTTPS://JURNAL.STMIKASIA.AC.ID/INDEX.PHP/JITIKA/ARTICLE/VIEW/485/>)

Analisis Algoritma Round Robin pada Penjadwalan CPU (<https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/481>)

Tri Dharma Putra, Rakhmat Purnomo

85-90

PDF (BAHASA INDONESIA) (<HTTPS://JURNAL.STMIKASIA.AC.ID/INDEX.PHP/JITIKA/ARTICLE/VIEW/481/>)

Analisis Quality of Service (QoS) Jaringan Wireless 2.4 GHz dan 5 GHz di Dalam Ruang dengan Hambatan Kaca (<https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/>)

Stefanus Eko Prasetyo, Elvin Tan

103-114

PDF (BAHASA INDONESIA) (<HTTPS://JURNAL.STMIKASIA.AC.ID/INDEX.PHP/JITIKA/ARTICLE/VIEW/609/>)

Implementasi Sistem Layanan Pengaduan Masyarakat Kecamatan Medan Amplas Berbasis Website (<https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/611>)

Yudha Sansena

91-102

PDF (BAHASA INDONESIA) (<HTTPS://JURNAL.STMIKASIA.AC.ID/INDEX.PHP/JITIKA/ARTICLE/VIEW/611/>)

Penerapan Rule-Based Expert System (RBES) Dalam Perancangan Aplikasi Sistem P

Untuk Mendiagnosa Penyakit Infeksi Saluran Pernapasan Akut (ISPA) Berbasis Andr (<https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/589>)

Nisaul Fadila, Rinabi Tanamal

115-122

PDF (BAHASA INDONESIA) ([HTTPS://JURNAL.STMIKASIA.AC.ID/INDEX.PHP/JITIKA/ARTICLE/VIEW/589/](https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/589/):

Penerapan Fitur Warna dan Tekstur untuk Identifikasi Kerusakan Mutu Biji Kopi Arab (Coffea Arabica) di Kabupaten Bondowoso (<https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/593>)

Zilvanhisna Emka Fitri, Brilyan Andi Syahbana, Abdul Madjid, Arizal Mujibtamala Nanda Imron

123-128

PDF (BAHASA INDONESIA) ([HTTPS://JURNAL.STMIKASIA.AC.ID/INDEX.PHP/JITIKA/ARTICLE/VIEW/593/](https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/593/):

Penentuan Kelompok Status Gizi Balita dengan Menggunakan Metode K-Means (<https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/594>)

Oktania Purwaningrum, Yudha Yunanto Putra, Amalia Anjani Arifiyanti

129-136

PDF (BAHASA INDONESIA) ([HTTPS://JURNAL.STMIKASIA.AC.ID/INDEX.PHP/JITIKA/ARTICLE/VIEW/594/](https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/594/):

Analisa Sentimen Pada Twitter Terhadap Kenaikan Tarif Dasar Listrik Dengan Metod Naïve Bayes (<https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/557>)

Adi Kusuma, Agung Nugroho

137-146

PDF (BAHASA INDONESIA) ([HTTPS://JURNAL.STMIKASIA.AC.ID/INDEX.PHP/JITIKA/ARTICLE/VIEW/557/](https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/557/):

#image .top-bar-placeholder, .image-viewer.full-viewer

Information

For Readers (<https://jurnal.stmikasia.ac.id/index.php/jitika/information/readers>)

For Authors (<https://jurnal.stmikasia.ac.id/index.php/jitika/information/authors>)

For Librarians (<https://jurnal.stmikasia.ac.id/index.php/jitika/information/librarians>)

Open Journal Systems (<http://pkp.sfu.ca/ojs/>)

Publication Etics (<https://jurnal.stmikasia.ac.id/index.php/jitika/pubetic>)

Journal Description (<https://jurnal.stmikasia.ac.id/index.php/jitika/descjournal>)

Journal History (<https://jurnal.stmikasia.ac.id/index.php/jitika/hysjournal>)

Scientific Publication (<https://jurnal.stmikasia.ac.id/index.php/jitika/pubsci>)

Statistics Detail (<http://statcounter.com/p11573448/?guest=1>)



[_\(https://wa.me/6285730349141\)](https://wa.me/6285730349141)

Language

English (https://jurnal.stmikasia.ac.id/index.php/jitika/user/setLocale/en_US?source=%2Findex.php%2Fjitika%2Fissue%2Fview%2F35)

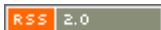
Bahasa Indonesia (https://jurnal.stmikasia.ac.id/index.php/jitika/user/setLocale/id_ID?source=%2Findex.php%2Fjitika%2Fissue%2Fview%2F35)

MAKE SUBMIT (<HTTPS://JURNAL.STMIKASIA.AC.ID/INDEX.PHP/JITIKA/ABOUT/SUBMISSIONS>)

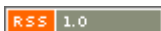
Current Issue



(<https://jurnal.stmikasia.ac.id/index.php/jitika/gateway/plugin/WebFeedGatewayPlugin/atom>)



(<https://jurnal.stmikasia.ac.id/index.php/jitika/gateway/plugin/WebFeedGatewayPlugin/rss2>)



(<https://jurnal.stmikasia.ac.id/index.php/jitika/gateway/plugin/WebFeedGatewayPlugin/rss>)

Published By:

Lembaga Penelitian, Pengembangan, dan Pengabdian Masyarakat (LP2M)
INSTITUT TEKNOLOGI DAN BISNIS ASIA MALANG

ISSN 2580-8397 (online) - 0852-730X (Print)

Statistic (<http://statcounter.com/p11573448/?guest=1>)**Details** (<http://statcounter.com/p11573448/?guest=1>)



(<http://creativecommons.org/licenses/by/4.0/>)

This work is licensed under a Creative Commons Attribution 4.0 International License
(<http://creativecommons.org/licenses/by/4.0/>)

powered by OJS | Open Journal Systems

(<https://jurnal.stmikasia.ac.id/index.php/jitika/about/aboutThisPublishingSystem>)

PKP | PUBLIC KNOWLEDGE PROJECT (<http://pkp.sfu.ca/ojs>)

© 2020 Copyright: **StackIt Mod**

Analisis Algoritma Round Robin pada Penjadwalan CPU

Tri Dharma Putra¹, Rakhmat Purnomo²

^{1,2} Universitas Bhayangkara Jakarta Raya

¹tri.dharma.putra@dsn.ubharajaya.ac.id, ²rakhmat.purnomo@dsn.ubharajaya.ac.id

ABSTRAK. Penjadwalan adalah konsep penting dalam sistem operasi multiprosesor dan *multitasking* pada sistem operasi waktu-nyata dengan mengalihkan proses pada CPU. Algoritma Round Robin adalah algoritma yang terkenal pada penjadwalan CPU. Algoritma Round Robin memberikan waktu *quantum* antara pengalihan proses. Memilih waktu *quantum* dalam Algoritma Round Robin sangatlah penting, waktu *quantum* besar akan mengakibatkan *context switching* lebih sedikit, sementara waktu *quantum* lebih kecil akan mengakibatkan *context switching* yang lebih sering. Algoritma Round Robin yang efisien adalah jumlah *context switching* lebih rendah. Untuk waktu tunggu, ide dasarnya adalah untuk mendapatkan waktu tunggu rata-rata yang lebih kecil, sehingga sistem lebih efisien. *Turn around time* juga harus minimum, yang berarti juga lebih efisien. Dua studi kasus didiskusikan untuk memahami algoritma ini dengan lebih mendalam. Dilakukan analisis pada data set proses 1 sampai proses 5 (P1-P5), dengan data set *bursttime*: 20mdetik, 34mdetik, 25mdetik, 62mdetik, 67mdetik. Pada analisis studi kasus 1 digunakan *quantum* 25mdetik, dan pada studi kasus kedua digunakan *quantum* 15mdetik. Context switching Pada studi kasus 2 ada sebanyak 17, sementara pada pada studi kasus 1, hanya terdapat 10 context switching. Total burst-time pada studi kasus 1 dan studi kasus 2 adalah 208 mdetik, Waktu tunggu rata-rata pada studi kasus 1 adalah 82 mdetik, sedangkan untuk studi kasus 2 adalah 105,8 mdetik. Rata-rata Turn Around Time pada studi kasus 1 adalah 123,6mdetik, sedangkan pada studi kasus 2 adalah 146,6 mdetik.

Kata kunci: Algoritma Round Robin, waktu quantum, context switching, rata-rata waktu tunggu, rata-rata turn around time

ABSTRACT. Scheduling is a key concept in computer multitasking and multiprocessing operating system design and in real-time operating system design by switching the CPU among process. Round Robin Algorithm is a wellknown algorithm in CPU scheduling. Round Robin Algorithm provides a time quantum between context switching. Choosing the time quantum in Round Robin Algorithm is very crucial, large quantum time will result in lower context switching, while small quantum time wil result in higher context switching. For waiting time, the idea for waiting time is to get smaller waiting time which is more efficient. Turn around time also should be minimum, which is more efficient. Two case studies are discussed to understand this algorithm more deeply. We did analysis with data set process 1 till process 5 (P1-P5), with burst time dataset: 20ms, 34ms, 25ms, 62ms, 67ms. In case study 1, we use quantum 25m, and in case study 2, we use quantum 15ms. Context switching in case study 2 is 17 where on case study 1, only 10 context switchings. Total burst-time in case study 1 and case study 2 are 208 ms. Average waiting time in case study 1 is 82 ms, where in case study 2 is 105.8 ms. Average Turn Around Time in case study 1 is 123.6ms, where in case study 2 is 146.6 ms.

Keywords: Round Robin Algorithm, quantum time, context switching, average waiting time, average turn around time

1. PENDAHULUAN

Sistem operasi adalah pengelola sumber daya. Sumber daya yang dikelola oleh sistem operasi adalah perangkat keras, unit penyimpanan, perangkat input, perangkat output dan data. Sistem operasi melakukan banyak fungsi seperti mengimplentasikan antar muka pengguna, berbagai perangkat keras antar pengguna, memfasilitasi input/output, perhitungan penggunaan sumber daya, mengatur data, dll. Penjadwalan proses adalah salah satu dari fungsi yang dilakukan oleh sistem operasi. Penjadwalan CPU adalah tugas dalam memilih proses dari antrian siap dan mengalokasikan CPU padanya (Amar Ranjan Dash, 2015). Apabila CPU menjadi *idle*, proses menunggu dari antrian siap dipilih dan CPU dialokasikan padanya.

Sebuah sistem operasi adalah sebuah program yang mengelola perangkat keras komputer dan menyediakan dasar untuk aplikasi program and bertindak sebagai penengah antara pengguna komputer dan perangkat keras (Md. Sohrawordi, 2019). Sistem operasi menyediakan antarmuka antara pengguna dan perangkat keras komputer yang membantu pengguna menangani sistem dengan cara yang nyaman. Sistem operasi modern menjadi lebih kompleks ketika sistem beralih dari tugas tunggal ke lingkungan multitasking (Tithi Paul, 2019).

Sistem operasi modern mendukung lingkungan *multitasking* dimana proses berjalan bersamaan. Pada sistem dengan prosesor tunggal, hanya satu proses yang dapat berjalan pada CPU pada satu waktu. Proses

lainnya yang berada di antrian siap harus menunggu sampai CPU menjadi bebas. Sistem operasi harus memutuskan melalui *scheduler*, urutan eksekusi dari proses. Objektif dari *multiprogramming* atau *multitasking* adalah untuk membuat beberapa proses berjalan setiap saat, untuk memaksimalkan utilisasi CPU.

Penjadwalan adalah fungsi sistem operasi yang fundamental. Hampir semua sumber daya komputer dijadwalkan sebelum digunakan. CPU, tentu saja, merupakan salah satu sumber daya komputer utama. Karena itu, penjadwalannya menjadi penting sekali pada desain sistem operasi. Penjadwalan CPU menentukan proses mana yang berjalan ketika ada beberapa proses yang dapat dieksekusi. Penjadwalan CPU adalah penting karena dapat memiliki efek pada utilisasi sumber daya dan kinerja keseluruhan sistem. Secara umum kita ingin mengoptimalkan perilaku sistem (Datta, 2015).

Ide dasar *multiprogramming* adalah mengatur beberapa proses untuk dieksekusi oleh CPU. Yaitu ketika proses mengalokasikan sumber daya bagi CPU. Ketika ada lebih dari satu proses yang dapat dieksekusi, hanya satu proses yang akan dialokasikan ke sumber daya. Ada beberapa algoritma, FCFS (First Come First Serve), *Priority Scheduling*, and *Shortest Job First (SJF) Scheduling*, dan Algoritma Round Robin. Penjadwalan CPU adalah masalah dalam memutuskan proses mana yang siap di antrian harus dialokasikan ke CPU (Putra, 2020).

Penjadwalan dapat merupakan *preemptive* atau *nonpreemptive*. Pada penjadwalan nonpreemptive, CPU tidak dapat mengalihkan proses ke proses lain sampai waktu penyelesaian dituntaskan, sedangkan penjadwalan *preemptive*, proses yang sedang dieksekusi mungkin dipaksa untuk mengalihkan proses sedang berjalan oleh proses lain berdasarkan kriteria penjadwalan (Dhruv, 2019). Pada Algoritma Round Robin, proses dialihkan ke proses berikutnya berdasarkan waktu *quantum* yang ditentukan. Sehingga Algoritma Round Robin merupakan algoritma *preemptive*.

Jadi kita dapat menyimpulkan bahwa algoritma penjadwalan yang baik untuk sistem waktu nyata dan time sharing harus memiliki karakteristik berikut:

- *Context switching* yang minimum
- Utilisasi CPU yang maksimum
- *Throughput* yang maksimum
- *Turnaround time* yang minimum
- Waktu tunggu yang minimum, dan
- Waktu respons yang minimum.

2. METODE PENELITIAN

Metode merupakan teknik yang digunakan pada pemecahan suatu masalah untuk menghasilkan hasil yang sesuai dengan yang diharapkan. Penulis menganalisis algoritma round robin dalam jurnal ini berdasarkan berbagai sumber, jurnal dan sumber-sumber lain untuk digunakan dalam penelitian ini, semuanya terpercaya dan digunakan dalam penulisan jurnal penelitian ini, semuanya tentu saja berkaitan dengan algoritma round robin yang kita bahas di sini.

Metode penelitian pada jurnal ini adalah studi pustaka yang mendalam. Literatur studi dilakukan dari berbagai sumber literatur dan jurnal. Beberapa penelitian terkait dibandingkan berdasarkan literatur. Ada beberapa macam varian Algoritma Round Robin di literatur, salah satunya adalah dengan waktu *quantum* yang bervariasi. Namun dalam studi kasus dalam jurnal ini dilakukan dengan waktu *quantum* yang tetap. Dua studi kasus Algoritma Round Robin dibahas secara mendalam satu persatu.

Round robin adalah metode pada penjadwalan proses yang membagi semua proses dengan porsi waktu yang sama. Proses dimana waktu prosesnya lebih pendek akan selesai lebih dahulu, daripada proses yang waktu prosesnya lebih lama. Metode ini membagi proses dengan *quantum* yang sama pada setiap proses. Berapa lama nilai *quantum* terhadap proses telah ditentukan sebelumnya. Tidak ada prioritas dalam proses dan diberikan *quantum* atau *time-slice* yang sama. Round robin dibuat sehingga memenuhi kebutuhan time-sharing. Dalam jurnal ini semua proses memiliki waktu *quantum* yang sama.

Keuntungan Algoritma Round Robin adalah sebagai berikut (J. R. Indusree, 2017):

- *Starvation* pada proses dapat dihindari, karena setiap proses mendapatkan giliran untuk dieksekusi oleh CPU.
- Proses dengan *burst-time* lebih kecil mendapat keuntungan karena dieksekusi lebih dulu dibandingkan dengan algoritma yang lain.
- Algoritma Round Robin adalah *preemptive*.

Tujuan penjadwalan sistem adalah untuk :

- *Throughput*: Throughput adalah jumlah proses yang diselesaikan dalam satu unit waktu. Algoritma penjadwalan harus didesain sedemikian rupa sehingga *throughput* sistem menjadi maksimal.
- *Utilisasi CPU*: Ini adalah persentase waktu bahwa CPU sibuk dalam mengeksekusi proses. Tujuan dasar penjadwalan adalah menjaga prosesor sibuk sepanjang waktu.
- *Fairness*: Semua proses dalam sistem harus diperlakukan dalam cara yang sama kecuali ada beberapa proses yang disukai atau diprioritaskan. Jika seperti ini, semua proses dengan prioritas lebih rendah harus diabaikan untuk menghindari *starvation*.
- *Context Switching*: Adalah prosedur menyimpan status proses aktif dan memulihkan status proses lain untuk CPU ketika sistem harus mulai mengeksekusi proses berikutnya. *Context switching* adalah *overhead* dari sistem dan membuat waktu tidak efektif dari CPU. Algoritma penjadwalan harus didesain untuk meminimalkan *context switching*.

Jurnal ini diatur dalam bab-bab berikut ini. Jurnal ini dibagi menjadi lima bab. Bab 1 adalah Pendahuluan. Di bab satu ini dibahas konsep penjadwalan sistem operasi secara umum. Bab dua adalah Metode Penelitian. Pada bab ini dibahas metode penelitian yang dilakukan. Yaitu literatur studi yang mendalam. Bab 3 merupakan pembahasan. Pada pembahasan ini dianalisis dua case studi untuk Algoritma Round Robin. Bab 5 adalah Kesimpulan. Dan yang terakhir adalah Daftar Pustaka.

3. HASIL DAN PEMBAHASAN

Algoritma Round Robin didesain khususnya untuk sistem *time-sharing*. Masing-masing proses ditetapkan interval waktu, yang disebut waktu *quantum*. Masing-masing proses proses disediakan waktu tetap untuk mengeksekusi sesuai waktu *quantum*. Waktu *quantum* umumnya adalah 10 hingga 100 mdetik. Antrian siap diperlakukan sebagai antrian sirkuler (Arpita Sharma, 2015). Sehingga proses dialihkan berurutan dari P1 hingga P5 dan kembali ke P1 hingga ke P5 kembali secara sirkuler.

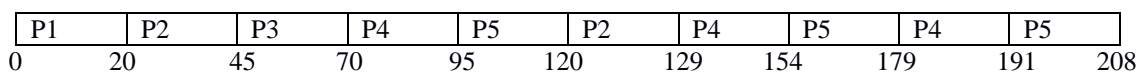
a. Studi Kasus 1

Diberikan lima proses P1 sampai P5 pada tabel 1 di bawah ini. Dengan *burst-time* masing-masingnya. Pada studi kasus ini diberikan waktu *quantum* sama dengan 25 mdetik.

Tabel 1. Lima Proses dan masing-masing *Burst-Time* CPU dengan waktu *quantum* = 25 mdetik

Nama Proses	Burst-Time CPU (mdetik)
P1	20
P2	34
P3	25
P4	62
P5	67

Perhatikan Gambar 1. Gantt Chart studi kasus 1. Proses akan bergiliran dari P1 ke P5 dan kembali ke P1 lagi, dengan urutannya. Proses dimulai dengan P1 dengan *burst-time* adalah 20. Setelah itu adalah P2, dengan *burst-time* adalah 34, sehingga tersisa 9, dengan *quantum* 25 sehingga sampai di 45. P3, 45 ditambah 25 menjadi 70, dengan sisa *burst-time* 0, berarti proses P3 selesai. Kemudian P4 masuk sampai 95. Pada P4 sisa proses adalah 37. Kemudian beralih ke P5 sampai ke 120, dengan sisa *burst-time* 42. Dari P5 beralih ke P1 kembali, namun P1 sudah selesai dieksekusi, sehingga masuk P2, sisa P2 adalah 9, hingga sampai ke 129. Dari P2 beralih ke P3, namun P3 sudah selesai, sehingga lanjut ke P4. P4 bersisa 37, sehingga sampai ke 154, dengan sisa P4 adalah 12. Dari P4 beralih ke P5, sisa P5 adalah 42, sehingga dikurangi 25 menjadi 17. P4 beralih ke P5 pada 179. Sistem secara sirkuler berlanjut ke P1 kembali. P1 sudah selesai, lanjut P2, P2 juga sudah selesai, sehingga lanjut urutan ke P3, P3 juga sudah selesai lanjut ke P4. Menjadi 191, karena sisa P4 adalah 12. Akhirnya P5 masuk dengan menjadi 208.



Gambar 1. Gantt Chart Studi Kasus 1 dengan *quantum* = 25

Untuk menghitung waktu tunggu setiap proses perhatikan Tabel 2.:

Tabel 2. Menghitung waktu tunggu rata-rata proses

Proses	Waiting Time
P1	0
P2	$20 + (120-45)= 95$
P3	45
P4	$70 + (129-95)+(179-154)=129$
P5	$95+(154-120)+(191-179)=141$

Sehingga waktu tunggu rata-rata adalah $(0+95+45+129+141)/5=82$ mdetik. Untuk menghitung rata-rata Turn Around Time perhatikan tabel 3. Di bawah ini:

Tabel 3. Menghitung rata-rata Turn Around Time

Proses	Waktu tiba	Burst-time	Waktu mulai	Waktu selesai	Turn Around Time
P1	0	20	0	20	20
P2	0	34	20	129	129
P3	0	25	45	70	70
P4	0	62	70	191	191
P5	0	67	95	208	208

Sehingga rata-rata Turn Around Time adalah $(20+129+70+191+208)/5=123,6$ mdetik.

b. Studi Kasus 2

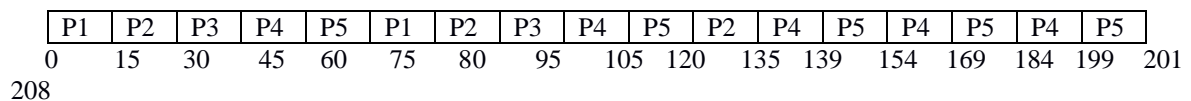
Pada studi kasus 2 ini, perhatikan Tabel 4. di bawah ini. Proses dan *burst-time* sama, namun waktu *quantum* yang digunakan adalah 15. Lebih sedikit waktu *quantum* maka *context switching* akan menjadi lebih banyak.

Tabel 4. Lima Proses dan masing-masing *Burst-Time* CPU dengan waktu *quantum* = 15

Nama Proses	Burst-Time CPU
P1	20
P2	34
P3	25
P4	62
P5	67

Proses P1 dimulai, karena quantum 15 maka proses bersisa 5. Kemudian proses P2 masuk, dan proses P2 bersisa 19. Proses P3 masuk dan bersisa 10. Kemudian P4 masuk dan bersisa 47. Setelah itu proses P5 masuk dan bersisa 52. Proses kemudian kembali ke P1 lagi, secara sirkuler. P1 masuk dan tidak bersisa, dari 75 menjadi 80 karena P1 hanya bersisa 5. Kemudian P2 masuk dari 19 menjadi bersisa 4, dengan data di gantt

chart menjadi 95. Kemudian P3 masuk sebanyak 10, gantt chart menjadi 105. Berikutnya P4 masuk, sisa sebelumnya 47, sehingga gantt chart menjadi 120. Dengan proses bersisa 32. Berikutnya P5 masuk, status sekarang adalah 52, sehingga sisa menjadi 37, pada gantt chart menjadi 135. Kemudian proses beroperasi secara sirkuler, kembali ke P1. P1 status tidak ada lagi proses tersisa sehingga dilewatkan. P2 masuk dengan *burst-time* tersisa 4, sehingga di gantt chart menjadi 139 dari 135. Kemudian proses beralih ke P3. P3 ternyata sudah selesai tidak ada *burst-time* tersisa, sehingga lanjut ke P4. P4 masuk, dari tersisa 32 menjadi tersisa 17, di gantt chart menjadi 154, yaitu ditambahkan 139 tambah 15 menjadi 154. Berikutnya P5 masuk dengan sisa yang ada 37, sehingga sisa menjadi 22, di gantt chart menjadi 169. Kemudian proses secara sirkuler kembali ke P1. Pada P1 proses sudah selesai, sehingga lanjut ke P2. P2 juga sudah selesai sehingga lanjut ke P3, P3 juga sudah selesai, sehingga dilanjutkan ke P4. Pada P4 sisa *burst-time* adalah 17, sehingga di gantt chart dari 169 menjadi 184. Sisa *burst-time* pada P4 ini adalah 2. Proses kemudian berlanjut ke P5. Pada P5, sisa proses yang ada 22, sehingga di gantt chart menjadi 199, dengan *burst-time* tersisa 7. Proses kemudian secara sirkuler balik lagi ke P1. P1 sudah selesai, sehingga lanjut ke P2. P2 sudah selesai sehingga lanjut ke P3, P3 sudah selesai sehingga lanjut ke P4. Pada P4 *burst-time* tersisa 2, sehingga di gantt chart menjadi 201 dari 199. Berikutnya P5, yang hanya tersisa 7, sehingga di gantt chart menjadi 208 dari 201. Akhirnya semua proses selesai.



Gambar 2. Gantt Chart Studi Kasus 2 dengan waktu *quantum* = 15

Untuk perhitungan waktu tunggu rata-rata perhatikan Tabel 5. di bawah ini:

Tabel 5. Menghitung waktu tunggu rata-rata proses

Nama Proses	Waktu Tunggu
P1	$0+(75-15)=60$
P2	$15+(80-30)+(135-95)=109$
P3	$30+(95-45)=80$
P4	$45+(105-60)+(139-120)+(169-154)+(199-184)=139$
P5	$60+(120-75)+(154-135)+(184-169)+(201-199)=141$

Maka waktu tunggu rata-rata adalah $(60+109+80+139+141)/5=105,8$ mdetik. Untuk menghitung rata-rata *Turn Around Time* perhatikan Tabel 6. di bawah ini:

Tabel 6. Menghitung rata-rata *Turn Around Time*

Proses	Waktu Tiba	Burst-time	Waktu mulai	Waktu selesai	Turn Around Time
P1	0	20	0	80	80
P2	0	34	15	139	139
P3	0	25	30	105	105
P4	0	62	45	201	201
P5	0	67	60	208	208

Sehingga rata-rata *Turn Around Time* adalah $(80+139+105+201+208)/5=146,6$ mdetik.

4. KESIMPULAN

Context switching yang lebih banyak terjadi pada waktu *quantum* = 15. Pada studi kasus 2 dengan waktu *quantum* 15, ada sebanyak 17 *context switching*, sementara pada waktu *quantum* = 25, pada studi kasus 1, hanya terdapat 10 *context switching*.

Total *burst-time* untuk waktu *quantum* = 25 pada studi kasus 1 adalah 208 mdetik, sedangkan pada waktu *quantum* = 15, adalah juga sama 208 mdetik, nilai ini harus sama, yang membuktikan bahwa perhitungan gantt chart sudah benar pada kedua studi kasus.

Waktu tunggu rata-rata untuk waktu *quantum*=25 pada studi kasus 1 adalah 82 mdetik, sedangkan waktu tunggu rata-rata untuk studi kasus 2, dengan waktu *quantum* 15 adalah 105,8 mdetik. Sehingga waktu tunggu yang lebih efisien adalah pada *quantum* yang sama dengan 25. Karena waktu tunggu nya lebih kecil.

Rata-rata *Turn Around Time* untuk waktu *quantum* = 25 pada studi kasus 1 adalah 123,6, sedangkan rata-rata *Turn Around Time* untuk waktu *quantum* = 15 pada studi kasus 2 adalah 146,6 mdetik. Sehingga rata-rata *Turn Around Time* yang lebih efisien adalah pada waktu *quantum* yang sama dengan 25, karena rata-rata *Turn Around Time*-nya lebih kecil.

5. SARAN

Diusulkan untuk studi kasus Algoritma Round Robin berikutnya adalah dengan menggunakan waktu *quantum* yang bervariasi, sehingga lebih mengefisienkan sistem.

DAFTAR RUJUKAN

- Amar Ranjan Dash, S. K. (2015). An Optimized Round Robin CPU Scheduling Algorithm with Dynamic Time Quantum. *International Journal of Computer Science, Engineering and Information Technology*, Vol. 5, No. 1, 7-26.
- Arpita Sharma, M. G. (2015). Analysis of Adaptive Round Robin Algorithm and Proposed Round Robin Remaining Time Algorithm. *International Journal of Computer Science and Mobile Computing*, Vol 4, Issue 12, 139-147.
- Datta, L. (2015). Efficient Round Robin Scheduling Algorithm with Dynamic Time Slice. *International Journal of Education and Management Engineering*, 10-19.
- Dhruv, R. (2019). Round Robin Scheduling Algorithm Based on Dynamic Time Quantum. *International Journal of Engineering and Advanced Technology*, Volume X, Issue X, 593-595.
- J. R. Indusree, B. P. (2017). Enhanced Round Robin CPU Scheduling with Burst Time Based Time Quantum. *IOP Conference Series: Material Science and Engineering*, 1-8.
- Md. Sohrawordi, U. A. (2019). A Modified Round Robin CPU Scheduling Algorithm with Dynamic Time Quantum. *International Journal of Advanced Research (IJAR)*, 422-429.
- Putra, T. D. (2020). Analysis of Preemptive Shortest Job First (SJF) Algorithm in CPU Scheduling. *International Journal of Advanced Research in Computer and Communication Engineering*, Vol 9, Issue 4, April, 41-45.
- Tithi Paul, R. H. (2019). Improved Round Robin Scheduling Algorithm with Progressive Time Quantum. *International Journal of Computer Application Volume 178, No. 49*, 30-36.



Plagiarism Checker X Originality Report

Similarity Found: 11%

Date: Thursday, September 02, 2021

Statistics: 317 words Plagiarized / 2901 Total words

Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

Analisis Algoritma Round Robin pada Penjadwalan CPU Tri Dharma Putra¹, Rakhmat Purnomo² 1Universitas Bhayangkara Jakarta Raya 2Universitas Bhayangkara Jakarta Raya 1tri.dharma.putra@dsn.ubharajaya.ac.id, 2rakhmat.purnomo@dsn.ubharajaya.ac.id
ABSTRAK. Penjadwalan adalah konsep penting dalam sistem operasi multiprosesor dan multitasking pada sistem operasi waktu-nyata dengan mengalihkan proses pada CPU.

Algoritma Round Robin adalah algoritma yang terkenal pada penjadwalan CPU. Algoritma Round Robin memberikan waktu quantum antara pengalihan proses. Memilih waktu quantum dalam Algoritma Round Robin sangatlah penting, waktu quantum besar akan mengakibatkan context switching lebih sedikit, sementara waktu quantum lebih kecil akan mengakibatkan context switching yang lebih sering. Algoritma Round Robin yang efisien adalah jumlah context switching lebih rendah.

Untuk waktu tunggu, ide dasarnya adalah untuk mendapatkan waktu tunggu rata-rata yang lebih kecil, sehingga sistem lebih efisien. Turn around time juga harus minimum, yang berarti juga lebih efisien. Dua studi kasus didiskusikan untuk memahami algoritma ini dengan lebih mendalam. Dilakukan analisis pada data set proses 1 sampai proses 5 (P1-P5), dengan data set burstime: 20mdetik, 34mdetik, 25mdetik, 62mdetik, 67mdetik.

Pada analisis studi kasus 1 digunakan quantum 25mdetik, dan pada studi kasus kedua digunakan quantum 15mdetik. Context switching Pada studi kasus 2 ada sebanyak 17, sementara pada pada studi kasus 1, hanya terdapat 10 context switching. Total burst-time pada studi kasus 1 dan studi kasus 2 adalah 208 mdetik, Waktu tunggu rata-rata pada studi kasus 1 adalah 82 mdetik, sedangkan untuk studi kasus 2 adalah 105,8 mdetik. Rata-rata Turn Around Time pada studi kasus 1 adalah 123,6mdetik, sedangkan pada studi kasus 2 adalah 146,6 mdetik.

Kata kunci: Algoritma Round Robin, waktu quantum, context switching, rata-rata waktu tunggu, rata-rata turn around time ABSTRACT. Scheduling is a key concept in computer multitasking and multiprocessing operating system design and in real-time operating system design by switching the CPU among process. Round Robin Algorithm is a wellknown algorithm in CPU scheduling. Round Robin Algorithm provides a time quantum between context switching.

Choosing the time quantum in Round Robin Algorithm is very crucial, large quantum time will result in lower context switching, while small quantum time wil result in higher context switching. For waiting time, the idea for waiting time is to get smaller waiting time which is more efficient. Turn around time also should be minimum, which is more efficient. Two case studies are discussed to understand this algorithm more deeply.

We did analysis with data set process 1 till process 5 (P1-P5), with burst time dataset: 20ms, 34ms, 25ms, 62ms, 67ms. In case study 1, we use quantum 25m, and in case study 2, we use quantum 15ms. Context switching in case study 2 is 17 where on case study 1, only 10 context switchings. Total burst-time in case study 1 and case study 2 are 208 ms. Average waiting time in case study 1 is 82 ms, where in case study 2 is 105.8 ms.

Average Turn Around Time in case study 1 is 123.6ms, where in case study 2 is 146.6 ms. Keywords: Round Robin Algorithm, quantum time, context switching, average waiting time, average turn around time PENDAHULUAN Sistem operasi adalah pengelola sumber daya. Sumber daya yang dikelola oleh sistem operasi adalah perangkat keras, unit penyimpanan, perangkat input, perangkat output dan data.

Sistem operasi melakukan banyak fungsi seperti mengimplementasikan antar muka pengguna, berbagai perangkat keras antar pengguna, memfasilitasi input/output, perhitungan penggunaan sumber daya, mengatur data, dll. Penjadwalan proses adalah salah satu dari fungsi yang dilakukan oleh sistem operasi. Penjadwalan CPU adalah tugas dalam memilih proses dari antrian siap dan mengalokasikan CPU padanya (Amar Ranjan Dash, 2015).

Apabila CPU menjadi idle, proses menunggu dari antrian siap dipilih dan CPU dialokasikan padanya. Sebuah sistem operasi adalah sebuah program yang mengelola perangkat keras komputer dan menyediakan dasar untuk aplikasi program and bertindak sebagai penengah antara pengguna komputer dan perangkat keras (Md. Sohrawordi, 2019).

Sistem operasi menyediakan antarmuka antara pengguna dan perangkat keras

komputer yang membantu pengguna menangani sistem dengan cara yang nyaman. Sistem operasi modern menjadi lebih kompleks ketika sistem beralih dari tugas tunggal ke lingkungan multitasking (Tithi Paul, 2019). Sistem operasi modern mendukung lingkungan multitasking dimana proses berjalan bersamaan.

Pada sistem dengan prosesor tunggal, hanya satu proses yang dapat berjalan pada CPU pada satu waktu. Proses lainnya yang berada di antrian siap harus menunggu sampai CPU menjadi bebas. Sistem operasi harus memutuskan melalui scheduler, urutan eksekusi dari proses. Objektif dari multiprogramming atau multitasking adalah untuk membuat beberapa proses berjalan setiap saat. untuk memaksimalkan utilisasi CPU.

Penjadwalan adalah fungsi sistem operasi yang fundamental. Hampir semua sumber daya komputer dijadwalkan sebelum digunakan. CPU, tentu saja, merupakan salah satu sumber daya komputer utama. Karena itu, penjadwalannya menjadi penting sekali pada desain sistem operasi. Penjadwalan CPU menentukan proses mana yang berjalan ketika ada beberapa proses yang dapat dieksekusi.

Penjadwalan CPU adalah penting karena dapat memiliki efek pada utilisasi sumber daya dan kinerja keseluruhan sistem. Secara umum kita ingin mengoptimalkan perilaku sistem (Datta, 2015). Ide dasar multiprogramming adalah mengatur beberapa proses untuk dieksekusi oleh CPU. Yaitu ketika proses mengalokasikan sumber daya bagi CPU.

Ketika ada lebih dari satu proses yang dapat dieksekusi, hanya satu proses yang akan dialokasikan ke sumber daya. Ada beberapa algoritma, FCFS (First Come First Serve), Priority Scheduling, and Shortest Job First (SJF) Scheduling, dan Algoritma Round Robin. Penjadwalan CPU adalah masalah dalam memutuskan proses mana yang siap di antrian harus dialokasikan ke CPU (Putra, 2020).

Penjadwalan dapat merupakan preemptive atau nonpreemptive. Pada penjadwalan nonpreemptive, CPU tidak dapat mengalihkan proses ke proses lain sampai waktu penyelesaian dituntaskan, sedangkan penjadwalan preemptive, proses yang sedang dieksekusi mungkin dipaksa untuk mengalihkan proses sedang berjalan oleh proses lain berdasarkan kriteria penjadwalan (Dhruv, 2019).

Pada Algoritma Round Robin, proses dialihkan ke proses berikutnya berdasarkan waktu quantum yang ditentukan. Sehingga Algoritma Round Robin merupakan algoritma preemptive. Jadi kita dapat menyimpulkan bahwa algoritma penjadwalan yang baik untuk sistem waktu nyata dan time sharing harus memiliki karakteristik berikut: Context switching yang minimum Utilisasi CPU yang maksimum Throughput yang maksimum Turnaround time yang minimum Waktu tunggu yang minimum, dan Waktu respons

yang minimum.

METODE PENELITIAN Metode merupakan teknik yang digunakan pada pemecahan suatu masalah untuk menghasilkan hasil yang sesuai dengan yang diharapkan. Penulis menganalisis algoritma round robin dalam jurnal ini berdasarkan berbagai sumber, jurnal dan sumber-sumber lain untuk digunakan dalam penelitian ini, semuanya terpercaya dan digunakan dalam penulisan jurnal penelitian ini, semuanya tentu saja berkaitan dengan algoritma round robin yang kita bahas di sini.

Metode penelitian pada jurnal ini adalah studi pustaka yang mendalam. Literatur studi dilakukan dari berbagai sumber literatur dan jurnal. Beberapa penelitian terkait dibandingkan berdasarkan literatur. Ada beberapa macam varian Algoritma Round Robin di literatur, salah satunya adalah dengan waktu quantum yang bervariasi.

Namun dalam studi kasus dalam jurnal ini dilakukan dengan waktu quantum yang tetap. Dua studi kasus Algoritma Round Robin dibahas secara mendalam satu persatu. Round robin adalah metode pada penjadwalan proses yang membagi semua proses dengan **porsi waktu yang sama**. Proses dimana waktu prosesnya lebih pendek akan selesai lebih dahulu, daripada proses yang waktu prosesnya lebih lama.

Metode ini membagi proses dengan quantum yang sama pada setiap proses. Berapa lama nilai quantum terhadap proses telah ditentukan sebelumnya. Tidak ada prioritas dalam proses dan diberikan quantum atau time-slice yang sama. Round robin dibuat sehingga memenuhi kebutuhan time-sharing. Dalam jurnal ini semua proses memiliki waktu quantum yang sama. Keuntungan **Algoritma Round Robin adalah** sebagai berikut (J. R.

Indusree, 2017): Starvation pada proses dapat dihindari, karena setiap proses mendapatkan giliran untuk dieksekusi oleh CPU. Proses dengan burst-time lebih kecil mendapat keuntungan karena dieksekusi lebih dulu dibandingkan dengan algoritma yang lain. **Algoritma Round Robin adalah** preemptive. Tujuan penjadwalan sistem adalah untuk : Throughput: Throughput adalah jumlah proses yang **diselesaikan dalam satu unit waktu**.

Algoritma penjadwalan harus didesain sedemikian rupa sehingga throughput sistem menjadi maksimal. Utilisasi CPU: Ini adalah persentase waktu bahwa CPU sibuk dalam mengeksekusi proses. Tujuan dasar penjadwalan adalah menjaga prosesor sibuk sepanjang waktu. Fairness: Semua proses dalam sistem harus diperlakukan dalam cara yang sama kecuali ada beberapa proses yang disukai atau diprioritaskan.

Jika seperti ini, semua proses dengan prioritas lebih rendah harus diabaikan untuk menghindari starvation. Context Switching: Adalah prosedur menyimpan status proses aktif dan memulihkan status proses lain untuk CPU ketika sistem harus mulai mengeksekusi proses berikutnya. Context switching adalah overhead dari sistem dan membuat waktu tidak efektif dari CPU. Algoritma penjadwalan harus didesain untuk meminimalkan context switching.

Jurnal ini diatur dalam bab-bab berikut ini. Jurnal ini dibagi menjadi lima bab. Bab 1 adalah Pendahuluan. Di bab satu ini dibahas konsep penjadwalan sistem operasi secara umum. Bab dua adalah Metode Penelitian. Pada bab ini dibahas metode penelitian yang dilakukan. Yaitu literatur studi yang mendalam. Bab 3 merupakan pembahasan. Pada pembahasan ini dianalisis dua case studi untuk Algoritma Round Robin. Bab 5 adalah Kesimpulan.

Dan yang terakhir adalah Daftar Pustaka. HASIL DAN PEMBAHASAN Algoritma Round Robin didesain khususnya untuk sistem time-sharing. Masing-masing proses ditetapkan interval waktu, yang disebut waktu quantum. Masing-masing proses disediakan waktu tetap untuk mengeksekusi sesuai waktu quantum. Waktu quantum umumnya adalah 10 hingga 100 mdetik. Antrian siap diperlakukan sebagai antrian sirkuler (Arpita Sharma, 2015).

Sehingga proses dialihkan berurutan dari P1 hingga P5 dan kembali ke P1 hingga ke P5 kembali secara sirkuler. Studi Kasus 1 Diberikan lima proses P1 sampai P5 pada tabel 1 di bawah ini. Dengan burst-time masing-masingnya. Pada studi kasus ini diberikan waktu quantum sama dengan 25 mdetik. Tabel 1.

Lima Proses dan masing-masing Burst-Time CPU dengan waktu quantum = 25 mdetik
Nama Proses _Burst-Time CPU (mdetik) _P1_20 _P2_34 _P3_25 _P4_62 _P5_67
__ Perhatikan Gambar 1. Gantt Chart studi kasus 1. Proses akan bergiliran dari P1 ke P5 dan kembali ke P1 lagi, dengan urutannya. Proses dimulai dengan P1 dengan burst-time adalah 20.

Setelah itu adalah P2, dengan burst-time adalah 34, sehingga tersisa 9, dengan quantum 25 sehingga sampai di 45. P3, 45 ditambah 25 menjadi 70, dengan sisa burst-time 0, berarti proses P3 selesai. Kemudian P4 masuk sampai 95. Pada P4 sisa proses adalah 37. Kemudian beralih ke P5 sampai ke 120, dengan sisa burst-time 42.

Dari P5 beralih ke P1 kembali, namun P1 sudah selesai dieksekusi, sehingga masuk P2, sisa P2 adalah 9, hingga sampai ke 129. Dari P2 beralih ke P3, namun P3 sudah selesai, sehingga lanjut ke P4. P4 bersisa 37, sehingga sampai ke 154, dengan sisa P4 adalah 12.

Dari P4 beralih ke P5, sisa P5 adalah 42, sehingga dikurangi 25 menjadi 17. P4 beralih ke P5 pada 179. Sistem secara sirkuler berlanjut ke P1 kembali.

P1 sudah selesai, lanjut P2, P2 juga sudah selesai, sehingga lanjut urutan ke P3, P3 juga sudah selesai lanjut ke P4. Menjadi 191, karena sisa P4 adalah 12. Akhirnya P5 masuk dengan menjadi 208. P1 P2 P3 P4 P5 P2 P4 P5 P4 P5 0 20 45 70 95 120 129 154 179 191 208 Gambar 1. Gantt Chart Studi Kasus 1 dengan quantum = 25 Untuk menghitung waktu tunggu setiap proses perhatikan Tabel 2.: Tabel 2.

Menghitung waktu tunggu rata-rata proses $\text{Proses_Waiting Time} = \frac{P1_0 + P2_20 + (120-45) + 95 + P3_45 + P4_70 + (129-95) + (179-154) + 129 + P5_95 + (154-120) + (191-179)}{5} = 141$ Sehingga waktu tunggu rata-rata adalah $(0+95+45+129+141)/5=82$ mdetik. Untuk menghitung rata-rata Turn Around Time perhatikan tabel 3. Di bawah ini: Tabel 3. Menghitung rata-rata Turn Around Time Proses Waktu tiba Burst-time Waktu mulai Waktu selesai Turn Around Time P1 0 20 20 20 P2 0 34 20 129 129 P3 0 25 45 70 70 P4 0 62 70 191 191 P5 0 67 95 208 208 Sehingga rata-rata Turn Around Time adalah $(20+129+70+191+208)/5=123,6$ mdetik.

Studi Kasus 2 Pada studi kasus 2 ini, perhatikan Tabel 4. di bawah ini. Proses dan burst-time sama, namun waktu quantum yang digunakan adalah 15. Lebih sedikit waktu quantum maka context switching akan menjadi lebih banyak. Tabel 4. Lima Proses dan masing-masing Burst-Time CPU dengan waktu quantum = 15 Nama Proses Burst-Time CPU P1 20 P2 34 P3 25 P4 62 P5 67 Proses P1 dimulai, karena quantum 15 maka proses bersisa 5. Kemudian proses P2 masuk, dan proses P2 bersisa 19. Proses P3 masuk dan bersisa 10.

Kemudian P4 masuk dan bersisa 47. Setelah itu proses P5 masuk dan bersisa 52. Proses kemudian kembali ke P1 lagi, secara sirkuler. P1 masuk dan tidak bersisa, dari 75 menjadi 80 karena P1 hanya bersisa 5. Kemudian P2 masuk dari 19 menjadi bersisa 4, dengan data di gantt chart menjadi 95. Kemudian P3 masuk sebanyak 10, gantt chart menjadi 105. Berikutnya P4 masuk, sisa sebelumnya 47, sehingga gantt chart menjadi 120. Dengan proses bersisa 32.

Berikutnya P5 masuk, status sekarang adalah 52, sehingga sisa menjadi 37, pada gantt chart menjadi 135. Kemudian proses beroperasi secara sirkuler, kembali ke P1. P1 status tidak ada lagi proses tersisa sehingga dilewatkan. P2 masuk dengan burst-time tersisa 4, sehingga di gantt chart menjadi 139 dari 135. Kemudian proses beralih ke P3. P3 ternyata sudah selesai tidak ada burst-time tersisa, sehingga lanjut ke P4.

P4 masuk, dari tersisa 32 menjadi tersisa 17, di gantt chart menjadi 154, yaitu ditambahkan 139 tambah 15 menjadi 154. Berikutnya P5 masuk dengan sisa yang ada 37, sehingga sisa menjadi 22, di gantt chart menjadi 169. Kemudian proses secara sirkuler kembali ke P1. Pada P1 proses sudah selesai, sehingga lanjut ke P2. P2 juga sudah selesai sehingga lanjut ke P3, P3 juga sudah selesai, sehingga dilanjutkan ke P4.

Pada P4 sisa burst-time adalah 17, sehingga di gantt chart dari 169 menjadi 184. Sisa burst-time pada P4 ini adalah 2. Proses kemudian berlanjut ke P5. Pada P5, sisa proses yang ada 22, sehingga di gantt chart menjadi 199, dengan burst-time tersisa 7. Proses kemudian secara sirkuler balik lagi ke P1. P1 sudah selesai, sehingga lanjut ke P2. P2 sudah selesai sehingga lanjut ke P3, P3 sudah selesai sehingga lanjut ke P4.

Pada P4 burst-time tersisa 2, sehingga di gantt chart menjadi 201 dari 199. Berikutnya P5, yang hanya tersisa 7, sehingga di gantt chart menjadi 208 dari 201. Akhirnya semua proses selesai. P1_P2_P3_P4_P5_P1_P2_P3_P4_P5_P2_P4_P5_P4_P5_P4_P5 __ 0 15 30 45 60 75 80 95 105 120 135 139 154 169 184 199 201 208 Gambar 2.

Gantt Chart Studi Kasus 2 dengan waktu quantum = 15 Untuk perhitungan waktu tunggu rata-rata perhatikan Tabel 5. di bawah ini: Tabel 5. Menghitung waktu tunggu rata-rata proses Nama Proses _Waktu Tunggu _P1_0+(75-15)=60 _P2_15+(80-30)+(135-95)=109 _P3_30+(95-45)=80 _P4_45+(105-60)+(139-120)+(169-154)+(199-184)=139 _P5_60+(120-75)+(154-135)+(184-169)+(201-199)=141 __ Maka waktu tunggu rata-rata adalah $(60+109+80+139+141)/5=105,8$ mdetik.

Untuk menghitung rata-rata Turn Around Time perhatikan Tabel 6. di bawah ini: Tabel 6. Menghitung rata-rata Turn Around Time Proses _Waktu Tiba _Burst-time _Waktu mulai _Waktu selesai _Turn Around Time __P1_0_20_0_80_80 __P2_0_34_15_139_139 __P3_0_25_30_105_105 __P4_0_62_45_201_201 __P5_0_67_60_208_208 __ Sehingga rata-rata Turn Around Time adalah $(80+139+105+201+208)/5=146,6$ mdetik. KESIMPULAN Context switching yang lebih banyak terjadi pada waktu quantum = 15.

Pada studi kasus 2 dengan waktu quantum 15, ada sebanyak 17 context switching, sementara pada waktu quantum = 25, pada studi kasus 1, hanya terdapat 10 context switching. Total burst-time untuk waktu quantum = 25 pada studi kasus 1 adalah 208 mdetik, sedangkan pada waktu quantum = 15, adalah juga sama 208 mdetik, nilai ini harus sama, yang membuktikan bahwa perhitungan gantt chart sudah benar pada kedua studi kasus.

Waktu tunggu rata-rata untuk waktu quantum=25 pada studi kasus 1 adalah 82 mdetik,

sedangkan waktu tunggu rata-rata untuk studi kasus 2, dengan waktu quantum 15 adalah 105,8 mdetik. Sehingga waktu tunggu yang lebih efisien adalah pada quantum yang sama dengan 25. Karena waktu tunggu nya lebih kecil. Rata-rata Turn Around Time untuk waktu quantum = 25 pada studi kasus 1 adalah 123,6, sedangkan rata-rata Turn Around Time untuk waktu quantum = 15 pada studi kasus 2 adalah 146,6 mdetik.

Sehingga rata-rata Turn Around Time yang lebih efisien adalah pada waktu quantum yang sama dengan 25, karena rata-rata Turn Around Time nya lebih kecil. SARAN Diusulkan untuk studi kasus Algoritma Round Robin berikutnya adalah dengan menggunakan waktu quantum yang bervariasi, sehingga lebih mengefisienkan sistem. DAFTAR RUJUKAN Amar Ranjan Dash, S. K. (2015). An Optimized Round Robin CPU Scheduling Algorithm with Dynamic Time Quantum.

International Journal of Computer Science, Engineering and Information Technology, Vol. 5, No. 1, 7-26. Arpita Sharma, M. G. (2015). Analysis of Adaptive Round Robin Algorithm and Proposed Round Robin Remaining Time Algorithm. International Journal of Computer Science and Mobile Computing, Vol 4. Issue 12, 139-147. Datta, L. (2015).

Efficient Round Robin Scheduling Algorithm with Dynamic Time Slice. International Journal of Education and Management Engineering, 10-19. Dhruv, R. (2019). Round Robin Scheduling Algorithm Based on Dynamic Time Quantum. International Journal of Engineering and Advanced Technology, Volume X, Issue X, 593-595. J. R. Indusree, B. P. (2017). Enhanced Round Robin CPU Scheduling with Burst Time Based Time Quantum. IOP Conference Series: Material Science and Engineering, 1-8. Md.

Sohrawordi, U. A. (2019). A Modified Round Robin CPU Scheduling Algorithm with Dynamic Time Quantum. International Journal of Advanced Research (IJAR), 422-429. Putra, T. D. (2020). Analysis of Preemptive Shortest Job First (SJF) Algorithm in CPU Scheduling. International Journal of Advanced Research in Computer and Communication Engineering, Vol 9, Issue 4, April, 41-45. Tithi Paul, R. H. (2019).

Improved Round Robin Scheduling Algorithm with Progressive Time Quantum. International Journal of Computer Application Volume 178, No. 49, 30-36.

INTERNET SOURCES:

<1% -

<http://publikasi.dinus.ac.id/index.php/technoc/article/downloadSuppFile/3638/531>

<1% - <https://wintra89.blogspot.com/2016/01/makalah-scheduling.html>

<1% - <https://id.collaborate-anz.com/round-robin-scheduling-c-programming>

<1% - <https://www.slideshare.net/ajbs25/teori-antrian-64679367>

<1% - <https://refidelia.blogspot.com/2015/03/>

1% - https://www.researchgate.net/profile/Taqwa-Al-Tameemi/publication/334454781_CPU_scheduling_visualization/links/5d2b9735458515c11c3152cc/CPU-scheduling-visualization.pdf

<1% - https://www.researchgate.net/publication/281615591_Efficient_Round_Robin_Scheduling_Algorithm_with_Dynamic_Time_Slice

<1% - https://www.researchgate.net/publication/341621055_Analysis_of_Preemptive_Shortest_Job_First_SJF_Algorithm_in_CPU_Scheduling

<1% - <http://dahlan.unimal.ac.id/files/diktat/rpl/P5-Sistem-Operasi.ppt>

<1% - <https://blogsistemoperasi.wordpress.com/2017/09/21/fungsi-sistem-operasi-komputer/>

<1% - <https://aderian.blog.institutpendidikan.ac.id/sistem-operasi/>

<1% - <https://svcmistry.org/id/dictionary/what-s-the-difference-between-system-software-and-application-software/>

<1% - <https://qastack.id/unix/17917/what-is-the-relationship-between-system-calls-message-passing-and-interrupts>

1% - <https://galangromadhon.wordpress.com/2011/03/18/manajemen-proses-sistem-operasi/>

<1% - <https://ridhasariyanti636.wordpress.com/2014/04/10/makalah-so/>

<1% - <https://mediekaputra.wordpress.com/2011/03/21/sistem-operasi/>

<1% - <https://yusniaalfisyahrin.wordpress.com/2013/04/09/proses-penjadwalan-preemptive-shortest-job-first-psjf/>

<1% - <https://kurniadiputra.wordpress.com/2019/10/30/algortma-penjadwalan-proses/>

<1% - <https://id.scribd.com/doc/200716687/Modul-Sistem-Operasi>

<1% - https://www.researchgate.net/publication/326735061_Implementasi_Algoritma_Round_Robin_Pada_Sistem_Penjadwalan_Mata_Kuliah_Studi_Kasus_Universitas_Muhammadiyah_Bengkulu

<1% - <https://anyflip.com/pfnd/ikuf/basic/401-450>

<1% - <https://text-id.123dok.com/document/dzx27wvq-implementasi-load-balancing-dengan-metode-round-robin-dalam-software-defined-networking-sdn-menggunakan-controller>

-pox.html

<1% - <http://journal.unublitar.ac.id/ilkomnika/index.php/ilkomnika/article/download/7/3>

<1% - <http://kuliah.dinus.ac.id/ika/so4.html>

<1% - <https://sirulpoliwangi.blogspot.com/2011/07/uas-semester-2-jaringan-1.html>

<1% -

<https://www.lowongandosen.com/2017/02/kriteria-dan-algoritma-penjadwalan-cpu.html>

<1% -

<https://dwinovi08.blogspot.com/2015/09/penjadwalan-prosesor-manajemen-memori.html>

<1% - http://digilib.uinsgd.ac.id/4515/4/4_bab1.pdf

<1% - <http://scholar.unand.ac.id/33557/2/2.%20BAB%20I%20WT%20M.Farizd.pdf>

<1% - <http://library.binus.ac.id/eColls/eThesisdoc/Bab2Doc/TSA-2012-0032%202.doc>

<1% - <https://anyflip.com/pfnd/ikuf/basic/451-497>

<1% - <https://elvandri.wordpress.com/>

<1% -

<https://andreputrawibowo26.wordpress.com/2019/10/17/tugas-4-penjadwalan-proses-pertemuan-3-dan-4/>

<1% -

https://www.academia.edu/3194200/ANALISIS_KINERJA_DAN_PEMETAAN_RUTE_BUS_DAMRI_BANDARA_DENGAN_GIS_OPEN_SOURCE_KOTA_MAKASSAR

<1% - <https://www.eriesjournal.com/index.php/eries/article/view/375>

1% -

<https://1library.net/title/analysis-adaptive-round-robin-algorithm-proposed-remaining-algorithm>

<1% - https://link.springer.com/chapter/10.1007/978-981-10-0557-2_117

1% - <https://scholar.google.com/citations?user=g-7N8yAAAAAJ>

<1% -

https://www.researchgate.net/publication/220670335_RASA_a_new_grid_task_scheduling_algorithm

<1% - <https://www.science.gov/topicpages/i/ictac+lifetime-prediction+round-robin>

1% -

https://www.academia.edu/38653211/A_MODIFIED_ROUND_ROBIN_CPU_SCHEDULING_ALGORITHM_WITH_DYNAMIC_TIME_QUANTUM

<1% -

https://www.researchgate.net/publication/351330519_A_Review_on_the_CPU_Scheduling_Algorithms_Comparative_Study

<1% - <https://link.springer.com/article/10.1007/s41870-021-00630-0>



UNIVERSITAS BHAYANGKARA JAKARTA RAYA
FAKULTAS ILMU KOMPUTER

Kampus I: Jl. Harsono RM No.67, Ragunan, Pasar Minggu, Jakarta Selatan 12550
Telepon: (021) 27808121 - 27808882
Kampus II: Jl. Raya Perjuangan, Bekasi Utara, Jawa Barat
Telepon: (021) 88955882 Fax.: (021) 88955871
Web: www.ubharajaya.ac.id/fasilkom/. Email: fasilkom@ubharajaya.ac.id

SURAT TUGAS

Nomor : ST/095/IX/2021/B-FASILKOM-UBJ

1. Dasar: Kalender Akademik Ubhara Jaya Tahun Akademik 2021/2022.
2. Dalam rangka mewujudkan Tri Dharma Perguruan Tinggi untuk Dosen di Universitas Bhayangkara Jakarta Raya maka dihimbau untuk melakukan penelitian.
3. Sehubungan dengan hal tersebut diatas, maka Dekan Fakultas Ilmu Komputer Universitas Bhayangkara Jakarta Raya menugaskan:

No.	NAMA	NIDN	JABATAN	KETERANGAN
1.	Tri Dharma Putra, S.T., M.Sc.	0302117101	Dosen Tetap Prodi Informatika	Sebagai Penulis Pertama
2.	Rakhmat Purnomo, S.Pd., S.Kom., M.Kom	0322108201	Dosen Tetap Prodi Informatika	Sebagai Penulis Kedua

Membuat Artikel Ilmiah dengan judul '**Analisis Algoritma Round Robin Pada Penjadwalan CPU**' pada Media Jurnal Ilmiah Teknologi Informasi Asia, Vol. 15, No. 2, September 2021, E-ISSN : 2580-8397.

4. Demikian penugasan ini agar dapat dilaksanakan dengan penuh rasa tanggung jawab.

Bekasi, 01 September 2021
DEKAN FAKULTAS ILMU KOMPUTER


Dr. TYASTUTI SRI LESTARI, S.Si., M.M
NIP: 1408206

Paraf:

1. Ka. Prodi Informatika 