

Pengembangan *Network Automation* Menggunakan *WinAutomation* Studi Kasus Data Center IBM Indonesia

Yuda Samudra¹

¹Fakultas Teknik, Program Studi Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia
Email: dosen02623@unpam.ac.id, yudasamudra00@gmail.com

Abstrak—Replikasi data pada sebuah sistem SAP dapat menggunakan layanan *Data Center* dan *Disaster Recovery Center* dengan menggunakan sistem *Load Balancing* atau *Failover* pada perangkat *router*. Gangguan jaringan *internal* maupun *external* juga dapat terjadi yang tidak dapat diprediksi akan mengganggu kelancaran bisnis dan replikasi data. Contohnya gangguan jaringan seperti *intermittent* atau *downtime*, tetapi fungsi *Load Balancing* atau *Failover* tidak dapat berjalan dengan baik sehingga mengganggu fungsi replikasi data *Oracle Data Guard (ODG)* pada *system recovery DC (Data Center)* ke *DRC (Disaster Recovery Center)*. Maka penggunaan *Robot Automation* dapat meminimalisir gangguan jaringan internet guna mendapatkan *Service Level Agreement (SLA)* yang tinggi sesuai dengan *Service Delivery* berbasis *framework* untuk *Service Management*, yaitu *ITIL*. Untuk mencapai *SLA (Service Level Agreement)* yang tinggi dapat menggunakan pedoman *ITIL (IT Infrastructure Library)* sebagai kerangka kerja IT dengan *framework ITSM (IT Services Management)*. *Robot Automation* adalah suatu program yang dirancang untuk otomatisasi yang dapat dirancang untuk menangani kegagalan fungsi perpindahan jalur jaringan internet menggunakan *WinAutomation*. Hasil pengukuran kerja menggunakan survey kuisioner dengan partisipan terbatas pada lingkungan kerja *Data Center* dengan metode pengujian Beta menggunakan perhitungan skala *Likert*. Berdasarkan pengujian Beta, dapat ditarik kesimpulan bahwa sistem yang dibangun mudah digunakan, bermanfaat dan dapat membantu pengguna dalam membantu operasional secara cepat dan tepat ketika terdapat gangguan jaringan baik *internal* maupun *external* dengan rata-rata persentase keberhasilan 78,4 %.

Kata Kunci: *Network Automation, Load Balancing, Failover, Cisco, Juniper, ITIL, ITSM, Oracle Data Guard, Data Center, Disaster Recovery Center*

Abstract—*Data replication on a SAP system can use the Data Center and Disaster Recovery Center services using a Load Balancing or Failover system on a router device. Internal and external network disruptions may also occur which cannot be predicted which will disrupt the smooth running of business and data replication. For example network disruptions such as intermittent or downtime, but the Load Balancing or Failover function cannot run properly so that it interferes with the Oracle Data Guard (ODG) data replication function in the DC (Data Center) recovery system to the DRC (Disaster Recovery Center). Then the use of Robot Automation can minimize internet network disruptions in order to get a high Service Level Agreement (SLA) in accordance with a framework-based Service Delivery for Service Management, namely ITIL. To achieve a high SLA (Service Level Agreement), ITIL (IT Infrastructure Library) guidelines can be used as an IT framework with the ITSM (IT Services Management) framework. Robot Automation is a program designed for automation that can be designed to handle malfunctions of internet network switching using WinAutomation. The results of work measurement using a questionnaire survey with limited participants in the Data Center work environment with the Beta testing method using a Likert scale calculation. Based on Beta testing, it can be concluded that the system built is easy to use, useful and can assist users in assisting operations quickly and precisely when there is network disruption both internal and external with an average success percentage of 78.4%.*

Keywords: *Network Automation, Load Balancing, Failover, Cisco, Juniper, ITIL, ITSM, Oracle Data Guard, Data Center, Disaster Recovery Center*

1. PENDAHULUAN

Di era digitalisasi saat ini, perkembangan teknologi sangatlah cepat baik untuk mengatasi suatu masalah ataupun sebagai bentuk inovasi dari suatu teknologi yang digunakan untuk pengelolaan data dibutuhkan sangat cepat dan tepat untuk digunakan secara berkala guna menghasilkan informasi yang tepat dan akurat serta efisiensi. Salah satu bentuk perkembangan teknologi saat ini adalah Teknik *Network Automation* yang sangat dibutuhkan untuk menangani gangguan network ketika dimana suatu *main link* mengalami gangguan dan dapat di proses melalui *Automation System* untuk berpindah ke *secondary link*.

Pada prosesnya, *Automation* ini merupakan program aplikasi yang ketika dijalankan akan melakukan konfigurasi secara otomatis pada perangkat Router yang berbeda, karena permasalahan *Load Balancing* dari suatu Router tidak dapat berjalan dengan baik karena perbedaan *Vendor* dari Router tersebut sehingga ketika terjadi gangguan pada *main link*, koneksi jaringan yang menghubungkan *Branch - Data Center - Disaster Recovery Center* tidak dapat berpindah secara otomatis, karena teknologi *Load Balancing* tidak dapat berjalan pada perangkat yang berbeda ketika terdapat *limit speed connection* yang berbeda.

Berdasarkan *ITIL (IT Infrastructure Library)*, teknis *Delivery Services* menggunakan *Based Framework* untuk *ITSM (IT Services Management)*, sedangkan banyak ditemukan proses *Delivery Services* tidak sesuai dengan aturan hingga terjadi kesalahan prosedural yang mengakibatkan tidak tercapainya *SLA (Service Level Agreement)* yang berdampak pada penurunan performa dari penyedia *Services*. Proses *problem solving* yang dianggap terlalu lama karena perbedaan *device* dan *speed limit connection* menghambat proses *Delivery Service* yang tidak menggunakan *Based Framework* untuk *ITSM* menjadi suatu pelanggaran yang berdasarkan dari *Statement Of Work* antara penyedia jasa dan *client*.

WinAutomation adalah suatu program yang dirancang untuk pengelolaan manajemen *network link* ketika terjadi gangguan dari *Internet Service Provider* sehingga terganggu jaringan koneksi *client* pada jaringan *SAP*, karena pada dasarnya sistem *SAP* menggunakan *main link* sebagai koneksi utama dan *secondary link* untuk koneksi alternatif ketika mengalami gangguan, maka ketika teknologi *Load Balancing* tidak dapat berjalan dikarenakan tidak memenuhi suatu syarat, dengan *Automation* dari *WinAutomation* dapat mengatasi masalah tersebut sebagai pilihan alternatif.

Berdasarkan uraian diatas maka diperlukan suatu proses manajemen *network link* yang terstruktur secara prosedural, sehingga proses *Delivery Services* pada *connection network link* tersampaikan dengan baik untuk meningkatkan kepuasan pelanggan berdasarkan *Service Level Agreement*.

Metode yang dapat digunakan untuk menghasilkan sistem informasi ini adalah *ITIL* yang merupakan *Best Practice* untuk *ITSM (IT Service Management)* yang digunakan untuk *Delivery Services*. Untuk menyampaikan pelayanan kepada *client*, tidak secara acak menggunakan metode yang tersedia, melainkan berdasarkan prosedural yang diakui secara umum dan menghasilkan kesepakatan diantara kedua belah pihak antara penyedia layanan dan pelanggan. Didalam *ITIL* terdapat lima komponen utama untuk menghasilkan informasi berdasarkan keinginan pelanggan, yaitu *ITIL Service Strategy*, *ITIL Service Design*, *ITIL Service Transition*, *ITIL Service Operation*, dan *ITIL Continual Service Improvement*. Teknis pelayanan ini digunakan untuk menghasilkan informasi yang tujuannya untuk mencapai kepuasan pelanggan berdasarkan *Service Level Agreement*.

Berdasarkan masalah yang ada di atas, maka dibuat karya ilmiah dengan judul: **PENGEMBANGAN NETWORK AUTOMATION MENGGUNAKAN WINAUTOMATION.**

2. METODOLOGI PENELITIAN

2.1 Penerapan Service Level Agreement (SLA)

Penerapan dalam pencapaian *Service Level Agreement* yang didapatkan berdasarkan dari jumlah *incident* hingga *problem* berdasarkan kesepakatan dari *Statement of Work* antara kedua belah pihak. Berikut seperti yang ditunjukkan pada “Gambar 2.1 Contoh Komponen SLA” untuk contoh data yang digunakan sebagai perbandingan dalam pencapaian *SLA* pada saat akhir laporan.

SLA							
2016							
Measured KPI	Target	Jul	Aug	Sep	Oct	Nov	Dec
Data Center Support	98%	100%	100%	98.3%	100%	100%	100%
SAP Production Servers & Storage Availability	99.5%	100%	100%	100%	100%	100%	100%

2017							
Measured KPI	Target	Jan	Feb	Mar	Apr	May	Jun
Data Center Support	98%	100%	100%	100%	100%	100%	100%
SAP Production Servers & Storage Availability	99.5%	100%	100%	100%	100%	100%	100%

2018							
Measured KPI	Target	Jul	Aug	Sep	Oct	Nov	Dec
Data Center Support	98%	100%	100%	100%	100%	100%	100%
SAP Production Servers & Storage Availability	99.5%	100%	100%	100%	100%	100%	100%

Gambar 1. Contoh Komponen SLA

Setiap hasil SLA didapatkan dengan rumus yang didapat dari pedoman *ITIL* seperti yang ditunjukkan pada “Gambar 2.2 Rumus Perhitungan Pencapaian SLA”

Service Availability

$$\begin{aligned}
 A &= (t_1 - t_0 - t_1 - t_2) * 100 / (t_1 - t_1 - t_2) \\
 &= ((31 * 24) - 0 - 0 - 0) * 100 / ((31 * 24) - 0 - 0) \\
 &= 720 * 100 / 720 \\
 &= 100
 \end{aligned}$$

Where A is availability (%). So, service availability is 100%

Gambar 2. Rumus Perhitungan Pencapaian SLA

Jika target pada *performance server & network availability* dibawah standar, maka dapat disimpulkan sementara bahwa target SLA tidak tercapai, tetapi dapat dibantu dengan komponen lain agar target lainnya tercapai.

2.2 Skala Likert

Kuesioner adalah teknik yang digunakan secara luas untuk memperoleh informasi dari subjek. Kuesioner relatif ekonomis, memuat pertanyaan yang sama bagi seluruh subjek dan dapat memastikan kerahasiaan subjek. Kuesioner dapat menggunakan pertanyaan atau pernyataan, tetapi dalam banyak kasus subjek merespon pada sesuatu yang ditulis secara khusus (Hamdi, Bahruddin, 2014).

Pada pengujian kuesioner ini menggunakan metode *Skala Likert* untuk mengetahui hasil kepuasan pelanggan terhadap uji coba aplikasi. Skala *Likert* merupakan skala yang mengukur kesetujuan atau ketidaksetujuan seseorang terhadap serangkaian pernyataan berkaitan dengan keyakinan atau perilaku mengenai suatu obyek tertentu. Format skala *Likert* merupakan perpaduan antara kesetujuan dan ketidaksetujuan. Skala ini dikembangkan oleh *Rensis Likert* sehingga dikenal dengan *Skala Likert* (Hamdi, Bahruddin, 2014).

<p>RUMUS:</p> $\frac{T \times Pn}{Y}$ <p>T = Total jumlah responden yg memilih Pn = Pilihan angka Skor Likert</p>	<p>RUMUS INDEX % :</p> $\frac{\text{Total Skor} / Y \times 100}{Y}$ <p>Y = Skor tertinggi Likert x jumlah responden</p>
--	---

Gambar 2 Rumus Hitung Skala Likert Dan Hitung Skor Index

Skala ini umumnya menggunakan lima angka penilaian seperti yang ditunjukkan pada “Gambar 2.3 Rumus Hitung Skala Likert Dan Hitung Skor Index”, yaitu (Hamdi, Bahruddin, 2014) :

1. (1) Sangat Tidak Setuju.
2. (2) Tidak Setuju.
3. (3) Netral.
4. (4) Setuju.
5. (5) Sangat Setuju.

3. ANALISA DAN PEMBAHASAN

3.1 Analisis Sistem

Analisis sistem ini digunakan untuk proses penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan solusi atau perbaikan. Dari hasil analisis tersebut dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efisien. Sistem yang dibuat merupakan Automation System Robotics untuk mengubah konektivitas jaringan ketika terjadi gangguan (Intermittent atau Outage). Objek penelitian ini dilakukan di Data Center milik IBM Indonesia

3.1.1 Analisis Masalah

Analisis masalah adalah langkah awal dari analisis sistem yang diperlukan untuk mengetahui permasalahan apa saja yang terjadi didalam sistem yang telah berjalan. Berdasarkan analisis sistem yang dilakukan dengan cara mengamati proses *problem solving* saat terjadi gangguan dari *ISP* (*Intermittent* atau *Outage* Jaringan) dirasakan tidak efisien dari sisi *Engineer* maupun *Timing*.

3.1.2 Analisis Metode

Load Balancing adalah fungsi standar dari perangkat lunak *router Cisco IOS*, dan tersedia di semua *platform router*. Itu melekat pada proses penerusan di *router* dan secara otomatis diaktifkan jika tabel routing memiliki banyak jalur ke tujuan. Ini didasarkan pada protokol routing standar, seperti *Routing Information Protocol (RIP)*, *RIPv2*, *Enhanced Interior Gateway Routing Protocol (EIGRP)*, *Buka Shortest Path First (OSPF)*, dan *Interior Gateway Routing Protocol (IGRP)*, atau berasal dari konfigurasi secara statis rute dan mekanisme penerusan paket. Hal ini memungkinkan *router* untuk menggunakan banyak jalur ke tujuan saat meneruskan paket.

Otomatisasi jaringan (*Network Automation*) adalah metodologi di mana perangkat lunak secara otomatis mengkonfigurasi, ketentuan, mengelola dan menguji perangkat jaringan. Ini digunakan oleh perusahaan dan penyedia layanan untuk meningkatkan efisiensi dan mengurangi kesalahan manusia dan biaya operasi. Alat otomasi jaringan mendukung fungsi mulai dari pemetaan jaringan dasar dan penemuan perangkat, hingga alur kerja yang lebih kompleks seperti manajemen konfigurasi jaringan dan penyediaan sumber daya jaringan virtual.

3.2 Perancangan Penelitian

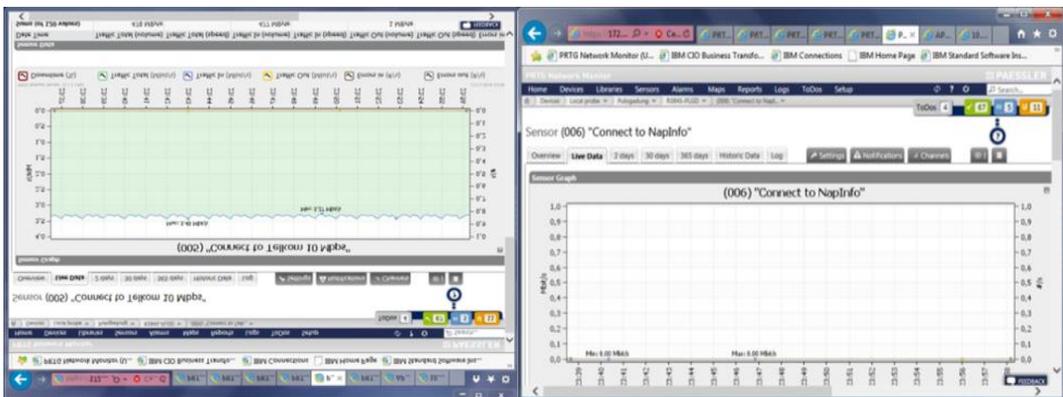
Perancangan penelitian dilakukan dengan analisis terhadap kebutuhan bisnis dan tujuan target bisnis dari *client*. Tahapan dalam perancangan ini adalah:

1. Analisis
2. Desain
3. Verifikasi
4. Evaluasi

4. IMPLEMENTASI

4.1 Hasil Analisis Monitoring

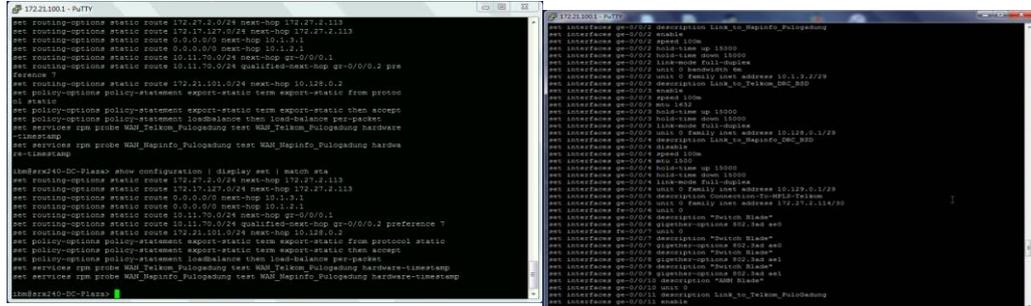
Berdasarkan hasil implementasi system *Automation*, maka sebagai acuan yang menjadi tolak ukur keberhasilan adalah *traffic graphic* pada layar *monitoring tools PRTG*. Dalam kasus uji coba adalah memindahkan koneksi *Primary Link to Secondary Link*, dimana jika *Primary Link* mengalami gangguan (dalam uji coba akan dimatikan secara paksa) maka pada *monitoring tools primary link* akan terdapat indikasi *alarm notification* sebagai tanda koneksi mengalami gangguan dan di saat yang sama jika proses perpindahan jaringan berhasil dilakukan, maka layar *monitoring tools secondary link* akan menunjukkan *traffic graphic up*.



Gambar 4. Default Traffic Graphic Primary Link & Default Traffic Graphic Secondary Link

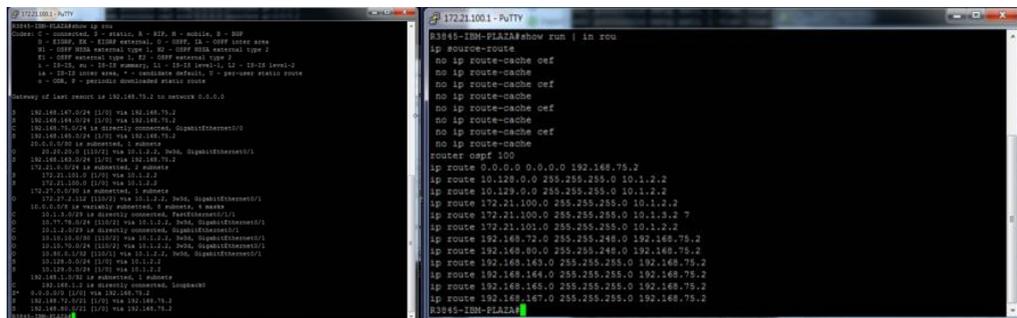
Koneksi *Primary Link* menggunakan *Telkom Link* seperti yang ditunjukkan pada “*Default Traffic Graphic Primary Link*” akan terlihat dari *system monitoring PRTG* dengan menampilkan *graphic traffic* sesuai dengan *bandwidth usable*. Sedangkan koneksi *Secondary Link* menggunakan *NapInfo Link* seperti yang ditunjukkan pada “*Default Traffic Graphic Secondary Link*” akan terlihat dari *system monitoring PRTG* tetapi tidak menampilkan *graphic traffic* sesuai dengan *bandwidth usable* karena digunakan sebagai *Backup Link* yang dimana jika *Primary Link* mengalami gangguan,

maka *Secondary Link* ini yang akan digunakan untuk konektivitas yang sebelumnya digunakan melalui *Primary Link*.



Gambar 5 Routing Table On Juniper OS (SRX240) & Configuration On Juniper OS (SRX240)

Setiap perangkat router memiliki konfigurasi yang menghubungkan setiap perangkat baik *DC-DR* maupun *DC-BRANCHES*, maka konfigurasi sudah disesuaikan dengan kebutuhan *client*. Pada konfigurasi *router DC* yaitu perangkat *Juniper* dengan *JUNOS* seperti yang ditunjukkan pada “Gambar 4.2 *Routing Table On Juniper OS (SRX240) & Configuration On Juniper OS (SRX240)*” memiliki *interface port* yang mengarah ke *DRC* dan *BRANCHES*. Melalui *router DC* ini yang nantinya semua konfigurasi dijalankan untuk merubah konfigurasi perangkat *router DC* maupun perangkat *router DRC* dan perangkat *router BRANCHES* dan semua proses ini melalui *terminal Putty* sebagai penghubung ke *router* yang bersangkutan.



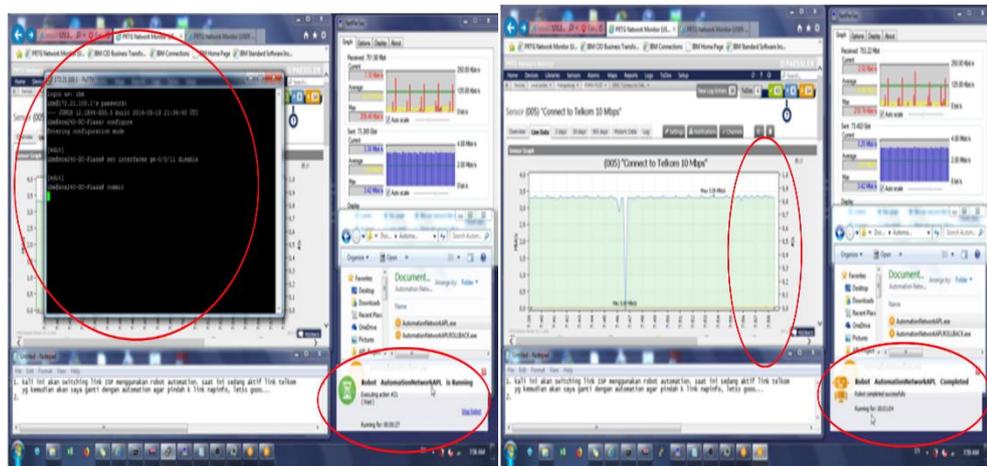
Gambar 6. Configuration On Cisco iOS (R3845) & Routing Table On Cisco iOS (R3845)

Berbeda dengan perangkat pada *DC*, perangkat pada *BRANCHES* menggunakan *router CISCO R3845* dengan konfigurasi seperti yang ditunjukkan pada “Gambar 4.3 *Configuration On Cisco iOS (R3845) & Routing Table On Cisco iOS (R3845)*” yang dimana untuk sistemnya menggunakan *IOS (Interconnecting Operation System)* dan memiliki konfigurasi pada setiap *port* yang berbeda mengarah langsung ke *DC* dengan masing masing *port* memiliki fungsionalitas tersendiri dan pada penelitian ini hanya difokuskan untuk *port* yang digunakan sebagai *port Primary Link* dan *port Secondary Link*.

4.1.1 Hasil Pengujian



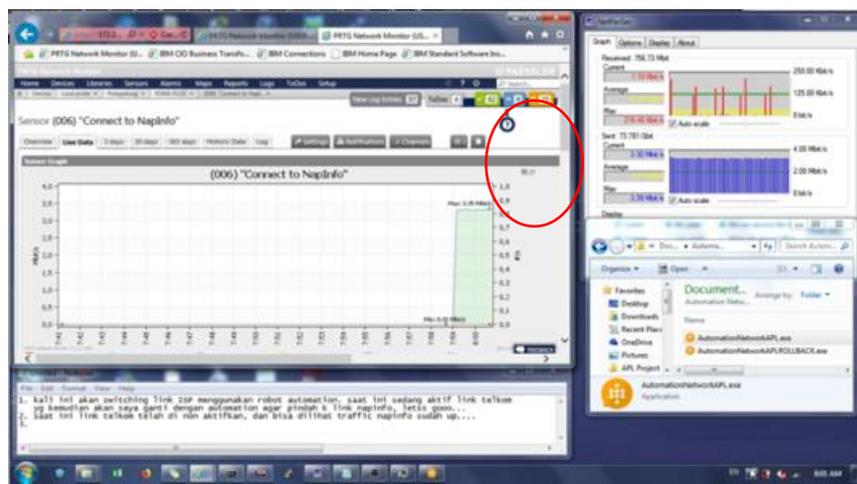
Gambar 7. Pop Up Robotic Notification



Gambar 8. Running Automation for Switching Primary Link Finished Automation for Switching Primary Link

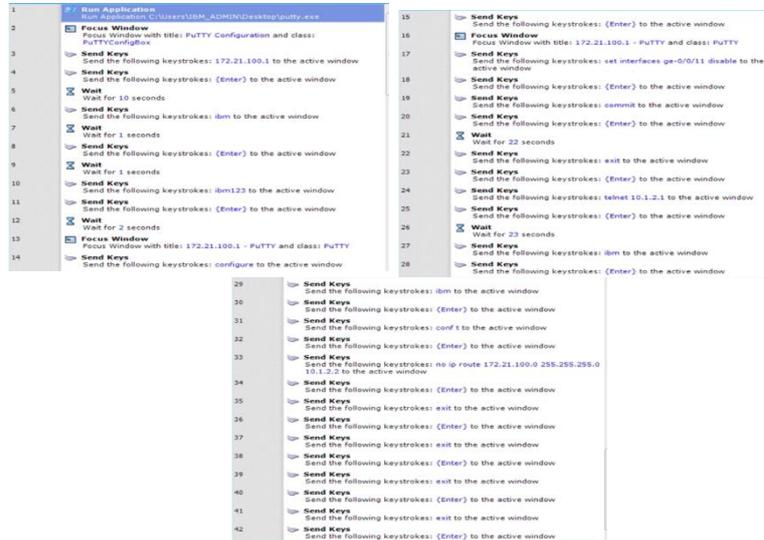
Hasil pengujian *Network Automation* ini berdasarkan program yang dijalankan akan memunculkan *pop up notification* seperti yang ditunjukkan pada “Gambar 7. *Pop Up Robotic Notification*” dengan detail pada “Gambar 4.5 *Running Automation for Switching Primary Link Finished Automation for Switching Primary Link*” dengan keterangan *Robot Running* dimana semua proses akan dijalankan langsung dengan eksekusi berdasarkan alur yang dibuat dalam *Robot Designer* sesuai dengan urutan yang telah ditentukan. Ketika selesai dijalankan, pada *graphic traffic primary link* akan menunjukkan *alert* pada *graphic* berupa titik-titik merah seperti yang ditunjukkan pada “Gambar 4.5 *Running Automation for Switching Primary Link Finished Automation for Switching Primary Link*” yang menunjukkan bahwa *link* tersebut mengalami gangguan dan pada proses lain pada *graphic traffic secondary link* juga menunjukkan titik-titik merah yang menunjukkan bahwa sedang terjadi gangguan, jika koneksi berhasil dipindahkan antara *primary link* ke *secondary link*, maka *graphic traffic secondary link* akan secara otomatis menampilkan *graphic traffic* sesuai dengan *limit bandwidth usable*.

Dalam hasil pengujian, waktu yang dibutuhkan untuk eksekusi semua aktivitas *Robot Network Automation* tidak lebih dari 64 detik, proses yang ini yang diharapkan sebagai efisiensi waktu dalam *problem solving*.



Gambar 9. Connection has been Switched to Secondary Link

Dalam hasil pengujian lainnya, ketika koneksi berhasil dipindahkan sesuai yang telah dijelaskan sebelumnya, maka dalam *graphic traffic secondary link* akan menampilkan *limit bandwidth usable*.

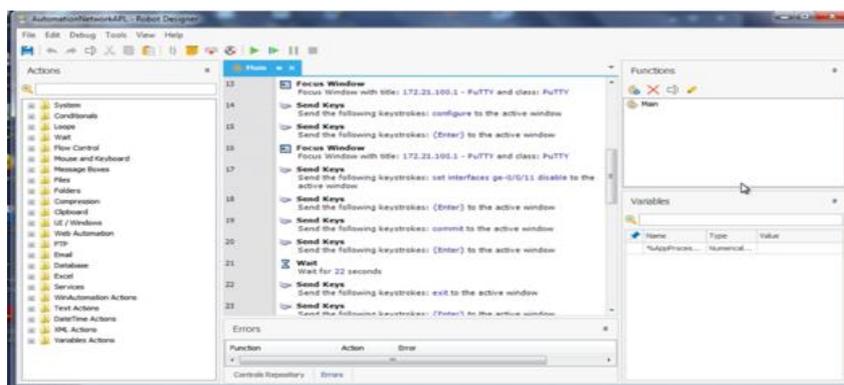


Dengan indikasi ini maka proses perpindah koneksi berhasil dilakukan.

4.1.2 Pembahasan

4.1.2.1 Robotics Designer

Pembahasan pengembangan sistem ini adalah suatu proses penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan solusi atau perbaikan. Dari hasil analisis tersebut dapat dirancang atau diperbaiki menjadi sebuah sistem yang lebih efektif dan efisien. Sistem yang dibuat merupakan *Automation System Robotics* untuk mengubah konektivitas jaringan ketika terjadi gangguan (*Intermittent* atau *Outage*). Objek penelitian ini dilakukan di *Data Center* milik IBM Indonesia



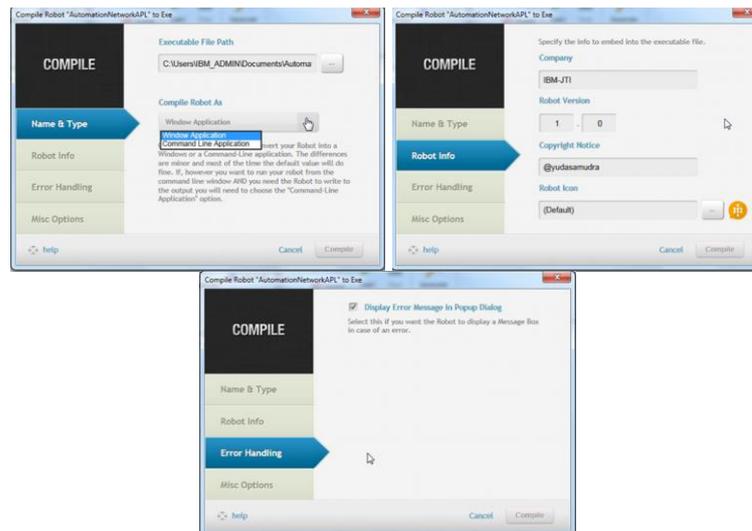
Gambar 10. Robot Designer Network Automation

Tampilan ketika membuat alur dari *Network Automation* pada *WinAutomation*, pada tampilan ini alur proses disebut sebagai *Robot Designer* dan setiap proses memiliki fungsi masing-masing seperti yang ditunjukkan pada “Gambar 4.7 *Robot Designer Network Automation*”.

Berikut akan dijabarkan *step definitions* dari alur proses pembuatan *Network Automation* yang dimana setiap langkah nya akan dieksekusi ketika sudah di generate dalam bentuk *.EXE file*. Berikut langkah-langkah *step definitions* seperti yang ditunjukkan pada “Gambar 4.8 *Robot Designer Network Automation Details*”:

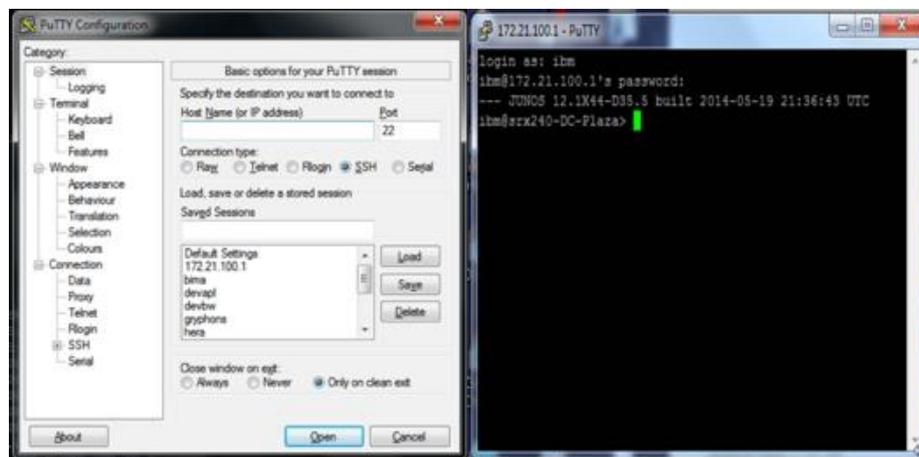
Gambar 11. Robot Designer Network Automation Details

Berdasarkan langkah-langkah diatas, maka jika sudah dilakukan uji coba internal, dapat di generate menjadi aplikasi yang siap digunakan secara instan hanya dengan double click. Untuk proses generate dilakukan menjadi aplikasi berbasis Windows yaitu berekstensi .exe yang dimana komponen lainnya dapat diatur sesuai kebutuhan pengguna. Berikut screen on untuk generate aplikasi ini seperti yang ditunjukkan pada “Gambar 4.9 Generate and Compile Program Robotics” :



Gambar 12 Generate and Compile Program Robotics

4.1.2.2 Terminal Putty



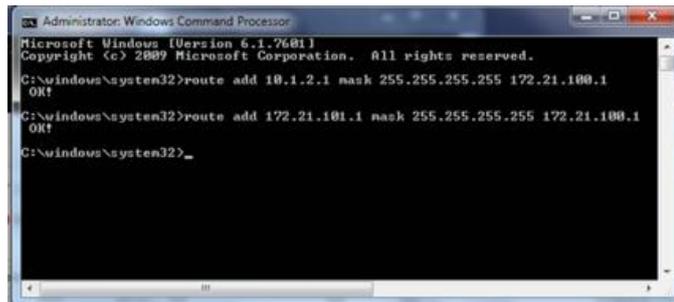
Gambar 13 Putty Screen Login Access

Terminal Putty seperti yang ditunjukkan pada “Gambar 4.10 Putty Screen Login Access” digunakan untuk mengakses perangkat router DC, router DRC, router BRANCHES. Connection type menggunakan SSH dan Telnet yang disesuaikan dengan konfigurasi masing-masing router. Untuk Connection Type SSH maka akan default port adalah 22 digunakan untuk mengakses router DC, sedangkan untuk mengakses router BRANCHES yang menggunakan router CISCO akan menggunakan Connection Type Telnet, maka menggunakan default port 23. Khusus akses menggunakan Connection Type Telnet port 23, maka routing table dalam sistem Data Center IBM diharuskan menambahkan routing table kemana tujuan routing table itu akan diakses, karena yang akan diakses adalah perangkat router CISCO BRANCHES maka akan ditambahkan

routing table router CISCO BRANCHES dalam komponen Windows sebagai bridge antara kedua router sehingga dapat langsung diakses dengan Putty. Untuk menambahkan routing table di dalam Windows, maka dapat dilakukan dengan menggunakan CMD dengan parameter berikut :

“Route Add <Destination IP Address> mask <Subnet Mask> <IP Address Device Bridge>”

Berikut adalah screen on ketika menambahkan routing table pada Windows dilakukan menggunakan CMD seperti yang ditunjukkan pada “Gambar 4.11 Add Routing Table On Windows System” dan “Gambar 4.12 Checking Routing Table On Windows System”, yaitu :



Gambar 14. Add Routing Table On Windows System



Gambar 15. Checking Routing Table On Windows System

4.1.3 Pengujian Alpha

Pengujian Alpha dilakukan pada sisi pengembangan yang merekam semua kesalahan dan masalah pemakaian. Pengujian Alpha dilakukan pada sebuah lingkungan yang terkendali, yaitu pada ruang lingkup aplikasi berjalan yang sudah siap digunakan. Pengguna akan menggunakan aplikasi dan pengembang mencatat setiap masukan atau tindakan yang dilakukan oleh pengguna (Simarmata, 2010).

4.1.3.1 Rencana Blackbox Testing

Rencana pengujian berfokus terhadap fungsi yang ada di dalam aplikasi yang dibangun, menu fungsional dari aplikasi berfungsi sesuai yang diharapkan atau tidak. Berikut ini tabel 4.1 rencana pengujian dari aplikasi yang dibangun.

Tabel 2. Rencana Pengujian Aplikasi

No	Komponen Pengujian	Fitur Pengujian	Jenis Pengujian
1	Running Application	Open Putty	Blackbox
		Input Parameter Putty	Blackbox
		Execute process in Router OS	Blackbox
		Pop Up Robotics	Blackbox

4.1.3.2 Blackbox Testing

Pengujian fungsional yang digunakan untuk menguji sistem yang baru adalah metode pengujian *Alpha*. Metode yang digunakan dalam pengujian ini adalah *Blackbox* yang berfokus pada persyaratan fungsional dari sistem yang dibangun. Untuk pengujian *Blackbox* ini dilakukan sebagai pengguna sistem.

4.1.3.3 Kasus Dan Hasil Pengujian

Kasus dan hasil pengujian berisi pemaparan dari rencana pengujian yang telah disusun pada skenario pengujian. Pengujian ini dilakukan secara *blackbox* dengan hanya memperhatikan masukan ke dalam sistem dan keluaran dari masukan tersebut. Berdasarkan rencana pengujian, maka dapat dilakukan pengujian pada aplikasi *Network Automation* sebagai berikut:

Tabel 3. Pengujian Running Application

No	Komponen Pengujian	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
1	Running Application	Open Putty	Robot will open Putty from Windows Apps	[V] Success [] Not Success
2		Input Parameter Putty	Parameter has been set on internal robot and will gave input on column text or click button in terminal Putty	[V] Success [] Not Success
3		Execute process in Router OS	After open terminal Putty, process will be execute all parameter inputs for acces router Juniper and Cisco	[V] Success [] Not Success
4		Pop Up Robotics	When process is running, pop up will be show as notification to owner	[V] Success [] Not Success

4.1.4 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan dengan tujuan untuk mengetahui sejauh mana kualitas sistem pada aplikasi yang dibangun, apakah sudah memenuhi harapan atau belum. Pengujian dilakukan secara objektif dimana pengujian dilakukan secara langsung kepada responden. Pengujian ini dilakukan kepada 30 orang yang telah menggunakan aplikasi.

4.1.4.1 Skenario Pengujian Beta

Pengujian Beta ini menggunakan perhitungan *Skala Likert*, yaitu adalah suatu *skala psikometrik* yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Penggunaan yang penelitian yang sering menggunakan skala ini adalah bila penelitian menggunakan jenis penelitian *Survei Deskriptif* (Gambaran). Sewaktu menanggapi pertanyaan dalam *Skala Likert*, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia.

<p>RUMUS :</p> <p>$T \times P_n$</p> <p>T = Total jumlah responden yg memilih P_n = Pilihan angka Skor Likert</p>	<p>RUMUS INDEX % :</p> <p>$\frac{\text{Total Skor}}{Y} \times 100$</p> <p>Y = Skor tertinggi Likert x jumlah responden</p>
---	--

Gambar 16. Rumus Hitung Skala Likert Dan Hitung Skor Index

4.1.5 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian *Alpha* yang telah dilakukan, dapat disimpulkan bahwa sistem secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

Berdasarkan pengujian beta pada aplikasi, dapat ditarik kesimpulan bahwa sistem yang dibangun mudah digunakan, bermanfaat dan dapat membantu pengguna dalam membantu operasional secara cepat dan tepat ketika terdapat gangguan jaringan baik *internal* maupun *external* yang ada dengan lebih efektif, hal ini sesuai dengan persentase jawaban setiap pengguna atau responden terhadap pertanyaan (kuesioner) yang telah disebarakan.

5. KESIMPULAN

Kesimpulan Berdasarkan hasil survei yang dilakukan kepada responden terkait di *Network Automation* di lingkungan *DATA CENTER IBM INDONESIA*, maka dapat diambil kesimpulan bahwa Aplikasi ini dapat membantu *Engineer* baik *Dedicate* maupun *Non-Dedicated* untuk melakukan problem solving secara cepat dari sebelum adanya *Network Automation* ini di implementasikan. Informasi penggunaan dan *configuration path* yang disajikan dengan menerapkan *Network Automation* menjadi lebih efektif dan terperinci berdasarkan *point-to-point* dari *routing table* dalam memberikan informasi menggunakan *Network Automation* yang digunakan untuk *switching connection link*.

Aplikasi ini juga membantu project lain jika di implemntasikan sesuai karakteristik project dengan *Base On Windows Application* yaitu *Putty*

REFERENCES

- Doerr, Bryan. (2007), Shared Data Center Disaster Recovery Systems And Methods. Ballwin: United States Patent.
- Hamdi, Asep Saepul. Bahruddin, E. (2014). "Metode Penelitian Kuantitatif Aplikasi Dalam Pendidikan". Yogyakarta: Deepublish.
- Juniper Network, Inc. (2015). Network Automation And Orchestration. White Paper, pp. 1-8
- Limited, Axelos, *ITIL Foundation: ITIL 4 Edition*, 4th ed. London: The Stationery Office, 2019
- Marrone, Mauricio. (2014). IT Service Management: A Cross-national Study of ITIL Adoption. Australia: *Macquarie University*.
- MIHĂILĂ, Paul., BĂLAN, Titus., CURPEN, Radu., SANDU, Florin. (2017). Network Automation and Abstraction using Python Programming Methods. Electronics and Computers Department, Faculty of Electrical Engineering and Computer Science: Transilvania University.
- Prameswari, Nur Shabrina dan Susanto, Tony Dwi. (2016). Pembuatan Service Level Requirement, Service Level Agreement dan Operational Level Agreement pada Layanan Help Desk SAP Berdasarkan Kerangka Kerja ITIL Versi 2011. JURNAL TEKNIK ITS Vol. 5, No. 2, pp. A453-A457
- Rajagopalan, S. (2020). An Overview of Server Load Balancing. International Journal of Trend in Research and Development, Volume 7(2), pp. 231-232
- Singh, Gurpreet. (2014). Configuring Oracle Dataguard Using Logical Standby Database. Gurukul Vidyapeeth Institute of Engineering & Technology, Banur, Punjab, India: CSE/IT Department.
- Singh, Ravindra Kumar., Chaudhari, Narendra S., dan Saxena, Kanak. (2012). Load Balancing in IP/MPLS Networks: A Survey. Communications and Network, 4, 151-156