

**PENERAPAN *LOAD BALANCER* DI *AWS CLOUD* UNTUK
MENDUKUNG *HIGH AVAILABILITY* PADA APLIKASI *WEB***

TUGAS AKHIR

Disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer



Putra Budiansyah

1232921001

**TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS BAKRIE
JAKARTA
2026**

HALAMAN PERNYATAAN ORISINALITAS

**Tugas akhir ini adalah karya saya sendiri,
dan semua sumber baik yang dikutip maupun ditujuk
telah saya nyatakan dengan benar.**

Nama : Putra Budiansyah

NIM : 1232921001

Tanda Tangan :

A handwritten signature in black ink, consisting of a stylized 'P' followed by a horizontal line and a small flourish.

Tanggal : 23 Juni 2026

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

Nama : Putra Budiansyah
NIM : 1232921001
Program Studi : Teknik Informatika
Fakultas : Fakultas Teknik dan Ilmu Komputer
Judul Skripsi : Penerapan *Load Balancer* di *AWS Cloud* untuk
Mendukung *High Availability* pada Aplikasi *Web*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Bakrie.

DEWAN PENGUJI

Pembimbing : Yusuf Lestanto, S.T., M.Sc., MIEEE. (..16 Juli 2026)

Penguji 1 : Guson Prasamuarso Kuntarto, S.T., M.Sc. (..4 Juli 2026 22:55 WIB)

Penguji 2 : Dimas Aryo Anggoro, S.Kom., M.Sc. (..13 Juli 2026, 09:25 WIB)

Ditetapkan di Jakarta

Tanggal 23 Juni 2026

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

Sebagai civitas akademik Universitas Bakrie, saya yang bertanda tangan di bawah ini:

Nama : Putra Budiansyah
NIM : 1232921001
Program Studi : Teknik Informatika
Fakultas : Fakultas Teknik dan Ilmu Komputer

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Bakrie **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-free Right)** atas karya ilmiah saya yang berjudul:

**PENERAPAN *LOAD BALANCER* DI *AWS CLOUD* UNTUK
MENDUKUNG *HIGH AVAILABILITY* PADA APLIKASI *WEB***

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Bakrie berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta untuk kepentingan akademis. Demikian pernyataan ini saya buat sebenarnya.

Dibuat di : Jakarta
Pada tanggal : 23 Juni 2026

Yang membuat pernyataan,



Putra Budiansyah
NIM. 1232921001

KATA PENGANTAR

Puji dan Syukur kehadiran Tuhan Yang Maha Esa atas segala Rahmat dan kasih karunia-Nya sehingga penulis dapat menyelesaikan buku tugas akhir ini dengan judul “Penerapan *Load Balancer* di *AWS Cloud* untuk Mendukung *High Availability* di Aplikasi *Web*” Tugas akhir ini disusun sebagai persyaratan untuk mendapat gelar sarjana pada program studi Teknik Informatika, Universitas Bakrie.

Dalam pembuatan tugas akhir ini tidak terlepas dari banyak dukungan maupun bantuan dari banyak pihak yang turut berpartisipasi dalam memberikan masukan dan saran kepada penulis.

Pembuatan buku tugas akhir ini masih banyak kekurangan. Oleh sebab itu, sarana dan kritik yang membangun dari para pembaca untuk penyempurnaan buku proyek akhir ini sangat diharapkan, semua pihak dapat memberikan masukan. Dan penulis berharap buku tugas akhir ini dapat bermanfaat untuk para pembaca.

Jakarta, 23 Juni 2026

Penulis

UNGKAPAN TERIMA KASIH

Puji dan syukur kehadiran Allah SWT karena atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik. Tugas Akhir dengan judul “**Penerapan *Load Balancer* di *Aws Cloud* untuk Mendukung *High Availability* Pada Aplikasi *Web***” ini ditulis untuk memenuhi salah satu syarat dalam menyelesaikan perkuliahan pendidikan strata satu (S1) pada Program Studi Informatika, Universitas Bakrie. Banyak pihak yang telah membantu penulis dalam penelitian dan penulisan Tugas Akhir ini, baik itu berupa bimbingan, saran, maupun dukungan secara moril dan materil. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih dan penghargaan yang setulus-tulusnya kepada:

1. Prof. Ir. Sofia W. Alisjahbana, M.Sc., Ph.D., IPU., ASEAN Eng. selaku Rektor Universitas Bakrie;
2. Dr. Mohammad Ihsan, S.T., M.T., M.Sc. selaku Dekan Fakultas Teknik dan Ilmu Komputer Universitas Bakrie;
3. Ir. Irwan Adhicandra, S.T., M.Sc., Ph.D., SMIEEE. selaku Ketua Program Studi Teknik Informatika Universitas Bakrie;
4. Bapak Yusuf Lestanto, S.T., M.Sc, selaku dosen pembimbing akademik yang telah meluangkan waktunya untuk selalu memberikan arahan selama penulis menyelesaikan studi;
5. Seluruh Bapak/Ibu dosen Program Studi Informatika Universitas Bakrie, yang telah memberikan banyak ilmu, pengetahuan, wawasan kepada penulis selama perkuliahan;
6. Kedua orang tua yang selalu memberikan semangat, motivasi dan doa yang sangat berarti bagi penulis;
7. Seluruh pihak yang terlibat dalam penyusunan Tugas Akhir yang tidak dapat penulis sebutkan satu persatu.

Dengan segala keterbatasan yang ada, penulis menyadari bahwa penyusunan Tugas Akhir ini masih jauh dari kesempurnaan. Untuk itu, saran dan

kritik akan selalu diterima agar penulis dapat memperbaiki setiap kekurangan untuk kesempurnaan di masa mendatang. Akhirnya, sekali lagi penulis menyampaikan ucapan terima kasih dan semoga Allah SWT membalas segala kebaikan serta melimpahkan berkat dan rahmat-Nya kepada semua pihak yang telah membantu selama ini. Penulis berharap Tugas Akhir ini berguna dan bermanfaat bagi kita semua.

Jakarta, 23 Juni 2026

Penulis

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	iv
KATA PENGANTAR.....	v
UNGKAPAN TERIMA KASIH	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xii
ABSTRAK	xiii
<i>ABSTRACT</i>	xiv
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	5
BAB II	6
TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terkait	6
2.2 <i>Cloud Computing</i> dan AWS	7
2.2.1 Tujuan menggunakan <i>Cloud Computing</i>	7
2.2.2 Penggunaan AWS.....	9
2.3 Implementasi <i>Load Balancer</i> pada AWS.....	13
2.3.1 Konsep <i>Load Balancing</i>	13
2.3.2 Jenis-jenis <i>Load Balancer</i> di AWS.....	14
2.3.3 Cara Kerja <i>Load Balancer</i>	18
2.3.4 AWS <i>Elastic Load Balancer</i> (ELB)	18

2.4 <i>High Availability</i> pada Aplikasi <i>Web</i>	19
2.4.1 Konsep <i>High Availability</i>	19
2.4.2 Hubungan <i>Load Balancer</i> dengan <i>High Availability</i>	19
BAB III	20
METODOLOGI PENELITIAN	20
3.1 Kerangka Penelitian	20
3.2 Alat Penelitian	22
3.3 Pengembangan Sistem	23
3.4 Objek Penelitian	25
3.5 Jenis Penelitian	25
3.6 Metode Pengujian	25
BAB IV	27
HASIL DAN PEMBAHASAN	27
4.1 Hasil	27
4.1.1 Implementasi <i>AWS Load Balancer</i> untuk Aplikasi <i>Web</i> ...	29
4.1.2 Penggunaan <i>Load Balancer</i> untuk Menjaga <i>High Availability</i> pada Aplikasi <i>Web</i>	38
4.1.3 Pengujian <i>Failover</i> (Simulasi Kegagalan <i>Server</i>)	58
4.2 Pembahasan	61
4.2.1 Implementasi <i>AWS Load Balancer</i> untuk Aplikasi <i>Web</i> ...	61
4.2.2 Penggunaan <i>Load Balancer</i> untuk Menjaga Performa <i>High</i> <i>Availability</i> pada Aplikasi <i>Web</i>	61
4.2.3 Pengujian <i>Failover</i> (Simulasi Kegagalan <i>Server</i>)	62
BAB V	64
KESIMPULAN DAN SARAN	64
5.1 Kesimpulan	64
5.2 Saran	64
DAFTAR PUSTAKA	66
LAMPIRAN	69

DAFTAR GAMBAR

Gambar 2. 1 <i>Load Balancing</i> [18]	14
Gambar 3. 1 Diagram Alur Penelitian.....	20
Gambar 4. 1 <i>Output</i> Proses <i>Deployment</i> Infrastruktur Menggunakan <i>Terraform</i> pada AWS.....	28
Gambar 4. 2 Tampilan Konfigurasi VPC di AWS.....	30
Gambar 4. 3 Konfigurasi <i>subnet ha-lab-public-subnet-1</i>	31
Gambar 4. 4 Konfigurasi <i>ha-lab-public-subnet-2</i>	32
Gambar 4. 5 Konfigurasi <i>web_server_1</i>	33
Gambar 4. 6 Tampilan halaman <i>Web</i> pada <i>web_server_1</i>	34
Gambar 4. 7 Tampilan hasil konfigurasi <i>Load Balancer</i>	35
Gambar 4. 8 <i>Resource Map</i> pada <i>Load Balancer</i>	36
Gambar 4. 9 Tampilan halaman <i>web</i> melalui <i>Load Balancer</i>	37
Gambar 4. 10 Diagram Arsitektur Sistem.....	38
Gambar 4. 11 Hasil <i>Request</i> HTTP ke <i>Application Load Balancer</i>	39
Gambar 4. 12 Konfigurasi <i>Load Testing</i> dengan 10 <i>Virtual User</i>	41
Gambar 4. 13 Hasil Pengujian <i>Load Testing</i> dengan 10 <i>Virtual User</i>	42
Gambar 4. 14 Konfigurasi <i>Load Testing</i> dengan 50 <i>Virtual User</i>	43
Gambar 4. 15 Hasil Pengujian <i>Load Testing</i> dengan 50 <i>Virtual User</i>	44
Gambar 4. 16 <i>Load test</i> pertama pada <i>Load Balancer</i>	46
Gambar 4. 17 <i>Load test</i> kedua pada <i>Load Balancer</i>	47
Gambar 4. 18 <i>Load test</i> ketiga pada <i>Load Balancer</i>	48
Gambar 4. 19 <i>Load test</i> pertama <i>direct</i> langsung ke <i>server</i>	50
Gambar 4. 20 <i>Load test</i> kedua <i>direct</i> langsung ke <i>server</i>	51
Gambar 4. 21 <i>Load test</i> ketiga <i>direct</i> langsung ke <i>server</i>	52
Gambar 4. 22 Konfigurasi <i>Load Testing</i> dengan 100 <i>Virtual User</i>	55
Gambar 4. 23 Hasil Pengujian <i>Load Testing</i> dengan 100 <i>Virtual User</i>	56
Gambar 4. 24 Kondisi Awal – 2 <i>Target Healthy</i>	58
Gambar 4. 25 Kondisi <i>Target Instance</i> setelah Salah Satu <i>Instance</i> Dihentikan....	59
Gambar 4. 26 Tampilan <i>Web</i> melalui <i>Load Balancer</i> setelah Terjadi <i>Failover</i>	60

DAFTAR TABEL

Tabel 2. 1 Rincian Biaya <i>Cloud</i> vs <i>On-premise</i> [11].....	8
Tabel 2. 2 Perbandingan Platform Layanan Komputasi Awan [16]	12
Tabel 4. 1 Hasil System Integration Testing	39
Tabel 4. 2 Konfigurasi Pengujian Load Testing	45
Tabel 4. 3 Hasil Pengujian Melalui ALB 3 Iterasi.....	49
Tabel 4. 4 Hasil Pengujian Direct IP 3 Iterasi.....	53
Tabel 4. 5 Perbandingan Tanpa Load Balancer vs Dengan Load Balancer.....	54
Tabel 4. 6 Hasil Load Testing (Dengan Load Balancer)	57
Tabel 4. 7 Rangkuman Hasil Penelitian.....	63

DAFTAR LAMPIRAN

Lampiran 1. 1: Script Konfigurasi VPC via Infrastucture as A Code.....	71
Lampiran 1. 2: Script Konfigurasi Web Server	73
Lampiran 1. 3: Script Konfigurasi Application Load Balancer	75
Lampiran 1. 4: Script Konfigurasi Load Testing	77
Lampiran 2. 1 Load Test Pertama di Load Balancer	79
Lampiran 2. 2 Load Test Kedua di Load Balancer	79
Lampiran 2. 3 Load Test Ketiga di Load Balancer	80
Lampiran 2. 4 Load Test Pertama Direct Langsung ke Server	80
Lampiran 2. 5 Load Test Kedua Direct Langsung ke Server.....	81
Lampiran 2. 6 Load Test Ketiga Direct Langsung ke Server	81

ABSTRAK

Seiring dengan meningkatnya kebutuhan akan aplikasi web yang andal, kemampuan sistem untuk tetap tersedia meskipun terjadi gangguan menjadi sangat penting. Penelitian ini mengeksplorasi penerapan *load balancer* dalam infrastruktur *AWS Cloud* sebagai solusi untuk menjaga ketersediaan layanan aplikasi web. Penelitian ini bertujuan untuk memastikan bahwa aplikasi web dapat tetap berjalan secara konsisten meskipun terjadi gangguan pada salah satu *server*. Dengan menggunakan layanan *load balancer* pada platform *AWS Cloud*, penelitian ini menguji bagaimana lalu lintas dapat dialihkan ke *server* lain yang aktif untuk menjaga stabilitas layanan. Simulasi dilakukan dengan cara mematikan salah satu *server* secara terkontrol untuk melihat bagaimana sistem dapat mempertahankan performa serta aksesibilitas aplikasi. Hal ini dilakukan mengingat semakin meningkatnya tuntutan terhadap aplikasi web yang dapat diandalkan dan selalu tersedia, kemampuan sistem untuk tetap tersedia meskipun terjadi gangguan menjadi sangat penting. Penelitian ini mengeksplorasi penerapan *load balancer* dalam infrastruktur *AWS Cloud* sebagai solusi untuk menjaga ketersediaan layanan aplikasi web. Pengujian melibatkan skenario simulasi kerusakan *server* untuk mengevaluasi kemampuan *load balancer* dalam mendistribusikan trafik secara optimal ke *server* yang berfungsi dengan normal. Hasil penelitian menunjukkan bahwa implementasi *Application Load Balancer* (ALB) pada *AWS Cloud* berhasil mendistribusikan trafik ke dua *web server* secara efektif menggunakan metode *round-robin*. Berdasarkan pengujian *load* sebanyak 50 *virtual users* selama 60 detik, sistem mampu mempertahankan performa yang stabil dengan rata-rata *response time* sebesar 12,22 ms, *throughput* sebesar 97,26 *request* per detik, total *request* sebanyak 5.867 *request*, dan *error rate* 0%. Selain itu, pengujian *failover* menunjukkan bahwa ALB mampu mendeteksi kegagalan *server* melalui mekanisme *health check* dan mengalihkan trafik secara otomatis ke *server* yang masih aktif, sehingga layanan tetap tersedia tanpa *downtime* yang signifikan.

Kata Kunci: Aplikasi Web, *AWS Cloud*, *High Availability*, *Load Balancer*, *Round Robin*

ABSTRACT

As the need for reliable web applications increases, the system's ability to remain available despite disruptions becomes crucial. This study explores the implementation of a load balancer in the AWS Cloud infrastructure as a solution to maintain the availability of web application services. This study aims to ensure that web applications can continue to run consistently even if a disruption occurs on one of the servers. Using the load balancer service on the AWS Cloud platform, this study examines how traffic can be redirected to another active server to maintain service stability. Simulations were conducted by controlling the shutdown of one of the servers to see how the system can maintain application performance and accessibility. This was done considering the increasing demand for reliable and always available web applications, the ability of the system to remain available despite disruptions is crucial. This study explores the implementation of a load balancer in the AWS Cloud infrastructure as a solution to maintain the availability of web application services. Testing involved a simulated server failure scenario to evaluate the load balancer's ability to optimally distribute traffic to the normally functioning servers. The results show that the implementation of an Application Load Balancer (ALB) on the AWS Cloud successfully distributed traffic to two web servers effectively using a round-robin method. Based on load testing of 50 virtual users for 60 seconds, the system was able to maintain stable performance with an average response time of 12.22 ms, a throughput of 97.26 requests per second, a total of 5,867 requests, and a 0% error rate. Furthermore, failover testing showed that ALB was able to detect server failures through a health check mechanism and automatically redirect traffic to the remaining active server, so that services remained available without significant downtime.

Keywords: AWS Cloud, High Availability, Load Balance, Round Robin, Web Application.