**OTOMATISASI LOAD BALANCING DAN FAILOVER DENGAN MONITORING REAL-TIME PADA MIKROTIK BERBASIS PYTHON**

**TUGAS AKHIR**

Diajukan untuk memenuhi syarat-syarat memperoleh gelar Ahli Madya

Oleh :

**Christopher Aldrinovito Andriawan**

**233140700111039**



**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2025**

**BAB I**

**PENDAHULUAN**

* 1. Latar Belakang

Sebagian besar organisasi dihadapkan pada tantangan dalam mengelola load balancing dan failover secara manual, yang sering kali memerlukan waktu dan sumber daya manusia yang signifikan. Menurut laporan (Gartner, 2023), hanya sekitar 35% aktivitas jaringan yang telah diotomatisasi, sedangkan sisanya masih dikelola secara manual, yang berisiko menimbulkan kesalahan manusia dan memperlambat waktu pemulihan jaringan[1].

Mikrotik, sebagai salah satu perangkat router yang populer dan terjangkau, memang menyediakan berbagai fitur seperti scripting dan policy-based routing untuk menangani masalah ini. Namun, pengaturannya yang rumit dan membutuhkan keterampilan teknis mendalam membuat banyak organisasi, terutama skala kecil-menengah, kesulitan untuk menerapkannya secara otomatis dalam lingkungan jaringan yang dinamis. Selain itu, menurut laporan SolarWinds IT Trends Report 2022, 49% profesional TI hanya memiliki visibilitas terhadap sekitar setengah atau kurang dari aplikasi dan infrastruktur mereka, yang berdampak pada kemampuan mereka dalam mendeteksi anomali dan menganalisis akar penyebab masalah secara efektif[2]. Teori load balancing dan failover merupakan dua konsep dasar dalam jaringan komputer.

Load balancing bertujuan untuk mendistribusikan lalu lintas jaringan secara merata ke beberapa jalur atau server guna menghindari overload pada satu titik dan meningkatkan performa keseluruhan sistem. Sementara itu, failover adalah mekanisme pemulihan otomatis yang mengalihkan trafik ke jalur cadangan jika jalur utama mengalami gangguan, guna memastikan ketersediaan layanan jaringan. Konsep-konsep ini sangat penting dalam arsitektur jaringan modern, khususnya pada sistem dengan kebutuhan high availability dan redundansi (Tanenbaum & Wetherall, 2011; Kurose & Ross, 2021)[3]. Penelitian oleh Wiwik W., I., S., Ikhram H. (2023) membuktikan bahwa metode Per Connection Classifier (PCC) pada MikroTik dapat mendistribusikan trafik ke dua jalur internet sekaligus menyediakan sistem failover otomatis, yang terbukti meningkatkan stabilitas dan ketersediaankoneksi jaringan dalam pengujian nyata[4].

Penelitian oleh Farhan R. P. (2024) menunjukkan bahwa implementasi otomatisasi jaringan menggunakan Python dapat mengurangi waktu instalasi aplikasi dari rata-rata 4 menit 50 detik menjadi sekitar 30 detik pada sistem operasi CentOS, dan dari 4 menit 22 detik menjadi sekitar 28 detik pada Ubuntu[5]. Hal ini menunjukkan peningkatan efisiensi yang signifikan dibandingkan dengan metode manual.​ Penelitian oleh Dede R., Hidra A., Indri R. (2020) menunjukkan bahwa integrasi Prometheus dan Grafana dengan notifikasi Telegram memungkinkan administrator jaringan untuk mendapatkan informasi kondisi server dan layanan secara cepat, bahkan saat tidak berada di pusat pengawasan[6]. Hal ini membantu dalam menjaga stabilitas server dan layanan yang berjalan, serta memungkinkan deteksi dan penanganan masalah secara lebih efisien.​

Ide dari tugas akhir ini adalah mengimplementasikan sistem otomatisasi load balancing dan failover dengan monitoring real-time menggunakan Python. Dengan memanfaatkan API MikroTik dan SNMP, sistem ini akan secara otomatis mendeteksi kegagalan jaringan, mengganti jalur dengan lebih cepat, dan memberikan notifikasi kepada administrator melalui email, telegram, whatsapp, atau SMS. Tujuan utama dari penelitian ini adalah untuk menciptakan solusi yang dapat meningkatkan efisiensi dan ketersediaan jaringan, serta mengurangi intervensi manual yang memakan waktu.

1.1 Rumusan Masalah

* Bagaimana cara mengotomatisasi load balancing dan failover dengan monitoring real-time pada Mikrotik berbasis Python?
  1. Tujuan
* Untuk menerapkan sistem otomasi konfigurasi load balancing dan failover pada Mikrotik berbasis Python
  1. Manfaat
* Menyediakan solusi otomatisasi sederhana dan efektif menggunakan perangkat terjangkau seperti MikroTik.
* Membantu mengurangi downtime dan meningkatkan respons terhadap gangguan jaringan.
* Menambah referensi tentang pemanfaatan Python dalam otomatisasi jaringan.
  1. Batasan Masalah
     1. Topologi Jaringan:

Penelitian ini hanya menggunakan satu perangkat MikroTik RouterOS yang memiliki dua atau lebih jalur koneksi internet (multi-WAN), yang akan digunakan sebagai objek implementasi load balancing dan failover.

* + 1. Jenis Load Balancing:

Load balancing yang diimplementasikan dibatasi pada metode PCC (Per Connection Classifier) dan/atau ECMP (Equal Cost Multi-Path Routing), sesuai dengan konfigurasi standar di MikroTik.

* + 1. Failover:

Mekanisme failover yang dikembangkan hanya berfungsi untuk mendeteksi gateway yang tidak aktif dan secara otomatis mengalihkan koneksi ke jalur Cadangan secara real-time, tanpa intervensi manual.

* + 1. Monitoring Real-Time:

Monitoring dilakukan dengan script Python yang dijalankan secara periodik (internal waktu tertentu) untuk memantau status koneksi internet pada setiap jalur, berdasarkan parameter seperti ping response, latency, dan packet loss.

* + 1. Bahasa Pemrograman dan Tools:

Otomatisasi dikembangkan menggunakan Python 3, dengan bantuan library seperti paramiko (untuk SSH ke MikroTik), subprocess, atau requests. MikroTik diakses melalui SSH atau API tergantung kebutuhan sistem.

* + 1. Antarmuka Sistem:

Sistem monitoring berbasis teks/CLI (Command Line Interface) atau dashboard sederhana berbasis web local, tidak mencakup pengembangan antarmuka grafis professional atau berbasis cloud.

* + 1. Sistem Operasi dan Versi MikroTik:

Sistem diuji pada MikroTik RouterOS versi terbaru yang kompatibel dengan fitur scripting dan remote access via SSH/API, tanpa modifikasi firmware atau penggunaan perangkat tambahan lain.

* + 1. Lingkup Uji Coba:

Uji coba sistem dilakukan pada jaringan lokal (LAN) skala kecil dengan maksimal 2-3 jalur ISP dan kurang dari 10 klien, untuk memastikan efisiensi dan reliabilitas fungsionalitas otomatisasi.

**BAB II**

**KAJIAN PUSTAKA**

* 1. Landasan Teori
     1. Load Balancing

Load balancing bertugas untuk membagi dan menyeimbangkan beban trafik pada dua atau lebih ISP secara seimbang, sehingga output yang diperoleh ketika menggunakan teknik ini adalah memaksimalkan throughput, meminimalkan waktu respon, dan menghindari kelebihan beban pada satu jalur ISP ketika salah satu ISP mengalami pemutusan sambungan(Wiharti et al., 2023).

* + 1. Failover

Failover merupakan perpindahan jalur secara

otomatis pada jaringan dengan minimal ada lebih dari satu

sumber internet, dengan adanya failover downtime pada

jaringan akan semangkin kecil(Panggabean & Kuswanto, 2023).

* + 1. Monitoring Jaringan

Monitoring jaringan adalah salah satu fungsi dari manajemen yang berguna untuk menganalisis apakah

jaringan masih cukup layak untuk digunakan atau perlu tambahan kapasitas(Izzaturrahmah et al., 2025).

* + 1. MikroTik RouterOS

RouterOS adalah sistem operasi perangkat MikroTik(MikroTik, 2025).

* + 1. Bahasa Pemrograman Python

Bahasa pemrograman Python merupakan bahasa pemrograman multi platform yang banyak digunakan pada aplikasi teknologi

saat ini dan mendatang khususnya dalam bidang IoT(Trie Maya Kadarina & Muhammad Hafizd Ibnu Hajar, 2025).

* 1. Penelitian Terkait
     1. Penelitian oleh Aziz Setyawan Hidayat

Judul: Implementasi Load Balancing Dengan Metode PCC Pada Balai Besar Pelatihan Kesehatan (BBPK) Jakarta

Penelitian ini bertujuan untuk mengimplementasikan dua layanan ISP dengan metode load balancing PCC dan Failover agar tidak terjadi gagalnya koneksi ketika salah satu ISP bermasalah dan membagi beban traffic pada ISP(Hidayat et al., 2021).

* + 1. Penelitian oleh Ryo Pambudi

Judul: Implementasi Policy Base Routing dan Failover Menggunakan Router Mikrotik untuk Membagi Jalur Akses Internet di FMIPA Unnes

Dari hasil penelitian tersebut, dapat disimpulkan bahwa membagi jalur antara internet dan sistem informasi suatu instansi dapat dilakukan dan mengurangi beban pada kecepatan jalur akses, yang pada penelitianpenelitian sebelumnya hanya dilakukan untuk membagi beban penggunaan internet saja(Pambudi & Muslim, 2017).

* + 1. Penelitian oleh Ariella Giovanni Wahyuriyanto

Judul: Implementasi Sistem Monitoring Jaringan Berbasis Simple Network Management Protocol Dengan Notifikasi Telegram

Penelitian ini bertujuan untuk mendorong penerapan LibreNMS sebagai sistem manajemen jaringan yang dapat meningkatkan efisiensi pengelolaan oleh tim UP2TI(Wahyuriyanto & Yulianton, 2024).

* + 1. Penelitian oleh Muchamad Mashudi

Penelitian yang dilakukan yaitu membandingkan library Paramiko dan Netmiko untuk konfigurasi vlan 2 sampai vlan 10 pada multi-layer switch. Selain itu, penelitian ini juga meneliti mengenai standarisasi untuk otomasi perangkat berbeda vendor dengan mengunakan driver NTAF (Network Test Automation Forum)(Ayasso & Mohammad-Djafari, 2010).

* 1. Kerangka Pemikiran

Dalam era digital saat ini, kebutuhan akan koneksi internet yang stabil dan andal menjadi sangat krusial, terutama bagi organisasi yang bergantung pada layanan daring. Untuk memastikan kontinuitas layanan, diperlukan sistem jaringan yang mampu mendistribusikan beban lalu lintas secara merata dan memiliki mekanisme pemulihan otomatis saat terjadi gangguan(Irfan Oktavianto dan et al., 2019). Teknologi load balancing dan failover pada perangkat MikroTik menjadi solusi yang umum digunakan untuk memenuhi kebutuhan tersebut.

Metode Per Connection Classifier (PCC) merupakan salah satu teknik load balancing yang efektif dalam mendistribusikan lalu lintas berdasarkan koneksi individu(Wiharti et al., 2023). Dengan memanfaatkan metode ini, jaringan dapat mengoptimalkan penggunaan beberapa jalur koneksi internet secara bersamaan(Mustofa & Ramayanti, 2020). Namun, untuk meningkatkan keandalan sistem, diperlukan integrasi dengan mekanisme failover yang memungkinkan perpindahan otomatis ke jalur alternatif saat terjadi kegagalan pada salah satu koneksi. Implementasi kombinasi PCC dan failover pada MikroTik telah dibahas dalam penelitian oleh (Suryanto et al. 2018), yang menunjukkan efektivitas metode ini dalam menjaga kestabilan koneksi internet(Wiharti et al., 2023).

Selain itu, kemampuan untuk memantau status koneksi secara real-time dan memberikan notifikasi saat terjadi gangguan sangat penting dalam manajemen jaringan. Penggunaan bot Telegram sebagai alat monitoring telah dibahas dalam penelitian oleh Oktavianto dan Prayogi (2019), di mana mereka mengembangkan sistem monitoring jaringan dengan metode Equal Cost Multi Path (ECMP) yang terintegrasi dengan bot Telegram untuk memberikan notifikasi saat terjadi gangguan pada server(Mustofa & Ramayanti, 2020).

Untuk meningkatkan efisiensi dan mengurangi intervensi manual, otomatisasi konfigurasi jaringan menggunakan skrip Python menjadi solusi yang tepat. (Fahmi et al. 2021) Dalam penelitiannya menunjukkan bahwa penggunaan skrip Python dengan library Paramiko dapat mempermudah proses konfigurasi dan manajemen perangkat MikroTik secara otomatis, sehingga mengurangi potensi kesalahan manusia dan mempercepat proses konfigurasi(Maisyaroh et al., 2021).

Berdasarkan studi-studi tersebut, dapat disimpulkan bahwa integrasi antara metode load balancing PCC, mekanisme failover, monitoring real-time menggunakan bot Telegram, dan otomatisasi konfigurasi dengan skrip Python dapat menciptakan sistem jaringan yang andal, efisien, dan responsif terhadap gangguan. Kerangka pemikiran ini menjadi dasar dalam pengembangan sistem otomatisasi load balancing dan failover dengan monitoring real-time pada MikroTik berbasis Python.

**BAB III**

**METODE PENYELESAIAN TUGAS AKHIR**

* 1. Jenis Penelitian

Penelitian ini termasuk dalam jenis penelitian terapan (applied research), yaitu penelitian yang bertujuan untuk mengembangkan dan menerapkan Solusi praktis terhadap permasalahan nyata di bidang jaringan komputer, khususnya dalam pengelolaan koneksi internet menggunakan perangkat MikroTik.

Selain itu, metode yang digunakan dalam penelitian ini bersifat eksperimen rekayasa (engineering experimental), di mana peneliti merancang, mengimplementasikan, dan menguji sistem otomatisasi menggunakan bahasa pemrograman Python untuk melakukan monitoring real-time serta eksekusi load balancing dan failover pada perangkat jaringan.

Ciri khas dari jenis penelitian ini adalah adanya:

* Perancangan sistem berdasarkan kebutuhan dan permasalahan riil,
* Penggunaan tools teknis dan scripting dalam implementasi,
* Evaluasi sistem berdasarkan hasil uji coba terhadap kondisi koneksi yang bervariasi.

Dengan pendekatan ini, penelitian tidak hanya bersifat konseptual, namun menghasilkan prototipe sistem yang dapat diimplementasikan langsung dalam lingkungan jaringan nyata.

* 1. Metode Pengumpulan Data

Dalam penelitian ini, data yang dikumpulkan bersifat primer dan sekunder, yang digunakan untuk merancang serta mengevaluasi sistem otomatisasi load balancing dan failover berbasis Python pada perangkat MikroTik. Adapun metode pengumpulan data yang digunakan antara lain:

1. Studi Literatur

Metode ini digunakan untuk memperoleh dasar teori serta referensi teknis yang mendukung perancangan sistem. Studi dilakukan terhadap:

* Dokumen resmi MikroTik (RouterOS),
* Dokumentasi pustaka Python (seperti paramiko, subprocess, dan ping3),
* Artikel ilmiah, skripsi, serta jurnal terkait topik load balancing, failover, dan monitoring jaringan.

1. Observasi Langsung

Peneliti melakukan pengamatan langsung terhadap performa koneksi internet dari beberapa ISP pada perangkat MikroTik. Data yang dikumpulkan meliputi:

* Stabilitas koneksi,
* Waktu respon (latency),
* Kejadian gangguan (link down),
* Efektivitas metode failover manual yang digunakan sebelumnya.

1. Eksperimen dan Simulasi

* Memutuskan koneksi salah satu ISP secara sengaja (simulasi link failure),
* Mengamati kecepatan failover,
* Mencatat log hasil monitoring dan eksekusi perintah otomatis,
* Menguji kembali proses failback saat koneksi kembali normal.

1. Dokumentasi Sistem

Selama proses pengembangan, semua hasil pengujian dicatat dan didokumentasikan secara sistematis, termasuk tangkapan layar konfigurasi, script Python yang digunakan, hasil output, dan log monitoring real-time sebagai data pendukung.

* 1. Perancangan Sistem

Perancangan sistem dalam penelitian ini bertujuan untuk membangun sebuah mekanisme otomatis yang mampu melakukan monitoring koneksi internet secara real-time, serta mengeksekusi proses load balancing dan failover secara dinamis pada perangkat MikroTik, dengan bantuan bahasa pemrograman Python.

* + 1. Arsitektur Sistem

Secara umum, sistem terdiri dari dua komponen utama:

1. Perangkat MikroTik Router – berperan sebagai gateway jaringan dan pusat konfigurasi routing.
2. Skrip Python – bertugas untuk melakukan monitoring konektivitas dan mengeksekusi perintah otomatis ke MikroTik melalui protokol SSH.

Sistem bekerja dalam siklus berulang (loop) dengan tahapan sebagai berikut:

1. Python melakukan ping ke IP publik (misalnya 8.8.8.8) melalui masing-masing gateway ISP.
2. Jika salah satu jalur ISP terdeteksi down, Python akan mengeksekusi perintah failover ke MikroTik menggunakan SSH.
3. Jika semua jalur aktif, Python dapat menerapkan load balancing berdasarkan logika tertentu (misal round-robin atau PCC).
4. Python juga mencatat hasil monitoring dalam file log dan dapat diintegrasikan dengan sistem notifikasi seperti Telegram atau email (opsional).
   * 1. Diagram Blok Sistem
     2. Komponen Perangkat Lunak

* Python 3.13.3 – digunakan untuk scripting otomatisasi.
* Library Python:
  + Ping3/os – untuk monitoring status koneksi (ICMP ping).
  + Paramiko – untuk SSH remote ke MikroTik.
  + time – untuk pengulangan dan jeda.
  + logging – untuk pencatatan log hasil monitoring.
* RouterOS Mikrotik – sistem operasi router yang dikonfigurasi menggunakan CLI via SSH.
* (Opsional) Telegram Bot API – untuk memberikan notifikasi real-time kepada admin jaringan.
  + 1. Skema Alur Kerja Sistem
    2. Konfigurasi Dasar MikroTik

Untuk mendukung sistem ini, MikroTik perlu dikonfigurasi dengan:

* Static Routes dengan check-gateway=ping,
* IP Firewall Mangle (jika menggunakan metode PCC),
* Script untuk logging atau fallback manual (opsional),
* Akses SSH diaktifkan, agar Python dapat melakukan kontrol jarak jauh.
  1. Tools dan Bahan

Dalam pelaksanaan penelitian ini, digunakan berbagai tools (perangkat lunak dan keras) serta bahan pendukung yang menunjang proses perancangan, implementasi, dan pengujian sistem. Berikut adalah rincian tools dan bahan yang digunakan:

* + 1. Perangkat Keras (Hardware)

|  |  |  |
| --- | --- | --- |
| No. | Nama Alat | Spesifikasi / Keterangan |
| 1. | Laptop / PC | Intel Core i7 Gen-13, RAM 24 GB, SSD 512 GB |
| 2. | Router MikroTik | MikroTik RB750Gr3 (hEX) atau setara |
| 3. | Modem / Internet Gateway (ONT) | 2 buah modem/router dari ISP berbeda |
| 4. | Kabel UTP | CAT5e atau CAT6 |
| 5. | Switch | Jika dibutuhkan untuk manajemen jaringan |

* + 1. Perangkat Lunak (Software)

|  |  |  |
| --- | --- | --- |
| No. | Nama Software | Keterangan |
| 1. | Python 3.13.3 | Bahasa pemrograman utama untuk scripting |
| 2. | Winbox | GUI untuk konfigurasi MikroTik |
| 3. | Terminal (Linux / Windows) | Untuk menjalankan skrip Python |
| 4. | RouterOS (MikroTik) | Sistem operasi perangkat router MikroTik |
| 5. | Telegram (opsional) | Untuk integrasi notifikasi real-time |

* + 1. Library dan Modul Python

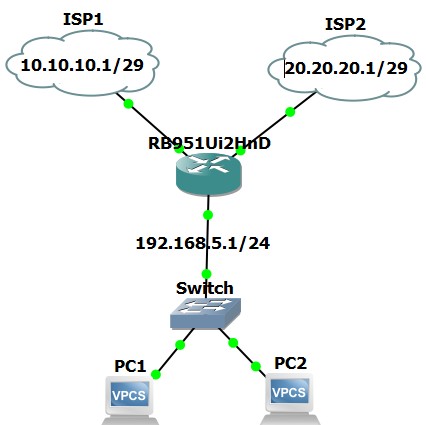
|  |  |  |
| --- | --- | --- |
| No. | Library Python | Fungsi |
| 1. | paramiko | SSH remote ke MikroTik |
| 2. | ping3 / os.system(‘ping’) | Monitoring status koneksi (ICMP ping) |
| 3. | time | Delay dan pengulangan proses monitoring |
| 4. | logging | Menyimpan log hasil monitoring system |
| 5. | requests | Untuk akses API Telegram atau monitoring eksternal |

* + 1. Bahan Pendukung

|  |  |  |
| --- | --- | --- |
| No. | Bahan | Keterangan |
| 1. | Koneksi Internet | Minimal 2 koneksi dari ISP yang berbeda |
| 2. | Dokumentasi MikroTik | Sebagai referensi konfigurasi routing |
| 3. | Artikel / jurnal ilmiah | Penelitian terkait load balancing & failover |

* + 1. Perancangan dan Implementasi

Untuk mempermudah dalam implementasi maka perlu dibuat topologi jaringan seperti yang ditunjukkan pada Gambar 1.



* 1. Tahapan Implementasi  
     Tahapan implementasi dilakukan secara sistematis agar sistem dapat berjalan sesuai dengan kebutuhan. Implementasi dibagi menjadi beberapa tahap utama, yaitu perencanaan, pengembangan, pengujian, dan dokumentasi. Berikut penjabaran setiap tahapan:
     1. Analisis Kebutuhan Sistem

Tahapan ini bertujuan untuk mengidentifikasi seluruh kebutuhan yang diperlukan, baik dari sisi perangkat keras, perangkat lunak, maupun fungsi sistem. Hasil dari tahap ini menjadi dasar dalam menentukan struktur program dan konfigurasi jaringan.

* Menentukan perangkat MikroTik yang digunakan
* Menentukan jumlah dan jenis koneksi ISP
* Menentukan tools pemrograman yang digunakan (Python + libraries)
* Menentukan metode load balancing dan failover
  + 1. Perancangan Sistem

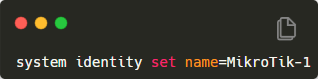
Pada tahap ini dilakukan perancangan struktur logika monitoring dan kontrol jaringan, baik secara manual di MikroTik maupun otomatis dengan Python.

* Membuat flowchart dan diagram alur kerja sistem
* Merancang konfigurasi routing MikroTik (static route dan mangle)
* Menyusun kerangka skrip Python (monitoring, logging, SSH command)
  + 1. Konfigurasi MikroTik

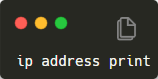
Konfigurasi dasar MikroTik dilakukan untuk memastikan sistem siap menerima kontrol dari skrip Python. Pengaturan mencakup:

* Menambahkan dua atau lebih gateway internet
* Menyusun static route dengan parameter check-gateway=ping
* Mengaktifkan akses SSH untuk remote script
* Konfigurasi NAT dan firewall bila diperlukan
  + - 1. Konfigurasi MikroTik-1

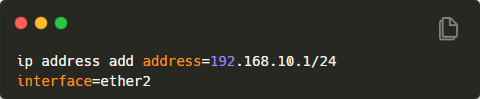
1. Gunakan Identity sesuai devicenya



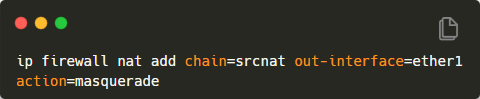
1. Verifikasi ke internet



1. Masukkan IP Address setiap interface sesuai topologi

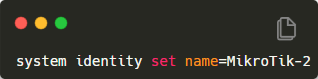


1. Setup firewall NAT

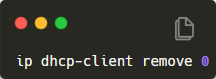


* + - 1. Konfigurasi MikroTik-2

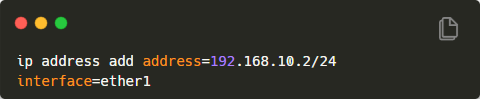
1. Gunakan Identity sesuai devicenya



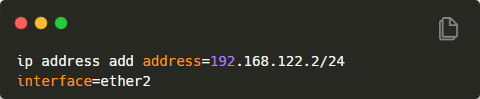
1. Hapus DHCP Client



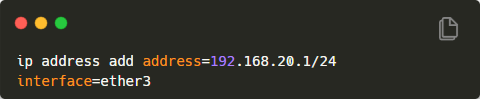
1. Masukkan IP Address setiap interface sesuai topologi



Interface yang mengarah ke ISP 1

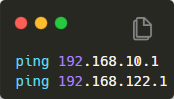


Interface yang mengarah ke ISP 2

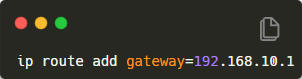


Interface yang mengarah ke LAN

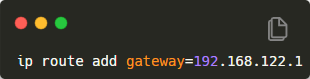
1. Ping ke MikroTik-1 dan Internet



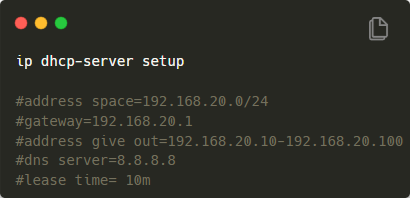
1. Buat default route lewat interface router tetangga (MikroTik-1) dan internet



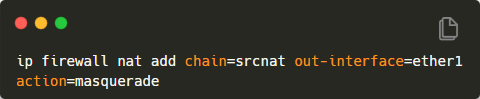
Disable terlebih dahulu routing yang mengarah ke ISP 1



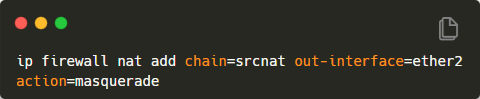
1. Setup DHCP Server untuk LAN



1. Setup firewall NAT

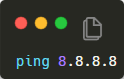


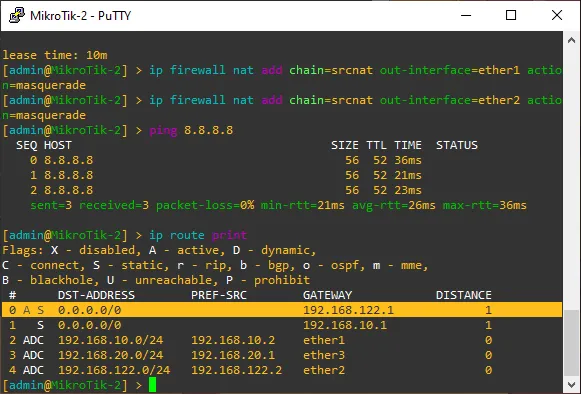
Firewall NAT ke ISP 1



Firewall NAT ke ISP 2

1. Verifikasi ke internet



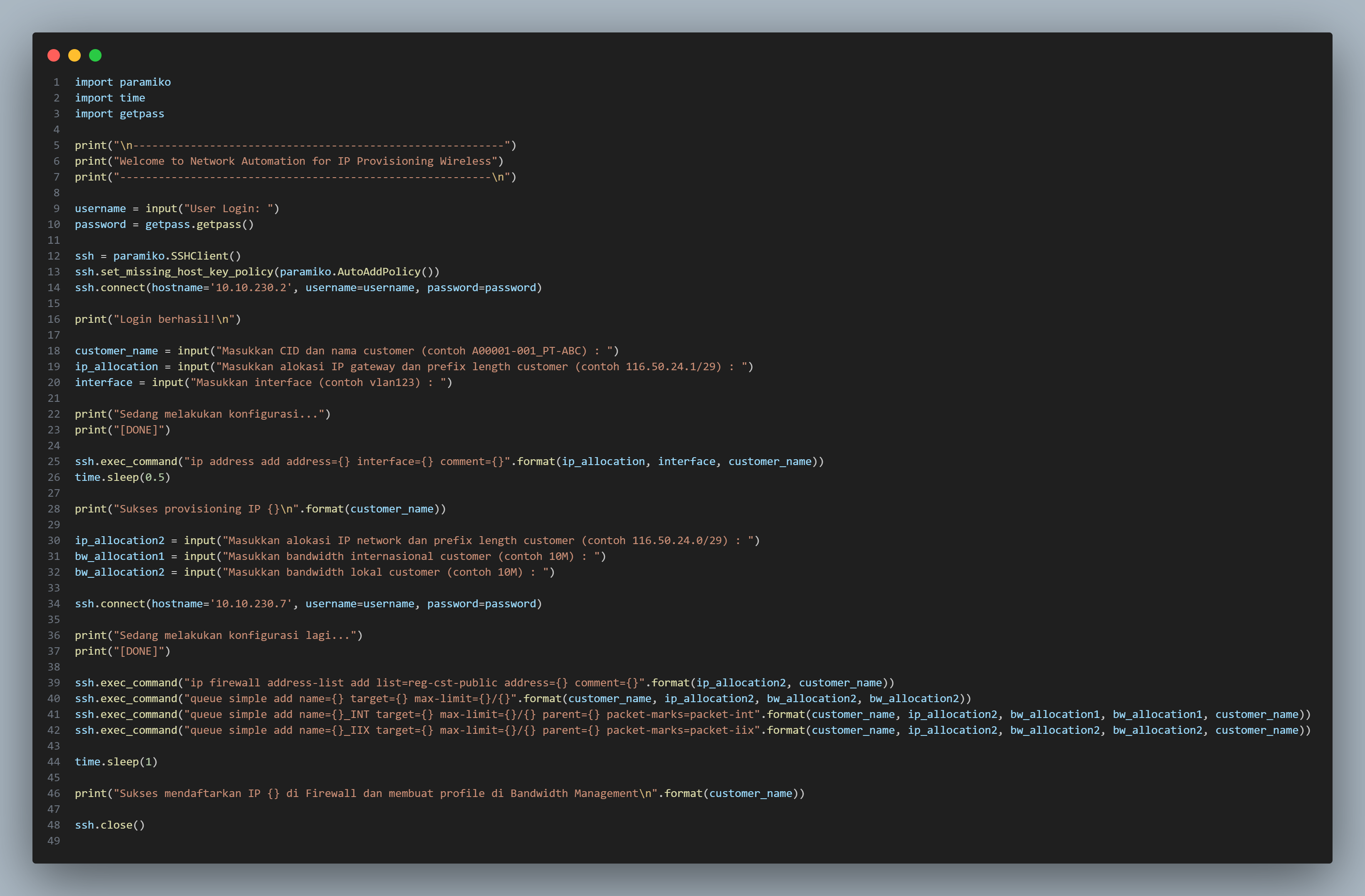


ISP 2 aktif secara default oleh MikroTik-2

* + 1. Pembuatan Skrip Python

Skrip Python dikembangkan untuk melakukan monitoring koneksi internet dan mengatur failover/load balancing secara otomatis. Fitur utama meliputi:

* Monitoring IP gateway tiap ISP dengan metode ping
* Deteksi otomatis koneksi down atau up
* Eksekusi perintah routing melalui SSH menggunakan paramiko
* Logging hasil monitoring ke dalam file log
* (Opsional) Mengirim notifikasi real-time via Telegram
  + - 1. Instalasi Paramiko
      2. Membuat file script dengan nama provisioning1.py:



* + 1. Pengujian Sistem

Pengujian dilakukan dengan skenario berikut:

* Simulasi koneksi internet putus (ISP down)
* Pengamatan reaksi sistem terhadap gangguan (failover aktif)
* Pengujian kembali saat koneksi normal (failback aktif)
* Analisis log hasil monitoring
* Uji efektivitas load balancing (jika digunakan)
* Hasil dan Pembahasan

|  |  |  |
| --- | --- | --- |
| No. | Percobaan | Waktu |
| 1. | Livestreaming via Youtube  Dengan kualitas video 1080p  Dengan durasi 3:18 | 2:32:31 |
| 2. | Livestreaming via Youtube  Dengan kualitas video 720p  Dengan durasi 3:18 | 1:38:04 |
| 3. | Unduh file Raspberry Lite via FTP  Dengan ukuran 294 MB  Pada Server Repository UGM | 14:11:18 |
| 4. | Unduh file Raspberry Lite via HTTP  Dengan ukuran 294 MB  Pada server Repository UGM | 14:05:17 |

* + 1. Dokumentasi dan Evaluasi

Setiap langkah dan hasil dari implementasi didokumentasikan dengan baik untuk keperluan laporan dan pengujian. Evaluasi dilakukan untuk menilai:

* Kinerja sistem dalam menangani gangguan
* Ketapatan logika monitoring
* Kecepatan eksekusi failover
* Kemudahan pemeliharaan sistem
  1. Metode Pengujian

Pengujian dilakukan untuk memastikan bahwa sistem otomatisasi load balancing dan failover yang dikembangkan dapat berjalan dengan baik dan sesuai dengan tujuan.

Pengujian dilakukan secara terstruktur menggunakan metode black-box testing dan pengujian skenario, dengan fokus pada fungsi sistem, bukan kode internal.

* + 1. Jenis Pengujian

1. Black-Box Testing

Pengujian ini dilakukan dengan cara mengamati output dari sistem berdasarkan input tertentu tanpa melihat proses internal dalam kode program. Dalam konteks ini, dilakukan pengujian terhadap:

* Fungsi monitoring koneksi
* Respons sistem terhadap perubahan status ISP (up/down)
* Eksekusi otomatis perintah routing di MikroTik
* Notifikasi/log aktivitas sistem

1. Skenario Pengujian

Beberapa scenario berikut dirancang untuk menguji keandalan sistem:

|  |  |  |
| --- | --- | --- |
| No. | Skenario | Tujuan Pengujian |
| 1. | ISP 1 aktif, ISP 2 aktif | Memastikan sistem default berjalan normal |
| 2. | ISP 1 mati, ISP 2 aktif | Menguji fungsi failover ke ISP 2 |
| 3. | ISP 1 hidup kembali setelah mati | Menguji fungsi failback ke ISP 1 |
| 4. | Kedua ISP mati | Menguji deteksi kondisi no-connection |
| 5. | Sistem mengirim log saat perubahan status koneksi | Memastikan fitur monitoring/log berjalan |

* + 1. Parameter Pengujian

Beberapa parameter yang diamati selama pengujian antara lain:

* Waktu Deteksi Gangguan: Berapa lama waktu yang dibutuhkan sistem untuk mendeteksi bahwa sebuah koneksi down.
* Waktu Peralihan (Switch Time): Seberapa cepat system melakukan failover ke koneksi cadangan.
* Stabilitas Failback: Apakah sistem dapat kembali menggunakan koneksi utama saat sudah normal.
* Konsistensi Logging: Apakah semua aktivitas tercatat dalam log secara akurat.
* Notifikasi (Opsional): Jika diaktifkan, apakah sistem dapat mengirim notifikasi ke Telegram atau media lain.
  + 1. Dokumentasi Hasil Pengujian

Hasil dari setiap skenario akan dicatat dan disusun dalam bentuk tabel evaluasi. Setiap hasil akan dievaluasi berdasarkan:

* Berhasil / Gagal
* Waktu respon
* Dampak terhadap jaringan
* Analisis log output
* Catatan tambahan (jika ada error)