

**PENGEMBANGAN *WEB CONFERENCING* BERBASIS
BIGBLUEBUTTON UNTUK MENINGKATKAN KOLABORATIF
LEARNING MAHASISWA PTIK UNS**



SKRIPSI

Oleh :

**MOHAMMAD RIFKI AFFANDI Z
K3515037**

**FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS SEBELAS MARET
SURAKARTA**

2019

DAFTAR ISI

HALAMAN JUDUL.....
DAFTAR ISI.....	i
DAFTAR GAMBAR.....	ii
DAFTAR TABEL.....	iii
BAB 1 PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	2
C. Tujuan Pengembangan.....	2
D. Spesifikasi produk yang diharapkan.....	3
E. Pentingnya Pengembangan.....	3
F. Asumsi dan Keterbatasan Pengembangan.....	3
G. Definisi Istilah.....	4
BAB 2 KAJIAN PUSTAKA.....	6
A. KAJIAN TEORI.....	6
1. Server.....	6
2. Ansible.....	8
3. DevOps.....	10
4. Software Defined Network Networking (SDN).....	12
5. Linux.....	14
6. Secure Shell (SSH).....	15
7. CentOS.....	16
8. Jaringan Komputer.....	17
9. Perangkat Keras Jaringan Komputer.....	19
B. KERANGKA BERPIKIR.....	21
BAB 3 Metode Penelitian.....	23
A. Tempat dan Waktu Penelitian.....	23
1. Tempat Penelitian.....	23
2. Waktu Penelitian.....	23
B. Metode Penelitian.....	24
C. Prosedur Penelitian.....	24
1. Tahap <i>Analysis</i>	25
2. Tahap <i>Design</i>	26
3. Tahap <i>Simulation Prototype</i>	27
4. Tahap <i>Implementation</i>	27
D. Teknik Pengujian.....	27
DAFTAR PUSTAKA.....	29

DAFTAR GAMBAR

Gambar 2.1: Arsitektur Ansible.....	8
Gambar 2.2: Perintah Ad-Hoc di Ansible untuk test ping.....	9
Gambar 2.3: Contoh playbook ketika ingin membuat web.....	10
Gambar 2.4: Siklus DevOps.....	12
Gambar 2.5: Arsitektur SDN (Software Defined Networking).....	14
Gambar 2.6: Cara Kerja SSH.....	16
Gambar 2.7: Release history RHEL.....	16
Gambar 2.8: Macam - macam jaringan komputer.....	18
Gambar 2.9: Perangkat Keras Jaringan Komputer.....	21
Gambar 2.10: Kerangka berpikir.....	22
Gambar 3.1: NDLC Siklus.....	25
Gambar 3.2: Topologi Jaringan.....	26

DAFTAR TABEL

Table 3.1: Waktu Penelitian.....	23
Table 3.2: Kebutuhan.....	25

BAB 1

PENDAHULUAN

A. Latar Belakang

E-Learning merupakan salah satu bentuk pendidikan yang bersifat jarak jauh, yang mana kegiatan belajar mengajar dilakukan melalui media internet (Suyanto, 2005). Rosenberg (2001) menekankan bahwa e-learning merujuk pada tata penggunaan internet dalam mengirimkan serangkaian solusi dalam meningkatkan pengetahuan serta keterampilan. Adapun hal ini merupakan senada dengan Cambell (2002), Kamarga (2002) yang intinya menekankan penggunaan internet dalam pendidikan dalam hakekat e-learning.

Teknologi pembelajaran terus berkembang. Sedangkan pada prinsipnya teknologi tersebut dikelompokkan menjadi dua, yaitu : *Technology based learning* dan *Technology based web-learning*. Adapun *based learning* merupakan *Audio Information Technologies* (radio, audio tape, voice mail telephone) dan *Video Information Technologies* (video tape, video text, video messaging). Sedangkan *web-learning* pada dasarnya ialah *Data Information Technologies* (buletin board, internet, e-mail, tele-collaboration) (Suyanto, 2005).

Zaman modern seperti sekarang ini penggunaan internet di Indonesia sudah sangatlah mudah, serta jaringan yang disediakan cukup untuk melakukan *stream* baik itu video ataupun *audio*. Kenyataan yang didapatkan ketika proses pembelajaran terhenti dan tidak dapat diadakan dikarenakan dosen yang berhalangan hadir selalu menjadi satu arah.

Pembelajaran jarak jauh merupakan salah satu solusi untuk mengatasi masalah tersebut yang mana pembelajaran jarak jauh ini mensyaratkan kemandirian belajar untuk peserta (Tahar & Enceng, 2006).

Berdasarkan pengamatan peneliti selama kuliah di Pendidikan Teknik Informatika dan Komputer FKIP UNS, ada beberapa mata kuliah yang terkendala

dengan berbagai faktor, mulai dari dosen sedang melakukan dinas, mata kuliah yang kadang jadwalnya tubrukkan dengan mata kuliah lain, baik itu dari satu angkatan maupun angkatan yang lain, dan lain – lain.

Adapun solusi yang penulis terapkan disini ialah membangun sebuah media sebagai jembatan agar proses kegiatan belajar mengajar di Pendidikan Teknik Informatika dan Komputer FKIP UNS tidak terganggu dan tidak terundur. Solusi yang peneliti kembangkan ialah merupakan web conferencing, yang mana sebagai jembatan kegiatan belajar mengajar antara dosen dan mahasiswa yang berjalan secara langsung dan terkomunikasi dua arah, yang memanfaatkan teknologi dan internet.

Dari masalah itulah maka timbul penelitian ini dengan judul “PENGEMBANGAN WEB CONFERENCE BIGBLUEBUTTON UNTUK MENINGKATKAN KOLABORATIF LEARNING MAHASISWA PTIK UNS”. Semoga media ini dapat membantu dalam kegiatan belajar mengajar di Pendidikan Teknik Informatika dan Komputer FKIP UNS.

B. Rumusan Masalah

Adapun rumusan masalah yang didapatkan berdasarkan dengan apa yang ada pada masalah yang melatar belakangi bab ini adalah sebagai berikut :

1. Bagaimana membangun web conferencing untuk PTIK UNS?
2. Bagaimana cara menangani mata kuliah yang tertunda, apabila dosen berhalangan untuk hadir atau sedang dinas ke luar kota?
3. Bagaimana keefektifan kegiatan belajar mengajar dengan menggunakan media web conferencing?

C. Tujuan Penelitian

Maka adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Membangun web conferencing PTIK sebagai media pembelajaran tambahan.
2. Membantu mahasiswa dalam perkuliahan, sehingga tidak ada perkuliahan yang tertunda.

3. Membuat kegiatan belajar mengajar perkuliahan tidak ada yang melebihi target.

D. Manfaat Penelitian

Sebagaimana disebutkan dalam tujuan penelitian diatas, adapun manfaat dari penelitian ini adalah :

1. Dapat membangun sebuah web conferencing yang membantu mahasiswa PTIK dalam mata perkuliahan.
2. Membangun sebuah web conferencing yang bertujuan sebagai pengganti pembelajaran konvensional.

E. Asumsi dan Keterbatasan Pengembangan

1. Asumsi

Asumsi dalam pengembangan penelitian ini adalah sebagai berikut :

- a. Kegiatan belajar mengajar di perkuliahan yang masih terpaut dengan tatap muka secara langsung.
- b. Mahasiswa dapat mengenal salah satu macam dari web conference.

2. Keterbatasan Pengembangan

Dalam penelitian ini memiliki beberapa keterbatasan pengembangan antara lain sebagai berikut :

- a. Hanya mengkaitkan dengan prodi PTIK FKIP UNS angkatan 2016.
- b. Dalam penelitian ini hanya akan menggunakan *dedicated server* yang berpusat di Labkom PTIK FKIP UNS.
- c. Proses pengembangan ini menggunakan *server* fisik dengan PC lab komputer PTIK UNS berjumlah 1 PC.
- d. *Software* yang akan digunakan dalam pengembangan ini berbasis *Open Source*.

F. Definisi Istilah

Dalam penelitian ini terdapat beberapa istilah yang digunakan diantaranya seperti BigBlueButton, Kegiatan Belajar Mengajar, , Otomasi dan Linux.

1. BigBlueButton

Ansible adalah alat otomasi jaringan, dimana Ansible dapat mengkonfigurasi jaringan, menyebarkan update untuk perangkat lunak, dan mengatur tugas – tugas yang biasanya dilakukan manual oleh *engineer* (Geerling, 2015).

2. Web Conferencing

3. *Server*

Server adalah sebuah komputer baik fisik maupun virtual yang mempunyai tugas untuk mengatur, melayani dan membatasi segala sesuatu yang berkaitan dengan akses – akses *client* dalam suatu jaringan komputer (Santoso & Assegaf, 2017).

4. Otomasi

Otomasi adalah teknik dalam mengatur jaringan yang mengadopsi otomasi dan meningkatkan efisiensi dalam konfigurasi jaringan (J Edelman, SS Lowe, 2018).

5. Linux

Linux adalah sebuah sistem operasi komputer yang berlisensi sumber kode terbuka dibawah lisensi *GNU General Public License (GPL)* (Sutarti, Siswanto, & Subandi, 2018).

BAB 2

KAJIAN PUSTAKA

A. KAJIAN TEORI

1. *Server*

a. Pengertian *Server*

Server atau *PC Server* merupakan bagian dari pengembangan jaringan komputer yang sampai saat ini masih sulit dipisahkan. Terutama seperti bagian *client server* yang berada pada sistem jaringan yang terdiri dari beberapa komputer yang kemudian difungsikan sebagai *server* dan komputer lainnya sebagai *client*. Jadi pada dasarnya disini ialah tugas *server* itu sendiri melayani para *client*-nya dalam sebuah jaringan (Wahyono, 2007).

Server juga bisa diartikan dengan “hati” dari jaringan komputer itu sendiri. Karena *server* itu sendiri biasanya menangani puluhan *client* maka spesifikasi yang dibutuhkan untuk *server* ini bisa dikatakan merupakan komputer berspesifikasi tinggi dengan RAM yang besar dan penyimpanan *storage* yang besar dan tidak lupa dengan internet dengan kecepatan tinggi. *Server* sendiri terdiri dari berbagai macam *server* tergantung dari fungsi dan akan digunakan untuk apa *server* tersebut seperti *web server*, *DNS Server*, *mail server* dan masih banyak lagi (Informatika, Sains, & Akprind, 2018).

Adapun jenis – jenis *server* antara lain (Ray, 2015) :

1. *Server Platform* : *Server Platform* adalah sebuah *hardware* atau *software* dasar untuk sistem yang bertugas sebagai mesin yang menggerakkan *server*.
2. *Server Aplikasi* : biasa dikenal dengan istilah *middleware*, berfungsi sebagai pengolah perintah dari *client* dan *database*.
3. *Server Audio / Video* : *Server* ini menyediakan kemampuan multimedia ke situs *web* dengan membantu menyiarkan *streaming* konten multimedia untuk pengguna.

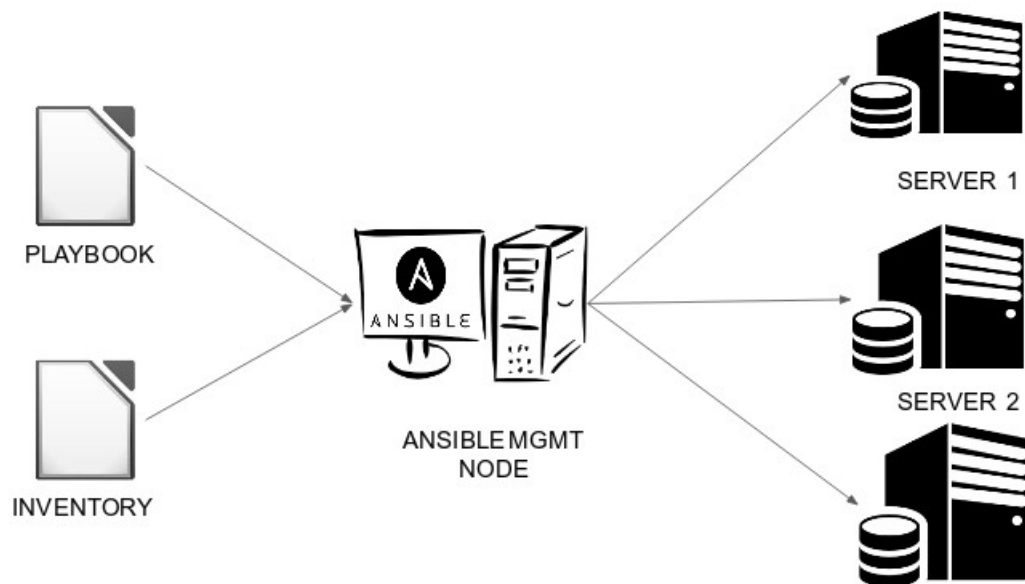
4. *Chat Server* : *server* yang bertanggung jawab menangani *chatting* beberapa *user*.
5. *Fax Server* : bertugas untuk mengirimkan dan menerima fax dalam jaringan komputer.
6. *FTP Server* : bertugas untuk mentransfer data, dari atau menuju *client*.
7. *Groupware Server* : merupakan sebuah *software* yang dirancang agar memungkinkan adanya kerja sama antara pengguna terlepas dari lokasi, yang terhubung dalam *internet* atau *intranet* perusahaan dan berfungsi bersama – sama dalam suasana *virtual*.
8. *IRC Server* : Internet Relay Chat Server (IRC Server) yang bertujuan untuk user berdiskusi dalam jaringan.
9. *List Server* : bertugas untuk mengelola milis.
10. *Mail Server* : *Server* yang bertugas untuk menangani surel.
11. *News Server* : berfungsi sebagai sumber pengiriman berbagai macam berita untuk banyak kelompok.
12. *Proxy Server* : *server* yang berfungsi sebagai mediator antara *client* dan *server* diluar jaringan yang bertugas untuk menyaring dan membatasi permintaan, meningkatkan kinerja koneksi dan *sharing*.
13. *Telnet Server* : untuk masuk ke komputer *host* (utama) dengan cara remot dari komputer lain diluar jaringan.
14. *Virtual Server* : sebuah metode untuk mengurangi jumlah *server* fisik yang bertujuan untuk mengurangi biaya *cost*.
15. *Web Server* : menyediakan konten *web browser*, HTTP (*HyperText Transfer Protocol*) digunakan untuk berkomunikasi dengan *web server*.

2. Ansible

b. Pengertian Ansible

Ansible adalah alat otomatisasi jaringan, dimana Ansible dapat mengkonfigurasi jaringan, menyebarkan update untuk perangkat lunak, dan mengatur tugas – tugas yang biasanya dilakukan manual oleh *engineer* (Geerling, 2015). Ansible sangat membantu terutama untuk *Network Engineer* dalam mengkonfigurasi *server* baik itu dua *server* atau bahkan ratusan *server*; didalam tugasnya Ansible berfungsi untuk membuat otomatisasi didalam *server*; jadi dalam hal ini *Network Engineer* tidak perlu melakukan pekerjaan yang sama seperti melakukan update berulang – ulang ke masing – masing *server*, tapi cukup dilakukan di satu *server* utama dan perintah tersebut akan serta merta dijalankan ke berbagai *server* yang terhubung dengan *server* utama. *Server* Ansible umumnya melakukan tugasnya dengan menyebarkan berbagai macam perintah yang dikirimkan atau update perangkat lunak di sumber daya lingkungan produksi (PetroChina, PetroChina, & PetroChina, 2018).

Untuk struktur Ansible akan dijelaskan pada gambar 2.1 berikut ini



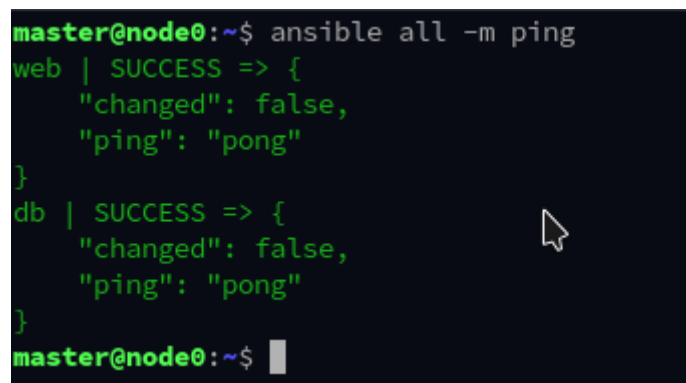
Gambar 2.1: Arsitektur Ansible

Pada gambar diatas terdapat satu buah master node yang berfungsi sebagai pengelola yang mana sudah diatur sedemikian rupa agar node master ini mampu mengatur dan mengotomatisasi *server – server* yang ada.

c. Ad-Hoc

Ad-Hoc adalah suatu baris perintah dalam Ansible, yang mana perintah ini dapat langsung dijalankan tanpa perlu menyimpannya ke suatu file tertentu. Ad-Hoc ini sangat membantu anda untuk memahami bagaimana Ansible bekerja untuk pertama kali. Ad-Hoc ini sangat dianjurkan untuk memahami Ansible sebelum mempelajari Ansible Playbook (Hardion et al., 2013).

Dalam praktiknya Ad-Hoc itu adalah sebuah kumpulan perintah yang dijalankan melalui *command line* yang biasanya akan diawali dengan perintah seperti gambar 2.2 berikut :



```
master@node0:~$ ansible all -m ping
web | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
db | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
master@node0:~$
```

Gambar 2.2: Perintah Ad-Hoc di Ansible untuk test ping

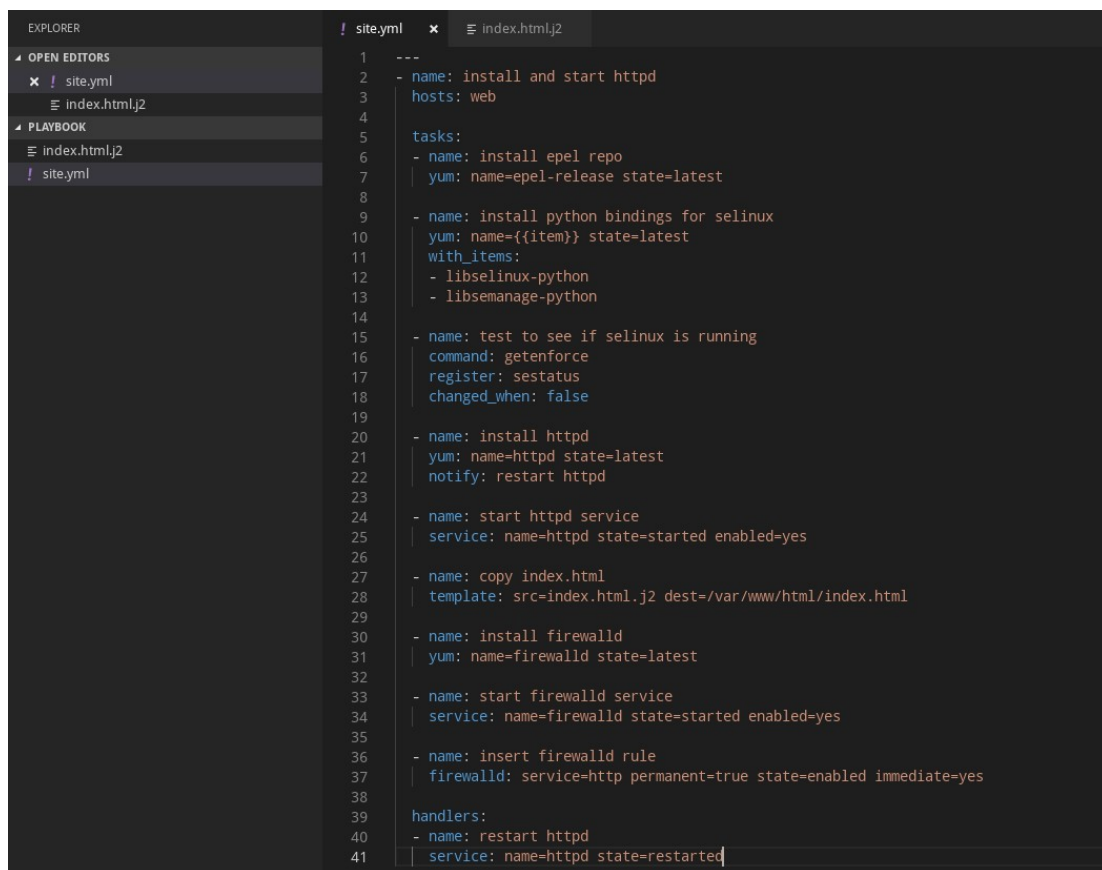
pada perintah tersebut terlihat *server* master sedang mencoba test ping ke dua *node server* yang lain yaitu web dan db (*database*). Dan setelah perintah “ansible all -m ping dijalankan” maka muncul balasan dari dua *server* yang dikendalikan oleh master *node*.

d. Playbook

Playbook adalah tulisan berbasis kode dengan format YAML yang mana kode tersebut ditulis di sebuah file yang bertujuan untuk menjaga agar tindakan pengetikkan tidak dilakukan secara berulang seperti Ad-Hoc (Hardion et al., 2013).

Pada dasarnya Playbook sendiri merupakan bahasa konfigurasi, penyebaran dan orkestrasi pada Ansible. Playbook sendiri terdiri dari satuan instruksi yang dapat dibaca oleh manusia, Playbook ini sendiri memiliki beberapa fungsi tertentu seperti misalnya : mengumpulkan informasi yang berguna, membuat cadangan *file network*, mengelola *file* jaringan, atau memvalidasi konektivitas. Dan adapun Playbook ini dapat dibagikan dan digunakan kembali oleh banyak tim untuk membuat otomasi berulang (Redhat, 2019).

Pada gambar 2.3 berikut adalah merupakan salah satu contoh playbook



```
1 ---
2 - name: install and start httpd
3   hosts: web
4
5   tasks:
6     - name: install epel repo
7       yum: name=epel-release state=latest
8
9     - name: install python bindings for selinux
10      yum: name={{item}} state=latest
11      with_items:
12        - libselinux-python
13        - libsemanage-python
14
15     - name: test to see if selinux is running
16       command: getenforce
17       register: sestatus
18       changed_when: false
19
20     - name: install httpd
21       yum: name=httpd state=latest
22       notify: restart httpd
23
24     - name: start httpd service
25       service: name=httpd state=started enabled=yes
26
27     - name: copy index.html
28       template: src=index.html.j2 dest=/var/www/html/index.html
29
30     - name: install firewall
31       yum: name=firewalld state=latest
32
33     - name: start firewalld service
34       service: name=firewalld state=started enabled=yes
35
36     - name: insert firewalld rule
37       firewallld: service=http permanent=true state=enabled immediate=yes
38
39   handlers:
40     - name: restart httpd
41       service: name=httpd state=restarted
```

Gambar 2.3: Contoh playbook ketika ingin membuat web

3. DevOps

DevOps merupakan sebuah metode yang dapat menjembatani dari dua aksi, yaitu pengembangan (*Develop*) dan pengoperasian (*Operation*), Tetapi pada hal ini DevOps tidak hanya sebatas pada hal tersebut diantaranya juga mencakup semua aspek yang

berkaitan dalam membantu dan mempercepat, mengoptimalkan dan pengiriman paket perangkat lunak berkualitas tinggi. DevOps berfokus pada kecepatan pengiriman, pengujian berkelanjutan pada produksi sebuah perangkat lunak (Technology, 2015).

Adapun beberapa fase yang ada didalam metode DevOps (Singh, Thakur, Chaurasiya, & Nagdev, 2016) antara lain :

A) *Continuous Planning*

Continuous Planning merupakan tindakan yang berupa pemeriksaan sementara yang bertujuan untuk menilai kembali kebutuhan apa saja yang akan dibutuhkan nantinya. Karena setiap kebutuhan setiap waktu dapat berubah sesuai dengan keadaan yang sedang terjadi.

B) *Continuous Integration*

Continuous Integration merupakan tahapan dimana perubahan yang telah lama dilakukan tidak disimpan lama pada suatu tempat. Jadi sebuah perubahan terus mengalami perubahan dan tidak dibiarkan terus menerus terpendam dalam sistem.

C) *Continuous Deployment*

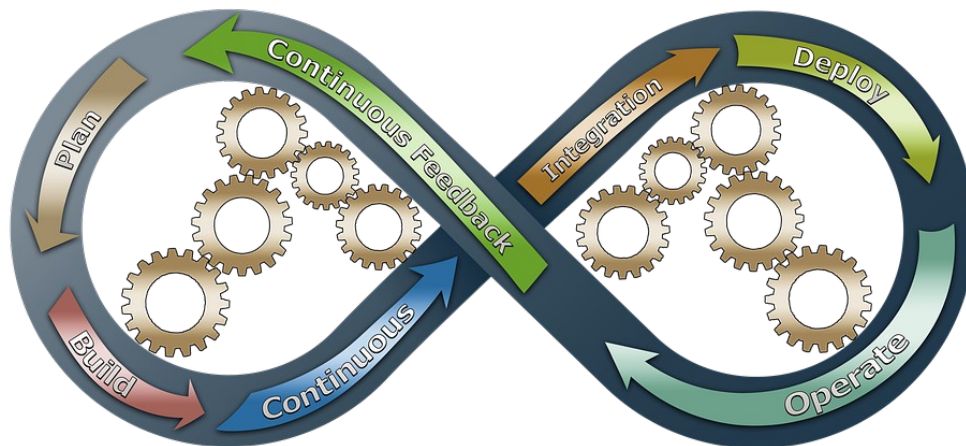
Continuous Deployment adalah bagian penting dari DevOps itu sendiri, yang mana proses *deploy* aplikasi bertindak secara terus menerus dan tidak dilakukan secara manual.

D) *Continuous Testing*

Dalam tahap ini *engineer* diharuskan untuk terus menerus melakukan otomatisasi setiap uji kasus yang mana proses tersebut merupakan proses yang memerlukan pengulangan dari waktu ke waktu.

E) *Continuous Monitoring*

Di bagian ini adalah tahap terakhir dimana mengamati berbagai paramater kualitas secara keseluruhan.



Gambar 2.4: Siklus DevOps

Pada gambar 2.4 diperlihatkan alur dari DevOps yang selalu berputar dan berulang – ulang.

4. *Software Defined Network Networking (SDN)*

a. Pengertian SDN (*Software Defined Networking*)

Konsep utama SDN ialah memusatkan atau sentralisasi jaringan dengan terpusat pada suatu *control plane*. Konsep SDN ini sangat memudahkan operator atau administrator jaringan dalam mengelola jaringan yang dikendalikan (Budi & Haryadi, 2018). Karena dalam implementasinya jaringan yang digunakan saat ini melakukan konfigurasi *node* secara terpisah, dan tujuan dari SDN adalah untuk meniadakan hal tersebut.

b. Arsitektur SDN

Adapun arsitektur SDN ini distandarisasi oleh ONF (*Open Network Foundation*) (Budi & Haryadi, 2018). Karakteristik yang dipunyai SDN seperti memisahkan control plane dan data plane, dipercaya dapat meningkatkan dukungan terhadap jaringan di masa depan.

Arsitektur SDN ini dapat dibagi kedalam 3 layer antara lain :

1. *Infrastructure layer*

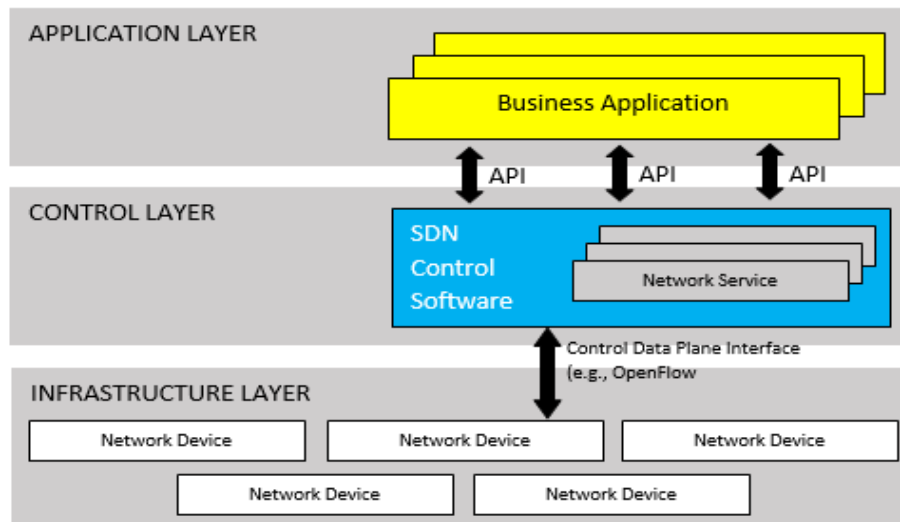
Terdiri dari elemen jaringan yang dapat mengatur SDN Datapath sesuai dengan instruksi yang diberikan melalui *Control-Data-Plane Interface* (CDPI)

2. *Control layer*

Bertugas untuk mentranslasikan kebutuhan aplikasi dengan infrastruktur dengan cara memberikan informasi yang relevan dan dibutuhkan SDN.

3. *Application layer*

Berada pada lapis teratas dalam SDN, berkomunikasi dengan *Northbound Interface* (NBI).



Gambar 2.5: *Arsitektur SDN (Software Defined Networking)*

5. Linux

a. Pengertian Linux

Linux merupakan suatu sistem operasi sumber terbuka (*Open Source*) yang mana memungkinkan semua orang untuk mengubah, menyebarluaskan serta memodifikasinya tanpa melanggar HAKI. Adapun kebebasan dalam Linux ini membuat Linux mudah berkembang dengan cepat dan makin mudah dikembangkan oleh para pengembang yang membuat para pengembang tersebut membuat distronya masing – masing, sehingga pengguna tinggal memilih distro Linux apa yang ingin dipakai, yang mana setiap distro itu mempunyai kelebihan dan kekurangan tersendiri (Sabiq, 2016).

Nama Linux sendiri berasal dari pengembang pertamanya yang bernama Linus Torvalds, yang sebetulnya awal pengembangan Linux ini sendiri bermula pada

kumpulan suatu *software* bersamaan dengan *kernel* untuk menyusun suatu sistem operasi yang lengkap (Raharja, 2001).

b. Sejarah Linux

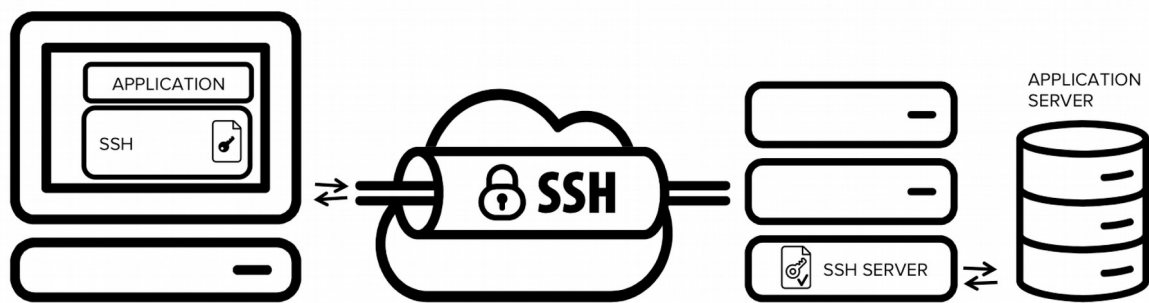
Linus Torvalds, sudah dikenal oleh banyak penggiat *Open Source* dimana dia adalah seorang mahasiswa yang berasal dari Finlandia. Mulanya Linux adalah sebuah proyek yang terinspirasi dari Minix, yakni sistem UNIX kecil yang dikembangkan oleh Andrew Tanenbaum. Pada bulan Agustus tahun 1991 muncullah Linux versi 0.01 kemudian setelah versi 0.01 dirilis Linus mengumumkan versi resmi dari Linux itu sendiri yaitu versi 0.02 dan hanya dapat menjalankan shell bash (GNU Bourne Again Shell) dan gcc (GNU C Compiler) (Raharja, 2001).

Seiring berkembangnya zaman dan proyek Linux ini merupakan proyek sumber terbuka yang mana siapa saja bisa memodifikasi dan membuatnya secara cuma – cuma, perkembangan Linux ini makin lama makin pesat dan kemudian dikenal dengan sebutan Distro Linux. Distro merupakan bundel dari *kernel* Linux , beserta sistem dasar Linux, tools, *program* instalasi, *tools basic*, dan program – program lainnya yang bermanfaat sesuai dengan tujuan dibuatnya distro tersebut.

6. *Secure Shell* (SSH)

Secure Shell atau SSH adalah pendekatan berbasis perangkat lunak yang populer dalam keamanan jaringan dikarenakan keamanannya. SSH akan secara otomatis melakukan enkripsi data dari data yang dikirimkan dalam sebuah jaringan. Dan ketika data telah sampai ditujuan SSH akan secara otomatis melakukan deskripsi.

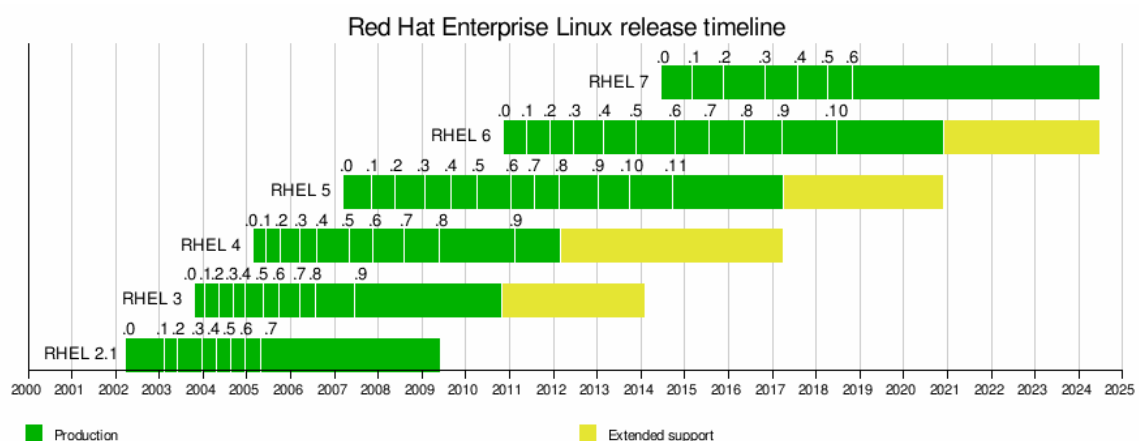
Hasilnya berupa inkripsi yang transparan, maka seorang pengguna atau *user* dapat bekerja seperti biasa tanpa mengetahui bahwa komunikanya telah dienkripsi secara aman oleh jaringan (Ray, 2015).



Gambar 2.6: Cara Kerja SSH

7. CentOS

CentOS singkatan dari *Community ENTERprise Operating System* (Sistem Operasi Perusahaan buatan komunitas/masyarakat). Adalah sistem operasi sumber terbuka berbasis RHEL (*Red Hat Enterprise Linux*). Proyek sistem operasi ini berusaha kompetibel dengan sistem operasi hulunya (RHEL). Dan tentunya paket – paket yang terinstall pada CentOS ini adalah RPM (Pribadi, 2013) .



Gambar 2.7: Release history RHEL

pada gambar 2.7 terlihat RHEL sekarang sudah memasuki versi 7 dan begitu pula dengan CentOS yang mengikuti *release history* dari RHEL ini.

8. Jaringan Komputer

Jaringan komputer dapat diartikan sebagai suatu himpunan interkoneksi antar komputer. Dapat diartikan dua komputer yang membentuk satu buah jaringan ialah dua komputer yang dapat berinteraksi satu sama lain dan dapat bertukar informasi (Supriyadi & Gartina, 2007).

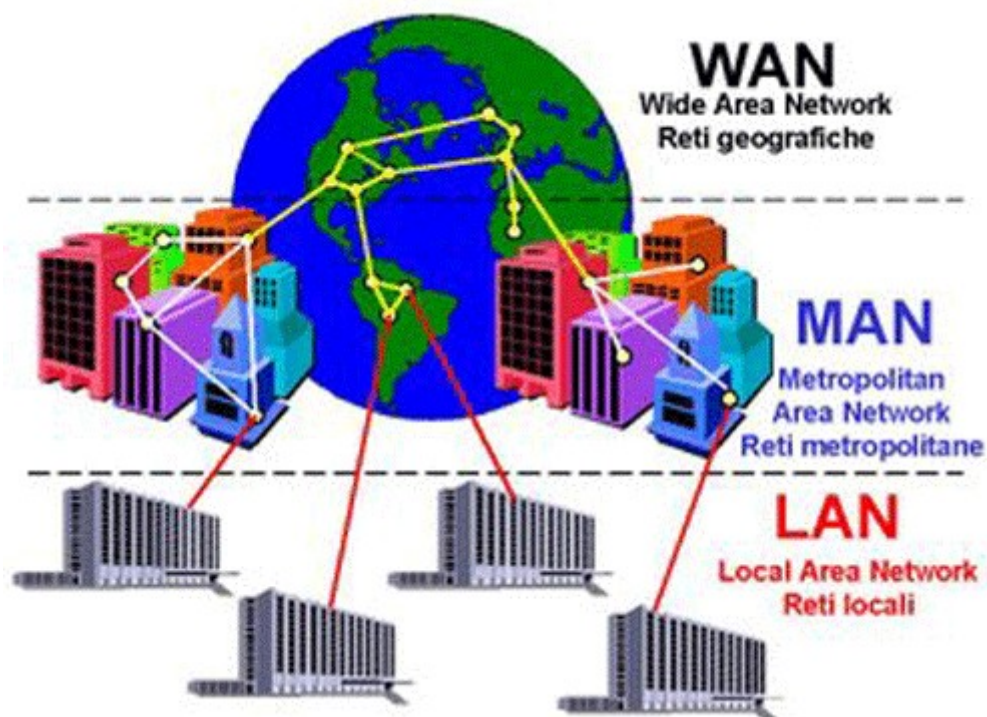
Dalam gambaran umum, sebuah jaringan mempunyai beberapa manfaat dibandingkan dengan sebuah komputer yang berdiri sendiri (Supriyadi & Gartina, 2007). yaitu :

1. Efisiensi dalam manajemen sumber daya. Misalnya ketika di dalam sebuah jaringan banyak pengguna yang bisa terkoneksi dalam satu printer tunggal dengan kualitas tinggi. Dibandingkan dengan memakai printer kualitas rendah di masing – masing meja.
2. Jaringan dapat membantu dalam hal mempertahankan informasi agar tetap handal dan tidak tertinggal (*up to date*).
3. Jaringan juga dapat membantu dalam hal *sharing* data atau berbagi data antar pengguna.
4. Jaringan memungkinkan kelompok kerja berkomunikasi secara efisien. Sebagai contoh dalam penggunaan surat elektronik yang merupakan substansi sebagian besar sistem jaringan.
5. Jaringan juga membantu menangani klien menjadi lebih efektif. Seperti akses jarak jauh ke data terpusat yang memungkinkan karyawan dapat melayani klien di lapangan dan klien dapat langsung berkomunikasi dengan karyawan.

Adapun komputer jaringan dibagi atas lima jenis, yaitu :

1. Local Area Network (LAN), : merupakan sebuah jaringan internal di dalam sebuah gedung atau kampus. LAN seringkali digunakan untuk menghubungkan komputer – komputer workstation dan pribadi. Dalam perusahaan misalnya, berbagi sumber daya secara bersama – sama.

2. Metropolitan Area Network (MAN) : merupakan sebuah jaringan yang sama seperti LAN akan tetapi memiliki cangkupan yang lebih luas. MAN ini dapat mencakup luas antar kantor – kantor yang berdekatan.
3. Wide Area Network (WAN) : merupakan jaringan yang cakupannya ke daerah geografis yang luas. Seringkali mencakup sebuah negara bahkan benua.
4. Internet : merupakan sebuah jaringan yang memungkinkan menghubungkan orang lain antar suatu jaringan dalam satu waktu dalam jaringan yang saling terhubung (terinterkoneksi).
5. Jaringan tanpa kabel : biasa dikenal dengan sebutan *wireless* jaringan ini merupakan sebuah solusi terhadap suatu jaringan yang tidak bisa dilakukan dengan jaringan yang menggunakan kabel.



Gambar 2.8: Macam - macam jaringan komputer

Berikut merupakan ilustrasi dari macam – macam jaringan komputer yang terdapat pada gambar 2.8.

9. Perangkat Keras Jaringan Komputer

Suatu komputer yang terhubung tidak akan dapat terhubung jika tanpa perantara atau bisa disebut dengan perangkat keras jaringan komputer. Perangkat keras jaringan komputer ini dibutuhkan untuk membangun suatu jaringan komputer (James, 2015).

Adapun macam – macam perangkat keras jaringan komputer adalah sebagai berikut :

1. *Network Interface Card* (NIC) dan *Ethernet Card*/Kartu Jaringan

Kartu jaringan merupakan perangkat yang menyediakan media untuk menghubungkan antar komputer. Kebanyakan kartu ini bersifat internal yaitu kartu jaringan yang dipasang secara ekspansi kedalam komputer.

2. Hub dan Switch

Hub adalah sebuah perangkat yang bertugas menyatukan kabel – kabel jaringan dari setiap *workstation*, *server* atau perangkat lain. Sementara Switch adalah sebuah konsentrator yang samaa dengan Hub. Perbedaan dari kedua perangkat tersebut terdapat pada penyaluran data di jaringan.

3. Repeaters

Alat yang berfungsi sebagai penguat sinyal. Contoh yang paling sederhana dari fungsi alat ini ialah pada sebuah jaringan LAN yang menggunakan sebuah topologi jaringan komputer dengan menggunakan kabel *unshielded twisted pair*. Karena panjang kabelnya mempunyai batas, maka untuk menguatkan sinyal tersebut digunakanlah repeater agar sinyal tersebut makin kuat.

4. Bridges

Perangkat yang bertugas membagi satu buah jaringan kedalam dua buah jaringan. Perangkat ini digunakan untuk membuat suatu jaringan menjadi lebih efesien, dikarenakan pertumbuhan jaringan yang cepat sehingga dibutuhkan Bridges.

5. Routers

Routers adalah sebuah perangkat jaringan yang mengartikan sebuah informasi dari satu jaringan ke jaringan yang lain. Perangkat ini bisa diartikan hampir sama seperti

Bridges akan tetapi sedikit lebih pintar. Router akan mencari jalur terbaik dalam sebuah jaringan untuk mengirimkan sebuah pesan berdasarkan alamat tujuan dan alamat asal.

6. Komputer

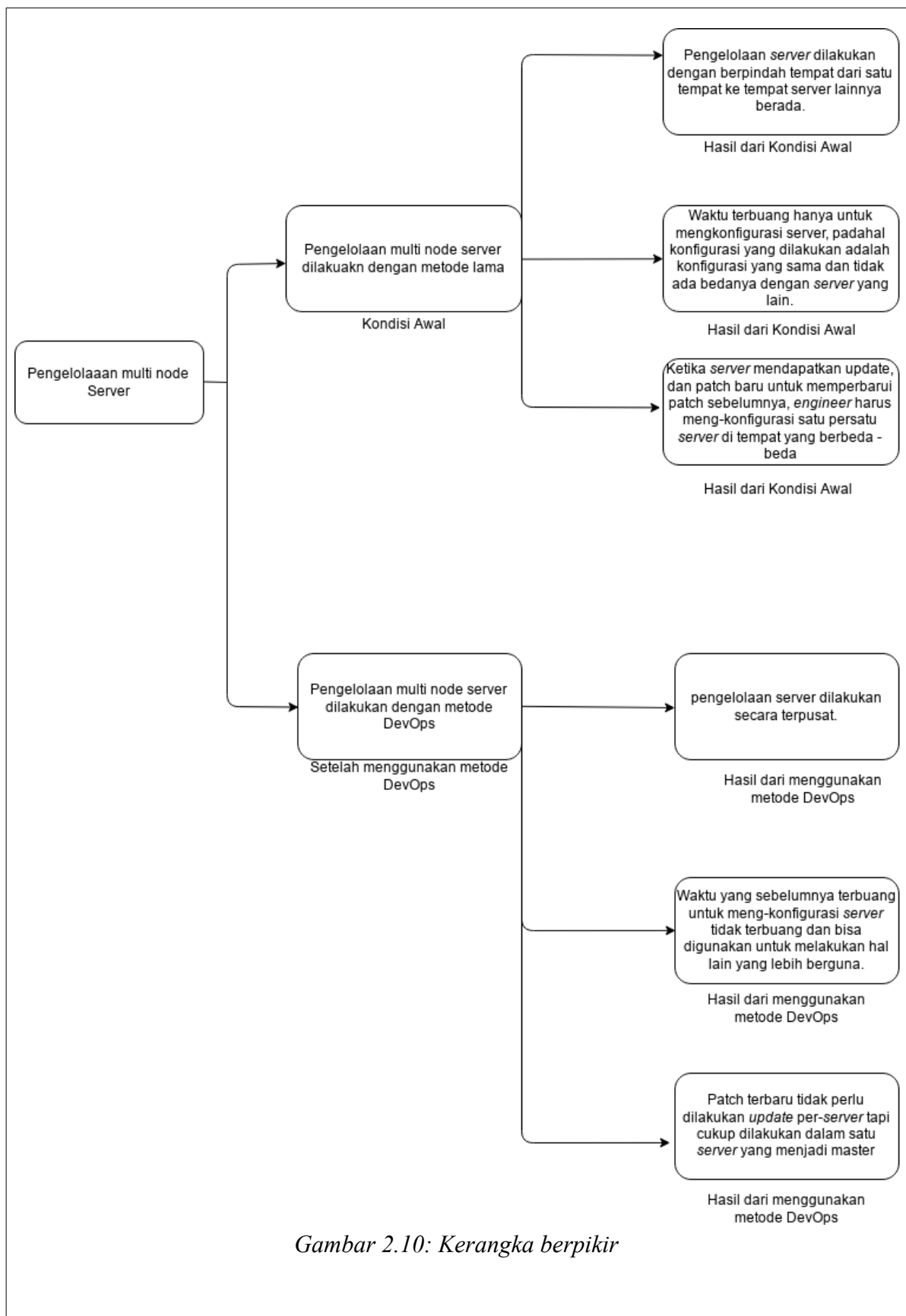
Komputer atau server adalah sebuah komputer yang berpusat menjadi pengelola dan pusat bagi komputer lainnya. Karena komputer ini menjadi pusat maka komputer ini harus mempunyai sesuatu yang lebih dibandingkan dengan komputer biasa.



Gambar 2.9: Perangkat Keras Jaringan Komputer

B. KERANGKA BERPIKIR

Dalam pengerjaannya, pengelolaan banyak *server* masih menggunakan cara yang tradisional yaitu dengan mengkonfigurasi satu persatu *server* secara bergantian yang mana itu akan memakan banyak waktu dari sisi pengerjaannya. Maka lahirlah metode DevOps yang akan memangkas waktu tersebut agar waktu yang dipakai oleh *network engineer*, serta dapat mengelola lebih dari satu *server* secara cepat tanpa adanya kesalahan yang berarti. Maka menggunakan metode DevOps adalah upaya untuk memecahkan masalah tersebut, berikut uraian dalam kerangka berpikir yang tertuang dalam gambar 2.10



Gambar 2.10: Kerangka berpikir

BAB 3

Metode Penelitian

A. Tempat dan Waktu Penelitian

1. Tempat Penelitian

Pelaksanaan penelitian ini akan dilakukan di Lab Komputer Pendidikan Teknik Informatika dan Komputer, Fakultas Keguruan dan Ilmu Pendidikan Universitas Sebelas Maret yang beralamatkan di Jl. Ahmad Yani No. 200 Pabelan, Kartasura, Kabupaten Sukoharjo.

2. Waktu Penelitian

Penelitian ini akan dilaksanakan pada bulan Mei minggu ke-1 dimana proses penelitian ini akan menggunakan Desain pengembangan NDLC yang terdiri dari *Analysis*, *Design*, *Simulation Prototyping*, *Implementation*, *Monitoring* dan *Management*.

Tahap pertama yakni *Analysis* yang akan dilakukan pada minggu ke-1 di bulan Mei samapai minggu ke-2 Mei.

Tahap kedua dan ketiga akan dilakukan berdekatan yakni pada pertengahan Juni dan Akhir Juli akan dilakukan *Design* dan *Simulation Prototyping*.

Tahap keempat dan kelima akan dilakukan sama seperti tahap kedua dan ketiga diatas yakni pada minggu ke-4 bulan Juli sampai dengan minggu ke-2 bulan Agustus

Table 3.1: Waktu Penelitian

Jenis Kegiatan	Bulan				
	Mey	Juni	Juli	Agu stus	Septe mber
1. <i>Analysis</i>					
• Analisa Kebutuhan					
2. <i>Design</i>					
• Design Topologi Jaringan					
3. <i>Simulation</i>					
• Simulasi multi node <i>server</i>					

4. <i>Implementation</i>					
• <i>Implementation multi node server</i>					
5. <i>Evaluation</i>					
• Evaluasi penghitungan MTBF dan MTTR serta Availability					
6. <i>Pelaporan</i>					
• Menyusun laporan skripsi					
• Ujian Skripsi					
• Revisi					

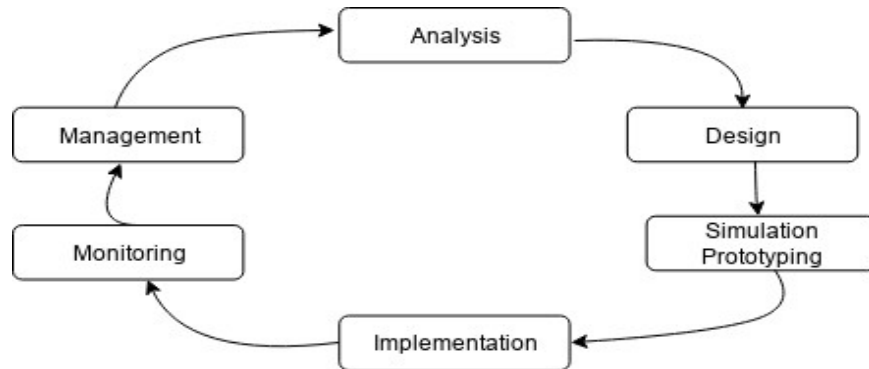
B. Metode Penelitian

Metode penelitian yang akan dilakukan dalam penelitian ini adalah R&D (*Research and Development*) sebagai bentuk metode penelitiannya. Pada hakikatnya metode penelitian R&D ini merupakan sebuah metode penelitian yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tertentu (Haryati, 2012).

C. Prosedur Penelitian

Network Development Lifecycle (NDLC) merupakan sebuah metode yang digunakan untuk merancang protokol atau membangun sebuah jaringan yang dapat memungkinkan terjadinya pemantauan statistik dan kinerja suatu jaringan. Meliputi analisis jaringan yang bersifat fisik dan logis seperti misalnya skema *routing*, pengalaman jaringan, prioritas lalu lintas data, serta keamanan dan manajemen (James, 2015).

Berikut merupakan tahapan dalam NDLC yang dijabarkan pada gambar 3.1.



Gambar 3.1: NDLC Siklus

Pada NDLC terdapat beberapa tahapan seperti yang terdapat pada gambar 3.1 diantaranya ialah *Analysis*, *Design*, *Simulation Prototyping*, *Implementation*, *Monitoring* dan *Management*.

Berikut adalah beberapa tahapan *Network Development Lifecycle* yang tertuang di penelitian (Kahn, 1982)

1. Tahap *Analysis*

Pada tahap *analysis* ini adalah tahap awal dari berbagai macam analisa seperti analisa kebutuhan yang diperlukan, analisa permasalahan yang muncul, sampai analisa topologi jaringan yang akan dibuat.

Pada penelitian kali ini adapun beberapa kebutuhan yang diperlukan dijelaskan pada tabel 3.2 sebagai berikut :

Table 3.2: Kebutuhan

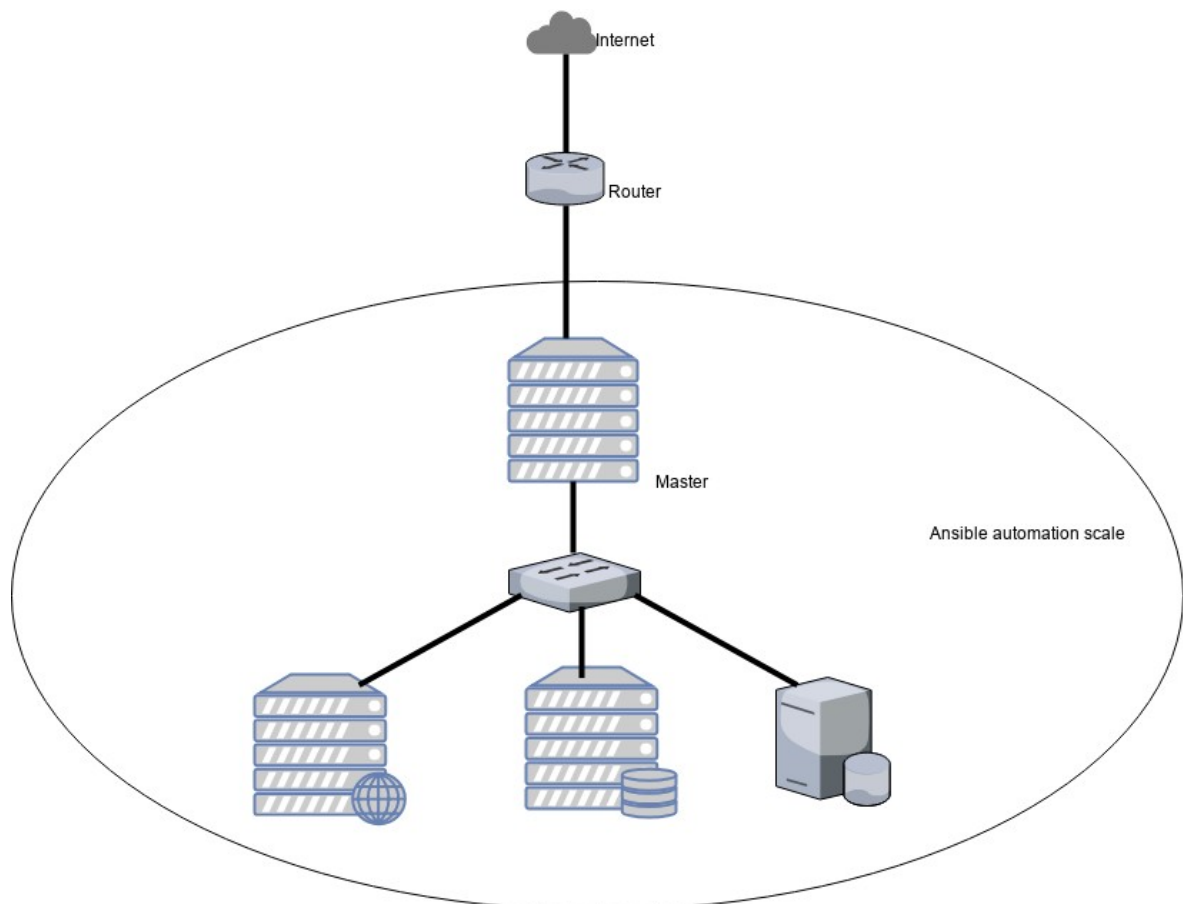
No	Nama Barang	Qty
1.	<i>Server</i>	3
2.	Router	2
3.	Switch	1
4.	Kabel RJ45	5

2. Tahap *Design*

Setelah analisa dibuat dan mendapatkan data – data yang diperlukan, maka pada tahap ini memulai perancangan *design* topologi jaringan yang akan dibangun. Adapun *design* yang dibuat antara lain berupa *design* struktur topologi, *design* akses data, *design* tata *layout* perkabelan dan sebagainya. Adapun *output* yang biasanya didapatkan dari tahapan ini ialah berupa :

- Gambar – gambar topologi
- Gambar – gambar *detail* estimasi kebutuhan yang ada

Adapun topologi yang akan digunakan pada penelitian kali ini akan dijelaskan pada gambar 3.2 berikut



Gambar 3.2: Topologi Jaringan

3. Tahap *Simulation Prototype*

Adapun sebelum melakukan pengujian secara langsung *prototype* akan membantu dalam melihat simulasi dari rancangan yang telah dibuat, adapun pada tahapan ini dibantu dengan beberapa *tools* yang biasanya digunakan oleh para *network engineer* antara lain : *packet tracer*, GNS3 dan lain – lain. Dan bahkan tidak sedikit para *network engineer* yang hanya menggunakan Visio dan *tools* pembuat topologi jaringan lainnya.

4. Tahap *Implementation*

Setelah 3 tahap yang telah disebutkan selesai, mulailah ketahapan yang sedikit memakan waktu lebih lama yakni tahap implementasi. Dimana semua rencana yang direncanakan sebelumnya akan diterapkan pada tahap ini. Pada tahap ini juga yang menentukan berhasil atau gagalnya *project* yang akan dibangun.

Adapun beberapa masalah yang sering muncul pada tahapan ini ialah :

- a. Jadwal yang tidak tepat karena faktor – faktor penghambat
- b. Masalah dana atau anggaran dan perubahan kebijakan
- c. Peralatan pendukung dari *vendor*
- d. *team work* yang tidak solid

Maka dari itu dibutuhkannya manajemen project dan manajemen resiko untuk meminimalisir itu semua.

Pada tahap ini dilakukan implementasi sesungguhnya dari hasil simulasi, dimana *server* akan dibangun dengan teknologi otomasi dengan menggunakan Ansible.

D. Teknik Pengujian

a. *Reliability*

Reliability adalah suatu pengujian kelayakan suatu sistem atau perangkat dalam mengerjakan tugas pada kondisi tertentu dan dalam selang waktu tertentu pula. Untuk mengetahui *Reliability* suatu perangkat dapat menggunakan perhitungan *Mean Time Between Failure* (MTBF). MTBF sendiri diperoleh dari tingkat kegagalan sebuah sistem atau perangkat (Aji, Hatta, & Wihidiyat, 2018).

$$MTBF = \text{Total Operating Time} / \text{Number of Failure}$$

b. *Maintainability*

Maintainability adalah kemampuan suatu sistem atau perangkat dalam pemulihan dalam kegagalan sistem. Pemeliharaan ini dikenal sebagai *Mean Time To Repair* (MTTR). MTTR adalah waktu perbaikan suatu sistem atau perangkat untuk kembali kedalam kondisi normal (Aji et al., 2018).

$$MTTR = \text{Total Down Time} / \text{Number of Repairs}$$

c. *Availability*

Availability adalah tingkat suatu sistem atau komponen beroperasi dan dapat diakses untuk digunakan, dan ini dinyatakan probabilitas. Ketersediaan ini diketahui dari hasil perbandingan antara MTBF dan MTTR

$$Availability = MTBF / (MTBF + MTTR)$$

DAFTAR PUSTAKA

- Aji, A. F., Hatta, P., & Wihidiyat, E. S. (2018). QoS and RMA performance analysis for wireless mesh network implementation. *2018 International Conference on Information and Communications Technology, ICOIACT 2018, 2018-Janua*, 254–258. <https://doi.org/10.1109/ICOIACT.2018.8350716>
- Budi, B., & Haryadi, S. (2018). *Simulasi Mobility pada Software Defined Simulasi Mobility pada Software Defined Networking*. 2018(November 2016), 26–27.
- Geerling, J. (2015). *Ansible for DevOps Server and configuration management for humans*.
- Hardion, V., Spruce, D., Lindberg, M., Milan Otero, A., Lidon-Simon, J., Jan Jamroz, J., & Persson, A. (2013). Configuration Management of the Control System. *International Conference on Accelerator & Large Experimental Physics Control Systems*, 1114–1117. Retrieved from https://www.researchgate.net/publication/320126636_CONFIGURATION_MANAGEMENT_OF_THE_CONTROL_SYSTEM
- Haryati, S. (2012). Research and Development (R&D) Sebagai Salah Satu Model Penelitian dalam Bidang Pendidikan. *Fkip Utm*, 37(1), 11–26. Retrieved from https://www.academia.edu/15666277/RESEARCH_AND_DEVELOPMENT_R_and_D_SEBAGAI_SALAH_SATU_MODEL_PENELITIAN_DALAM_BIDANG_PENDIDIKAN_Oleh
- Informatika, T., Sains, I., & Akprind, T. (2018). *OPTIMALISASI KINERJA SERVER MENGGUNAKAN MANAJEMEN DNS OPTIMIZING SERVER PERFORMANCE USING DNS MANAGEMENT*. 5(2), 153–163.
- J Edelman, SS Lowe, M. O. (2018). *Network Programmability and Automation: Skills for the Next-Generation Network Engineer*.
- James. (2015). Desain Topologi Jaringan Kabel Nirkabel PDII-LIPI dengan Cisco Three-Layered Hierarchical menggunakan NDLC. *Jurnal Elkomika*, 4(1). <https://doi.org/10.26760/elkomika.v4i1.47>
- Kahn, J. (1982). Network Development Life Cycle. *Jurnal Network Development Life Cycle*, (April), 13–33.
- Nelson, J. (2016). DevOps as a Lean Strategy. *Becoming a Lean Library*, 123–130. <https://doi.org/10.1016/B978-1-84334-779-8.00009-4>
- PetroChina, W. yiran, PetroChina, Z. tongyang, & PetroChina, G. yidong. (2018). Design and implementation of continuous integration scheme based on Jenkins and

- Ansible. *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 245–249. <https://doi.org/10.1109/ICAIBD.2018.8396203>
- Pribadi, P. T. (2013). Implementasi High-Availability Vpn Client Pada Jaringan Komputer Fakultas Hukum Universitas Udayana. *Program Studi Teknik Informatika, Jurusan Ilmu Komputer*, 6(1), 17–24.
- Raharja, R. A. (2001). *Pengenalan Linux*.
- Ray, E. (2015). *Pengembangan Aplikasi Monitoring Server Berbasis Mobile Web Dengan Sistem Notifikasi Universitas Islam Negeri Syarif Hidayatullah Jakarta*.
- Redhat. (2019). *AUTOMATE YOUR NETWORK WITH RED HAT*.
- Sabiq, A. (2016). *METODE FUZZY AHP DAN FUZZY TOPSIS UNTUK PEMILIHAN*. (May).
- Santoso, & Assegaf, S. (2017). Analisis Dan Rancang Bangun Sistem Layanan Proxy Server Pada SMK Unggul Sakti Jambi. *Jurnal Manajemen Sistem Informasi*, 2(1), 260–277.
- Singh, N. K., Thakur, S., Chaurasiya, H., & Nagdev, H. (2016). Automated provisioning of application in IAAS cloud using Ansible configuration management. *Proceedings on 2015 1st International Conference on Next Generation Computing Technologies, NGCT 2015*, (September), 81–85. <https://doi.org/10.1109/NGCT.2015.7375087>
- Supriyadi, A., & Gartina, D. (2007). Memilih Topologi Jaringan dan Hardware dalam Desain Sebuah Jaringan Komputer. *Informatika Pertanian*, 16(2), 1037–1053.
- Sutarti, Siswanto, & Subandi, A. (2018). Implementasi Dan Analisis QoS (Quality of Service) Pada VoIP (Voice Over Internet Protocol) Berbasis Linux. *Jurnal PROSISKO*, 5(2), 92–101.
- Suyanto, A. H. (2005). *MENGENAL E-LEARNING*.
- Tahar, I., & Enceng. (2006). Hubungan Kemandirian Belajar Dan Hasil Belajar Pada Pendidikan Jarak Jauh. *Jurnal Pendidikan Terbuka Dan Jarak Jauh*, 7(2), 91–201.
- Technology, I. C. (2015). *Understanding DevOps & Bridging the gap from Continuous Integration to Continuous Delivery*. (Intech), 78–82.
- Wahyono, T. (2007). *Building & Maintenance PC Server*. PT Gramedia.