

**ANALISA PERBANDINGAN KINERJA NGINX DAN APACHE
PADA LINGKUNGAN VIRTUAL DAN DOCKER STUDI KASUS
APLIKASI PLN MARKETPLACE**

**Oleh :
Falyan Zuril
065120702**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
BOGOR
2024**

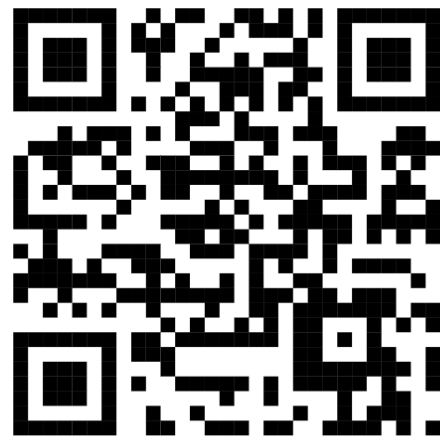
LEMBAR PERSEMBAHAN

Skripsi ini penulis persembahkan kepada ke dua orang tua tercinta saya, yang telah membantu memberikan dukungan dalam berbagai bentuk yang tidak bisa terukur berapapun besarnya. Terima kasih untuk semua dukungan yang telah diberikan, kasih sayang dan pengorbanan yang dilakukan demi memberikan yang terbaik untuk saya.

Terima kasih atas setiap dukungan dan motivasi yang selalu membuat penulis yakin untuk terus berkarya. Tidak ada kata yang cukup untuk menggambarkan betapa besarnya rasa terima kasih ini, Semoga karya sederhana ini bisa menjadi bukti kecil dari dedikasi dan cinta yang telah kalian berikan.



Linked In



Github

HALAMAN PENGESAHAN

Judul : Analisa Perbandingan Kinerja Nginx dan Apache pada
Lingkungan Virtual dan Docker Studi Kasus Aplikasi PLN
Marketplace
Nama : Falyan Zuril
NPM : 065120702

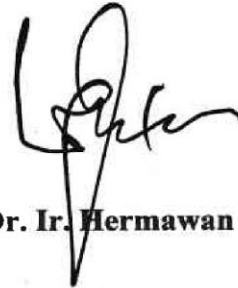
Mengesahkan,

Pembimbing Pendamping
Program Studi Ilmu Komputer
FMIPA – UNPAK



Mohamad Iqbal Suriansyah, M.Kom.

Pembimbing Utama
Program Studi Ilmu Komputer
FMIPA – UNPAK



Dr. Ir. Hermawan Thaheer.

Mengetahui,


Ketua Program Studi Ilmu Komputer.
FMIPA – UNPAK.



Arie Qur'ania, M.Kom.

Dekan

FMIPA – UNPAK



Asep Denih, S.Kom., M.Sc., Ph.D.

PERNYATAAN KEASLIAN KARYA TULIS

Dengan ini saya menyatakan bahwa :

Sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasi atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian-bagian dimana sumber informasinya dicantumkan dengan cara referensi yang semestinya.

Demikian pernyataan ini saya buat dengan sebenar-benarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai dengan peraturan yang berlaku.

Bogor, Agustus 2024



Falyan Zuril
065120702

PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Saya yang bertanda tangan dibawah ini:

Nama : Falyan Zuril
NPM : 065120702
Jukul Skripsi : Analisa Perbandingan Kinerja Nginx dan Apache pada
Lingkungan Virtual dan Docker Studi Kasus Aplikasi PLN
Marketplace

Dengan ini saya menyatakan bahwa Paten dan Hak Cipta dari produk Skripsi dan Tugas Akhir di atas adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dari karya tulis saya kepada Universitas Pakuan.

Bogor, Agustus 2024



Falyan Zuril
065120702

RIWAYAT HIDUP



Falyan Zuril lahir pada tanggal 12 Februari 1998 di Talu. Putra pasangan dari bapak Syahri Edy dan ibu Zuraima, merupakan anak bungsu dari 2 bersaudara. Bertempat tinggal di Kp. Pinang Sei Jernih Talu, Kecamatan Talamau, Kabupaten Pasaman Barat, Provinsi Sumatera Barat. Pendidikan yang pernah di tempuh : Sekolah Dasar di SD Negeri 04 Talamau pada tahun 2004, lulus pada tahun 2010. Kemudian melanjutkan ke Madrasah Tsanawiyah PPMM Sei Jernih Talu, dan lulus pada tahun 2013, meneruskan pendidikan ke SMA Negeri 1 Talamau dan lulus pada tahun 2016. Kemudian melanjutkan pendidikan di Institut Pertanian Bogor pada fakultas Diploma 3 dengan jurusan Teknik Komputer dan lulus pada tahun 2019, dan penulis tercatat sebagai mahasiswa perguruan tinggi swasta Universitas Pakuan Bogor pada Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Program Studi Ilmu Komputer pada tahun 2020. Penulis melaksanakan penelitian untuk menyelesaikan skripsi ini di Laboratorium Workshop agar tercapainya suatu keberhasilan penelitian maka penulis mengikuti arahan yang diberikan dosen pembimbing dengan melaksanakan penelitian pada tanggal 01 Januari sampai dengan 25 April 2024. Dengan ketekunan dan mempunyai motivasi yang tinggi untuk terus belajar dan berusaha, penulis berhasil menyelesaikan pengerjaan tugas akhir skripsi ini. Semoga dengan penulisan skripsi ini mampu memberikan kontribusi positif bagi teknologi baru. Akhir kata penulis mengucapkan rasa syukur yang sebesar-besarnya atas terselesaikannya skripsi yang berjudul “Analisa Perbandingan Kinerja Nginx dan Apache pada Lingkungan Virtual dan Docker Studi Kasus Aplikasi PLN Marketplace”.

RINGKASAN

Di era digital, aplikasi berbasis web menjadi tulang punggung berbagai layanan, termasuk platform e-commerce seperti PLN Marketplace yang dikembangkan oleh PT PLN. Penelitian ini mengevaluasi kinerja dua server web terkemuka, Nginx dan Apache, dalam lingkungan Docker dan Virtual Machine. Dengan menggunakan pendekatan kuantitatif dan metode Network Development Life Cycle (NDLC), penelitian ini menganalisis parameter-parameter Quality of Service (QoS) yang penting seperti latency, jitter, throughput, packet loss, penggunaan CPU, dan konsumsi RAM. Evaluasi kinerja menggunakan alat Apache JMeter dan Apache Bench di tiga skenario pengguna (500, 1000, dan 1500 pengguna). Temuan ini mengungkapkan bahwa Nginx mengungguli Apache dalam menangani permintaan pengguna yang tinggi, terutama dalam hal throughput, waktu respons, dan efisiensi sumber daya. Arsitektur asinkron dan event-driven Nginx memungkinkan stabilitas dan kinerja yang lebih baik, terutama di lingkungan terkontainerisasi seperti Docker. Sebaliknya, Apache menunjukkan keunggulan dalam hal kompatibilitas dengan aplikasi yang kompleks dan dukungan modul yang luas, meskipun kinerjanya kurang optimal di bawah beban pengguna yang berat. Penelitian ini menyimpulkan bahwa Nginx adalah server web yang lebih cocok untuk menangani aplikasi dengan permintaan tinggi seperti PLN Marketplace, terutama di lingkungan Docker. Hasil penelitian ini memberikan rekomendasi berbasis data bagi para pengembang untuk mengoptimalkan pemilihan server web, meningkatkan efisiensi dan skalabilitas sistem berbasis web.

Kata kunci: Apache, Docker, Nginx, Virtual, Web Server

KATA PENGANTAR

Puji syukur kehadirat Allah SWT, karena rahmat dan hidayah- Nya Penulis dapat menyelesaikan skripsi ini berjudul “**Analisa Perbandingan Kinerja Nginx dan Apache pada Lingkungan Virtual dan Docker Studi Kasus Aplikasi PLN Marketplace**”. Penulisan tugas akhir ini merupakan salah satu syarat memperoleh gelar Sarjana Komputer di Program Studi Ilmu Komputer FMIPA UNPAK Bogor.

Dalam penulisan tugas akhir ini, penulis dengan senang hati ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Dr. Ir. Hermawan Thaheer., selaku Pembimbing Utama yang telah memberikan dorongan moril dan motivasi kepada penulis.
2. Mohamad Iqbal Suriansyah, M.Kom, selaku Pembimbing Pendamping yang telah memberikan bimbingan, semangat dan motivasi.
3. Orang tua yang telah memberi pengertian, semangat baik moral maupun materil, serta doa dan kasih sayang kepada anaknya.
4. Teman-teman Ilkom Universitas Pakuan angkatan 2020, dan 2021 yang telah menemani perjuangan penulis dalam menempuh perkuliahan
5. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu yang telah memberikan segala dukungan, semangat, bantuan secara langsung maupun tidak langsung.

Saran dan kritik yang membangun dalam penulisan tugas akhir ini akan diterima dengan senang hati. Mudah-mudahan Allah SWT akan membalas semua kebaikan kepada semua pihak yang membantu. Akhir kata, semoga laporan ini dapat bermanfaat bagi kita semua. Amin.

Bogor , Agustus 2024

Falyan Zuril
065120702

DAFTAR ISI

LEMBAR PERSEMBAHAN	i
HALAMAN PENGESAHAN	ii
PERNYATAAN KEASLIAN KARYA TULIS	iii
PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA	iv
RIWAYAT HIDUP	v
RINGKAKASAN	vi
KATA PENGANTAR	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB I	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.1.1. Tujuan Penelitian	2
1.1.2. Ruang Lingkup Penelitian	2
1.1.3. Manfaat Penelitian	3
BAB II	4
TINJAUAN PUSTAKA	4
2.1. Landasan Teori	4
2.1.1. Docker.....	4
2.1.2. Virtual Machine	4
2.1.3. Web Server	5
2.1.4. Apache	5
2.1.5. Nginx	5
2.1.7. Arsitektur	6
2.1.6. Quality of Service	6
2.1.7. Trthroughput.....	6
2.1.8. Delay	7
2.1.9. Packet Loss	7
2.1.10. Jitter	8
2.1.11. CPU (Central Processing Unit).....	8
2.1.12. RAM (Random Access Memory).....	8
2.1.13. Network Troughput.....	8
2.1.14. Time Taken	9
2.1.15. Netcraft	9
2.1.16. Apache Bench.....	9
2.1.17. Apache Jmeter.....	9
2.1.18. PHP	10
2.2. Penelitian Terdahulu	10

2.3. Tabel Perbandingan	12
BAB III.....	13
METODOLOGI PENELITIAN.....	13
3.1. Metode Penelitian	13
3.1.1. Defenisi Masalah dan Analisis	13
3.1.2. Alat dan Bahan.....	14
3.1.3. Analisis Parameter	15
3.1.4. Perancangan	15
3.1.6. Pengujian	17
3.1.7. Evaluasi.....	17
3.1.8. Dokumentasi	17
3.2. Waktu dan Jadwal Penelitian.....	18
3.2.1. Waktu dan Tempat Penelitian.....	18
BAB IV.....	19
IMPLEMENTASI.....	19
4.1. Pengujian	19
4.1.1. Pengumpulan Data.....	20
4.1.2. Analisis Statistik Data.....	31
BAB V	34
HASIL DAN PEMBAHASAN.....	34
5.1. Hasil Penelitian	34
5.1.1. Hasil Uji T Student	35
5.1.2. Hasil uji T-Hitung dan T-Table	38
5.2. Pembahasan Hasil Penelitian.....	41
5.2.1. Lingkungan docker dan Virtual menggunakan Apache Jmeter.....	41
5.2.2. Lingkungan docker dan Virtual menggunakan Apache Bench	41
BAB VI.....	42
KESIMPULAN DAN SARAN.....	42
6.1. Kesimpulan.....	42
6.2. Saran	42
DAFTAR PUSTAKA	43
LAMPIRAN	45

DAFTAR GAMBAR

Gambar 1 Docker Kontainer	4
Gambar 2 Virtual Machine.....	5
Gambar 3 Netcraft Survey.....	9
Gambar 4 Metode Penelitian.....	13
Gambar 5 Lingkungan Docker.....	15
Gambar 6 Arsitektur lingkungan docker.....	16
Gambar 7 Lingkungan Virtual	16
Gambar 8 Arsitektur lingkungan virtual	17
Gambar 9 Diagram Tree Pengujian Lingkungan Docker.....	19
Gambar 10 Diagram Tree Pengujian Lingkungan Virtual	19
Gambar 11 Hasil uji T Student Apache dan Nginx menggunakan Apache Jmeter .	35
Gambar 12 Grafik hasil Uji T Student Apache dan Nginx Menggunakan Apache Jmeter.....	36
Gambar 13 Hasil uji T Student Apache dan Nginx menggunakan Apache Bench..	37
Gambar 14 Grafik hasil Uji T Student Apache dan Nginx Menggunakan Apache Bench	38
Gambar 15 Grafik T-tabel dan T-hitung throughput dan Delay Apache dan Nginx dengan Apache Jmeter	39
Gambar 16 Grafik T-tabel dan T-hitung Jitter dan Packet Loss Apache dan Nginx dengan Apache Jmeter	39
Gambar 17 Grafik T-tabel dan T-hitung CPU dan RAM Apache dan Nginx dengan Apache Jmeter.....	39
Gambar 18 Grafik T-tabel dan T-hitung Network Throughput Apache dan Nginx dengan Apache Jmeter	40
Gambar 19 Grafik T-tabel dan T-hitung Throughput dan Network Throughput Apache dan Nginx dengan Apache Bench	40
Gambar 20 Grafik T-tabel dan T-hitung Jitter dan Time Taken Apache dan Nginx dengan Apache Bench	40
Gambar 21 Grafik T-tabel dan T-hitung CPU dan RAM Apache dan Nginx dengan Apache Bench.....	41

DAFTAR TABEL

Tabel 1 Kategori Throughput.....	7
Tabel 2 Kategori Delay	7
Tabel 3 Kategori Packet Loss	8
Tabel 4 Kategori Jitter.....	8
Tabel 5 Perbandingan Penelitian.....	12
Tabel 6 Alat dan Bahan.....	14
Tabel 8 Data Pengujian Nginx (Docker – Apache Jmeter) – 500 user	20
Tabel 9 Data Pengujian Nginx (Docker – Apache Jmeter) – 1000 user	20
Tabel 10 Data Pengujian Nginx (Docker – Apache Jmeter) – 1500 user	21
Tabel 11 Data Pengujian Nginx (Docker – Apache Bench) – 500 user.....	21
Tabel 12 Data Pengujian Nginx (Docker – Apache Bench) – 1000 user.....	22
Tabel 13 Data Pengujian Nginx (Docker – Apache Bench) – 1500 user.....	22
Tabel 14 Data Pengujian Nginx (Virtual – Apache Jmeter) – 500 user.....	23
Tabel 15 Data Pengujian Nginx (Virtual – Apache Jmeter) – 1000 user.....	23
Tabel 16 Data Pengujian Nginx (Virtual – Apache Jmeter) – 1500 user.....	24
Tabel 17 Data Pengujian Nginx (Virtual – Apache Bench) – 500 user	24
Tabel 18 Data Pengujian Nginx (Virtual – Apache Bench) – 1000 user	25
Tabel 19 Data Pengujian Nginx (Virtual – Apache Bench) – 1500 user	25
Tabel 20 Data Pengujian Apache (Docker – Apache Jmeter) – 500 user.....	26
Tabel 21 Data Pengujian Apache (Docker – Apache Jmeter) – 1000 user.....	26
Tabel 22 Data Pengujian Apache (Docker – Apache Jmeter) – 1500 user	27
Tabel 23 Data Pengujian Apache (Docker – Apache Bench) – 500 user	27
Tabel 24 Data Pengujian Apache (Docker – Apache Bench) – 1000 user	28
Tabel 25 Data Pengujian Apache (Docker – Apache Bench) – 1500 user	28
Tabel 26 Data Pengujian Apache (Virtual – Apache Jmeter) – 500 user	29
Tabel 27 Data Pengujian Apache (Virtual – Apache Jmeter) – 1000 user	29
Tabel 28 Data Pengujian Apache (Virtual – Apache Jmeter) – 1500 user	30
Tabel 29 Data Pengujian Apache (Virtual – Apache Bench) – 500 user.....	30
Tabel 30 Data Pengujian Apache (Virtual – Apache Bench) – 1000 user.....	31
Tabel 31 Data Pengujian Apache (Virtual – Apache Bench) – 1500 user.....	31
Tabel 32 Rata rata Apache dan Nginx dilingkungan Docker dengan Apache Jmeter	32
Tabel 33 Rata - rata Apache dan Nginx dilingkungan Docker dengan Apache Bench	32
Tabel 34 Rata - rata Apache dan Nginx dilingkungan Virtual dengan Apache Jmeter	33
Tabel 35 Rata - rata Apache dan Nginx dilingkungan Virtual dengan Apache Bench	33
Tabel 36 Hasil Uji T Student Apache dan Nginx Menggunakan Apache Jmeter.....	36
Tabel 37 Hasil Uji T Student Apache dan Nginx Menggunakan Apache Bench	37

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam dunia *modern* popularitas dan kemajuan dalam teknologi web sangat penting bagi setiap organisasi untuk memiliki situs web yang fungsionalnya bagus dan menarik secara visual. Salah satu unsur paling mendasar dari pengembangan web adalah pemrograman web yang dicapai dengan bantuan bahasa pemrograman.

Perkembangan teknologi informasi saat ini khususnya pada bidang telekomunikasi sangat cepat membuat banyak layanan baru yang muncul. Bahkan sekarang semua komunikasi, terutama pada komunikasi data bisa dilakukan dengan sistem jaringan yang telah terintegrasi. Berkembangnya sistem jaringan komputer tidak lepas dari suatu teknologi yang disebut dengan internet. Internet sudah menjadi hal pokok, bahkan bisa dikatakan hal wajib dalam kemajuan era teknologi sekarang. Ketika mengakses internet maka seseorang tidak akan lepas dari apa yang disebut dengan web server, didalam internet ada dua interaksi hubungan yang selalu terjadi yaitu antara client dan server. Oleh karena itu dibutuhkan web server yang handal dalam menangani jumlah permintaan yang banyak dari pengguna internet. Web Server menunggu permintaan dari *client* yang menggunakan browser seperti Netscape Navigator, Internet Explorer, Mozilla, dan program browser lainnya. Jika ada permintaan dari browser, maka Web Server akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser, menurut (Adnan, 2016a) proses ini akan dilakukan seara timbal balik dan terus menerus. Melalui proses ini, web server memuat dan mengirim halaman yang diminta untuk disajikan di browser pengguna.

Diera sekarang aplikasi web sudah dibangun dalam struktur mikroservis. Arsitektur Mikroservis adalah salah satu cara yang membuat satu aplikasi besar dipecah-pecah menjadi beberapa bagian layanan yang lebih kecil dan terhubung satu sama lainnya dengan menggunakan API (*Application Programming Interface*). Arsitektur Mikroservis bergantung dengan HTTP dan TCP untuk mengkoneksikan atau menyambungkan setiap layanannya, maka web server merupakan core yang penting dalam melakukan proses tersebut (Unsong et al., 2021). Penelitian kali ini akan menggunakan Apache dan Nginx sebagai objek perbandingan performa.

PLN Marketplace adalah sebuah aplikasi *e-commerce* yang dimiliki oleh PT PLN. Aplikasi ini dibangun dengan arsitektur mikroservis, saat ini terus dilakukan pengembangan oleh tim software house PLN untuk membangun API terbaik, dimana API yang dibangun menggunakan bahasa pemograman PHP, tentu saja PHP ini membutuhkan web server agar service API ini dapat berjalan sebagaimana mestinya. Dalam menjalankan service API pemograman PHP membutuhkan sebuah Web Server. Service API itu sendiri merupakan halaman situs informasi yang dapat diakses dengan cepat melalui browser. Banyak jenis-jenis web server yang dapat digunakan, diantaranya Apache Web server, Apache Tomcat, Microsoft Internet Information Service (IIS) , Nginx, Lighttpd, Zeus Web server, dan lainnya. Menurut Busran et al., (2020) masing-masing dari *software* tersebut tentu mempunyai keunggulan maupun kelemahan, baik dari segi performa maupun dari segi fitur yang disediakan.

Mengacu dari penelitian terdahulu pengujian Apache dan Nginx menggunakan Apache Bench menghasilkan bahwa Nginx menunjukkan kinerja yang lebih baik dalam menangani *request* dari *client*, dengan rata-rata waktu penyelesaian request

yang lebih cepat dibandingkan Apache. Meskipun demikian, terdapat perbedaan hasil tergantung pada parameter yang diamati. Misalnya, Apache mungkin unggul dalam hal response time dan throughput, namun Nginx mungkin lebih baik dalam hal jumlah transaksi. Berdasarkan hasil uji performa dari penelitian terdahulu, disarankan untuk menggunakan Nginx jika sistem website akan menghadapi banyaknya *client request*, mengingat kinerjanya yang lebih cepat dalam menangani *request*. Namun, untuk keperluan tertentu, seperti *throughput* yang lebih tinggi, Apache mungkin menjadi pilihan yang lebih baik. Berdasarkan penelitian perbandingan performa antara Apache dan Nginx terdahulu, maka diangkatlah topik Analisa Perbandingan Kinerja Nginx dan Apache pada Lingkungan Virtual dan Docker Studi Kasus Aplikasi PLN Marketplace untuk mencari tahu perbedaan antara keduanya pada API aplikasi ini.

Analisa Perbandingan Kinerja Nginx dan Apache pada Lingkungan Virtual dan Docker Studi Kasus Aplikasi PLN Marketplace ini dilakukan untuk mencari tahu web server mana yang terbaik atau web server mana yang mempunyai performa terbaik untuk dijalankan di lingkungan *Virtual Machine* dan lingkungan Docker, agar nantinya dapat digunakan pada servis API aplikasi PLN Marketplace ini. Kedua web server tersebut yang bertindak sebagai web server akan diuji menggunakan aplikasi Apache Jmeter dan Apache Bench, dengan parameter QOS (*Quality of Service*) yaitu *Delay (Latency)*, *Jitter*, *Througput*, *Error (Packet Loss)*, *Network Troughput*, *Time Taken* dan sebagai perbandingan utilitas diukur juga penggunaan CPU dan RAM. Apache Jmeter akan dijalankan menggunakan t (waktu) selama 120s, sedangkan pengujian dengan Apache Bench dijalankan tanpa menggunakan variabel t (waktu). Serta penelitian ini menggunakan analisa statistika dan uji T *Student* agar dapat mengambil keputusan mana web server yang mempunyai performa terbaik untuk aplikasi PLN Marketplace itu sendiri.

1.1.1. Tujuan Penelitian

Berdasarkan latar belakang yang telah diuraikan maka tujuan dari penelitian ini adalah untuk menganalisa kinerja dan mencari tau performa terbaik dari Web Server Apache dan Nginx, dimana parameter yang diuji tersebut merupakan acuan yang relevan untuk mengetahui kinerja dari kedua web server.

1.1.2. Ruang Lingkup Penelitian

Ruang lingkup penelitian ini dibatasi pada :

- a. Penelitian dilakukan pada lingkungan Docker dan *Virtual Machine*, pada Docker menggunakan *docker compose* untuk mengatur *multi Docker*, sedang untuk lingkungan virtual akan menggunakan *virtual machine* dari VPS (*Virtual Private Server*).
- b. Web Server yang digunakan adalah Apache dan Nginx, dua objek ini adalah Web Server yang dikategorikan web *server* terbaik menurut data Netcraft.
- c. API yang diuji ada 1 API, yaitu GET API semua produk yang tersedia pada database aplikasi staging PLN Marketplace.
- d. Parameter yang diukur sebagai perbandingan adalah kualitas layanan atau yang sering disebut dengan QOS (*Quality of Service*) yaitu *Delay (Latency)*, *Jitter*, *Througput*, *Error (Packet Loss)*, *Network Throughput*, *Time Taken* dan sebagai ukuran utilitas diukur juga penggunaan CPU dan RAM.
- e. Sistem Operasi yang digunakan adalah Ubuntu 20.04 LTS.

1.1.3. Manfaat Penelitian

Manfaat penelitian ini adalah :

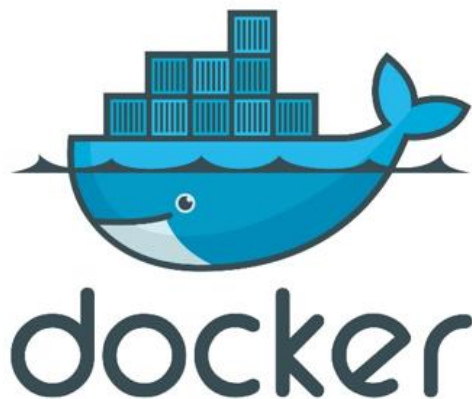
- a. Memudahkan pengelola dalam menentukan performa Web Server terbaik
- b. Mengimplementasikan penggunaan Web Server untuk kestabilan dalam penggunaan API aplikasi PLN Marketplace.
- c. Mengetahui perbedaan kinerja dari kedua Web Server dilingkungan Docker dan *Virtual Machine*.
- d. Menganalisa penggunaan QOS (*Quality of Service*) yaitu *Delay (Latency)*, *Jitter*, *Throughput*, *Error (Packet Loss)*, *Network Throughput*, *Time Taken*, CPU, dan RAM

BAB II TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1. Docker

Teknik Docker kontainer ini mengisolasi suatu sistem sehingga tidak mengganggu sistem yang lainnya atau bisa juga disebut suatu teknik virtualisasi berbasis linux dengan cara *containerization*, Virtualisasi berbasis linux dan *container* adalah teknik yang tidak menerapkan *Hypervisor*, dimana *Hypervisor* hanya mengisolasi proses tanpa mengisolasi perangkat keras, Kernel, dan operating system, sehingga dapat mengurangi *overhead* pada perangkat keras dan juga performanya lebih baik dari virtualisasi mesin. Salah satu virtualisasi berbasis linux dan kontainer adalah Docker (Dame et al., 2023). Kontainer adalah sebuah Operating System Virtual yang berlaku sebagai pembungkus yang membungkus aplikasi bersama dengan *dependency* dan *environment*-nya. Setiap kontainer ini memiliki proses yang tertutup dari kontainer lain maupun dengan *host OS*-nya. Prinsip kontainer ini diibaratkan kapal kargo, yang mana kapal kargo itu adalah sebuah aplikasi yang kompleks dan kontainernya adalah layanan - layanan kecil yang dimuat dan saling terkoneksi dengan menggunakan *Rest API* untuk menjalankan aplikasinya (Unsong et al., 2021).

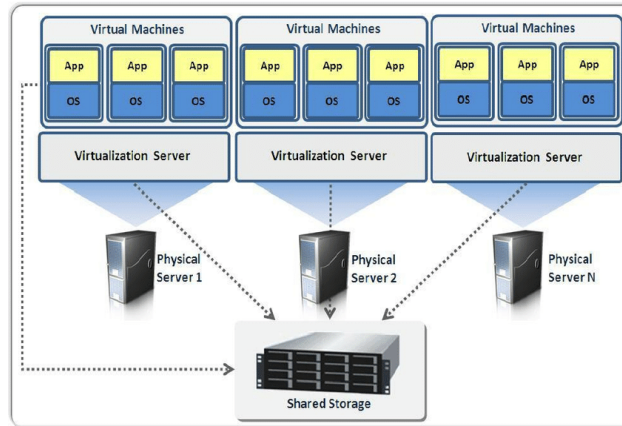


Gambar 1 Docker Kontainer

2.1.2. Virtual Machine

Menurut Dwiyatno et al., (2020) secara sederhana, virtualisasi adalah lawan kata dari physical machine atau mesin fisik. Physical machine adalah wujud server yang memiliki berbagai komponen seperti power supply, mainboard, memory, disk, dan sebagainya. Virtualisasi adalah aplikasi seolah-olah berjalan sendirian dalam satu mesin, tetapi sebenarnya virtualisasi berjalan di atas mesin lain, bersama-sama dengan aplikasi lain. Virtualisasi merupakan suatu teknik yang memungkinkan untuk mengatur dan membagi beberapa sistem operasi didalam satu server. Virtualisasi memiliki fungsi utama yaitu menghemat biaya pembelian server, karena dengan menggunakan konsep virtualisasi pengembang dapat mempunyai satu server dengan banyak operating system. Terdapat beberapa jenis virtualisasi salah satunya adalah virtualisasi kontainerisasi. Teknik ini mempermudah dan menghemat pembangunan serta

pengembangan sistem, jenis virtualisasi kontainer (ringan) membantu pada bidang DevOps sedangkan virtualisasi *hypervisor* (berat) lebih membantu menghemat dalam bidang pengembangan server karena tipe ini memvirtualisasi *machine virtual* secara sempurna.



Gambar 2 Virtual Machine

2.1.3. Web Server

Web Server merupakan *software* yang memberikan layanan berbasis data yang berfungsi menerima permintaan dari HTTP atau HTTPS pada *web browser* dan mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web atau berbentuk dokumen HTML (Jiwandono, 2021). Web Server menunggu permintaan dari client yang menggunakan *browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla*, dan program browser lainnya. Jika ada permintaan dari *browser*, maka Web Server akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser (Oki, 2022). Kinerja web server yang baik sangat mempengaruhi hubungan yang terjadi antara *client* dan server (Adnan, 2016).

2.1.4. Apache

Apache web *server* adalah *unix-based* Web Server, Apache awalnya dikembangkan berbasis kode pada NCSA HTTPD 1.3 yang kemudian diprogram ulang menjadi sebuah web *server* yang paling banyak digunakan saat ini. Apache kini menjadi Web Server yang paling populer dan banyak digunakan lebih dari 42% dari berbagai domain website yang ada di internet. Apache memiliki fitur yang sangat lengkap mulai dari performa yang tinggi, fungsionalitas, efisiensi, serta kecepatan. Apache juga merupakan Web Server berbasis *opensource* (Chandra, 2019).

2.1.5. Nginx

Nginx atau biasa disebut "*Engine-x*", adalah *opensource* Web Server. Nginx selain digunakan sebagai Web Server juga memiliki fitur untuk digunakan sebagai *reverse proxy*, *HTTP cache*, dan *load balancer* (Azi et al., 2023). Nginx dibuat oleh Igor Sysoev dan dirilis ke publik pada bulan Oktober 2004. Saat awal dirilis Igor meyakinkan publik bahwa Nginx dapat menjadi jawaban untuk mengatasi permasalahan yang ada pada saat itu yaitu permasalahan performa Web Server jika memiliki koneksi aktif lebih dari 10.000 koneksi secara bersamaan. Nginx menawarkan penggunaan memori yang lebih rendah dibandingkan Web Server

lainnya dan juga beberapa fitur seperti: *reverse proxy*, IPv6, *load balancing*, *FastCGI* support, web sockets, handling static files, TLS/SSL (Chandra, 2019).

2.1.7. Arsitektur

Apache adalah sebuah web server yang dibuat tahun 1995 lebih tua dibanding dengan Nginx, Apache ditujukan untuk menyediakan web server yang aman, stabil dan fleksibel. Sebagai contoh, Apache mengelola modul *multi-process-module* untuk mempertahankan koneksi yang sama namun dengan cara yang berbeda. Apache memiliki arsitektur Prefork MPM yaitu menggunakan proses terpisah untuk setiap permintaan, sehingga setiap permintaan ditangani oleh satu proses. Setiap proses bekerja secara independen. Hal tersebut sangat menguntungkan karena administrator dapat mengaplikasikan kriteria untuk tiap koneksi, tetapi server ini membutuhkan lebih banyak *resources* karena kerjanya cukup berat. Sedangkan teknologi Nginx merupakan pengembangan dari Apache sehingga sekarang ini berkembang secara pesat sejak dipublikasi pada tahun 2004, dan sudah mulai banyak digunakan. Nginx dikenal dengan arsitektur *asynchronous* dan *events-driven architecture*, yang mana *single-threaded asynchronous architecture* yang didesain untuk menangani banyak permintaan secara efisien dalam satu proses atau beberapa *worker processes*. Nginx lebih dikenal karena server tidak terlalu membutuhkan banyak *resources* sehingga Nginx lebih ringan saat digunakan dibandingkan dengan Apache (Perbedaan Nginx vs Apache: Pilih Server Web Mana yang Lebih Baik?, 2024.). Menurut Unsong et al., (2021a) Nginx juga lebih efisien saat menampilkan konten statis.

2.1.6. Quality of Service

Quality of Service adalah teknik untuk mengelola Throughput, Delay, Jitter dan Packet Loss untuk aliran dalam jaringan. Tujuan dari mekanisme QoS adalah mempengaruhi setidaknya satu diantara empat parameter dasar QoS yang telah ditentukan. QoS didesain untuk membantu *end user* menjadi lebih produktif dengan memastikan bahwa *user* mendapatkan performansi yang handal dari aplikasi berbasis jaringan. QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda. QoS merupakan suatu tantangan yang besar dalam jaringan berbasis IP dan internet secara keseluruhan (Aprianto Budiman et al., 2020a).

2.1.7. Throughput

Throughput merupakan *bandwidth* yang sesungguhnya. Jika *bandwidth* adalah batas maksimal maka *throughput* adalah data yang sesungguhnya, contohnya jika memakai Internet dengan menggunakan *bandwidth* 10 Mbps dan kecepatan yang tertera adalah 5 Mbps pada saat download file kecepatan yang tertera 5 Mbps ini yang disebut dengan *throughput*. *Throughput* bisa diukur dengan kondisi jaringan dan waktu tertentu yang dipakai untuk transfer file dengan ukuran tertentu. Contoh jika *bandwidth*-nya 32 kbps lalu ketika mengunduh file 265 kbps dari Internet, file seharusnya terunduh di komputer hanya dalam waktu 8 detik saja dari perhitungan $265/32$, lalu jika file yang terunduh dalam waktu 10 detik maka *bandwidth* yang realnya adalah $265 \text{ kb} / 10 \text{ detik}$ yaitu 25,6 kbps (Unsong et al., 2021).

Tabel 1 Kategori *Throughput*

Kategori <i>Throughput</i>	<i>Index</i>	<i>Throughput</i>
Sangat bagus	76 – 100%	4
Bagus	51 – 75%	3
Sedang	26 – 50%	2
Buruk	< 25%	1

Persamaan perhitungan *throughput* :

$$\textit{Throughput} = \text{Paket data diterima} / \text{Lama Pengamatan}$$

2.1.8. Delay

Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya (Aprianto Budiman et al., 2020a). *Delay* di dalam jaringan dapat digolongkan sebagai berikut:

- a. *Packetization Delay*
Delay yang disebabkan oleh waktu yang diperlukan untuk proses pembentukan paket IP dari informasi *user*. *Delay* ini hanya terjadi sekali saja, yaitu di sumber informasi.
- b. *Queuing Delay*
Delay ini disebabkan oleh waktu proses yang diperlukan oleh router dalam menangani transmisi paket di jaringan. Umumnya *delay* ini sangat kecil, kurang lebih sekitar 100 *micro second*.
- c. *Delay Propagation*
Delay ini adalah proses perjalanan informasi selama di dalam media transmisi, misalnya kabel SDH, *coaxial* atau tembaga, menyebabkan *delay* yang disebut dengan *delay* propagasi.

Tabel 2 Kategori *Delay*

Kategori <i>Delay</i>	<i>Index</i>	<i>Throughput</i>
Sangat bagus	< 150 m/s	4
Bagus	150 s/d 300 m/s	3
Sedang	300 s/d 450 m/s	2
Buruk	> 450 m/s	1

2.1.9. Packet Loss

Packet Loss didefinisikan sebagai kegagalan transmisi paket IP mencapai tujuannya. Kegagalan paket tersebut mencapai tujuan, dapat disebabkan oleh beberapa kemungkinan (Aprianto Budiman et al., 2020b), diantaranya yaitu:

- a. Terjadinya *overload* trafik didalam jaringan.
- b. Tabrakan (*congestion*) dalam jaringan.
- c. *Error* yang terjadi pada media fisik.
- d. Kegagalan yang terjadi pada sisi penerima antara lain bisa disebabkan karena *overflow* yang terjadi pada *buffer*.

Tabel 3 Kategori *Packet Loss*

Kategori Packet Loss	Index	Througput
Sangat bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Buruk	25%	1

2.1.10. Jitter

Jitter merupakan variasi *delay* antar paket yang terjadi pada jaringan IP. Besarnya nilai *jitter* akan sangat dipengaruhi oleh *variasi* beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan IP. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion* dengan demikian nilai *jitter* akan semakin besar. Semakin besar nilai *jitter* akan mengakibatkan nilai QoS akan semakin turun. Untuk mendapatkan nilai QoS jaringan yang baik, nilai *jitter* harus dijaga seminimum mungkin (Aprianto Budiman et al., 2020a).

Tabel 4 Kategori *Jitter*

Kategori Jitter	Index	Througput
Sangat bagus	0 m/s	4
Bagus	1 s/d 75 m/s	3
Sedang	76 s/d 125 m/s	2
Buruk	>225 m/s	1

2.1.11. CPU (Central Processing Unit)

CPU merupakan bagian komputer yang berfungsi untuk menangani semua instruksi yang diterimanya baik dari *hardware* maupun dari *software* selama komputer digunakan. CPU berbentuk *chip* persegi di mana bagian bawahnya ada ratusan pin konektor. Komponen primer dari CPU adalah *arithmetic logic* untuk yang menampilkan matematika, logika dan hasil operasi secara langsung (Output, 2021a).

2.1.12. RAM (Random Access Memory)

RAM merupakan salah satu perangkat keras komputer yang penting dalam membantu meningkatkan kinerja dari sebuah komputer. Pengertian *Random Acces Memory* atau RAM adalah sebuah perangkat keras komputer yang mempunyai fungsi untuk penyimpanan data dan sebuah instruksi-intruksi program yang sedang kita jalankan dan sebelumnya sudah di eksekusi oleh sebuah *processor* (Output, 2021b).

2.1.13. Network Troughput

Menurut (Ammar et al., 2016) Network Troughput secara singkat adalah parameter untuk menyatakan nilai download/upload data (KB/s) pada suatu lalu lintas jaringan atau disebut juga dengan Network Througput. Network Througput ini sebenarnya adalah ukuran lajur data yang berjalan saat proses download/upload (KB) terjadi. Semakin besar Network Troughput maka semakin bagus kualitas kecepatan transfer data. Network Througput diukur dalam *byte per second* (bps), *megabyte per second* (Mbps), atau *gigabyte per second* (Gbps), menggambarkan jumlah data yang berhasil dipindahkan dari satu tempat ke tempat lain dalam periode waktu tertentu. Ini

mencakup pengukuran efektivitas jaringan dalam menangani *traffic* data (Apa itu Throughput? Arti, Fungsi, Contoh, FAQs 2024 | RevoU, 2024).

Persamaan perhitungan *throughput* :

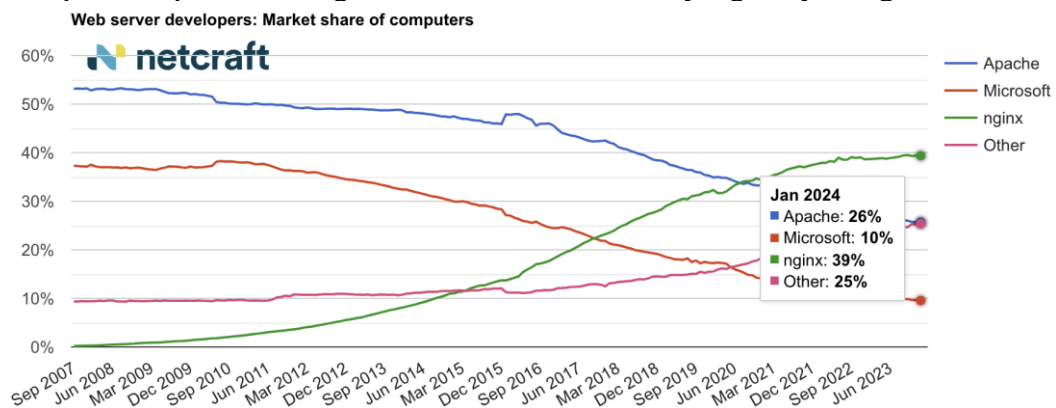
$$\text{Network Throughput} = \text{Data Length (bytes)} / \text{Time Taken} \times 1024$$

2.1.14. Time Taken

Time Taken adalah parameter atau nilai yang ditunjukkan oleh suatu proses dalam transfer data yang menandakan lamanya seluruh data diproses.

2.1.15. Netcraft

Netcraft adalah perusahaan jasa internet yang berbasis di Bath, Inggris. *Netcraft* Menyediakan data penelitian dan analisis terhadap berbagai aspek internet. *Netcraft* telah menjelajahi internet sejak tahun 1994 dan memiliki keenangan pada market share dari Web Server, sistem operasi, *hosting providers*, ISP, transaksi terenkripsi, bisnis secara elektronik, bahasa-bahasa *scripting* dan teknologi konten di Internet Survei yang dilakukan oleh netcraft, menampilkan web server Apache dan Nginx termasuk web server yang banyak digunakan. Tercatat sebanyak 26.09% website dunia menggunakan Web Server Apache dan Nginx penggunaanya sebanyak 39.00% (February 2024 Web Server Survey | Netcraft, 2024). Maka dari itu pemilihan objek Apache dan Nginx ini juga mengacu kepada survei dari netcraft yang menampilkan Apache dan Nginx termasuk Web Server yang banyak digunakan.



Gambar 3 Netcraft Survey

2.1.16. Apache Bench

Apache Benchmarking Tool adalah sebuah alat yang dibuat oleh Apache Organization yang dapat bekerja untuk mengukur performansi pada *Hypertext Transfer Protocol* (HTTP) web Server (Oki, 2022).

2.1.17. Apache Jmeter

Aplikasi Jmeter adalah perangkat lunak sumber terbuka, aplikasi Java 100% murni yang dirancang untuk memuat perilaku fungsional pengujian dan mengukur kinerja. Awalnya dirancang untuk menguji Aplikasi Web tetapi sejak itu diperluas ke fungsi pengujian lainnya (Apache JMeter - Apache JMeterTM.)

2.1.18. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side*, maka *script* dari PHP nantinya akan diproses didalam sebuah web server. Jenis web server yang sering digunakan bersama dengan PHP antara lain Apache, Nginx, IIS (*Internet Information Server*) dan LiteSpeed (Agustine et al., 2023)

2.2. Penelitian Terdahulu

Berikut penelitian sebelumnya tentang analisa kinerja Apache dan Nginx telah banyak diperkenalkan diantaranya:

- 1 Penulis : Andri Jiwandono
Judul : Analisa Perbandingan Kinerja Web Server Apache, Nginx, dan Litespeed dengan menggunakan Metode Stress Test
Tahun : 2021
Isi : Pada penelitian ini dilakukan dengan melakukan perbandingan 3 objek yaitu, Apache, Nginx dan litespeed. Penelitian ini dilakukan pada lingkunagn virtual dan hasil perbandinganya dilakukan menggunakan stress test dengan tools Jmeter. Hasil dari penelitian ini beragam, ada yang diliat dari sisi *throughput*, keamanan, kecepatan sent sehingga mendapati hasil yang bervariasi.
- 2 Penulis : Howard Christopher Yoel Unsong dan Justinus Andjarwirawan
Judul : Analisa Kinerja Apache dan Nginx dalam Arsitektur *Microservice* Menggunakan Siege
Tahun : 2021
Isi : Pada penelitian ini dilakukan dalam lingkungan Docker, kemudian percobaan untuk Web Server ini melakukan 9 kombinasi percobaan dengan jumlah concurrent *user* yang terdiri dari 250000, 500000 dan 750000, sedangkan untuk waktu akan terdiri dari 2 menit, 4 menit dan 6 menit. berdasarkan hasil perhitungan rata-rata untuk response time dan *throughput* maka Web Server Apache lebih unggul dibandingkan dengan Web Server Nginx, *tetapi* berdasarkan untuk jumlah dari transaksi Web Server Nginx lebih banyak daripada Web Server Apache.
- 3 Penulis : Albert Yakobus Chandra
Judul : Analisis Performansi Antara Apache & Nginx Web Server dalam Menangani *Client Request*
Tahun : 2019
Isi : Pada penelitian dilakukan dalam lingkungan Docker, Pengujian yang dilakukan menggunakan tools Apache Bench untuk melakukan benchmarking dari sisi banyak nya *client request* yang bervariasi mulai dari 100 request sampai dengan 1000000 request dan waktu yang dibutuhkan untuk

- menyelesaikan request tersebut. Hasil uji pada penelitian ini memberikan hasil benchmarking yang menunjukkan bahwa dari sisi penggunaan waktu, Nginx menggunakan waktu lebih sedikit daripada Apache dalam menyelesaikan *client* request.
- 4 Penulis : Lady Agustin Fitriana dan Syarah Seimahuira
- Judul : Penerapan Metode SAW dalam Analisa Perbandingan Performa Web Server (Apache, Nginx, Lighttpd, Iis) pada Bahasa Pemrograman PHP
- Tahun : 2023
- Isi : Penelitian ini dilakukan pada lingkungan virtual dengan melakukan deployment bahasa pemrograman PHP. Berdasarkan penelitian yang dilakukan mengenai proses perbandingan performa Web Server pada bahasa pemrograman PHP, maka ditarik kesimpulan bahwa terdapat perbedaan hasil antara pengolahan data menggunakan metode *Simple Additive Weighting* (SAW) dengan uji coba menggunakan Apache Bench. Pengujian menggunakan Apache Bench menghasilkan webserver Nginx sebagai webserver paling optimal, sedangkan pengujian dengan SAW menghasilkan IIS sebagai server paling optimal. Sistem pendukung keputusan mampu membantu pengambil keputusan dalam memproses data atau informasi dan memecahkan masalah yang sangat kompleks dan tidak terstruktur. Mampu menghasilkan solusi alternatif yang lebih cepat dan hasil yang dapat diandalkan.
- 6 Penulis : Ardian Dwi Praba dan Hariyanto
- Judul : Performansi Web Server Apache dan Nginx Pada Aplikasi penjualan Online
- Tahun : 2020
- Isi : Penelitian dilakukan dilingkungan virtual. Pengujian pada penelitian ini menggunakan tools Apache Bench untuk melakukan benchmarking dari sisi banyaknya permintaan dari pengguna mulai dari 500 sampai dengan 15000 permintaan dan waktu yang dibutuhkan oleh sever untuk menyelesaikan permintaan tersebut. Hasil pada pengujian banchmarking ini menunjukkan bahwa dari sisi kecepatan waktu, Nginx menggunakan waktu lebih sedikit dibandingkan dengan Apache dalam menyelesaikan permintaan dari pengguna.

2.3. Tabel Perbandingan

Tabel perbandingan penelitian menunjukkan beberapa perbedaan yaitu :

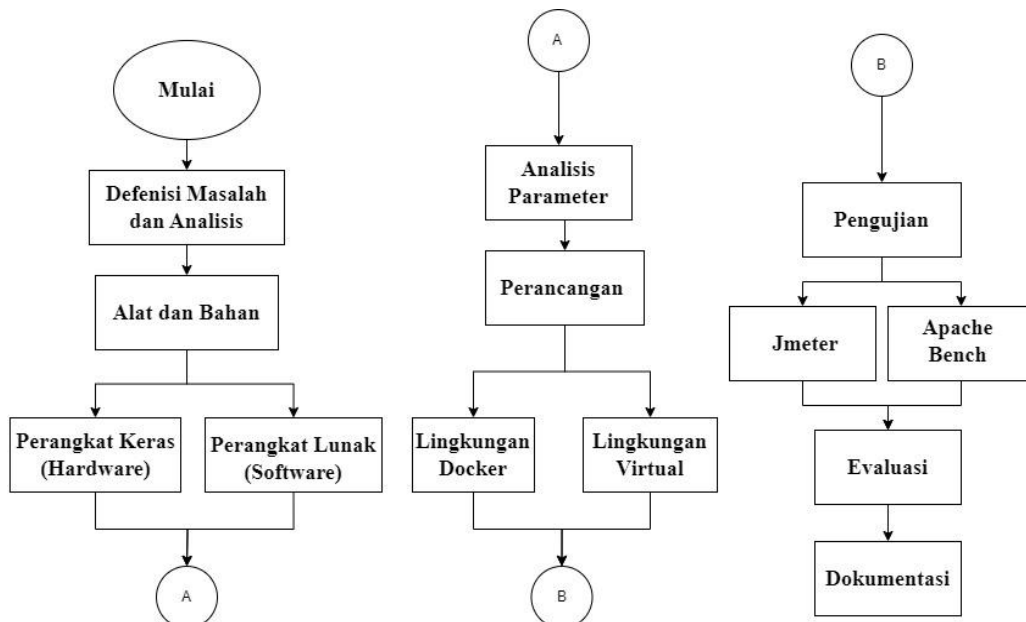
Tabel 5 Perbandingan Penelitian

No	Peneliti dan Tahun	Lingkungan		Alat Uji			Web server			
		Docker	Virtual	Apache Jmeter	Apache Bench	Siege	Apache	Nginx	Litespeed	IIS
1	Andri Jiwandono (2021)		✓	✓			✓	✓	✓	
2	Howard Christopher Yoel Unsong, dan Justinus Andjarwirawan (2021)	✓				✓	✓			
3	Albert Yakobus Chandra (2019)	✓			✓					
4	Lady Agustin Fitriana dan Syarah Seimahaira (2023)				✓			✓		✓
5	Ardian Dwi Praba dan Hariyanto (2020)		✓		✓			✓		
6	Falyan Zuril (2024)	✓	✓	✓	✓		✓	✓		

BAB III METODOLOGI PENELITIAN

3.1. Metode Penelitian

Metode untuk menganalisa perbandingan yang digunakan dalam penelitian “Analisa Perbandingan Kinerja Nginx dan Apache pada Lingkungan Virtual dan Docker Studi Kasus Aplikasi PLN Marketplace” ini menggunakan metode NDLC atau yang lebih dikenal dengan *Network Development Life Cycle*. Metode *Network Development Life Cycle* adalah teknik analisis terstruktur yang digunakan untuk merencanakan dan mengolah proses pengembangan sebuah sistem. NDLC ialah model yang mendeskripsikan proses siklus pengembangan dan perancangan suatu jaringan komputer. Sebagaimana model pengembangan sistem jaringan komputer ini lebih teruntuk sistem *software*, NDLC memiliki elemen yang mendefinisikan tahapan, fase, langkah, atau prosedur proses yang spesifik. Kata *cycle* pada NDLC merupakan pusat deskriptif dari siklus pembangunan sistem jaringan yang digambarkan secara menyeluruh, agar proses dan tahapan pembangunan sistem jaringan dalam berjalan secara kesinambungan (Danil Fajri et al., 2023.), berikutnya ditambahkan juga sebuah metode analisis statistik deskriptif untuk mengambil nilai rata – rata dari setiap pengujian yang dilakukan, analisa statistik deskriptif digunakan untuk menjelaskan atau memberikan gambaran mengenai karakteristik dari serangkaian data tanpa mengambil kesimpulan umum (Leni Masnidar Nasution, 2017). Setelah melakukan analisa statistik akan dilanjutkan dengan uji T student yaitu mengambil hipotesis awal atau H0 dan setelah analisa statistik didapatkan maka akan diambil hipotesis alternatif atau H1 (hipotesa kerja). Penelitian ini akan dilakukan tahapan seperti yang ditunjukkan pada gambar 4.



Gambar 4 Metode Penelitian

3.1.1. Defenisi Masalah dan Analisis

Penelitian ini memiliki beberapa masalah mengenai layanan Web Server yaitu:

- a. Kurangnya informasi Web Server yang tahan dengan beban permintaan yang besar meliputi banyaknya jumlah *users* yang mengakses web *server* tersebut.
- b. Adanya kekurangan kinerja pada setiap Web Server sehingga dilakukan perbandingan kinerja Web Server Apache dan Nginx pada lingkungan Docker dan lingkungan virtual.

Dari beberapa permasalahan diatas akan dilakukan pengujian Web Server Apache dan Nginx dengan berpatokan atau mengambil parameter QOS (Quality of Service) yaitu *Jitter*, *Delay*, *Througput*, Network Troughput, Time Taken dan *Packet Loss*, serta akan diukur juga bagaimana efisiensi dari penggunaan CPU dan RAM dari Apache dan Nginx pada lingkungan Docker dan lingkungan virtual.

3.1.2. Alat dan Bahan

Alat dan bahan yang digunakan pada penelitian terbagi kedalam dua kategori yaitu alat untuk perangkat keras (*Hardware*) dan alat untuk perangkat lunak (*Software*) yaitu:

Tabel 6 Alat dan Bahan

No	Nama Alat	<i>Hardware</i>	<i>Software</i>	Keterangan
1	Prosesor Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz	✓		Prosesor server virtual
2	RAM (Random Acces Memory) DDR3	✓		RAM server Virtual
3	Laptop Acer Swift 3 SF315 Ryzen 5	✓		Laptop untuk <i>remote server</i>
4	Ubuntu 20.04 LTS		✓	Sistem Operasi mesin virtual
5	Apache		✓	Web Server untuk menjalankan pemograman PHP
6	Nginx		✓	Web Server untuk menjalankan pemograman PHP
7	Apache Bench		✓	Alat untuk pengujian Web Server
8	Jmeter		✓	Alat untuk pengujian Web Server
9	Windows 10		✓	Sistem operasi pada leptop untuk <i>remote server</i>
10	Docker		✓	Alat untuk kontainerisasi
11	Docker <i>Compose</i>		✓	Alat untuk mendefinisikan dan menjalankan aplikasi Docker multi-kontainer
12	Postgresql		✓	Alat untuk mengatur tatakelola database

13	Python3		✓	Alat bantu untuk menghitung uji T Student
----	---------	--	---	---

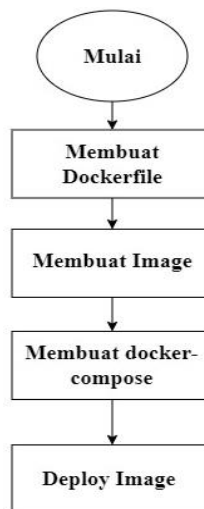
3.1.3. Analisis Parameter

Tahapan ini akan dirancang sebuah proses pengiriman data dari *client* menuju server, sehingga dapat diketahui parameter yang diukur yaitu, *Jitter*, *Delay*, *Througput*, *Network Troughput*, *Time Taken* dan *Packet Loss*, serta akan diukur juga bagaimana efisiensi dari penggunaan CPU dan RAM dari Apache dan Nginx. Kesimpulan akan diambil dari hasil perbandingan QOS ini.

3.1.4. Perancangan

Tahapan ini membuat desain topologi yang akan digunakan untuk menguji perbandingan kinerja Web Server Apache dan Nginx. serta juga mendesign arsitektual infrastuktur lalu menentukan spesifikasi alat pada lingkungan Docker dan spesifikasi pada lingkungan Virtual. Tahapan ini dibagi menjadi dua bagian yaitu pada lingkungan Docker dan lingkungan Virtual.

a. Lingkungan Docker



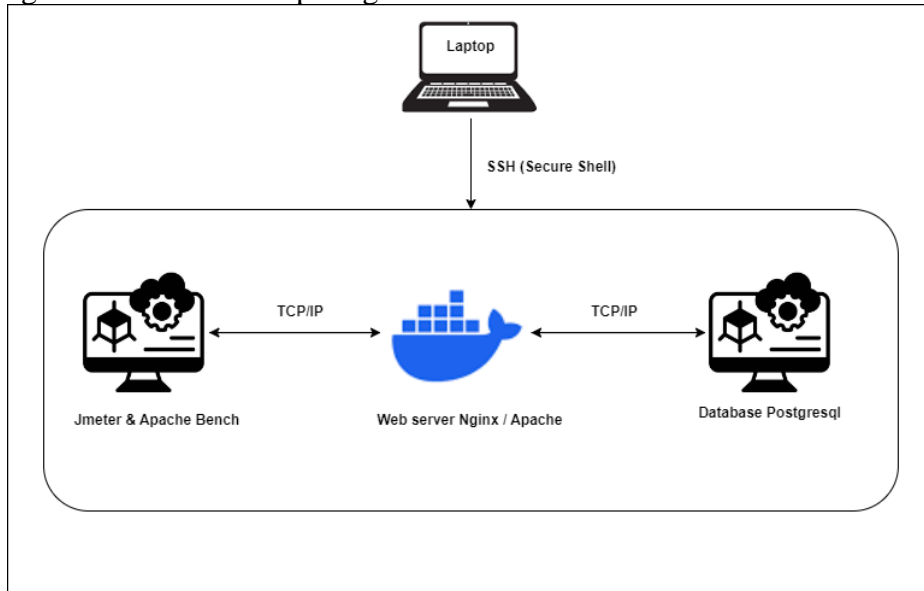
Gambar 5 Lingkungan Docker

Tahapan membuat Dockerfile adalah sebuah tahapan untuk mendefinisikan kebutuhan objek aplikasi untuk berjalan. Dockerfile adalah dokumen teks yang berisi sebuah perintah untuk membangun sebuah image. Dengan menggunakan perintah Docker build pengguna dapat mengeksekusi beberapa instruksi baris perintah secara berurutan. Dockerfile ini digunakan untuk membuat sebuah lingkungan yang terisolasi serta mendefinisikan kebutuhan kebutuhan untuk sebuah objek yang akan berjalan menjadi sebuah kontainer.

Tahapan membuat image adalah proses membangun komponen yang akan diisolasi menjadi kontainer. Image ini dibangun dari Dockerfile yang telah didefinisikan sesuai kebutuhan objek aplikasi yang akan dijalankan.

Tahapan membuat Docker compose adalah bagian untuk mendefinisikan kontainer yang akan berjalan. Mendefinisikan yang dimaksud adalah mendefinisikan nama layanan, kebutuhan RAM, CPU dan lain – lain.

Tahapan deploy image adalah bagian untuk merubah image menjadi kontianer yang akan berjalan sebagai sebuah layanan aplikasi. Adapun model arsitektur untuk lingkungan Docker adalah seperti gambar 6.



Gambar 6 Arsitektur lingkungan docker

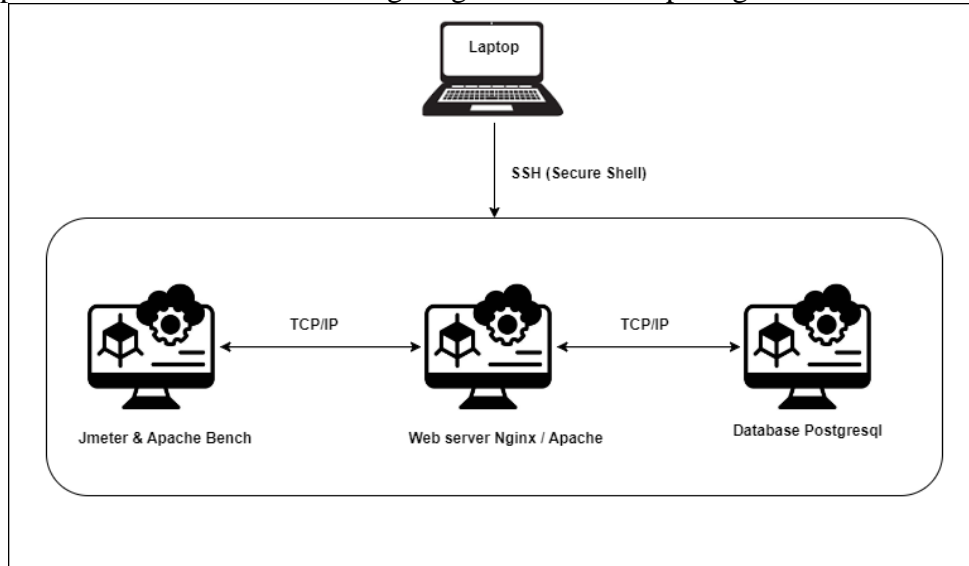
b. Lingkungan Virtual



Gambar 7 Lingkungan Virtual

Tahapan perancangan instal Web Server adalah menginstal Web Server pada server yang akan dijadi tempat aplikasi berjalan. Tahapan persiapan requirement PHP adalah tahapan untuk menentukan kebutuhan agar aplikasi bisa berjalan dengan

baik pada Web Server yang berjalan. Tahapan konfigurasi adalah tahapan mengkonfigurasi Web Server agar berjalan sesuai kebutuhan aplikasi. Tahapan deploy Web Server adalah tahapan menjalankan Web Server yang telah dikonfigurasi. Adapun model arsitektur untuk lingkungan virtual ini seperti gambar 8.



Gambar 8 Arsitektur lingkungan virtual

3.1.6. Pengujian

Tahapan ini melakukan pengujian dengan dua alat yaitu Apache Jmeter dan Apache Bench. Hasil pengujian akan mengumpulkan data berupa pengukuran QOS, lalu dilakukan perbandingan melalui proses pengiriman data dari *client* menuju server. Pengujian akan dilakukan dengan skenario 500 *users*, 1000 *users*, dan 1500 *users*, serta setiap pengujian antara Web Server Apache dan Nginx akan dilakukan 10 kali pengulangan agar mendapatkan data untuk dianalisa secara statistik deskriptif.

3.1.7. Evaluasi

Setelah data didapatkan dari Apache Jmeter dan Apache Bench, selanjutnya data dievaluasi dengan menghitung rata – rata QOS yang dihasilkan, hal ini agar menemukan keunggulan dan kekurangan dari masing-masing Web Server. Keunggulan akan diketahui setelah perbandingan dari uji hipotesa awal (H0) dan uji hipotesa alternatif (H1) dibandingkan.

3.1.8. Dokumentasi

Tahap akhir adalah melakukan dokumentasi agar hasil dan evaluasi komparasi Web Server Apache dan Nginx dapat disimpan dengan baik dan diakses lagi apabila diperlukan kemudian hari.

3.2. Waktu dan Jadwal Penelitian

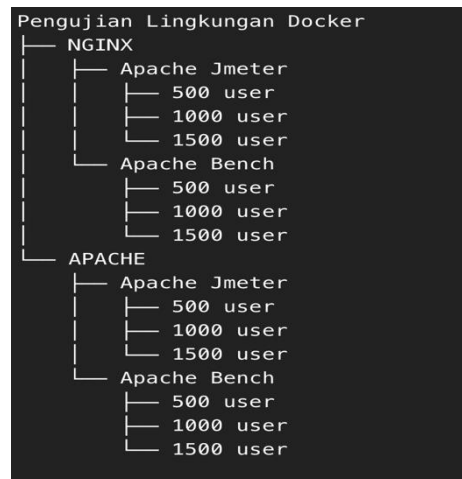
3.2.1. Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan mulai dari bulan Januari 2024 sampai bulan April 2024 (4 bulan). Waktu pelaksanaan dilaksanakan di lab komputer prodi ilmu komputer unpak.

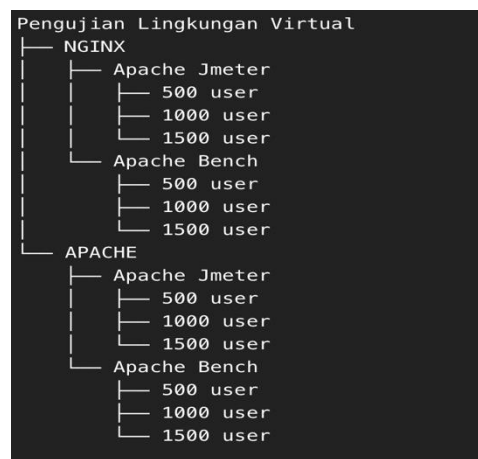
BAB IV IMPLEMENTASI

4.1. Pengujian

Pengujian dalam penelitian ini terdiri dari beberapa bagian yaitu pengujian untuk pengumpulan data, analisis statistik data, dan visualisasi melalui grafik. Data yang digunakan dalam penelitian ini adalah data produk dari aplikasi PLN *Marketplace* dilingkup *staging*. Skenario pengujian dilakukan dengan mengakses API semua produk yang berjumlah 450 data yang terdapat pada database aplikasi *staging* PLN *Marketplace*. API ini akan diakses dengan 500 user, 1000 user, 1500 user dan masing masing dijalankan dengan alat uji Apache Jmeter dan Apache Bench sebanyak 10 kali percobaan. Berikut gambar 9 dan gambar 10 adalah diagram *tree* dari skenario yang akan dilaksanakan untuk pengambilan data pengujian.



Gambar 9 Diagram *Tree* Pengujian Lingkungan Docker



Gambar 10 Diagram *Tree* Pengujian Lingkungan Virtual

4.1.1. Pengumpulan Data

Proses pengumpulan data dilakukan pengujian dengan Apache Jmeter yang akan dijalankan selama 120 s untuk setiap jumlah transaksi yang dijalankan sedangkan pengujian dengan Apache Bench dilakukan hanya dengan menjalankan transaksi yang ada dengan jumlah user tertentu.

- a. Data pengujian Nginx dengan Apache Jmeter dilingkungan Docker

Berikut adalah tabel data pengujian web server Nginx menggunakan Apache Jmeter pada lingkungan docker dengan jumlah 500 user, 1000 user, dan 1500 user.

Tabel 7 Data Pengujian Nginx (Docker – Apache Jmeter) – 500 user

PERCOBAAN KE	JUMLAH USER	NGINX						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Throughput (KB/s)
1	500	4,18	89,81	128	0	3,71	5,08	146,80
2	500	4,18	90,08	241	0	3,88	5,1	146,85
3	500	4,18	90,7	121	0	3,89	4,48	146,80
4	500	4,18	92,62	87	0	3,68	5,14	146,80
5	500	4,18	92,39	128	0	4,01	5,41	146,80
6	500	4,18	91,01	120	0	3,86	5,27	146,83
7	500	4,18	92,2	95	0	3,42	5,17	146,90
8	500	4,18	89,3	155	0	4,06	5,14	146,86
9	500	4,18	89,8	122	0	3,77	5,21	146,86
10	500	4,18	91,53	102	0	3,93	5,23	146,79

Tabel 8 Data Pengujian Nginx (Docker – Apache Jmeter) – 1000 user

PERCOBAAN KE	JUMLAH USER	NGINX						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Throughput (KB/s)
1	1000	8,36	92,32	162	0	4,46	8,96	293,49
2	1000	8,35	92,36	113	0	4,48	10,59	293,37
3	1000	8,35	91,06	152	0	3,88	10,14	293,43
4	1000	8,35	92,32	105	0	4,54	10,27	293,35
5	1000	8,35	92,92	124	0	4,42	10,21	293,33
6	1000	8,36	93,06	110	0	4,45	9,77	293,59
7	1000	8,35	92,07	105	0	4,48	10,26	293,41
8	1000	8,35	93,1	102	0	4,01	10,4	293,33
9	1000	8,36	94,32	165	0	4,46	10,42	293,53
10	1000	8,35	89,1	149	0	4,2	10,1	293,42

Tabel 9 Data Pengujian Nginx (Docker – Apache Jmeter) – 1500 user

PERCOBAAN KE	JUMLAH USER	NGINX						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Throughput (KB/s)
1	1500	12,53	96,39	204	0	4,91	16,14	439,69
2	1500	12,53	92,72	236	0	4,76	16,61	439,74
3	1500	12,53	93,69	111	0	5,12	15,87	439,79
4	1500	12,54	96,09	177	0	4,68	15,57	440,13
5	1500	12,52	93	117	0	4,24	16,22	439,64
6	1500	12,53	92,54	145	0	4,9	13,36	439,83
7	1500	12,53	93,97	142	0	5,11	15,16	439,98
8	1500	12,53	93,62	154	0	5,15	15,92	439,94
9	1500	12,53	94,18	133	0	5,17	15,87	439,82
10	1500	12,53	94,06	121	0	4,9	15,37	439,79

b. Data pengujian Nginx dengan Apache Bench dilingkungan Docker

Berikut adalah tabel data pengujian web server Nginx menggunakan Apache Bench pada lingkungan docker dengan jumlah 500 user, 1000 user, dan 1500 user.

Tabel 10 Data Pengujian Nginx (Docker – Apache Bench) – 500 user

PERCOBAAN KE	JUMLAH USER	NGINX					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Throughput (Kb/s)	RAM (%)	CPU (%)
1	500	11,74	111	42,581	412,07	5,04	13,51
2	500	11,42	250	43,792	400,68	5,04	13,51
3	500	11,83	63	42,272	415,09	5,22	12,36
4	500	11,84	80	42,222	415,58	5,38	11,54
5	500	11,8	60	42,376	414,06	5,07	13,32
6	500	11,68	105	42,817	409,89	5,04	11,41
7	500	11,67	77	42,852	409,47	4,88	10,4
8	500	11,67	70	42,845	409,54	5,28	11,57
9	500	11,36	135	44,01	398,78	5,06	12,83
10	500	11,8	58	42,387	413,96	5,55	11,36

Tabel 11 Data Pengujian Nginx (Docker – Apache Bench) – 1000 user

PERCOBAAN KE	JUMLAH USER	NGINX					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM (%)	CPU (%)
1	1000	11,5	150	86,975	403,48	5,14	13,7
2	1000	11,82	137	84,588	414,87	5,33	13,68
3	1000	11,65	83	85,808	408,97	5,3	13,58
4	1000	11,35	97	88,087	398,39	5,22	13,37
5	1000	11,3	91	88,461	396,7	5,4	13,79
6	1000	11,73	87	85,287	411,47	5,1	13,86
7	1000	11,76	97	85,037	412,68	5,27	14,09
8	1000	11,61	97	86,138	407,4	5,46	13,88
9	1000	11,2	91	89,297	392,99	5,16	13,83
10	1000	11,62	68	86,083	407,66	5,18	13,68

Tabel 12 Data Pengujian Nginx (Docker – Apache Bench) – 1500 user

PERCOBAAN KE	JUMLAH USER	NGINX					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM(%)	CPU(%)
1	1500	12,15	126	123,417	426,51	5,33	14,72
2	1500	12,03	84	124,658	422,27	5,37	13,6
3	1500	12,13	82	123,621	425,81	5,89	13,59
4	1500	11,56	116	129,724	405,78	5,55	13,61
5	1500	11,59	93	129,446	406,65	5,95	13,53
6	1500	11,83	92	126,763	415,25	5,34	13,99
7	1500	12,03	114	124,671	422,22	5,87	14,41
8	1500	11,29	112	132,866	396,18	5,43	13,76
9	1500	11,79	124	127,178	413,9	5,45	14,18
10	1500	11,39	264	131,724	399,62	5,5	14,03

c. Data pengujian Nginx dengan Apache Jmeter dilingkungan Virtual

Berikut adalah tabel data pengujian web server Nginx menggunakan Apache Jmeter pada lingkungan virtual dengan jumlah 500 user, 1000 user, dan 1500 user.

Tabel 13 Data Pengujian Nginx (Virtual – Apache Jmeter) – 500 user

PERCOBAAN KE	JUMLAH USER	NGINX						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Troughput (KB/s)
1	500	4,18	70,77	70	0	27	4,07	189,75
2	500	4,18	71,46	93	0	27	4,07	189,83
3	500	4,18	69,84	90	0	27	3,97	189,91
4	500	4,18	68,61	74	0	27	3,93	189,77
5	500	4,19	69,61	108	0	27	4,03	190
6	500	4,18	71,32	238	0	27	4,05	189,79
7	500	4,18	69,54	98	0	27	4,35	189,72
8	500	4,18	66,97	102	0	27	4,15	189,71
9	500	4,18	69,33	132	0	27	4,12	189,73
10	500	4,18	67,35	94	0	27	3,75	189,92

Tabel 14 Data Pengujian Nginx (Virtual – Apache Jmeter) – 1000 user

PERCOBAAN KE	JUMLAH USER	NGINX						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Troughput (KB/s)
1	1000	8,36	67,3	161	0	26	7,3	379,59
2	1000	8,36	67,92	85	0	27	7,78	379,48
3	1000	8,35	68,95	92	0	27	7,63	379,24
4	1000	8,36	69,04	117	0	27	7,37	379,52
5	1000	8,36	68,46	83	0	27	7,35	379,38
6	1000	8,35	68,92	69	0	27	7,3	379,19
7	1000	8,36	69,22	88	0	28	7,88	379,35
8	1000	8,36	68,82	86	0	27	7,8	379,36
9	1000	8,35	69,72	91	0	27	7,82	379,26
10	1000	8,36	69,26	89	0	26	7,2	379,40

Tabel 15 Data Pengujian Nginx (Virtual – Apache Jmeter) – 1500 user

PERCOBAAN KE	JUMLAH USER	NGINX						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Troughput (KB/s)
1	1500	12,53	68,90	130	0	27	11,4	568,66
2	1500	12,53	68,37	158	0	26	10,09	568,55
3	1500	12,53	68,61	173	0	26	10,8	569,01
4	1500	12,52	69,72	95	0	26	10,6	568,46
5	1500	12,53	68,98	88	0	26	10,7	568,93
6	1500	12,53	69,84	96	0	26	10,8	568,89
7	1500	12,52	69,82	87	0	26	10,8	568,56
8	1500	12,53	68,67	134	0	27	10,9	568,95
9	1500	12,54	68,69	175	0	26	11	569,15
10	1500	12,53	69,73	109	0	26	10,9	568,90

d. Data pengujian Nginx dengan Apache Bench dilingkungan Virtual

Berikut adalah tabel data pengujian web server Nginx menggunakan Apache Bench pada lingkungan virtual dengan jumlah 500 user, 1000 user, dan 1500 user.

Tabel 16 Data Pengujian Nginx (Virtual – Apache Bench) – 500 user

PERCOBAAN KE	JUMLAH USER	NGINX					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM (%)	CPU(%)
1	500	14,76	51	33,886	669,1	26	12,9
2	500	14,68	48	34,064	665,6	27	7,15
3	500	14,71	48	33,999	666,88	27	12,9
4	500	14,74	49	33,912	668,55	27	11
5	500	14,73	22	33,941	668,01	26	5,58
6	500	14,48	54	34,521	656,78	26	12,4
7	500	14,21	51	35,174	644,59	26	12,4
8	500	14,12	49	35,414	640,22	26	12,1
9	500	14,18	53	35,256	643,08	26	8,75
10	500	14,09	45	35,484	638,96	26	8,35

Tabel 17 Data Pengujian Nginx (Virtual – Apache Bench) – 1000 user

PERCOBAAN KE	JUMLAH USER	NGINX					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM (%)	CPU(%)
1	1000	13,71	57	72,954	621,56	26	12
2	1000	13,46	58	74,301	610,3	26	11,9
3	1000	13,58	60	73,625	615,9	26	12,1
4	1000	13,21	75	75,7	599,02	26	11,5
5	1000	13,21	54	75,701	599,01	26	11,9
6	1000	13,24	64	75,532	600,35	26	11,5
7	1000	13,16	79	75,998	596,67	26	11,6
8	1000	13	55	76,949	589,3	26	11,5
9	1000	13,4	46	74,637	607,55	26	12,1
10	1000	13,47	55	74,212	611,03	26	11,9

Tabel 18 Data Pengujian Nginx (Virtual – Apache Bench) – 1500 user

PERCOBAAN KE	JUMLAH USER	NGINX					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM (%)	CPU(%)
1	1500	13,3	64	111,707	608,9	26	11,4
2	1500	12,77	74	117,466	579,05	26	11,3
3	1500	12,43	76	120,707	563,5	26	10,6
4	1500	12,6	103	119,043	571,,38	26	11,6
5	1500	12,36	88	121,339	560,57	26	11
6	1500	12,35	78	121,443	560,09	26	11,1
7	1500	12,35	77	121,411	560,23	26	11,1
8	1500	12,6	107	124,394	546,8	26	11
9	1500	12,29	83	122,03	557,39	26	10,7
10	1500	12,1	74	124,922	544,49	26	10,5

e. Data pengujian Apache dengan Apache Jmeter dilingkungan Docker

Berikut adalah tabel data pengujian web server Apache menggunakan Apache Jmeter pada lingkungan docker dengan jumlah 500 user, 1000 user, dan 1500 user.

Tabel 19 Data Pengujian Apache (Docker – Apache Jmeter) – 500 user

PERCOBAAN KE	JUMLAH USER	APACHE						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU(%)	Network Troughput (KB/s)
1	500	4,17	261,31	202	0	5,58	22,51	146,71
2	500	4,17	257,51	160	0	5,68	22,43	146,77
3	500	4,17	267,47	158	0	5,61	23,39	146,78
4	500	4,18	260,61	198	0	5,71	22,17	146,88
5	500	4,17	261,33	227	0	5,63	22,39	146,80
6	500	4,18	252,47	229	0	5,64	21,1	146,86
7	500	4,17	264,62	208	0	5,66	23,98	146,82
8	500	4,18	249,96	185	0	5,66	21,09	146,85
9	500	4,17	252,12	201	0	5,68	21,87	146,77
10	500	4,18	259,75	208	0	5,67	22,27	146,89

Tabel 20 Data Pengujian Apache (Docker – Apache Jmeter) – 1000 user

PERCOBAAN KE	JUMLAH USER	APACHE						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Troughput (KB/s)
1	1000	4,25	63359,6	126848	78,4	49,95	49,95	38,94
2	1000	4,33	53608,14	120965	65	50,3	50,01	58,96
3	1000	8,34	297,57	929	0	9,51	45,66	293,27
4	1000	8,34	269,47	395	0	8,07	45,22	293,38
5	1000	3,98	68500,07	141335	74,9	50,03	50,03	41,11
6	1000	8,19	503,53	4629	0	19,53	45,95	287,98
7	1000	8,34	263,57	278	0	8,1	44,61	293,28
8	1000	8,35	261,78	476	0	8,13	39,3	293,52
9	1000	8,34	260,03	351	0	8,24	43,92	293,44
10	1000	8,34	266,3	288	0	8,26	44,99	293,37

Tabel 21 Data Pengujian Apache (Docker – Apache Jmeter) – 1500 user

PERCOBAAN KE	JUMLAH USER	APACHE						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Troughput (KB/s)
1	1500	4,41	125607,7	267302	90,93	49,78	50,02	22,39
2	1500	5,01	115607,7	237302	88,12	49,91	49,95	23,20
3	1500	4,45	120813,77	267095	88,07	50,03	50	26,86
4	1500	4,65	119268,89	234724	92,27	50,03	50,05	21,71
5	1500	4,45	117144,7	257905	90,27	50,03	49,94	23,59
6	1500	4,24	130841,36	275286	91,13	50,03	49,95	21,32
7	1500	4,41	125907,8	264999	90,67	50,03	50,03	22,93
8	1500	4,46	122479,72	263359	92,07	50,03	50,04	21,10
9	1500	12,54	68,69	175	0	50,03	49,88	569,15
10	1500	4,3	123929,77	276481	87,87	50,03	50,02	26,22

f. Data pengujian Apache dengan Apache Bench dilingkungan Docker

Berikut adalah tabel data pengujian web server Apache menggunakan Apache Bench pada lingkungan docker dengan jumlah 500 user, 1000 user, dan 1500 user.

Tabel 22 Data Pengujian Apache (Docker – Apache Bench) – 500 user

PERCOBAAN KE	JUMLAH USER	APACHE					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM (%)	CPU (%)
1	500	3,97	1049	126,022	139,31	8,37	20,59
2	500	4,15	167	120,616	145,55	8,37	20,33
3	500	4,3	184	116,323	150,93	8,37	20,91
4	500	3,82	177	130,969	134,05	8,4	20,66
5	500	3,99	121	125,402	140	8,41	20,85
6	500	3,92	178	127,417	137,78	8,42	20,72
7	500	4,08	199	122,416	143,41	8,43	20,67
8	500	3,94	175	127,001	138,24	8,54	20,68
9	500	4,08	236	122,444	143,38	8,54	20,53
10	500	3,76	140	132,885	132,12	8,47	20,9

Tabel 23 Data Pengujian Apache (Docker – Apache Bench) – 1000 user

PERCOBAAN KE	JUMLAH USER	APACHE					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM (%)	CPU (%)
1	1000	3,81	193	262,161	133,93	8,48	20,89
2	1000	3,81	192	262,277	133,87	8,5	20,72
3	1000	3,82	183	261,455	134,3	8,54	20,95
4	1000	3,86	152	259,103	135,51	8,55	20,66
5	1000	3,89	186	256,282	136,7	8,58	20,87
6	1000	4,01	1000	249,655	140,64	8,61	20,69
7	1000	3,75	254	266,372	131,82	8,64	20,9
8	1000	3,87	162	258,576	135,79	8,65	20,63
9	1000	3,87	135	258,693	135,73	8,66	20,87
10	1000	3,91	105	255,859	137,23	8,69	20,82

Tabel 24 Data Pengujian Apache (Docker – Apache Bench) – 1500 user

PERCOBAAN KE	JUMLAH USER	APACHE					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM (%)	CPU (%)
1	1500	3,82	158	392,443	134,21	8,7	20,81
2	1500	3,85	197	389,722	135,14	8,74	20,76
3	1500	3,88	377	386,477	136,28	8,76	20,71
4	1500	3,93	190	381,834	137,94	8,81	20,92
5	1500	3,85	172	389,966	135,06	8,83	20,8
6	1500	3,84	1487	391,094	134,67	8,87	21,08
7	1500	3,87	171	387,124	136,05	8,9	20,66
8	1500	3,92	167	382,179	137,81	8,94	20,81
9	1500	3,95	112	379,681	138,72	8,95	20,72
10	1500	3,91	141	383,718	137,26	8,99	20,78

g. Data pengujian Apache dengan Apache Bench dilingkungan Virtual

Berikut adalah tabel data pengujian web server Apache menggunakan Apache Jmeter pada lingkungan virtual dengan jumlah 500 user, 1000 user, dan 1500 user.

Tabel 25 Data Pengujian Apache (Virtual – Apache Jmeter) – 500 user

PERCOBAAN KE	JUMLAH USER	APACHE						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Troughput (KB/s)
1	500	4,18	74,47	84	0	19	4,44	189,75
2	500	4,18	77,1	162	0	18	4,22	189,84
3	500	4,18	74,63	86	0	18	4,54	189,72
4	500	4,18	74,11	74	0	18	4,45	189,71
5	500	4,18	73,07	76	0	19	4,29	189,70
6	500	4,18	73,7	139	0	19	4,25	189,84
7	500	4,18	75,79	88	0	19	4,49	189,70
8	500	4,18	77,32	136	0	19	4,3	189,77
9	500	4,18	76,45	79	0	19	4,25	189,72
10	500	4,18	76,25	92	0	19	4,28	189,74

Tabel 26 Data Pengujian Apache (Virtual – Apache Jmeter) – 1000 user

PERCOBAAN KE	JUMLAH USER	APACHE						
		Troughput (tps)	Delay (ms)	Jitter(ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Troughput (KB/s)
1	1000	8,36	74,05	99	0	18	8,13	379,40
2	1000	8,35	74,23	188	0	19	8,12	379,04
3	1000	8,35	74,27	113	0	19	8,28	379,14
4	1000	8,35	75,03	99	0	18	7,98	379,09
5	1000	8,35	73,18	81	0	19	8,08	379,18
6	1000	8,35	74,44	101	0	18	8,18	379,14
7	1000	8,35	74,81	89	0	18	8,61	379,18
8	1000	8,35	75,69	131	0	19	8,65	379,17
9	1000	8,36	74,57	103	0	19	7,85	378,28
10	1000	8,36	74,39	105	0	18	8,02	379,37

Tabel 27 Data Pengujian Apache (Virtual – Apache Jmeter) – 1500 user

PERCOBAAN KE	JUMLAH USER	APACHE						
		Troughput (tps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	RAM (%)	CPU (%)	Network Troughput (KB/s)
1	1500	12,53	76,28	125	0	19	12,4	568,65
2	1500	12,53	73,38	128	0	18	11,9	568,69
3	1500	12,53	73,56	96	0	18	11,6	568,77
4	1500	12,53	74,32	109	0	18	12	568,63
5	1500	12,52	73,43	125	0	18	11,9	568,42
6	1500	12,53	73,1	108	0	18	12	568,80
7	1500	12,54	74,32	119	0	18	11,9	568,95
8	1500	12,53	73,12	92	0	18	12	568,66
9	1500	12,53	74,62	125	0	18	13,6	568,83
10	1500	12,53	73,59	110	0	19	11,8	568,57

h. Data pengujian Apache dengan Apache Bench dilingkungan Virtual

Berikut adalah tabel data pengujian web server Apache menggunakan Apache Bench pada lingkungan virtual dengan jumlah 500 user, 1000 user, dan 1500 user.

Tabel 28 Data Pengujian Apache (Virtual – Apache Bench) – 500 user

PERCOBAAN KE	JUMLAH USER	APACHE					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM(%)	CPU(%)
1	500	13,81	34	36,202	626,3	18	13,2
2	500	13,86	49	36,086	628,31	19	12,9
3	500	13,57	45	36,857	615,17	19	13
4	500	12,54	114	39,879	568,55	18	21,4
5	500	13,43	53	37,236	608,92	19	12,5
6	500	13,51	49	37,014	612,56	19	12,6
7	500	13,43	65	37,238	608,87	19	12,7
8	500	13,31	57	37,337	603,71	19	12,3
9	500	13,59	45	36,786	626,36	29	13
10	500	13,42	58	37,269	608,37	19	12,5

Tabel 29 Data Pengujian Apache (Virtual – Apache Bench) – 1000 user

PERCOBAAN KE	JUMLAH USER	APACHE					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM (%)	CPU (%)
1	1000	13,22	52	75,671	599,26	18	13,3
2	1000	13,53	55	73,895	613,67	19	13,1
3	1000	13,68	56	73,094	620,39	19	12,8
4	1000	13,53	56	73,914	613,5	19	13
5	1000	13,3	67	75,172	603,24	18	12,6
6	1000	13,01	58	76,861	589,98	19	11,9
7	1000	13,31	64	75,156	603,37	19	12,3
8	1000	13,01	56	76,864	589,96	19	12,1
9	1000	13,2	54	75,781	598,39	18	12,8
10	1000	13,03	57	76,774	590,65	19	12,9

Tabel 30 Data Pengujian Apache (Virtual – Apache Bench) – 1500 user

PERCOBAAN KE	JUMLAH USER	APACHE					
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (KB/s)	RAM (%)	CPU (%)
1	1500	13,01	92	115,339	589,74	18	12,7
2	1500	13,24	99	113,297	600,37	18	12,7
3	1500	12,97	61	115,672	588,04	19	12,4
4	1500	12,96	53	115,752	587,64	18	12,3
5	1500	12,78	81	117,333	579,71	18	12,2
6	1500	13,2	60	113,632	598,6	18	12,2
7	1500	13,4	54	111,914	607,79	19	11,6
8	1500	13,52	69	110,915	613,27	19	12,9
9	1500	14,22	52	105,516	644,64	19	13
10	1500	14,38	48	104,329	651,97	18	13,2

4.1.2. Analisis Statistik Data

Untuk menentukan perbandingan antara web server Apache dan Nginx maka dilakukan perhitungan hasil performansi dengan analisa statistika. Statistika merupakan bentuk bilangan yang disusun dalam daftar atau tabel yang menggambarkan suatu kejadian, datanya diperoleh dari hasil pengamatan yang disajikan dalam bentuk bilangan-bilangan. Menurut Lutfiana (2020) statistik ini juga

merupakan sekumpulan data yang digunakan dalam menjelaskan masalah dan menarik kesimpulan yang benar, tentunya melalui beberapa proses.

Lutfiana (2020) mengatakan statistik meliputi pengumpulan data, pengolahan sebuah data, menganalisis sebuah data, dan menarik kesimpulan berdasarkan data yang ada. Data adalah sejumlah informasi yang dapat memberikan gambaran mengenai suatu keadaan bahan suatu masalah. Sebuah data bisa dikatakan memenuhi syarat jika obyektif yang artinya jika data dikumpulkan harus ada gambaran keadaan sebenarnya, kedua relevan yang artinya jika sebuah data yang dikumpulkan mempunyai kaitannya dengan sebuah permasalahan yang akan diteliti, ketiga representatif yang artinya jika data yang dikumpulkan melalui teknik sampling, sesuai zaman, harus bisa mewakili dan menggambarkan keadaan populasinya, dapat dipercaya yang artinya jika sebuah data yang dikumpulkan diperoleh dari sumber yang akurat. Berikut adalah tabel jumlah perhitungan statistika dengan menghitung rata – rata dari data yang telah disajikan, hasil setiap parameter yang sudah ditentukan telah dikerucutkan menjadi 4 tabel sesuai dengan alat uji dan lingkungan web server masing - masing.

Tabel 31 Rata rata Apache dan Nginx dilingkungan Docker dengan Apache Jmeter

NO	JUMLAH USER	NGINX							ALAT PENGUJIAN
		Troughput (tps)	Delay(ms)	Jitter(ms)	Packet Loss (%)	RAM(%)	CPU(%)	Network Troughput (Kb/s)	
1	500	4,18	90,944	129,9	0	4	5	146,829	APACHE JMETER
2	1000	8,353	92,263	128,7	0	4	10	293,425	
3	1500	12,53	94,026	154	0	5	16	439,835	
NO	JUMLAH USER	APACHE							
		Troughput (tps)	Delay(ms)	Jitter(ms)	Packet Loss (%)	RAM(%)	CPU(%)	Network Troughput (Kb/s)	
1	500	4,174	258,715	197,6	0	6	22	146,813	
2	1000	7,08	18759,006	39649,4	22	22	46	218,275	
3	1500	4,486	122401,113	259781,2	90	50	50	77,847	

Tabel 32 Rata - rata Apache dan Nginx dilingkungan Docker dengan Apache Bench

NO	JUMLAH USER	NGINX						ALAT PENGUJIAN
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (Kb/s)	RAM(%)	CPU(%)	
1	500	11,681	100,9	42,8154	409,912	5	12	APACHE BENCH
2	1000	11,554	99,8	86,5761	405,461	5	14	
3	1500	11,779	120,7	127,4068	413,419	6	14	
NO	JUMLAH USER	APACHE						
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (Kb/s)	RAM(%)	CPU(%)	
1	500	4,001	262,6	125,1495	140,477	8	21	
2	1000	3,86	256,2	259,0433	135,552	9	21	
3	1500	3,882	317,2	386,4238	136,314	9	21	

Tabel 33 Rata - rata Apache dan Nginx dilingkungan Virtual dengan Apache Jmeter

NO	JUMLAH USER	NGINX							ALAT PENGUJIAN
		Troughput (tps)	Delay(ms)	Jitter(ms)	Packet Loss (%)	RAM(%)	CPU(%)	Network Troughput (Kb/s)	
1	500	4,18	69,48	109,9	0	27	4	189,813	APACHE JMETER
2	1000	8,357	68,761	96,1	0	27	8	379,377	
3	1500	12,529	69,1588889	124,5	0	26	10	568,806	
NO	JUMLAH USER	APACHE							
		Troughput (tps)	Delay(ms)	Jitter(ms)	Packet Loss (%)	RAM(%)	CPU(%)	Network Troughput (Kb/s)	
1	500	4,18	75,289	101,6	0	19	4	189,749	
2	1000	8,353	74,446	110,9	0	19	8	379,099	
3	1500	12,53	73,972	113,7	0	18	12	568,697	

Tabel 34 Rata - rata Apache dan Nginx dilingkungan Virtual dengan Apache Bench

NO	JUMLAH USER	NGINX						ALAT PENGUJIAN
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (Kb/s)	RAM(%)	CPU(%)	
1	500	14,47	47	34,5651	656,177	26	10	APACHE BENCH
2	1000	13,344	60,3	74,9609	605,069	26	12	
3	1500	12,515	82,4	120,4462	564,5577778	26	11	
NO	JUMLAH USER	APACHE						
		Troughput (tps)	Jitter (ms)	Time Taken (s)	Network Troughput (Kb/s)	RAM(%)	CPU(%)	
1	500	13,447	56,9	37,1904	610,712	10	14	
2	1000	13,282	57,5	75,3182	602,241	19	13	
3	1500	13,368	66,9	112,3699	606,177	18	13	

BAB V HASIL DAN PEMBAHASAN

5.1. Hasil Penelitian

Penelitian ini menggunakan metode *Network Development Life Cycle*. Metode *Network Development Life Cycle* seperti yang sudah dijelaskan pada bab 3, selanjutnya ditambahkan juga sebuah metode analisis statistik deskriptif untuk mengambil nilai rata – rata dari setiap pengujian, metode ini telah dilakukan pada bab 4, untuk hasil hipotesa maka pada bab 5 ini akan melakukan uji T student dengan data sampel independen. Pengujian uji T dilakukan dengan mengambil hipotesa awal **H0** yaitu tidak terdapat perbedaan signifikan dari data statistik pengujian yang dilakukan, artinya hasil dari uji T akan menunjukkan bahwa ada atau tidaknya perbedaan signifikan antara kelompok-kelompok yang diuji. Hipotesa ini dibantu dengan bahasa pemograman python untuk mencari hasil hipotesis kerja **H1**. Sebelum mencapai hasil hipotesa kerja **H1** maka dicari terlebih dahulu **T-hitung** dan ukuran probabilitasnya. **T-hitung** adalah nilai yang dihitung dari data sampel menggunakan rumus uji T, sedangkan ukuran probabilitas adalah ukuran yang digunakan dalam statistik untuk membantu menentukan signifikansi hasil dari suatu uji hipotesis. Asumsi penting dalam uji t ini adalah bahwa kedua kelompok memiliki variansi yang sama (*equal variance*). Interpretasi dari uji T ini adalah jika ukuran probabilitas lebih kecil dari tingkat signifikasi maka **H1** akan menghasilkan terima **H0** yaitu tidak terdapat perbedaan yang signifikan pada data statistik pengujian, dan sebaliknya jika ukuran propabilitas lebih besar dari tingkat signifikasi maka **H1** akan menghasilkan tolak **H0** ada perbedaan yang signifikan pada data statistik pengujian. Berikut adalah rumus secara tertulis untuk melakukan uji T student.

Rumus:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Dimana:

- \bar{X}_1 dan \bar{X}_2 = rata – rata sampel dari dua kelompok
- s_1 dan s_2 = standar deviasi sampel dari dua kelompok
- n_1 dan n_2 = ukuran sampel dari dua kelompok

Rumus Standar Deviasi

$$s = \sqrt{\frac{\sum(X_i - \bar{X})^2}{n - 1}}$$

5.1.1. Hasil Uji T Student

Berikut adalah hasil uji T *student* yang dilakukan dengan bahasa pemrograman python3. Python ini memiliki pustaka *scipy* yang menyediakan fungsi statistik. Berikut hasil hipotesa yang telah dilakukan.

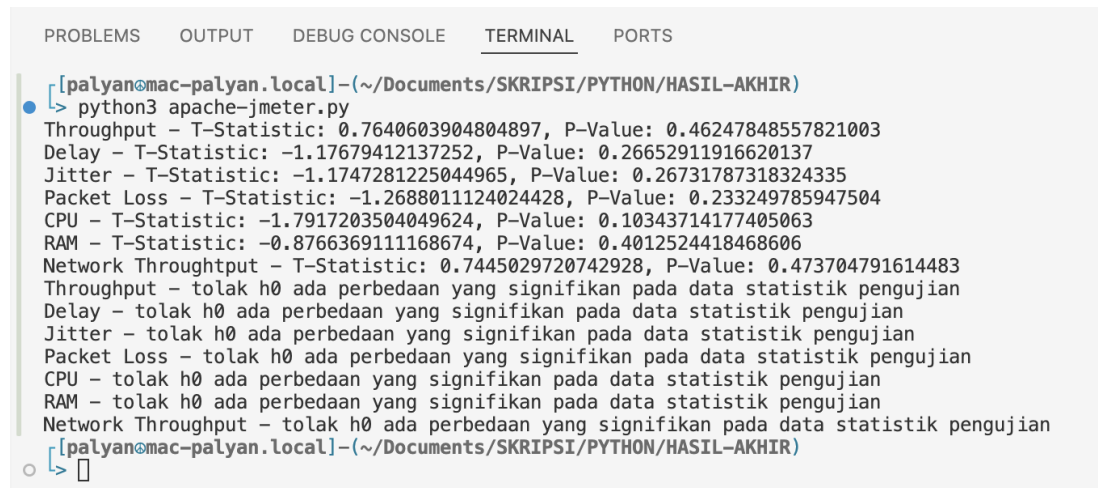
- a. Hasil uji T *Quality of Service* Nginx dan Apache pada lingkungan docker dan virtual dengan Apache Jmeter.

Diketahui:

H₀ = Tidak terdapat perbedaan yang signifikan pada data statistik pengujian

alpha = 0,05 (5 %) tingkat signifikansi

Berikut hasil pengukuran dengan python yang dijalankan untuk mencari hasil **T-hitung** dan sekaligus memberikan output **H₁**. ukuran probabilitas diwakilkan oleh **p_value**, sedangkan **T-Hitung** diwakilkan **T-statistic**, serta tingkat signifikansi diwakilkan oleh variabel **alpha**. Hasil pengukuran menunjukkan seperti gambar 11.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[palyan@mac-palyan.local]-(~/Documents/SKRIPSI/PYTHON/HASIL-AKHIR)
└─> python3 apache-jmeter.py
Throughput - T-Statistic: 0.7640603904804897, P-Value: 0.46247848557821003
Delay - T-Statistic: -1.17679412137252, P-Value: 0.26652911916620137
Jitter - T-Statistic: -1.1747281225044965, P-Value: 0.26731787318324335
Packet Loss - T-Statistic: -1.2688011124024428, P-Value: 0.233249785947504
CPU - T-Statistic: -1.7917203504049624, P-Value: 0.10343714177405063
RAM - T-Statistic: -0.8766369111168674, P-Value: 0.4012524418468606
Network Throughput - T-Statistic: 0.7445029720742928, P-Value: 0.473704791614483
Throughput - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
Delay - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
Jitter - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
Packet Loss - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
CPU - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
RAM - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
Network Throughput - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
[palyan@mac-palyan.local]-(~/Documents/SKRIPSI/PYTHON/HASIL-AKHIR)
└─> □
```

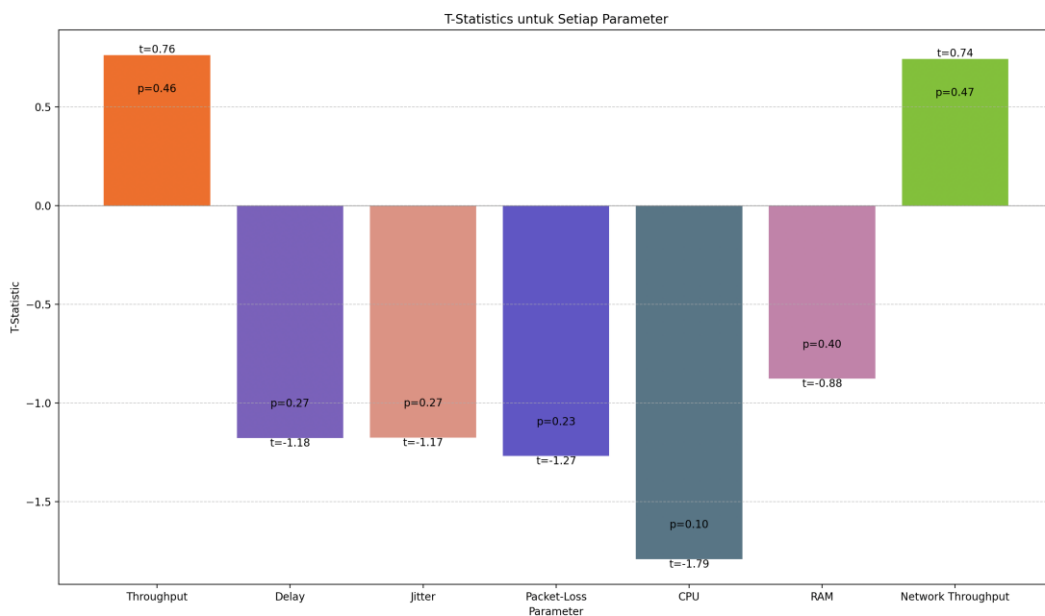
Gambar 11 Hasil uji T Student Apache dan Nginx menggunakan Apache Jmeter

Jika dituliskan kedalam bentuk tabel hasilnya seperti tabel 36.

Tabel 35 Hasil Uji T Student Apache dan Nginx Menggunakan Apache Jmeter

No	PARAMETER	T-Statistic (T-Hitung)	P-Value	Result
1	Throughput	0.7640603904804897	0.46247848557821003	tolak H0 ada perbedaan yang signifikan pada data statistik uji
2	Delay	-1.17679412137252	0.26652911916620137	tolak H0 ada perbedaan yang signifikan pada data statistik uji
3	Jitter	-1.1747281225044965	0.26731787318324335	tolak H0 ada perbedaan yang signifikan pada data statistik uji
4	Packet Loss	-1.2688011124024428	0.233249785947504	tolak H0 ada perbedaan yang signifikan pada data statistik uji
5	CPU	-1.7917203504049624	0.10343714177405063	tolak H0 ada perbedaan yang signifikan pada data statistik uji
6	RAM	-0.8766369111168674	0.4012524418468606	tolak H0 ada perbedaan yang signifikan pada data statistik uji
7	Network Throughput	0.7445029720742928	0.473704791614483	tolak H0 ada perbedaan yang signifikan pada data statistik uji

Hasil uji T *student* dengan statistik deskriptif perlu digambarkan dalam bentuk grafik, berikut hasil grafik pengujian seperti gambar 12.



Gambar 12 Grafik hasil Uji T *Student* Apache dan Nginx Menggunakan Apache Jmeter

Grafik pada gambar 12 memvisualisasikan ukuran **p_value** dan nilai **T-hitung**, jika **T-hitung** bernilai negatif itu berarti total rata – rata pada objek kedua lebih besar dari pada objek pertama, dan pada pengujian ini objek pertama adalah Web Server Nginx dan objek kedua Web Server Apache.

- b. Hasil uji T *Quality of Service* Nginx dan Apache pada lingkungan docker dan virtual dengan Apache Bench.

Diketahui:

H₀ = Tidak terdapat perbedaan yang signifikan pada data statistik pengujian

T-tabel = 0,05 (5 %) tingkat signifikansi

Berikut adalah *script* python yang dijalankan untuk mencari hasil **T-hitung** dan sekaligus memberikan output **H₁**. ukuran probabilitas diwakilkan oleh **p_value**, sedangkan **T-Hitung** diwakilkan **T-statistic**, serta tingkat signifikansi diwakilkan oleh variabel **alpha**. Hasil pengukuran menunjukkan seperti gambar 13.

```

[~] [palyan@mac-palyan.local] (~/.Documents/SKRIPSI/PYTHON/HASIL-AKHIR)
└─> python3 apache-bench.py
Throughput - T-Statistic: 1.8094261826233207, P-Value: 0.10049630483445104
Network Throughput - T-Statistic: 1.1977651479423532, P-Value: 0.2586272337890861
Jitter - T-Statistic: -1.6592927801827668, P-Value: 0.12804559937129006
Time Taken - T-Statistic: -1.5147248778061257, P-Value: 0.16079115895839954
CPU - T-Statistic: -2.7160723812755565, P-Value: 0.02170617431949711
RAM - T-Statistic: 0.7231210564144186, P-Value: 0.4861747617810296
Throughput - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
Network Throughput - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
Jitter - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
Time Taken - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian
CPU - terima H0 tidak ada perbedaan yang signifikan pada data statistik pengujian
RAM - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian

```

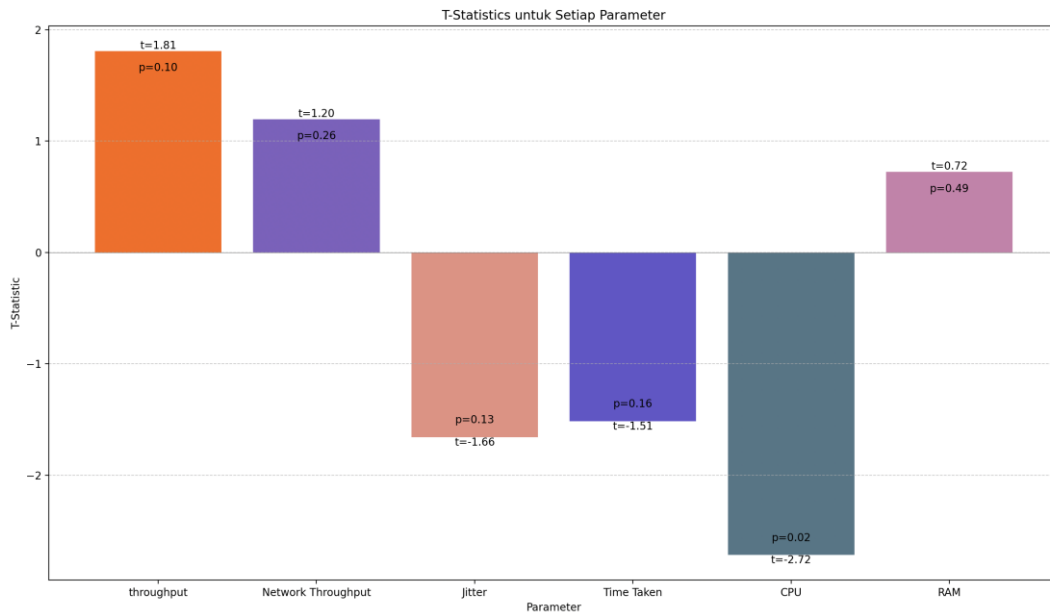
Gambar 13 Hasil uji T *Student* Apache dan Nginx menggunakan Apache Bench

Jika dituliskan kedalam bentuk tabel hasilnya seperti tabel 37.

Tabel 36 Hasil Uji T *Student* Apache dan Nginx Menggunakan Apache Bench

No	PARAMETER	T-Statistic (T-count)	P-Value	Result
1	Throughput	1.8094261826233207	0.10049630483445104	tolak H ₀ ada perbedaan yang signifikan pada data statistik uji
2	Network Troughput	1.1977651479423532	0.2586272337890861	tolak H ₀ ada perbedaan yang signifikan pada data statistik uji
3	Jitter	-1.6592927801827668	0.12804559937129006	tolak H ₀ ada perbedaan yang signifikan pada data statistik uji
4	Time Taken	-1.5147248778061257	0.16079115895839954	tolak H ₀ ada perbedaan yang signifikan pada data statistik uji
5	CPU	-2.7160723812755565	0.02170617431949711	terima H ₀ tidak ada perbedaan yang signifikan pada data uji statistik
6	RAM	0.7231210564144186	0.4861747617810296	tolak H ₀ ada perbedaan yang signifikan pada data statistik uji

Hasil uji *T student* dengan statistik deskriptif perlu digambarkan dalam bentuk grafik, berikut hasil grafik pengujian seperti gambar 14.



Gambar 14 Grafik hasil Uji T Student Apache dan Nginx Menggunakan Apache Bench

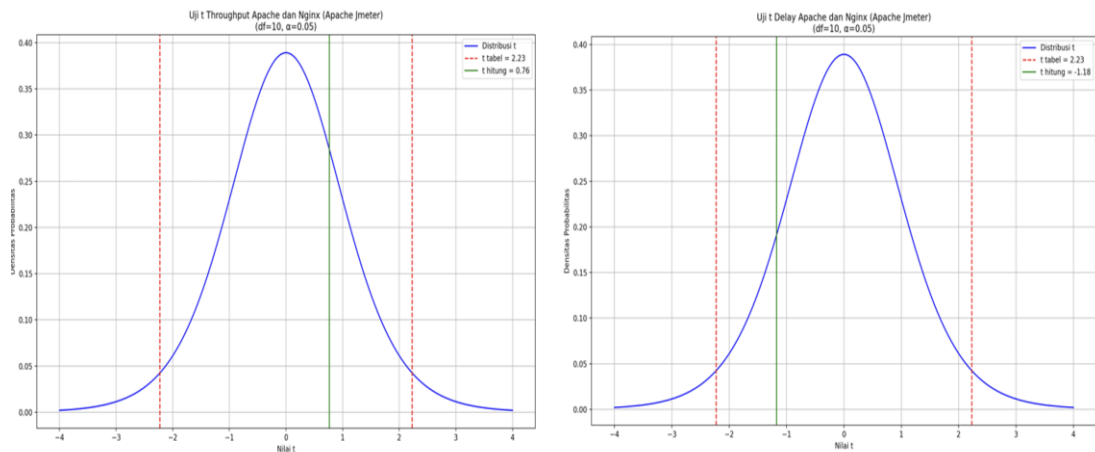
Grafik pada gambar 14 memvisualisasikan ukuran **p_value** dan nilai **T-hitung**, jika T-hitung bernilai negatif itu berarti total rata – rata pada objek kedua lebih besar dari pada objek pertama, dan pada pengujian ini objek pertama adalah Web Server Nginx dan objek kedua Web Server Apache.

5.1.2. Hasil uji T-Hitung dan T-Table

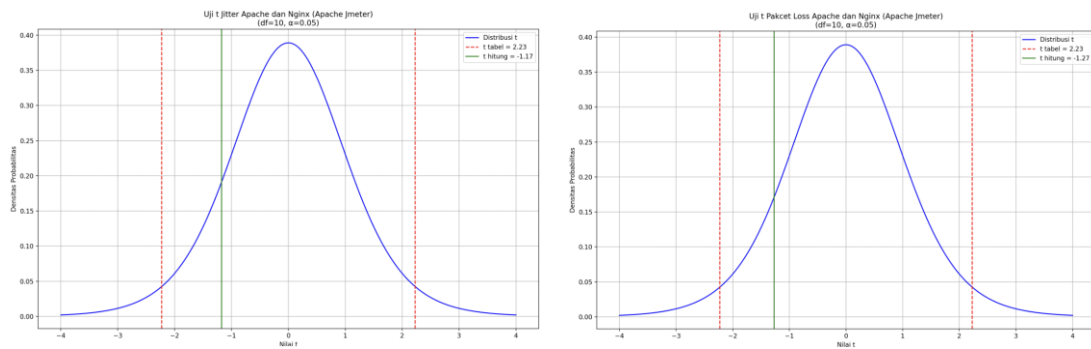
Hasil uji T-hitung dan T-tabel akan dikemas kedalam bentuk grafik agar memudahkan untuk memahami secara visual, berikut adalah hasil grafik dari setiap parameter yang telah diukur.

a. Hasil uji T-Hitung dan T-Table dengan Apache Jmeter

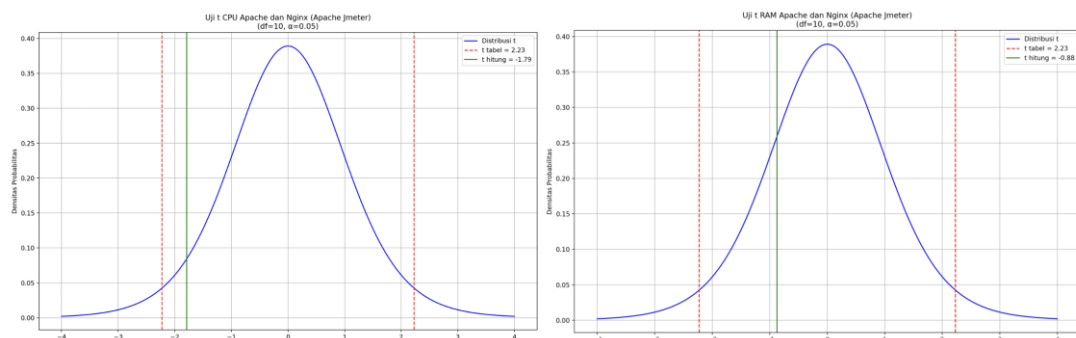
Berikut adalah grafik uji T parameter QOS Apache dan Nginx menggunakan alat uji test Apache Jmeter.



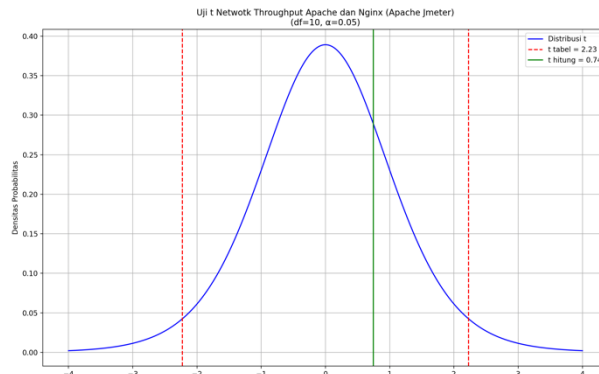
Gambar 15 Grafik T-tabel dan T-hitung throughput dan Delay Apache dan Nginx dengan Apache Jmeter



Gambar 16 Grafik T-tabel dan T-hitung Jitter dan Packet Loss Apache dan Nginx dengan Apache Jmeter



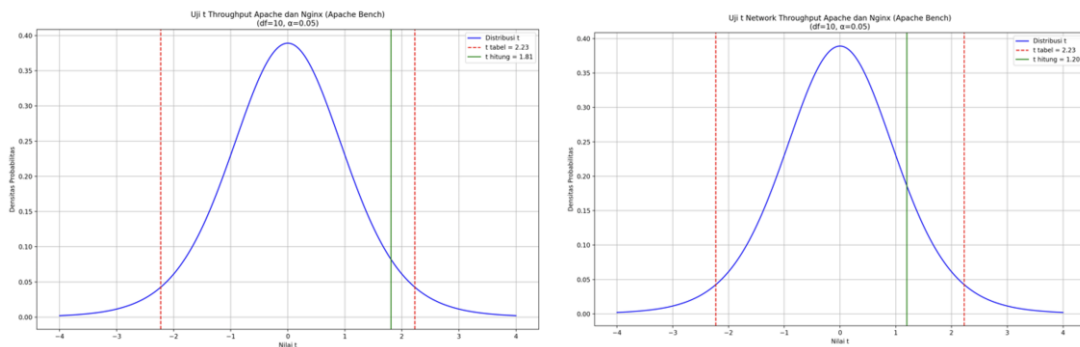
Gambar 17 Grafik T-tabel dan T-hitung CPU dan RAM Apache dan Nginx dengan Apache Jmeter



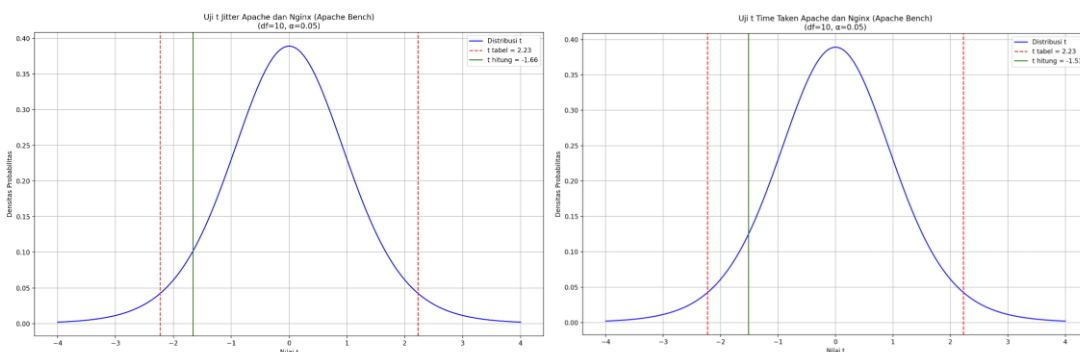
Gambar 18 Grafik T-tabel dan T-hitung Network Throughput Apache dan Nginx dengan Apache Jmeter

b. Hasil uji T-Hitung dan T-Table dengan Apache Jmeter

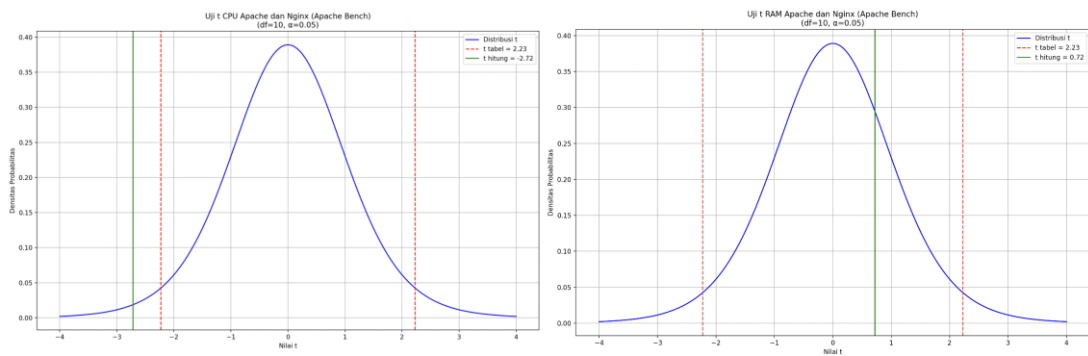
Berikut adalah grafik uji T parameter QOS Apache dan Nginx menggunakan alat uji test Apache Bench.



Gambar 19 Grafik T-tabel dan T-hitung Throughput dan Network Throughput Apache dan Nginx dengan Apache Bench



Gambar 20 Grafik T-tabel dan T-hitung Jitter dan Time Taken Apache dan Nginx dengan Apache Bench



Gambar 21 Grafik T-tabel dan T-hitung CPU dan RAM Apache dan Nginx dengan Apache Bench

5.2. Pembahasan Hasil Penelitian

5.2.1. Lingkungan docker dan Virtual menggunakan Apache Jmeter

Pengujian Nginx dan Apache untuk semua QOS pada lingkungan docker dan virtual menggunakan Apache Jmeter adalah tolak **H0** ada perbedaan yang signifikan artinya **H1** menolak **H0**. Parameter - parameter QOS yang diukur menunjukkan hasil ada perbedaan yang signifikan atau ukuran signifikan (p_value) \geq tingkat signifikansi (α). Pengujian ini menunjukkan bahwa salah satu dari web server tersebut ada yang lebih baik performansinya.

5.2.2. Lingkungan docker dan Virtual menggunakan Apache Bench

Pengujian Nginx dan Apache untuk semua QOS pada lingkungan docker dan virtual menggunakan Apache Bench adalah tolak **H0** ada perbedaan yang signifikan artinya **H1** menolak **H0**. Parameter – parameter QOS yang diukur menunjukkan hasil ada perbedaan yang signifikan atau ukuran signifikan (p_value) \geq tingkat signifikansi (α). Namun ada salah satu parameter yang menerima **H0** yaitu pada bagian CPU. Pengujian ini menunjukkan bahwa ada bukti yang cukup untuk menyatakan bahwa ada perbedaan atau efek yang diukur dalam parameter-parameter tersebut, dan dapat dinyatakan bahwa salah satu dari web server tersebut ada yang lebih baik performansinya.

BAB VI KESIMPULAN DAN SARAN

6.1. Kesimpulan

Penelitian ini dilakukan dengan harapan dapat memberikan hasil uji performansi pada kedua web server yang populer digunakan saat ini yaitu Nginx dan Apache dalam menangani permintaan layanan. Penelitian ini telah menerapkan metode *Network Development Life Cycle* untuk menganalisis data penelitian secara terstruktur mulai dari merencanakan dan mengolah proses pengembangan sebuah sistem. Penelitian ini juga menggunakan analisis statistika sebagai bentuk pengolahan data agar dapat mengambil nilai rata – rata dari setiap sampel pengujian, dan pengukuran signifikansi menggunakan uji T Student dengan data sampel independen.

Dari pengujian ini, dapat disimpulkan bahwa ada bukti yang cukup untuk menyatakan bahwa ada perbedaan atau efek yang diukur dalam parameter-parameter tersebut, dan dapat dinyatakan bahwa salah satu dari web server tersebut ada yang lebih baik performansinya. Namun hanya satu dari enam parameter menunjukkan perbedaan yaitu menerima **H0** ketika pengujian dengan Apache Bench. Pengujian dengan Apache Bench menunjukan parameter CPU terima **H0** tidak terdapat perbedaan yang signifikan pada data statistik pengujian, tetapi hal ini tidak menunjukan bukti yang cukup kuat untuk menganggap bahwa kedua web server tersebut sama performansinya. Maka dari sampel yang telah diuji dapat ditarik kesimpulan bahwa salah satu dari web server ini ada yang lebih baik performansinya, yaitu web server Nginx karena memiliki performansi QOS yang lebih efisien dibandingkan dengan QOS web server Apache.

6.2. Saran

Berdasarkan penelitian ini, beberapa saran yang dapat diberikan:

- a. Perlu tambahan pengaturan konfigurasi untuk setiap web server yang dilakukan baik itu pada lingkungan docker maupun virtual agar mendapatkan hasil performansi yang maksimal, pada penelitian ini masih menggunakan konfigurasi bawaan pabrik (*default*).
- b. Pemilihan dari web server antara nginx dan apache sebaiknya dilihat dari kebutuhan masing - masing aplikasi, jika dilihat dari performansi kecepatan nilai QOS maka web server yang cocok untuk menangani aplikasi PHP adalah web server Nginx.
- c. Berdasarkan hasil yang diperoleh maka penelitian ini dapat dilanjutkan dengan menguji performansi kedua web server tersebut dari aspek lain seperti *concurrency level* dan jumlah url yang diakses lebih tinggi atau ditingkat pada waktu yang bersamaan.
- d. Percobaan penelitian bisa diukur juga dengan alat uji selain Apache Jmeter dan Apache Bench seperti K6, Load Runner dan lain – lain.
- e. Endpoint uji API bisa dilakukan dengan metode POST.
- f. Mengacu kepada keterbatasan biaya dalam segi resource penggunaan RAM pada Apache lebih efisien, Jika ingin lebih mengurangi biaya resource web server apache bisa sebagai pilihan.

DAFTAR PUSTAKA

- Adnan, F. (2016). *ANALISIS PERBANDINGAN PERFORMA WEB SERVER APACHE DAN NGINX MENGGUNAKAN HTTPERF PADA VPS DENGAN SISTEM OPERASI CENTOS*. Retrieved from <http://www.kegel.com/c10k.html>
- Agustine, Lady, & Seimahaira, S. (2023). Penerapan Metode SAW dalam Analisa Perbandingan Performa Web server (Apache, Nginx, Lighttpd, Iis) pada Bahasa Pemrograman PHP. *Remik*, 7(1), 409–420. doi: 10.33395/remik.v7i1.12075
- Ammar, F., & Hanafi, H. (2016). Analisis Transfer Rate Wireless Local Area Network Dengan Standar Ieee 802.11a Dan Ieee 802.11G Pada Kanal Line of Sight. *Jurnal Ecotipe (Electronic, Control, Telecommunication, Information, and Power Engineering)*, 3(1), 31–39. doi: 10.33019/ecotipe.v3i1.28
- Apa itu Throughput? Arti, Fungsi, Contoh, FAQs 2024 | RevoU*. (2024). Retrieved from <https://revou.co/kosakata/throughput>
- Apache JMeter - Apache JMeterTM*. (2024). Retrieved from <https://jmeter.apache.org/>
- Aprianto Budiman, M. Ficky Duskarnaen, & Hamidillah Ajie. (2020a). Analisis Quality of Service (Qos) Pada Jaringan Internet Smk Negeri 7 Jakarta. *PINTER : Jurnal Pendidikan Teknik Informatika Dan Komputer*, 4(2), 32–36. doi: 10.21009/pinter.4.2.6
- Aprianto Budiman, M. Ficky Duskarnaen, & Hamidillah Ajie. (2020b). Analisis Quality of Service (Qos) Pada Jaringan Internet Smk Negeri 7 Jakarta. *PINTER : Jurnal Pendidikan Teknik Informatika Dan Komputer*, 4(2), 32–36. doi: 10.21009/pinter.4.2.6
- Azi, M. N. A., Arifwidodo, B., & Wahyudi, E. (2023). Analisis Performansi Web Server Saat Menangani Permintaan Client Menggunakan Metode Reserve Proxy Caching dan Varnish. *Journal of Telecommunication, Electronics, and Control Engineering (JTECE)*, 5(1), 14–21. doi: 10.20895/jtece.v5i1.843
- Busran, & Ridwan, A. (2020). Analisis Perbandingan Performa Apache Web Server dan Nginx Menggunakan Apache Jmeter. *Jurnal Teknoif*, 8(2), 87–92.
- Chandra, A. Y. (2019). Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request. *Jurnal Sistem Dan Informatika (JSI)*, 14(1), 48–56. doi: 10.30864/jsi.v14i1.248
- Dame, A. H., Zailani, A. U., Kunci, K., Implementasi, :, & Server, W. (2023). Implementasi Webserver Berbasis Docker Dan Linux. *Jurnal Ilmu Komputer Dan Pendidikan*, 1(5), 1084–1090. Retrieved from <https://journal.mediapublikasi.id/index.php/logic>
- Danil Fajri, R., & Djutalov, R. (2023). Implementasi Jaringan Hotspot Menggunakan Mikrotik untuk RT RW.Net Dengan Menggunakan Metode Network Development Life Cycle (NDLC) Pada Kampung Kelapa Indah Tangerang. In *Jurnal Ilmu Komputer dan Pendidikan* (Vol. 1, Issue 6). Retrieved from <https://journal.mediapublikasi.id/index.php/logic>
- Dwiyatno, S., Rakhmat, E., & Gustiawan, O. (2020). *IMPLEMENTASI VIRTUALISASI SERVER BERBASIS DOCKER CONTAINER*. 7(2).
- February 2024 Web Server Survey | Netcraft*. (2024). Retrieved from <https://www.netcraft.com/blog/february-2024-web-server-survey/>
- Jiwandono, A. (2021). *Analisa Perbandingan Kinerja Web Server Apache, Nginx, Dan Litespeed Dengan Menggunakan Metode Stress Test*. 1–78.

- Lutfiana, M. (2020). *ANALISIS JURNAL STATISTIKA DALAM PENGELOLAAN DATA DAN NILAI RAPORT SISWA DI SDN PANGKAH WETAN*.
- Oki, R. A. (2022). Analisis Perbandingan Performa Web Server Apache Dan Lighttpd Menggunakan Httperf Pada Vps Dengan Sistem Operasi Ubuntu Server. *Jurnal Vokasional Teknik Elektronika & Informatika*, 19.
- Output, P. (2021a). *PERKEMBANGAN TEKNOLOGI PERANGKAT KOMPUTER (MEMORY - PERKEMBANGAN TEKNOLOGI PERANGKAT KOMPUTER (MEMORY – PROCESSOR – INPUT OUTPUT) Oleh : I Gede Suputra Widharma. September. doi: 10.13140/RG.2.2.34916.30082*
- Output, P. (2021b). *PERKEMBANGAN TEKNOLOGI PERANGKAT KOMPUTER (MEMORY - PERKEMBANGAN TEKNOLOGI PERANGKAT KOMPUTER (MEMORY – PROCESSOR – INPUT OUTPUT) Oleh : I Gede Suputra Widharma. September. doi: 10.13140/RG.2.2.34916.30082*
- Perbedaan Nginx vs Apache: Pilih Server Web Mana yang Lebih Baik?* (2024). Retrieved from <https://www.exabytes.co.id/blog/perbedaan-nginx-vs-apache/>
- STATISTIK DESKRIPTIF* Leni Masnidar Nasution. (2017).
- Unsong, H. C. Y., & Andjarwirawan, J. (2021). Analisa Kinerja Apache dan Nginx dalam Arsitektur Microservice Menggunakan Siege. *Jurnal Infra*, 09(02), 41–45.

LAMPIRAN

Script Python menguji Nginx dan Apache dilingkungan docker dan virtual dengan Apache Jmeter

```
from scipy import stats

#THROUGHPUT
throughput_NGINX = [4.18, 8.35, 12.53, 4.18, 8.35, 12.52]
throughput_APACHE = [4.17, 7.08, 4.48, 4.18, 8.35, 12.53]

t_statistic_throughput, p_value_throughput = stats.ttest_ind(throughput_NGINX, throughput_APACHE, equal_var=True)
print(f"Throughput - T-Statistic: {t_statistic_throughput}, P-Value: {p_value_throughput}")

#DELAY
delay_NGINX = [90.94, 92.26, 94.02, 69.48, 68.76, 69.15]
delay_APACHE = [258.715, 18759, 122401.11, 75.28, 74.44, 73.97]

t_statistic_delay, p_value_delay = stats.ttest_ind(delay_NGINX, delay_APACHE, equal_var=True)
print(f"Delay - T-Statistic: {t_statistic_delay}, P-Value: {p_value_delay}")

#JITTER
jitter_NGINX = [129.9, 128.7, 154, 109.9, 96.1, 124.5]
jitter_APACHE = [197.6, 39649.4, 259781.2, 101.6, 110.9, 113.7]

t_statistic_jitter, p_value_jitter = stats.ttest_ind(jitter_NGINX, jitter_APACHE, equal_var=True)
print(f"Jitter - T-Statistic: {t_statistic_jitter}, P-Value: {p_value_jitter}")

#PACKET LOSS
packet_loss_NGINX = [0, 0, 0, 0, 0, 0]
packet_loss_APACHE = [0, 22, 90, 0, 0, 0]

t_statistic_packet_loss, p_value_packet_loss = stats.ttest_ind(packet_loss_NGINX, packet_loss_APACHE, equal_var=True)
print(f"Packet Loss - T-Statistic: {t_statistic_packet_loss}, P-Value: {p_value_packet_loss}")

#CPU
cpu_NGINX = [5, 10, 16, 4, 8, 10]
cpu_APACHE = [22, 46, 50, 4, 8, 12]

t_statistic_cpu, p_value_cpu = stats.ttest_ind(cpu_NGINX, cpu_APACHE, equal_var=True)
print(f"CPU - T-Statistic: {t_statistic_cpu}, P-Value: {p_value_cpu}")

#RAM
ram_NGINX = [4, 4, 5, 27, 27, 26]
ram_APACHE = [6, 22, 50, 19, 19, 18]

t_statistic_ram, p_value_ram = stats.ttest_ind(ram_NGINX, ram_APACHE, equal_var=True)
```

```

print(f"RAM - T-Statistic: {t_statistic_ram}, P-Value: {p_value_ram}")

#Network Throughput
network_throughput_NGINX = [146.82, 293.42, 439.83, 189.81, 379.37, 568.80]
network_throughput_APACHE = [146.81, 218.27, 77.84, 189.74, 379.09, 568.69]

t_statistic_network_throughput, p_value_network_throughput = stats.ttest_ind(network_throughput_NGINX, network_throughput_APACHE,
equal_var=True)
print(f"Network Throughput - T-Statistic: {t_statistic_network_throughput}, P-Value: {p_value_network_throughput}")

alpha = 0.05

def check_significance(p_value, parameter):
    if p_value < alpha:
        print(f"{parameter} - terima H0 tidak terdapat perbedaan yang signifikan pada data statistik pengujian")
    else:
        print(f"{parameter} - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian")

check_significance(p_value_throughput, "Throughput")
check_significance(p_value_delay, "Delay")
check_significance(p_value_jitter, "Jitter")
check_significance(p_value_packet_loss, "Packet Loss")
check_significance(p_value_cpu, "CPU")
check_significance(p_value_ram, "RAM")
check_significance(p_value_network_throughput, "Network Throughput")
check_significance(p_value_ram, "RAM")
check_significance(p_value_network_throughput, "Network Throughput")

```

Script menampilkan grafik pengujian Nginx dan Apache dilingkungan docker dan virtual dengan Apache Jmeter

```

import matplotlib.pyplot as plt
from scipy import stats

# Data dua kelompok yang berbeda untuk setiap parameter
throughput_NGINX = [4.18, 8.35, 12.53, 4.18, 8.35, 12.52]
throughput_APACHE = [4.17, 7.08, 4.48, 4.18, 8.35, 12.53]

delay_NGINX = [90.94, 92.26, 94.02, 69.48, 68.76, 69.15]
delay_APACHE = [258.715, 18759, 122401.11, 75.28, 74.44, 73.97]

jitter_NGINX = [129.9, 128.7, 154, 109.9, 96.1, 124.5]
jitter_APACHE = [197.6, 39649.4, 259781.2, 101.6, 110.9, 113.7]

packet_loss_NGINX = [0, 0, 0, 0, 0, 0]
packet_loss_APACHE = [0, 22, 90, 0, 0, 0]

cpu_NGINX = [5, 10, 16, 4, 8, 10]
cpu_APACHE = [22, 46, 50, 4, 8, 12]

```

```

ram_NGINX = [4, 4, 5, 27, 27, 26]
ram_APACHE = [6, 22, 50, 19, 19, 18]

#Network Throughput
network_throughput_NGINX = [146.82, 293.42, 439.83, 189.81, 379.37, 568.80]
network_throughput_APACHE = [146.81, 218.27, 77.84, 189.74, 379.09, 568.69]

# Melakukan uji t untuk setiap parameter
t_statistic_throughput, p_value_throughput = stats.ttest_ind(throughput_NGINX, throughput_APACHE, equal_var=True)
t_statistic_delay, p_value_delay = stats.ttest_ind(delay_NGINX, delay_APACHE, equal_var=True)
t_statistic_jitter, p_value_jitter = stats.ttest_ind(jitter_NGINX, jitter_APACHE, equal_var=True)
t_statistic_packet_loss, p_value_packet_loss = stats.ttest_ind(packet_loss_NGINX, packet_loss_APACHE, equal_var=True)
t_statistic_cpu, p_value_cpu = stats.ttest_ind(cpu_NGINX, cpu_APACHE, equal_var=True)
t_statistic_ram, p_value_ram = stats.ttest_ind(ram_NGINX, ram_APACHE, equal_var=True)
t_statistic_network_throughput, p_value_network_throughput = stats.ttest_ind(network_throughput_NGINX, network_throughput_APACHE,
equal_var=True)

# Data untuk grafik
parameters = ['Throughput', 'Delay', 'Jitter', 'Packet-Loss', 'CPU', 'RAM', 'Network Throughput']
t_statistics = [t_statistic_throughput, t_statistic_delay, t_statistic_jitter, t_statistic_packet_loss, t_statistic_cpu, t_statistic_ram,
t_statistic_network_throughput]
p_values = [p_value_throughput, p_value_delay, p_value_jitter, p_value_packet_loss, p_value_cpu, p_value_ram, p_value_network_throughput]

# Membuat grafik batang
plt.figure(figsize=(10, 6))
bars = plt.bar(parameters, t_statistics, color=['#FF6500', '#7E60BF', '#E78F81', '#6256CA', '#507687', '#CB80AB', '#6EC207'])
plt.axhline(0, color='gray', linewidth=0.5) # Garis nol untuk referensi
plt.title('T-Statistics untuk Setiap Parameter')
plt.xlabel('Parameter')
plt.ylabel('T-Statistic')
plt.grid(axis='y', linestyle='--', linewidth=0.7, alpha=0.7)

# Menampilkan nilai p-value dan t-statistic di atas setiap batang
for i, (bar, t_stat, p_val) in enumerate(zip(bars, t_statistics, p_values)):
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height, f't={t_stat:.2f}', ha='center', va='bottom' if height > 0 else 'top')
    plt.text(bar.get_x() + bar.get_width()/2, height - 0.2 if height > 0 else height + 0.2, f'p={p_val:.2f}', ha='center', va='bottom' if height > 0 else
'top')

plt.tight_layout()

# Menampilkan grafik
plt.show()

```

Script Python menguji Nginx dan Apache dilingkungan docker dan virtual dengan Apache Bench

```
from scipy import stats

#THROUGHPUT
throughput_NGINX = [11.68, 11.55, 11.77, 14.47, 13.34, 12.51]
throughput_APACHE = [4.001, 3.86, 3.88, 13.44, 13.28, 13.36]

t_statistic_throughput, p_value_throughput = stats.ttest_ind(throughput_NGINX, throughput_APACHE, equal_var=True)
print(f"Throughput - T-Statistic: {t_statistic_throughput}, P-Value: {p_value_throughput}")

#network_throughput
network_throughput_NGINX = [409.91, 405.46, 413.41, 656.17, 605.06, 564.55]
network_throughput_APACHE = [140.47, 135.55, 136.31, 610.71, 602.24, 606.17]

t_statistic_network_throughput, p_value_network_throughput = stats.ttest_ind(network_throughput_NGINX, network_throughput_APACHE,
equal_var=True)
print(f"Network Throughput - T-Statistic: {t_statistic_network_throughput}, P-Value: {p_value_network_throughput}")

#JITTER
jitter_NGINX = [100.9, 99.8, 120.7, 47, 60.3, 82.4]
jitter_APACHE = [262.6, 256.2, 317.2, 56.9, 57.5, 66.9]

t_statistic_jitter, p_value_jitter = stats.ttest_ind(jitter_NGINX, jitter_APACHE, equal_var=True)
print(f"Jitter - T-Statistic: {t_statistic_jitter}, P-Value: {p_value_jitter}")

#Time Taken
time_taken_NGINX = [42.81, 86.57, 127.40, 34.56, 74.96, 120.44]
time_taken_APACHE = [125.14, 259.04, 386.42, 37.19, 75.31, 112.36]

t_statistic_time_taken, p_value_time_taken = stats.ttest_ind(time_taken_NGINX, time_taken_APACHE, equal_var=True)
print(f"Time Taken - T-Statistic: {t_statistic_time_taken}, P-Value: {p_value_time_taken}")

#CPU
cpu_NGINX = [12, 14, 14, 10, 12, 11]
cpu_APACHE = [21, 21, 21, 14, 13, 13]

t_statistic_cpu, p_value_cpu = stats.ttest_ind(cpu_NGINX, cpu_APACHE, equal_var=True)
print(f"CPU - T-Statistic: {t_statistic_cpu}, P-Value: {p_value_cpu}")

#RAM
ram_NGINX = [5, 5, 6, 26, 26, 26]
ram_APACHE = [8, 8, 9, 10, 19, 18]

t_statistic_ram, p_value_ram = stats.ttest_ind(ram_NGINX, ram_APACHE, equal_var=True)
print(f"RAM - T-Statistic: {t_statistic_ram}, P-Value: {p_value_ram}")
```

```

alpha = 0.05

def check_significance(p_value, parameter):
    if p_value < alpha:
        print(f"{parameter} - terima H0 tidak terdapat perbedaan yang signifikan pada data statistik pengujian")
    else:
        print(f"{parameter} - tolak H0 ada perbedaan yang signifikan pada data statistik pengujian")

check_significance(p_value_throughput, "Throughput")
check_significance(p_value_network_throughput, "Network Throughput")
check_significance(p_value_jitter, "Jitter")
check_significance(p_value_time_taken, "Time Taken")
check_significance(p_value_cpu, "CPU")
check_significance(p_value_ram, "RAM")

```

Script menampilkan grafik pengujian Nginx dan Apache dilingkungan docker dan virtual dengan Apache Bench

```

import matplotlib.pyplot as plt
from scipy import stats

# Data dua kelompok yang berbeda untuk setiap parameter
throughput_NGINX = [11.68, 11.55, 11.77, 14.47, 13.34, 12.51]
throughput_APACHE = [4.001, 3.86, 3.88, 13.44, 13.28, 13.36]

network_throughput_NGINX = [409.91, 405.46, 413.41, 656.17, 605.06, 564.55]
network_throughput_APACHE = [140.47, 135.55, 136.31, 610.71, 602.24, 606.17]

jitter_NGINX = [100.9, 99.8, 120.7, 47, 60.3, 82.4]
jitter_APACHE = [262.6, 256.2, 317.2, 56.9, 57.5, 66.9]

time_taken_NGINX = [42.81, 86.57, 127.40, 34.56, 74.96, 120.44]
time_taken_APACHE = [125.14, 259.04, 386.42, 37.19, 75.31, 112.36]

cpu_NGINX = [12, 14, 14, 10, 12, 11]
cpu_APACHE = [21, 21, 21, 14, 13, 13]

ram_NGINX = [5, 5, 6, 26, 26, 26]
ram_APACHE = [8, 8, 9, 10, 19, 18]

# Melakukan uji t untuk setiap parameter
t_statistic_throughput, p_value_throughput = stats.ttest_ind(throughput_NGINX, throughput_APACHE, equal_var=True)
t_statistic_network_throughput, p_value_network_throughput = stats.ttest_ind(network_throughput_NGINX, network_throughput_APACHE,
equal_var=True)
t_statistic_jitter, p_value_jitter = stats.ttest_ind(jitter_NGINX, jitter_APACHE, equal_var=True)
t_statistic_time_taken, p_value_time_taken = stats.ttest_ind(time_taken_NGINX, time_taken_APACHE, equal_var=True)

```

```

t_statistic_cpu, p_value_cpu = stats.ttest_ind(cpu_NGINX, cpu_APACHE, equal_var=True)
t_statistic_ram, p_value_ram = stats.ttest_ind(ram_NGINX, ram_APACHE, equal_var=True)

# Data untuk grafik
parameters = ['throughput', 'Network Throughput', 'Jitter', 'Time Taken', 'CPU', 'RAM']
t_statistics = [t_statistic_throughput, t_statistic_network_throughput, t_statistic_jitter, t_statistic_time_taken, t_statistic_cpu, t_statistic_ram]
p_values = [p_value_throughput, p_value_network_throughput, p_value_jitter, p_value_time_taken, p_value_cpu, p_value_ram]

# Membuat grafik batang
plt.figure(figsize=(10, 6))
bars = plt.bar(parameters, t_statistics, color=['#FF6500', '#7E60BF', '#E78F81', '#6256CA', '#507687', '#CB80AB'])
plt.axhline(0, color='gray', linewidth=0.5) # Garis nol untuk referensi
plt.title("T-Statistics untuk Setiap Parameter")
plt.xlabel("Parameter")
plt.ylabel("T-Statistic")
plt.grid(axis='y', linestyle='--', linewidth=0.7, alpha=0.7)

# Menampilkan nilai p-value dan t-statistic di atas setiap batang
for i, (bar, t_stat, p_val) in enumerate(zip(bars, t_statistics, p_values)):
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height, f't={t_stat:.2f}', ha='center', va='bottom' if height > 0 else 'top')
    plt.text(bar.get_x() + bar.get_width()/2, height - 0.2 if height > 0 else height + 0.2, f'p={p_val:.2f}', ha='center', va='bottom' if height > 0 else
'top')

plt.tight_layout()

# Menampilkan grafik
plt.show()

```