Jurnal Teknik Informatika dan Teknologi Informasi
Volume. 5 Nomor. 3 Desember 2025
E-ISSN .: 2827-9387, P-ISSN .: 2827-9379, Hal 305-320
DOI: https://doi.org/10.55606/jutiti.v5i3.6313
Tersedia: https://journalshub.org/index.php/jutiti

Simulasi Jaringan untuk Sistem Terdistribusi Website "Simple Restaurant App" dengan GNS3

Rafel Sutra Dharma [1*], Tri Sutisno [2], Manatap Dolok Lauro [3], Ari Wijaya [4] , George Wielianto[5] , Michael [6] , Firzi Ilham Bagusti [7]
[1-7] Prodi Teknik Informatika, Universitas Tarumanagara, Jakarta Barat, Indonesia
**Penulis Korespondensi: rafel.sutradharma@gmail.com [1]*

Abstract. *Distributed systems, integral to contemporary computing, involve the coordinated functioning of autonomous nodes across interconnected networks. Architectural models like client-server and peer-to-peer configurations define their structure, presenting both advantages and challenges. Challenges stem from the intricacies of managing geographically dispersed nodes, encompassing issues of data consistency, fault tolerance, scalability, and security. Overcoming these hurdles demands advanced algorithms and protocols, especially for achieving consistency and resilience in the face of failures. Scalability is a critical consideration, and security concerns add complexity to their design. Current trends in distributed systems, such as edge computing, serverless architectures, and blockchain technologies, aim to address these challenges and enhance system capabilities. Edge computing optimizes proximity to data sources, serverless architectures streamline resource utilization, and blockchain offers decentralized and tamper-resistant solutions. Understanding these architectures, addressing challenges, and embracing emerging trends are pivotal for constructing robust and efficient distributed systems that align with the demands of the interconnected digital landscape.*

Keywords: *Data Consistency; Distributed Systems; Fault Tolerance; Scalability; System Security.*

Abstrak. Sistem terdistribusi, yang merupakan bagian integral dari komputasi kontemporer, melibatkan fungsi terkoordinasi dari node-node otonom di seluruh jaringan yang saling terhubung. Model arsitektur seperti konfigurasi klien-server dan peer-to-peer mendefinisikan strukturnya, menghadirkan keuntungan sekaligus tantangan. Tantangan bermula dari kerumitan pengelolaan node yang tersebar secara geografis, yang mencakup masalah konsistensi data, toleransi kesalahan, skalabilitas, dan keamanan. Mengatasi hambatan ini membutuhkan algoritma dan protokol yang canggih, terutama untuk mencapai konsistensi dan ketahanan dalam menghadapi kegagalan. Skalabilitas merupakan pertimbangan penting, dan masalah keamanan menambah kompleksitas pada desainnya. Tren terkini dalam sistem terdistribusi, seperti komputasi tepi, arsitektur tanpa server, dan teknologi blockchain, bertujuan untuk mengatasi tantangan ini dan meningkatkan kapabilitas sistem. Komputasi tepi mengoptimalkan kedekatan dengan sumber data, arsitektur tanpa server menyederhanakan pemanfaatan sumber daya, dan blockchain menawarkan solusi terdesentralisasi dan tahan gangguan. Memahami arsitektur ini, mengatasi tantangan, dan merangkul tren yang sedang berkembang sangat penting untuk membangun sistem terdistribusi yang tangguh dan efisien yang selaras dengan tuntutan lanskap digital yang saling terhubung.

Kata kunci: Fault tolerance; Keamanan sistem; Konsistensi data; Sistem terdistribusi; Skalabilitas.

1. LATAR BELAKANG

Sistem jaringan merupakan suatu kerangka kerja yang memfasilitasi perangkat atau komputer untuk terhubung satu sama lain guna berbagi informasi, sumber daya, dan layanan. Komponen utama dalam sistem jaringan komputer melibatkan perangkat seperti server, kartu jaringan, kabel dan konektor, hub, switch, serta bridge. Lebih dari sekadar menyediakan konektivitas, sistem jaringan mampu beroperasi secara virtual, dapat diskalakan sesuai permintaan, dan menawarkan keamanan data. Fungsi sistem jaringan mencakup kemudahan pengiriman data antar perangkat dengan kecepatan tinggi, kemungkinan komunikasi langsung atau tidak langsung antar pengguna, kolaborasi efisien, akses cepat terhadap informasi, dan penyediaan fitur keamanan seperti firewall.

Naskah Masuk: 16 Juni 2025; Revisi: 05 Agustus 2025; Diterima: 23 September 2025; Tersedia: 04 Desember 2025.

Penelitian ini bertujuan untuk mensimulasikan sistem jaringan yang menghubungkan Web Client, Web Server, dan Cloud Server melalui perangkat seperti router, switch, firewall, dan NAT yang telah dikonfigurasi agar terhubung secara terintegrasi. Simulasi ini bertujuan meningkatkan efisiensi operasional, memberikan akses fleksibel terhadap aplikasi dan layanan, serta memastikan keamanan data. Penggunaan Cloud SaaS diusulkan sebagai solusi untuk meningkatkan efisiensi, aksesibilitas, dan keamanan. SaaS memungkinkan manajemen data yang efisien, akses aplikasi dari berbagai perangkat dengan koneksi internet, dan sistem penyimpanan terhubung dengan cloud, yang memberikan keakuratan dan efisiensi waktu dalam manajemen data.

Pembuatan jaringan untuk sistem terdistribusi website "Simple Restaurant App" ini melibatkan beberapa teknologi dan aplikasi, dengan GNS3 sebagai software utama untuk simulasi jaringan komputer. GNS3 memfasilitasi pembuatan jaringan yang realistis dengan berbagai jenis perangkat seperti firewall dan perangkat Cisco.

2. KAJIAN TEORITIS

Jaringan dan Keamanan Komputer

Jaringan komputer, sebagai suatu rangkaian komputer yang saling terhubung, memungkinkan pertukaran data dan berbagi sumber daya melalui penggunaan protokol komunikasi baik pada teknologi fisik maupun nirkabel. Dengan adanya jaringan ini, pengguna dapat dengan mudah berinteraksi dan berkomunikasi antar komputer, menggunakan perangkat lunak, serta menjalankan aplikasi.

Implementasi keamanan jaringan pada komputer dilakukan dengan tujuan melindungi sistem dari berbagai ancaman seperti kebocoran data, virus, dan malware. Fokus utama keamanan jaringan adalah menciptakan lingkungan yang terbebas dari potensi ancaman yang dapat menimbulkan kerusakan atau peristiwa tidak diinginkan, seperti kebocoran data. Dalam konteks ini, firewall menjadi salah satu teknologi keamanan yang efektif dengan kemampuannya memantau data secara selektif dalam jaringan, mencegah serangan virus dan malware, serta mengatur akses yang tidak diinginkan.

Network address translation merupakan sebuah jaringan yang dapat menghubungkan jaringan pribadi terhubung ke jaringan umum. *Network address translation* ini memiliki fungsi untuk mengurangi penggandaan alamat IP pada jaringan.

Subnetting merupakan pembagian jaringan IP dengan membagi jaringan menjadi dua bagian atau lebih. *Subnetting* ini mempunyai kemampuan untuk mengurangi kepadatan jaringan untuk menghindari kemacetan jaringan dan juga dapat mengoptimalkan jaringan.

Transmission control protocol merupakan salah satu penunjang komunikasi yang banyak digunakan oleh orang-orang di internet dengan cara saling bertukar data antar komputer pada jaringan internet. fungsi utama dari TCP ini adalah melakukan *sharing* data yang dimiliki antara komputer yang terkoneksi.

Application layer merupakan nama dari salah satu bagian atau lapisan pada *Open System Interconnection* (OSI). *Application layer* ini merupakan *layer* yang unik dikarenakan hanya *application layer* inilah yang dapat berinteraksi langsung dengan *user* atau pengguna. fungsi dari *application layer* ini adalah menyediakan *interface* antara jaringan dan aplikasi, dan menampilkan gambar pada jaringan.

Aplikasi Terdistribusi

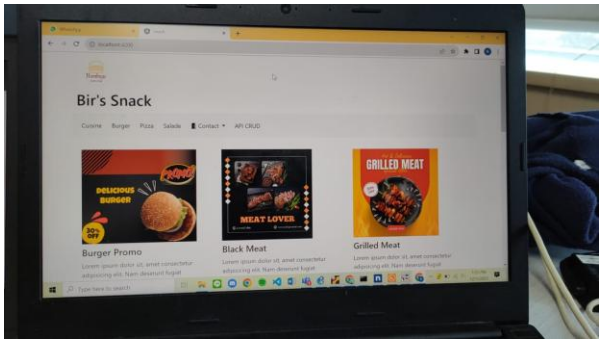
Aplikasi terdistribusi merupakan program perangkat lunak yang bergerak pada komputer yang berbeda pada suatu jaringan, aplikasi terdistribusi ini umumnya terbentuk dari atas aplikasi *backend* maupun *frontend*.

Aplikasi terdistribusi yang digunakan pada project ini adalah website “Simple Restaurant App” yang termasuk pada jenis arsitektur *cloud server* yang dibuat dengan menggunakan beberapa bahasa seperti, *Hypertext Markup Language* (HTML) yang banyak digunakan untuk membuat rangka dasar dan menyusun konten yang terdapat pada halaman web yang akan ditampilkan pada halaman *browser*. *Cascading Style Sheets* (CSS) merupakan bahasa pemrograman yang banyak digunakan untuk mengatur tata letak konten serta mendesain tampilan pada sebuah website yang dibuat dengan HTML.

Javascript merupakan bahasa pemrograman yang dapat dikatakan memiliki tingkatan yang tinggi dan banyak digunakan untuk membuat halaman web yang responsif atau interaktif.

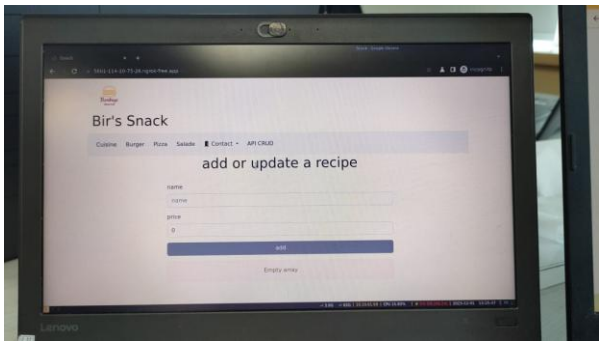
Aplikasi ini dapat digunakan oleh karyawan yang bekerja secara *work from office* (WFO), *work from home* (WFH), dengan cukup terhubung ke internet karyawan dapat mencatat pekerjaannya dimanapun dan kapanpun. aplikasi ini juga memiliki beberapa fitur seperti berikut:

Pada gambar 1 terdapat *Home Page* untuk menampilkan *navigation bar* dan API CRUD yang berguna untuk menambahkan menu makanan pada website ini. Terdapat tampilan pilihan menu dibawah *navigation bar* yang tersedia.



Gambar 1. Home Page

Pada gambar 2 terdapat halaman *add or update a recipe* untuk menambahkan menu ke dalam database. *User* dapat mengisi nama *menu* pada *textbox name* dan mengisi harga *menu* pada *textbox price.* Terdapat tombol *submit* pada bagian bawah *textbox price* yang berfungsi untuk menginput data dari *user* ke dalam database.



Gambar 2. Back-end

3. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimen dan simulasi jaringan dengan pendekatan Network Simulation-Based Analysis. Metode ini dipilih karena penelitian berfokus pada pembangunan, konfigurasi, dan pengujian jaringan terdistribusi yang menghubungkan Web Client, Web Server, dan Cloud Server melalui perangkat jaringan yang disimulasikan menggunakan GNS3. Tahapan penelitian dijelaskan sebagai berikut:

Studi Literatur

Tahap ini dilakukan untuk mengumpulkan teori dan referensi terkait sistem terdistribusi, jaringan komputer, keamanan jaringan, firewall Cisco ASA, konsep DMZ, NAT, subnetting, serta teknologi pendukung seperti GNS3, Angular, JSON Server, dan Ngrok. Literatur diperoleh dari jurnal penelitian, buku akademik, dokumentasi resmi, dan sumber ilmiah lainnya. Studi literatur digunakan sebagai dasar dalam merancang jaringan serta menentukan konfigurasi dan teknologi yang sesuai.

Analisis Kebutuhan Sistem

Tahap ini bertujuan untuk mengidentifikasi kebutuhan perangkat jaringan dan aplikasi yang akan disimulasikan. Analisis kebutuhan meliputi:

a) Penentuan perangkat virtual seperti router, switch, firewall, NAT, dan client.
b) Penetapan struktur jaringan yang terdiri dari tiga zona: inside, DMZ, dan outside.
c) Identifikasi kebutuhan akses pengguna Work From Office (WFO) dan Work From Home (WFH).
d) Kebutuhan server aplikasi berupa frontend Angular dan backend JSON Server.
e) Kebutuhan keamanan seperti pengaturan access-list, NAT, dan security level pada Cisco ASA.

Perancangan Arsitektur Jaringan

Pada tahap ini dirancang topologi jaringan yang akan digunakan pada simulasi. Perancangan mencakup:

a) Router R3 untuk jaringan internal kantor.
b) Router R1 sebagai penghubung ke jaringan luar dan Internet.
c) Cisco ASA sebagai firewall utama yang membagi zona inside–DMZ–outside.
d) Web Server yang ditempatkan pada zona DMZ menggunakan Angular dan JSON Server.
e) Web Client (PuppyLinux) sebagai klien WFH yang berada di zona outside.
f) Hasil perancangan divisualisasikan dalam diagram topologi untuk mempermudah proses implementasi.

Konfigurasi Jaringan pada GNS3

Tahap ini merupakan implementasi langsung dari rancangan arsitektur. Konfigurasi yang dilakukan meliputi:

a) Pemberian IP address pada seluruh perangkat (router, switch, dan client).
b) Pengaturan default gateway dan DNS pada setiap PC.
c) Pembuatan Static Route untuk menghubungkan seluruh subnet.
d) Konfigurasi Cisco ASA, meliputi:

a. Penetapan security level untuk inside, DMZ, dan outside.
b. Pembuatan NAT rules untuk menghubungkan antar zona.
c. Pembuatan access-list untuk menentukan hak akses ke WebServer.

e) Pengujian konektivitas menggunakan perintah seperti ping, show ip interface brief, dan show ip route.

Implementasi Server Aplikasi

Pada tahap ini dijalankan aplikasi web "Simple Restaurant App" sebagai Web Server pada zona DMZ. Implementasi meliputi:

a) Menjalankan backend menggunakan perintah npm run json-server.
b) Menjalankan frontend Angular menggunakan perintah npm run start.
c) Mengaktifkan Ngrok untuk menyediakan alamat URL yang dapat diakses dari jaringan outside (WFH).
d) Memastikan Web Server dapat diakses baik dari jaringan inside maupun outside.

Pengujian dan Validasi Sistem

Pengujian dilakukan untuk memastikan seluruh komponen jaringan berfungsi sesuai desain. Pengujian meliputi:

a) Konektivitas antar perangkat di jaringan internal (PC1, PC2, PC3, PC4).
b) Akses ke Internet dari perangkat internal dan WFH.
c) Akses ke Web Server dari jaringan inside dan outside melalui Ngrok.
d) Pengujian fungsi aplikasi seperti penambahan dan penghapusan data (CRUD).
e) Validasi keamanan firewall melalui pengujian access-list, NAT, dan routing.

Hasil pengujian dicatat untuk dianalisis lebih lanjut.

Dokumentasi dan Analisis Hasil

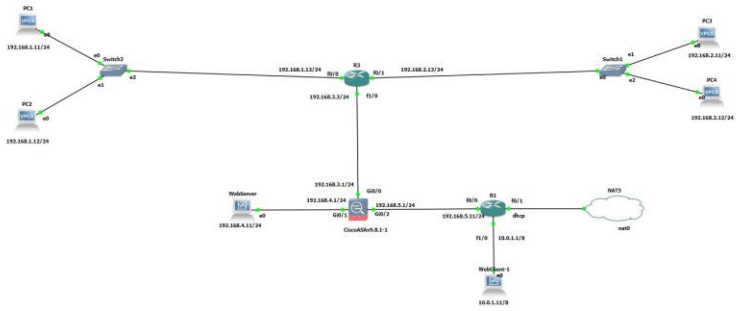
Seluruh proses konfigurasi, hasil pengujian, dan temuan selama simulasi didokumentasikan secara sistematis. Analisis dilakukan untuk menilai:

a) Keberhasilan implementasi jaringan terdistribusi.
b) Ketersediaan akses bagi pengguna WFO dan WFH.
c) Tingkat keamanan jaringan berdasarkan konfigurasi firewall dan NAT.
d) Potensi kelemahan serta rekomendasi perbaikan untuk pengembangan sistem di masa mendatang.

4. HASIL DAN PEMBAHASAN

Rancangan Jaringan

Pada Gambar 3 dijelaskan bahwa *Router 1(R1), NAT5* dan *WebClient(*Puppylinux*/WFH)* merupakan bagian outside. Dan *WebServer* merupakan bagian *DMZ(Demitrilisasi Zone).* DMZ yang kami gunakan merupakan *WebServer* yang kami buat memakai angular dan membuat link menggunakan ngrok. *WebServer* yang kami buat dapat diakses dari *PC1(internal), dan WebClient(PuppyLinux/WFH)(outside).* Dan yang terakhir ada *Router 3(R3), PC1, PC2, PC3, dan PC4* yang kita buat sebagai kantor internal.



Gambar 3. Layout jaringan

Konfigurasi Jaringan

Selanjutnya dilakukan *show* IP pada PC1(Internal). Pada gambar 4, dapat dilihat dari "sh ip" pada PC1(Internal) didapatkan IP 192.168.1.11/24 dengan *gateway* 192.168.1.13 dan DNS 8.8.8.8.

```
PC1>
PC1> sh ip

NAME        : PC1[1]
IP/MASK     : 192.168.1.11/24
GATEWAY     : 192.168.1.13
DNS         : 8.8.8.8
MAC         : 00:50:79:66:68:03
LPORT       : 10038
RHOST:PORT  : 127.0.0.1:10039
MTU:        : 1500
```

Gambar 4. Hasil dari "sh ip" PC1

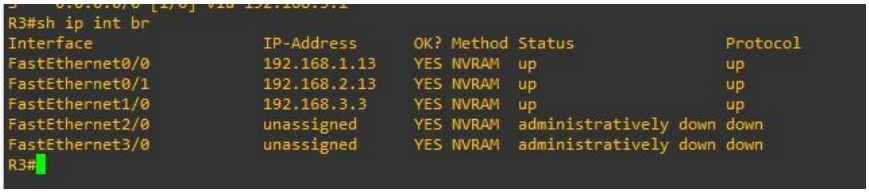
Selanjutnya dilakukan *show* IP pada PC3(Internal) Pada gambar 5, dapat dilihat dari "sh ip" pada PC3 didapatkan IP 192.168.2.11/24 dengan *gateway* 192.168.2.13 dan DNS 8.8.8.8.

```
PC3> sh ip

NAME        : PC3[1]
IP/MASK     : 192.168.2.11/24
GATEWAY     : 192.168.2.13
DNS         : 8.8.8.8
MAC         : 00:50:79:66:68:00
LPORT       : 10032
RHOST:PORT  : 127.0.0.1:10033
MTU:        : 1500
```

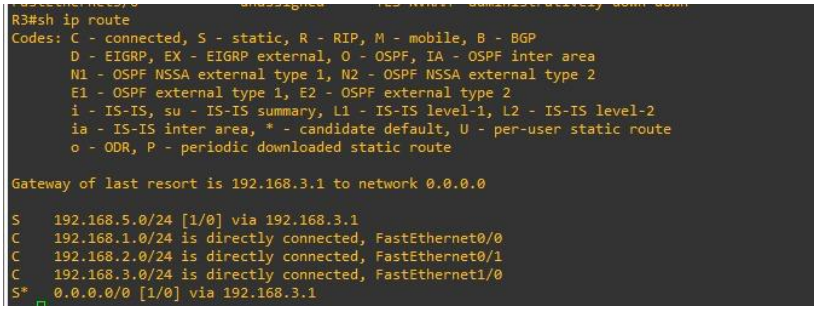
Gambar 5. Hasil dari "sh ip" PC3

Pada gambar 6 menunjukkan "*show IP interface brief*" pada R3 yang bertujuan untuk menampilkan status IP yang terhubung. *FastEthernet*0/0 terhubung dengan IP 192.168.1.13, *FastEthernet0/1* terhubung dengan IP 192.168.2.13, dan *FastEthernet*1/0 dengan IP 192.168.3.3.



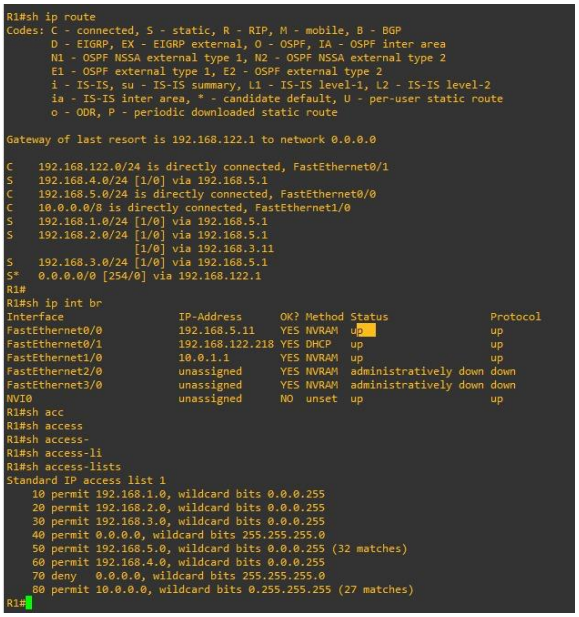
Gambar 6. Hasil dari command "sh ip int br" pada R3

Gambar 7 menjelaskan, *Command* "*sh ip route*" dilakukan untuk menunjukkan *IP* yang ada pada router R3. *FastEthernet0*/0 terhubung dengan 192.168.1.0/24, *FastEthernet0*/1 terhubung dengan 192.168.2.0/24, *FastEthernet*3/0 terhubung dengan 192.168.3.0. Lalu kami menghubungkan dengan antar IP menggunakan Static Route.



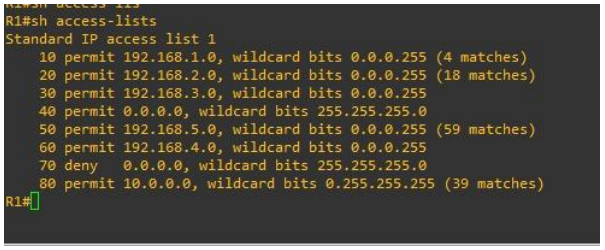
Gambar 7. Command "sh ip route" R3

Pada gambar 8, menunjukan *sh ip route* dari R1 dan menunjukan *sh ip int br*. Disini FastEthernet 0/1 terhubung dengan IP 192.168.122.0/24(dhcp), FastEthernet 0/0 terhubung dengan 192.168.5.0/24, FastEthernet 1/0 dengan IP 10.0.0.0/8. Disini kami menggunakan Static Route untuk menghubungkan semua IP address yg berada di R1.



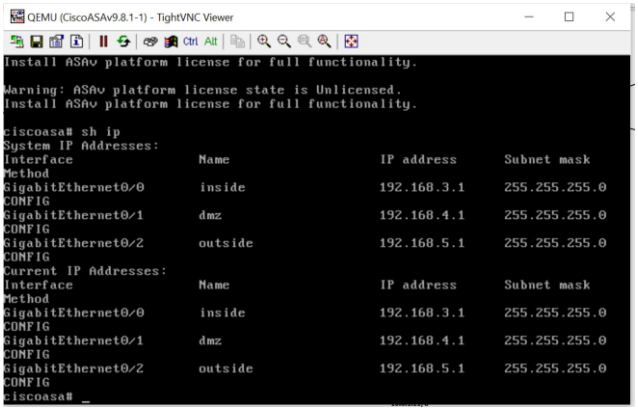
Gambar 8. "Show ip route dan show ip int brt" R1

Selanjutnya pada gambar 9 kita akan menunjukan access-list dari R1. Access-list ini berguna untuk memberi akses kepada IP yang mau diberik akses Internet.



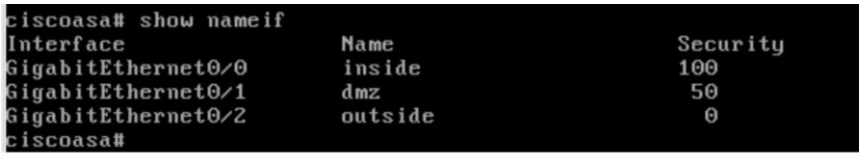
Gambar 9. Show access-list R1

Pada gambar 10 kami menunjukan *sh ip* pada cisco. Disini GI0/0 dengan IP 192.168.3.1 kami tetapkan sebagai bagian inside. GI0/1 dengan IP 192.168.4.1 kami tetapkan sebagai bagian DMZ. GI0/2 dengan ip 192.168.5.1 kami tetapkan sebagai bagian outside.



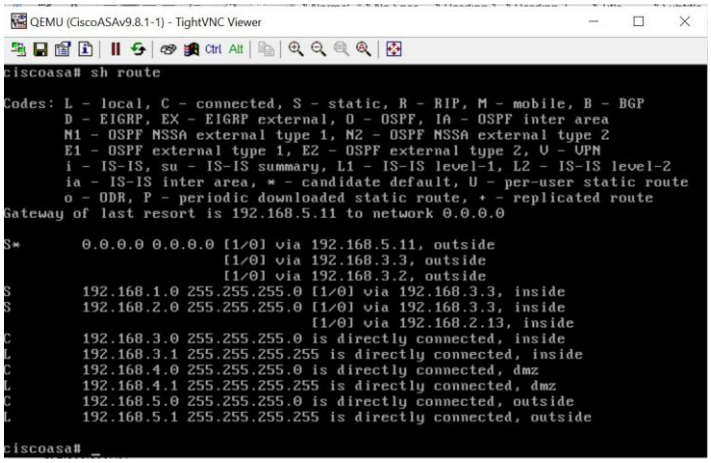
Gambar 10. Command “Sh ip” Cisco

Gambar 11 menunjukan security level yang kita atur untuk setiap bagian-bagian kantor. Yang pertama ada bagian inside yang kita atur security nya 100, DMZ 50, dan yang terakhir outside 0.



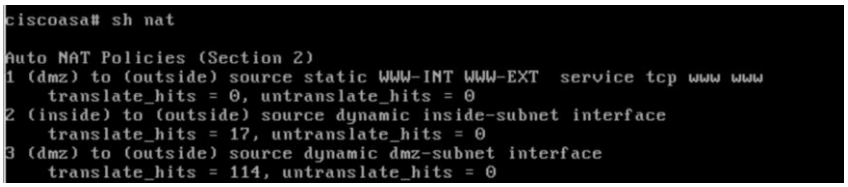
Gambar 11. Command “show nameif” Cisco

Gambar 12 menunjukkan IP 192.168.1.0, IP 192.168.2.0, IP 192.168.3.0, dan IP 192.168.3.1 menunjukan inside. IP 192.168.4.0, IP 192.168.4.1 menunjukan DMZ. Dan yang terakhir IP 192.168.5.0, IP 192.168.5.1 menunjukan outside.



Gambar 12. Command "sh route" cisco ASA

Pada gambar 13 *WWW-INT* dan *WWW-EXT*, menyambungkan antara DMZ dengan *outside. Inside-subnet interface menyambungkan antara inside dengan outside. DMZ-subnet interface menyambungkan antara DMZ dengan outside.*

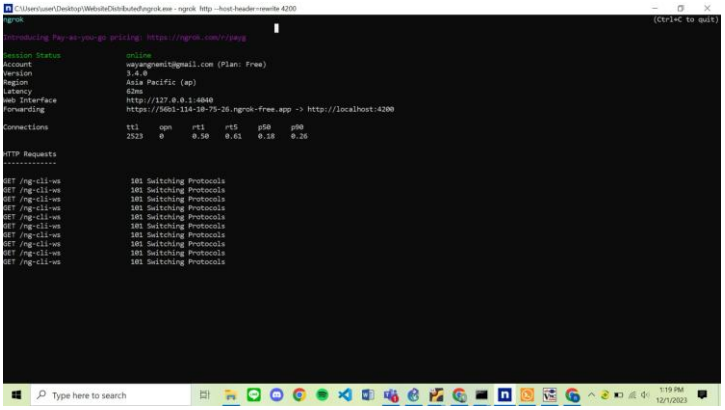


Gambar 13. Command "sh nat" Cisco

Gambar 14 menunjukkan memberikan akses kepada *WWW-INT* untuk 192.168.4.11.

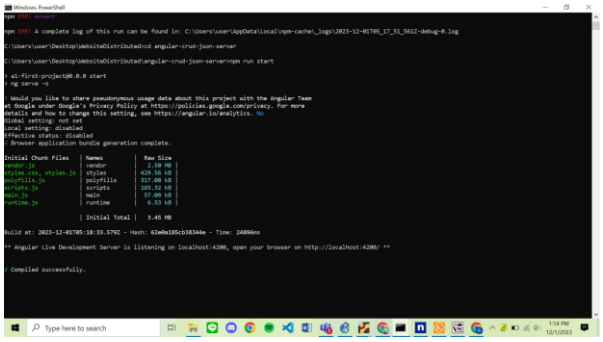
Gambar 14. Command "sh access-list" cisco

Gambar 15 menunjukan bahwa program ngrok.exe sedang berjalan dan sedang "online". Saat program ini dijalankan maka WebServer yang kami buat sudah dapat diakses melalui inside dan outside. Dan ngrok.exe ini juga membuat link *https://dc25-140-213-0-192.ngrok-free.app (saat program dijalankan maka otomatis akan membuat suatu link secara random untuk mengakses WebServer yang sudah kami buat)* agar dapat diakses melalui inside maupun outside.



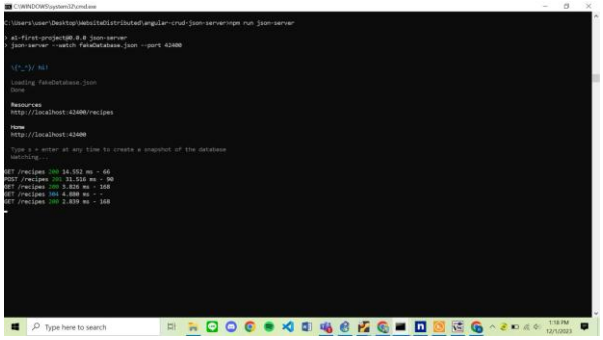
Gambar 15. Program ngrok.exe

Gambar 16 menunjukan CMD dari folder C;\Users\user\Dekstop\WebDistributed\angular-crud-json-server. Lalu di dalam CMD ini kami memberi perintah npm run start untuk menjalankan Front-end dari WebServer yang telah kami buat



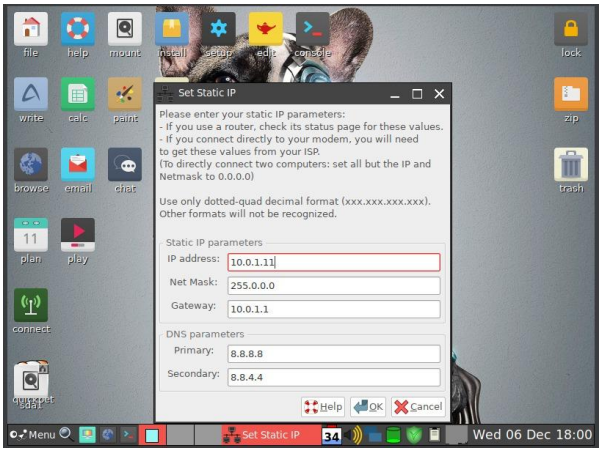
Gambar 16. Command Prompt(Front-end)

Gambar 17 menunjukan Back-end dari WebServer kami. Cara menjalankannya kami harus mengakses folder yang sama seperti Front-end. Berbeda dari yang sebelumnya perintah yang kita jalankan adalah “npm run json-server” untuk menjalakan Back-end WebServer kami.



Gambar 17. Command Prompt(Back-end)

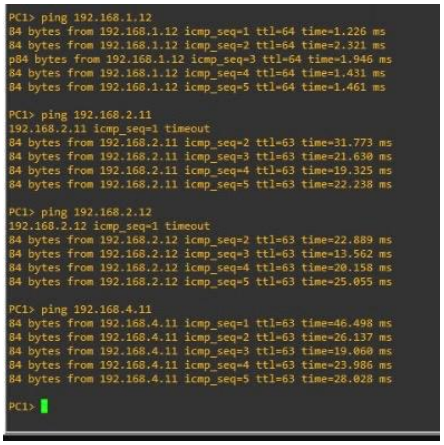
Pada gambar 18 kita memberikan IP address pada WebClient(PuppyLinux) “10.0.1.11” dengan gateway “10.0.1.1”



Gambar 18. set IP WebClient

Hasil Uji Coba

Pada gambar 19 kami menunjukan koneksi internal dan DMZ. Disini kami coba untuk ping PC2, PC3, PC4, dan DMZ.

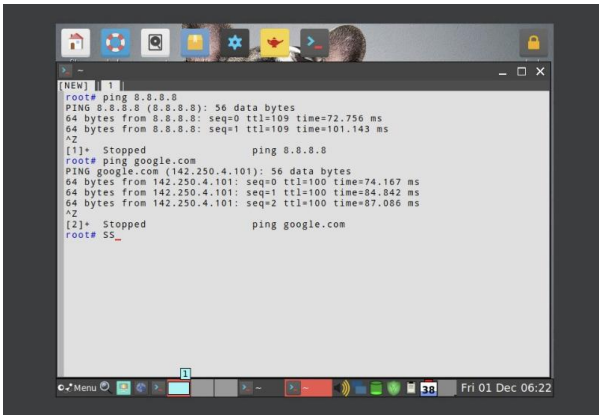


Gambar 19. Ping ke PC1, PC2, PC3, dan DMZ

Pada Gambar 20 kami menunjukan bahwa PC1(internal) bisa mengakses internet(google.com) dan WebServer yang kami buat sebagai DMZ(Demilitrisasi Zone)

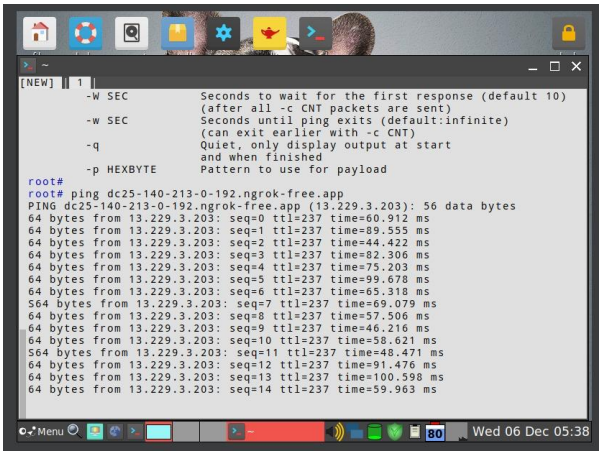
Gambar 20 menunjukan *PC1 bisa akses internet, google.com dan DMZ/WebServer(*https*://726a-103-74-170-5.ngrok-free.app)*

Gambar 21 menunjukan bahwa WebClient/PuppyLinuz(WFH) bisa mengakses Internet dan ping 8.8.8.8



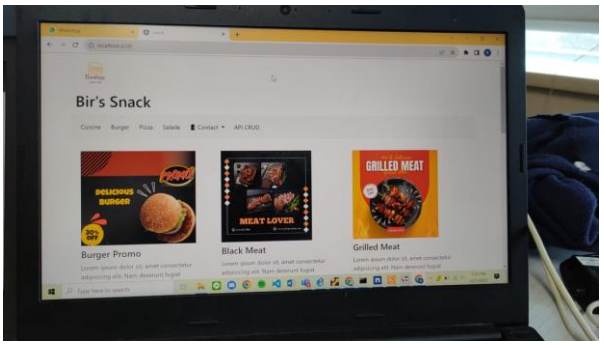
Gambar 21. Command "Ping 8.8.8.8, ping google.com" WebClient(PuppyLinux)

Gambar 22 menunjukan bahwa kita bisa mengakes WebServer yang kami buat melalui WebClient/PuppLinux(WFH).



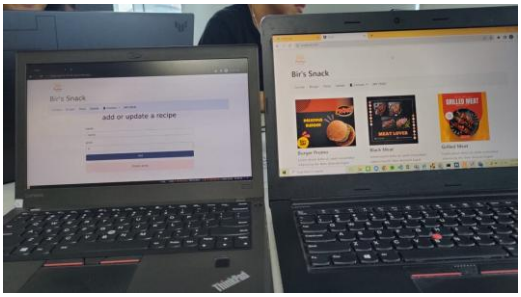
Gambar 22 *Command "Ping https://dc25-140-213-0-192.ngrok-free.app" WebClient(PuppyLinux)*

Gambar 23 menunjukan WebServer yang sudah kami buat didalam localhost 4200.



Gambar 23 *WebServer localhost 4200*

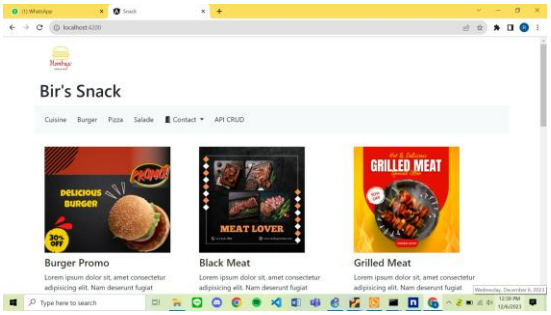
Gambar 24 menunjukan kami dapat mengakses WebServer kami dari Laptop yang berbeda dan dari HandPhone



Gambar 24. *WebServer dapat diakses melalui remote*

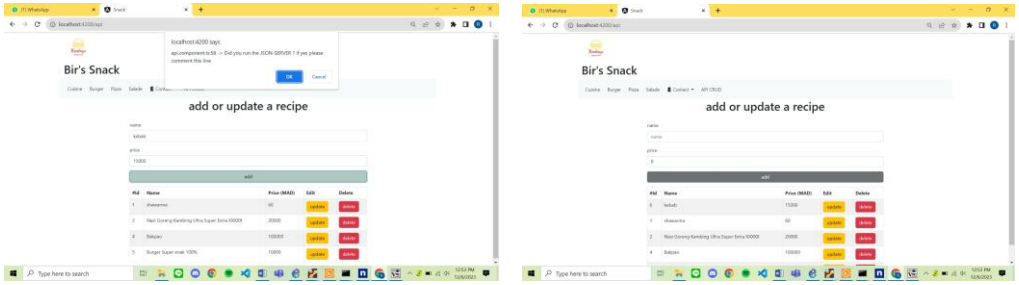
Gambar 25. *WebServer dapat diakses melalui remote*

Gambar 26 menunjukan Tampilan HomePage(Front-end) dari WebServer kami



Gambar 26. tampilan HomePage WebServer

Gambar 27 kami akan menunjukan Back-end yang sudah kami buat untuk WebServer kami. Kami disini membuat agar Pengguna bisa menambahkan dan menghapus menu.



Gambar 27. tampilan Back-end WebServer

5. KESIMPULAN DAN SARAN

Pada jaringan terdistribusi yang telah disimulasikan, *server* yang digunakan sebagai wadah untuk melakukan pekerjaan secara jarak jauh (*work from home*) ataupun secara jarak dekat (*work from office*), dengan menyediakan keamanan *firewall* yang berupa *Cisco ASA.*

Dengan simulasi jaringan ini, baik karyawan maupun pemimpin perusahaan dapat mengalami kemudahan karena semua aktivitas dapat dilakukan secara online melalui jaringan yang memungkinkan akses dari jarak jauh dan dapat digunakan secara bersamaan.

Namun, dalam pelaksanaan simulasi jaringan, terdeteksi beberapa kelemahan dalam aspek keamanan. Oleh karena itu, diperlukan peningkatan dengan mengintegrasikan penggunaan VPN sebagai lapisan keamanan tambahan pada jaringan, sekaligus memanfaatkan protokol HTTPS sebagai langkah perlindungan data pengguna yang lebih efisien. Pengembangan jaringan ke depan akan difokuskan pada peningkatan kapasitas untuk mendukung skala yang lebih besar, serta peningkatan keamanan guna membuat jaringan lebih tahan terhadap upaya penyusupan dan ancaman siber. Kemajuan jaringan ini dapat dijadikan sebagai platform untuk lebih mempermudah semua pekerja dalam memaksimalkan kinerja mereka demi kemajuan perusahaan.

DAFTAR REFERENSI

Desks, P. G., Guide, M. K., Keyboards, G. M., Keyboards, B. M., Keyboards, R. M., Backlight, B. M., & Earbuds, G. (2021). *Best Linux distro to run from USB*. Computer Software. https://computerstationnation.com/best-linux-distro-to-run-from-usb/

Fain, Y., & Moiseev, A. (2017). *Angular 2 development with TypeScript* (p. 456). Manning Publications Company. https://sundaramdesign.com/sites/default/files/angular-2-development-with-typescript-yakov-fain-anton-moiseev-d7bea4f.pdf

Khadafi, S., Nurmuslimah, S., & Anggakusuma, F. K. (2019). Implementasi firewall dan port knocking sebagai keamanan data transfer pada FTP server berbasiskan Linux Ubuntu server. *Network Engineering Research Operation*, *4*(3), 180–188. http://download.garuda.kemdikbud.go.id/article.php?article=1620052&val=11044&title=IMPLEMENTASI%20FIREWALL%20DAN%20PORT%20KNOCKING%20SEBAGAI%20KEAMANAN%20DATA%20TRANSFER%20PADA%20FTP%20SERVER%20BERBASISKAN%20LINUX%20UBUNTU%20SERVER

Liu, L. Y., & Shyamasundar, R. K. (1990). Static analysis of real-time distributed systems. *IEEE Transactions on Software Engineering*, *16*(4), 373–388. https://ieeexplore.ieee.org/abstract/document/54290

Mullender, S. (Ed.). (1990). *Distributed systems*. ACM. https://dl.acm.org/doi/abs/10.1145/90417

Neuman, B. C. (1994). *Scale in distributed systems* (Report No. 68). ISI/USC. https://cs.uwaterloo.ca/~brecht/courses/epfl/Possible-Readings/general/scale-dist-sys-neuman-readings-dcs-1994.pdf

Praghash, K., Eswar, V. S., Roy, J. Y., Alagarsamy, A., & Arunmetha, S. (2021, December). Tunnel-based intra network controller using NGROK framework for smart cities. In *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 39–43). IEEE. https://ieeexplore.ieee.org/abstract/document/9676036

Rose, M. E. (1953). The analysis of angular correlation and angular distribution data. *Physical Review, 91*(3), 610. https://journals.aps.org/pr/abstract/10.1103/PhysRev.91.610

Suryana, T. (2021). *Implementasi komunikasi web server Nodemcu ESP8266 dan web server Apache MySQL untuk otomatisasi dan kontrol peralatan elektronik jarak jauh via internet*. https://repository.unikom.ac.id/68717/

Terplan, K., & Huntington-Lee, J. (1997). *Applications for distributed systems and network management*. International Thomson Publishing. https://dl.acm.org/doi/abs/10.5555/550612

Tutang, T. (2016). Implementasi Network Address Translation (NAT) menggunakan Kerio Control versi 7.4.1 di Pusat Penelitian Bioteknologi–LIPI. *Baca: Jurnal Dokumentasi dan Informasi, 36*(1), 97–108. https://jurnalbaca.pdii.lipi.go.id/index.php/baca/article/view/162/0

Widharma, I. G. S. (n.d.). *Pengamanan sistem jaringan komputer dengan teknologi firewall*. https://www.researchgate.net/profile/I-Gede-Widharma-2/publication/346965331_PENGAMANAN_SISTEM_JARINGAN_KOMPUTER_DENGAN_TEKNOLOGI_FIREWALL_I_Gede_Suputra_Widharma_and_The_A_Team/links/5fd4d5fd299bf14088041120/PENGAMANAN-SISTEM-JARINGAN-KOMPUTER-DENGAN-TEKNOLOGI-FIREWALL-I-Gede-Suputra-Widharma-and-The-A-Team.pdf

Zhang, Y., Kameda, H., & Hung, S. L. (1997). Comparison of dynamic and static load-balancing strategies in heterogeneous distributed systems. *IEE Proceedings – Computers and Digital Techniques, 144*(2), 100–106. https://digital-library.theiet.org/content/journals/10.1049/ip-cdt_19970951