

DOKUMEN DESAIN



PyNetScanner

Network Monitoring Tool

Richard Firmansyah

NPM: 714240047 | Kelas: 2C-D4-TI

Dosen Pengampu:

M. Yusril Helmi Setyawan, S.Kom., M.Kom., SFPC.

1. Deskripsi Aplikasi

Pada era digital seperti sekarang, jaringan komputer sudah menjadi bagian penting dalam kehidupan sehari-hari. Mulai dari browsing internet, video call, hingga transfer data antar perangkat, semuanya membutuhkan koneksi jaringan yang stabil. Namun terkadang kita mengalami masalah seperti koneksi lambat, tidak bisa mengakses website tertentu, atau perangkat tidak terdeteksi di jaringan.

PyNetScanner hadir sebagai solusi untuk membantu mendiagnosis dan memahami kondisi jaringan. Aplikasi ini saya kembangkan menggunakan bahasa pemrograman Python karena Python memiliki library yang lengkap untuk network programming dan mudah dipelajari.

Untuk tampilan antarmuka, saya menggunakan CustomTkinter yang merupakan pengembangan dari Tkinter standard. CustomTkinter memberikan tampilan yang lebih modern dengan dukungan tema dark mode yang nyaman untuk mata.

1.1 Latar Belakang Pembuatan

Ide pembuatan aplikasi ini muncul ketika saya sering mengalami masalah koneksi di rumah. Setiap kali internet lambat, saya harus membuka Command Prompt dan mengetik perintah ping secara manual. Hal ini cukup merepotkan, apalagi kalau ingin mengecek banyak hal sekaligus seperti port yang terbuka atau rute paket data.

Dari pengalaman tersebut, saya berpikir untuk membuat aplikasi yang bisa melakukan semua fungsi diagnostik jaringan dalam satu tempat dengan tampilan yang user-friendly.

1.2 Tujuan Aplikasi

Aplikasi PyNetScanner dibuat dengan beberapa tujuan utama:

- A. Mempermudah proses troubleshooting jaringan tanpa perlu menghafal command
- B. Menyediakan informasi jaringan yang lengkap dalam satu aplikasi
- C. Mengimplementasikan konsep Network Programming yang dipelajari di perkuliahan
- D. Membuat tools yang bisa digunakan untuk keperluan sehari-hari

1.3 Konsep Network Programming

Dalam pengembangan aplikasi ini, saya mengimplementasikan beberapa konsep penting dalam Network Programming:

a) *Socket Programming*

Socket adalah endpoint untuk komunikasi dua arah antar program dalam jaringan. Di aplikasi ini, socket digunakan untuk fitur Port Scanner dan Service Detection. Saya menggunakan TCP socket untuk mengecek apakah sebuah port terbuka atau tidak dengan cara mencoba membuat koneksi ke port tersebut.

b) *ICMP Protocol*

Internet Control Message Protocol (ICMP) digunakan untuk fitur Ping. Protokol ini bekerja dengan mengirimkan paket Echo Request ke target dan menunggu Echo Reply. Dari sini kita bisa mengukur waktu response (latency) dan mengetahui apakah host bisa dijangkau atau tidak.

c) *DNS Resolution*

Domain Name System (DNS) berfungsi untuk mengkonversi nama domain menjadi alamat IP. Misalnya ketika kita mengetik ulbi.ac.id, DNS akan menerjemahkannya menjadi IP seperti 104.21.43.145. Fitur DNS Lookup menggunakan konsep ini.

d) *Multi-threading*

Karena operasi jaringan bisa memakan waktu lama (misalnya menunggu timeout), saya menggunakan threading agar aplikasi tidak freeze saat menjalankan operasi. Setiap operasi jaringan dijalankan di thread terpisah sehingga GUI tetap responsif.

2. Fitur dan Penjelasan

PyNetScanner dilengkapi dengan 13 fitur yang bisa digunakan untuk berbagai keperluan diagnostik jaringan. Berikut penjelasan detail masing-masing fitur:

2.1 Ping Tester

Fitur ini digunakan untuk mengecek apakah sebuah host (komputer/server) bisa dijangkau melalui jaringan. Cara kerjanya mirip dengan command "ping" di Command Prompt.

Ketika kita melakukan ping, aplikasi akan mengirim paket ICMP Echo Request ke target. Jika target aktif dan bisa dijangkau, maka akan mengirim balik Echo Reply. Dari proses ini kita bisa mendapatkan informasi:

- A. Packets Sent - Jumlah paket yang dikirim
- B. Packets Received - Jumlah paket yang diterima kembali
- C. Packet Loss - Persentase paket yang hilang
- D. Latency (Min/Avg/Max) - Waktu tempuh paket dalam milidetik

Contoh penggunaan: Untuk mengecek apakah koneksi internet berfungsi, kita bisa ping ke google.com atau 8.8.8.8 (DNS Google).

2.2 Port Scanner

Port Scanner berfungsi untuk memindai port-port yang terbuka pada sebuah host. Setiap layanan jaringan berjalan di port tertentu, misalnya HTTP di port 80, HTTPS di port 443, SSH di port 22, dan seterusnya.

Aplikasi ini menggunakan metode TCP Connect Scan, yaitu mencoba membuat koneksi TCP ke setiap port yang dicek. Jika koneksi berhasil (port merespons), berarti port

tersebut terbuka. Jika timeout atau ditolak, berarti port tertutup.

Fitur ini berguna untuk:

- A. Mengecek layanan apa saja yang berjalan di server
- B. Security audit untuk menemukan port yang tidak seharusnya terbuka
- C. Troubleshooting ketika aplikasi tidak bisa terhubung ke server

2.3 DNS Lookup

DNS Lookup mengkonversi nama domain menjadi alamat IP. Fitur ini menggunakan resolver DNS sistem untuk melakukan query A record.

Contoh: Ketika kita lookup "ulbi.ac.id", aplikasi akan menampilkan alamat IP 104.21.43.145 dan 172.67.180.76 (Cloudflare).

2.4 Reverse DNS

Kebalikan dari DNS Lookup, Reverse DNS mencari nama domain dari alamat IP menggunakan PTR record. Tidak semua IP memiliki reverse DNS, tergantung konfigurasi dari pemilik IP tersebut.

2.5 Network Info

Fitur ini menampilkan informasi jaringan lokal komputer, termasuk:

- A. Hostname - Nama komputer
- B. Local IP - Alamat IP komputer di jaringan lokal
- C. Default Gateway - Alamat router

- D. Network Interfaces - Daftar adapter jaringan beserta MAC address

Informasi ini berguna untuk troubleshooting masalah konfigurasi jaringan di komputer.

2.6 Service Detection

Service Detection atau Banner Grabbing adalah teknik untuk mengetahui jenis layanan yang berjalan di sebuah port. Caranya dengan membuat koneksi ke port dan membaca banner/response awal yang dikirim server.

Misalnya ketika connect ke port 22, server SSH biasanya akan mengirim banner seperti "SSH-2.0-OpenSSH_8.0". Dari sini kita bisa tahu versi SSH yang digunakan.

2.7 Traceroute

Traceroute melacak jalur yang dilalui paket data dari komputer kita menuju target. Cara kerjanya dengan mengirim paket dengan TTL (Time To Live) bertahap, mulai dari 1, 2, 3, dan seterusnya.

Setiap router yang dilewati akan mengurangi TTL sebesar 1. Ketika TTL habis (menjadi 0), router akan mengirim pesan ICMP Time Exceeded beserta alamat IP-nya. Dari sini kita bisa melihat semua router (hop) yang dilewati.

Traceroute berguna untuk:

- A. Mengetahui jalur routing ke target
- B. Menemukan titik bottleneck atau masalah di jaringan
- C. Menganalisis latency di setiap hop

2.8 WHOIS Lookup

WHOIS adalah protokol untuk mencari informasi kepemilikan domain atau alamat IP. Dari WHOIS kita bisa mengetahui:

- A. Nama pemilik/pendaftar domain
- B. Tanggal registrasi dan expiry
- C. Name server yang digunakan
- D. Informasi kontak (jika tidak di-private)

Aplikasi ini menggunakan library python-whois untuk melakukan query WHOIS.

2.9 Speed Test

Fitur Speed Test mengukur kecepatan download internet. Cara kerjanya dengan mengunduh file dari server dan menghitung berapa lama waktu yang dibutuhkan.

Rumus perhitungannya sederhana:

$$\text{Kecepatan (Mbps)} = \frac{\text{Ukuran File (bit)}}{\text{Waktu Download (detik)}} / 1000000$$

2.10 Host Discovery

Host Discovery memindai jaringan lokal untuk menemukan perangkat yang aktif. Cara kerjanya dengan melakukan ping sweep ke range IP di subnet yang sama.

Misalnya jika komputer kita memiliki IP 192.168.1.10, maka aplikasi akan memindai 192.168.1.1 sampai 192.168.1.254 untuk menemukan host yang merespons.

2.11 MAC Vendor Lookup

Setiap perangkat jaringan memiliki MAC address unik. 3 byte pertama dari MAC address disebut OUI (Organizationally

Unique Identifier) yang menunjukkan vendor/produsen perangkat.

Contoh: MAC address yang diawali dengan 00:1A:2B kemungkinan adalah perangkat dari vendor tertentu. Fitur ini membantu mengidentifikasi jenis perangkat di jaringan.

2.12 GeolP Location

GeolP menggunakan layanan API eksternal untuk melacak lokasi geografis dari alamat IP publik. Informasi yang bisa didapat antara lain negara, kota, ISP, dan koordinat perkiraan.

Fitur ini berguna untuk mengetahui dari mana asal traffic atau mengecek lokasi server.

2.13 Activity Logging

Semua aktivitas yang dilakukan di aplikasi dicatat dalam log. Log ini bisa di-export ke file teks untuk dokumentasi. Setiap entry log berisi timestamp, jenis aksi, hasil, dan detail tambahan.

3. Workflow Aplikasi

Berikut adalah penjelasan detail alur kerja aplikasi PyNetScanner dari awal hingga akhir:

3.1 Tahap Input

Langkah pertama adalah user memasukkan target yang ingin diperiksa. Tergantung fitur yang dipilih, input bisa berupa:

- A. Alamat IP (contoh: 192.168.1.1, 8.8.8.8)
- B. Hostname/Domain (contoh: google.com, ulbi.ac.id)
- C. MAC Address (untuk MAC Vendor Lookup)
- D. Range port (untuk Port Scanner)

Aplikasi kemudian memvalidasi input untuk memastikan formatnya benar sebelum melanjutkan ke tahap berikutnya.

3.2 Tahap Processing

Setelah input divalidasi, aplikasi mempersiapkan operasi yang akan dijalankan. Pada tahap ini:

- A. Membuat thread baru agar GUI tidak freeze
- B. Menyiapkan parameter sesuai fitur yang dipilih
- C. Menampilkan status "Processing..." di GUI
- D. Mengupdate progress bar jika ada

Penggunaan thread terpisah sangat penting karena operasi jaringan bisa memakan waktu lama, terutama jika terjadi timeout.

3.3 Tahap Network Operation

Ini adalah tahap inti dimana operasi jaringan sebenarnya dijalankan. Beberapa metode yang digunakan:

Socket Connection

Untuk fitur Port Scanner dan Service Detection, aplikasi menggunakan socket TCP untuk membuat koneksi ke target. Contoh kode sederhana:

```
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.settimeout(1.0)
result = sock.connect_ex((host, port))
if result == 0:
    print("Port terbuka")
sock.close()
```

Subprocess

Untuk fitur Ping dan Traceroute, aplikasi menjalankan command sistem menggunakan subprocess. Ini karena Python standard tidak memiliki akses langsung ke raw socket untuk ICMP.

```
import subprocess
result = subprocess.run(
    ["ping", "-n", "4", "google.com"],
    capture_output=True, text=True
)
print(result.stdout)
```

HTTP Request

Untuk fitur Speed Test dan Geoloc, aplikasi menggunakan library requests untuk melakukan HTTP request ke API eksternal.

3.4 Tahap Output

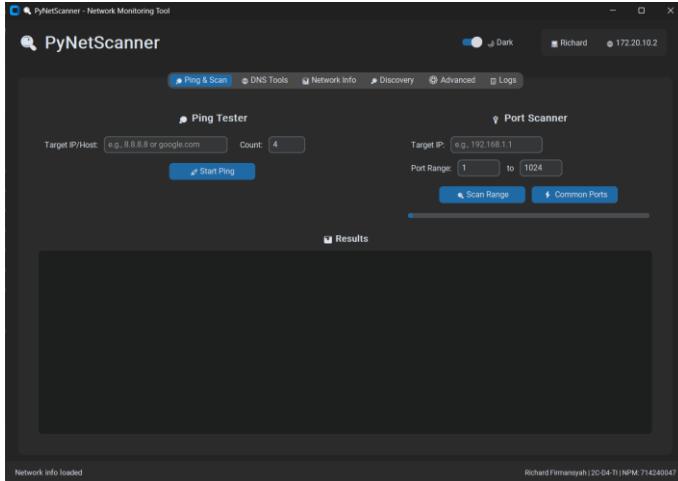
Setelah operasi selesai, hasil ditampilkan di area Results pada GUI. Format output disesuaikan dengan jenis fitur, misalnya tabel untuk Port Scanner atau list untuk Traceroute.

Selain ditampilkan di GUI, hasil juga dicatat ke Log Activity sehingga bisa direview atau di-export kemudian.

4. Tampilan UI Aplikasi

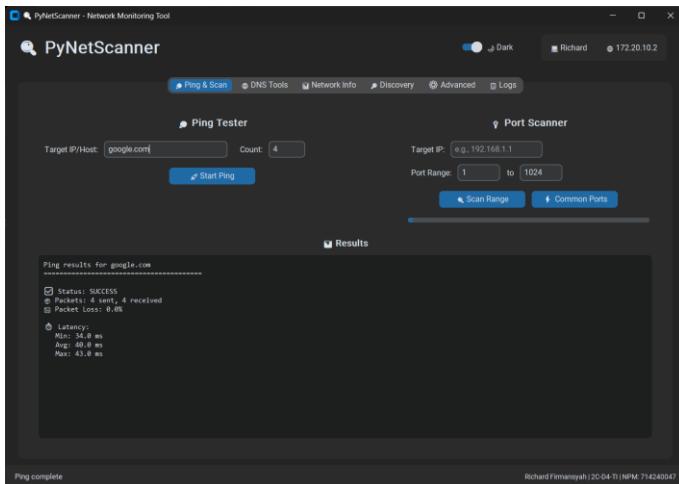
Aplikasi PyNetScanner menggunakan desain tab untuk mengorganisasi berbagai fitur. Berikut tampilan masing-masing bagian:

4.1 Tab Ping & Scan



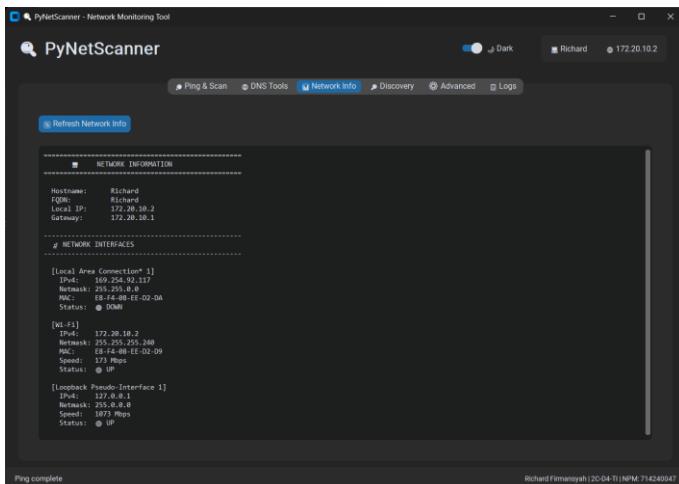
Tab utama berisi fitur Ping Tester dan Port Scanner. Di bagian atas terdapat field untuk memasukkan target IP/hostname. User bisa mengatur jumlah ping dan range port yang diinginkan.

4.2 Hasil Ping



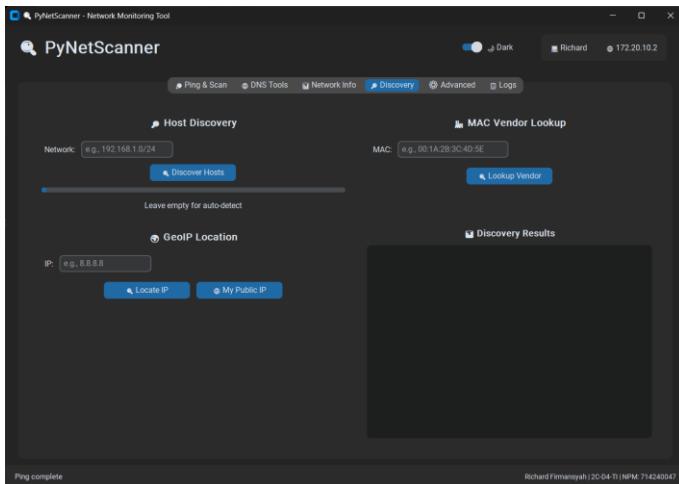
Hasil ping menampilkan statistik lengkap termasuk jumlah paket yang dikirim dan diterima, persentase packet loss, serta latency minimum, rata-rata, dan maksimum.

4.3 Network & DNS Tools



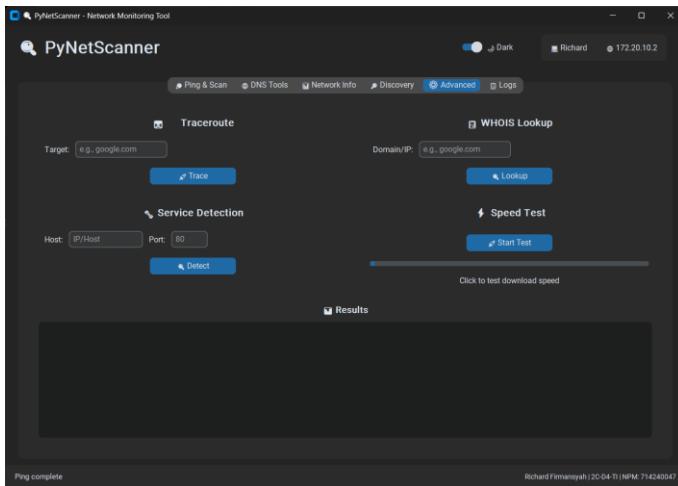
Tab ini menampilkan informasi jaringan lokal dan menyediakan tools DNS. User bisa melihat hostname, IP address, gateway, dan interface yang tersedia di komputer.

4.4 Discovery Tools



Tab Discovery berisi fitur untuk scanning jaringan lokal, lookup vendor MAC, dan pelacakan lokasi IP. Fitur Host Discovery sangat berguna untuk mengetahui perangkat apa saja yang terhubung ke jaringan yang sama.

4.5 Advanced Tools



Tab Advanced menyediakan tools lanjutan seperti Traceroute, WHOIS, Service Detection, dan Speed Test. Fitur-fitur ini lebih advanced dan biasanya digunakan untuk troubleshooting yang lebih mendalam.

5. Kesimpulan

5.1 Hasil Pengujian

Pengujian aplikasi dilakukan dengan mencoba semua fitur yang tersedia menggunakan berbagai target. Berikut ringkasan hasil pengujian:

| Test Case | Hasil |
|-----------------------|---|
| Ping ke google.com | Berhasil - Min 21ms, Max 27ms, Avg 23ms, 0% packet loss |
| Ping ke ulbi.ac.id | Berhasil - Min 188ms, Max 314ms, Avg 270ms, 0% packet loss |
| Ping ke 8.8.8.8 | Berhasil - Min 19ms, Max 37ms, Avg 25ms, 0% packet loss |
| Port Scan ulbi.ac.id | Berhasil - Port 80 (HTTP) dan 443 (HTTPS) terbuka |
| DNS Lookup ulbi.ac.id | Berhasil - Resolved ke IP 104.21.43.145 dan 172.67.180.76 |
| Traceroute ulbi.ac.id | Berhasil - Menampilkan 6 hop, berakhir di 104.21.43.145 |
| WHOIS ulbi.ac.id | Berhasil - Domain terdaftar di Cloudflare |
| Speed Test | Berhasil - Kecepatan download terukur sesuai koneksi |
| Host Discovery | Berhasil - Menemukan perangkat aktif di jaringan 192.168.100.x |
| GeolP Lookup | Berhasil - ulbi.ac.id terdeteksi lokasi di Indonesia (Cloudflare) |
| MAC Vendor Lookup | Berhasil - Mengidentifikasi vendor dari MAC address |
| Network Info | Berhasil - Gateway 192.168.100.1, IP lokal terdeteksi |

Dari pengujian di atas, semua fitur berfungsi dengan baik sesuai yang diharapkan. Beberapa catatan:

- Akurasi hasil tergantung pada kondisi jaringan saat pengujian
- Speed Test mungkin berbeda dengan hasil dari website speed test karena perbedaan server
- GeolP tidak selalu 100% akurat, terutama untuk IP dari ISP lokal

5.2 Manfaat di Dunia Nyata

Aplikasi ini memiliki banyak manfaat praktis yang bisa digunakan dalam berbagai situasi:

Untuk Administrator Jaringan

- Mempercepat proses troubleshooting dengan tools yang terintegrasi
- Monitoring kondisi jaringan secara berkala
- Inventory perangkat yang terhubung ke jaringan

Untuk Security Analyst

- Scanning port untuk menemukan layanan yang tidak seharusnya berjalan
- Service detection untuk mengidentifikasi versi software yang mungkin vulnerable
- Tracking IP mencurigakan dengan GeolP

Untuk Pengguna Umum

- Mengecek apakah internet berfungsi dengan baik
- Mengetahui kecepatan internet yang sebenarnya
- Memahami kondisi jaringan di rumah

5.3 Pengembangan Kedepan

Beberapa fitur yang bisa ditambahkan di masa depan untuk menyempurnakan aplikasi ini:

1. Fitur scheduling untuk monitoring otomatis
2. Export hasil ke format PDF atau Excel
3. Notifikasi alert ketika koneksi bermasalah
4. Interface berbasis web untuk akses remote
5. Integrasi dengan database untuk menyimpan history

Dengan pengembangan berkelanjutan, aplikasi ini diharapkan bisa menjadi tools yang semakin berguna dan lengkap untuk keperluan monitoring jaringan.