

Pemanfaatan Paramiko dan GlusterFS untuk Optimasi Sistem Operasi Linux sebagai Solusi Penyimpanan Distribusi

Anya Nur Defitri^{1*}, Jaren², M. Azrel Samudra³

¹Sistem Komputer, Universitas Sriwijaya, Indonesia

*Korespondensi: anyanurdefitri12@gmail.com

ARTICLE INFO

Article History:

- Received 27 March 2024
- Received in revised form 27 March 2024
- Accepted

ABSTRAK

Dalam era digital yang berkembang pesat saat ini, permintaan akan penyimpanan data yang efisien dan handal semakin meningkat. Sistem operasi Linux telah menjadi salah satu platform yang paling umum digunakan untuk infrastruktur penyimpanan, baik di lingkungan perusahaan maupun individu. Fokus utama dalam pengembangan sistem penyimpanan modern adalah teknologi yang memungkinkan manajemen penyimpanan distribusi. Salah satu solusi yang populer di lingkungan Linux adalah GlusterFS, yang memungkinkan penyediaan penyimpanan distribusi yang fleksibel dan dapat diperluas. Namun, pengelolaan dan pengaturan penyimpanan distribusi ini memerlukan alat manajemen yang sesuai. Di sisi lain, Paramiko adalah pustaka Python serbaguna yang digunakan untuk implementasi protokol SSH. Kemampuannya dalam manajemen remote menjadikannya pilihan menarik untuk diintegrasikan dalam pengelolaan sistem operasi Linux, termasuk dalam konteks pengelolaan penyimpanan distribusi.

Kata Kunci: sistem operasi, Linux, sistem file, Paramiko, GlusterFS

ABSTRACT

In today's rapidly evolving digital age, the demand for efficient and reliable data storage is increasing. The Linux operating system has become one of the most commonly used platforms for storage infrastructure, both in enterprise and individual environments. A major focus in the development of modern storage systems is the technology that enables distributed storage management. One of the popular solutions in the Linux environment is GlusterFS, which enables flexible and extensible provisioning of distributed storage. However, the management and organization of this distributed storage requires appropriate management tools. On the other hand, Paramiko is a versatile Python library used for SSH protocol implementation. Its capabilities in remote management make it an attractive option to integrate in the management of Linux operating systems, including in the context of managing distribution storage.

Keywords: operating system, Linux, file system, Paramiko, GlusterFS

1. PENDAHULUAN

Komputer tersusun dari empat komponen utama, yaitu perangkat keras (hardware), perangkat lunak (software), brainware, dan data. Sistem operasi berperan sebagai penghubung antara pengguna komputer (brainware) dengan perangkat keras (hardware). Peran sistem operasi sangatlah vital, karena sistem operasi lah yang memungkinkan pengguna untuk berinteraksi dengan perangkat keras dan menjalankan berbagai program[1].

Sistem operasi merupakan suatu perangkat lunak sistem yang mempunyai tugas mengelola sumber daya dari perangkat lunak dan perangkat keras agar dapat bekerja dengan baik serta memudahkan proses interaksi antara computer dengan user[2].

Sistem operasi umumnya bertanggung jawab atas pengelolaan semua sumber daya dalam sebuah sistem komputer dan menyediakan layanan kepada pengguna melalui panggilan sistem untuk mempermudah dan meningkatkan penggunaan sumber daya tersebut. Oleh karena itu, sistem operasi menjadi perangkat lunak yang sangat penting untuk mengoperasikan komputer dengan efisien. Linux, sebagai contoh sistem operasi, dikenal karena fleksibilitasnya. Linux dapat berfungsi secara mandiri sebagai PC tunggal atau digunakan sebagai workstation atau PC klien[3].

Salah satu inovasi penting yang berasal dari lingkungan Linux adalah GlusterFS, sebuah sistem file terdistribusi yang open source[4]. Dalam terminologi Gluster, terdapat volume yang merupakan bagian dari server yang menampung sistem file ruang kernel aktual untuk penyimpanan data dan pengeksposan data kepada klien. Setiap volume dapat dibangun oleh beberapa subvolume, umumnya di-hosting oleh server yang berbeda[5].

Pada penelitian ini bertujuan untuk mengoptimalkan sistem operasi Linux sebagai solusi penyimpanan distribusi. Optimalisasi ini akan dicapai melalui implementasi dan integrasi teknologi GlusterFS dan Paramiko. GlusterFS dan Paramiko adalah dua teknologi yang saling melengkapi untuk menghadirkan solusi penyimpanan data yang aman dan efisien. GlusterFS, sebagai sistem file terdistribusi, memungkinkan penyimpanan data di berbagai server yang terhubung dalam jaringan[6]. Hal ini memberikan skalabilitas dan redundansi yang tinggi, sehingga data dapat diakses dengan mudah dan aman dari berbagai lokasi. Paramiko adalah library python yang merupakan implementasi dari protokol SSH versi 2 yang mempersiapkan fungsionalitas client dan server. Semua perangkat jaringan yang bisa dikonfigurasi dengan SSH juga dapat menggunakan library ini[7].

Implementasi GlusterFS dan Paramiko dalam penelitian ini diharapkan dapat memberikan solusi inovatif untuk penyimpanan data yang terdistribusi dan aman. Integrasi kedua teknologi ini akan memungkinkan pengguna untuk memanfaatkan berbagai keuntungan, seperti skalabilitas, efisiensi, keamanan, dan aksesibilitas.

2. TINJAUAN PUSTAKA

2.1 Sistem Operasi

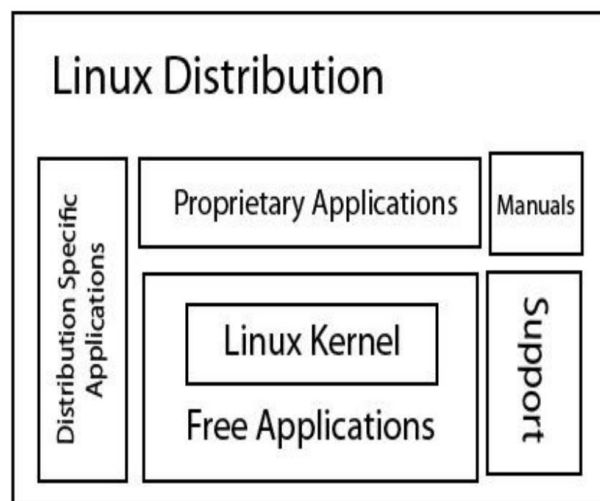
Sistem operasi dapat dibandingkan dengan pemerintah dalam suatu negara, karena mereka bertanggung jawab untuk menciptakan kondisi yang memungkinkan komputer menjalankan program dengan benar[8]. Menurut Abdul Kadir (2013:164), sistem operasi merupakan program utama di antara program-program yang ada dalam sistem komputer. Fungsinya mirip dengan program kontrol yang mengelola eksekusi program lain di dalam komputer. Sistem operasi memungkinkan pengguna untuk berinteraksi dengan perangkat keras secara lebih efisien dan memberikan cara yang lebih mudah bagi para pengembang untuk bekerja dengan perangkat keras tersebut[9].

2.2 Sistem Operasi Linux

Linux merupakan sebuah sistem operasi lengkap yang menyediakan semua kemampuan yang biasanya terkait dengan sistem Unix komersial. Linux memiliki kompilator yang sangat baik untuk bahasa pemrograman C, C++, Pascal, Modula-2 dan -3, Oberon, Smalltalk, dan Fortran. Editor seperti Emacs dan vi tersedia dalam beberapa versi. Keamanan kata sandi, perlindungan file, login ganda, memori virtual, dan pemrosesan multipel didukung sepenuhnya[10][9]. Jaringan untuk Linux telah berkembang hingga tahap di mana jaringan besar dapat mendukung login remote, shell remote, dan surat elektronik. Sebuah versi dari Network File System (NFS) telah dikembangkan untuk Linux. Ini memungkinkan berbagi sistem file di antara beberapa mesin, sehingga mengurangi penggunaan hard disk dan meminimalkan overhead administrasi sistem[6]. Sistem pemrosesan teks TeX/LaTeX telah dipindahkan ke Linux bersama dengan fasilitas pratinjau (ghostview dan xdvi) dan paket gambar (xfig dan idraw). Semua kemampuan ini tersedia secara gratis tanpa biaya lisensi.

2.3 Struktur Sistem Operasi Linux

Linux bukanlah hanya sebuah kernel sistem operasi yang berdiri sendiri. Meskipun kernel merupakan bagian inti yang sangat penting, namun sebuah sistem operasi Linux terdiri dari lebih dari sekadar kernel[9]. Sistem operasi berbasis Linux biasanya terbentuk dengan menggabungkan kernel Linux dengan berbagai alat tambahan seperti kompiler, konektor, perakitan, serta alat-alat tingkat lanjut seperti editor teks dan antarmuka grafis. Skema umum dari sistem operasi berbasis Linux dapat dilihat dalam Gambar 1.



Gambar 1. Skema Kernel Linux

Di sini, kernel Linux dipadukan dengan berbagai aplikasi dan program untuk membentuk apa yang kita kenal sebagai Distribusi Linux. Variasi antar distribusi terutama dipengaruhi oleh manajemen dan pengaturan yang berbeda.

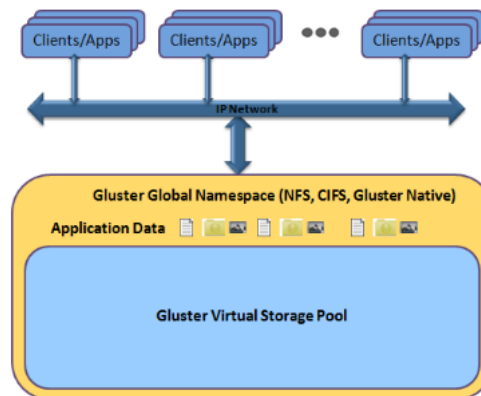
2.4 File System

Menurut Kamus Besar Bahasa Indonesia, kata "*file system*" atau "sistem fail" berasal dari kata "sistem" dan "fail". Sistem berarti 1) kumpulan elemen yang saling berhubungan secara teratur sehingga membentuk suatu totalitas, 2) susunan yang teratur dari perspektif, teori, asas, dan sebagainya, dan 3) metode[11][12][13]. Tidak seperti dokumen atau berkas, sistem file adalah cara untuk memberi nama pada file dan meletakkannya di media penyimpanan. Oleh karena itu, file system dapat didefinisikan sebagai cara untuk mengatur semua hal yang berhubungan dengan dokumen atau berkas sehingga susunannya menjadi teratur[5].

2.5 Gluster Fs

GlusterFS adalah aplikasi open source yang bermanfaat untuk proses berbagi file antar server. Ini adalah solusi terukur dan murah untuk kebutuhan sistem berkas terdistribusi yang memenuhi kebutuhan penyimpanan dalam berbagai lingkungan[5].

GlusterFS didesain untuk memenuhi kebutuhan ruang penyimpanan bagi pengguna dan dapat memberikan kinerja yang luar biasa untuk beban kerja yang beragam. Selain digunakan untuk memenuhi ruang penyimpanan, GlusterFS juga dapat digunakan untuk mem-backup data server apabila salah satu server down sehingga data tetap aman dan dapat sinkron kembali. GlusterFS juga dapat menangani ribuan client sehingga pemrosesan dapat dilakukan pada banyak gluster client sehingga dapat dilakukan pembagian request pada setiap server[5].



Gambar 2. Arsitektur GlusterFS

GlusterFS memungkinkan pengguna untuk menghilangkan sistem penyimpanan tertutup yang awalnya mandiri, mahal, dan menggunakan perangkat penyimpanan murah biasa untuk menerapkan kolam penyimpanan terpusat, yang dikelola secara terpusat, berskala, dan tervirtualisasi, dan kapasitas penyimpanannya dapat diperluas hingga kelas TB / PB. GlusterFS juga memiliki skalabilitas dan keandalan yang kuat juga[14].

2.6 Paramiko

Paramiko merupakan sebuah pustaka Python yang mengimplementasikan protokol SSH versi 2 dan menyediakan fungsionalitas client dan server. Semua perangkat jaringan yang dapat dikonfigurasi melalui SSH juga dapat menggunakan pustaka ini. Selain itu, Paramiko juga mampu menjalankan skrip konfigurasi secara langsung, yang kemudian akan dikirimkan kepada perangkat yang akan dikonfigurasi melalui protokol SSH[7][15].

Secure Shell (SSH) adalah protokol jaringan yang memungkinkan pertukaran data yang aman antara dua perangkat atau komputer melalui jaringan yang tidak aman, seperti internet[16]. Paramiko memberikan kemampuan untuk berkomunikasi dengan perangkat atau server yang menjalankan layanan SSH, melakukan operasi remote, dan mengirim perintah melalui koneksi yang aman. Pustaka ini menyediakan kelas-kelas yang memungkinkan pembuatan koneksi SSH, autentikasi pengguna, dan eksekusi perintah pada perangkat atau server jarak jauh. Dengan menggunakan Paramiko, pengembang dapat membuat skrip Python yang dapat berinteraksi dengan perangkat atau server SSH tanpa harus bergantung pada klien SSH eksternal. Ini sangat berguna dalam otomatisasi tugas-tugas sistem, manajemen server, dan pengelolaan infrastruktur jaringan secara remote[15].

3. METODELOGI PENELITIAN

3.1 Jenis Percobaan

Eksperimen ini bertujuan untuk mengevaluasi kinerja Paramiko dan GlusterFS dalam mengelola penyimpanan terdistribusi dan sinkronisasi file di berbagai node. Melalui percobaan ini, kita dapat mendalami cara kerja keduanya dalam menggabungkan kapasitas

penyimpanan dan menjaga konsistensi file di seluruh kluster. Tujuan utamanya adalah untuk memahami secara lebih mendalam kemampuan serta kelebihan dan kekurangan implementasi dan kinerja Paramiko dan GlusterFS dalam skenario ini.

3.2 Keperluan Perangkat

Adapun perangkat yang digunakan dalam percobaan ini adalah sebagai berikut:

1. Perangkat Lunak (*Software*)

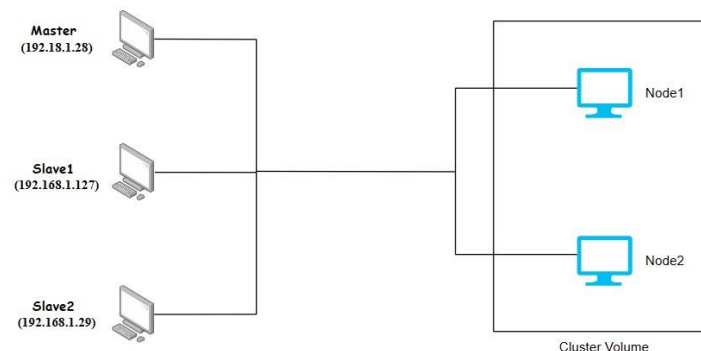
- a. *Microsoft Word Student 2019*
- b. *Mendeley Desktop*
- c. *Draw.io*

2. Sedangkan perangkat keras (*Hardware*) yang dibutuhkan antara lain:

- a. Laptop dengan prosessor 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
2.42 GHz
- b. *System type* 64-bit

3.3 Topologi Jaringan

Eksperimen ini menggunakan sebuah jaringan lokal untuk menghubungkan komputer-komputer dalam sebuah kluster. Setiap node dalam kluster Paramiko dan GlusterFS terkoneksi melalui *switch* jaringan, yang memungkinkan pertukaran data yang efisien di antara mereka. Dengan konfigurasi jaringan seperti ini, kluster Paramiko dan GlusterFS dapat berinteraksi secara langsung, yang memungkinkan implementasi penyimpanan terdistribusi dan sinkronisasi file yang optimal di seluruh node.



Gambar 3. Topologi Jaringan GlusterFS

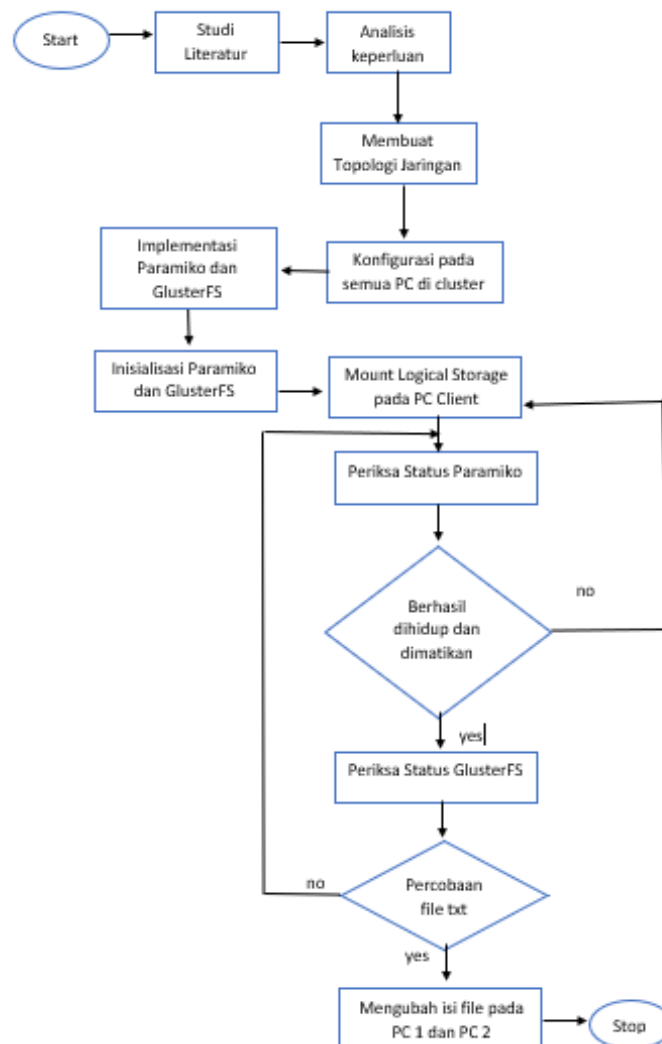
Pada gambar 3 menampilkan sebuah konfigurasi jaringan Paramiko dan GlusterFS yang terdiri dari tiga node: master, slave 1, dan slave 2, yang semua terkoneksi ke volume kluster. Node 1 memiliki koneksi langsung dengan Node 2. Perangkat yang digunakan dalam konfigurasi ini adalah sebagai berikut penjelasannya:

- Master merupakan komputer atau server pusat yang bertanggung jawab atas pengaturan dan kontrol utama. Master bisa berupa server berkekuatan tinggi atau mesin virtual dengan spesifikasi yang sesuai.

- Slave 1 dan Slave 2 adalah node tambahan yang berfungsi sebagai penyimpanan dan pengelolaan data. Node ini dapat berupa server fisik atau mesin virtual dengan spesifikasi yang tepat.
- Node 1 dan Node 2 merupakan node tambahan lainnya. Node 1 terhubung langsung ke Node 2. Node-node ini mungkin memiliki peran spesifik dalam menjaga koneksi antar-node atau fungsi lain dalam infrastruktur jaringan Paramiko dan GlusterFS.

3.4 Flowchart dari Percobaan

Tahap percobaan ini menguraikan langkah-langkah yang diambil untuk menyelesaikan masalah penelitian. Tujuannya adalah untuk menguji kinerja Paramiko dan GlusterFS dalam mengintegrasikan kapasitas penyimpanan terdistribusi dan sinkronisasi file antara node-node dalam sebuah kluster. Gambar 4 ini merupakan Flowchart yang dapat menjelaskan tahapan-tahapan yang akan dilakukan sebagai berikut :



Gambar 4. Flowchart

Berikut penjelasan dari flowchart diatas :

1. Mulai : Memulai langkah-langkah eksperimen.
2. Tinjauan Pustaka : Mengkaji literatur untuk memperoleh pemahaman tentang dasar-dasar GlusterFS, termasuk konsep, arsitektur, fitur-fitur, dan implementasi sebelumnya. Tinjauan literatur membantu dalam memperoleh pemahaman menyeluruh tentang topik penelitian.
3. Analisis Kebutuhan : Menganalisis kebutuhan untuk menentukan persyaratan dan keperluan yang diperlukan dalam implementasi GlusterFS. Ini mencakup identifikasi perangkat keras, perangkat lunak, dan konfigurasi jaringan yang diperlukan untuk menjalankan percobaan.
4. Rancangan Topologi : Merancang topologi jaringan GlusterFS yang akan digunakan dalam percobaan, termasuk jumlah dan konfigurasi node, serta hubungan antara mereka dalam kluster GlusterFS.
5. Konfigurasi Seluruh Node di Kluster : Melakukan konfigurasi awal pada semua node dalam kluster, termasuk instalasi perangkat lunak GlusterFS dan penyesuaian konfigurasi seperti pengaturan firewall dan pengaturan host untuk memungkinkan komunikasi yang lancar antara node-node.
6. Mount Penyimpanan Logis pada Klien : Melakukan mounting penyimpanan logis GlusterFS pada PC klien untuk akses data, memungkinkan PC klien untuk mengakses dan menggunakan penyimpanan yang dikelola oleh GlusterFS.
7. Periksa Status Paramiko: Diagram alir kemudian memeriksa status Paramiko, jika Ya: Jika Paramiko aktif dan berjalan, proses berlanjut ke langkah berikutnya. Jika Tidak: Jika Paramiko tidak aktif dan berjalan, proses melompat ke langkah 15 untuk memulai ulang Paramiko dan GlusterFS.
8. Periksa Status Gluster di Semua Node : Memeriksa status layanan GlusterFS pada semua node dalam kluster untuk memastikan semua node terhubung dan berfungsi dengan baik sebelum melanjutkan ke tahap berikutnya.
9. Percobaan dengan File txt : Melakukan percobaan dengan membuat, menyimpan, dan mengakses file txt pada volume GlusterFS yang telah dibuat untuk menguji ketersediaan dan konsistensi data di seluruh kluster GlusterFS.
10. Modifikasi Isi File pada Node B dan C : Melakukan percobaan dengan mengubah isi file pada Node B dan C untuk menguji sinkronisasi antara node, memastikan bahwa perubahan pada satu node tercermin secara otomatis di node lainnya.
11. Stop : Mengakhiri langkah-langkah eksperimen, memungkinkan peneliti untuk menganalisis hasil percobaan dan menarik kesimpulan dari eksperimen yang telah dilakukan.

4. HASIL DAN PEMBAHASAN

Dalam bab ini, akan dibahas secara mendalam hasil dari penerapan Paramiko dan GlusterFS sebagai sistem file di platform Linux. Fokus utama adalah pada konfigurasi yang akurat pada sistem operasi Linux untuk memastikan kinerja dan keandalan sistem file yang dijalankan. Berikut Langkah-langkah implementasi paramiko dan GlusterFS

4.1. Buat Direktori Gluster:

Buat direktori di setiap PC yang akan digunakan sebagai penyimpanan GlusterFS.

4.2. Edit File Konfigurasi GlusterFS:

Buka file `/etc/glusterfs/glusterfsd.conf` di setiap PC.

Tambahkan baris berikut ke bagian akhir file:

`volume-name all`

`brick /mnt/glusterfs/brick1`

Ganti `/mnt/glusterfs/brick1` dengan lokasi direktori GlusterFS di PC.

4.3. Jalankan Layanan GlusterFS:

Jalankan layanan GlusterFS di setiap PC:

`sudo service glusterfs-server start`

4.3.1 Buat Volume GlusterFS:

Buat volume GlusterFS:

`sudo gluster volume create all replica 3`

4.3.2 Mount Volume GlusterFS:

Mount volume GlusterFS di setiap PC:

`sudo mount -t glusterfs localhost:/all /mnt/glusterfs`

4.3.3 Skrip Otomatisasi dengan Paramiko:

Buat skrip Python menggunakan Paramiko untuk:

Mengkonfigurasi GlusterFS di semua PC.

Mengecek status volume GlusterFS di semua PC

```
import paramiko
from glusterfs import GlusterFS

class RemoteConnection:
    def __init__(self, ip, username, password):
        self.ip = ip
        self.username = username
        self.password = password
        self.ssh_client = paramiko.SSHClient()
        self.ssh_client.set_missing_host_key_policy(paramiko.AutoAddPol

    def connect(self):
        try:
            self.ssh_client.connect(self.ip, username=self.username, pa
            return True
        except paramiko.AuthenticationException:
            return False

    def execute_command(self, command):
        stdin, stdout, stderr = self.ssh_client.exec_command(command)
        output = stdout.read().decode('utf-8')
        return output.strip()

    def close(self):
        self.ssh_client.close()
```

Gambar 5. Kelas RemoteConnection

Kode tersebut terdiri dari dua tahap. Tahap pertama melibatkan impor modul seperti `paramiko` untuk koneksi SSH dan `glusterfs` untuk berinteraksi dengan sistem file GlusterFS.

Tahap kedua adalah definisi kelas `RemoteConnection`. Kelas ini digunakan untuk mengelola koneksi SSH ke sebuah PC. Metode yang didefinisikan meliputi `__init__()` untuk inisialisasi, `connect()` untuk melakukan koneksi SSH, `execute_command(command)` untuk menjalankan perintah di PC jarak jauh, dan `close()` untuk menutup koneksi SSH setelah selesai digunakan.

```
def mount_volume(pc, volume_name, mount_point):
    """
    Mount volume GlusterFS pada PC.

    Args:
        pc: Objek RemoteConnection yang mewakili PC.
        volume_name: Nama volume GlusterFS.
        mount_point: Titik mount di PC.

    Returns:
        True jika mount berhasil, False jika gagal.
    """
    try:
        # Buat objek GlusterFS
        glusterfs = GlusterFS(pc.ip)

        # Periksa apakah volume ada
        if volume_name not in glusterfs.list_volumes():
            print(f"Volume {volume_name} tidak ditemukan di {pc.ip}")
            return False

        # Mount volume
        glusterfs.mount(volume_name, mount_point)
        print(f"Volume {volume_name} berhasil di-mount di {pc.ip}")
        return True

    except Exception as e:
        print(f"Gagal mount volume {volume_name} di {pc.ip}: {e}")
        return False
```

Gambar 6. Mount_volume

Untuk melakukan mount volume GlusterFS pada sebuah PC yang ditentukan, kamu perlu menggunakan tiga argumen: objek PC (RemoteConnection), nama volume, dan titik

mount. Proses mount akan mencoba menghubungkan volume GlusterFS dengan sistem file PC. Hasilnya akan dicetak sebagai pesan keberhasilan atau kegagalan, dan nilai True atau False akan dikembalikan sesuai hasilnya.

```
def main():
    pc_a = RemoteConnection('192.168.9.178', 'slave', 'slave')
    pc_b = RemoteConnection('192.168.9.244', 'slain', 'slain')

    while True:
        print("*****Sistem Kontrol PC A dan PC B*****")
        print()

        choice = input("""
        A: Check Status PC A
        B: Check Status PC B
        C: Input Angka ke DFS
        D: Mount Volume GlusterFS
        Q: Quit
        Please enter your choice: """)

        if choice == 'A':
            if pc_a.connect():
                status_pc_a = pc_a.execute_command('uname -a')
                pc_a.close()
                print(f"Status PC A: {status_pc_a}")
            else:
                print("Gagal Koneksi SSH")
        elif choice == 'B':
            if pc_b.connect():
                status_pc_b = pc_b.execute_command('uname -a')
                pc_b.close()
                print(f"Status PC B: {status_pc_b}")
            else:
                print("Gagal koneksi SSH: Salah satu PC tidak terhubung.")
        elif choice == 'C':
            # Jalankan hitung.sh di PC A dan PC B
            jalankan_hitung_sh(pc_a)

            # Tunggu proses hitung.sh selesai
            tunggu_proses_selesai(pc_a)

            # Baca hasil perhitungan dari file .txt
            hasil_perhitungan = baca_nilai_dari_file()

            # Tampilkan hasil perhitungan
            print(f"Hasil perhitungan: {hasil_perhitungan}")
        elif choice == 'D':
            # Mount volume "all" di /mnt/glusterfs pada PC A dan PC B
            mount_volume(pc_a, "all", "/mnt/glusterfs")
            mount_volume(pc_b, "all", "/mnt/glusterfs")
        elif choice == 'Q':
            print("Keluar dari program.")
            break
        else:
            print("Pilihan tidak Valid")
```

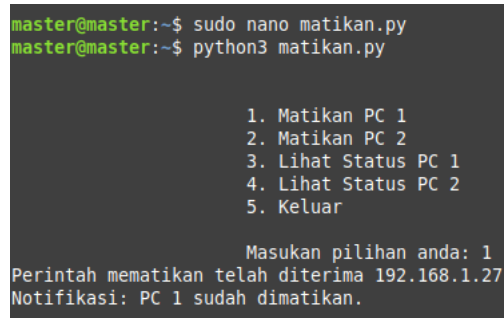
Gambar 7. main

Fungsi `main()` adalah inti dari program yang bertanggung jawab untuk menjalankan seluruh proses. Pertama-tama, program membuat objek `RemoteConnection` untuk PC A dan PC B dengan informasi koneksi yang diperlukan. Selanjutnya, program menampilkan menu pilihan kepada pengguna, termasuk opsi untuk memeriksa status PC, memasukkan angka ke DFS, me-mount volume GlusterFS, dan keluar dari program. Setelah menerima input pengguna, program akan menjalankan tindakan yang sesuai dengan pilihan mereka. Misalnya, untuk memeriksa status PC, program akan mencoba koneksi SSH dan menampilkan informasi sistem jika berhasil atau pesan kesalahan jika tidak. Opsi lainnya seperti memasukkan angka ke DFS dan me-mount volume GlusterFS juga akan diimplementasikan sesuai dengan kebutuhan. Jika pengguna memilih opsi yang tidak valid,

program akan memberikan pesan yang sesuai sebelum kembali menampilkan menu. Dengan demikian, fungsi `main()` mengatur alur program dan interaksi dengan pengguna secara efisien.

4.4 Analisis Hasil

Setelah melalui beberapa tahapan selanjutnya menganalisis hasil atau output dari code yang sudah dibuat tadi



```
master@master:~$ sudo nano matikan.py
master@master:~$ python3 matikan.py

1. Matikan PC 1
2. Matikan PC 2
3. Lihat Status PC 1
4. Lihat Status PC 2
5. Keluar

Masukan pilihan anda: 1
Perintah mematikan telah diterima 192.168.1.27
Notifikasi: PC 1 sudah dimatikan.
```

Gambar 8. Hasil

Ketika menjalankan program MeProgram, akan diminta untuk memilih antara opsi A, B, C, D, atau keluar (Q). Jika memilih opsi A, program akan mencoba untuk memeriksa status PC yang dipilih, yaitu PC A atau B. Jika koneksi berhasil, program akan menampilkan status dari PC yang dipilih, seperti contoh berikut: "Status PC A: Linux raspberrypi 5.10.103-v7l+ #1358 SMP Sel Tue Feb 22 18:10:14 GMT 2022 armv7l GNU/Linux." Namun, jika koneksi gagal, program akan memberikan pesan kesalahan yang menyatakan kegagalan koneksi SSH dengan salah satu dari dua kemungkinan pesan: "Gagal Koneksi SSH" atau "Gagal koneksi SSH: Salah satu PC tidak terhubung." Pesan-pesan ini membantu memberi pemahaman tentang hasil dari percobaan koneksi yang dilakukan oleh program.

Operasi mount untuk PC A dan PC B telah berhasil dilakukan. Volume all telah berhasil di-mount pada alamat IP 192.168.9.178 dan 192.168.9.244 secara sukses. Namun, jika terjadi kesalahan, seperti ketika volume tidak dapat ditemukan, akan muncul pesan yang menyatakan "Gagal mount volume all di 192.168.9.178: Volume tidak ditemukan". Dalam situasi tersebut, program akan menghentikan prosesnya dan menampilkan pesan "Keluar dari program". Dengan demikian, operasi mount akan diakhiri dan program berhenti berjalan.

KESIMPULAN

Dari hasil dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa penerapan GlusterFS sebagai sistem file pada platform Linux memiliki potensi yang signifikan untuk memenuhi permintaan akan penyimpanan data yang efisien dan handal dalam era digital saat ini. GlusterFS menyediakan solusi penyimpanan distribusi yang fleksibel dan dapat diperluas, sehingga cocok untuk digunakan baik dalam lingkungan perusahaan maupun individu.

Selain itu, integrasi Paramiko sebagai pustaka Python untuk implementasi protokol SSH menawarkan manfaat tambahan dalam pengelolaan sistem operasi Linux, termasuk dalam konteks pengelolaan penyimpanan distribusi. Kemampuan Paramiko dalam manajemen remote dapat meningkatkan efisiensi dan keterjangkauan dalam mengelola sistem penyimpanan distribusi yang menjalankan GlusterFS.

Dengan demikian, integrasi antara GlusterFS dan Paramiko memiliki potensi untuk meningkatkan kinerja, keandalan, dan manajemen penyimpanan distribusi pada platform Linux. Ini menawarkan solusi yang komprehensif dan terintegrasi untuk kebutuhan penyimpanan data yang semakin meningkat dalam lingkungan digital saat ini.

DAFTAR PUSTAKA

- [1] I. Yunianto and K. Adhiyarta, "Jurnal Review: Perbandingan Sistem Operasi Linux Dengan Sistem Operasi Windows," *JUPITER J. Comput. Inf. Technol.*, vol. 1, no. 1, pp. 1–7, 2020, doi: 10.53990/jupiter.v1i1.3.
- [2] Athallah and O. Gerry, "Sistem Operasi LINUX dan Pemanfaatannya dalam Sistem Informasi Manajemen," *Jurnal_PIK_Owen_Gerry_Athallah*, p. 5, 2022.
- [3] T. Yusnanto, S. Wahyudiono, and H. A. Wicaksono, "Analisa Kinerja Sistem Operasi Windows 10 Dengan Linux Mint Menggunakan Aplikasi Zxt Cam, Gnome System Monitor," *SENTRI J. Ris. Ilm.*, vol. 1, no. 2, pp. 288–296, 2022, doi: 10.55681/sentri.v1i2.210.
- [4] S. Purwantoro E.S.G.S, "Perbandingan Kinerja Clustered File System pada Cloud Storage menggunakan GlusterFS dan Ceph," *INOVTEK Polbeng - Seri Inform.*, vol. 7, no. 2, p. 319, 2022, doi: 10.35314/isi.v7i2.2753.
- [5] A. Setiawan and R. A. Rahmah, "PEMBUATAN FILE SYSTEM GLUSTERFS.pdf," vol. 3, no. 3, pp. 368 – 375, 2021.
- [6] M. Riasetiawan and N. Amien, "Random and Sequence Workload for Web-Scale Architecture for NFS, GlusterFS and MooseFS Performance Enhancement," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 3, pp. 449–457, 2022, doi: 10.14569/IJACSA.2022.0130354.
- [7] J. A. Ramadhan, A. Susilo, Y. Irawan, and A. Solehudin, "Dengan Pemrograman Python Berbasis Web (Studi Kasus : Smkn 3 Kota Bekasi)," vol. 7, no. 4, pp. 2756–2766, 2023.
- [8] R. U. Putri, J. Informatika, F. T. Industri, and U. I. Indonesia, "Forensik sistem file ext4 studi kasus file di sistem operasi linux," pp. 3–6, 2009.
- [9] W. Umar, "Pengembangan Perangkat Lunak Distribusi Gnu/Linux Dengan Fitur Khas Gorontalo," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 6, no. 1, p. 11, 2015, doi: 10.24176/simet.v6i1.231.
- [10] S. Bokhari, "The Linux Thin-Client-Concept," *Comput. Pract.*, pp. 75–80, 1999.
- [11] S. Farizy and T. Gunawan, "Analisa Performa Sistem Berkas Ext4 Pada Kondisi Tervirtualisasi," *Sainstech J. Penelit. dan Pengkaj. Sains dan Teknol.*, vol. 30, no. 1, pp. 70–75, 2020, doi: 10.37277/stch.v30i1.726.
- [12] J. Haritsah, A. Budiyo, and A. Widjajarto, "Analisis Proses Sistem Untuk Implementasi Interplanetary File System (IPFS) Pada Smart Contract Ethereum," *e-Proceeding Eng.*, vol. 6, no. 2, pp. 7890–7898, 2019.
- [13] M. T. Jones, "Anatomy of ext4 Get to know the fourth extended file system," pp. 1–6, 2009, [Online]. Available: <https://www.ibm.com/developerworks/library/l-anatomy-ext4/l-anatomy-ext4-pdf.pdf>
- [14] D. Xiao, C. Zhang, and X. Li, "The Performance Analysis of GlusterFS In Virtual Storage," no. Ameii, pp. 199–203, 2015, doi: 10.2991/ameii-15.2015.36.
- [15] L. G. Mauboy and T. Wellem, "Studi Perbandingan Library Untuk Implementasi Network Automation Menggunakan Paramiko Dan Netmiko Pada Router Mikrotik," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 4, p. 790, 2022, doi: 10.30865/jurikom.v9i4.4420.
- [16] D. A. Mutiara, K. N. Isnaini, and D. Suhartono, "Network Programmability for Network Issue Using Paramiko Library," *J. Tek. Inform.*, vol. 4, no. 4, pp. 751–758, 2023, doi: 10.52436/1.jutif.2023.4.4.691.