



SDN Implementation - Test on Mikrotik -

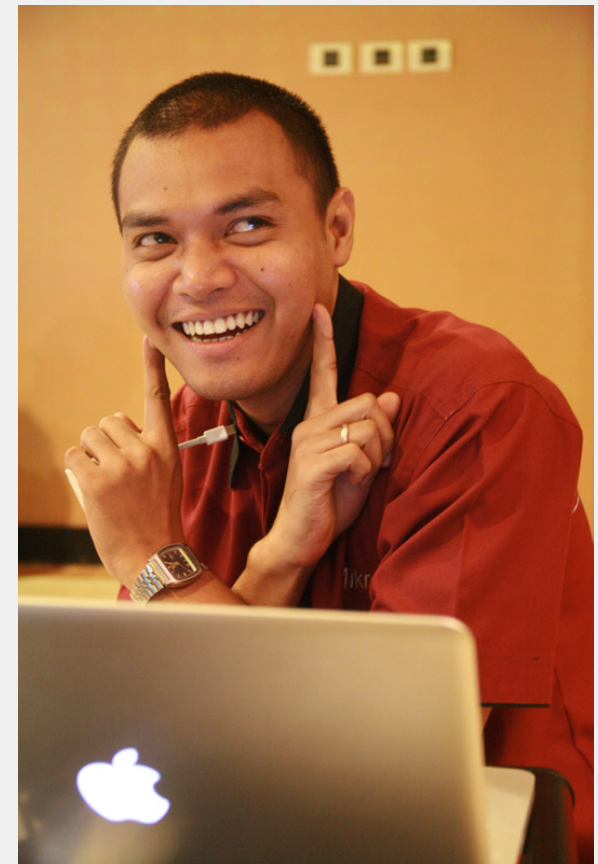
By: **Novan Chris**

CITRAWEB NUSA INFOMEDIA

www.mikrotik.co.id

Introduction

- **Novan Chris**
- Work for Citraweb / Citranet
 - Mikrotik Distributor & Training Partner, ISP
- Product Manager & Support Manager
- Mikrotik Certified Trainer
- MTCNA, MTCTCE, MTCRE, MTCWE, MTCUME, MTCINE



SDN ?

~~SOCIAL DEFINED NETWORK~~

SOFTWARE DEFINED NETWORKING

SDN ≠ OPENFLOW

~~SEKOLAH DASAR SINEGRI~~

SOFTWARE DEFINED NETWORKING

- New Network Architecture
 - Pemisahan Control Plane dan Data Plane.
- Open Programmability
 - Management network berbasis aplikasi.
- Open Vendor Compatibility
 - Tidak terikat pada salah satu brand dan type perangkat.
- Open Network Topology
 - Network topologi tidak menghambat pengembangan Jaringan.

SDN dikembangkan untuk management jaringan yang lebih baik dan mengoptimalkan penggunaannya.

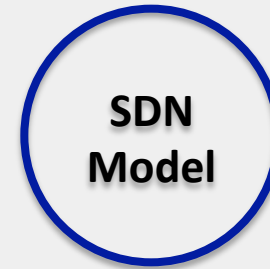
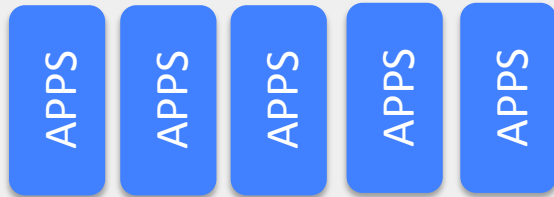
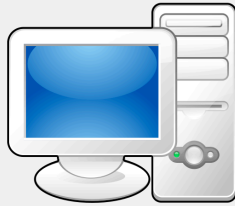
Pengembangan SDN

- mid 1990 - early 2000 – Programmable Network
 - Active Network Mulai memperkenalkan fungsi-fungsi programming yang bisa di implementasikan dalam Networking.
- 2001 - 2007 – Control & Data Planes Separation
 - IETF Mulai mengembangkan pemisahan Control Plane dan Data Plane pada Jaringan secara terbuka.
- 2007 -2010 – OpenFlow API and Network OS
 - ONF (Open Network Foundation) menghasilkan OpenFlow dan Network Operating System yang bisa mengimplementasi SDN secara praktis.
- Masih terus dikembangkan dan disempurnakan ...

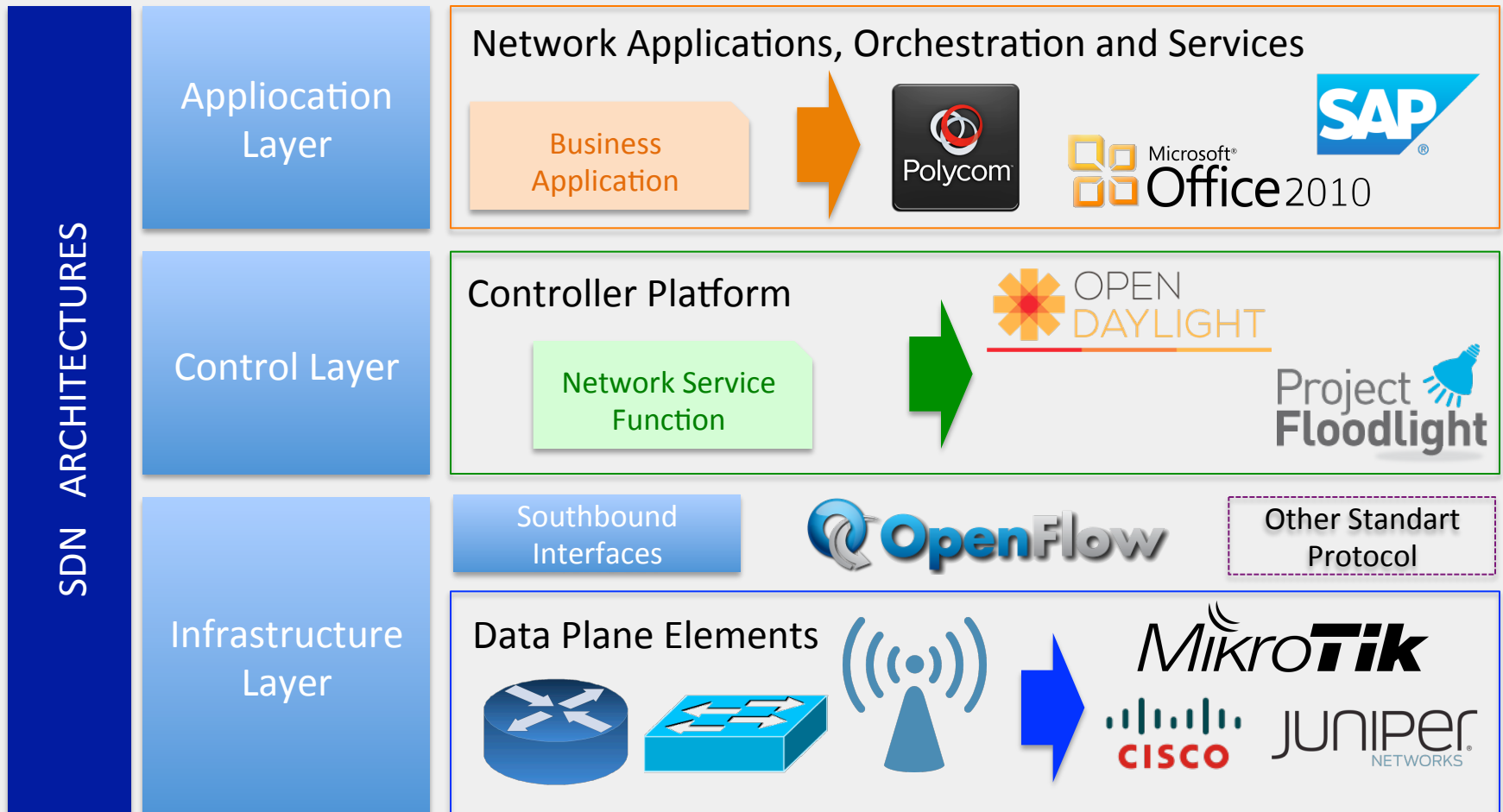
Fitur-fitur SDN

- Device Provisioning System
- Service Provisioning System
- Routing & Forwarding Controllers
- Centralized Control Plane (OpenFlow)
- Network Virtualization

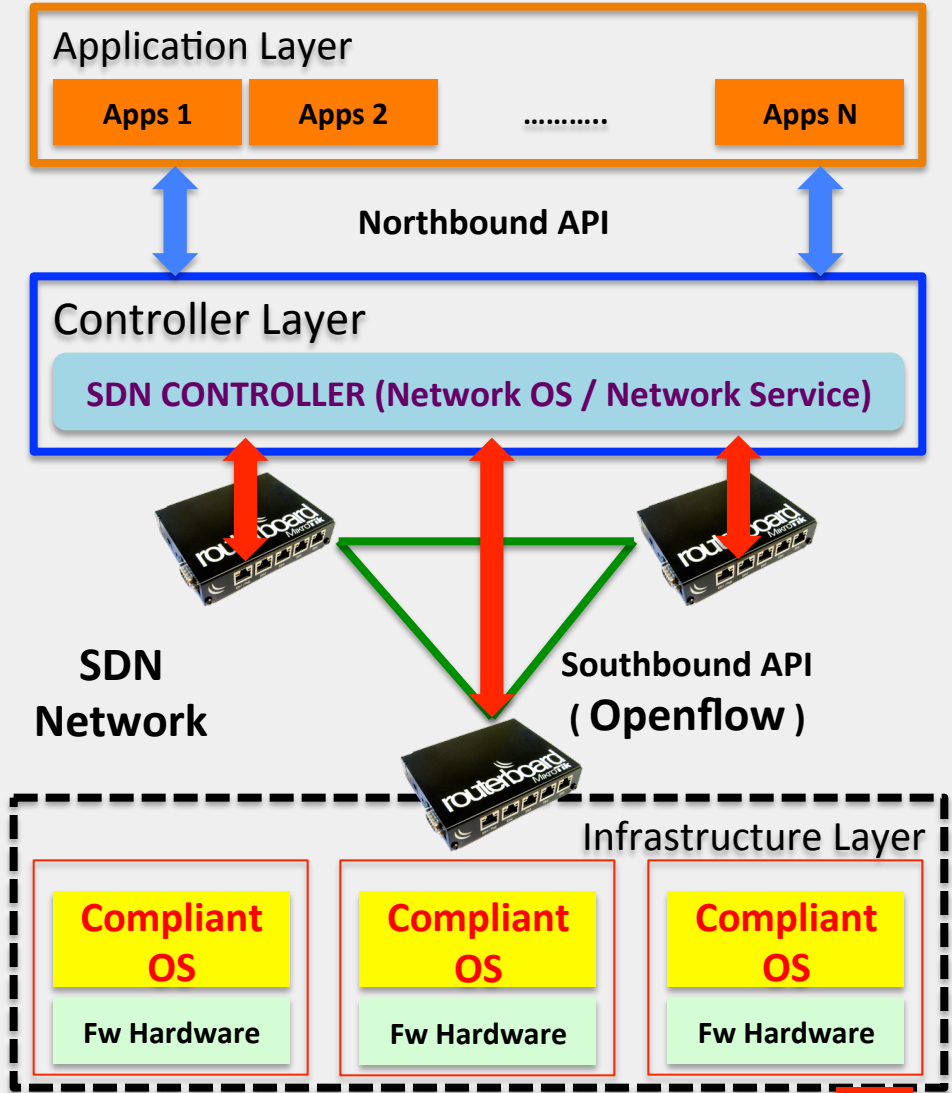
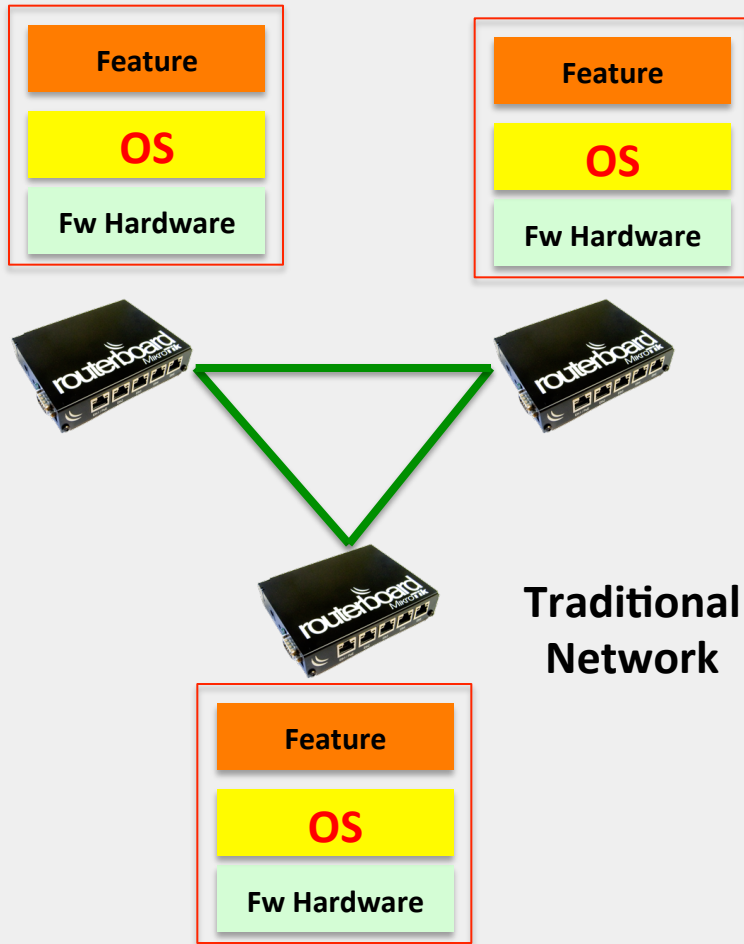
Arsitektur SDN



Arsitektur SDN

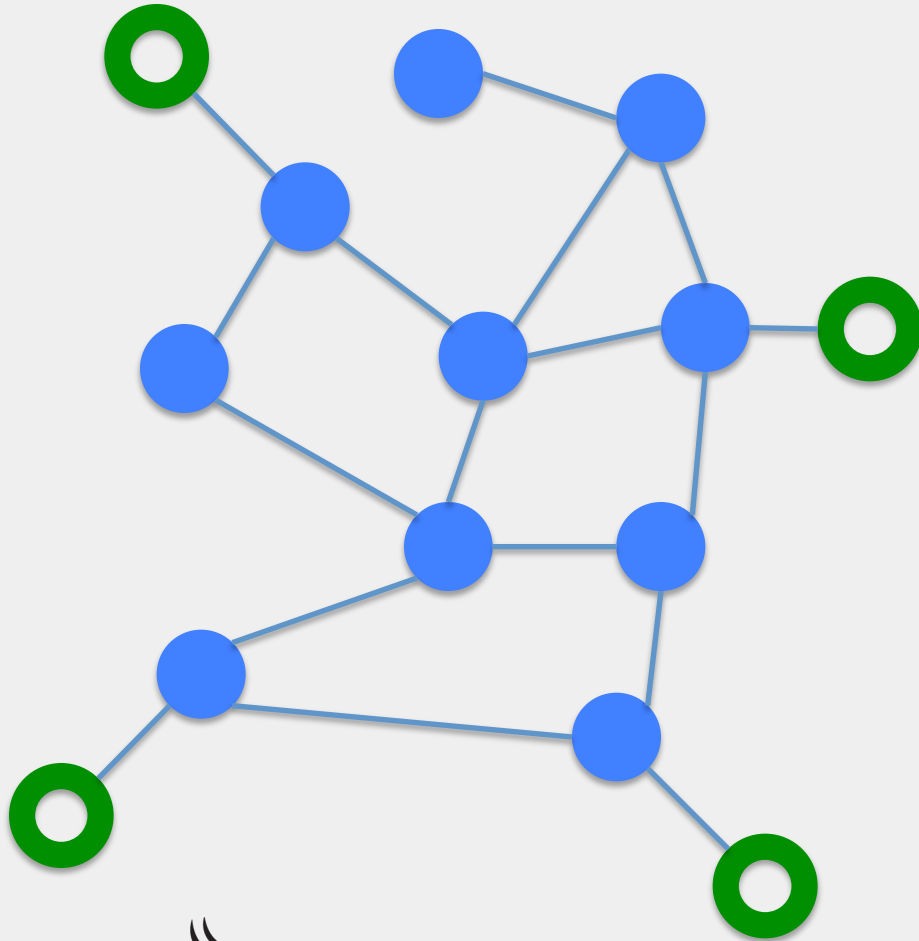


Arsitektur SDN

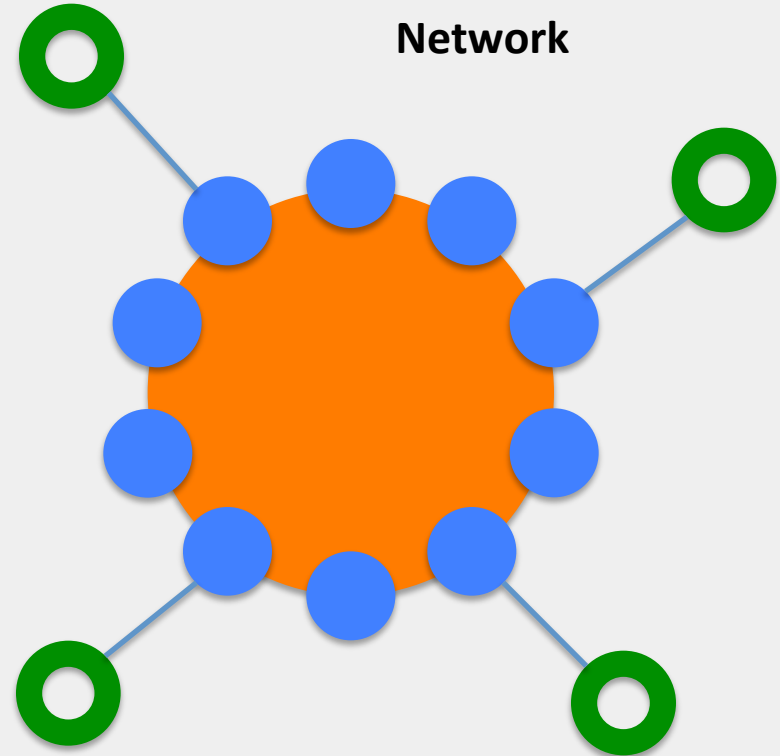


Arsitektur SDN

Traditional Network



SDN Network



Penggunaan SDN

- Campus Network
- Datacenter Network
- Service Provider Network
- Cloud Provider

- RT/RW Network ???
- Home Network ???????



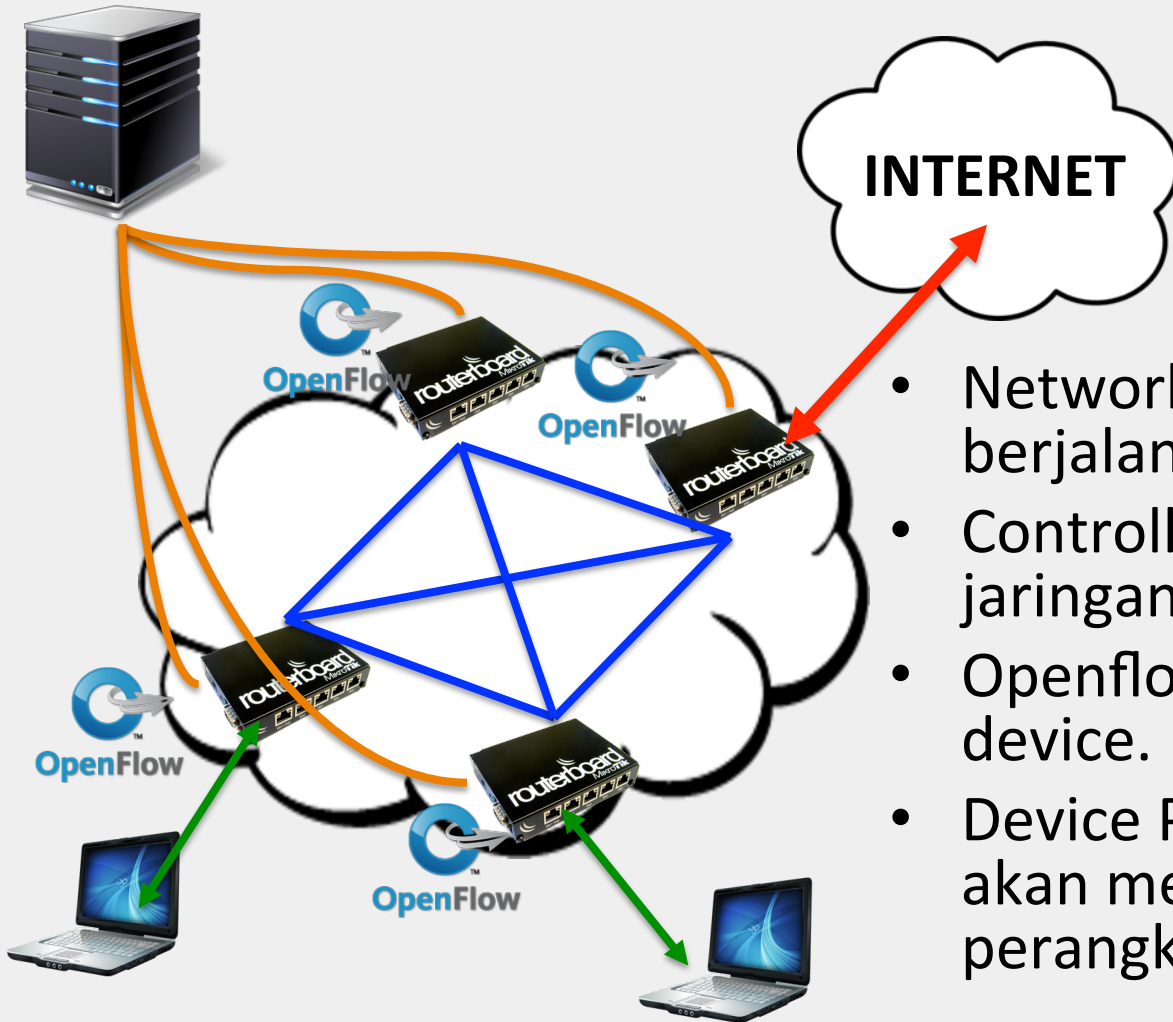
Mengapa SDN Dikembangkan

- Jaringan tidak lagi bersifat tertutup / proprietary dan juga semakin mudah implementasinya.
- Tidak disadari bahwa Jaringan yang dimiliki semakin besar dan dinamis (selalu berubah secara topologi) semakin rumit untuk di maintain.
- Perlu adanya platform baru yang memungkinkan para Pemilik Jaringan lebih mudah untuk mengatur infrastructure Jaringanannya. Supaya lebih optimal dan lebih fleksibel.

Kebutuhan Implementasi SDN

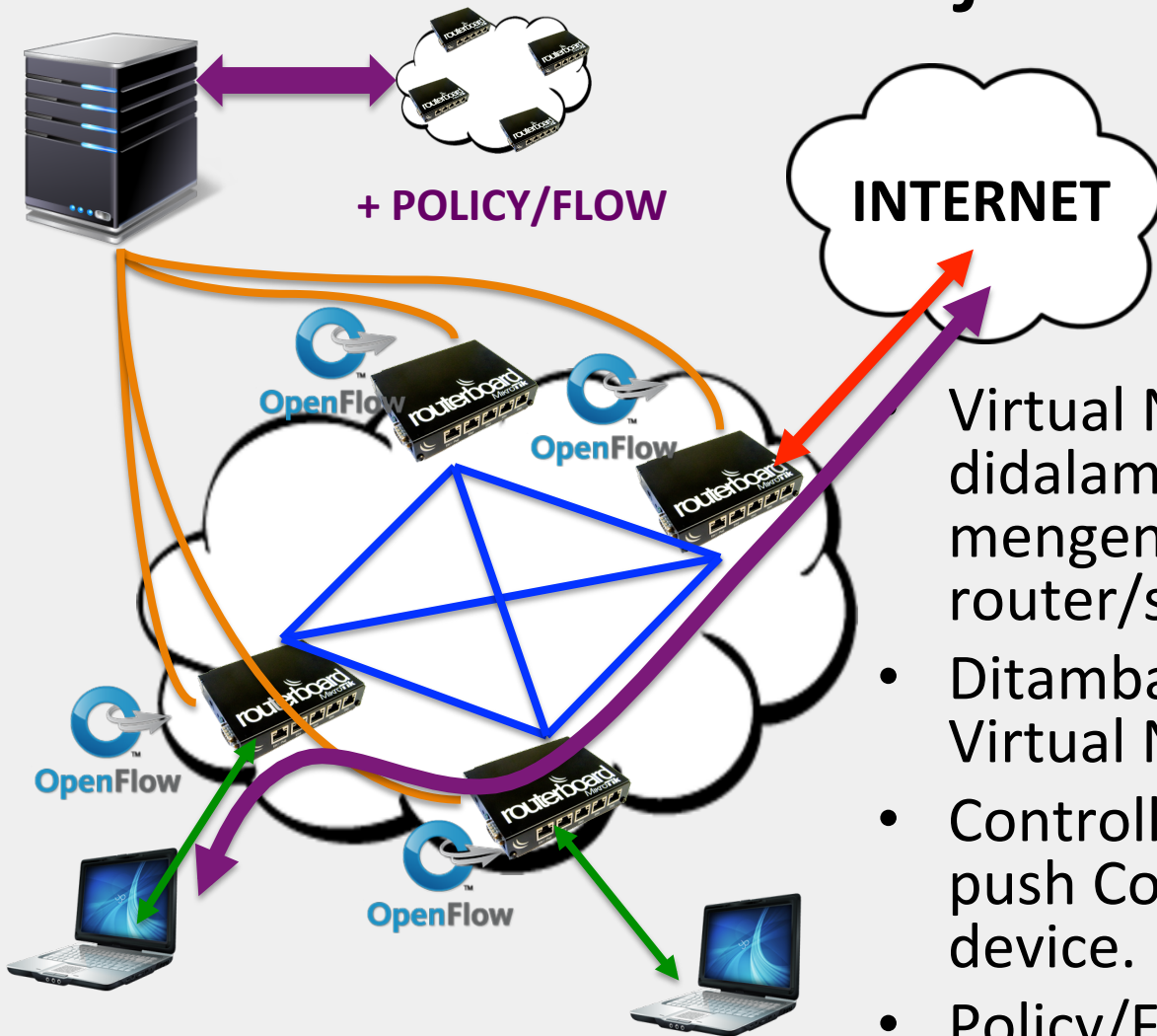
- Dari Arsitektur baru yang ditawarkan oleh SDN maka dibutuhkan sebuah perangkat baru dalam jaringan, yaitu **SDN Controller**.
- Network Device yang sudah ada dan sudah berjalan bertahun tahun perlu diganti dengan perangkat baru yang sudah support dengan standart SDN.
 - Tidak perlu diganti jika ada update software/ firmware dari perangkat tersebut.

Cara Kerja SDN



- Network yang sudah ada sudah berjalan secara normal.
- Controller ditambahkan dalam jaringan.
- Openflow diaktifkan di tiap device.
- Device Provisioning System akan mengenali semua perangkat.

Cara Kerja SDN



Virtual Network akan dibuat didalam controller, sekaligus mengenali interkoneksi antar router/switch.

- Ditambahkan policy / flow pada Virtual Network.
- Controller akan melakukan push Configuration ke setiap device.
- Policy/Flow akan diterapkan.

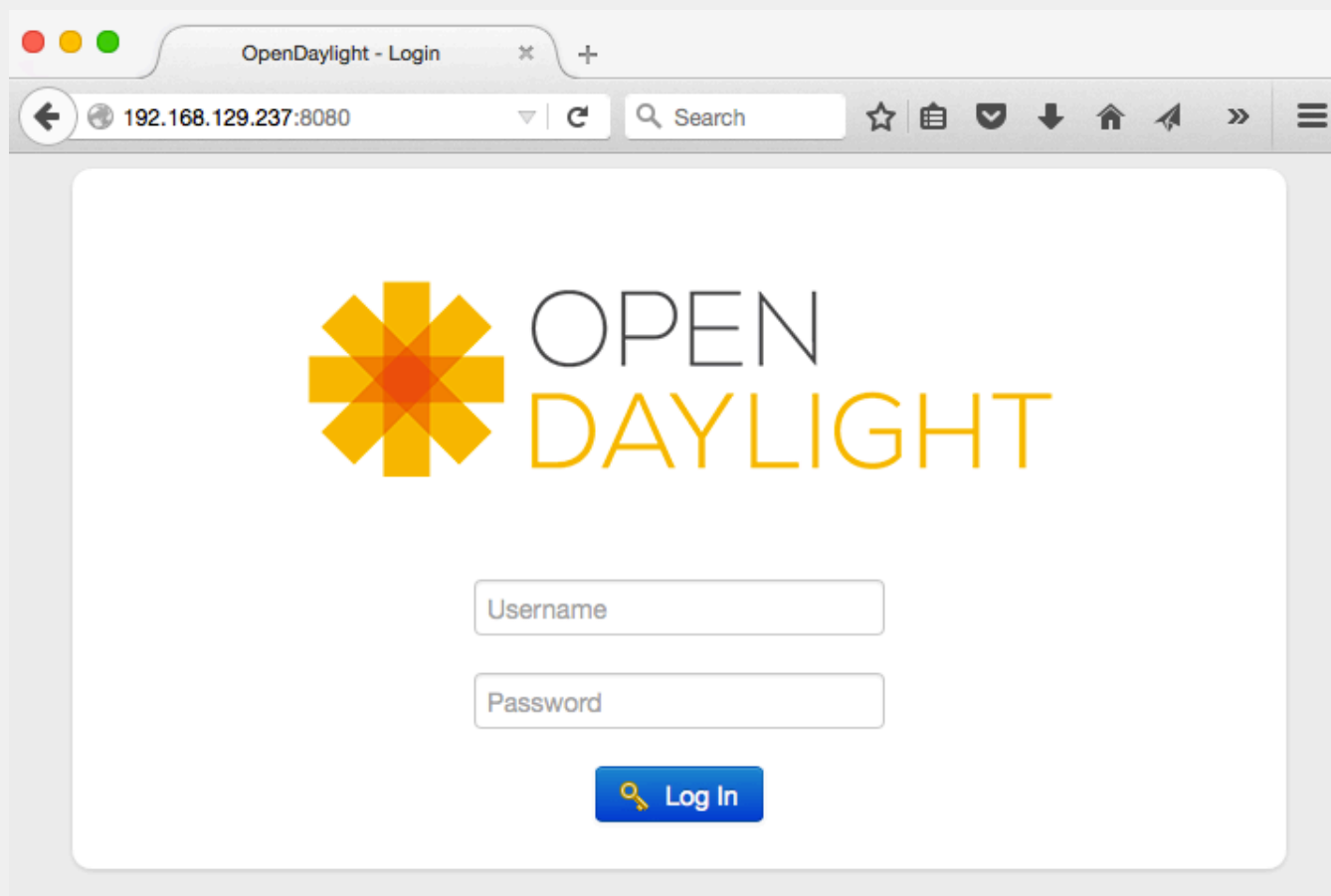
SDN Controller

- SDN Controller sebenarnya adalah sebuah Server
 - Hardware : Komputer berbasis Server dengan spesifikasi tinggi. (virtualisasi = irit...)
 - OS : berbasis Linux / Windows / OSX
 - Software Controller (JAVA Base): Opendaylight, Floodlight
- SDN Controller Masih terus dikembangkan dan disempurnakan.
- Beberapa Vendor Networking sudah mengembangkan controller mereka sendiri.

Mempersiapkan SDN Controller

- Siapkan server berbasis Linux
 - Pastikan sudah terinstall JVM dan JDK
 - apt-get install build-essential default-jdk ant python-dev
- Download software controller “Opendaylight”
 - Download controller :
 - <https://nexus.opendaylight.org/content/repositories/opendaylight.snapshot/org/opendaylight/controller/distribution.opendaylight/>
 - Unzip
 - Sudo ./opendaylight/run.sh

Akses ke Controller



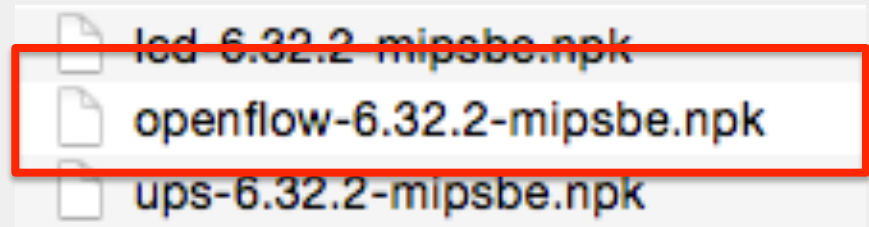
http://[ip-server]:8080 (**user:admin pass:admin**)

OpenFlow

- Protocol baru yang dikembangkan oleh ONF untuk mengadopsi konsep SDN.
- Dengan mengaktifkan Openflow pada Router/Switch memungkinkan Controller mengatur packet flow dari interface-interface yang ada didalamnya.
- Ketika openflow diaktifkan maka terjadilah pemisahan control plane dan data plane (forwarding) pada interface yang didaftarkan.

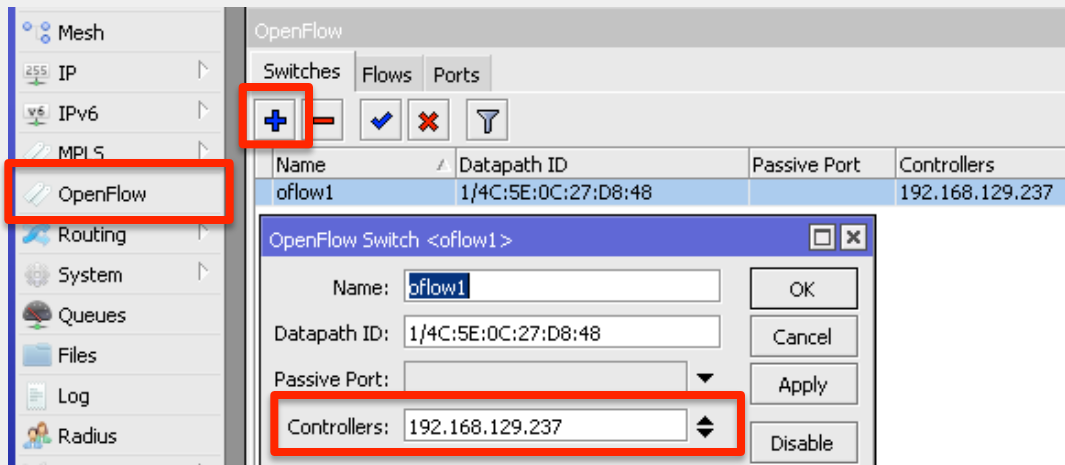
OpenFlow di Mikrotik

- Mikrotik sudah support dengan Openflow sejak di versi 6.rc8 (January 2013)
 - Openflow version 1.0.0
- Fungsi openflow bisa ditambahkan di Mikrotik dengan paket terpisah.
- Fungsi Openflow di Mikrotik masih dianggap EXPERIMENTAL (“Not for Production” - Wiki)



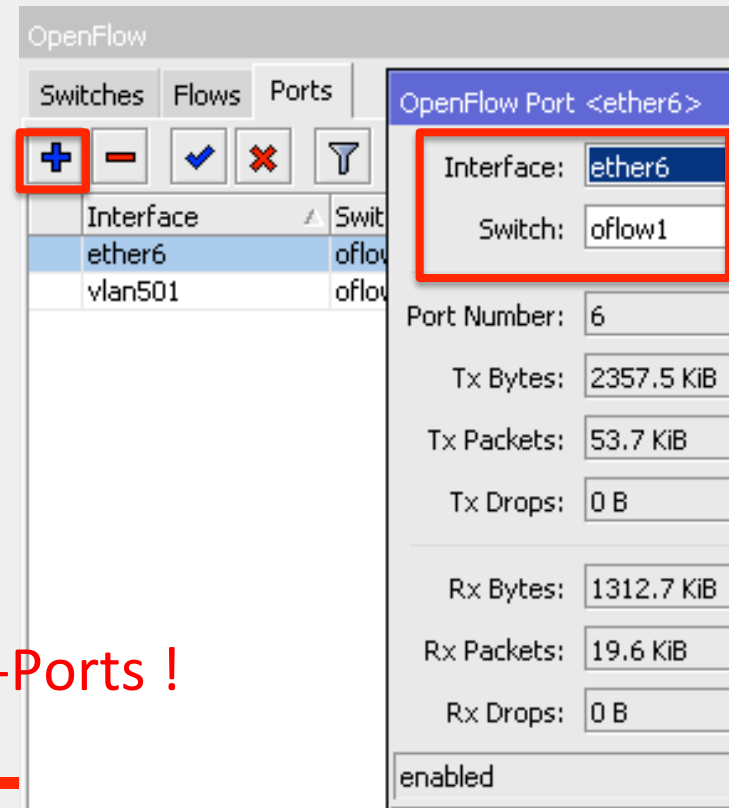
Implementasi Openflow

Setelah Openflow diaktifkan di mikrotik, router/switch bisa didaftarkan ke controller.

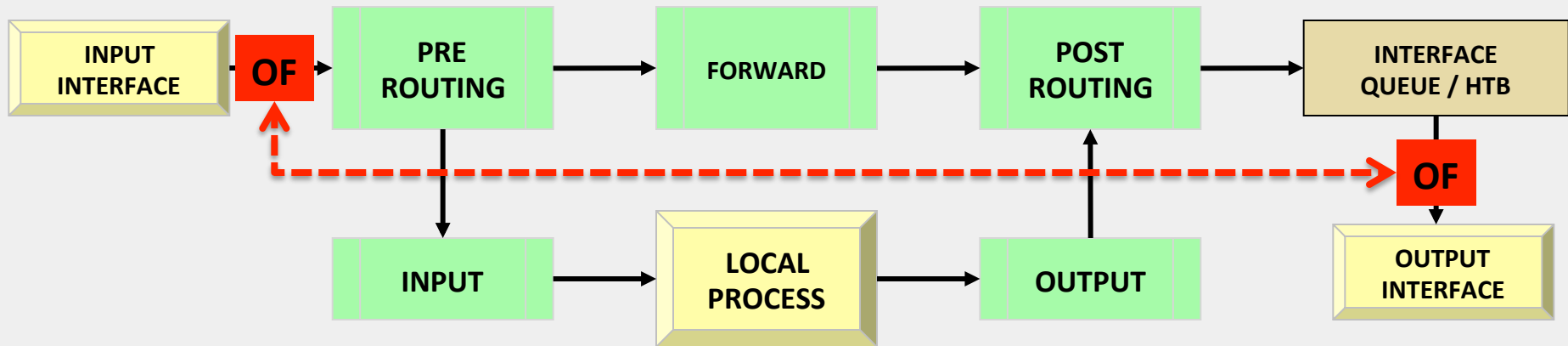


Daftarkan port-port yang akan dimasukkan kedalam openflow.

Mirip seperti Bridge-Ports !



Openflow - Packet Flow

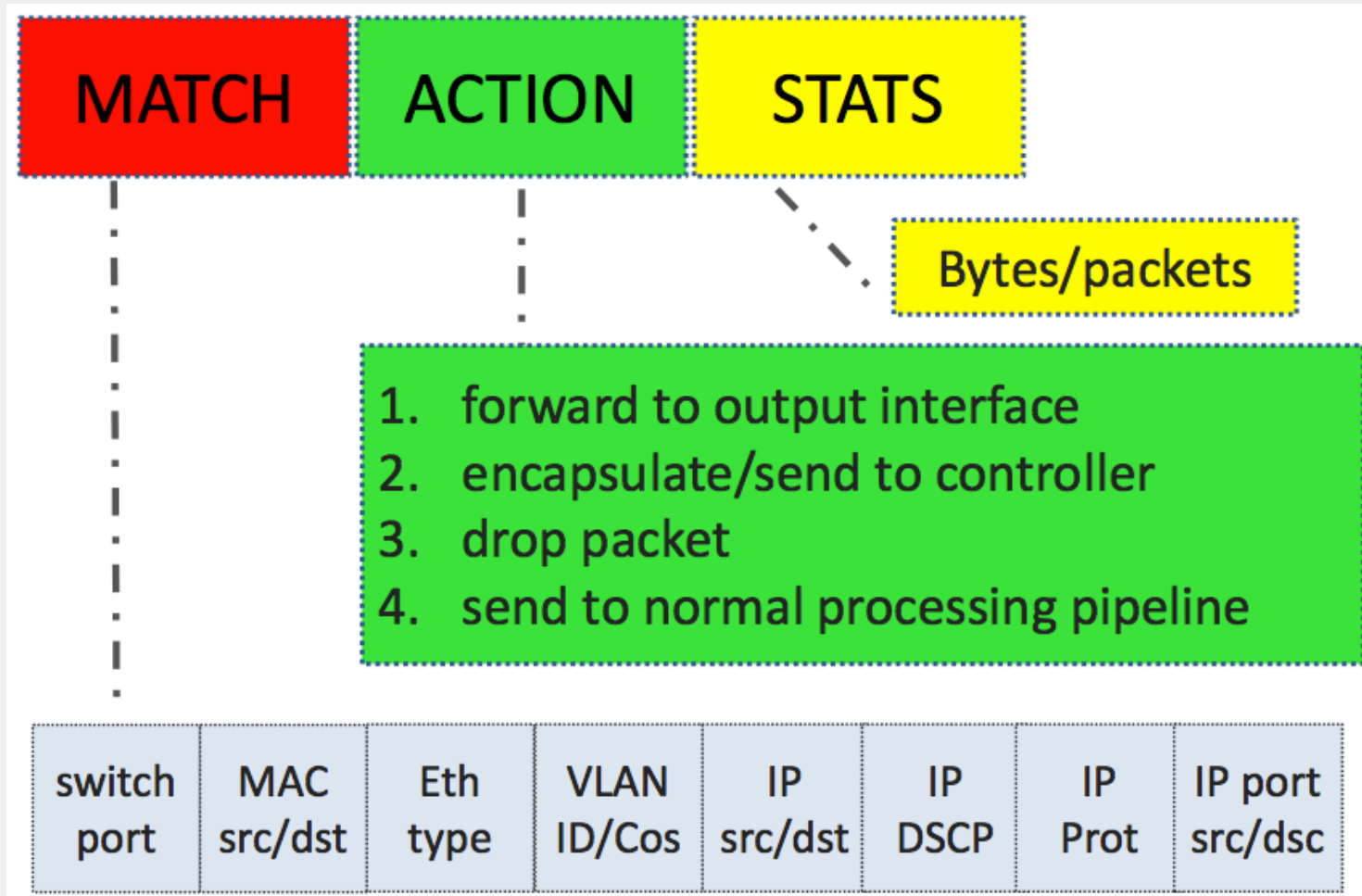


- Interface yang didaftarkan sebagai **Openflow Interface**, akan dikeluarkan dari logika packet-flow Mikrotik.
- Openflow bisa menggunakan Interface Ethernet dan juga Ethernet-like Interface (**seperti Mikrotik – Bridge Ports**) :
 - **Ethernet & Vlan**
 - **Eoip**
 - **VPLS**
 - **Bonding**
 - **Wireless Interface**

Openflow Flows

- Interface yang sudah dimasukkan ke dalam openflow akan menjalankan packet flow yang dibuat oleh Controller.
- Controller akan secara spesifik menentukan aliran data (flow) dari setiap interface yang ada di Openflow.
- Cara kerjanya hampir sama seperti di firewall mikrotik, flow di controller akan diklasifikasikan berdasarkan packet matcher dan selanjutnya menjalankan action tertentu.

Flow pada Openflow



Flow Matcher & Action

- Flow Matcher bisa mendeteksi traffic dari layer 1 - layer 4

switch port	MAC src/dst	Eth type	VLAN ID/Cos	IP src/dst	IP DSCP	IP Prot	IP port src/dsc
-------------	-------------	----------	-------------	------------	---------	---------	-----------------

- Action :
 - Port-to-port forwarding
 - Dikirim ke controller untuk diproses lebih lanjut
 - Drop Traffic !
 - Normal proses ke non-openflow network/device

Network Topology Discovery

Router Mikrotik akan dianggap sebagai Openflow Switch

The screenshot shows the OpenDaylight web interface. The browser address bar displays 192.168.129.237:8080. The page title is "OpenDaylight" and the user is logged in as "admin". The main content area shows a network topology diagram with five routers (Router1, Router2, Router3, Router4, Router5) and six hosts (172.21.100.1, 172.21.100.2, 172.21.100.3, 172.21.100.4, 172.21.100.5, 172.21.100.6). The routers are connected in a mesh topology. Router1 is connected to Router2 and Router3. Router2 is connected to Router1, Router3, and Router4. Router3 is connected to Router1, Router2, and Router5. Router4 is connected to Router2. Router5 is connected to Router3. Hosts are connected to their respective routers: 172.21.100.1 to Router1, 172.21.100.2 to Router2, 172.21.100.3 to Router3, 172.21.100.4 to Router4, 172.21.100.5 to Router5, and 172.21.100.6 to Router5.

Node Name	Node ID	Ports
Router1	OF 00:01:4c:5e:0c:27:d8:48	2
Router3	OF 00:01:00:0c:42:5d:15:dc	3
Router4	OF 00:01:00:0c:42:33:97:aa	2
Router5	OF 00:01:d4:ca:6d:29:4c:28	2
Router2	OF 00:01:00:0c:42:34:77:76	3

Controller akan melakukan discovery ke semua router/switch dan mendeteksi semua port yang ada, serta interkoneksi satu sama lain.

Menambahkan Flow di Controller

Ada dua metode menambahkan flow traffic di Controller Flow Table :

- **Reactive** – Ketika ada traffic yang masuk ke dalam jaringan, openflow agent akan memeriksa flow table. Jika tidak didapatkan flow yang cocok dengan traffic tersebut maka traffic tersebut akan dikirim ke controller dan menunggu controller untuk membuat flow yang sesuai.
- **Proactive** – flow akan dibuat otomatis di flow table menyesuaikan traffic yang lewat.

Flow Parameter

Add Flow Entry

Name

Flow Name

Node

Router3

Input Port

Please Select a Port

Please Select a Port

vlan503(1)

ether8(3)

eoip-tunnel1(6)

Layer 2

Ethernet Type

0x800

VLAN Identification Number

VLAN Identification Number

Range: 0 - 4095

VLAN Priority

VLAN Priority

Range: 0 - 7

Source MAC Address

3c:97:0e:75:c3:f7

Destination MAC Address

7c:d1:c3:e8:e6:99

Layer 3

Source IP Address

192.168.3.128

Destination IP Address

2001:2334::0/32

ToS Bits

ToS Bits

Range: 0 - 63

Actions

Controller

Please Select an Action

Controller

Drop

Flood

Flood All

Hardware Path

Output

Set VLAN ID

Software Path

Layer 4

Source Port

Source Port

Range: 0 - 65535

Destination Port

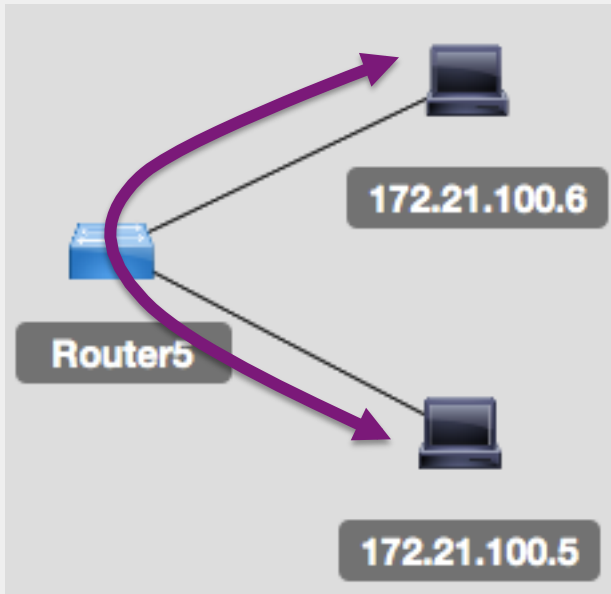
Destination Port

Range: 0 - 65535

Protocol

Protocol

Contoh Layer 1 flow



Flow 1	
Node	Router 5
Input Port	Port 1
Action	Output
Output Port	Port 2

Flow 2	
Node	Router 5
Input Port	Port 2
Action	Output
Output Port	Port 1

Router5 memiliki 2 port yang masing-masing terhubung dengan client. Flow harus dibuat supaya kedua client bisa saling berkomunikasi.

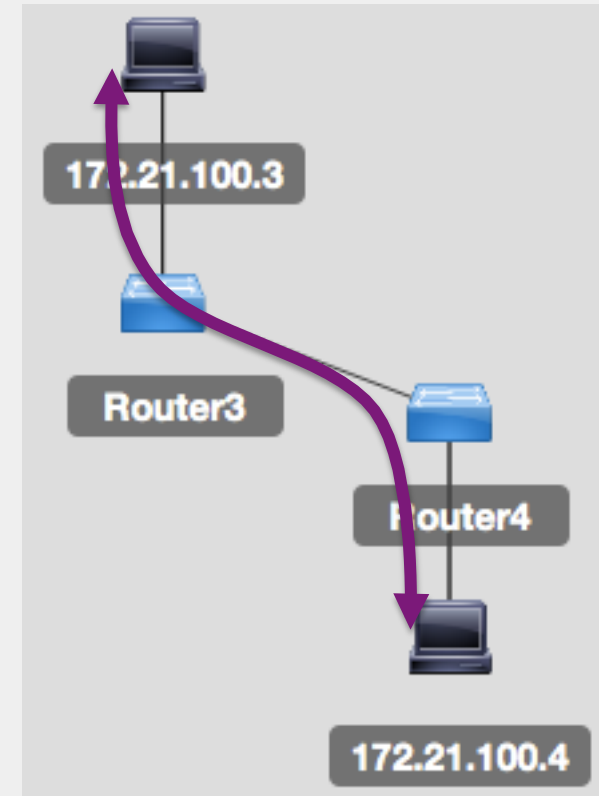
Contoh Layer 1 flow (Multi Switch)

Flow 1	
Node	Router 3
Input Port	Port 1
Action	Output
Output Port	Port 2

Flow 2	
Node	Router 3
Input Port	Port 2
Action	Output
Output Port	Port 1

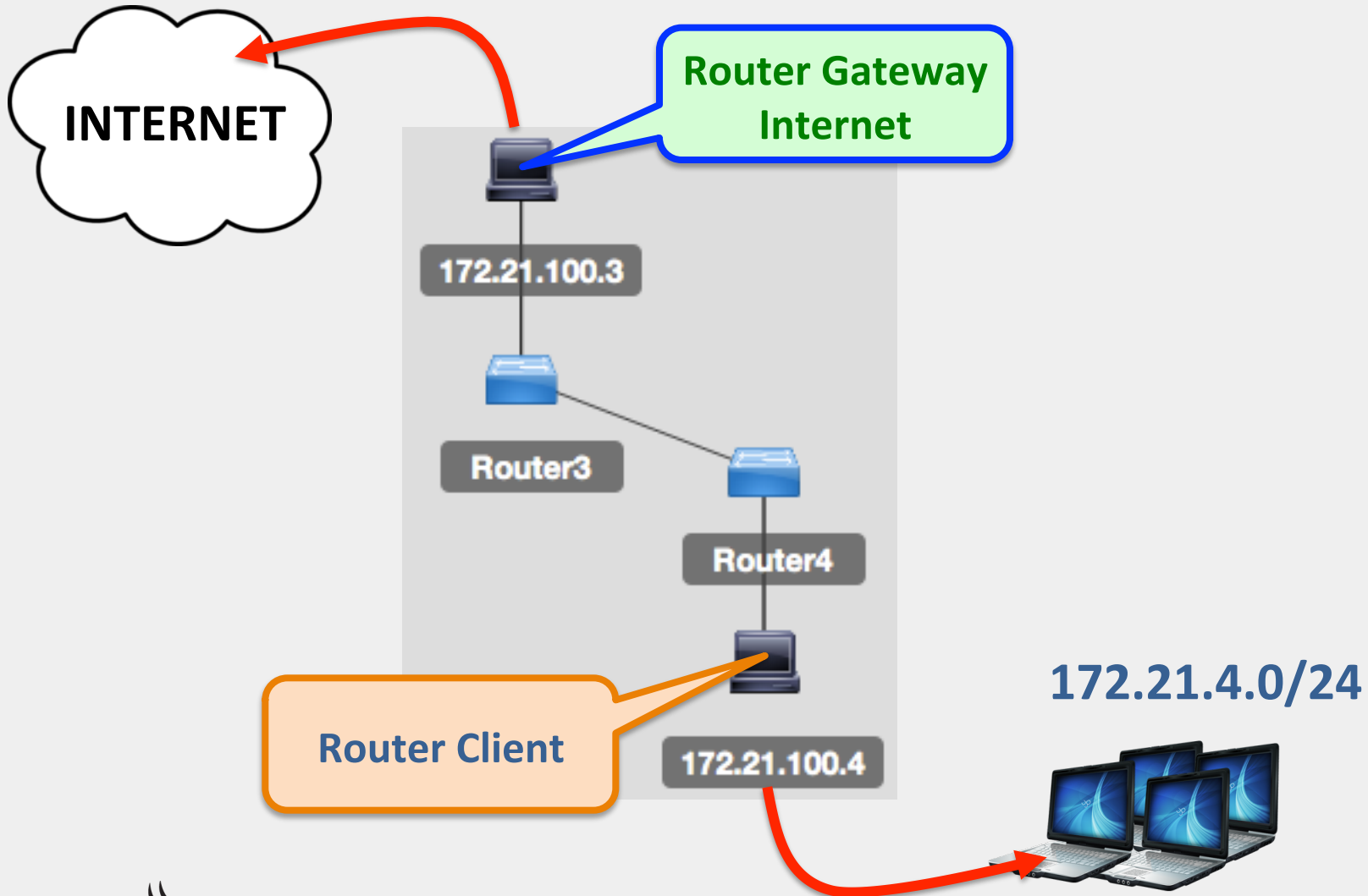
Flow 3	
Node	Router 4
Input Port	Port 1
Action	Output
Output Port	Port 2

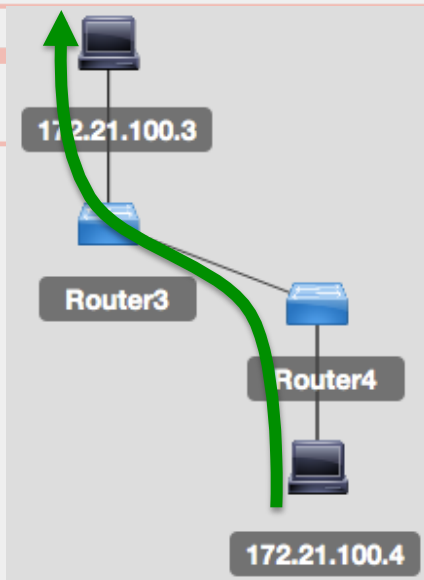
Flow 4	
Node	Router 4
Input Port	Port 2
Action	Output
Output Port	Port 1



2 host terhubung melewati 2 openflow switch

Contoh Layer 3 Flow

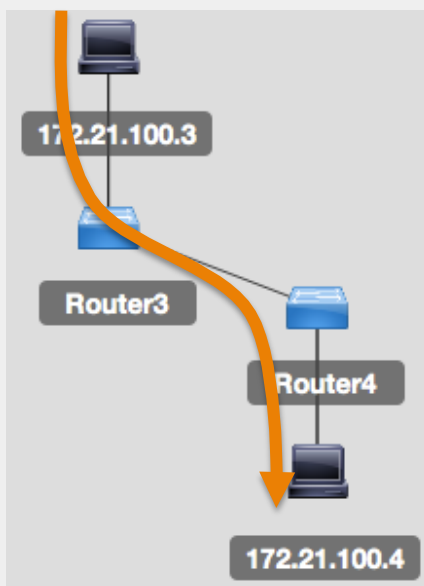




▲
 ▲
**U
P
L
O
A
D**
 ▼

Flow 1 : Client-to-Internet	
Node	Router 4
Input Port	From-Client
Eth-Type	0x800
Dst-Address	0.0.0.0/0
Action	Output
Output Port	To-Router3

Flow 2 : To Internet via Router3	
Node	Router 3
Input Port	From Router4
Eth-Type	0x800
Dst-Address	0.0.0.0/0
Action	Output
Output Port	To-Provider



**D
O
W
N
L
O
A
D**
 ▼

Flow 3 : Internet-to-Client	
Node	Router 3
Input Port	-
Eth-Type	0x800
Dst-Address	172.21.4.0/24
Action	Output
Output Port	To-Router4

Flow 4 : To Client via Router 4	
Node	Router 4
Input Port	-
Eth-Type	0x800
Dst-Address	172.21.4.0/24
Action	Output
Output Port	To-Client

BERSAMBUNG.... 😊

Conclusions

- SDN + Openflow = the Future....
- SDN tidak menggantikan fungsi networking yang sudah ada saat ini, SDN + Openflow mempermudah kita manage infrastructure jaringan yang sudah ada.
- Mikrotik baru mensupport Openflow Ver 1.0.0 sehingga belum bisa memanfaatkan fitur SDN lebih luas lagi.



Thank You !!

www.mikrotik.co.id

info@mikrotik.co.id

[@mikrotik_id](#)

Dijijinkan menggunakan sebagian atau seluruh materi pada modul ini, baik berupa ide, foto, tulisan, konfigurasi, diagram, selama untuk kepentingan pengajaran, dan memberikan kredit dan link ke www.mikrotik.co.id