

**PENERAPAN *LOAD BALANCING* PADA WEBSITE
SMART HEALTH MENGGUNAKAN
METODE *IP HASH***

PROJEK

Sebagai salah satu syarat untuk menyelesaikan Studi di
Program Studi Teknik Komputer DIII



OLEH:

**AZZA ADLIYAH
09030582226040**

**PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
JUNI 2025**

HALAMAN PENGESAHAN

PROJEK AKHIR

PENERAPAN LOAD BALANCING PADA WEBSITE SMART HEALTH

MENGGUNAKAN METODE IP HASH

Sebagai salah satu syarat untuk penyelesaian studi di
Program Studi D3 Teknik Komputer

Oleh:

AZZA ADLIYAH
09030582226040

Pembimbing 1 : **Dr. Ir. Ahmad Heryanto, M.T.**
NIP. 198701222015041002

Pembimbing 2 : **Adi Hermansyah, M.T.**
NIP. 198904302024211001

Mengetahui
Koordinator Program Studi Teknik Komputer



Dr. Ir. Ahmad Heryanto, M.T.
198701222015041002

HALAMAN PERSETUJUAN

Telah diuji dan lulus pada:

Hari : Jum'at

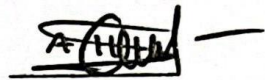
Tanggal : 13 Juni 2025

Tim Penguji :

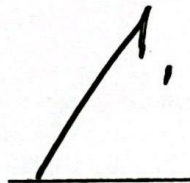
1. Ketua : Aditya P. P Prasetyo, S.Kom., M.T.



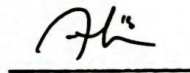
2. Pembimbing I : Dr. Ir. Ahmad Heryanto, M.T



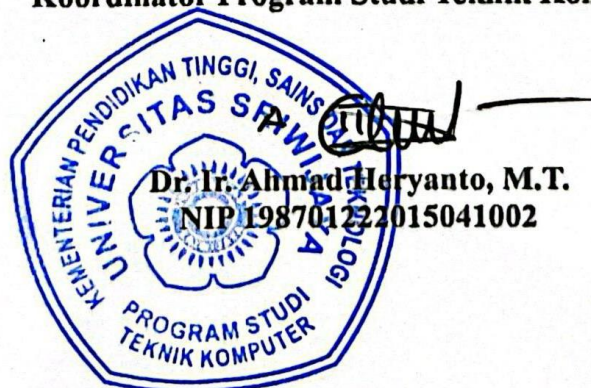
3. Pembimbing II : Adi Hermansyah, M.T.



4. Penguji : Winda Kurnia Sari, S.SI., M.Kom.



Mengetahui,
Koordinator Program Studi Teknik Komputer



HALAMAN PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Azza Adliyah
NIM : 09030582226040
Program Studi : Teknik Komputer
Judul Projek : Penerapan *Load balancing* pada Website
Smart Health Menggunakan Metode *IP Hash*

Hasil Pengecekan Software iThenticate/Turnitin : 15%

Menyatakan bahwa laporan Projek saya merupakan hasil karya sendiri dan bukan hasil penjiplakan atau plagiat. Apabila ditemukan unsur penjiplakan atau plagiat dalam laporan Projek ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya.

Demikian, pernyataan ini saya buat dalam keadaan sadar dan tanpa paksaan dari siapapun.



Palembang, 13 Juni 2025



Azza Adliyah
NIM 09030582226040

MOTTO DAN PERSEMBAHAN

MOTTO

"Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya. Dia mendapat (pahala) dari yang dikerjakannya dan mendapat (siksa) dari yang diperbuatnya"

(QS. Al-Baqarah: 286)

"Kehidupan bukan tentang seberapa cepat kita mencapai tujuan, tapi tentang seberapa bijak kita melangkah dan mencoba tumbuh dari setiap perjalanan yang kita lalui". (Azza Adliyah)

PERSEMBAHAN

Laporan ini kupersembahkan dengan tulus kepada kedua orang tua tercinta, atas doa yang tak pernah putus dan kasih sayang yang tak terbatas. Juga kepada Satryo Pangestu yang senantiasa mendukung dan menjadi penyemangat dari awal hingga akhir perkuliahan, kepada keluarga, ayuk-ayuk dan para dosen Teknik Komputer yang selalu menjadi sumber semangat untuk terus maju. Serta, kepada Almamater Kuning Kebanggaanku yang telah mengiringi langkah ini.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Proyek ini yang berjudul “Penerapan *load balancing* pada *Website Smart Health* Menggunakan Metode *IP Hash*”. Proyek ini disusun sebagai salah satu syarat untuk memenuhi syarat dalam menyelesaikan pendidikan di Program Studi Teknik Komputer (DIII) Universitas Sriwijaya.

Selama proses penyusunan Proyek ini, penulis banyak mendapatkan bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan kesehatan kepada penulis sehingga penulis dapat menyelesaikan Laporan Proyek ini dengan tepat waktu.
2. Kedapa cinta pertama dan pintu surga, Ayah dan Ama. Terima kasih atas segala kasih sayang, doa, dan dukungan moril dan materil yang tidak terhingga sehingga penulis mampu menyelesaikan studi diploma sampai selesai di Universitas Sriwijaya.
3. Kedua saudari tercinta, Khalishah dan Qonita. Terima kasih telah menjadi adik sekaligus teman cerita yang memberikan semangat, doa, dan dukungan untuk penulis dalam menyelesaikan proyek ini.
4. Bapak Prof. Dr. Erwin, S.Si., M.Si. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
5. Bapak Dr. Ir. Ahmad Heryanto, M.T. selaku Koordinator Program Studi Teknik Komputer Universitas Sriwijaya sekaligus menjadi Dosen Pembimbing I dalam pembuatan Laporan Proyek ini.
6. Bapak Adi Hermansyah, M.T. selaku Dosen Pembimbing II dalam pembuatan Laporan Proyek ini.
7. Bapak Rahmat Fadli Isnanto, S.Si., Msc selaku Dosen Pembimbing Akademik.
8. Admin jurusan teknik komputer yang telah berjasa dalam membantu permasalahan administrasi penulis.

9. Teruntuk pemilik NIM 09030582226008 saya mengucapkan terima kasih sebanyak-banyaknya karena selalu ada dan tak henti-hentinya memberikan semangat dan dukungan serta bantuan baik itu tenaga, pikiran, materi maupun moril kepada penulis untuk menyelesaikan studi ini. Laki-laki dengan nama lengkap Satryo Pangestu, terima kasih atas segala ketulusan dan kesabaranmu. Hadirmu menjadi semangat sekaligus pengingat bahwa perjalanan ini tidak pernah sendiri.
10. Teman-teman seangkatan mulai dari sekolah menengah pertama yg utama 'Niken, Cece, Zahwa, Afinah dan Adinda' hingga sekolah menengah atas terutama 'cewe-cewe istigfar' terimakasih sudah mempertahankan pertemanan yang tiada tara ini, semangat untuk kalian dimanapun berada.
11. Dan yang terakhir, kepada saudara tanpa ikatan darah Widya Anggraini, Mayang Arinda, Prita Salma, Asti Ratna, Dwi Aurelia, dan Msy Nayla Vira yang saya sebut hangat sebagai 'Ayuk-ayuk' Terima kasih atas tawa, dan semangat yang kalian berikan.

Penulis sadar bahwa masih terdapat banyak kekurangan dalam pengerjaan Projek ini. Oleh karena itu, kritik dan saran akan sangat membantu penulis untuk perbaikan laporan ini. Penulis berharap Projek ini dapat memberikan manfaat bagi pembaca. Terima kasih.

Palembang, 13 Juni 2025

Penulis

Azza Adliyah
NIM 09030582226040

PENERAPAN *LOAD BALANCING* PADA WEBSITE *SMART HEALTH* MENGUNAKAN METODE *IP HASH*

Oleh

AZZA ADLIYAH (09030582226040)

Jurusan Teknik Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya

Email : azzadlyhh@gmail.com

ABSTRAK

Penelitian ini mengembangkan "*Sistem Load balancing* pada Website *Smart Health* Menggunakan Metode *IP Hash*". Sistem *load balancing* dirancang untuk mendistribusikan beban lalu lintas *website* secara merata, sehingga pengguna layanan *Smart Health* dapat mengakses platform tanpa mengalami gangguan. Sistem dibangun menggunakan *docker container* dan *Nginx* sebagai arsitektur utama menunjukkan bahwa pengujian kualitas layanan sangat memuaskan, dengan rata-rata waktu respons menurun dari 1828,8 ms menjadi 1465,2 ms, rata-rata persentase error turun dari 17,67% menjadi 14,73%, dan throughput meningkat dari 201,86 menjadi 310,46 permintaan per detik. Selain itu, penggunaan CPU juga menurun drastis dari 62,92% menjadi 14,44% setelah diterapkan *load balancing*. Penelitian ini menyimpulkan bahwa penerapan *load balancing* menggunakan *docker container* dan *Nginx* dengan metode *IP Hash* mampu membuat sistem *website Smart Health* dapat memberikan layanan yang andal bahkan dalam kondisi trafik tinggi sekalipun.

Kata Kunci: *Load balancing, Smart Health, IP Hash, docker container, Nginx*

IMPLEMENTATION OF LOAD BALANCING ON THE SMART HEALTH WEBSITE USING THE IP HASH METHOD

By

AZZA ADLIYAH (09030582226040)

*Dept. Of Computer Enginnering, Faculty of Computer Science, Sriwijaya
University*

Email : azzadlyhh@gmail.com

ABSTRACT

This research develops a "Load Balancing System on the Smart Health Website Using the IP Hash Method." The load balancing system is designed to evenly distribute website traffic loads, allowing Smart Health users to access the platform without interruption. The system is built using Docker containers and NGINX as the core architecture. The results of the service quality tests show significant improvement: the average response time decreased from 1828.8 ms to 1465.2 ms, the average error rate dropped from 17.67% to 14.73%, and throughput increased from 201.86 to 310.46 requests per second. Additionally, CPU usage was significantly reduced from 62.92% to 14.44% after the implementation of load balancing. This study concludes that the application of load balancing using Docker containers and NGINX with the IP Hash method enables the Smart Health website system to deliver reliable services even under high traffic conditions.

Keywords: Load balancing, Smart Health, IP Hash, Docker container, NGINX

DAFTAR ISI

HALAMAN PENGESAHAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PERNYATAAN	iv
MOTTO DAN PERSEMBAHAN	v
KATA PENGANTAR	vi
ABSTRAK	8
ABSTRACT	9
DAFTAR ISI	10
DAFTAR GAMBAR	13
DAFTAR TABEL	15
DAFTAR LAMPIRAN	16
BAB I PENDAHULUAN	17
1.1 Latar Belakang	17
1.2 Rumusan Masalah	18
1.3 Batasan Masalah	18
1.4 Tujuan	19
1.5 Manfaat	19
1.6 Metodelogi Penelitian	20
BAB II DASAR TEORI	21
2.1 Penelitian Terdahulu	21
2.2 Sistem <i>Smart Health</i>	26
2.3 <i>Load balancing</i>	27
2.3.1 Konsep <i>Load balancing</i>	27
2.3.2 Faktor Adanya Load Balacing	27
2.4 Metode <i>Load balancing</i>	28
2.4.1 <i>IP Hash</i>	28
2.4.2 Round-Robin	29
2.4.3 Least Connection	29
2.4.4 <i>Rendon Allocation</i>	30
2.4.5 Least Response Time	31
2.4.6 Tabel Perbandingan	32

2.5	Metode <i>Load Balancing</i>	33
2.5.1	<i>Docker</i>	33
2.5.2	<i>Nginx</i>	34
2.6	<i>Visual Studio Code</i>	35
2.7	Portainer	35
2.8	Jmeter	35
2.9	Wireshark	36
BAB III	PERANCANGAN SISTEM	37
3.1	Kerangka Kerja Penelitian.....	37
3.2	Data dan Pengumpulan Data	38
3.3	Analisis Kebutuhan	39
3.3.1	Analisa Kebutuhan Perangkat Keras (<i>Hardware</i>)	39
3.3.2	Analisa Kebutuhan Perangkat Lunak (<i>Software</i>)	39
3.3.3	Internet.....	40
3.4	Perancangan Sistem.....	40
3.4.1	Perancangan Sistem Secara Umum.....	40
3.4.2	Design Topologi Sistem	40
3.4.3	Perancangan <i>Docker Image</i>	41
	Kode Program 3. 1 Konfigurasi <i>DockerFile</i>	44
3.4.4	Perancangan <i>Docker Container Web</i>	44
	Kode Program 3. 2 Konfigurasi <i>Container Web</i>	46
3.4.5	Perancangan <i>Docker Container Portainer</i>	48
3.4.6	Perancangan <i>Load Balancing</i> Menggunakan Metode <i>IP Hash</i>	50
3.5	Skenario Pengujian.....	55
3.5.1	Skenario Pengujian Pertama.....	56
3.5.2	Skenario Pengujian Kedua	57
3.5.3	Skenario Pengujian Ketiga	60
BAB IV	HASIL DAN PEMBAHASAN	62
4.1	Pengujian Skenario Pertama.....	62
4.2	Pengujian Skenario Kedua	66
4.3	Pengujian Skenario Ketiga	79
BAB V	PENUTUP	88

5.1	Kesimpulan.....	88
5.2	Saran.....	88
DAFTAR PUSTAKA.....		89

DAFTAR GAMBAR

Gambar 2. 1 Konsep <i>IP Hash</i> pada system jaringan.....	28
Gambar 2. 2 Konsep Round-Robin dalam sistem jaringan.....	29
Gambar 2. 3 Konsep <i>Least Connections</i> dalam sistem jaringan.....	30
Gambar 2. 4 Konsep <i>Random Allocation</i> dalam sistem jaringan.....	31
Gambar 2. 5 Konsep <i>Least Response Time</i> dalam sistem jaringan	31
Gambar 2. 6 <i>Docker</i>	33
Gambar 2. 7 <i>Nginx</i>	34
Gambar 2. 8 Proses Server Menggunakan <i>Nginx</i>	34
Gambar 2. 9 <i>Visual Studio Code</i>	35
Gambar 2. 10 <i>Portainer</i>	35
Gambar 2. 11 <i>Apache Jmeter</i>	36
Gambar 2. 12 <i>Wireshark</i>	36
Gambar 3. 1 Kerangka Alur Kerja Projek.....	37
Gambar 3. 2 Topologi <i>Load Balancing</i> Menggunakan <i>Docker dan Nginx</i>	41
Gambar 3. 3 Topologi Cara Kerja <i>Docker</i>	42
Gambar 3. 4 Topologi <i>docker compose</i>	43
Gambar 3. 5 Menjalankan <i>docker-compose</i>	47
Gambar 3. 6 Tampilan Empat <i>Node Website Smart Health</i> (a) <i>node 1</i> , (b) <i>node 2</i> , (c) <i>node 3</i> , (d) <i>node 4</i>	48
Gambar 3. 7 Tampilan <i>Portainer</i>	50
Gambar 3. 8 Cara Kerja Metode <i>IP Hash</i>	53
Gambar 3. 9 <i>Logging Access Nginx Web Smart Health</i>	54
Gambar 3. 10 Tampilan <i>Web Smart Health</i> Setelah <i>Load Balancing</i>	55
Gambar 3. 11 <i>Flowchart</i> Skenario Pengujian Pertama	57
Gambar 3. 12 <i>Flowchart</i> Skenario Pengujian Kedua	59
Gambar 3. 13 <i>Flowchart</i> Pengujian Skenario Ketiga	61
Gambar 4. 1 Simulasi Pengujian Skenario Kedua	68
Gambar 4. 2 Hasil Pengukuran 100 traffic dengan <i>load balancing</i>	68
Gambar 4. 3 Hasil Pengukuran 100 traffic dengan tanpa <i>load balancing</i>	69
Gambar 4. 4 Hasil Pengukuran 1000 traffic dengan <i>load balancing</i>	70
Gambar 4. 5 Hasil Pengukuran 1000 traffic dengan tanpa <i>load balancing</i>	71

Gambar 4. 6 Hasil Pengukuran 2000 traffic dengan <i>load balancing</i>	72
Gambar 4. 7 Hasil Pengukuran 2000 traffic dengan tanpa <i>load balancing</i>	73
Gambar 4. 8 Hasil Pengukuran 4000 traffic dengan <i>load balancing</i>	75
Gambar 4. 9 Hasil Pengukuran 4000 traffic dengan tanpa <i>load balancing</i>	76
Gambar 4. 10 Hasil Pengukuran 8000 traffic dengan <i>load balancing</i>	77
Gambar 4. 11 Hasil Pengukuran 8000 traffic dengan tanpa <i>load balancing</i>	78
Gambar 4. 12 Grafik perbandingan <i>CPU Utilazation</i> antara <i>single server</i> dan <i>load balancing</i> dengan beban 500 <i>request</i>	80
Gambar 4. 13 Grafik Perbandingan <i>CPU Utilazation</i> antara <i>single server</i> dan <i>load balancing</i> dengan beban 1000 <i>request</i>	82
Gambar 4. 14 Grafik Perbandingan <i>CPU Utilazation</i> antara <i>single server</i> dan <i>load balancing</i> dengan beban 2000 <i>request</i>	83
Gambar 4. 15 Grafik Perbandingan <i>CPU Utilazation</i> antara <i>single server</i> dan <i>load balancing</i> dengan beban 4000 <i>request</i>	85
Gambar 4. 16 Grafik Perbandingan <i>CPU Utilazation</i> antara <i>single server</i> dan <i>load balancing</i> dengan beban 5000 <i>request</i>	86

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	21
Tabel 2. 2 Tabel Perbandingan Metode <i>Load Balancing</i>	32
Tabel 4. 1 Hasil Pengujian <i>stress test</i> 100 traffic	62
Tabel 4. 2 Hasil Pengujian <i>stress test</i> 500 traffic	63
Tabel 4. 3 Hasil Pengujian <i>stress test</i> 1000 traffic	63
Tabel 4. 4 Hasil Pengujian <i>stress test</i> 5000 traffic	64
Tabel 4. 5 Hasil Pengujian <i>stress test</i> 10000 traffic	64
Tabel 4. 6 Hasil Pengujian Keseluruhan Skenario Pertama.....	65
Tabel 4. 7 Kategori <i>Throughput</i>	66
Tabel 4. 8 Kategori <i>Packet Loss</i>	67
Tabel 4. 9 Kategori <i>Delay</i>	67
Tabel 4. 10 Kategori <i>Jitter</i>	67
Tabel 4. 11 Hasil Perbandingan Pengujian <i>Quality of Service</i> 100 Traffic.....	70
Tabel 4. 12 Hasil Perbandingan Pengujian <i>Quality of Service</i> 1000 Traffic	72
Tabel 4. 13 Hasil Perbandingan Pengujian <i>Quality of Service</i> 2000 Traffic	74
Tabel 4. 14 Hasil Perbandingan Pengujian <i>Quality of Service</i> 4000 Traffic	76
Tabel 4. 15 Hasil Perbandingan Pengujian <i>Quality of Service</i> 8000 Traffic	79
Tabel 4. 16 Hasil Pengujian Keseluruhan Skenario Kedua	79
Tabel 4. 17 Hasil Pengujian <i>CPU Utilazation</i> 500 request	80
Tabel 4. 18 Hasil Pengujian <i>CPU Utilazation</i> 1000 request	81
Tabel 4. 19 Hasil Pengujian <i>CPU Utilazation</i> 2000 request	83
Tabel 4. 20 Hasil Pengujian <i>CPU Utilazation</i> 4000 request	84
Tabel 4. 21 Hasil Pengujian <i>CPU Utilazation</i> 5000 request	86
Tabel 4. 22 Rata-rata keseluruhan pengujian	87

DAFTAR LAMPIRAN

Lampiran 1 Kode Program	93
Lampiran 2 Kartu Konsultasi Pembimbing I.....	97
Lampiran 3 Kartu Konsultasi Pembimbing II	98
Lampiran 4 Surat Rekomendasi Ujian Projek Pembimbing I	99
Lampiran 5 Surat Rekomendasi Ujian Projek Pembimbing II	100
Lampiran 6 Verifikasi Hasil SULIET / USEPT	101
Lampiran 7 Surat Keterangan Projek	102
Lampiran 8 Turnitin.....	103
Lampiran 9 Form Revisi Penguji.....	104
Lampiran 10 Form Revisi Pembimbing 1	105
Lampiran 11 Form Revisi Pembimbing II.....	106

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini penggunaan internet sangat penting dan banyak orang yang menggunakan untuk mencari informasi [1]. Pada era digital saat ini, penggunaan internet semakin meningkat dan mengakibatkan lonjakan jumlah penggunaan *website* yang dapat menyebabkan *server* menjadi *overload*. Akibatnya *server* menurun dan sering terjadi gangguan pada layanan – layanan *web* tersebut [2]. Bertambahnya pemakaian internet menyebabkan permintaan akan layanan internet harus terus ditingkatkan [3]. Gangguan tersebut dapat menurunkan kualitas pengalaman pengguna terhadap *website* tersebut.

Smart Health adalah aplikasi berbasis *web* untuk memonitoring pasien disektor kesehatan [4]. Website ini dirancang untuk memberikan kualitas pelayanan bidang kesehatan dalam memonitoring kesehatan. Dalam meningkatkan sistem layanan *website Smart Health*, dibutuhkan suatu sistem *server* yang dapat mengatasi tingginya jumlah permintaan sehingga *website* tetap stabil dan dapat diakses baik oleh pengguna secara bersamaan [5]. Kondisi ini berkaitan dengan aspek skalabilitas sitem, dimana model pelayanan menggunakan beberapa *server* (jamak) menjadi salah satu solusi yang efektif [6]. Dalam melakukan penerapan *server* jamak, dibutuhkan salah satu solusi yaitu *docker* karena kemudahan dalam proses *deployment* (penyebaran) [7]. *docker* memungkinkan untuk membuat, menjalankan, melakukan percobaan dan meluncurkan aplikasi di dalam sebuah *container* [8].

docker sangat ringan dan mempunyai mekanisme yang lebih maju jika dibandingkan dengan perangkat lunak virtualisasi berbasis *hypervisor*. Sehingga penerapan *docker* pada *server* jamak adalah solusi paling efektif. Dalam melakukan penerapan *load balancing* pada *server* jamak dibutuhkan lah sebuah tools yang dapat membagi beban *server* secara merata yaitu *Nginx* [9].

Berbagai metode *load balancing* telah dikembangkan, salah satunya adalah *IP Hash*, yang menggunakan algoritma hashing berdasarkan alamat IP pengguna untuk

menentukan *server* yang akan melayani permintaan [10]. Metode ini memastikan distribusi beban kerja yang konsisten di seluruh *server* dan memiliki keunggulan dalam menjaga kestabilan pengelolaan sesi, karena pengguna dengan alamat IP yang sama akan selalu diarahkan ke *server* yang sama selama *server* tersebut tersedia.

Berdasarkan permasalahan yang telah diuraikan diatas, maka dibutuhkanlah suatu sistem yang dapat membagi beban kerja merata pada *website Smart Health*. Agar pasien pengguna *website* dapat menggunakan layanan tanpa ada masalah. Oleh karena itu diangkatlah judul projek ini yaitu “Penerapan *Load balancing* Pada *Website Smart Health* Menggunakan Metode *IP Hash*”. Dengan menggunakan *docker Container* dapat memungkinkan penggunaan *server* jamak meski resource terbatas. Dan dengan menggabungkan *docker* dan *Nginx* menggunakan metode *IP Hash* diharapkan sistem ini mampu menjaga kestabilan, meningkatkan kinerja, dan memastikan pengguna *website Smart Health* dapat mengakses tanpa mengalami gangguan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan diatas, maka penulis akan merumuskan masalah sebagai berikut :

1. Bagaimana merancang agar pengalaman pengguna (*User Interface / UX*) *Smart Health V2* optimal dan bebas hambatan saat mengakses *web*?
2. Bagaimana agar *website Smart Health* dapat berjalan di atas *container*?
3. Bagaimana penggunaan *load balancing* dengan metode *IP Hash* dapat mencegah gangguan jika salah satu *container* mengalami down?

1.3 Batasan Masalah

Dalam penelitian ini diperlukan Batasan masalah agar proses penelitian berjalan sesuai dengan tujuan pembuatan dan membatasi masalah yang akan dibahas, Adapun Batasan masalah tersebut sebagai berikut :

1. Pengujian dilakukan dengan membandingkan sebelum maupun sesudah diterapkannya *load balancing* berdasarkan tiga skenario pengujian.
2. Penerapan *container* menggunakan platform *docker* dalam sistem pengelolaan layanan web *Smart Health*.

3. Analisis implementasi *load balancing* dengan metode *IP Hash*, mencakup pengaturan dan performa sistem melalui parameter seperti *respon time*, *throughput*, *persentase error*, *delay*, *packet loss*, *jitter*, dan penggunaan *CPU*.

1.4 Tujuan

Tujuan dalam Proyek ini sebagai berikut :

1. Mengoptimalkan pengalaman pengguna (User Experience/UX) pada *website Smart Health* melalui pengujian usability dan identifikasi hambatan.
2. Menyelidiki penerapan teknologi *container* pada *website Smart Health*, termasuk konfigurasi lingkungan dan integrasi dengan sistem yang ada.
3. Menganalisis dan mengimplementasikan metode *load balancing IP Hash* untuk memastikan ketersediaan layanan secara konsisten, meskipun terjadi gangguan pada salah satu *container*, serta menjaga kualitas layanan yang responsif dan andal.

1.5 Manfaat

Manfaat dalam penelitian ini sebagai berikut :

1. Memberikan pengalaman pengguna yang lebih baik dengan akses layanan *Smart Health* yang cepat dan tanpa hambatan.
2. Meningkatkan efisiensi operasional melalui penerapan teknologi *container* yang andal dan terintegrasi.
3. Menjamin ketersediaan layanan *website* dengan stabilitas sistem yang terjaga melalui metode *load balancing IP Hash*.

1.6 Metodologi Penelitian

Metode yang digunakan penulis dalam penelitian ini sebagai berikut :

1. Studi Literatur

Pada tahap ini, penulis mencari dan mengkaji referensi yang relevan, seperti jurnal, buku, dan artikel terkait teknologi *container*, metode *load balancing*, khususnya *IP Hash*, serta implementasi dan pengujian sistem berbasis *web*. Studi ini bertujuan untuk memperoleh dasar teori yang mendukung perancangan dan analisis sistem.

2. Metode Konsultasi

Penulis berkonsultasi dengan dosen pembimbing dan pakar di bidang sistem berbasis *web* untuk mendapatkan masukan dan arahan terkait pengembangan dan implementasi metode yang digunakan.

3. Metode Perancangan Sistem

Tahap ini mencakup perancangan sistem, baik dari sisi infrastruktur (teknologi *container*) maupun perangkat lunak. Desain ini mencakup pengaturan *load balancing* dengan metode *IP Hash*, konfigurasi *server*, serta simulasi distribusi beban.

4. Metode Pengujian

Setelah perancangan selesai, sistem diuji untuk memastikan stabilitas, kecepatan akses, dan ketersediaan layanan. Pengujian dilakukan dengan mensimulasikan berbagai skenario beban untuk menilai kinerja *load balancing* dan pengalaman pengguna.

5. Metode Analisis dan Kesimpulan

Hasil pengujian dianalisis untuk mengevaluasi kinerja sistem dan metode *load balancing*. Analisis ini digunakan untuk mengidentifikasi kekurangan serta memberikan rekomendasi pengembangan di masa depan. Selanjutnya, penulis menyusun kesimpulan berdasarkan hasil penelitian.

DAFTAR PUSTAKA

- [1] M. Ilham, "Implementasi Sistem Load Balancing untuk Optimasi Kinerja pada Web Server Nginx Menggunakan Algoritma Ip Hash," vol. 9, pp. 60–68, 2024.
- [2] N. I. A. Ramadhan, T. A. Cahyanto, and M. Dasuki, "Analisis Algoritma Roundrobin Dan Source Ip Hash Untuk Optimasi Kinerja Load Balancing Webserver," pp. 1–6, 2020.
- [3] M. Tanujaya and C. Dewi, "Simulasi Implementasi Load Balancing pada Jaringan Internet dengan Metode Per Connection Classifier (PCC) dan Server dengan Metode IP Hash Menggunakan GNS3," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 1, pp. 238–246, 2024, doi: 10.35870/jtik.v8i1.1550.
- [4] I. Habibi, "SISTEM PEREKAM DETAK JANTUNG DAN SATURASI OKSIGEN BERBASIS SMART HEALTH," *repository.unsri.ac.id*, vol. 15, no. 1, pp. 37–48, 2024, [Online]. Available: <http://repository.unsri.ac.id/147010/>
- [5] M. D. Adila and T. Y. Hadiwandura, "Peningkatan Kinerja dan Skalabilitas Website E-Commerce Menggunakan Load Balancing," vol. 10, no. 2, pp. 428–442, 2024.
- [6] S. D. Riskiono and D. Pasha, "Analisis Metode Load Balancing Dalam Meningkatkan Kinerja Website E-Learning," *J. Teknoinfo*, vol. 14, no. 1, p. 22, 2020, doi: 10.33365/jti.v14i1.466.
- [7] F. R. Bik and Asmunin, "IMPLEMENTASI DOCKER UNTUK PENGELOLAAN BANYAK APLIKASI WEB (Studi Kasus : Jurusan Teknik Informatika UNESA)," *J. Manaj. Inform.*, vol. 7, no. 2, pp. 46–50, 2017.
- [8] S. Dwiyatno, E. Rachmat, A. P. Sari, and O. Gustiawan, "Implementasi Virtualisasi Server Berbasis Docker Container," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 7, no. 2, pp. 165–175, 2020, doi: 10.30656/prosisko.v7i2.2520.
- [9] V. K. Demidov, O. I. Pyvovar, and Y. G. Kravchenko, "Modern development methodologies for GIS oriented web application," *17th Int.*

- Conf. Geoinformatics - Theor. Appl. Asp.*, vol. 6, no. 1, 2018, doi: 10.3997/2214-4609.201801770.
- [10] M. Gokul and N. Rajkumar, "Source IP Hashing Algorithm for Load Balancing in Cloud Server," *6th Int. Conf. Mob. Comput. Sustain. Informatics, ICMCSI 2025 - Proc.*, pp. 794–801, 2025, doi: 10.1109/ICMCSI64620.2025.10883068.
- [11] C. Ma and Y. Chi, "Evaluation Test and Improvement of Load Balancing Algorithms of Nginx," *IEEE Access*, vol. 10, pp. 14311–14324, 2022, doi: 10.1109/ACCESS.2022.3146422.
- [12] S. Fugkeaw, S. Rattagool, P. Jiangthiranan, and P. Pholwiset, "FPRESSO: Fast and Privacy-Preserving SSO Authentication with Dynamic Load Balancing for Multi-Cloud-based Web Applications," *IEEE Access*, vol. 12, no. November, pp. 157888–157900, 2024, doi: 10.1109/ACCESS.2024.3485996.
- [13] Irlon, "PERBANDINGAN DAN IMPLEMENTASI LOAD BALANCER PADA WEB SERVER MENGGUNAKAN NGINX PADA GG SUSPENSION," *Ismetek*, vol. Vol. 18 No, no. 02, p. 35, 2024, [Online]. Available: <https://ismetek.itbu.ac.id/index.php/jurnal/issue/view/17/21>
- [14] A. Algiffary and T. Sutabri, "Indonesian Journal of Computer Science," *Indones. J. Comput. Sci.*, vol. 12, no. 2, pp. 284–301, 2023, [Online]. Available: <http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135>
- [15] F. Apriliansyah, I. Fitri, A. Iskandar, and R. Artikel, "Jurnal Teknologi dan Manajemen Informatika Implementasi Load Balancing Pada Web Server Menggunakan Nginx," *Int. J. Futur. Gener. Commun. Netw.*, vol. 13, no. 2, pp. 1449–1452, 2020, [Online]. Available: <http://http://jurnal.unmer.ac.id/index.php/jtmi>
- [16] E. Pentanugraha *et al.*, "Analisis Kinerja Load Balancing Webserver Menggunakan Haproxy Terintegrasi Dengan Grafana Sebagai Monitoring Dan Notifikasi Telegram," *JOINTECOMS (Journal Inf. Technol. Comput. Sci. p-ISSN)*, vol. 4, no. 1, pp. 2798–284, 2024.
- [17] N. Badisa *et al.*, "E cient Docker Image Optimization using Multi- Stage

Builds and Nginx for Enhanced Application Deployment Efficient Docker Image Optimization using Multi-Stage Builds and Nginx for Enhanced Application Deployment”.

- [18] R. T. Darmawan, “Implementasi Grafana untuk Auto-Scaling pada NGINX Web-Server di Lingkungan Docker Container,” *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 7, pp. 3163–3167, 2023.
- [19] K. Mishra and S. K. Majhi, “A binary Bird Swarm Optimization based load balancing algorithm for cloud computing environment,” *Open Comput. Sci.*, vol. 11, no. 1, pp. 146–160, 2021, doi: 10.1515/comp-2020-0215.
- [20] F. S. Mukti and F. E. Nugroho, “Upaya Pencapaian Status High Availability Server Menggunakan Metode Load Balancing Berbasis Klaster pada Database Server PT. XYZ,” *J. Tek. Inform. dan Sist. Inf.*, vol. 10, no. 1, pp. 757–767, 2023, [Online]. Available: <http://jurnal.mdp.ac.id>
- [21] M. R. Baihaqi, R. M. Negara, and R. Tulloh, “Analysis of Load Balancing Performance using Round Robin and IP Hash Algorithm on P4,” *2022 5th Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2022*, pp. 93–98, 2022, doi: 10.1109/ISRITI56927.2022.10052975.
- [22] X. Wu and D. C. O’Brien, “QoS-Driven Load Balancing in Hybrid LiFi and WiFi Networks,” *IEEE Trans. Wirel. Commun.*, vol. 21, no. 4, pp. 2136–2146, Apr. 2022, doi: 10.1109/TWC.2021.3109716.
- [23] S. Mohanty, S. Patra, S. Sarkar, P. Dube, and P. K. Pattnaik, “Load Balancing in Cloud Environment to Minimize Average Response Time,” *ESIC 2024 - 4th Int. Conf. Emerg. Syst. Intell. Comput. Proc.*, pp. 187–192, 2024, doi: 10.1109/ESIC60604.2024.10481627.
- [24] H. Babbar, S. Rani, A. K. Bashir, and R. Nawaz, “LBSMT: Load Balancing Switch Migration Algorithm for Cooperative Communication Intelligent Transportation Systems,” *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 3, pp. 1386–1395, Sep. 2022, doi: 10.1109/TGCN.2022.3162237.
- [25] G. P. Mattia, A. Pietrabissa, and R. Beraldi, “A Load Balancing Algorithm for Equalising Latency Across Fog or Edge Computing Nodes,” *IEEE Trans. Serv. Comput.*, vol. 16, no. 5, pp. 3129–3140, Sep. 2023, doi: 10.1109/TSC.2023.3265883.

- [26] A. Hermansyah *et al.*, “Heart Rate and Oxygen Saturation Internet of Things System (HROS-IoT) Uses Fuzzy Logic,” *Indones. J. Comput. Sci.*, vol. 13, no. 5, Oct. 2024, doi: 10.33022/IJCS.V13I5.4330.
- [27] I. Rijayana, “Teknologi Load Balancing Untuk Mengatasi Beban Server,” *Semin. Nas. Apl. Teknol. Inf.*, vol. 2005, no. Snati, pp. 79–75, 2005.
- [28] P. Satya Saputra, P. Aditya Pratama, and L. Putu Ary Sri Tjahyanti, “Perancangan Dan Komparasi Web Server Nginx Dengan Web Server Apache Serta Pemanfaatan Reverse Proxy Server Pada Nginx,” *J. Komput. dan Teknol. Sains*, vol. 2, no. 1, pp. 16–21, 2023.
- [29] B. Kurniawan and M. Romzi, “Implementasi Pemrograman Python Menggunakan Visual Studio Code,” *J. Pengabd. Masy.*, vol. 11, no. 2, pp. 1–9, 2020, doi: 10.31004/abdira.v2i3.202.
- [30] “Kubernetes and Docker Container Management Software.” Accessed: May 16, 2025. [Online]. Available: <https://www.portainer.io/>
- [31] N. Luh, A. S. Ginasari, K. S. Wibawa, N. Kadek, and A. Wirdiani, “Pengujian Stress Testing API Sistem Pelayanan dengan Apache JMeter,” 2021.
- [32] Ratnasari Ratnasari, Fernando Juliansyah, Gera Nugraha, Dede Wirta, and Didik Aribowo, “Quality of Service (QoS) Pada Jaringan Wi-Fi (Wireless Fidelity) FKIP UNTIRTA Menggunakan Software Wireshark,” *J. Elektron. dan Tek. Inform. Ter. JENTIK*), vol. 2, no. 2, pp. 70–76, 2024, doi: 10.59061/jentik.v2i2.672.