

## BAB 6

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan implementasi dan pengujian perangkat lunak automasi alat jaringan berbasis *web*, diperoleh beberapa kesimpulan sebagai berikut.

1. Perangkat lunak yang dibangun sudah dapat melakukan automasi konfigurasi alat jaringan tanpa harus mengetikkan secara manual baris perintah yang biasanya digunakan untuk melakukan berbagai konfigurasi alat jaringan.
2. *Library* Paramiko memiliki performa waktu yang cepat dan ukuran data yang relatif kecil. Namun berdasarkan implementasi perangkat lunak, diketahui bahwa Paramiko memiliki beberapa kekurangan dalam implementasinya, karena Paramiko tidak dapat mendeteksi kapan suatu perintah yang dikirimkan kepada alat jaringan telah selesai dilakukan, sehingga *library* Paramiko memerlukan suatu cara untuk menghambat pengiriman perintah agar tidak terjadi konflik pada saat suatu perintah belum selesai dijalankan.
3. *Library* Netmiko menyediakan kemudahan dalam mengimplementasi automasi karena pada Netmiko sudah dilengkapi dengan fitur untuk mendeteksi kapan perintah yang dikirimkan kepada alat jaringan sudah selesai dijalankan. Namun berdasarkan hasil pengujian, *library* Netmiko memiliki performa yang kurang baik terlihat dari lama waktu proses data yang dilakukan oleh Netmiko.
4. Secara keseluruhan, *library* Paramiko unggul dalam aspek performa ukuran data yang dikirimkan maupun yang diterima kembali. Sedangkan untuk aspek performa lama proses data antara *library* Paramiko dan NAPALM cukup sebanding dan tidak jauh berbeda. *Library* Netmiko memiliki performa yang buruk dibandingkan dengan *library* Paramiko dan NAPALM.

#### 6.2 Saran

Berdasarkan implementasi dan hasil pengujian, terdapat beberapa hal yang dapat dijadikan sebagai saran untuk penelitian selanjutnya, yaitu:

1. Melakukan penelitian dengan menggunakan alat jaringan nyata yang mendukung protokol NETCONF dan RESTCONF
2. Melakukan implementasi automasi alat jaringan dengan menggunakan protokol NETCONF dan RESTCONF.

## DAFTAR REFERENSI

- [1] Mihăilă, P., Bălan, T., Curpen, R., dan Sandu, F. (2017) Network automation and abstraction using python programming methods. *MACRo 2015*, **2**, 95–103.
- [2] Empson, S. (2008) *CCNA portable command guide*, 2nd edition. Cisco Systems, Inc., Indianapolis.
- [3] Leinwand, A. dan Pinsky, B. (2000) *Cisco router configuration*, 2nd edition. Cisco Press, Indianapolis.
- [4] George, N. (2016) *Mastering Django: Core*. Packt Publishing Ltd., Birmingham.
- [5] Chappell, L. dan Combs, G. (2013) *Wireshark 101: Essential skills for network analysis*, 2nd edition. Protocol Analysis Institute, Chappell University, Reno, Nevada.
- [6] Edelman, J., Lowe, S. S., dan Oswalt, M. (2018) *Network Programmability and Automation: Skills for the Next-Generation Network Engineer*. O'Reilly Media, Inc., Sebastopol.
- [7] Holzmann, G. J. (1991) *Design and Validation of Computer Protocols*. Prentice-Hall, Inc., Englewood Cliffs.
- [8] Barrett, D. J., Silverman, R. E., dan Byrnes, R. G. (2005) *SSH, the Secure Shell: The Definitive Guide*, 2nd edition. O'Reilly Media, Inc., Sebastopol.
- [9] Ylönen, T. dan Lonvick, C. (2006) The secure shell (ssh) protocol architecture. RFC 4251. RFC Editor, <http://www.rfc-editor.org>.
- [10] Claise, B., Clarke, J., dan Lindblad, J. (2019) *Network programmability with YANG: the structure of network automation with YANG, NETCONF, RESTCONF, and gNMI*. Addison-Wesley Professional, Boston.
- [11] Enns, R., Bjorklund, M., Schoenwaelder, J., dan Bierman, A. (2011) Network configuration protocol (netconf). RFC 6241. RFC Editor, <http://www.rfc-editor.org>.
- [12] Bjorklund, M. (2010) Yang - a data modeling language for the network configuration protocol (netconf). RFC 6020. RFC Editor, <http://www.rfc-editor.org>.
- [13] Bierman, A., Bjorklund, M., dan Watsen, K. (2017) Restconf protocol. RFC 8040. RFC Editor, <http://www.rfc-editor.org>.