

# Otomatisasi Perangkat Jaringan Komputer Menggunakan Ansible Pada Laboratorium Komputer

M Rifki Afandi<sup>a</sup>, Puspanda Hatta<sup>b</sup>, Agus Efendi<sup>c</sup>

<sup>abc</sup>Program Studi Pendidikan Teknik Informatika dan Komputer Universitas Sebelas Maret. Jl.Ir. Sutami 36 A, Surakarta, Jawa Tengah 57126

email: <sup>a</sup>[rifki.affandi3@gmail.com](mailto:rifki.affandi3@gmail.com), <sup>b</sup>[hatta.puspanda@staff.uns.ac.id](mailto:hatta.puspanda@staff.uns.ac.id), <sup>c</sup>[agusefendi@staff.uns.ac.id](mailto:agusefendi@staff.uns.ac.id)

**Abstract**—*Network automation is a solution in implementing some complicated or repetitive work. The standard configuration method on network devices performs by entering a network device. That looks easy to configure one or two network devices. that will look not very easy if one must configure many network devices such as routers and servers. Ansible served as a network automation tool on a large and small scale on a computer network, and Ansible served to simplify configuring network devices.*

**Index Terms**— *Network Automation, Ansible, Computer Laboratories, Network Device*

**Abstrak**—*Otomatisasi jaringan merupakan sebuah solusi dalam mengimplementasikan beberapa pekerjaan yang rumit atau berulang. Dalam metode tradisional konfigurasi pada perangkat jaringan dilakukan dengan masuk ke dalam perangkat jaringan untuk mengkonfigurasi perangkat jaringan tersebut, pekerjaan tersebut terlihat mudah bila yang dikonfigurasi masih satu atau dua perangkat jaringan, hal ini akan terlihat rumit jika terdapat banyak perangkat jaringan seperti router dan server yang harus dikonfigurasi. Ansible bertugas sebagai alat otomatisasi jaringan pada skala besar dan kecil pada sebuah jaringan komputer, serta Ansible bertugas untuk mempermudah pekerjaan dalam mengkonfigurasi perangkat jaringan.*

**Kata Kunci**—*Otomatisasi Jaringan, Ansible, Laboratorium Komputer, Perangkat Jaringan*

## I. PENDAHULUAN

Secara umum laboratorium komputer merupakan layanan yang harus ada pada instansi Pendidikan. Salah satu tujuan adanya laboratorium komputer adalah penunjang proses belajar mengajar dalam instansi Pendidikan. Untuk itu laboratoium komputer harus memenuhi kebutuhan penggunanya dengan pelayanan yang berkualitas[1]. Laboratorium komputer yang baik tidak hanya sekedar memenuhi pelayanan fasilitas dari segi fisik, namun juga harus mampu memenuhi ekpektasi pengguna dari segi non fisik seperti kualitas akses informasi serta aplikasi – aplikasi yang terdapat dalam komputer pada laboratorium tersebut.

Beberapa masalah yang sering ditemui pada laboratorium komputer ini ialah 1) Jika ditemukan *update* pada suatu *software* maka otomatis setiap komputer yang ada pada laboratorium tersebut harus di-*update* satu persatu, 2) Apabila ada router lebih dari satu dan konfigurasi router tersebut sama persis dan diharuskan mengkonfigurasi router tersebut satu persatu, dan masih banyak lagi[2].

Solusi yang dapat memecahkan masalah yang dijabarkan kemungkinan besar diterapkannya penggunaan SSH pada tiap perangkat agar memudahkan melakukan konfigurasi tanpa berpindah pada tiap perangkat jaringan komputer, akan menjadi masalah apabila dilakukan banyak konfigurasi pada ratusan perangkat jaringan komputer pada laboratorium komputer[3][4].

Dalam beberapa studi kasus yang ada tentang otomatisasi ini pernah dilakukan antara lain dengan Zero Touch Provisioning dan Terraform, akan tetapi dalam kasus tersebut ZTP hanya dapat diaplikasikan dengan beberapa perangkat tertentu (Cisco) dan resource yang mendukung layanan skala besar dan dalam beberapa kasus tersebut membutuhkan alur kerja yang rumit[5][6][7]. Terraform hanya bisa digunakan untuk melakukan otomatisasi infrastruktur cloud, seperti OpenStack, AWS, GCP dan infrastruktur cloud yang lainnya [8]. Karena keterbatasan yang bisa dilakukan oleh ZTP dan Terraform ini maka ansible adalah solusi dalam otomatisasi jaringan yang mana tidak membutuhkan banyak resource dan tidak membutuhkan banyak syarat yang ada ketika mengaplikasikan Ansible[9][10][11][12].

Ansible bertugas menjadi alat otomatisasi yang akan menghapus beberapa pekerjaan yang harusnya dilakukan berulang[13][14] di banyak server tapi tidak dilakukan karena memakai Ansible seperti

layaknya konfigurasi, instalasi dan lain – lain[15][16]. Ansible merupakan alat otomatisasi jaringan yang mulai dikenal banyak orang, dibanding dengan alat otomatisasi jaringan lain seperti CHEF, dan PUPPET tapi untuk keduanya dokumentasi pengelolaan masih sangat sedikit bila dibandingkan dengan Ansible.

## II. METODE

Network Development Life Cycle (NDLC) merupakan sebuah metode dalam membangun suatu jaringan dalam suatu tempat untuk mengembangkan atau merancang topologi jaringan untuk mengetahui statistic dan kinerja jaringan dengan pemantauan jaringan[17]. NDLC mempunyai 6 tahapan, Adapun beberapa tahapan NDLC sebagai berikut:

### 1. Analisis

Tahap awal ini merupakan tindakan analisis dari analisa kebutuhan, analisa permasalahan yang muncul, analisa keinginan pengguna, dan analisa topologi jaringan yang sudah ada.

### 2. Desain

Tahap desain adalah tahap dimana topologi jaringan dibuat yang mana diharapkan dari desain ini sudah menggambarkan seutuhnya dari kebutuhan yang ada. Desain disini bisa berupa topologi, desain akses data, desain tata layout perkabelan dan sebagainya yang mana itu semua akan memberikan gambaran yang jelas tentang proyek yang akan dibuat.

### 3. Simulasi

Tahap ini merupakan tahap simulasi dari jaringan yang akan dibuat dapat memanfaatkan *tools – tools* khusus di bidang jaringan seperti : GNS3, Cisco Packet Tracer dan lain – lain

### 4. Implementasi

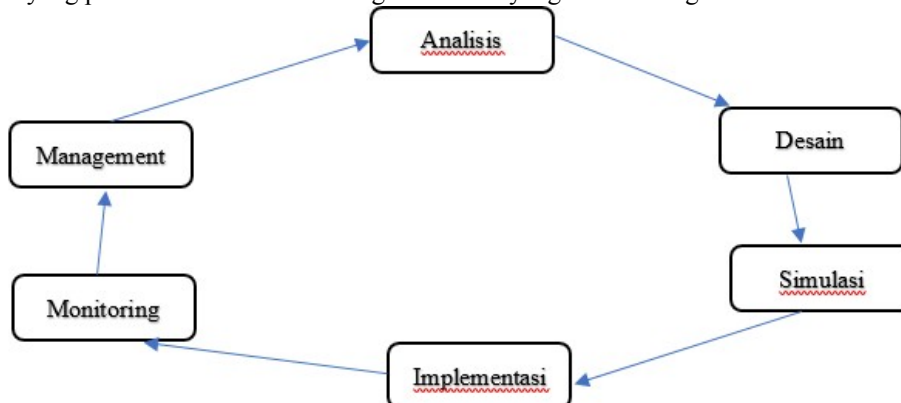
Tahapan kali ini akan memakan waktu yang lebih lama dari tahapan sebelumnya, tahapan ini merupakan aksi nyata dari simulasi yang telah dilakukan.

### 5. Monitoring

Setelah tahapan implementasi selesai dilakukan monitoring yang mana tahapan ini memonitor jaringan dan komunikasi agar sesuai dengan keinginan awal.

### 6. Management

Manajemen atau pengaturan menjadi salah satu perhatian khusus karena terdapat masalah aturan, kebijakan yang perlu dilakukan untuk mengatus sistem yang telah dibangun.



Gambar 1. Silkus Network Development Life Cycle

## III. HASIL DAN PEMBAHASAN

### A. Analisis

Hasil analisis dari kebutuhan untuk membangun perangkat dan topologi jaringan komputer pada laboratorium komputer Pendidikan Teknik Informatika dan Komputer FKIP UNS setelah dianalisa sebagai berikut:

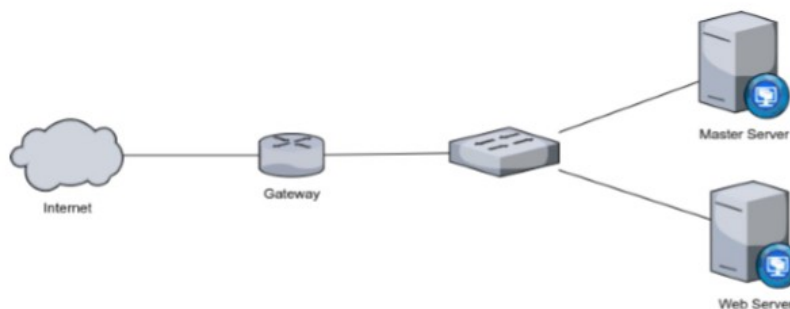
Tabel 1. Kebutuhan Barang

No.	Nama Barang	Qty
1.	PC Server	2
2.	Switch	2

No.	Nama Barang	Qty
3.	Kabel RJ45	8
4.	Keyboard	1
5.	Router Mikrotik	1
6.	Mouse	1

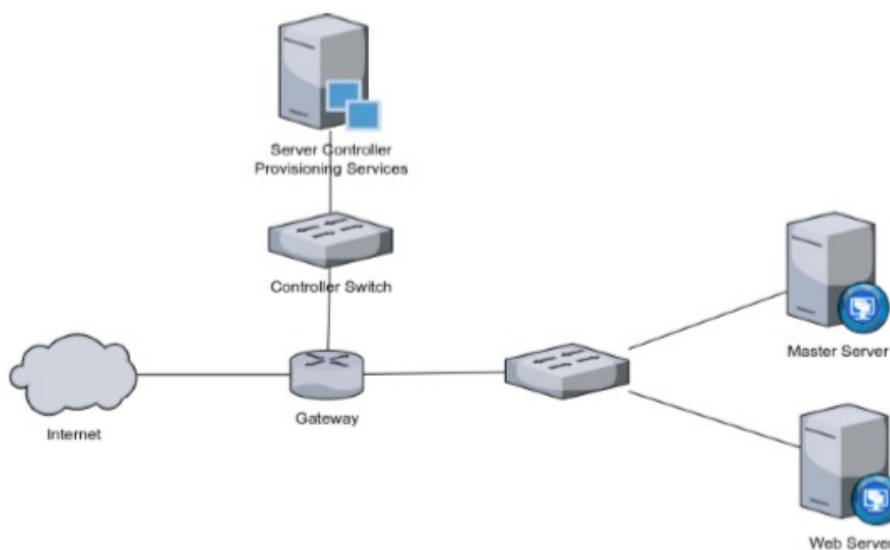
**B. Desain**

Setelah dibuat analisis tentang kebutuhan barang, dilanjutkan dengan desain topologi jaringan. Desain dari topologi jaringan yang sudah diotomatisasi dan sebelum diotomatisasi pada gambar 2 dan gambar 3



Gambar 2 Topologi Jaringan Sebelum Otomatisasi

Pada gambar 2 terlihat bahwa topologi jaringan belum memiliki server otomatisasi sehingga dalam pengelolaan dibutuhkan untuk mengkonfigurasi pada masing-masing perangkat jaringan komputer. Sedangkan pada gambar 3 ditambahkan server otomatisasi yang berfungsi sebagai pusat konfigurasi untuk mengirimkan konfigurasi pada perangkat jaringan yang dituju.



Gambar 3 Topologi Jaringan Setelah Otomatisasi

**C. Simulasi**

Sebelum diterapkan pada tahap implementasi, mekanisme otomatisasi ini jaringan diujikan terlebih dahulu dalam bentuk simulasi purwarupa. Simulasi purwarupa dilakukan pada lingkungan virtualisasi perangkat keras dengan bantuan tools Oracle VirtualBox. Adapun spesifikasi komputer yang digunakan untuk pengujian simulasi melalui VirtualBox ini mempunyai spesifikasi 16 GB RAM.

Tabel 2. Spesifikasi Node Simulasi

No.	Nama Node	Spesifikasi
1.	Master Server	4 GB RAM, 40 GB dan 2 Processor
2.	Router Mikrotik	64 MB RAM, 20 GB dan 1 Processor

#### D. Implementasi

Implementasi dimulai ketika tahap simulasi prototipe tidak mengalami kegagalan dan gangguan sama sekali. Pada tahap ini PC server menggunakan Ubuntu 18.04 sebagai sistem operasi default untuk menjalankan Ansible, dengan memanfaatkan 1 PC di laboratorium komputer PTIK FKIP UNS untuk dijadikan sebagai server.

Server yang ada diinstall Ubuntu 18.04 minimal install. Proses install ini membutuhkan sekitar 30 menit ditahap instalasi. Topologi yang dipakai sama dengan yang telah direncanakan sebelumnya pada gambar 2, setelah semua alat – alat yang dibutuhkan terpenuhi maka tahap implementasi dapat dijalankan.

Ansible diinstall pada server master kemudian dilakukan tahap instalasi ssh key pair untuk menghubungkan antara server master dan server lain, tak lupa juga memberikan ssh key pair ke router mikrotik agar router dapat terhubung ke server master. Setelah siap semua tahap otomatisasi dengan Ansible dapat dilaksanakan. Otomatisasi perangkat jaringan ini menggunakan Ansible Playbook untuk mengkonfigurasi masing – masing host. Berikut contoh Ansible Playbook sederhana terdapat pada Tabel 3.

Tabel 3 Ansible Playbook Mikrotik

```
---
- hosts: r1
  connection: network_cli
  gather_facts: no
  tasks:
  - name: Menambahkan IP Address Ether2
    routeros_command:
      commands: /ip address add address=192.168.10.1/24 interface=ether2
```

Pada bagian “routeros command” terdapat perintah untuk menambahkan ip address pada router mikrotik, apabila ingin menambah command hal tersebut sangat mungkin dilakukan dengan menambahkan beberapa command mikrotik yang diinginkan.

Pada bagian server terdapat pula Ansible Playbook yang harus dijalankan, untuk melakukan instalasi paket – paket untuk web server seperti contoh pada tabel 4 berikut

Tabel 4 Ansible Playbook untuk web server

```
---
- name: install and start httpd
  hosts: web
  tasks:
  - name: install epel repo
    yum: name=epel-release state=latest
  - name: install httpd
    yum: name=httpd state=latest
    notify: restart httpd
  - name: start httpd service
    service: name=httpd state=started enabled=yes
```

Pada bagian hosts terdapat keterangan “web”, hal tersebut dimaksudkan untuk hanya mengkonfigurasi host web dan host lainnya tidak ikut terkonfigurasi.

#### E. Tahap Pengujian

Tahap pengujian uji fungsionalitas ini didasarkan pada setting dasar mikrotik dan setting dasar penginstalan paket web server, adapun yang membedakan dengan setting mikrotik dan penginstalan paket web server pada umumnya ini ialah pada saat proses konfigurasi, jika pada umumnya setting dasar mikrotik dan penginstalan web server dengan cara tradisional yakni mikrotik menggunakan winbox dan web server dengan langsung melakukan konfigurasi melalui mesin yang diinstall dengan web server tersebut. Hal ini tidak dilakukan sepenuhnya dalam otomatisasi perangkat jaringan.

Beberapa hal yang menjadi acuan untuk pengujian keberhasilan fungsional pada perangkat jaringan otomatisasi menjadi 15 aspek penilaian.

Tabel 5 Pengujian Fungsional Otomatisasi Perangkat Jaringan

No	Aspek Pengujian	Hasil Pengujian		Keterangan
		Berhasil	Tidak Berhasil	
1.	Semua server dapat terkoneksi ke Internet	Berhasil		
2.	Ping antar server berhasil server master dapat menjalankan role Ansible	Berhasil		
3.	server master dapat menjalankan role Ansible	Berhasil		
4.	SSH antar server berhasil	Berhasil		

No	Aspek Pengujian	Hasil Pengujian		Keterangan
		Berhasil	Tidak Berhasil	
5.	Server ter-cluster dengan server lain.	Berhasil		
6.	Generate SSH dari server master ke slave server	Berhasil		
7.	Menginstall paket web server dari server master menggunakan Ansible	Berhasil		
8.	Menjalankan Playbook	Berhasil		
9.	Menjalankan paket web server dari server master	Berhasil		
10.	Cek konfigurasi server lain melalui server master	Berhasil		
11.	Ping router dan server master	Berhasil		
12.	Konfigurasi IP Address pada Mikrotik dapat dilakukan melalui server master	Berhasil		
13.	Konfigurasi DHCP server via server master	Berhasil		
14.	Konfigurasi NAT pada router mikrotik melalui server master	Berhasil		
15.	Menambahkan User baru melalui server master	Berhasil		

Dari hasil pengujian fungsional yang dilakukan baik untuk otomatisasi router dan server berjalan dengan baik. Ansible telah melakukan tugasnya untuk mengotomatisasi perangkat jaringan komputer di laboratorium komputer tanpa mengalami kegagalan. Bukan hanya pada server saja, akan tetapi otomatisasi dapat dilakukan pada router mikrotik yang telah dikonfigurasi sedemikian rupa sehingga bisa diotomatisasi dengan Ansible.

#### IV. KESIMPULAN

Dari hasil penelitian ini dapat disimpulkan beberapa hal, antara lain, otomatisasi jaringan dengan memanfaatkan Ansible sangat memungkinkan untuk dilakukan dengan tujuan untuk mengotomatisasi perangkat jaringan seperti router, server dan lain – lain. Ansible sudah melaksanakan tugasnya dengan baik tanpa adanya kesalahan dalam pengoperasiannya.

Dengan memanfaatkan Ansible pekerjaan yang harus dilakukan secara berulang tentu akan menghilang dan dilakukan dengan satu skrip untuk menyelesaikan pekerjaan tersebut. Diharapkan untuk penelitian selanjutnya lebih memanfaatkan Ansible lebih luas lagi seperti mengkonfigurasi melalui web atau mobile.

#### DAFTAR PUSTAKA

- [1] L. Hidayat, M. Mulyana, and M. Effendy, "Membangun Kepuasan Mahasiswa Pengguna Laboratorium Komputer," *JAS-PT J. Anal. Sist. Pendidik. Tinggi*, vol. 1, no. 2, p. 93, 2018, doi: 10.36339/jaspt.v1i2.87.
- [2] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "DevOps," *IEEE Softw.*, vol. 33, no. 3, pp. 94–100, 2016, doi: 10.1109/MS.2016.68.
- [3] W. F. Senjaya, T. Witono, and N. Alkhala, "Perancangan dan Evaluasi Usability Aplikasi Pengelolaan Laboratorium Komputer," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 3, no. 2, p. 101, 2017, doi: 10.20473/jisebi.3.2.101-106.
- [4] A. Kurniawan, H. N. Palit, and J. Adjarwirawan, "Eksplorasi Pemanfaatan Docker untuk Mempermudah Pengelolaan Instalasi Komputer di Laboratorium Komputer Teknik Informatika Universitas Kristen Petra," *Infra*, vol. 4, no. 2, pp. 2–7, 2016.
- [5] B. M. Andrus *et al.*, "Zero-touch provisioning of distributed video analytics in a software-defined metro-haul network with P4 processing," *Opt. InfoBase Conf. Pap.*, vol. Part F160-, pp. 1–3, 2019, doi: 10.1364/ofc.2019.m3z.10.
- [6] R. Vilalta *et al.*, "Zero-Touch Network Slicing Through Multi-Domain Transport Networks," *Int. Conf. Transparent Opt. Networks*, vol. 2018-July, pp. 2–5, 2018, doi: 10.1109/ICTON.2018.8473898.
- [7] Y. Demchenko *et al.*, "ZeroTouch Provisioning (ZTP) Model and infrastructure components for multi-provider cloud services provisioning," *Proc. - 2016 IEEE Int. Conf. Cloud Eng. Work. IC2EW 2016*, pp. 184–189, 2016, doi: 10.1109/IC2EW.2016.50.
- [8] L. R. De Carvalho and A. P. F. De Araujo, "Performance Comparison of Terraform and Cloudify as Multicloud Orchestrators," *Proc. - 20th IEEE/ACM Int. Symp. Clust. Cloud Internet Comput. CCGRID 2020*, pp. 380–389, 2020, doi: 10.1109/CCGrid49817.2020.00-55.
- [9] M. Artac, T. Borovssak, E. Di Nitto, M. Guerriero, and D. A. Tamburri, "DevOps: Introducing infrastructure-as-code," *Proc. - 2017 IEEE/ACM 39th Int. Conf. Softw. Eng. Companion, ICSE-C 2017*, pp. 497–498, 2017, doi: 10.1109/ICSE-C.2017.162.
- [10] Markus Borgenstrand, "Network Automation – the power of Ansible," p. 54, 2018.
- [11] I. P. Hariyadi and K. Marzuki, "Implementation Of Configuration Management Virtual Private Server Using Ansible," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 19, no. 2, pp. 347–357, 2020, doi: 10.30812/matrik.v19i2.724.
- [12] O. Lavriv, M. Klymash, G. Grynkevych, O. Tkachenko, and V. Vasylenko, "Method of cloud system disaster recovery based on 'Infrastructure as a code' concept," *14th Int. Conf. Adv. Trends Radioelectron. Telecommun. Comput. Eng.*

- TCSET 2018 - Proc.*, vol. 2018-April, pp. 1139–1142, 2018, doi: 10.1109/TCSET.2018.8336395.
- [13] W. yiran PetroChina, Z. tongyang PetroChina, and G. yidong PetroChina, “Design and implementation of continuous integration scheme based on Jenkins and Ansible,” *2018 Int. Conf. Artif. Intell. Big Data*, pp. 245–249, 2018, doi: 10.1109/ICAIBD.2018.8396203.
- [14] S. A. I. B. S. Arachchi and I. Perera, “Continuous integration and continuous delivery pipeline automation for agile software project management,” *MERCon 2018 - 4th Int. Multidiscip. Moratuwa Eng. Res. Conf.*, pp. 156–161, 2018, doi: 10.1109/MERCon.2018.8421965.
- [15] C. T. Solution, “Ansible and Docker: the Best Combination for an Efficient Software Product Management,” *Jun 28, 2017*, 2017. [https://medium.com/@cabot\\_solutions/ansible-and-docker-the-best-combination-for-an-efficient-software-product-management-28c86cfebe90](https://medium.com/@cabot_solutions/ansible-and-docker-the-best-combination-for-an-efficient-software-product-management-28c86cfebe90) (accessed Dec. 30, 2018).
- [16] J. Keating, *Mastering Ansible*. PacktPub, 2015.
- [17] K. Rianafirin and M. T. Kurniawan, “Design network security infrastructure cabling using network development life cycle methodology and ISO/IEC 27000 series in Yayasan Kesehatan (Yakes) Telkom Bandung,” *Proc. 2017 4th Int. Conf. Comput. Appl. Inf. Process. Technol. CAIPT 2017*, vol. 2018-Januari, pp. 1–6, 2018, doi: 10.1109/CAIPT.2017.8320681.

**M Rifki Affandi** merupakan mahasiswa Pendidikan Teknik Informatika dan Komputer Fakultas Keguruan dan Ilmu Pendidikan Universitas Sebelas Maret yang juga turut mendalami ilmu *Cloud Computing*, *Network Automation*, *DevOps* dan *Linux Administration*. Aktif di beberapa komunitas Open Source di Indonesia, seperti openSUSE Indonesia, BlankOn Linux, OpenStack Indonesia dan masih banyak lagi.

Puspanda Hatta, Meraih gelar sarjana komputer (S.Kom) dari Universitas Sebelas Maret pada tahun 2011. Kemudian meraih gelar Engineering (M.Eng) dari Universitas Gadjah Mada pada tahun 2014. Saat ini Penulis menjadi dosen Pendidikan Teknik Informatika dan Komputer Fakultas Keguruan dan Ilmu Pendidikan Universitas Sebelas Maret.

Agus Efendi, Meraih gelar sarjana (Drs) dari Universitas Negeri Surabaya pada tahun 1992. Kemudian meraih gelar Master (M.Pd) dari Universitas Negeri Yogyakarta pada tahun 1999. Kemudian meraih gelar doktor (Dr) dari Universitas Sebelas Maret. Saat ini Penulis menjadi dosen Pendidikan Bahasa dan Sastra Daerah Fakultas Keguruan dan Ilmu Pendidikan Universitas Sebelas Maret.