**ANALISIS PENINGKATAN KINERJA FTP SERVER MENGGUNAKAN LOAD BALANCING PADA CONTAINER**

Januar Al Amien, S.Kom,. M.Kom1, Doni Winarso, S.Kom,. M.Kom2

1Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau

2Sistem Informasi, Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau

[1januaralamien@umri.ac.id](mailto:1januaralamien@umri.ac.id), [2doniwinarso@umri.ac.id](mailto:2doniwinarso@umri.ac.id)

# Abstract

*Cloud computing is a technology that answers the challenge of the need for efficient computing technology. There are many things that can be implemented using cloud computing technologies such as web services, storage services, applications and others. Use of cloud computing using container technology can help in the management of applications and optimize the use of resources in the form of memory and processor usage on the server. In this research docker containers implemented by service of FTP (File Transfer Protocol). The FTP service is made into 3 containers within a single server computer. To handle load problems performance on the FTP server against overload requests, load balancing is used. Load balancing is a method to improve performance while reducing the performance load on FTP servers. Based on the test results, the use of multi container and load balancing in the FTP server in load with two algorithm least connection and raound robin handling has result of smaller memory usage and utilization of processor usage evenly.* *Both algorithms are recommended for handling loads for FTP servers and will be more efficient when applied to servers with the same specifications and loads*

***Keywords****: Cloud Computing, Docker, FTP, Load Balancing, HAProxy, Least Connection, Round Robin.*

# Abstrak

Cloud computing merupakan teknologi yang menjawab tantangan akan kebutuhan teknologi komputasi yang efisien. Terdapat banyak hal yang dapat diimplementasikan menggunakan teknologi cloud computing seperti web service, layanan penyimpanan, aplikasi dan lain-lain. Penerapan cloud computing dengan menggunakan teknologi container dapat membantu dalam pengelolaan aplikasi serta mengoptimalkan penggunaan sumber daya berupa penggunaan memory dan prosesor pada server. Dalam penelitian ini penerapan docker container diimplementasikan menggunakan layanan aplikasi FTP (File Transfer Protocol). Layanan FTP dibuat menjadi 3 container didalam satu computer server. Untuk menangani permasalahan beban kinerja pada FTP server terhadap permintaan yang terlalu berat (overload) digunakan load balancing. Load balancing merupakan metode untuk meningkatkan kinerja sekaligus mengurangi beban kinerja pada FTP server. Berdasarkan hasil pengujian, penerapan multi container serta load balancing didalam FTP server dalam penanganan beban dengan dua algortima least connection dan round robin memiliki hasil penggunaan memory yang lebih kecil dan pemanfaatan penggunaan prosesor yang merata kedua algoritma tersebut direkomendasikan untuk penanganan beban untuk ftp server dan akan lebih efisien apabila diterapkan pada server dengan spesifikasi dan beban yang sama.

**Kata Kunci**: *Cloud Computing, Docker, FTP, Load Balancing, HAProxy, Least Connection, Round Robin .*

# Pendahuluan

Universitas Muhammadiyah Riau program studi Teknik Informatika, merupakan sebuah instansi yang memerlukan sebuah sistem penyimpanan data terpadu repositori file, untuk para dosen dengan menggunakan ftp yaitu protokol yang digunakan untuk mentransfer file dari satu host ke host lain [1]. protokol ini akan divirtualisasikan menjadi 3 server da

lam satu server fisik, penerapan akan menggunkan aplikasi Docker *Container* untuk menangani *virtual server* dan sifatnya private, Private cloud merupakan sebuah mekanisme penyediaan resource IT secara menyeluruh, namun akan dideploy secara terpisah dari public cloud dan hanya bisa diakses melalui jaringan private, sehingga memiliki tingkat security yang lebih baik. Model ini cocok untuk perusahaan dengan skala enterprise yang sangat memperhatikan privasi dan keamanan data [2], dengan memanfaatkan sumber daya server yang di miliki oleh instansi menjadi solusi untuk mengatasi permasalah tersebut.

Docker *container* tidak seperti seperti Virtual mesin, dimana docker tidak menggunakan *hardware* atau virtualisasi. Program yang berjalan pada docker *container* berhubungan langsung dengan kernel linux pada *host* sistem operasi. *Container* memungkinkan mengisolasi lingkungan program, sehingga program dapat berjalan tanpa gangguan dari permasalahan di sistem operasi [3] . Virtualisasi container

Penerapan Load balancing dilakukan menggunakan haproxy untuk menangani kelebihan beban pada aplikasi, ketika beban tersebut menghabiskan seluruh resource yang ada, maka aplikasi akan mengalami kegagalan atau down [3]. Container memiliki keunggulan dari sisi kemudahkan deployment, maintenance, dan ringan, sehingga sangat mungkin untuk melakukan proses load balancing [3]. Fokus utama dalam penelitian ini Ada dua perbandingan algoritma yang akan di buat, algoritma *Least Connection* dan *Round Robin*. Dari kedua algoritma ini akan menjadi perbandingan untuk mengukur kinerja server container agar server bekerja dengan baik.

Dari data yang di dapat dari log, log ini akan membaca Context Switching adalah pertukaran proses pra-dieksekusi dari CPU dengan proses baru [4]. Dengan kata lain, pengalihan konteks adalah berapa kali proses mengalihkan eksekusi Algoritma penjadwalan dapat dioptimalkan dengan mengurangi waktu respons, respon time, waiting time dan dengan memaksimalkan CPU utilisation dan throughtput [4]. Log ini akan di peroses menggunakan Average waiting time untuk penghitungan rata respon time perbandingan algoritma Least Connection [5] dan Round Robin dengan menghitung jumlah dari waktu tunggu di setiap proses.

Berdasarkan penelitian [6] Least Connection Scheduling Algorithm dan Enhanced Round Robin Algorithm hasil menunjukkan bahwa algoritma yang diusulkan bekerja lebih baik. Berdasarkan average procesing time (ms), dimana ada 3 data center , Throttled Load Balancer (TLB) DC1 0,383 ms, DC2 0,472 ms, DC3 0,552 ms. Least connection Scheduling algorithm (LCSA) DC1 0,369 ms, DC2 0,449 ms, DC3 0,537 ms. Enhanced Round Robin Algorithm (ERRA) DC1 0,355 ms, DC2 0426 ms, DC3 0,517 ms.

Berdasarkan penelitian [7] kesimpulanya Round Robin, lebih baik untuk menyesuaikan menunggu rata-rata waktu yang diinginkan, perlihatkan pembagian waktu yang mewakili setiap proses dengan melakukan algoritma Round Robin dalam waktu kuantum yang berbeda nilai-nilai. Waktu kuantum memengaruhi hasil waktu tunggu rata-rata. Jika waktu kuantum lebih besar, Average Waiting Time akan menghasilkan nilai yang kecil, tetapi proses yang telah ada dalam antrian siap akan membutuhkan lebih banyak waktu untuk mendapatkan giliran. Algoritma Round Robin akan lebih efisien apabila diterapkan pada server dengan spesifikasi yang sama dan beban load yang juga sama[8]

# Metode Penelitian

Berdasarkan gambar 4 dibawah Kerangka konseptual merupakan suatu bentuk kerangka berpikir yang dapat digunakan sebagai pendekatan ilmiah dalam memecahkan masalah yang menggambarkan hubungan antar variabel dalam proses analisis.

* 1. Pengumpulan data

Pengumpulan data adalah merupakan langkah awal dalam melakukan penelitian yang akan di teliti. Penelitian ini menggunakan teknik observasi, wawancara, dan studi literatur sebagai berikut:

1. Observasi

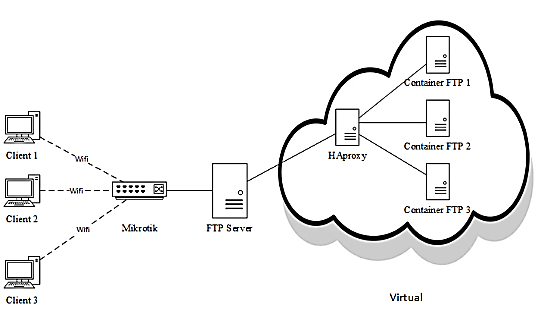
Teknik Observasi dilakukan untuk memperoleh data dengan cara mengamati secara langsung obyek yang berkaitan dengan kegiatan penelitian. Pada tahap ini peneliti melakukan pengamatan pada kegiatan penelitian terhadap kinerja container pada FTP server.

1. Studi Pustaka

Pada tahap ini dilakukan dengan pengumpulan informasi yang diperlukan dalam tahap pembuatan perangkat lunak. Informasi yang diperlukan diperoleh dengan mempelajari jurnal penelitian dan buku maupun browsing internet yang berkaitan dengan Cloud Computing, Docker, dan FTP server.

* 1. Perancangan sistem

Topolology jaringan menggambarkan bentuk konsep komputer yang terhubung ke jaringan yang saling terkoneksi. Adapun bentuk rancangan topology jaringan pada penelitian yang akan dibuat seperti pada Gambar 5.



Gambar 2. Topologi Jaringan

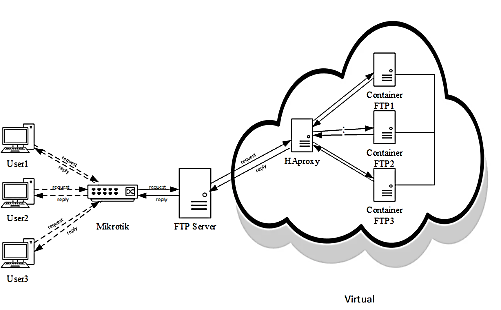
Berdasarkan gambar 5, didalam FTP server terdapat 3 *container* FTP dan HAProxy digunakan sebagai balancer yang akan menangani proses load balancing. Dari ketiga user yang melakukan *request* terhadap FTP server, proses *request* tersebut akan dibagikan ke dalam 3 *container* FTP yang akan membuat penanganan *request* yang dilakukan oleh 3 user dibagi rata kedalam *container* FTP.

* 1. Implementasi

bertujuan untuk mengimplementasi sinkronisasi data pada setiap container dan Perhitungan berlanjut sampai proses terakhir. Ini menghasilkan total waktu tunggu dan akhirnya waktu tunggu rata-rata diperoleh [7]. Data ini merujuk kepada format log haproxy.txt di website haproxy, yang nantinya data ini sudah di kelompokan sesuai format log haproxy

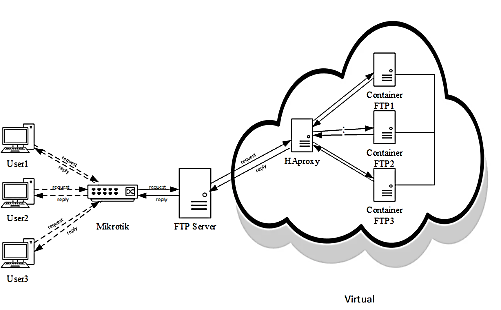
1. Skenario 1

Pengujian dilakukan dengan menggunakan 1 account user yang terdaftar pada 3 container FTP dan melakukan pengujian 3 percobaan login dan upload Video ukuran file 1,08GB, kedalam container FTP di lakukan pengamatan monitoring docker stats untuk baban CPU dan pemakaian memori dan Log file dari haproxy sebagai perbandingan Algoritma Least Connection dan Round Robin dihitung Average Waiting Time.

  
Gambar 3. Skenario 1 uplod Client Video

1. Skenario 2

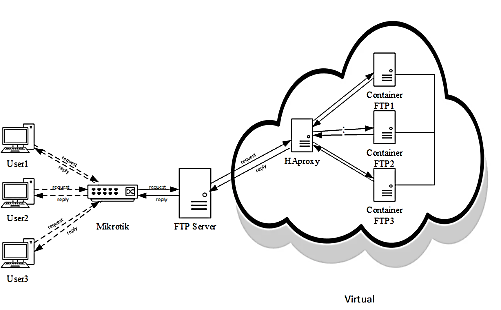
Pengujian dilakukan dengan menggunakan 1 account user yang terdaftar pada 3 container FTP dan melakukan pengujian 3 percobaan login dan upload RAR ukuran file 1,78GB, kedalam container FTP di lakukan pengamatan monitoring docker stats untuk baban CPU dan pemakaian memori dan Log file dari haproxy sebagai perbandingan Algoritma Least Connection dan Round Robin dihitung Average Waiting Time.



Gambar 4. Skenario 2 uplod Client File RAR

1. Skenario 3

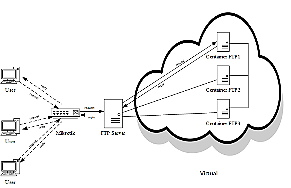
Pengujian dilakukan dengan menggunakan 1 account user yang terdaftar pada 3 container FTP dan melakukan pengujian 3 percobaan login dan upload Document ukuran file 1,17GB, kedalam container FTP di lakukan pengamatan monitoring docker stats untuk baban CPU dan pemakaian memori dan Log file dari haproxy sebagai perbandingan Algoritma Least Connection dan Round Robin dihitung Average Waiting Time.



Gambar 5. Skenario 3 uplod Client File DOC

1. Skenario 4

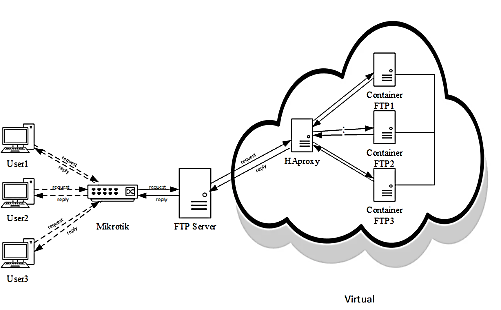
Pengujian dilakukan dengan menggunakan 1 account user yang terdaftar pada 3 container FTP dan melakukan pengujian 3 percobaan login dan upload PDF ukuran file 1,51GB, kedalam container FTP di lakukan monitoring docker stats untuk baban CPU dan pemakaian memori dan Log file dari haproxy sebagai perbandingan Algoritma Least Connection dan Round Robin dihitung Average Waiting Time.



Gambar 6. Skenario 4 uplod Client File PDF

* 1. Analisa

Bertujuan mendapatkan hasil kinerja penggunaan CPU dan memory pada saat penanganan load balancing dengan menggunakan Algoritma Least Connection dan Round Robin dengan 1 user yang melakukan aktifitas upload file kedalam 3 server container FTP di lakukan pengamatan monitoring docker stats dan Log file dari haproxydengan melakukan 3 kali percobaan .



Gambar 7. Topology Skenario empat Pengujian Kinerja Load Balancing Container FTP

Untuk perhitungna Proses adalah, Waiting Time adalah WT [7].

(2)

# 3. Hasil dan Pembahasan

Pada pengujian ini mengacu kepada skenario pengujian yang telah dirancang yang sudah di jelas kan di masing-masing skenario. Data diambil dari log haproxy data yang diambil untuk sample yaitu total Least Connection 5667, untuk Total Round Robin 5720 tota tercatat di log haproxy 11.387. Untuk Penghitungan rata-rata CPU dan Memory di lakukan pengamatan di docker stats.

Untuk mencari Avarage Waiting Time (AWT), terlebih dahulu jumlahkan Waiting Time (WT) adalah penjumlahan dari sama dengan “Total Time” (tt) adalah total waktu dalam milidetik yang terlewati antara penerimaan dan penutupan terakhir, Total Proses diambil dari jumlah "termination\_state" adalah kondisi sesi ketika sesi berakhir Status sesi pada pemutusan. Total Time dan Termination\_state di dapat dari format log haproxy.

TWT = WT1+WT2+WT3+..WTn

Total Waiting Time (TWT) di dapat dari pejumlahan dari Total Time (tt) sama dengan Waiting Time (WT)

Nantinya kita dapat melihat dari hasil skenario yang telah dibuat berdasarkan urutan uploud file video, RAR, Doc dan PDF

* 1. Pengujian Skenario satu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabel 1. Average CPU Usage and memory Least Connection | | | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Percobaan | CPU Usage | | | Total | | s1 | s2 | s3 | | Video 1 | 15,21% | 15,18% |  | 15,20% | | Video 2 |  | 19,60% | 15,27% | 17,44% | | Video 3 | 18,04% | 29,35% |  | 23,70% | | **Total** | **16,63%** | **21,38%** | **15,27%** | **18,78%** | | Percobaan | Memory Usage | | | Total | | s1 | s2 | s3 |  | | Video 1 | 0,10% | 0,13% | 0,20% | 0,18% | | Video 2 | 0,17% | 0,23% | 0,14% | 0,19% | | Video 3 | 0,17% | 0,26% | 0,13% | 0,17% | | **Total** | **0,15%** | **0,21%** | **0,16%** | **0,18%** |     Tabel 2. Average CPU Usage and memory Round Robin | | | | |
| **Percobaan** | CPU Usage | | | Total |
| s1 | s2 | s3 |
| **Video 1** | 16,92% |  |  | 22,10% |
| **Video 2** | 14,92% | 29,28% |  | 21,98% |
| **Video 3** | 15,13% |  | 28,83% | 24,59% |
| **Total** | **15,66%** | **29,28%** | **28,83%** | **22,89%** |
| **Percobaan** | Memory Usage | | | Total |
| s1 | s2 | s3 |
| **Video 1** | 0,31% | 0,33% | 0,22% | 0,27% |
| **Video 2** | 0,30% | 0,31% | 0,20% | 0,27% |
| **Video 3** | 0,30% | 0,30% | 0,20% | 0,27% |
| **Total** | **0,30%** | **0,31%** | **0,21%** | **0,27%** |

Tabel 3. Amount Total Time(ms) Least Connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Percobaan** | **Amount WaitingTime (ms) tt** | | | TWT |
| S1 | S2 | S3 |
| **Video 1** | 204928 | 104053 | 70707 | 379688 |
| **Video 2** | 0 | 104059 | 150849 | 254908 |
| **Video 3** | 167834 | 104043 | 70262 | 342139 |

Tabel 4. . Hitung termination\_state sama dengan Total proses Least Connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Percobaan** | **Hitung Proses** | | | Total Proses |
| S1 | S2 | S3 |
| **Video 1** | 3 | 2 | 2 | 7 |
| **Video 2** | 0 | 2 | 2 | 4 |
| **Video 3** | 3 | 2 | 1 | 6 |

Tabel 5. Amount Time(ms) Round Robin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Percobaan** | **Amount WaitingTime (ms) tt** | | | TWT |
| S1 | S2 | S3 |
| **Video 1** | 256947 | 0 | 0 | 256947 |
| **Video 2** | 104033 | 182328 | 63690 | 350051 |
| **Video 3** | 301858 | 0 | 333588 | 635446 |

Tabel 6. . Hitung termination\_state sama dengan Total proses Round Robin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Percobaan** | **Hitung Proses** | | | Total Proses |
| S1 | S2 | S3 |
| **Video 1** | 4 | 0 | 0 | 4 |
| **Video 2** | 2 | 4 | 1 | 7 |
| **Video 3** | 4 | 0 | 6 | 10 |

* 1. Pengujian Skenario dua

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabel 7. Average CPU Usage and memory Least Connection | | | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Percobaan | CPU Usage | | | Total | | s1 | s2 | s3 | | RAR 1 |  | 28,83% | 14,45% | 21,64% | | RAR 2 | 15,40% | 28,74% | 16,50% | 20,21% | | RAR 3 | 29,52% | 30,24% | 25,85% | 28,54% | | **Total** | **22,46%** | **29,27%** | **18,93%** | **23,46%** | | Percobaan | Memory Usage | | | Total | | s1 | s2 | s3 | | RAR 1 | 0,20% | 0,28% | 0,14% | 0,21% | | RAR 2 | 0,21% | 0,32% | 0,15% | 0,23% | | RAR 3 | 0,22% | 0,33% | 0,21% | 0,25% | | **Total** | **0,21%** | **0,31%** | **0,17%** | **0,23%** |     Tabel 8. Average CPU Usage and memory Round **Robin** | | | | |
| **Percobaan** | CPU Usage | | | Total |
| s1 | s2 | s3 |
| **RAR 1** | 14,48% | 15,24% |  | 20,03% |
| **RAR 2** | 30,64% | 14,97% | 14,48% | 15,16% |
| **RAR 3** | 14,63% | 14,89% | 15,96% | 16,72% |
| **Total** | **19,92%** | **15,03%** | **15,22%** | **17,30%** |
| **Percobaan** | Memory Usage | | | Total |
| s1 | s2 | s3 |
| **RAR 1** | 0,34% | 0,32% | 0,17% | 0,27% |
| **RAR 2** | 0,32% | 0,30% | 0,18% | 0,27% |
| **RAR 3** | 0,34% | 0,30% | 0,17% | 0,27% |
| **Total** | **0,33%** | **0,31%** | **0,17%** | **0,27%** |

Tabel 9. Amount Time(ms) Least Connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Amount WaitingTime (ms) tt** | | | TWT |
| S1 | S2 | S3 |
| RAR 1 | 85584 | 208091 | 158072 | 451747 |
| RAR 2 | 163084 | 104095 | 183292 | 450471 |
| RAR 3 | 104052 | 236578 | 109039 | 449669 |

Tabel 10. Hitung termination\_state sama dengan Total proses Least Connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Percobaan** | **Total Proses** | | | Total Proses |
| S1 | S2 | S3 |
| **RAR 1** | 2 | 4 | 3 | 9 |
| **RAR 2** | 3 | 2 | 4 | 9 |
| **RAR 3** | 2 | 5 | 2 | 9 |

Tabel 11. Amount Time(ms) Round Robin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Amount WaitingTime (ms) tt** | | | TWT |
| S1 | S2 | S3 |
| RAR 1 | 0 | 104033 | 78864 | 182897 |
| RAR 2 | 183103 | 104034 | 161927 | 449064 |
| RAR 3 | 104040 | 182183 | 162050 | 448273 |

Tabel 12. . Hitung termination\_state sama dengan Total proses Round Robin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Hitung Proses** | | | Total Proses |
| S1 | S2 | S3 |
| RAR 1 | 0 | 2 | 2 | 4 |
| RAR 2 | 4 | 2 | 3 | 9 |
| RAR 3 | 2 | 4 | 3 | 9 |

* 1. Pengujian Skenario tiga

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabel 13. Average CPU Usage and memory Least Connection | | | | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Percobaan | CPU Usage | | | Total | | s1 | s2 | s3 | | DOC 1 | 8,58% |  | 15,20% | 11,89% | | DOC 2 | 14,80% |  | 7,34% | 11,07% | | DOC 3 | 8,65% | 13,76% |  | 11,21% | | **Total** | **10,68%** | **13,76%** | **11,27%** | **11,39%** | | Percobaan | Memory Usage | | | Total | | s1 | s2 | s3 | | DOC 1 | 0,27% | 0,35% | 0,21% | 0,28% | | DOC 2 | 0,28% | 0,36% | 0,24% | 0,29% | | DOC 3 | 0,37% | 31,00% | 0,25% | 10,54% | | **Total** | **0,31%** | **10,57%** | **0,23%** | **3,70%** |     Tabel 14. CPU Usage and memory Round Robin | | | | | |
| **Percobaan** | CPU Usage | | | Total |
| s1 | s2 | s3 |
| **DOC 1** | 24,66% |  |  | 16,14% |
| **DOC 2** | 16,14% |  |  | 18,31% |
| **DOC 3** | 18,31% |  |  | 6,57% |
| **Total** | **19,70%** | **0,00%** | **0,00%** | **13,67%** |
| **Percobaan** | Memory Usage | | | Total |
| s1 | s2 | s3 |
| **DOC 1** | 0,34% | 0,30% | 0,18% | 0,28% |
| **DOC 2** | 0,35% | 0,31% | 0,17% | 0,27% |
| **DOC 3** | 0,34% | 0,30% | 0,18% | 0,27% |
| **Total** | **0,34%** | **0,30%** | **0,18%** | **0,27%** |

Tabel 15. Amount Time(ms) Least Connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Amount WaitingTime (ms) tt** | | | TWT |
| S1 | S2 | S3 |
| DOC 1 | 155344 | 0 | 282222 | 437566 |
| DOC 2 | 261360 | 54063 | 262437 | 577860 |
| DOC 3 | 101496 | 161306 | 0 | 262802 |

Tabel 16. . Hitung termination\_state sama dengan Total proses Least Connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Count Proses** | | | Total Proses |
| S1 | S2 | S3 |
| DOC 1 | 53 | 0 | 47 | 100 |
| DOC 2 | 49 | 1 | 49 | 99 |
| DOC 3 | 49 | 50 | 0 | 99 |

Tabel 17. Amount Time(ms) Round Robin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Amount WaitingTime (ms) tt** | | | TWT |
| S1 | S2 | S3 | Waktu |
| DOC 1 | 199515 | 54299 | 54039 | 307853 |
| DOC 2 | 202319 | 0 | 54290 | 256609 |
| DOC 3 | 1071879 | 0 | 0 | 1071879 |

Tabel 18. . Hitung termination\_state sama dengan Total proses Round Robin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Hitung Proses** | | | Total Proses |
| S1 | S2 | S3 |
| DOC 1 | 96 | 1 | 1 | 98 |
| DOC 2 | 96 | 0 | 1 | 97 |
| DOC 3 | 98 | 0 | 0 | 98 |

* 1. Pengujian Skenario empat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabel 19. CPU Usage and memory Least Connection | | | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Percobaan | CPU Usage | | | Total | | s1 | s2 | s3 | | PDF1 |  | 8,86% | 8,85% | 8,86% | | PDF2 |  | 8,51% | 7,48% | 8,00% | | PDF3 |  | 8,16% | 8.03% | 8,16% | | Total | 0,00% | 8,51% | 8,17% | 8,34% | | Percobaan | Memory Usage | | | Total | | s1 | s2 | s3 | | PDF1 | 0,32% | 0,40% | 0,40% | 0,37% | | PDF2 | 0,33% | 0,43% | 0,31% | 0,36% | | PDF3 | 0,27% | 0,34% | 0,29% | 0,30% | | Total | 0,31% | 0,39% | 0,33% | 0,34% |   Tabel 26. CPU Usage and memory Round Robin | | | | |
| **Percobaan** | CPU Usage | | | Total |
| s1 | s2 | s3 |
| **PDF1** |  | 8,45% | 8,35% | 9,57% |
| **PDF2** |  | 9,87% | 9,27% | 8,27% |
| **PDF3** |  | 8,22% | 8,32% | 5,83% |
| **Total** | **0,00%** | **8,85%** | **8,65%** | **7,89%** |
| **Percobaan** | Memory Usage | | | Total |
| s1 | s2 | s3 |
| **PDF1** | 0,30% | 0,34% | 0,19% | 0,29% |
| **PDF2** | 0,30% | 0,35% | 0,21% | 0,29% |
| **PDF3** | 0,30% | 0,35% | 0,21% | 0,28% |
| **Total** | **0,30%** | **0,35%** | **0,20%** | **0,29%** |

Tabel 20. Amount Time(ms) Least Connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Amount Time (ms) tt** | | | TWT |
| S1 | S2 | S3 |
| PDF1 | 0 | 301761 | 301023 | 602784 |
| PDF2 | 54066 | 302929 | 303213 | 660208 |
| PDF3 | 300369 | 54078 | 300930 | 655377 |

Tabel 21. . Hitung termination\_state sama dengan Total proses Least Connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Hitung Proses** | | | Total Proses |
| S1 | S2 | S3 |
| PDF1 | 0 | 893 | 894 | 1787 |
| PDF2 | 1 | 894 | 891 | 1786 |
| PDF3 | 876 | 1 | 875 | 1752 |

Tabel 22. Amount Time(ms) Round Robin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Amount WaitingTime (ms) tt** | | | TWT |
| S1 | S2 | S3 | Waktu |
| PDF1 | 0 | 164227 | 167020 | 331247 |
| PDF2 | 56644 | 111437 | 111307 | 279388 |
| PDF3 | 0 | 108812 | 108919 | 217731 |

Tabel 23. . Hitung termination\_state sama dengan Total proses Round Robin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan | **Count Proses** | | | Total Proses |
| S1 | S2 | S3 |
| DOC 1 | 96 | 1 | 1 | 98 |
| PDF1 | 0 | 897 | 898 | 1795 |
| PDF2 | 1 | 896 | 897 | 1794 |
| PDF3 | 0 | 898 | 897 | 1795 |

* 1. Pengujian Avarge Waiting Time Least Connection dan Round Robin

Pada pengujian menghitung rata-rata waktu tunggu satu proses dapat dihitung :

Dari total pejumlahan Waiting Time WT dan total proses di hitung berapa kali total WT memproses maka di dapat Total Average Waiting Time.

Tabel 24. Average Waiting Time Least Connection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan File | S1 | S2 | S3 | AWT |
| Video 1 | 68309 | 0 | 0 | 54241 |
| Video 2 | 0 | 52030 | 75425 | 63727 |
| Video 3 | 55945 | 0 | 70262 | 57023 |
| Total | 41418 | 17343 | 48562 | 58330 |
| RAR 1 | 0 | 52023 | 52691 | 50194 |
| RAR 2 | 54361 | 52048 | 45823 | 50052 |
| RAR 3 | 52026 | 47316 | 54520 | 49963 |
| Total | 35462 | 50462 | 51011 | 50070 |
| DOC 1 | 2931 | 0 | 6005 | 4376 |
| DOC 2 | 5334 | 0 | 5356 | 5837 |
| DOC 3 | 2071 | 0 | 0 | 2655 |
| Total | 3445 | 0 | 3787 | 4289 |
| PDF1 | 0 | 338 | 337 | 337 |
| PDF2 | 54066 | 339 | 340 | 370 |
| PDF3 | 0 | 54078 | 344 | 374 |
| Total | 18022 | 18252 | 340 | 1081 |

* 1. Pengujian Avarge Waiting Time Round Robin

Tabel 25. Average Waiting Time Round Robin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Percobaan File | S1 | S2 | S3 | AWT |
| Video 1 | 64237 | 0 | 0 | 64237 |
| Video 2 | 52017 | 45582 | 63690 | 50007 |
| Video 3 | 75465 | 0 | 55598 | 63545 |
| Total | 63906 | 15194 | 39763 | 59263 |
| RAR 1 | 0 | 52017 | 39432 | 45724 |
| RAR 2 | 45776 | 52017 | 53976 | 49896 |
| RAR 3 | 52020 | 45546 | 54017 | 49808 |
| Total | 32599 | 49860 | 49141 | 48476 |
| DOC 1 | 2078 | 54299 | 54039 | 3141 |
| DOC 2 | 2107 | 0 | 54290 | 2645 |
| DOC 3 | 10938 | 0 | 0 | 10938 |
| Total | 5041 | 18100 | 36110 | 5575 |
| PDF1 | 0 | 183 | 186 | 185 |
| PDF2 | 56644 | 124 | 124 | 156 |
| PDF3 | 0 | 121 | 121 | 121 |
| Total | 18881 | 143 | 144 | 462 |

* 1. Hasil grafik CPU dan memory dari masing-masing skenario

Gambar 8. Hasil Video Perbandingan CPU Least Connection dan Round Robin

Gambar 9. Hasil Video Perbandingan Memori Least Connection dan Round Robin

Berdasarkan gambar 8 dan 9 dapat di hasilkan skenario 1 beban pada cpu least connection 18,78% dan round robin 22,89%, dan pemakaian memory least connection 0,18% dan round Robin 0,27% ketika client uplod video ke 3 server virtual container dalam 3 kali percobaan.

Gambar 10. Hasil Video Perbandingan CPU Least Connection dan Round Robin

Gambar 11. Hasil Video Perbandingan Memori Least Connection dan Round Robin

Berdasarkan gambar 10 dan 11 dapat di hasilkan skenario 1 beban pada cpu least connection 23,46% dan round robin 17,30%, dan pemakaian memoryleast connection 0,23% dan round Robin 0,27% ketika client uplod RAR ke 3 server virtual container dalam 3 kali percobaan.

Gambar 12. Hasil Video Perbandingan CPU Least Connection dan Round Robin

Gambar 13. Hasil Video Perbandingan Memori Least Connection dan Round Robin

Berdasarkan gambar 12 dan 13 dapat di hasilkan skenario 1 beban pada cpu least connection 11,39% dan round robin 13,67%, dan pemakaian memoryleast connection 11,39% dan round Robin 0,27% ketika client uplod DOC ke 3 server virtual container dalam 3 kali percobaan.

Gambar 14. Hasil Video Perbandingan CPU Least Connection dan Round Robin

Gambar 15. Hasil Video Perbandingan Memori Least Connection dan Round Robin

Berdasarkan gambar 14 dan 15 dapat di hasilkan skenario 1 beban pada cpu least connection 8,34% dan round robin 7,89%, dan pemakaian memoryleast connection 8,34% dan round Robin 0,29% ketika client uplod PDF ke 3 server virtual container dalam 3 kali percobaan.

* 1. Hasil grafik Average Waiting Time Least Connection Dan Round Robin dari masing-masing skenario

Gambar 16. Hasil Video Perbandingan Avarge Waiting Time

Pada gambar 16 grafik total Avarege Waiting Time ada peningkatan pada round robin yaitu 59263ms sedangkan least connection 58330ms, ketika client uploud file video ke 3 server container.

Gambar 17. Hasil RAR Perbandingan Avarge Waiting Time

Pada gambar 16 grafik total Avarege Waiting Time ada peningkatan pada least connection yaitu 50070ms sedangkan Round Robin 48476ms, ketika client uploud file RAR ke 3 server container.

Gambar 18. Hasil DOC Perbandingan Avarge Waiting Time

Pada gambar 16 grafik total Avarege Waiting Time ada peningkatan pada round robin yaitu 5575ms sedangkan least connection 4289ms, ketika client uploud file DOC ke 3 server container

Gambar 19. Hasil DOC Perbandingan Avarge Waiting Time

Pada gambar 16 grafik total Avarege Waiting Time ada peningkatan pada least connection yaitu 360ms sedangkan least connection 154ms, ketika client uploud file video ke 3 server container

1. **KESIMPULAN**

Berdasarkan hasil yang diperoleh melalui pengamatan dan ujicoba, maka dapat ditarik kesimpulan sebagai berikut:

1. Penerapan server FTP pada Universitas Muhammadiyah Riau program studi Teknik Informatika membantu dalam memenuhi kebutuhan Dosen akan penyimpanan data yang terpusat, dikarenakan masing-masing dosen memiliki akses kedalam sesuai dengan akun masing-masing tanpa tercapur dengan yang lain.
2. Penerapan load balancing dapat mengalihkan service yang mati ke service yang lain, sehingga proses pengiriman data tidak terganggu dikarenakan service yang mati.
3. Penerapan multi container dan balancing pada server FTP membantu mengoptimalkan kinerja pada server FTP dalam penanganan request atau beban yang banyak dan mampu membagi kerja pada setiap container untuk menjaga kestabilan memory dan prosesor yang tetap membuat sistem berjalan baik.
4. Perbandingan dua algoritma least connection dan Round Robin tidak terlalu jauh dari beban CPU dan Pemakaian Memori, begitu juga waktu di bagi dengan proses, algoritma least connection dan Round Robin kedua algortima tergantung kecepatan proses pada beban proses tersebut direkomendasikan untuk penanganan beban untuk ftp server dan akan lebih efisien apabila diterapkan pada server dengan spesifikasi yang sama dan beban yang juga sama.

# Ucapan Terimakasih

Terimakasih kepada kemenristekdikti yang sudah membantu mendanai penelitian sebagai penelitian dosen pemula dan juga terimakasih kepada Universitas muhammadiya riau yang sudah memfasilitasi penelitian untuk pengambilan data ini khususnya fakultas ilmu komputer, prodi Teknik Informatika dan Unit Pelaksana terpadu Teknologi Informasi Pangakalan Data (UPT TIPD).

# Daftar Rujukan

[1] C. M. Babu and O. L. Chandana, “File Transfer Protocol in Cloud Computing,” vol. 2, no. March, pp. 665–667, 2014.

[2] I. Santiko and R. Rosidi, “Pemanfaatan Private Cloud Storage Sebagai Media Penyimpanan Data E-Learning Pada Lembaga Pendidikan,” *J. Tek. Inform.*, vol. 10, no. 2, pp. 137–146, 2018.

[3] M. A. Nugroho and R. Kartadie, “Analisis Kinerja Penerapan Container untuk Load Balancing Web Server,” *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 1, no. 02, pp. 7–15, 2016.

[4] M. M. Tajwar, M. N. Pathan, L. Hussaini, and A. Abubakar, “CPU scheduling with a round robin algorithm based on an effective time slice,” *J. Inf. Process. Syst.*, vol. 13, no. 4, pp. 941–950, 2017.

[5] Y. L. Oktavianus, “Membangun Sistem Cloud Computing dengan Implementasi Load Balancing dan Pengujian Algoritma Penjadwalan Linux Virtual Server pada FTP Server,” *J. Nas. Tek. Elektro*, vol. 2, no. 1, pp. 25–30, 2013.

[6] G. Thejesvi and T. Anuradha, “Distribution of Work Load at Main Controller Level Using Enhanced Round Robin Scheduling Algorithm in A Public Cloud,” *Int. J. Comput. Sci. Eng. (e-IJCSE 2347-269)*, no. 12, pp. 75–78, 2015.

[7] A. P. U. Siahaan, “Comparison Analysis of CPU Scheduling : FCFS, SJF and Round Robin,” *Int. J. Eng. Dev. Res.*, vol. 4, no. 3, pp. 124–131, 2016.

[8] L. Fani, I. Ardy, A. Bhawiyuga, and W. Yahya, “Implementasi Load Balancer Berdasarkan Server Status pada Arsitektur Software Defined Network ( SDN ),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 5, pp. 2135–2143, 2018.