

DIKTAT
PRAKTIKUM APLIKASI BERBASIS JARINGAN



OLEH
I PUTU HARIYADI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) BUMIGORA MATARAM

www.stmikbumigora.ac.id

DIKTAT
PRAKTIKUM APLIKASI BERBASIS JARINGAN

Oleh:
I Putu Hariyadi

Telah digunakan di STMIK Bumigora Mataram
Jurusan Teknik Informatika Strata 1 (S1)

Mataram, 7 Maret 2016
Kaprodi S1 Teknik Informatika



(Galih Hendro Martono, S.Kom, M.Eng)

NIK. 14.6.194

KATA PENGANTAR

Puji syukur penyusun panjatkan kepada Tuhan Yang Maha Esa atas berkat dan rahmatnya sehingga “**Diktat Praktikum Aplikasi Berbasis Jaringan**” ini dapat terselesaikan. Tidak lupa penyusun mengucapkan terimakasih kepada semua pihak di STMIK Bumigora Mataram, yang tidak dapat disebutkan satu per satu baik secara langsung maupun tidak langsung telah membantu dalam penyusunan diktat ini. Penyusun menyadari bahwa diktat ini masih jauh dari sempurna. Untuk itu kritik dan saran demi pengembangan diktat ini sangat diharapkan. Kritik dan saran dapat dikirimkan melalui email dengan alamat: *putu.hariyadi@stmikbumigora.ac.id*. Terimakasih.

Mataram, 7 Maret 2016

Penyusun

DAFTAR ISI

Halaman Judul	(i)
Halaman Pengesahan	(ii)
KATA PENGANTAR	(iii)
DAFTAR ISI	(iv)
PENDAHULUAN	(1)
Latar Belakang	(1)
Tujuan	(1)
Manfaat	(1)
BAB 1 MIKROTIK ROUTEROS	(2)
BAB 2 ADMINISTRASI & KONFIGURASI FITUR-FITUR MIKROTIK ROUTEROS SEBAGAI GATEWAY	(20)
BAB 3 IP HOTSPOT MIKROTIK ROUTEROS	(35)
BAB 4 MIKROTIK PHP API	(39)
BAB 5 MIKROTIK PHP API PADA SYSTEM USER	(59)
BAB 6 MIKROTIK PHP API PADA IP HOTSPOT	(76)
BAB 7 MIKROTIK PHP API PADA FIREWALL FILTER	(92)
BAB 8 KONEKSI PHP KE DATABASE MYSQL MENGGUNAKAN MYSQLI EXTENSION	(98)
DAFTAR REFERENSI	(107)

PENDAHULUAN

Latar Belakang

Penyusunan diktat ini diawali dengan adanya penyesuaian tujuan pembelajaran atau kemampuan yang diharapkan dapat dimiliki oleh mahasiswa setelah menyelesaikan perkuliahan Praktikum Aplikasi Berbasis Jaringan yaitu memiliki kemampuan dalam membuat aplikasi berbasis jaringan untuk manajemen Mikrotik RouterOS dengan antarmuka web. Tujuan pembelajaran tersebut berdampak pada perubahan materi belajar yang harus disampaikan pada praktikum. Untuk itu diperlukan adanya bahan ajar yang dapat mendukung proses pembelajaran sehingga tujuan pembelajaran dapat tercapai.

Tujuan

Diktat ini dibuat dengan tujuan menyediakan bahan ajar yang dapat digunakan oleh mahasiswa dalam mempelajari materi-materi terkait praktek membuat aplikasi berbasis jaringan menggunakan bahasa pemrograman PHP, dan database MySQL yang diintegrasikan dengan API PHP Class untuk berinteraksi dengan Mikrotik RouterOS..

Manfaat

Dengan adanya diktat ini diharapkan dapat mempermudah dan meningkatkan kemandirian mahasiswa dalam mempelajari materi terkait aplikasi berbasis jaringan sehingga kompetensi dari matakuliah ini tercapai.

BAB I

Mikrotik RouterOS

Tujuan:

Mahasiswa memahami mengenai instalasi & konfigurasi Mikrotik RouterOS.

Materi:

A. Pengenalan Mikrotik RouterOS

MikroTik RouterOS TM adalah sistem operasi router dan perangkat lunak yang dapat diinstalasi pada PC biasa atau MikroTik RouterBOARD hardware TM sehingga dapat berfungsi sebagai sebuah dedicated router.

Mikrotik RouterBoard Hardware



Akses ke Mikrotik

Untuk dapat mengakses mikrotik dapat dilakukan melalui beberapa cara yaitu:

- a) Console (lokal).
- b) Web.
- c) Telnet.

- d) SSH (dengan bantuan program aplikasi Putty).
- e) Winbox (program berbasis GUI untuk mengkonfigurasi Mikrotik).

Sebelum dapat melakukan konfigurasi, Anda harus melalui proses otentikasi login terlebih dahulu. Default user dari mikrotik adalah:

User : **admin**

Password : **kosong (blank)**

Diperlukan konfigurasi pengalamatan IP pada salah satu interface dari Mikrotik, sebelum dapat mengakses Mikrotik secara remote melalui antar muka web, telnet, SSH, atau winbox.

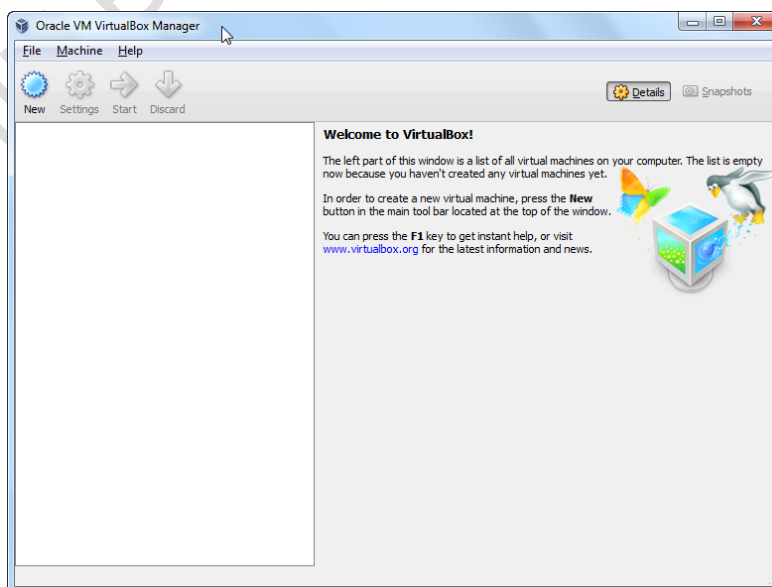
B. Pengenalan Virtualisasi menggunakan Oracle VM VirtualBox

VirtualBox merupakan aplikasi virtualisasi yang bersifat *cross-platform*. Melalui virtualisasi dapat dijalankan beberapa sistem operasi secara bersamaan (Sumber: VirtualBox Manual).

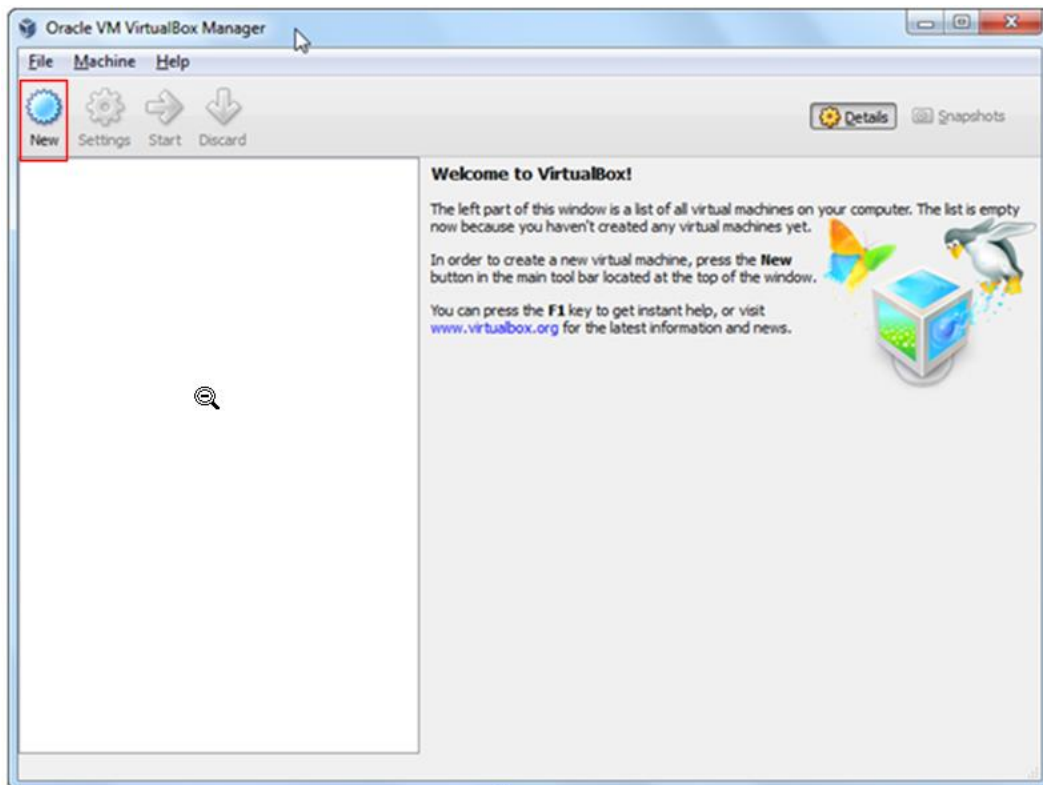
C. Instalasi & Konfigurasi VirtualBox

Adapun langkah-langkah instalasi & konfigurasi VirtualBox adalah sebagai berikut:

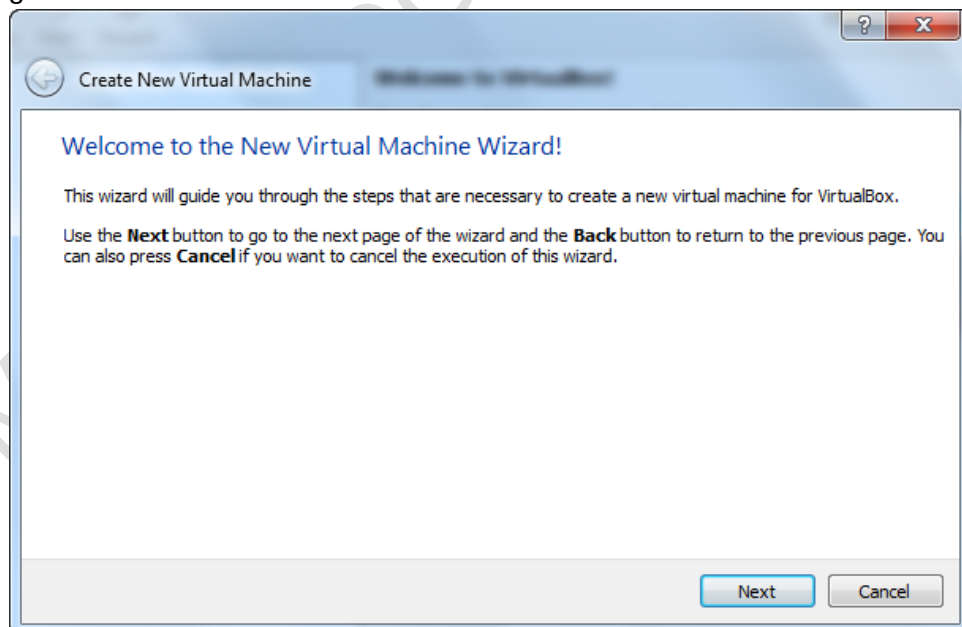
1. Jalankan aplikasi **VirtualBox** melalui menu **Start > All Programs > Oracle VM VirtualBox > Oracle VM VirtualBox**.
2. Tampil aplikasi *Oracle VM VirtualBox Manager*, seperti ditunjukkan pada gambar berikut:



3. Untuk membuat VM baru, klik tombol “**New**” pada Toolbar, seperti ditunjukkan pada gambar berikut:

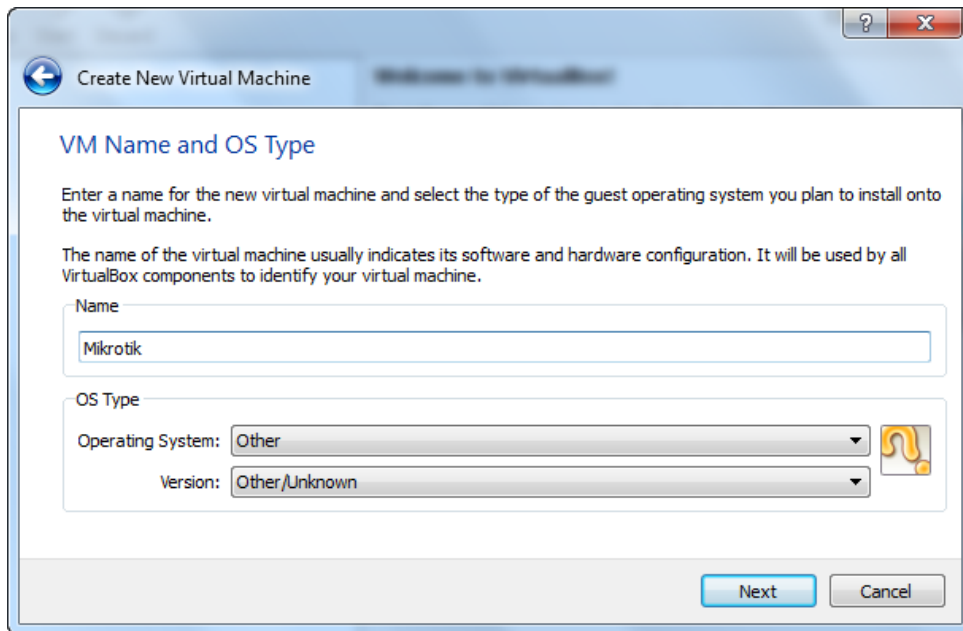


4. Tampil kotak dialog *Welcome to the New Virtual Machine Wizard*, seperti ditunjukkan pada gambar berikut:



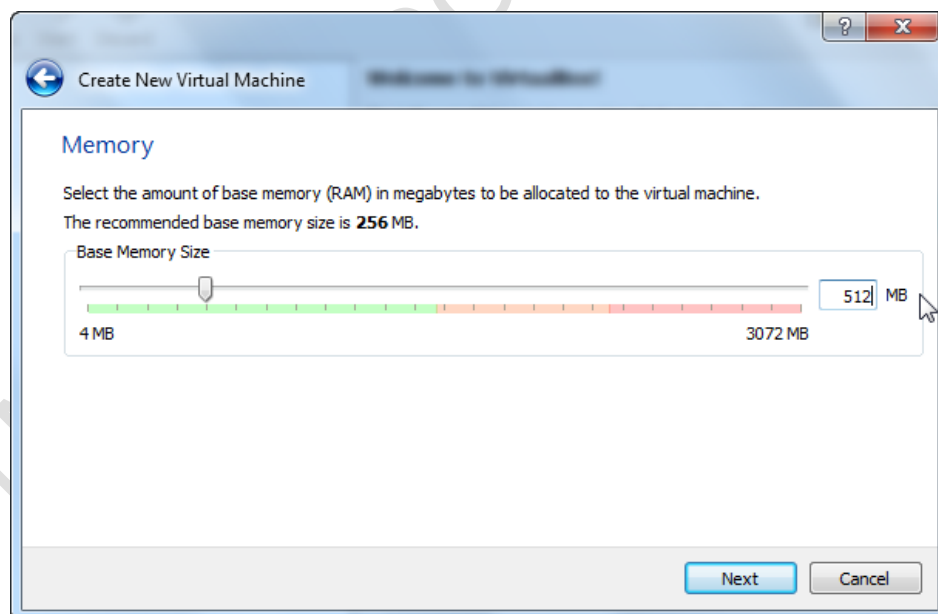
Klik tombol **Next** untuk memulai pembuatan *virtual machine* baru.

5. Tampil kotak dialog *VM Name and OS Type*, untuk menentukan nama pengenalan bagi VM yang dibuat, dan jenis sistem operasi yang akan diinstalasi pada VM, seperti ditunjukkan pada gambar berikut:



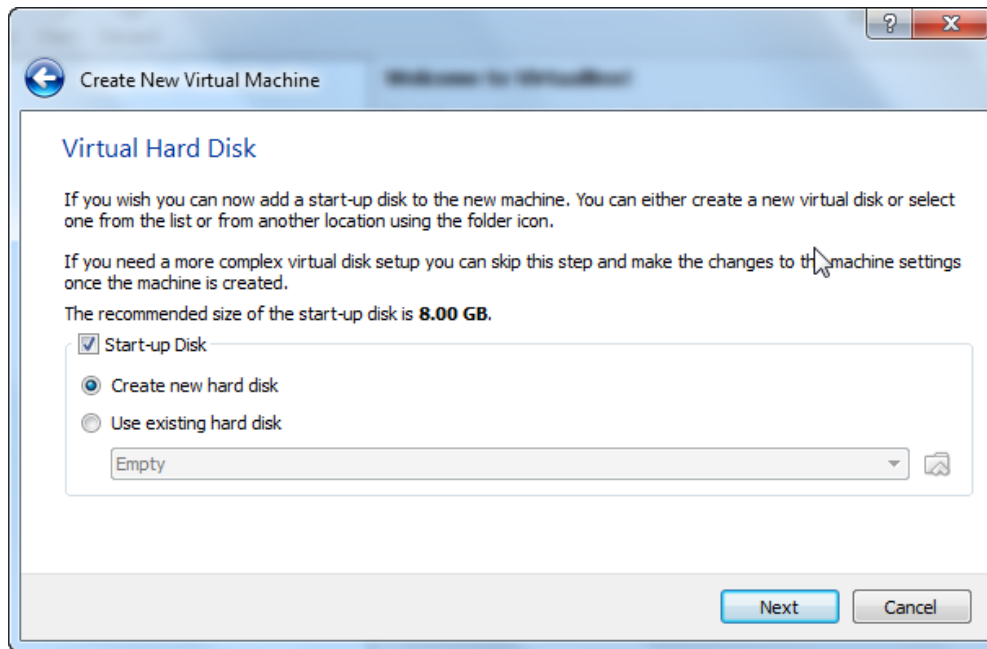
Pada isian dari parameter “Name”, masukkan nama pengenal dari VM yang dibuat, sebagai contoh Mikrotik. Selanjutnya pada pilihan jenis dan versi sistem operasi yang diinstalasi, akan secara otomatis tertera, *Operating System* adalah *Other*, dan *Version* adalah *Other/Unknown*. Klik tombol **Next**, untuk melanjutkan.

6. Tampil kotak dialog *Memory*, yang digunakan untuk menentukan besar memory yang dialokasikan untuk VM yang dibuat, seperti ditunjukkan pada gambar berikut:



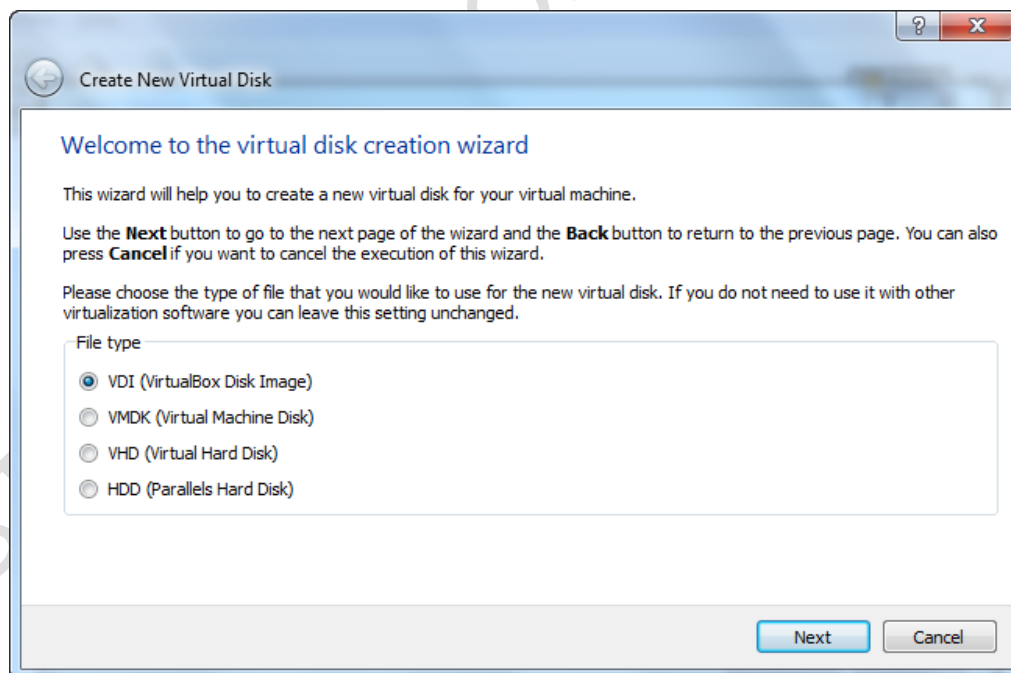
Pada isian dari parameter *Base Memory Size*, masukkan nilai besar memory yang dialokasikan, sebagai contoh alokasikan 512MB. Klik tombol **Next** untuk melanjutkan.

7. Tampil kotak dialog *Virtual Hard Disk*, untuk menambahkan hardisk pada VM, seperti ditunjukkan pada gambar berikut:



Cek pada pilihan *Start-up Disk*, dan pilih pada pilihan *Create new hard disk* untuk membuat hardisk baru pada virtual machine. Klik tombol **Next** untuk melanjutkan.

8. Tampil kotak dialog *Welcome to the virtual disk creation wizard*, untuk menentukan tipe file yang digunakan untuk menyimpan virtual disk yang baru, seperti ditunjukkan pada gambar berikut:



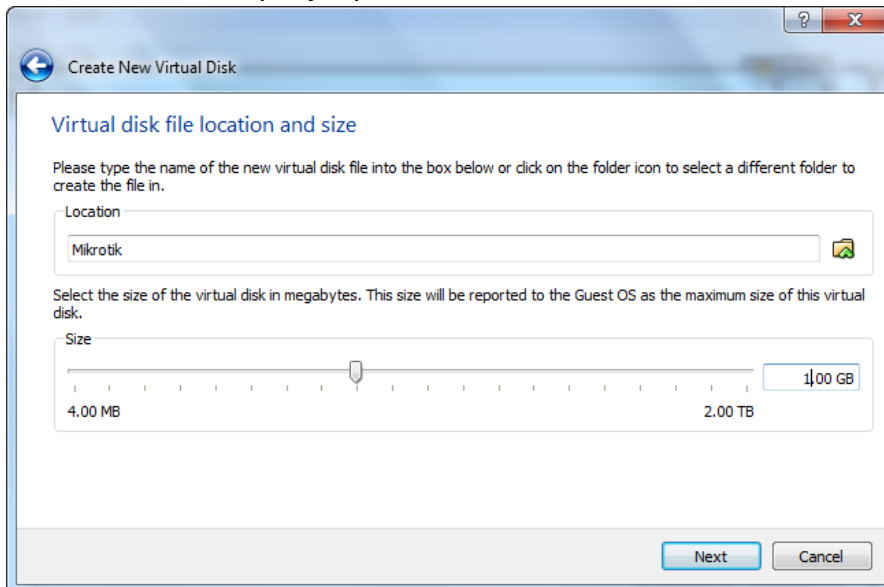
Terdapat 4 tipe file yang didukung oleh VirtualBox, yaitu:

- VDI (VirtualBox Disk Image), merupakan format yang dimiliki oleh VirtualBox.
- VMDK (Virtual Machine Disk), merupakan format yang paling populer dan digunakan oleh virtualisasi lainnya seperti VMWare.

- VHD (Virtual Hard Disk), merupakan format yang didukung oleh Microsoft.
- HDD (Parallels Hard Disk).

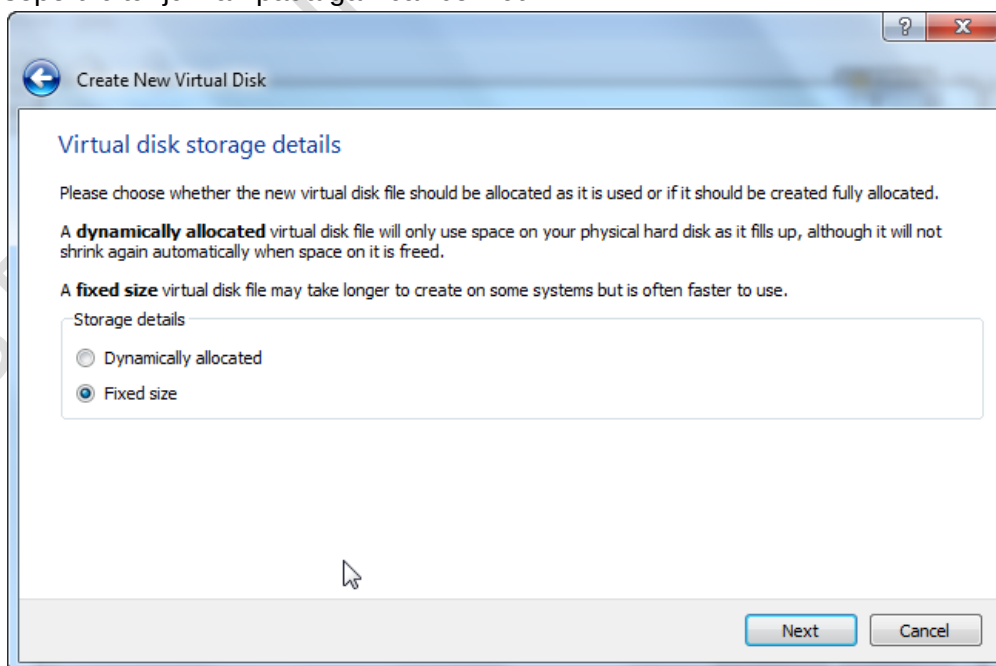
Pilih pada pilihan *VDI (VirtualBox Disk Image)*. Klik tombol **Next** untuk melanjutkan.

9. Tampil kotak dialog *Virtual disk file location and size*, yang digunakan untuk menentukan lokasi penyimpanan file, dan ukuran dari virtual disk.



Pada parameter *Location*, tentukan lokasi penyimpan file virtual disk, dan pada parameter *size* tentukan ukuran dari virtual disk, sebagai contoh 1 GB. Klik tombol **Next** untuk melanjutkan.

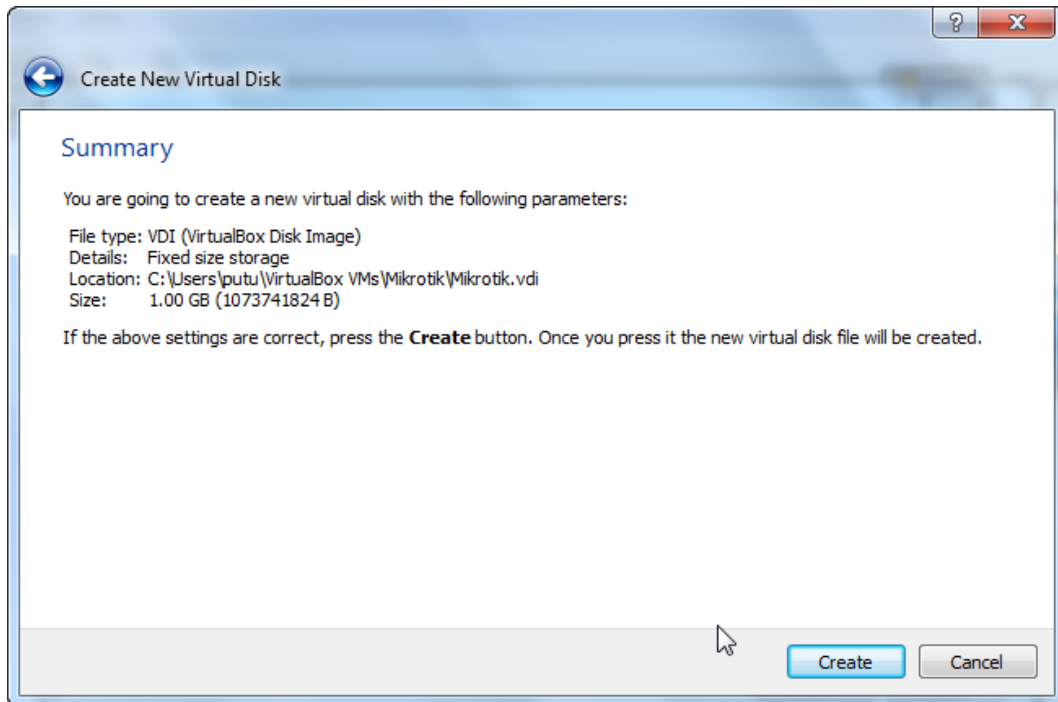
10. Tampil kotak dialog *Virtual disk storage details*, digunakan untuk menentukan apakah virtual disk yang baru dibuat dialokasikan berdasarkan penggunaan (dinamis), atau langsung sesuai dengan ukuran yang telah ditentukan diawal (tetap), seperti ditunjukkan pada gambar berikut:



Pada parameter *Storage details*, terdapat 2 pilihan yaitu *Dynamically allocated*, dan *Fixed Size*.

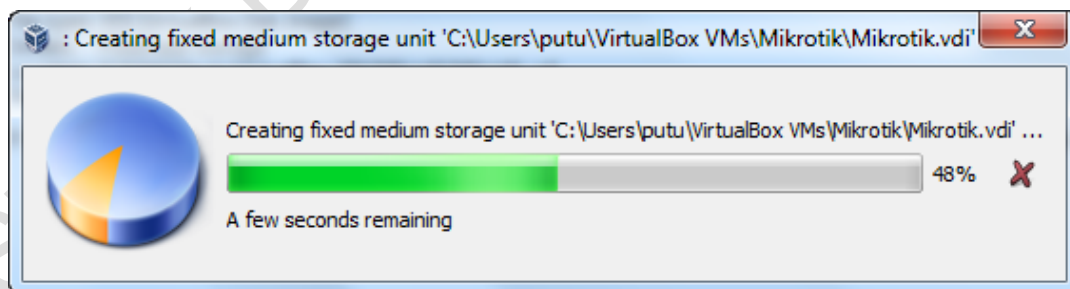
Pilih *Fixed size*. Klik **Next** untuk melanjutkan.

11. Tampil kotak dialog *Summary*, yang menampilkan informasi ringkasan parameter-parameter yang telah dikonfigurasi sebelumnya, seperti ditunjukkan pada gambar berikut:

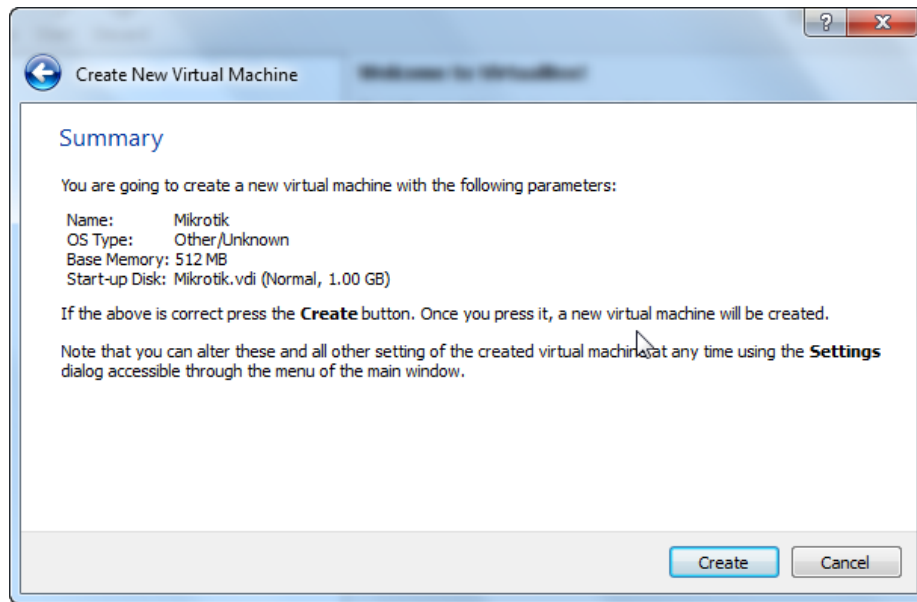


Klik tombol **Create**, untuk memulai pembuatan VM.

12. Tampilkan kotak dialog yang menampilkan informasi proses pembuatan virtual disk, seperti ditunjukkan pada gambar berikut:

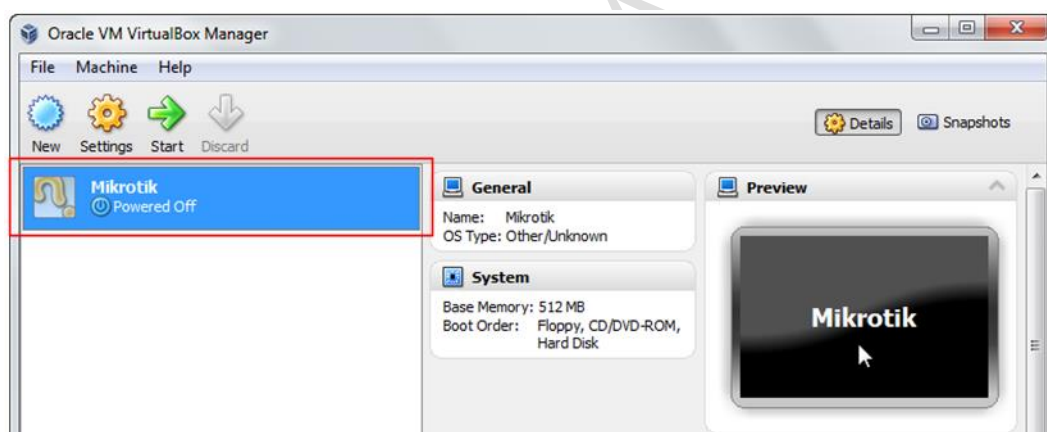


13. Tampil kotak dialog *Summary*, yang menampilkan informasi ringkasan pembuatan VM, seperti ditunjukkan pada gambar berikut:

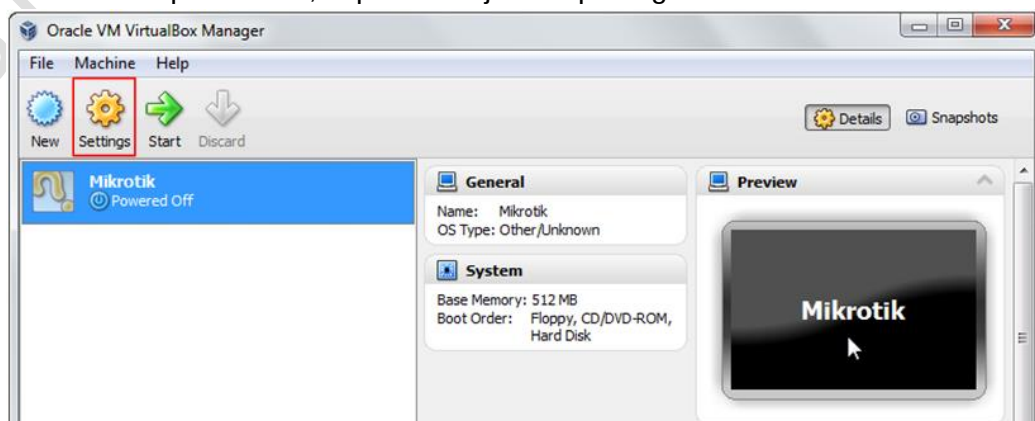


Klik tombol **Create**, untuk melanjutkan.

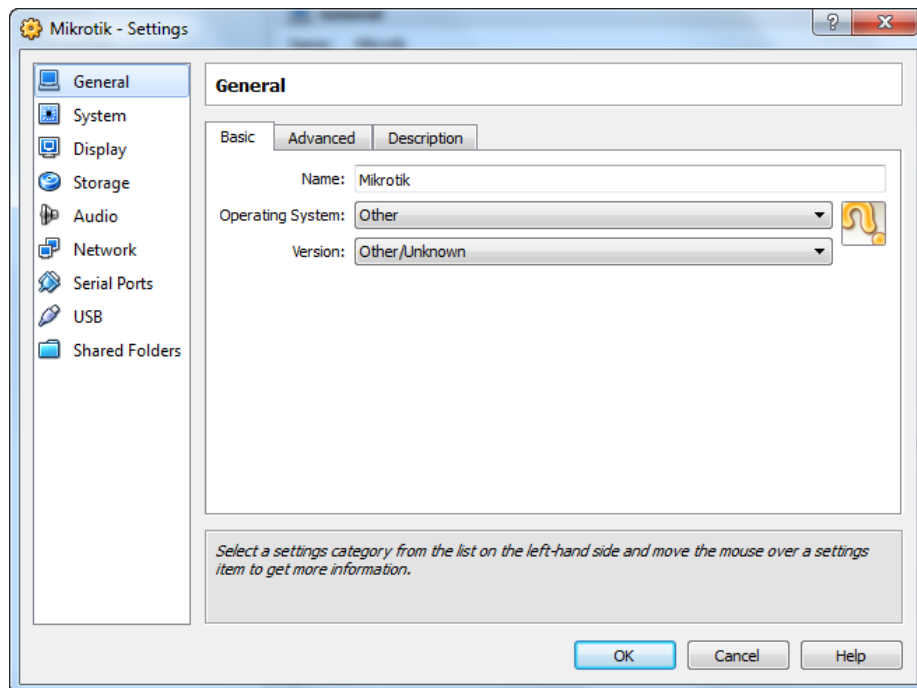
14. Tampil kotak dialog aplikasi *Oracle VM VirtualBox Manager*, terlihat pada panel sebelah kiri virtual machine yang dibuat yaitu dengan nama pengenal *Mikrotik*.



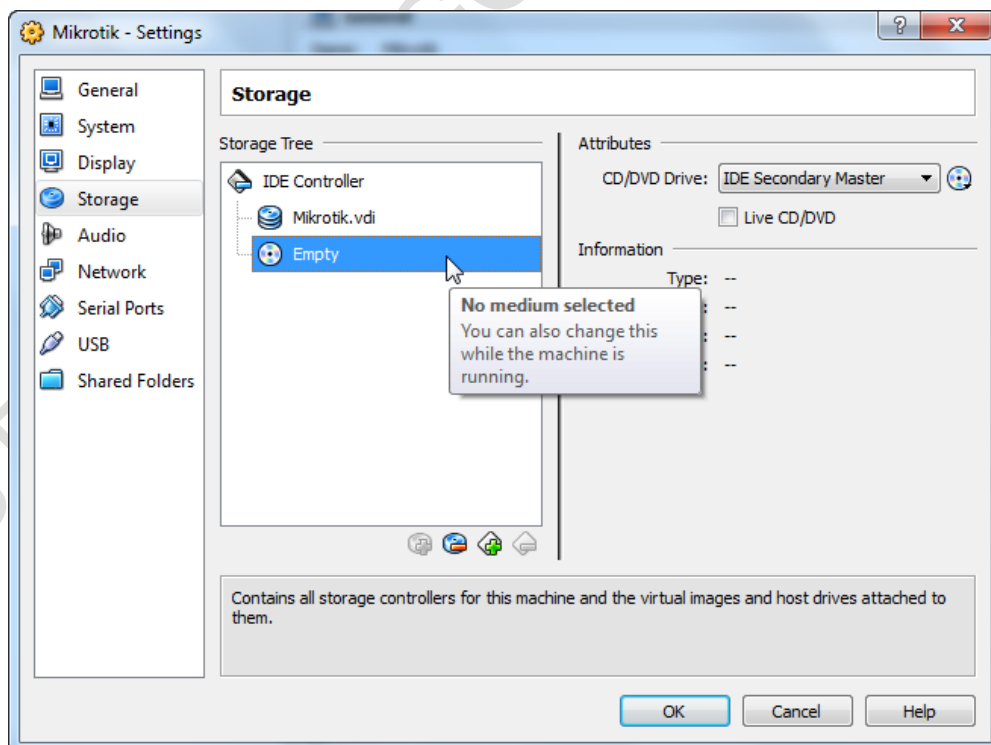
15. Pada toolbar dari Oracle VM VirtualBox Manager, klik tombol *Settings*, untuk mengatur lokasi file ISO instalasi Mikrotik RouterOS yang digunakan untuk instalasi, dan Network Adapter untuk menentukan jenis adapter, serta mengaktifkan dukungan Network Adapter kedua, seperti ditunjukkan pada gambar berikut:



16. Tampil kotak dialog *Settings*, yang digunakan untuk melakukan pengaturan lebih lanjut pada Virtual Machine yang telah dibuat, seperti ditunjukkan pada gambar berikut:

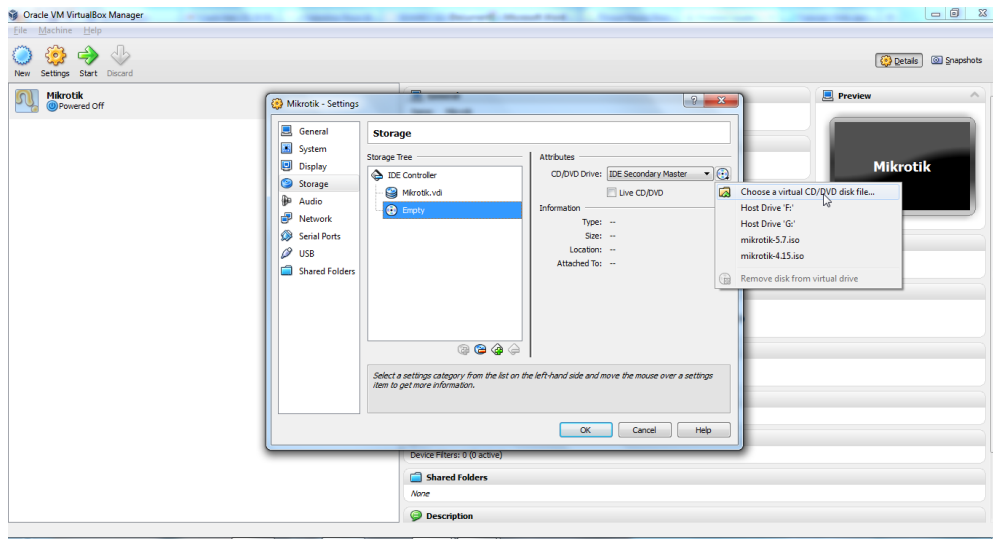


17. Pada panel sebelah kiri dari kotak dialog *Settings*, pilih *Storage*, untuk mengatur lokasi file ISO Mikrotik RouterOS, seperti ditunjukkan pada gambar berikut:

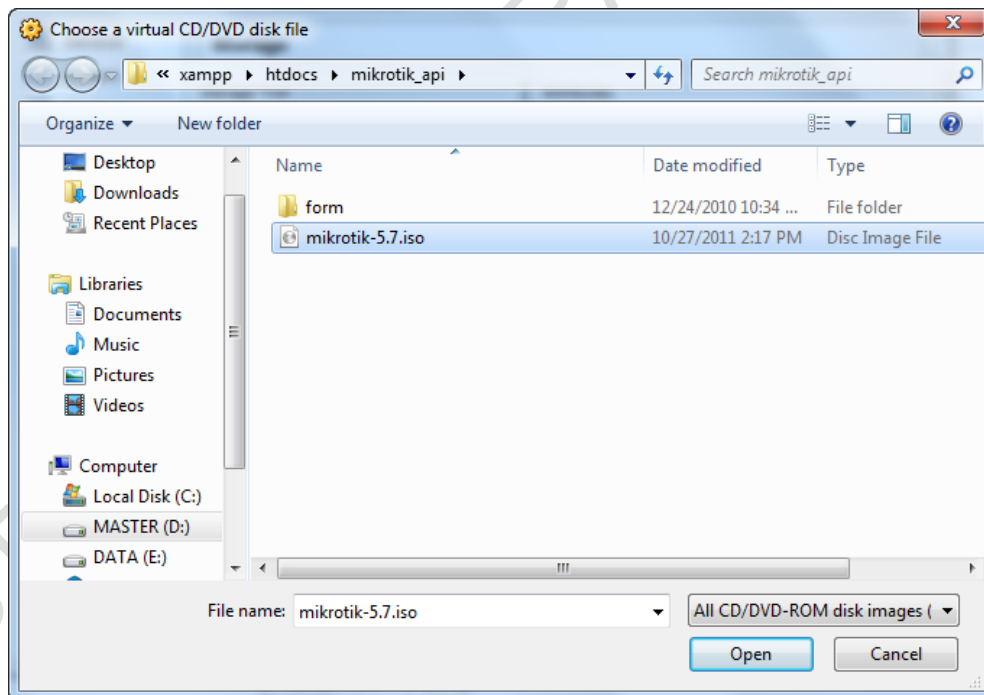


Pada panel *Storage* sebelah kanan, lakukan konfigurasi dari parameter-paramater berikut:

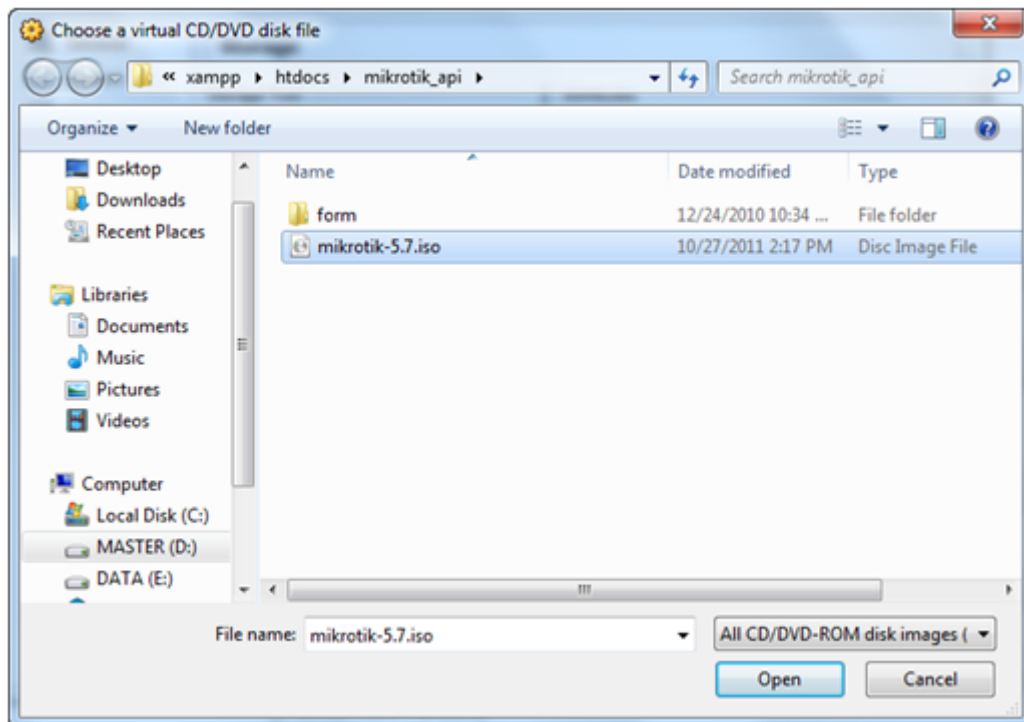
- Pada parameter *Storage Tree*, pilih *IDE Controller > CDROM/DVDROM Drive (Empty)*.
- Pada parameter *Attributes* sebelah kanan, pilih *Choose a virtual CD/DVD disk file*, seperti ditunjukkan pada gambar berikut:



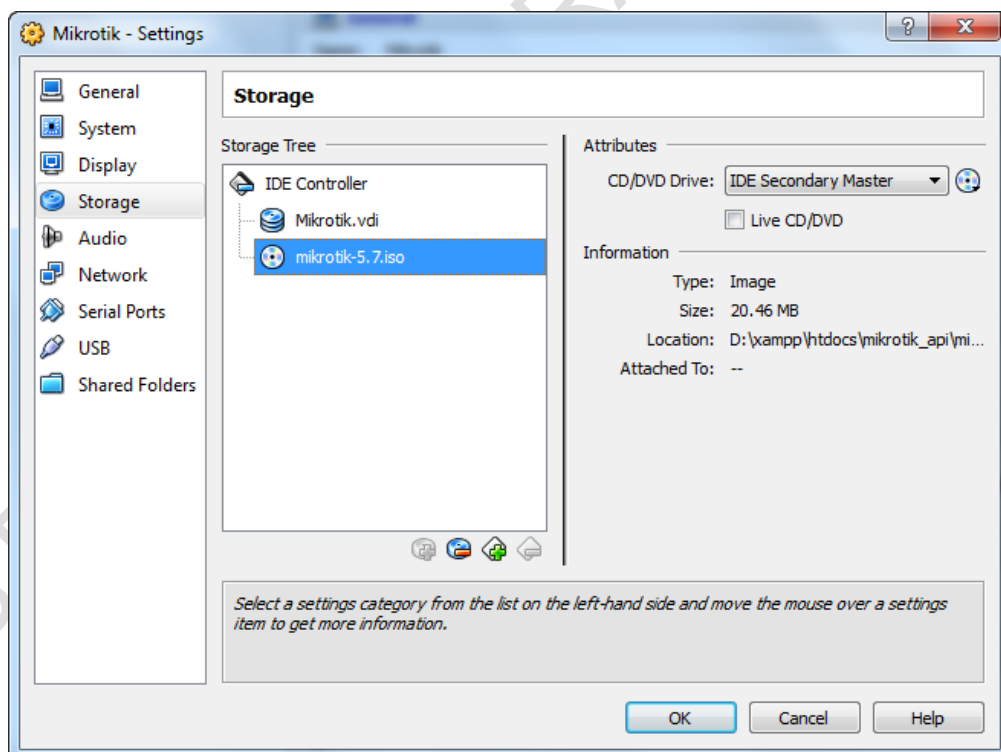
Selanjutnya tampil kotak dialog untuk mengarahkan ke lokasi file ISO Mikrotik RouterOS, seperti ditunjukkan pada gambar berikut:



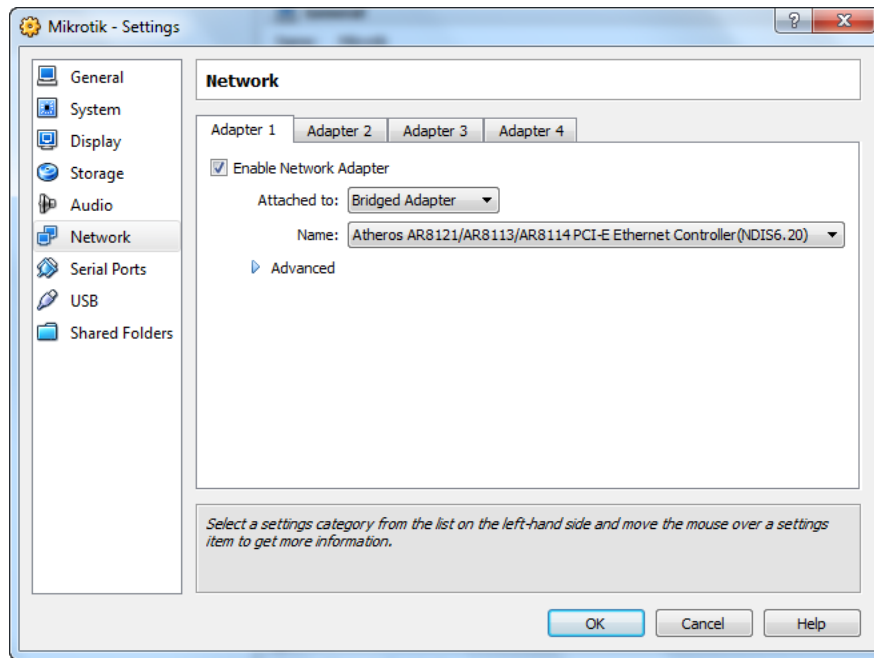
Sebagai contoh, pilih file *mikrotik-5.7.iso*, dan klik tombol Open, seperti ditunjukkan pada gambar berikut:



Hasil dari pengaturan CDROM/DVD yang mengarah ke lokasi file ISO Mikrotik RouterOS ditunjukkan seperti pada gambar berikut:



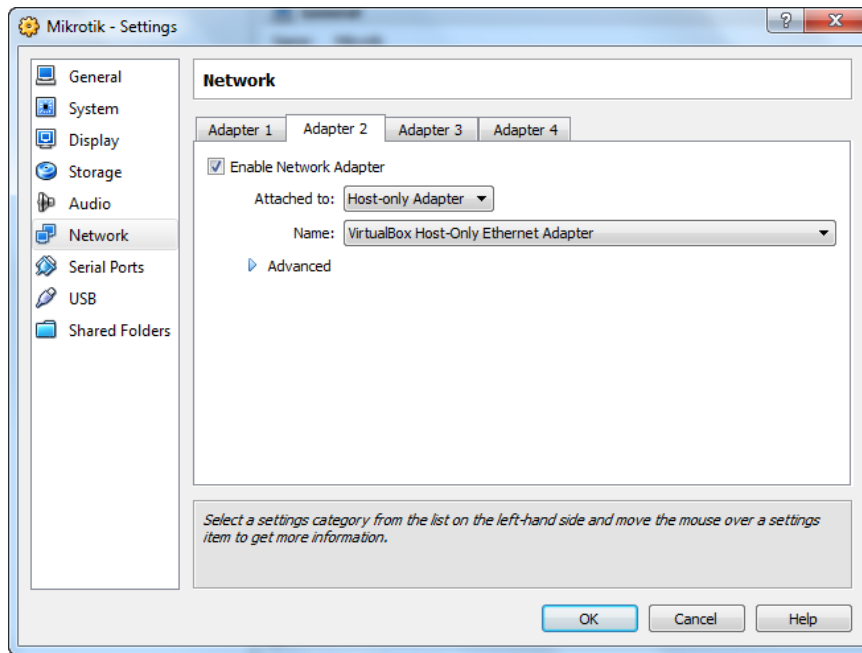
Pilih tab *Adapter 1*, untuk mengatur konfigurasi Network Adapter yang pertama, seperti ditunjukkan pada gambar berikut:



Konfigurasi parameter-paramater berikut:

- Cek pada pilihan Enable Network Adapter, untuk mengaktifkan Network Adapter pertama.
- Pilih Bridged Adapter pada pilihan parameter Attached to:.
- Pilih jenis *Network Adapter Local Area Connection* fisik yang terinstalasi pada sistem operasi host, sebagai contoh *Atheros AR8121/AR8112/AR8114 PCI-E Ethernet Controller (NDIS6.20)* pada pilihan parameter Name:.

Selanjutnya pilih tab *Adapter 2*, untuk mengatur konfigurasi Network Adapter yang kedua, seperti ditunjukkan pada gambar berikut:



Konfigurasi parameter-paramater berikut:

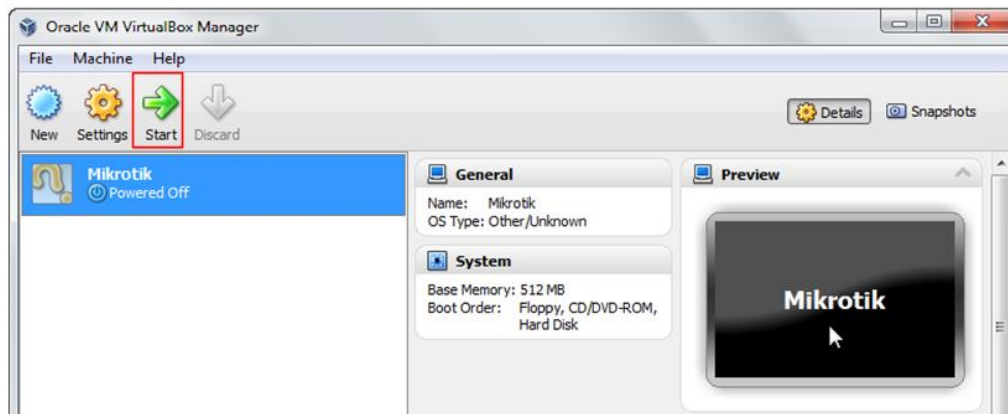
- Cek pada pilihan *Enable Network Adapter*, untuk mengaktifkan Network Adapter kedua.
- Pilih *Host-only Adapter* pada pilihan parameter *Attached to:*. Jenis adapter ini digunakan agar *guest operating system Mikrotik* dapat berkomunikasi dengan *host operating system Windows*.
- Pilih *VirtualBox Host-Only Ethernet Adapter* pada pilihan parameter *Name:*.

Klik tombol **OK**, untuk menyimpan perubahan yang telah dilakukan.

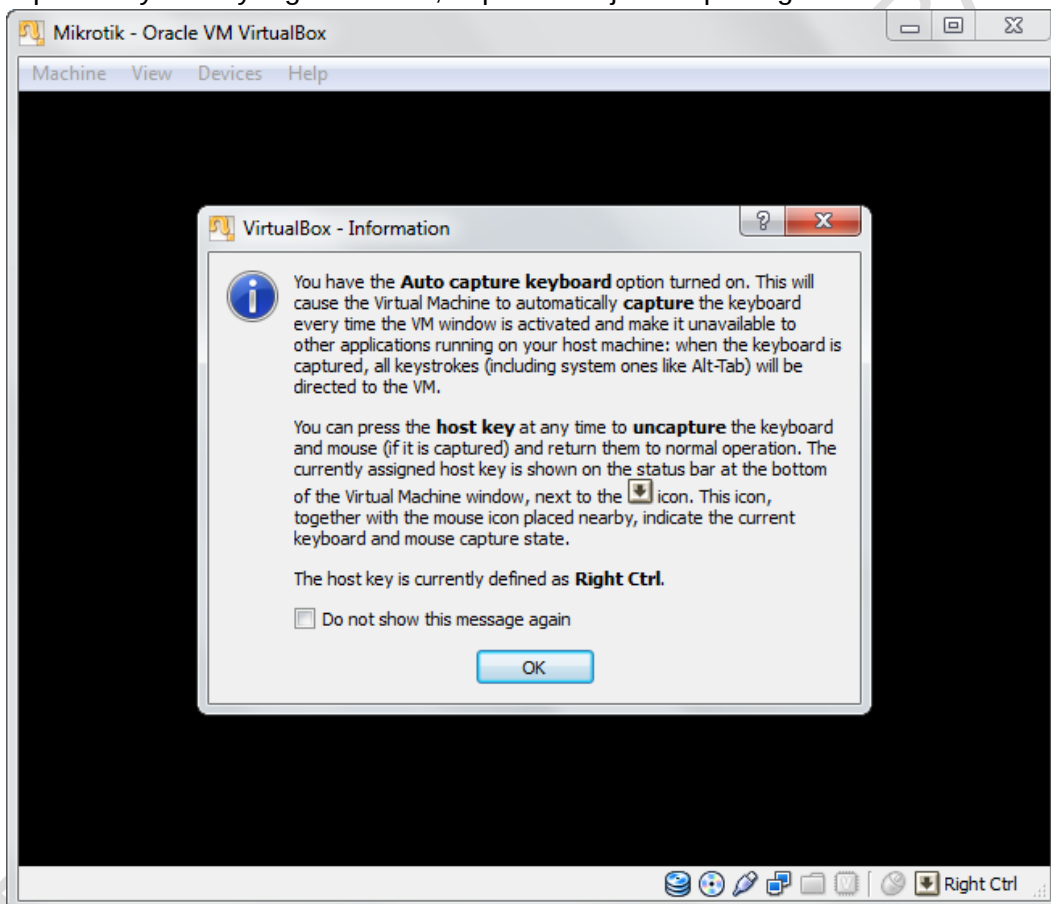
D. Instalasi & Konfigurasi Dasar Mikrotik RouterOS pada VirtualBox

Adapun langkah-langkah instalasi & konfigurasi dasar Mikrotik RouterOS pada VirtualBox adalah sebagai berikut:

1. Aktifkan VM yang telah dibuat, untuk melakukan instalasi Mikrotik RouterOS, dengan memilih nama VM yang telah dibuat pada tahap sebelumnya, yaitu *Mikrotik*, yang terdapat pada panel sebelah kiri. Selanjutnya pada toolbar klik tombol *Start*, untuk mengaktifkan VM, seperti ditunjukkan pada gambar berikut:

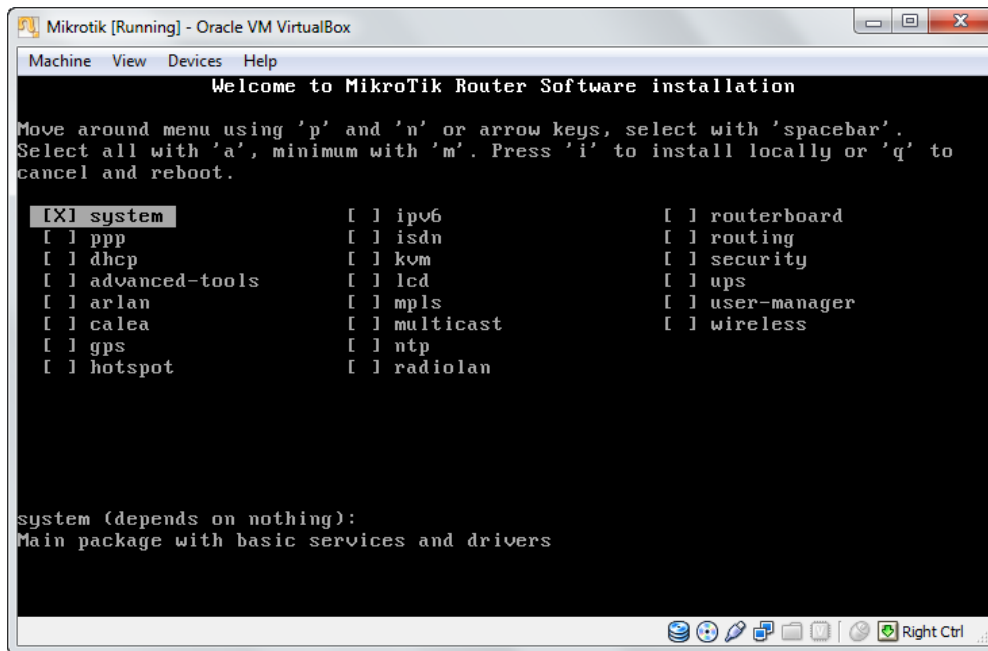


2. Tampil kotak dialog *VirtualBox Information* yang menginformasikan tentang fitur Auto capture keyboard yang telah aktif, seperti ditunjukkan pada gambar berikut:



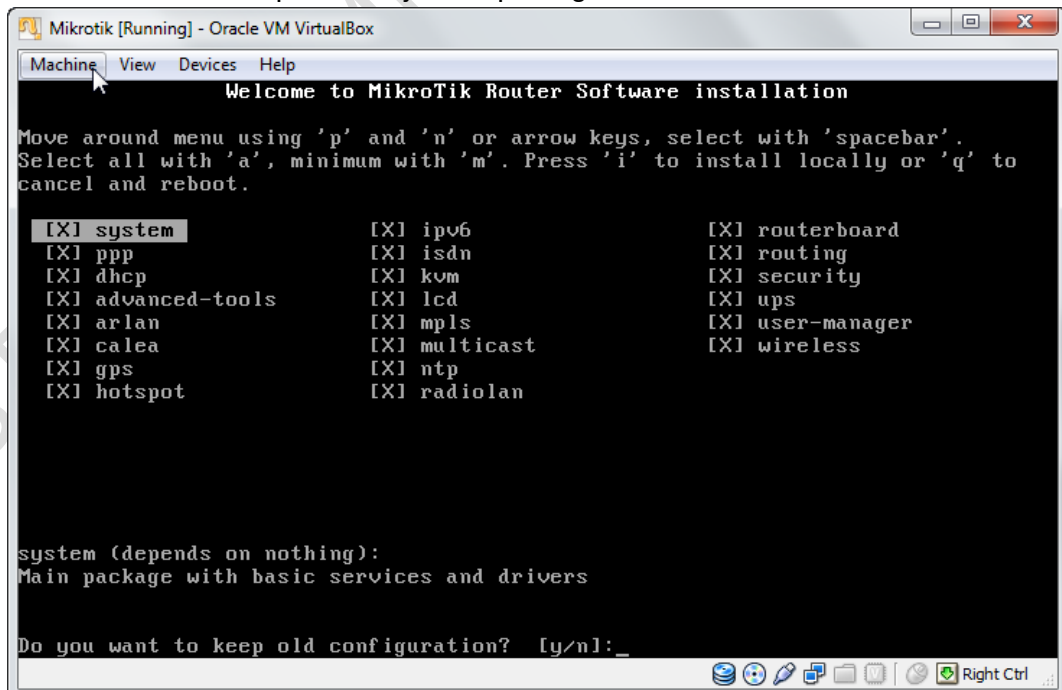
Klik tombol **OK**, untuk melanjutkan.

3. Terlihat pesan-pesan booting instalasi Mikrotik RouterOS. Selanjutnya tampil menu untuk memilih paket-paket Mikrotik RouterOS yang akan diinstalasi, seperti ditunjukkan pada gambar berikut:



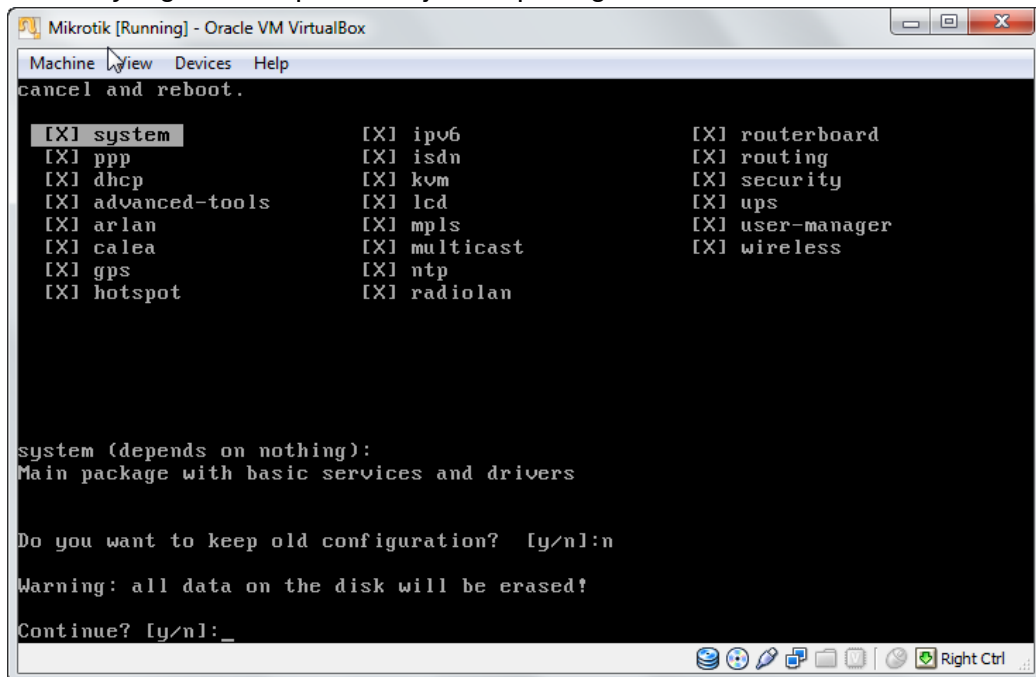
Terdapat beberapa tombol navigasi yang digunakan untuk berpindah-pindah ke pilihan paket-paket Mikrotik RouterOS yang akan diinstalasi, yaitu tombol 'p' (*previous*), 'n' (*next*), atau dapat menggunakan tombol panah (*arrow keys*). Paket-paket yang akan diinstalasi dapat dipilih menggunakan tombol spasi (*spacebar*), dan sebaliknya untuk menghilangkan seleksi pada paket terpilih juga menggunakan tombol spasi. Untuk memilih semua paket, digunakan tombol 'a' (*all*), dan sebaliknya untuk menginstalasi paket *minimum*, digunakan tombol 'm'.

Sebagai contoh, tekan 'a' untuk menginstalasi semua paket. Tekan 'i' (**install**) untuk memulai instalasi, seperti ditunjukkan pada gambar berikut:

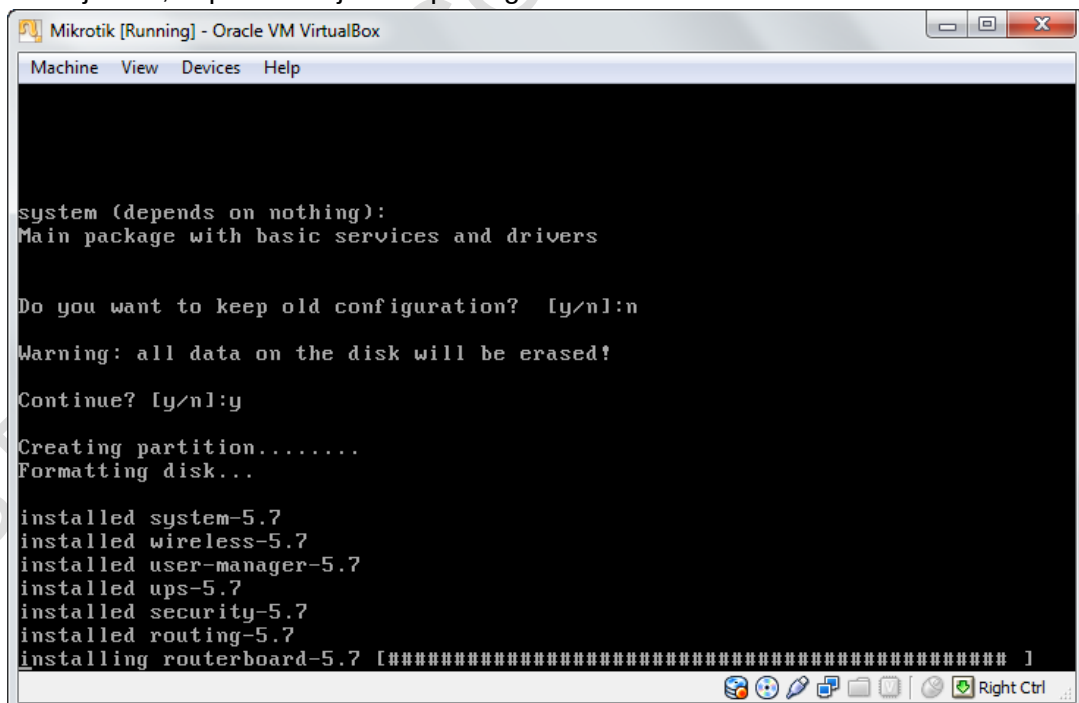


Tampil pesan *Do you want to keep old configuration? [y/n]:* yang menginformasikan apakah akan melakukan penyimpanan terhadap konfigurasi Mikrotik yang

sebelumnya telah dilakukan. Ketik 'n', untuk mengabaikan penyimpanan konfigurasi Mikrotik yang lama, seperti ditunjukkan pada gambar berikut:

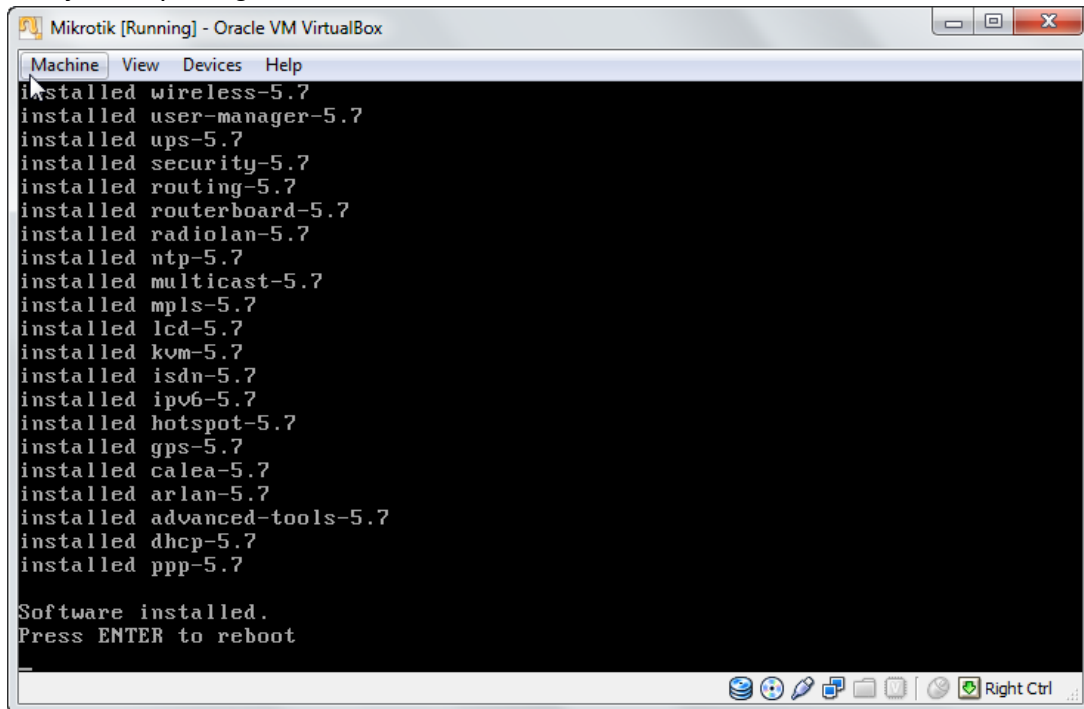


Tampil pesan Warning: all data on the disk will be erased! Continue? [y/n] yang memperingatkan bahwa semua data pada disk akan dihapus. Ketik 'y' untuk melanjutkan, dan sebaliknya 'n' untuk membatalkan instalasi. Ketik 'y', untuk melanjutkan, seperti ditunjukkan pada gambar berikut:

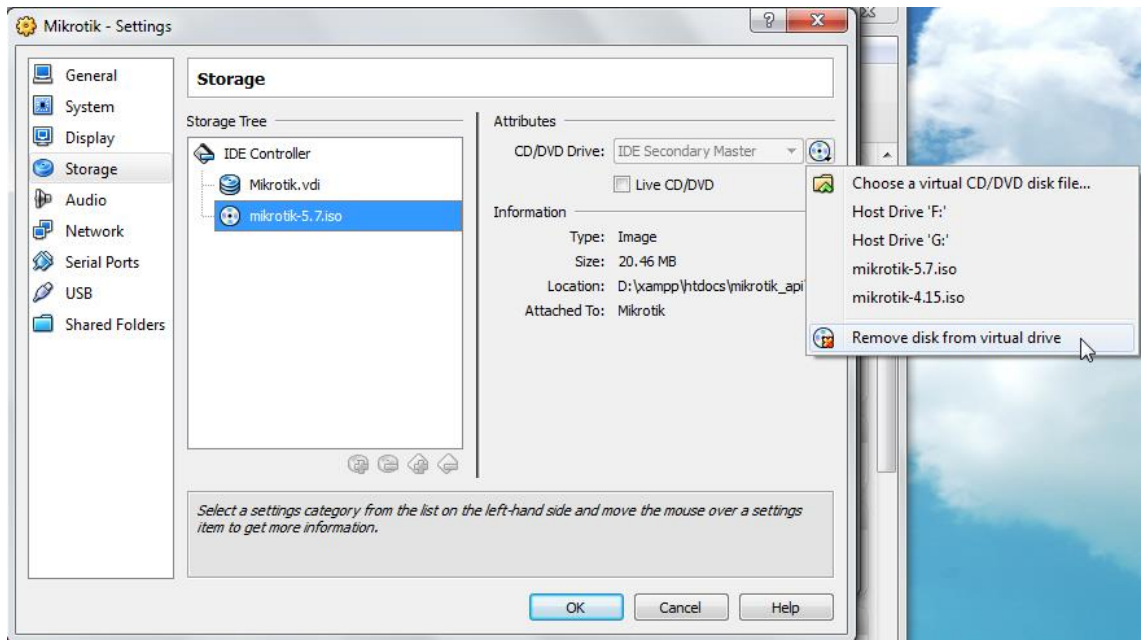


Selanjutnya akan terlihat proses pembuatan partisi, pemformatan disk, dan instalasi paket-paket Mikrotik RouterOS yang dipilih.

Tampil informasi yang menyatakan perangkat lunak telah terinstalasi, dan selanjutnya lakukan penekanan tombol *Enter* untuk melakukan *reboot*, seperti ditunjukkan pada gambar berikut:



Sebelum melakukan penekanan tombol **Enter**, lakukan pengaturan agar VM melepaskan akses pada CDROM/DVD yang masih mengarah ke lokasi file ISO Mikrotik RouterOS. Pada menu bar pilih menu **Machine > Settings >** pada panel sebelah kiri pilih **Storage**, maka selanjutnya pada panel sebelah kanan terlihat detail pengaturan untuk *Storage*. Pada panel detail *Storage* sebelah kanan terdapat bagian *Storage Tree*, lakukan seleksi pada pilihan **IDE Controller > CD/DVD Mikrotik-5.7.iso**. Selanjutnya pada bagian *Attributes*, lakukan perubahan pilihan pada parameter *CD/DVD Drive* ke *Remove disk from virtual drive*, seperti ditunjukkan pada gambar berikut:



Klik tombol **OK**, untuk menyimpan perubahan. Selanjutnya lakukan penekanan tombol **Enter** untuk melakukan *reboot*.

4. Setelah proses booting selesai, maka akan tampil menu login dari Mikrotik RouterOS. Gunakan user '**admin**', dan tanpa password (kosong) untuk login, seperti ditunjukkan pada gambar berikut:

```
MikroTik 5.7
MikroTik Login: admin
Password: _
```

Tekan *Enter* untuk menampilkan prompt dari Mikrotik RouterOS.

Selamat, Anda telah sukses membuat virtualisasi Mikrotik RouterOS pada VirtualBox



BAB 2**ADMINISTRASI & KONFIGURASI
FITUR-FITUR MIKROTIK ROUTEROS SEBAGAI GATEWAY****Tujuan:**

Mahasiswa memahami mengenai administrasi dan konfigurasi Mikrotik RouterOS sebagai gateway untuk koneksi Internet.

Materi:**Akses ke Mikrotik**

Untuk dapat mengakses mikrotik dapat dilakukan melalui beberapa cara yaitu:

- a) Console (lokal).
- b) Web.
- c) Telnet.
- d) SSH (dengan bantuan program aplikasi Putty).
- e) Winbox (program berbasis GUI untuk mengkonfigurasi Mikrotik).

Sebelum dapat melakukan konfigurasi, Anda harus melalui proses otentikasi login terlebih dahulu. Default user dari mikrotik adalah:

User : **admin**

Password : **kosong (blank)**

Diperlukan konfigurasi pengalamatan IP pada salah satu interface dari Mikrotik, sebelum dapat mengakses Mikrotik secara remote melalui antar muka web, telnet, SSH, atau winbox.

Konfigurasi Pengalamatan IP pada interface

Untuk mengkonfigurasi alamat IP pada interface dari mikrotik, eksekusi perintah berikut:

```
[admin@mikrotik] > interface print
```

```
[admin@mikrotik] > ip address add address=192.168.1.1/24  
interface=ether1
```

Untuk memverifikasi alamat IP yang telah dikonfigurasi pada interface ether1, eksekusi perintah berikut:

```
[admin@mikrotik] > ip address print
```

Konfigurasi Dasar Mikrotik

1. Mengubah hostname:

```
[admin@mikrotik] > system identity set name=stmikbumigora.intra.net
```

2. Mengubah password user:

```
[admin@stmikbumigora.intra.net] > password
```

old password:

new password: *****

retype new password: *****

Mikrotik mengatur user/pengguna yang terhubung ke router dari console, serial, telnet, SSH atau winbox. User dapat diotentikasi menggunakan user yang tersimpan di lokal atau di server RADIUS. Masing-masing user, masuk sebagai anggota dari group, yang menentukan hak akses dari user tersebut. User default adalah *admin*, sebagai anggota dari group "**full**".

Untuk menampilkan user yang ada di Mikrotik, eksekusi perintah berikut:

```
[admin@stmikbumigora.intra.net] > user print
```

Flags: X - disabled

```
#
GROUP
ADDRESS

0    ;;; system default user

    admin
full
```

Terdapat 3 group default bawaan dari Mikrotik yaitu:

- a) Read
- b) Write
- c) Full

Untuk menampilkan informasi group yang ada, eksekusi perintah berikut:

```
[admin@stmikbumigora.intra.net] > user group print

0                                     name="read"
policy=local,telnet,ssh,reboot,read,test,winbox,password,web,sniff,!
ftp,!write,!policy

1                                     name="write"
policy=local,telnet,ssh,reboot,read,write,test,winbox,password,web,s
niff,!ftp,!policy

2                                     name="full"
policy=local,telnet,ssh,ftp,reboot,read,write,policy,test,winbox,pas
sword,web,sniff
```

Untuk membuat user baru dengan hak akses read, eksekusi perintah berikut:

```
[admin@stmikbumigora.intra.net] > user add name=operator group=read
```

Untuk menampilkan informasi user yang ada di Mikrotik, eksekusi perintah berikut:

```
[admin@stmikbumigora.intra.net] > user print
```

Flags: X - disabled

#	NAME
0	admin
1	operator

GROUP
ADDRESS

0 ;;; system default user
full
0.0.0.0/0

1
read
0.0.0.0/0

3. Menampilkan informasi waktu dari sistem:

```
[admin@stmikbumigora.intra.net] > system clock print
```

time: 19:22:03

date: dec/26/2009

time-zone-name: "manual"

gmt-offset: +00:00

4. Menampilkan informasi interface (LAN Card):

```
[admin@stmikbumigora.intra.net] > interface print
```

Flags: D - dynamic, X - disabled, R - running, S - slave

#	NAME
0	ether1

TYPE MTU

ether 1500

```

1                                R                                ether2
ether                            1500

```

Terlihat ada 2 interface yang terpasang yaitu ether1 & ether2.

5. Mengubah nama interface untuk mempermudah mengidentifikasi atau memberikan gambaran mengenai interface:

a. Mengubah interface ether1 dengan nama "local", karena interface ini terhubung ke LAN

```
[admin@stmikbumigora.intra.net] > interface set ether1 name=local
```

b. Mengubah interface ether2 dengan nama "public", karena interface ini terhubung ke Internet

```
[admin@stmikbumigora.intra.net] > interface set ether2 name=public
```

6. Menampilkan informasi interface:

```
[admin@stmikbumigora.intra.net] > interface print
```

Flags: D - dynamic, X - disabled, R - running, S - slave

#	TYPE	MTU		NAME
0	ether	1500	R	local
1	ether	1500	R	public

7. Mengatur alamat ip pada interface local yang terhubung ke LAN dengan alamat IP 192.168.1.1 dan subnetmask 255.255.255.0:

```
[admin@stmikbumigora.intra.net] > ip address add address
192.168.1.1/24 interface=local
```

8. Mengatur alamat ip pada interface public yang terhubung ke Internet dengan alamat IP 202.134.1.1 dan subnetmask 255.255.255.0:

```
[admin@stmikbumigora.intra.net] > ip address add address
202.134.1.1/24 interface=public
```

9. Menampilkan informasi alamat-alamat ip pada masing-masing interface:

```
[admin@stmikbumigora.intra.net] > ip address print
```

Flags: X - disabled, I - invalid, D - dynamic

#	ADDRESS	NETWORK	BROADCAST	INTERFACE
0	192.168.1.1/24	192.168.1.0	192.168.1.255	local
1	202.134.1.1/24	202.134.1.0	202.134.1.255	public

10. Menampilkan informasi tabel routing

```
[admin@stmikbumigora.intra.net] > ip route print
```

Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,

B - blackhole, U - unreachable, P - prohibit

#	DST-ADDRESS	PREF-SRC	GATEWAY-STATE	GATEWAY
DISTANCE	INTERFACE			
0	ADC	192.168.1.0/24		192.168.1.1
0	local			
1	ADC	202.134.1.0/24		202.134.1.1
0	public			

11. Mengatur alamat default gateway.

```
[admin@stmikbumigora.intra.net] > ip route add gateway=202.134.1.2
```

12. Menampilkan informasi routing tabel setelah ditambahkan konfigurasi default gateway

```
[admin@stmikbumigora.intra.net] > ip route print
```

Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,

B - blackhole, U - unreachable, P - prohibit

#	DST-ADDRESS	PREF-SRC	GATEWAY-STATE	GATEWAY
0 A S	0.0.0.0/0		reachable	202.134.1.2
1	public			
1	ADC	192.168.1.0/24		192.168.1.1
0	local			
2	ADC	202.134.1.0/24		202.134.1.1
0	public			

Terlihat terdapat sebuah entry tambahan yang berfungsi sebagai default route. Paket-paket yang ditujukan diluar dari alamat jaringan dari masing-masing interface local (192.168.1.0/24) & public (202.134.1.0/24) akan dirutekan ke alamat default gateway 202.134.1.2 melalui interface keluar yaitu interface public.

13. Menampilkan informasi Domain Name System (DNS)

DNS merupakan sebuah system yang melakukan pemetaan/translasi nama domain ke alamat IP dan sebaliknya.

```
[admin@stmikbumigora.intra.net] > ip dns print
```

```
primary-dns: 0.0.0.0
```

```
secondary-dns: 0.0.0.0
```

```
allow-remote-requests: no
```

```
max-udp-packet-size: 512
```

```
cache-size: 2048KiB
```

```
cache-max-ttl: 1w
```

```
cache-used: 4KiB
```

14. Mengatur alamat Domain Name System (DNS).

Primary DNS merupakan server DNS utama, sedangkan secondary DNS adalah server DNS cadangan sebagai backup apabila server DNS utama mengalami permasalahan.

```
[admin@stmikbumigora.intra.net] > ip dns set primary-  
dns=202.134.1.10
```

```
[admin@stmikbumigora.intra.net] > ip dns set secondary-  
dns=61.94.192.12
```

Versi Mikrotik yang lebih baru, pengaturan DNS mengalami perubahan

```
[admin@workshop] > ip dns set servers=202.134.1.2,61.94.192.12
```

15. Menampilkan informasi DNS

```
[admin@stmikbumigora.intra.net] > ip dns print
```

```
primary-dns: 202.134.1.10
```

```
secondary-dns: 61.94.192.12
```

```
allow-remote-requests: no
```

```
max-udp-packet-size: 512
cache-size: 2048KiB
cache-max-ttl: 1w
cache-used: 5KiB
```

setelah dikonfigurasi pada langkah sebelumnya:

16. Memverifikasi koneksi ke alamat IP dari default gateway

```
[admin@stmikbumigora.intra.net] > ping 202.134.1.2
202.134.1.2 64 byte ping: ttl=128 time=13 ms
202.134.1.2 64 byte ping: ttl=128 time=1 ms
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 1/7.0/13 m
```

17. Memverifikasi koneksi ke alamat IP dari server DNS primary

```
[admin@stmikbumigora.intra.net] > ping 202.134.1.10
202.134.1.10 64 byte ping: ttl=128 time=1 ms
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 1/1.0/1 ms
```

18. Memverifikasi koneksi ke alamat IP dari server DNS secondary:

```
[admin@stmikbumigora.intra.net] > ping 61.94.192.12
61.94.192.12 64 byte ping: ttl=128 time=3 ms
61.94.192.12 64 byte ping: ttl=128 time<1 ms
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0/1.5/3 ms
```

19. Menampilkan informasi Network Address Translation (NAT)

NAT digunakan untuk mengubah alamat IP sumber atau alamat IP tujuan dari paket yang melalui router. NAT diperlukan oleh komputer client di LAN yang menggunakan alamat IP Private agar dapat mengakses Internet menggunakan alamat IP Public yang dimiliki oleh interface public dari router.

Terdapat 2 jenis NAT yaitu **source NAT** (*srcnat*), dan **destination NAT** (*dstnat*).

- a) **srcnat**: jenis NAT ini dilakukan pada paket yang berasal dari jaringan yang di-NAT. NAT router mengubah alamat IP sumber dari paket dengan alamat IP baru saat melalui router. Operasi sebaliknya dilakukan pada paket-paket balasan yang bergerak ke arah lainnya.
- b) **dstnat**: jenis NAT ini dilakukan pada paket yang ditujukan ke jaringan yang di-NAT. Umumnya digunakan untuk membuat host-host pada jaringan private dapat diakses dari Internet. NAT router yang melakukan dstnat akan mengubah alamat IP tujuan dari paket IP yang melalui router menuju jaringan private.

```
[admin@stmikbumigora.intra.net] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
```

20. Mengatur Network Address Translation (NAT).

```
[admin@stmikbumigora.intra.net] > ip firewall nat add
action=masquerade out-interface=public chain=srcnat
```

Keterangan paramater:

- **action:** aksi yang dilakukan pada paket yaitu masquerade, melakukan translasi alamat IP sumber ke alamat IP Public.
- **out-interface:** interface keluar bagi paket, yaitu interface public.
- **chain:** menentukan chain yang digunakan yaitu srcnat.

21. Menampilkan informasi NAT setelah dikonfigurasi pada langkah sebelumnya.

```
[admin@stmikbumigora.intra.net] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
0 chain=srcnat action=masquerade out-interface=public
```

22. Memverifikasi koneksi dari komputer client ke alamat IP default gateway

```
C:\>ping 202.134.1.2
```

Pinging 202.134.1.2 with 32 bytes of data:

```
Reply from 202.134.1.2: bytes=32 time=3ms TTL=127
```

```
Reply from 202.134.1.2: bytes=32 time=1ms TTL=127
```

```
Reply from 202.134.1.2: bytes=32 time=1ms TTL=127
```

```
Reply from 202.134.1.2: bytes=32 time=1ms TTL=127
```

Ping statistics for 202.134.1.2:

```
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 3ms, Average = 1ms

23. Memverifikasi koneksi dari komputer client ke alamat IP DNS Server utama:

```
C:\>ping 202.134.1.10
```

Pinging 202.134.1.10 with 32 bytes of data:

Reply from 202.134.1.10: bytes=32 time=1ms TTL=127

Ping statistics for 202.134.1.10:

Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 1ms, Average = 1ms

Control-C

^C

24. Memverifikasi koneksi dari komputer client ke alamat IP DNS Server kedua:

```
C:\>ping 61.94.192.12
```

Pinging 61.94.192.12 with 32 bytes of data:

Reply from 61.94.192.12: bytes=32 time=1ms TTL=127

Reply from 61.94.192.12: bytes=32 time<1ms TTL=127

Reply from 61.94.192.12: bytes=32 time=2ms TTL=127

Reply from 61.94.192.12: bytes=32 time=1ms TTL=127

Ping statistics for 61.94.192.12:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 2ms, Average = 1ms

25. Menampilkan informasi pool.

Pool merupakan jangkauan alamat yang akan didistribusikan oleh server Dynamic Host Configuration Protocol (DHCP) ke client-client di jaringan.

```
[admin@stmikbumigora.intra.net] > ip pool print
```

#	NAME
RANGES	

26. Membuat pool dengan nama "intranet-pool" dengan alokasi jangkauan ruang alamat IP yang didistribusikan mulai dari alamat IP 192.168.1.25 sampai dengan 192.168.1.50:

```
[admin@stmikbumigora.intra.net] > ip pool add name=intranet-pool  
ranges=192.168.1.25-192.168.1.50
```

27. Menampilkan informasi pool setelah dikonfigurasi pada langkah sebelumnya:

```
[admin@stmikbumigora.intra.net] > ip pool print
```

#	NAME
RANGES	
0	intranet-pool
192.168.1.25-192.168.1.50	

28. Menampilkan informasi alamat jaringan dan parameter-parameter lain dari server DHCP:

```
[admin@stmikbumigora.intra.net] > ip dhcp-server network print
```

#	ADDRESS	GATEWAY	DNS-SERVER	WINS-SERVER	DOMAIN
---	---------	---------	------------	-------------	--------

29. Mengatur alamat jaringan dari Server DHCP sesuai dengan alamat IP pada pool yang telah dibuat sebelumnya dan alamat default gateway serta alamat IP DNS:

```
[admin@stmikbumigora.intra.net] > ip dhcp-server network add
address=192.168.1.0/24 gateway=192.168.1.1 dns-
server=202.134.1.10,61.94.192.12 domain=telkom.net
```

30. Menampilkan informasi alamat jaringan dari Server DHCP setelah dikonfigurasi pada langkah sebelumnya:

```
[admin@stmikbumigora.intra.net] > ip dhcp-server network print
```

#	ADDRESS	GATEWAY	DNS-SERVER	WINS-SERVER	DOMAIN
0	192.168.1.0/24	192.168.1.1		202.134.1.10	telkom.net

31. Menerapkan pool dari Server DHCP yang telah dibuat sebelumnya pada interface local, agar client-client yang melakukan permintaan DHCP Request dapat memperoleh alamat IP dari Server DHCP:

```
[admin@stmikbumigora.intra.net] > ip dhcp-server add interface=local
address-pool=intranet-pool
```

32. Menampilkan informasi penerapan pool Server DHCP pada interface local:

```
[admin@stmikbumigora.intra.net] > ip dhcp-server print
```

Flags: X - disabled, I - invalid

#	NAME	INTERFACE	RELAY
ADDRESS-POOL	LEASE-TIME	ADD-ARP	
0	X dhcp1		local
	intranet-pool	3d	

33. Mengaktifkan Server DHCP "dhcp1" untuk pool "intranet-pool" pada interface local

```
[admin@stmikbumigora.intra.net] > ip dhcp-server enable 0
```

34. Menampilkan informasi penerapan pool server DHCP pada interface setelah diaktifkan pada langkah sebelumnya:

```
[admin@stmikbumigora.intra.net] > ip dhcp-server print
```

Flags: X - disabled, I - invalid

#	NAME	INTERFACE	RELAY
ADDRESS-POOL	LEASE-TIME	ADD-ARP	
0	dhcp1		local
	intranet-pool	3d	

BAB 3

IP HOTSPOT MIKROTIK ROUTEROS

Tujuan:

Mahasiswa memahami mengenai fitur hotspot di Mikrotik RouterOS.

Materi:**A. Pengenalan IP Hotspot**

Menurut manual Mikrotik, IP Hotspot merupakan sebuah fitur yang terdapat pada router Mikrotik yang dapat digunakan sebagai mekanisme untuk mengotentikasi & mengotorisasi pengguna ketika mengakses sumber daya jaringan. IP Hotspot tidak menyediakan fitur enkripsi pada trafik. Untuk melakukan login, pengguna dapat menggunakan browser apapun.

B. Konfigurasi IP Hotspot

Adapun langkah-langkah konfigurasi Mikrotik IP hotspot adalah sebagai berikut:

1. Mengatur fasilitas hotspot dari mikrotik

```
[admin@stmikbumigora.intra.net] > ip hotspot
```

```
[admin@stmikbumigora.intra.net] /ip hotspot> setup
```

```
Select interface to run HotSpot on
```

```
hotspot interface: local <--- antarmuka yang digunakan  
oleh hotspot
```

```
Set HotSpot address for interface
```

```
local address of network: 192.168.1.1/24 <-- mengatur alamat  
ip gateway bagi pengguna hotspot
```

```
masquerade network: yes <-- mengaktifkan fasilitas translasi  
alamat
```

```
Set pool for HotSpot addresses
```

```
address pool of network: 192.168.1.25-192.168.1.50 <--  
menentukan alamat pool yang digunakan
```

Select hotspot SSL certificate

select certificate: **none** <-- menentukan sertifikat Secure Socket Layer (SSL) yang digunakan

Select SMTP server

ip address of smtp server: **0.0.0.0** <-- mengatur alamat ip dari server SMTP

Setup DNS configuration

dns servers: **202.134.1.10,61.94.192.12** <-- mengatur alamat ip dari server dns

DNS name of local hotspot server

dns name: **telkom.net** <-- nama domain

Create local hotspot user

name of local hotspot user: **admin** <-- nama dari user hotspot lokal

password for the user: **123456** <-- password untuk user admin

2. Menampilkan informasi hotspot:

```
[admin@stmikbumigora.intra.net] > ip hotspot print
```

Flags: X - disabled, I - invalid, S - HTTPS

#	NAME	PROFILE	INTERFACE
	ADDRESS-POOL		IDLE-
	TIMEOUT		
0	hotspot1		local
	intranet-pool	hsprof1	5m

3. Menampilkan informasi user lokal pada hotspot:

```
[admin@stmikbumigora.intra.net] > ip hotspot user print
```

Flags: X - disabled, D - dynamic

#	SERVER	PROFILE	UPTIME	NAME
	ADDRESS			

```
0
default                                0s                                admin
```

4. Membuat user baru di hotspot:

```
[admin@stmikbumigora.intra.net] > ip hotspot user
```

```
[admin@stmikbumigora.intra.net] /ip hotspot user> add
name=putu password=123456
```

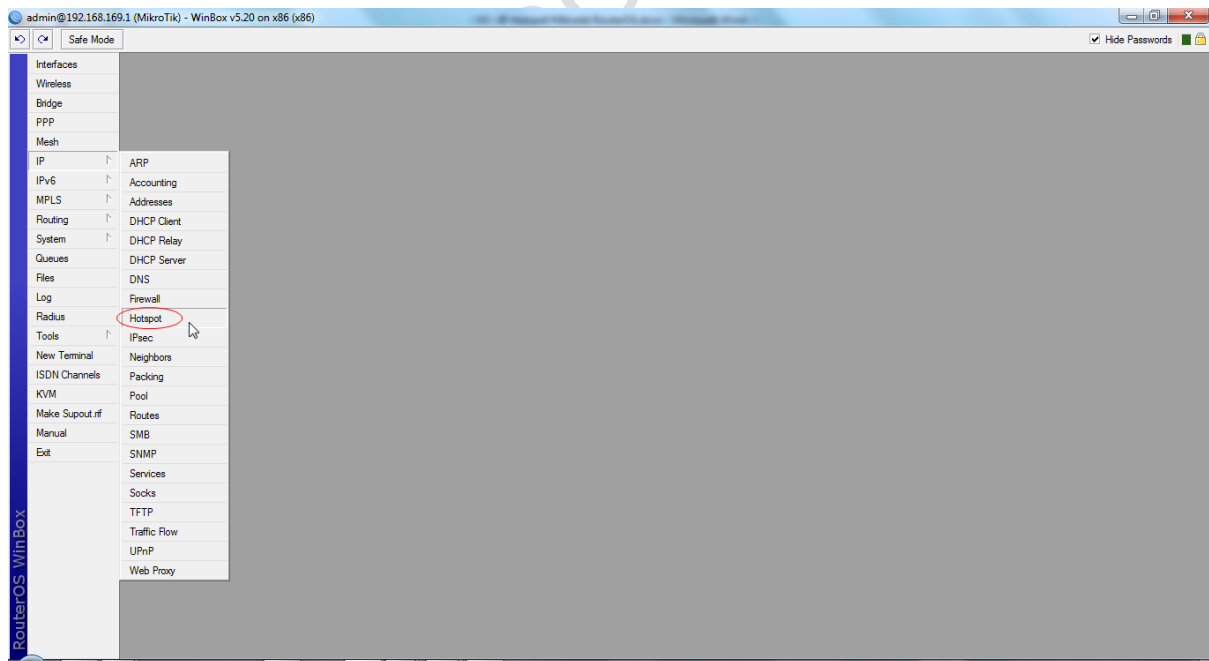
```
[admin@stmikbumigora.intra.net] /ip hotspot user> print
```

Flags: X - disabled, D - dynamic

#	SERVER	PROFILE	UPTIME	NAME
0	default		0s	admin
1	default		0s	putu

C. Manajemen User Hotspot

Manajemen User hotspot dapat dilakukan via **Winbox** dengan mengakses menu **IP → Hotspot**, seperti terlihat pada gambar berikut ini:



Selanjutnya akan tampil kotak dialog **Hotspot**, pilih tab "**Users**", seperti terlihat pada gambar berikut ini:

Hotspot

Server Profiles Users User Profiles Active Hosts IP Bindings Service Ports Walled Garden Walled Garden IP List Cookies ...

+ - ✓ ✕ [icon] [icon] 00 Reset All Counters Find

Server	#	Name	Address	MAC Address	Profile	Uptime	
[icon] all		admin			default	00:00:00	
::: stmik bumigora mataram							
[icon] all		stmik		AA:BB:CC:DD:EE...	default	00:00:00	

2 items

BAB 4

Mikrotik PHP API

Tujuan:

Mahasiswa memahami mengenai Mikrotik PHP API.

Materi:**Pengenalan API**

Application Programming Interface (API) adalah sebuah antarmuka yang diimplementasikan oleh perangkat lunak untuk berinteraksi dengan perangkat lunak lainnya. API memfasilitasi interaksi diantara perangkat lunak yang berbeda, serupa dengan cara sebuah antarmuka (*interface*) pengguna memfasilitasi interaksi antara manusia dengan komputer.

API diimplementasikan oleh aplikasi, library, dan sistem operasi untuk menentukan konvensi pemanggilan dan digunakan untuk mengakses layanannya. API meliputi spesifikasi untuk rutin, struktur data, objek class, dan protokol yang digunakan untuk melakukan komunikasi. Sebuah API merupakan abstraksi yang menggambarkan antarmuka untuk berinteraksi dengan sekumpulan fungsi yang digunakan oleh komponen-komponen dari sistem perangkat lunak.

Mikrotik menyediakan RouterOS API mulai dari RouterOS versi 3. RouterOS API menggunakan port komunikasi **8728** yang **dinonaktifkan secara default**. Informasi service-service apa saja yang aktif pada Mikrotik dapat ditampilkan menggunakan perintah **/ip service print**, seperti berikut:

```
[admin@stmikbumigora.intra.net] > ip service print
```

Flags: X - disabled, I - invalid

#	NAME
PORT ADDRESS CERTIFICATE	
0	telnet
23 0.0.0.0/0	
1	ftp
21 0.0.0.0/0	
2	www
80 0.0.0.0/0	
3	ssh
22 0.0.0.0/0	
4	www-ssl
443 0.0.0.0/0 none X	
5	api
8728 0.0.0.0/0	
6	winbox
8291 0.0.0.0/0	

Terlihat bahwa API masih nonaktif. Untuk mengaktifkan API dapat menggunakan perintah **/ip service enable 5** atau menggunakan perintah **/ip service enable api**, seperti berikut:

```
[admin@stmikbumigora.intra.net] > ip service enable 5
```

Untuk memverifikasi apakah API telah berhasil diaktifkan, gunakan perintah **/ip service print**, seperti berikut:

```
[admin@stmikbumigora.intra.net] > ip service print
```

Flags: X - disabled, I - invalid

#	NAME	PORT ADDRESS CERTIFICATE
---	------	--------------------------

0	telnet	23	0.0.0.0/0	
1	ftp	21	0.0.0.0/0	
2	www	80	0.0.0.0/0	
3	ssh	22	0.0.0.0/0	
4	X www-ssl	443	0.0.0.0/0	none
5	api	8728	0.0.0.0/0	
6	winbox	8291	0.0.0.0/0	

Komunikasi dengan router melalui API berlangsung menggunakan **API sentences** yang terdiri dari perintah API (**command**) dan atribut-atribut (**attribute**). *API sentence* dikirim menggunakan bentuk yang sangat spesifik yaitu dapat terdiri dari beberapa baris atau kata-kata, ketika dikirim ke router masing-masing kata harus mempunyai prefix, dan kata terakhir pada *API sentence* harus diterminasi dengan 0 (harus berisi byte yang diatur semuanya dengan 0).

Perintah (Command)

Perintah API adalah perintah yang berasal dari **Command Line Interface (CLI)**. Sintak dari perintah diturunkan dari *CLI* dan menyertakan path *CLI* dan perintah itu sendiri, sebagai contoh:

```
/ip address print
```

Perintah API yang berasal dari perintah *CLI* tersebut adalah:

```
/ip/address/print
```

Pada contoh diatas, **/ip/address** adalah *path* (lokasi) dan **print** adalah perintah itu sendiri. Sebuah perintah harus diawali dengan dengan tanda `'/'`. Spasi pada perintah *CLI* diganti dengan tanda `'/'`.

Atribut (Attribute)

Masing-masing API sentence dapat mempunyai atribut-atribut. Daftar atribut lengkap dapat diperoleh dari CLI dengan menekan tombol **?** atau **double Tab**. Sebagai contoh setelah penekanan tombol **?** pada akhir dari penulisan perintah **/ip address add** untuk mengetahui atribut dari perintah **/ip address add** akan menampilkan atribut-atribut berikut:

```
[admin@MikroTik] > ip address add
```

```
Creates new item with specified property values.
```

```
address -- Local IP address
```

```
broadcast -- Broadcast address
```

```
comment -- Short description of the item
```

```
copy-from -- Item number
```

```
disabled -- Defines whether item is ignored or used
```

```
interface -- Interface name
```

```
netmask -- Network mask
```

```
network -- Network prefix
```

Sedangkan hasil dari penggunaan *double Tab* adalah sebagai berikut:

```
[admin@MikroTik] > ip address add
```

```
broadcast    comment    copy-from    disabled    netmask    network    address  
interface
```

Contoh dari API sentence:

```
/ip/address/add  
=address=192.168.88.1/24  
=interface=ether1
```

Eksekusi perintah ini berfungsi untuk menambahkan alamat IP pada interface **ether1**.

Perintah yang diikuti oleh argumen harus diawali dengan tanda '=' diikuti oleh nama argumen, kemudian diikuti oleh tanda '=' lainnya, dan nilai dari argumen, sebagai contoh **=address=192.168.88.1/24**.

Perintah yang diikuti oleh argumen yang spesifik diawali dengan tanda '.', sebagai contoh **.id**.

Perintah yang memiliki parameter *query* tambahan diawali dengan tanda '?', sebagai contoh:

```
/interface/print
```

?type=ether

Atribut tanpa nilai

Perintah di RouterOS mempunyai atribut-atribut yang tidak mempunyai nilai apapun yang diatur, jika atribut tersebut digunakan ia hanya mengindikasikan bahwa mereka harus digunakan, dan nilai, jika diberikan akan diabaikan. Sebagai contoh:

```
/ip/address/print
```

```
=follow=
```

API PHP Class

Mikrotik RouterOS API diimplementasikan dalam beberapa bahasa pemrograman termasuk PHP. **API PHP Class** dibuat oleh **Denis Basta**, dan dapat diperoleh pada situs Wiki dari Mikrotik dengan alamat http://wiki.mikrotik.com/wiki/API_PHP_class. PHP Class ini bekerja dengan API RouterOS versi 3. API ini dapat digunakan untuk mengambil/membaca dan mengatur konfigurasi dari RouterOS. Berikut ini adalah *Mikrotik API PHP Class*:

```
<?php
//
// RouterOS API class
// Author: Denis Basta
//
// read() function altered by Nick Barnes to take into account the placing
// of the "!done" reply and also correct calculation of the reply length.
///
// read() function altered by Ben Menking (ben@infotechsc.com); removed
// echo statement that dumped byte data to screen
//
```

```

////////////////////
// Revised by: Jeremy Jefferson (http://jeremyj.com)
// January 8, 2010
//
//      Fixed write function in order to allow for queries to be executed
//

class routeros_api {

    var $debug = false;                // Show debug information
    var $error_no;                     // Variable for storing
connection error number, if any
    var $error_str;                   // Variable for storing
connection error text, if any

    var $attempts = 5;                 // Connection attempt count
    var $connected = false;            // Connection state
    var $delay = 3;                    // Delay between connection
attempts in seconds
    var $port = 8728;                  // Port to connect to
    var $timeout = 3;                  // Connection attempt timeout
and data read timeout

    var $socket;                       // Variable for storing
socket resource

    /*****
    *
    *****/

    function debug($text) {

        if ($this->debug)
            echo $text . "\n";

    }

    /*****
    *
    *****/

    function encode_length($length) {

        if ($length < 0x80) {

            $length = chr($length);

        }
        else
            if ($length < 0x4000) {

                $length |= 0x8000;

                $length = chr( ($length >> 8) & 0xFF) . chr($length
& 0xFF);

            }
        else
            if ($length < 0x200000) {

```

```

        $length |= 0xC00000;

        $length = chr( ($length >> 8) & 0xFF) . chr(
($length >> 8) & 0xFF) . chr($length & 0xFF);

    }
    else
    if ($length < 0x10000000) {

        $length |= 0xE0000000;

        $length = chr( ($length >> 8) & 0xFF) . chr(
($length >> 8) & 0xFF) . chr( ($length >> 8) & 0xFF) . chr($length & 0xFF);

    }
    else
    if ($length >= 0x10000000)
        $length = chr(0xF0) . chr( ($length >> 8) & 0xFF) .
chr( ($length >> 8) & 0xFF) . chr( ($length >> 8) & 0xFF) . chr($length &
0xFF);

    return $length;

}

/*****
*
*****/

function connect($ip, $login, $password) {

    for ($ATTEMPT = 1; $ATTEMPT <= $this->attempts; $ATTEMPT++)
    {

        $this->connected = false;

        $this->debug('Connection attempt #' . $ATTEMPT . '
to ' . $ip . ':' . $this->port . '...');

        if ($this->socket = @fsockopen($ip, $this->port,
$this->error_no, $this->error_str, $this->timeout) ) {

            socket_set_timeout($this->socket, $this-
>timeout);

            $this->write('/login');

            $RESPONSE = $this->read(false);

            if ($RESPONSE[0] == '!done') {

                if (preg_match_all('/[^\=]+/i',
$RESPONSE[1], $MATCHES) ) {

                    if ($MATCHES[0][0] == 'ret'

&& strlen($MATCHES[0][1]) == 32) {

                        $this-
>write('/login', false);

```

```

        $this->write('=name='
. $login, false);
        $this-
>write('=response=00' . md5(chr(0) . $password . pack('H*', $MATCHES[0][1])
) );

        $RESPONSE = $this-

>read(false);

        if ($RESPONSE[0] ==

'!done') {

                $this-

                break;

        }

        }

        }

        }

        fclose($this->socket);

    }

    sleep($this->delay);

}

if ($this->connected)
    $this->debug('Connected...');
else
    $this->debug('Error...');

return $this->connected;

}

/*****
*
*****/

function disconnect() {

    fclose($this->socket);

    $this->connected = false;

    $this->debug('Disconnected...');

}

/*****
*
*****/

```

```

function parse_response($response) {

    if (is_array($response) ) {

        $PARSED = array();
        $CURRENT = null;

        foreach ($response as $x) {

            if (in_array($x, array('!fatal', '!re',
'!trap') ) ) {

                if ($x == '!re')
                    $CURRENT = &$PARSED[];
                else
                    $CURRENT = &$PARSED[$x][];

            }
            else
                if ($x != '!done') {

                    if (preg_match_all('/[^\s=]+/i', $x,
$MATCHES) )
                        $CURRENT[$MATCHES[0][0]] =
(isset($MATCHES[0][1]) ? $MATCHES[0][1] : '');

                }

            }

        }

        return $PARSED;

    }
    else
        return array();

}

/*****
*
*****/

function array_change_key_name(&$array) {
    if (is_array($array) ) {
        foreach ($array as $k => $v) {
            $tmp = str_replace("-", "_", $k);
            $tmp = str_replace("/", "_", $tmp);
            if ($tmp) {
                $array_new[$tmp] = $v;
            } else {
                $array_new[$k] = $v;
            }
        }
        return $array_new;
    } else {
        return $array;
    }
}

/*****/

```

```

*
*****/

function parse_response4smarty($response) {
    if (is_array($response) ) {
        $PARSED = array();
        $CURRENT = null;
        foreach ($response as $x) {
            if (in_array($x, array('!fatal', '!re',
'!trap') ) ) {
                if ($x == '!re')
                    $CURRENT = &$PARSED[];
                else
                    $CURRENT = &$PARSED[$x][0];
            }
            else
                if ($x != '!done') {
                    if (preg_match_all('/[^\s]+/i', $x,
$MATCHES) )
                        $CURRENT[$MATCHES[0][0]] =
(isset($MATCHES[0][1]) ? $MATCHES[0][1] : '');
                }
                foreach ($PARSED as $key => $value) {
                    $PARSED[$key] = $this-
>array_change_key_name($value);
                }
                return $PARSED;
            }
            else {
                return array();
            }
        }
    }

    /*****
    *
    *****/

    function read($parse = true) {

        $RESPONSE = array();

        while (true) {

            // Read the first byte of input which gives us some or all of the
length
            // of the remaining reply.
            $BYTE = ord(fread($this->socket, 1) );
            $LENGTH = 0;

            // If the first bit is set then we need to remove the first four
bits, shift left 8
            // and then read another byte in.
            // We repeat this for the second and third bits.
            // If the fourth bit is set, we need to remove anything left in
the first byte
            // and then read in yet another byte.
            if ($BYTE & 128) {
                if (($BYTE & 192) == 128) {

```

```

        $LENGTH = (($BYTE & 63) << 8 ) + ord(fread($this->socket,
1)) ;
    } else {
        if (($BYTE & 224) == 192) {
            $LENGTH = (($BYTE & 31) << 8 ) + ord(fread($this->socket,
1)) ;
            $LENGTH = ($LENGTH << 8 ) + ord(fread($this->socket,
1)) ;
        } else {
            if (($BYTE & 240) == 224) {
                $LENGTH = (($BYTE & 15) << 8 ) + ord(fread($this-
>socket, 1)) ;
                $LENGTH = ($LENGTH << 8 ) + ord(fread($this->socket,
1)) ;
                $LENGTH = ($LENGTH << 8 ) + ord(fread($this->socket,
1)) ;
            } else {
                $LENGTH = ord(fread($this->socket, 1)) ;
                $LENGTH = ($LENGTH << 8 ) + ord(fread($this->socket,
1)) ;
                $LENGTH = ($LENGTH << 8 ) + ord(fread($this->socket,
1)) ;
                $LENGTH = ($LENGTH << 8 ) + ord(fread($this->socket,
1)) ;
            }
        }
    }
} else {
    $LENGTH = $BYTE;
}

// If we have got more characters to read, read them in.
if ($LENGTH > 0) {
    $_ = "";
    $retlen=0;
    while ($retlen < $LENGTH) {
        $toread = $LENGTH - $retlen ;
        $_ .= fread($this->socket, $toread);
        $retlen = strlen($_);
    }
    $RESPONSE[] = $_ ;
    $this->debug('>>> [' . $retlen . '/' . $LENGTH . ' bytes
read.']);
}

// If we get a !done, make a note of it.
if ($_ == "!done")
    $receiveddone=true;

$STATUS = socket_get_status($this->socket);

if ($LENGTH > 0)
    $this->debug('>>> [' . $LENGTH . ', ' . $STATUS['unread_bytes']
. ']' . ' . $_);

if ( (!$this->connected && !$STATUS['unread_bytes']) ||
    ($this->connected && !$STATUS['unread_bytes'] && $receiveddone)
)
    break;

```

```

    }

    if ($parse)
        $RESPONSE = $this->parse_response($RESPONSE);

    return $RESPONSE;
}

/*****
 *
 *****/

function write($command, $param2 = true) {

    if ($command) {

        $data = explode("\n", $command);

        foreach ($data as $com) {
            $com = trim($com);
            fwrite($this->socket, $this-
>encode_length(strlen($com) ) . $com);
            $this->debug('<<< [' . strlen($com) . ']' . ' ' .
$com);
        }

        if (gettype($param2) == 'integer') {

            fwrite($this->socket, $this-
>encode_length(strlen('.tag=' . $param2) ) . '.tag=' . $param2 . chr(0) );

            $this->debug('<<< [' . strlen('.tag=' .
$param2) . ']' . '.tag=' . $param2);
        }
        else
            if (gettype($param2) == 'boolean')
                fwrite($this->socket, ($param2 ? chr(0) :
'' ) );

        return true;
    }
    else
        return false;
}

}
?>

```

Penggunaan Mikrotik PHP API

Terdapat 6 tahapan yang dilalui untuk melakukan manajemen Mikrotik RouterOS melalui PHP yaitu:

1. Menyisipkan file **class routeros_api** pada file PHP menggunakan function "**include()**", seperti berikut:

```
<?php
    include('routeros_api.class.php');
?>
```

2. Membuat **instance (object)** dari **class routeros_api**.

Untuk membuat instance dari class, digunakan perintah berikut:

```
<?php
    $API = new routeros_api();
?>
```

Variable **\$API** digunakan untuk menampung hasil dari pembuatan instance (object). Melalui variable ini dapat diakses method-method yang dimiliki oleh class routeros_api.

3. Koneksi ke Mikrotik RouterOS.

Untuk dapat terkoneksi ke Mikrotik RouterOS, diperlukan informasi alamat ip atau nama host dari router, nama user, dan sandi yang dapat digunakan untuk melewati proses otentikasi. Secara default telah dibuatkan user "admin", dengan password **blank**. Method "**connect**" dengan parameter hostname, username, dan password, dapat digunakan untuk terkoneksi ke router, seperti berikut:

```
<?php
    $hostname = '192.168.1.1';
    $username = 'admin';
    $password = '';
    $API->connect($hostname, $username, $password);
?>
```

4. Mengeksekusi perintah Mikrotik *Command Line Interface (CLI)*.

Untuk mengeksekusi perintah, digunakan *method* “**write**”, seperti berikut:

```
<?php
    $API->write('perintah_mikrotik');
?>
```

Method *write* ini memerlukan parameter perintah Mikrotik yang akan dieksekusi, sebagai contoh untuk menampilkan informasi seluruh data user hotspot, perintah yang digunakan adalah **/ip/hotspot/user/getall**, seperti berikut:

```
<?php
    $API->write('/ip/hotspot/user/getall');
?>
```

5. Membaca hasil dari eksekusi perintah Mikrotik.

Untuk membaca hasil dari eksekusi perintah, digunakan *method* “**read**”, seperti berikut:

```
<?php
    $API->read();
?>
```

6. Memutuskan koneksi dari Mikrotik RouterOS.

Untuk memutuskan koneksi, digunakan *method* "**disconnect**", seperti berikut:

```
<?php

    $API->disconnect();

?>
```

Berikut adalah contoh lengkap tahapan-tahapan mengakses Mikrotik menggunakan *PHP API Class*:

```
<?php
    // Menyisipkan Mikrotik RouterOS API
    include('routeros_api.class.php');

    /*
     * Sebelum dapat melakukan aktivitas manajemen router
     Mikrotik, pengguna harus melalui proses otentikasi login
     terlebih dahulu.
     * Untuk kebutuhan otentikasi login ke router Mikrotik
     diperlukan informasi alamat IP dari Mikrotik, username &
     password.
     */

    // Variable hostname digunakan untuk menampung alamat IP
    dari router Mikrotik
    $hostname = '10.0.0.1';
    // Variable username digunakan untuk menampung nama
    pengguna yang digunakan sebagai user saat login ke Mikrotik
    $username = 'admin';
    // Variable password digunakan untuk menampung sandi
    pengguna yang digunakan sebagai password saat login ke
    Mikrotik
    $password = 'admin';

    // Membuat instance (objek) dari class routeros_api
    $API = new routeros_api();

    // Membuat koneksi ke router Mikrotik
    if ($API->connect($hostname, $username, $password))
    {
        // Memanggil method "write" dari objek yang
        digunakan untuk mengeksekusi perintah2 dari router Mikrotik
```

```

        // Parameter "/user/getall" pada method "write"
berfungsi untuk mengambil daftar user yang terdapat di router
Mikrotik
        $API->write('/user/getall');
        /* Memanggil method "read" dari objek yang
digunakan untuk membaca data hasil dari eksekusi method
"write"
        * dari objek pada langkah sebelumnya.
        * Hasil dari pemanggilan method "read" dari
objek ditampung pada variable "$users"
        */
        $users = $API->read();
        // Memutuskan koneksi ke router Mikrotik
        $API->disconnect();

        // Membuat tabel
        echo "<table border='1' cellpadding='2' cellspacing='2'>";
        // Membuat baris (row) sebagai header (judul kolom) dari
tabel
        echo "<tr>";

        // Membuat perulangan untuk membaca isi variable "$users"
yang menampung nama dari kolom
        foreach ($users[0] as $key => $i) {
            echo "<td bgcolor='grey' align='center'><b>" .
strtoupper($key) . "</b></td>";
        }
        // Membuat kolom dengan nama AKSI
        echo "<td bgcolor='grey' align='center'><b>AKSI</b></td>";
        echo "</tr>";

        // Membuat perulangan untuk menampilkan data yang
ditampung pada variable "users"
        foreach ($users as $data)
        {
            // Membuat baris (row) dari tabel
            echo "<tr>";
            foreach ($data as $key => $i)
            {

                echo "<td bgcolor='white'>$i</td>";

            }
            // Membuat kolom
            echo "<td bgcolor='white'>";
            // Membuat hyperlink HAPUS untuk menghapus (remove)
user
            echo "<a href='hapus_user.php?id=". $data['.id']
.'">Hapus</a>";
            echo "</td>";

```

```

        echo "</tr>";
    }
    echo "</table>";
    }
    else
    {
        return ("Gagal melakukan koneksi ke router
Mikrotik!");
    }
    echo "<a href='tambah_user.php'>Tambah User</a>";
?>

```

Deskripsi Perintah

- Perintah "**getall**" tersedia di API sebagai alias dari perintah "**print**" pada console.
- Hasil dari *getall* akan mengandung item **=.id=** yang merupakan property internal.
- Perintah "**print**" **berbeda dari console**, dimana digunakan proses mengambil data tertentu/spesifik. Perintah ini digunakan bersama argumen "**.proplist**" untuk menandai item-item apa saja yang diambil, sebagai contoh:

```

<?php
    $API->write('/ip/hotspot/user/print', false);
    $API->write('=.proplist=name', false);
    $API->write('=.proplist=password', false);
    $API->write('=.proplist=mac-address', false);
    $API->write('=.proplist=comment', false);
    $API->write('?.id='.$id);
?>

```

Contoh diatas digunakan untuk mengambil data user hotspot tertentu berdasarkan **.id** yang dikirim sebagai paramater. Item/atribut-atribut

yang akan diambil dicantumkan pada argumen **".proplist"**. Terlihat terdapat 4 atribut yang diambil dari data hotspot user sebagai hasil eksekusi perintah **/ip/hotspot/user/print**, yaitu **name**, **password**, **mac-address**, dan **comment**. Ketika terdapat lebih dari satu atribut yang akan diambil, setelah paramater **"=.proplist="** dari atribut sebelumnya, terdapat parameter kedua yang harus dicantumkan yaitu berupa nilai **"false"**, untuk menyatakan masih terdapat atribut lainnya yang akan diambil, sebaliknya jika ini adalah atribut terakhir maka pencantuman parameter kedua dengan nilai **"false"** tidak diperlukan.

Melakukan Query

Perintah **"print"** dapat menerima query untuk membatasi sekumpulan item-item yang ditampilkan.

- Query diawali dengan tanda **'?'**.
- Urutan dari query sangat signifikan. Query dievaluasi mulai dari yang paling awal.
- Query dievaluasi untuk masing-masing item pada daftar. Jika query sukses, maka item akan diproses, sebaliknya jika query gagal, maka item tersebut akan diabaikan.
- Query dievaluasi menggunakan tumpukan nilai boolean. Tumpukan dapat mengandung nilai **'true'** hingga tak terhingga. Pada akhir dari evaluasi, jika tumpukan mengandung paling sedikit sebuah nilai **'false'**, query akan gagal.
- Query beroperasi mengikuti aturan berikut:

?name : query ini akan menghasilkan hasil **'true'** jika item memiliki nilai dari property name, **'false'** jika tidak.

Berikut adalah contoh penggunaan query:

```
<?php
    $API->write('?id='.$id);
?>
```

BAB 5**Mikrotik PHP API PADA SYSTEM USER****Tujuan:**

Mahasiswa memahami mengenai Mikrotik PHP API untuk manajemen user sistem Mikrotik.

Materi:**A. Pengenalan System User**

Mikrotik mengatur user/pengguna yang terhubung ke router dari console, serial, telnet, SSH atau winbox. User dapat diotentikasi menggunakan user yang tersimpan di lokal atau di server RADIUS. Masing-masing user, masuk sebagai anggota dari group, yang menentukan hak akses dari user tersebut. User default adalah *admin*, sebagai anggota dari group "**full**". Terdapat 3 group default bawaan dari Mikrotik yaitu:

- a) Read
- b) Write
- c) Full

B. Perintah-perintah Mikrotik CLI untuk manajemen user sistem Mikrotik.**1. Membuat user sistem baru**

```
[admin@MikroTik] > user add name=stmik group=full  
address=192.168.88.2/32 password=bumigora mac-  
comment="stmik bumigora mataram"
```

2. Mengubah user sistem

```
[admin@MikroTik] > user edit 1  
value-name: comment
```

3. Menghapus user sistem

```
[admin@MikroTik] > user remove 1
```

4. Menampilkan seluruh data user sistem

```
[admin@MikroTik] > user print
```

5. Menonaktifkan user sistem tertentu

```
[admin@MikroTik] > user disable 1
```

6. Mengaktifkan user sistem tertentu

```
[admin@MikroTik] > user enable 1
```

C. Perintah-perintah Mikrotik PHP API untuk manajemen user sistem.

1. Membuat user sistem baru

```
/user/add
=name=?
=group=?
=address=?
=comment=?
=password=?
=disabled=?
```

2. Mengubah user sistem

```
/user/set
=.id=?
=name=?
=group=?
=address=?
=comment=?
=password=?
=disabled=?
```

3. Menghapus user sistem

```
/user/remove
=.id=?
```

4. Menampilkan seluruh data user sistem

```
/user/getall
```

5. Menonaktifkan user sistem tertentu

```
/user/disable
=.id=?
```

6. Mengaktifkan user sistem tertentu

```
/user/enable
=.id=?
```

D. Studi Kasus aplikasi manajemen user sistem Mikrotik

1. Buat file dengan nama "**config.php**" yang digunakan untuk melakukan koneksi dari PHP ke router Mikrotik, dengan kode program sebagai berikut:

```
<?php
// menyisipkan file class mikrotik php api
include("routeros_api.class.php");

// membuat instance atau object dari class
$API = new routeros_api();

// mendeklarasikan variable untuk koneksi ke mikrotik
$mikrotik_hostname = "192.168.88.1";
$mikrotik_username = "admin";
$mikrotik_password = "";

// membuat & mengecek koneksi ke router mikrotik
if (! $API->connect($mikrotik_hostname,
$mikrotik_username, $mikrotik_password))
{
    die("Koneksi ke Mikrotik Gagal dilakukan!<br>");
}
?>
```

2. Buat file dengan nama "**index.php**" yang digunakan untuk menampilkan informasi user sistem yang terdapat di router mikrotik, dengan kode program sebagai berikut:

```
<html>
<head>
<title>Mikrotik RouterOS System User</title>
</head>
<body>
<?php
// menyisipkan file config.php
include("config.php");

// mengeksekusi perintah Mikrotik CLI
$API->write("/user/getall");

// membaca hasil eksekusi perintah tersebut
$users = $API->read();

// menampilkan isi variable users
echo "<h3>Daftar User Sistem Mikrotik RouterOS</h3>";
echo "<form name=\"form\">";
echo "<table width=\"780px\" border=\"0\"
cellpadding=\"2\" cellspacing=\"2\">";
echo "<tr bgcolor=\"#0066CC\" height=\"15px\">";
echo "<th>NO.</th>";
echo "<th>.ID</th>";
echo "<th>NAME</th>";
echo "<th>GROUP</th>";
echo "<th>ADDRESS</th>";
echo "<th>DISABLED</th>";
```

```

echo "<th>COMMENT</th>";
echo "<th>ACTION</th>";
echo "</tr>";
$i = 1;
foreach ($users as $data)
{
    echo "<tr bgcolor=\"#CCE6FF\" height=\"15px\">";
    echo "<td align=\"right\">$i.</td>";
    echo "<td>".$data['id'].</td>";
    echo "<td>".$data['name'].</td>";
    echo "<td align=\"center\">".$data['group'].</td>";
    echo "<td>".$data['address'].</td>";
    echo
align="center">".$data['disabled'].</td>";
    echo "<td>".$data['comment'].</td>";

    echo "<td align=\"center\">";
    echo
href="update_user.php?id=".$data['id'].>Update</a> | ";
    echo
href="delete_user.php?id=".$data['id'].>Delete</a> | ";
    // mengecek nilai dari argumen disabled
    if ($data['disabled']=='true')
    {
        echo
href="enable_user.php?id=".$data['id'].>Enable</a>;

    }
    else
    {
        echo
href="disable_user.php?id=".$data['id'].>Disable</a>;

    }
    echo "</td>";
    echo "</tr>";
    $i++;
}
echo "<tr>";
echo "<td colspan=\"8\"><a href=\"add_user.php\">Tambah
User</a></td>";
echo "</tr>";
echo "</table>";
echo "</form>";
// memutuskan koneksi dari router mikrotik
$API->disconnect();
?>
</body>
</html>

```

3. Buat file dengan nama **"add_user.php"** yang digunakan untuk memasukkan data user sistem baru, dengan kode program sebagai berikut:

```

<html>
<head>
<title>Mikrotik RouterOS - Add New User</title>

```

```

<style type="text/css">
<!--
.style1 {color: #FF0000}
-->
</style>
</head>

<body>
<?php
    if (isset($_POST['btnSubmit']))
    {
        if (empty($_POST['txtname']))
        {
            $error[] = 'Name is required!';
        }

        if (empty($_POST['lstgroup']))
        {
            $error[] = 'Group is required!';
        }

        if (empty($_POST['txtcomment']))
        {
            $error[] = 'Comment is required!';
        }

        if (empty($_POST['txtpassword']))
        {
            $error[] = 'Password is required!';
        }

        if (empty($_POST['txtretype_password']))
        {
            $error[] = 'Retype Password is required!';
        }

        if ($_POST['txtpassword'] !=
$_POST['txtretype_password'])
        {
            $error[] = 'The passwords you entered do not
match. Please try again!';
        }

        if (empty($_POST['rbstatus']))
        {
            $error[] = 'Status is required!';
        }

        if (count($error) > 0)
        {
            echo "<ul>";
            foreach ($error as $data)
            {
                echo "<li>$data</li>";
            }
        }
    }
}

```

```

        echo "</ul>";
    }
    else
    {
        echo      "<strong>Name      :</strong>"      ".
$_POST['txtname'] . "<br />";
        echo      "<strong>Group      :</strong>"      ".
$_POST['lstgroup'] . "<br />";
        echo      "<strong>Allowed Address :</strong>"  ".
$_POST['txtallowed_address'] . "<br />";
        echo      "<strong>Comment      :</strong>"      ".
$_POST['txtcomment'] . "<br />";
        echo      "<strong>Password      :</strong>"      ".
$_POST['txtpassword'] . "<br />";
        echo      "<strong>Status      :</strong>"      ".
$_POST['rbstatus'] . "<br />";

        // mengambil nilai isian dari form untuk
pembuatan user
        $name = $_POST['txtname'];
        $group = $_POST['lstgroup'];
        $address = $_POST['txtallowed_address'];
        $comment = $_POST['txtcomment'];
        $password = $_POST['txtpassword'];
        $status = $_POST['txtstatus'];

        // mengatur nilai argumen disabled berdasarkan
isian status
        if ($status == 'Enable')
        {
            $status = 'false';
        }
        else
        {
            $status = 'true';
        }

        // menyisipkan file config.php
include("config.php");

        // mengeksekusi perintah Mikrotik CLI
$API->write("/user/add", false);
$API->write("=name=$name", false);
$API->write("=group=$group", false);
$API->write("=address=$address", false);
$API->write("=comment=$comment", false);
$API->write("=password=$password", false);
$API->write("=disabled=$status");

        // membaca hasil eksekusi perintah tersebut
$API->read();

        echo "Data user berhasil ditambahkan!<br>";

        // memutuskan koneksi dari router mikrotik

```

```

$API->disconnect();
    }
}
?>
<form action="add_user.php" method="post" name="frmuser"
id="frmuser">
    <table width="600" border="0" cellspacing="2"
cellpadding="2">
        <tr bgcolor="#0066CC">
            <th colspan="3" scope="row">Mikrotik RouterOS - New
User</th>
        </tr>
        <tr bgcolor="#CCE6FF">
            <th colspan="3" scope="row"><div align="left">Note:
<span class="style1">*</span> Fields must be filled
</div></th>
        </tr>
        <tr bgcolor="#CCE6FF">
            <th width="161" scope="row">Name<span
class="style1">*</span></th>
            <td width="5"><strong>:</strong></td>
            <td width="414"><input name="txtname" type="text"
id="txtname" value="<?php if (isset($_POST['txtname'])) {
echo $_POST['txtname']; } ?>" size="40" maxlength="30"></td>
        </tr>
        <tr bgcolor="#CCE6FF">
            <th scope="row">Group<span
class="style1">*</span></th>
            <td><strong>:</strong></td>
            <td><select name="lstgroup" id="lstgroup">
                <option value="">Choose...</option>
                <option value="read" <?php if
(isset($_POST['lstgroup']) && $_POST['lstgroup'] == 'read')
{ echo "selected"; } ?>>Read</option>
                <option value="write" <?php if
(isset($_POST['lstgroup']) && $_POST['lstgroup'] == 'write')
{ echo "selected"; } ?>>Write</option>
                <option value="full" <?php if
(isset($_POST['lstgroup']) && $_POST['lstgroup'] == 'full')
{ echo "selected"; } ?>>Full</option>
            </select></td>
        </tr>
        <tr bgcolor="#CCE6FF">
            <th scope="row">Allowed Address </th>
            <td><strong>:</strong></td>
            <td><input name="txtallowed_address" type="text"
id="txtallowed_address" value="<?php if
(isset($_POST['txtallowed_address'])) { echo
$_POST['txtallowed_address']; } ?>" size="15"
maxlength="15"></td>
        </tr>
        <tr bgcolor="#CCE6FF">
            <th scope="row">Comment<span
class="style1">*</span></th>
            <td><strong>:</strong></td>

```

```

        <td><input          name="txtcomment"          type="text"
id="txtcomment"          value="<?php
(isset($_POST['txtcomment'])) { echo $_POST['txtcomment']; }
?>" size="40" maxlength="255"></td>
</tr>
<tr bgcolor="#CCE6FF">
    <th                                scope="row">Password<span
class="style1">*</span></th>
    <td><strong>:</strong></td>
    <td><input          name="txtpassword"          type="password"
id="txtpassword" size="25" maxlength="20"></td>
</tr>
<tr bgcolor="#CCE6FF">
    <th                                scope="row">Re-type Password <span
class="style1">*</span> </th>
    <td><strong>:</strong></td>
    <td><input name="txtretype_password" type="password"
id="txtretype_password" size="25" maxlength="20"></td>
</tr>
<tr bgcolor="#CCE6FF">
    <th                                scope="row">Status<span
class="style1">*</span></th>
    <td><strong>:</strong></td>
    <td><input name="rbstatus" type="radio" value="enable"
<?php if (isset($_POST['rbstatus']) && $_POST['rbstatus'] ==
'enable') { echo "checked"; } ?>>
        Enable
        <input name="rbstatus" type="radio" value="disable"
<?php if (isset($_POST['rbstatus']) && $_POST['rbstatus'] ==
'disable') { echo "checked"; } ?>>
        Disable</td>
</tr>
<tr bgcolor="#CCE6FF">
    <th scope="row" colspan="2">&nbsp;</th>
    <td><input          name="btnSubmit"          type="submit"
id="btnSubmit" value="Submit">
    <input name="btnReset" type="reset" id="btnReset"
value="Reset"></td>
</tr>
</table>
</form>
<a href="index.php">Tampil Data User</a>
</body>
</html>

```

4. Buat file dengan nama "edit_ip_hotspot_user.php" yang digunakan untuk mengubah data user hotspot tertentu, dengan kode program sebagai berikut:

```

<?php
    ob_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

```

```

<meta      http-equiv="Content-Type"      content="text/html;
charset=utf-8" />
<title>Edit User Hotspot</title>
</head>

<body>
<?php
    include 'mikrotik.php';

    if      (isset($_POST['btnsubmit']))      &&
$_POST['btnsubmit']=='Submit')
    {
        if (empty($_POST['txtnama']))
        {
            echo "Nama harus diisi!";
        }
        elseif (empty($_POST['txtsandi']))
        {
            echo "Sandi harus diisi!";
        }
        elseif (empty($_POST['txtulangisandi']))
        {
            echo "Ulangi Sandi harus diisi!";
        }
        elseif      ($_POST['txtsandi']      !=
$_POST['txtulangisandi'])
        {
            echo "Sandi dan Ulangi Sandi harus sama!";
        }
        elseif (empty($_POST['txtalamat_mac']))
        {
            echo "Alamat MAC harus diisi!";
        }
        else
        {
            $id = $_POST['id'];
            $nama = $_POST['txtnama'];
            $sandi = $_POST['txtsandi'];
            $alamat_mac = $_POST['txtalamat_mac'];
            $komentar = $_POST['txtkomentar'];

            if (edit_ip_hotspot_user($id, $nama, $sandi,
$alamat_mac, $komentar))
            {
                header('location: ip_hotspot_user.php');
            }
            else
            {
                echo "Perubahan data user hotspot gagal
dilakukan!";
            }
        }
    }
}

```

```

    }

    }

}

// Mengambil data user hotspot tertentu dan
menampilkannya di form
if (isset($_GET['id']) && !empty($_GET['id']))
{
    // Mengambil nilai parameter querystring dengan nama
    id
    $id = $_GET['id'];

    // Memanggil function untuk mengambil data user
    hotspot tertentu
    $hotspot_users = detail_ip_hotspot_user($id);
    foreach ($hotspot_users as $key => $data)
    {
        $id = $data['.id'];
        $nama = $data['name'];
        $sandi = $data['password'];
        $ulangisandi = $data['password'];
        $alamat_mac = $data['mac-address'];
        $komentar = $data['comment'];
    }
}

?>
<form id="frmAddHotspotUser" name="frmAddHotspotUser"
method="post" action="edit_ip_hotspot_user.php">
    <table width="36%" border="0">
        <tr>
            <th width="36%" scope="row">Name</th>
            <td width="2%">:</td>
            <td width="62%"><label>
                <input type="text" name="txtnama" id="txtnama"
value="<?php if (!empty($nama)) { echo $nama; } else { echo
$_POST['txtnama']; }?>" />
            </label><input type="hidden" name="id" value="<?php
echo $id; ?>";</td>
        </tr>
        <tr>
            <th scope="row">Password</th>
            <td>:</td>
            <td><input type="password" name="txtsandi"
id="txtsandi" value="<?php if (!empty($sandi)) { echo
$sandi; } else { echo $_POST['txtsandi']; }?>" /></td>
        </tr>
        <tr>
            <th scope="row">Retype Password</th>
            <td>:</td>
            <td><input type="password" name="txtulangisandi"
id="txtulangisandi" value="<?php if (!empty($ulangisandi)) {
echo $ulangisandi; } else { echo $_POST['txtulangisandi'];
}?>" /></td>
        </tr>
    </table>

```

```

<tr>
  <th scope="row">MAC Address</th>
  <td>:</td>
  <td><input type="text" name="txtalamat_mac"
id="txtalamat_mac" value="<?php if (!empty($alamat_mac)) {
echo $alamat_mac; } else { echo $_POST['txtalamat_mac'];
}?>" /></td>
</tr>
<tr>
  <th scope="row">Comment</th>
  <td>:</td>
  <td><input type="text" name="txtkomentar"
id="txtkomentar" value="<?php if (!empty($komentar)) { echo
$komentar; } else { echo $_POST['txtkomentar']; }?>" /></td>
</tr>
<tr>
  <th scope="row">&nbsp;</th>
  <td>&nbsp;</td>
  <td><label>
    <input type="submit" name="btnsubmit" id="btnsubmit"
value="Submit" />
    <input type="reset" name="btnreset" id="btnreset"
value="Reset" />
  </label></td>
</tr>
</table>
</form>
<p><a href="ip_hotspot_user.php">Tampil Data User
Hotspot</a></p>
</body>
</html>

```

5. Buat file dengan nama **"update_user.php"** yang digunakan untuk mengubah data user sistem tertentu, dengan kode program sebagai berikut:

```

<html>
<head>
<title>Mikrotik RouterOS - Update User</title>
<style type="text/css">
<!--
.style1 {color: #FF0000}
-->
</style>
</head>

<body>
<?php
// menyisipkan file config.php
include("config.php");

if (isset($_POST['btnSubmit']))
{
    if (empty($_POST['txtname']))
    {
        $error[] = 'Name is required!';
    }
}

```

```

if (empty($_POST['lstgroup']))
{
    $error[] = 'Group is required!';
}

if (empty($_POST['txtcomment']))
{
    $error[] = 'Comment is required!';
}

if (empty($_POST['txtpassword']))
{
    $error[] = 'Password is required!';
}

if (empty($_POST['txtretype_password']))
{
    $error[] = 'Retype Password is required!';
}

if (($_POST['txtpassword'] !=
$_POST['txtretype_password']))
{
    $error[] = 'The passwords you entered do not
match. Please try again!';
}

if (empty($_POST['rbstatus']))
{
    $error[] = 'Status is required!';
}

if (count($error) > 0)
{
    echo "<ul>";
    foreach ($error as $data)
    {
        echo "<li>$data</li>";
    }
    echo "</ul>";
}
else
{
    echo "<strong>Name :</strong> ".
$_POST['txtname'] . "<br />";
    echo "<strong>Group :</strong> ".
$_POST['lstgroup'] . "<br />";
    echo "<strong>Allowed Address :</strong> ".
$_POST['txtallowed_address'] . "<br />";
    echo "<strong>Comment :</strong> ".
$_POST['txtcomment'] . "<br />";
    echo "<strong>Password :</strong> ".
$_POST['txtpassword'] . "<br />";
}

```

```

        echo      "<strong>Status      :</strong>      ".
$_POST['rbstatus'] . "<br />";

        // mengambil nilai isian dari form untuk
pembuatan user
        $id = $_POST['id'];
        $name = $_POST['txtname'];
        $group = $_POST['lstgroup'];
        $address = $_POST['txtallowed_address'];
        $comment = $_POST['txtcomment'];
        $password = $_POST['txtpassword'];
        $status = $_POST['rbstatus'];

        // mengatur nilai argumen disabled berdasarkan
isian status
        if ($status == 'enable')
        {
            $status = 'false';
        }
        else
        {
            $status = 'true';
        }

        // mengeksekusi perintah Mikrotik CLI
        $API->write("/user/set", false);
        $API->write("=.id=$id", false);
        $API->write("=.name=$name", false);
        $API->write("=.group=$group", false);
        $API->write("=.address=$address", false);
        $API->write("=.comment=$comment", false);
        $API->write("=.password=$password", false);
        $API->write("=.disabled=$status");

        // membaca hasil eksekusi perintah tersebut
        $API->read();

        echo "Data user berhasil diubah!<br>";

        // memutuskan koneksi dari router mikrotik
        $API->disconnect();

    }
    else
    {
        // mengambil data user dari router mikrotik
        $id = $_GET['id'];

        // mengeksekusi perintah Mikrotik CLI
        $API->write("/user/print", false);
        $API->write("=.proplist=id", false);
        $API->write("=.proplist=name", false);
        $API->write("=.proplist=group", false);

```

```

$API->write("=.proplist=address", false);
$API->write("=.proplist=comment", false);
$API->write("=.proplist=password", false);
$API->write("=.proplist=disabled", false);
$API->write("?.id=$id");

// membaca hasil dari eksekusi perintah diatas
$users = $API->read();

// menampung isian masing-masing argumen user pada
variable tertentu
foreach ($users as $data)
{
    $name = $data['name'];
    $group = $data['group'];
    $address = $data['address'];
    $comment = $data['comment'];
    $status = $data['disabled'];

    if ($status == 'true')
    {
        $status = 'disable';
    }
    else
    {
        $status = 'enable';
    }
}
// memutuskan koneksi dari router mikrotik
$API->disconnect();
?>
<form action="update_user.php" method="post" name="frmuser"
id="frmuser">
    <table width="600" border="0" cellspacing="2"
cellpadding="2">
        <tr bgcolor="#0066CC">
            <th colspan="3" scope="row">Mikrotik RouterOS - Update
User</th>
        </tr>
        <tr bgcolor="#CCE6FF">
            <th colspan="3" scope="row"><div align="left">Note:
<span class="style1">*</span> Fields must be filled
</div></th>
        </tr>
        <tr bgcolor="#CCE6FF">
            <th width="161" scope="row">Name<span
class="style1">*</span></th>
            <td width="5"><strong>:</strong></td>
            <td width="414"><input name="txtname" type="text"
id="txtname" value="<?php if (isset($_POST['txtname'])) {
echo $_POST['txtname']; } else { echo $name; } ?>" size="40"
maxlength="30"></td>
        </tr>
        <tr bgcolor="#CCE6FF">

```

```

        <th                                     scope="row">Group<span
class="style1">*</span></th>
        <td><strong>:</strong></td>
        <td><select name="lstgroup" id="lstgroup">
            <option value="">Choose...</option>
            <option          value="read"          <?php          if
((isset($_POST['lstgroup'])) && $_POST['lstgroup'] == 'read')
|| (isset($group) && $group=='read')) { echo "selected"; }
?>>Read</option>
            <option          value="write"          <?php          if
((isset($_POST['lstgroup'])) && $_POST['lstgroup'] ==
'write') || (isset($group) && $group=='write')) { echo
"selected"; } ?>>Write</option>
            <option          value="full"          <?php          if
((isset($_POST['lstgroup'])) && $_POST['lstgroup'] == 'full')
|| (isset($group) && $group=='full')) { echo "selected"; }
?>>Full</option>
        </select></td>
    </tr>
    <tr bgcolor="#CCE6FF">
        <th scope="row">Allowed Address </th>
        <td><strong>:</strong></td>
        <td><input          name="txtallowed_address"          type="text"
id="txtallowed_address"          value="<?php          if
(isset($_POST['txtallowed_address']))          {          echo
$_POST['txtallowed_address']; } else { echo $address; } ?>"
size="15" maxlength="18"></td>
    </tr>
    <tr bgcolor="#CCE6FF">
        <th                                     scope="row">Comment<span
class="style1">*</span></th>
        <td><strong>:</strong></td>
        <td><input          name="txtcomment"          type="text"
id="txtcomment"          value="<?php          if
(isset($_POST['txtcomment'])) { echo $_POST['txtcomment']; }
else { echo $comment; } ?>" size="40" maxlength="255"></td>
    </tr>
    <tr bgcolor="#CCE6FF">
        <th                                     scope="row">Password<span
class="style1">*</span></th>
        <td><strong>:</strong></td>
        <td><input          name="txtpassword"          type="password"
id="txtpassword" size="25" maxlength="20"></td>
    </tr>
    <tr bgcolor="#CCE6FF">
        <th scope="row">Re-type          Password          <span
class="style1">*</span> </th>
        <td><strong>:</strong></td>
        <td><input          name="txtretype_password"          type="password"
id="txtretype_password" size="25" maxlength="20"></td>
    </tr>
    <tr bgcolor="#CCE6FF">
        <th                                     scope="row">Status<span
class="style1">*</span></th>
        <td><strong>:</strong></td>

```

```

        <td><input name="rbstatus" type="radio" value="enable"
<?php if ((isset($_POST['rbstatus'])) && $_POST['rbstatus']
== 'enable') || (isset($status) && $status == 'enable')) {
echo "checked"; } ?>>
        Enable
        <input name="rbstatus" type="radio" value="disable"
<?php if ((isset($_POST['rbstatus'])) && $_POST['rbstatus']
== 'disable') || (isset($status) && $status == 'disable')) {
echo "checked"; } ?>>
        Disable</td>
    </tr>
    <tr bgcolor="#CCE6FF">
        <th scope="row" colspan="2">&nbsp;  </th>
        <td><input name="btnSubmit" type="submit"
id="btnSubmit" value="Submit">
        <input name="btnReset" type="reset" id="btnReset"
value="Reset">
        <input type="hidden" name="id" value="<?php if
(isset($_POST['id'])) { echo $_POST['id']; } else { echo
$id; } ?>">
        </td>
    </tr>
</table>
</form>
<?php
}
?>
<a href="index.php">Tampil Data User</a>
</body>
</html>

```

6. Buat file dengan nama "delete_user.php" yang digunakan untuk menghapus data user sistem tertentu, dengan kode program sebagai berikut:

```

<?php
// menyisipkan file config.php
include("config.php");

// mengambil nilai id dari querystring
$id = $_GET['id'];

// mengeksekusi perintah Mikrotik CLI
$API->write("/user/remove", false);
$API->write("=.id=$id");

// membaca hasil eksekusi perintah tersebut
$API->read();

echo "User tersebut berhasil dihapus!<br>";
echo "<a href=\"index.php\">Tampil Data User</a>";
?>

```

7. Buat file dengan nama "disable_user.php" yang digunakan untuk menonaktifkan data user sistem tertentu, dengan kode program sebagai berikut:

```

<?php
// menyisipkan file config.php

```

```

include("config.php");

// mengambil nilai id dari querystring
$id = $_GET['id'];

// mengeksekusi perintah Mikrotik CLI
$API->write("/user/disable", false);
$API->write("=.id=$id");

// membaca hasil eksekusi perintah tersebut
$status = $API->read();

echo "User tersebut berhasil dinonaktifkan!<br>";
echo "<a href=\"index.php\">Tampil Data User</a>";
?>

```

8. Buat file dengan nama **"enable_user.php"** yang digunakan untuk menonaktifkan data user sistem tertentu, dengan kode program sebagai berikut:

```

<?php
// menyisipkan file config.php
include("config.php");

// mengambil nilai id dari querystring
$id = $_GET['id'];

// mengeksekusi perintah Mikrotik CLI
$API->write("/user/enable", false);
$API->write("=.id=$id");

// membaca hasil eksekusi perintah tersebut
$API->read();

echo "User tersebut berhasil diaktifkan!<br>";
echo "<a href=\"index.php\">Tampil Data User</a>";
?>

```

BAB 6

Mikrotik PHP API PADA IP HOTSPOT

Tujuan:

Mahasiswa memahami mengenai Mikrotik PHP API untuk manajemen user hotspot.

Materi:

A. Pengenalan IP Hotspot

Menurut manual Mikrotik, IP Hotspot merupakan sebuah fitur yang terdapat pada router Mikrotik yang dapat digunakan sebagai mekanisme untuk mengotentikasi & mengotorisasi pengguna ketika mengakses sumber daya jaringan. IP Hotspot tidak menyediakan fitur enkripsi pada trafik. Untuk melakukan login, pengguna dapat menggunakan browser apapun.

B. Perintah-perintah Mikrotik CLI untuk manajemen user hotspot.

1. Membuat user hotspot baru

```
[admin@MikroTik] > ip hotspot user add name=stmik
password=bumigora mac-address=aa:bb:cc:dd:ee:ff
comment="stmik bumigora mataram"
```

2. Mengubah user hotspot

```
[admin@MikroTik] > ip hotspot user edit 1
value-name: comment
```

3. Menghapus user hotspot

```
[admin@MikroTik] > ip hotspot user remove 1
```

4. Menampilkan seluruh data user hotspot

```
[admin@MikroTik] > ip hotspot user print
```

5. Menonaktifkan user hotspot tertentu

```
[admin@MikroTik] > ip hotspot user disable 1
```

6. Mengaktifkan user hotspot tertentu

```
[admin@MikroTik] > ip hotspot user enable 1
```

C. Perintah-perintah Mikrotik PHP API untuk manajemen user hotspot.

1. Membuat user hotspot baru

```
/ip/hotspot/user/add
=name=?
=password=?
=mac-address=?
```

```
=comment=?
```

2. Mengubah user hotspot

```
/ip/hotspot/user/set
```

```
=.id=?
```

```
=name=?
```

```
=password=?
```

```
=mac-address=?
```

```
=comment=?
```

3. Menghapus user hotspot

```
/ip/hotspot/user/remove
```

```
=.id=?
```

4. Menampilkan seluruh data user hotspot

```
/ip/hotspot/user/getall
```

5. Menonaktifkan user hotspot tertentu

```
/ip/hotspot/user/disable
```

```
=.id=?
```

6. Mengaktifkan user hotspot tertentu

```
/ip/hotspot/user/enable
```

```
=.id=?
```

7. Menampilkan informasi user hotspot tertentu secara detail

```
/ip/hotspot/user/print
```

```
=.proplist=.id
```

```
=.proplist=server
```

```
=.proplist=name
```

```
=.proplist=password
```

```
=.proplist=mac-address
```

```
=.proplist=comment
```

```
=.proplist=profile
```

```
=.proplist=disabled
```

```
? .id=?
```

D. Studi Kasus aplikasi manajemen user hotspot menggunakan Mikrotik

1. Buat file dengan nama "mikrotik.php" yang digunakan untuk melakukan koneksi dari PHP ke router Mikrotik, dan fungsi-fungsi untuk memanajemen user hotspot, dengan kode program sebagai berikut:

```
<?php
/* Copyright 2012 I Putu Hariyadi
   STMIK Bumigora Mataram */
```

```

// Koneksi ke Router Mikrotik
$hostname = '192.168.88.1';
$username = 'admin';
$password = '';

// Menyisipkan Mikrotik RouterOS API
include('routeros_api.class.php');

// Mengambil data user hotspot tertentu
function detail_ip_hotspot_user($id)
{
    // Deklarasi variable dengan scope global
    global $hostname, $username, $password;

    // Membuat instance (objek) dari class routeros_api
    $API = new routeros_api();

    // Membuat koneksi ke router Mikrotik
    if ($API->connect($hostname, $username, $password))
    {
        /*
        * Memanggil method "write" dari objek yang
        digunakan untuk mengeksekusi perintah2 dari router Mikrotik

        * Parameter "/ip/hotspot/user/print" digunakan
        untuk mengambil data salah user hotspot secara spesifik di
        Mikrotik

        * berdasarkan referensi argumen .id. Data yang
        diambil meliputi server, name, password, mac_address,
        comment, profile, dan disabled.
        */
        $API->write('/ip/hotspot/user/print', false);

        $API->write('=.proplist=id', false);
        $API->write('=.proplist=server', false);

        $API->write('=.proplist=name', false);
        $API->write('=.proplist=password', false);
        $API->write('=.proplist=mac-address', false);
        $API->write('=.proplist=comment', false);
        $API->write('=.proplist=profile', false);

        $API->write('=.proplist=disabled', false);
        $API->write('?.id='.$id);
        $hotspot_users = $API->read();

        // Memutuskan koneksi dari router mikrotik
        $API->disconnect();

        return ($hotspot_users);
    }
    else
    {
        return false;
    }
}

```

```

    }
}

// Menampilkan seluruh data user hotspot
function getHotspotUsers()
{
    // Deklarasi variable dengan scope global
    global $hostname, $username, $password;

    // Membuat instance (objek) dari class routeros_api
    $API = new routeros_api();

    // Membuat koneksi ke router Mikrotik
    if ($API->connect($hostname, $username, $password))
    {
        /*
        * Memanggil method "write" dari objek yang
        digunakan untuk mengeksekusi perintah2 dari router Mikrotik

        * Parameter "/ip/hotspot/user/getall"
        digunakan untuk mengambil data seluruh user hotspot di
        Mikrotik

        */
        $API->write('/ip/hotspot/user/getall');
        $hotspot_users = $API->read();

        // Memutuskan koneksi dari router mikrotik
        $API->disconnect();

        return ($hotspot_users);
    }
    else
    {
        return false;
    }
}

// Menambahkan user hotspot
function add_ip_hotspot_user($nama, $sandi, $alamat_mac,
$komentar)
{
    // Deklarasi variable dengan scope global
    global $hostname, $username, $password;

    // Membuat instance (objek) dari class routeros_api
    $API = new routeros_api();

    // Membuat koneksi ke router Mikrotik
    if ($API->connect($hostname, $username, $password))
    {
        /*
        * Memanggil method "write" dari objek yang
        digunakan untuk mengeksekusi perintah2 dari router Mikrotik

```

* Parameter "/ip/hotspot/user/add" digunakan untuk menambahkan user hotspot baru di Mikrotik
 * dengan argumen name, password, mac_address, dan comment.

```

*/
$API->write('/ip/hotspot/user/add', false);
$API->write('=name='.$nama, false);
$API->write('=password='.$sandi, false);
$API->write('=mac-address='.$alamat_mac, false);
$API->write('=comment='.$komentar);
$API->read();

// Memutuskan koneksi dari router mikrotik
$API->disconnect();

return true;
}
else
{
    return false;
}
}

// Mengubah data user hotspot tertentu
function edit_ip_hotspot_user($id, $nama, $sandi,
$alamat_mac, $komentar)
{
    // Deklarasi variable dengan scope global
    global $hostname, $username, $password;

    // Membuat instance (objek) dari class routers_api
    $API = new routers_api();

    // Membuat koneksi ke router Mikrotik
    if ($API->connect($hostname, $username, $password))
    {
        /*
        * Memanggil method "write" dari objek yang
        digunakan untuk mengeksekusi perintah2 dari router Mikrotik

        * Parameter "/ip/hotspot/user/set" digunakan
        untuk mengubah data user hotspot tertentu di Mikrotik
        * berdasarkan referensi argumen .id. Data yang
        diubah meliputi name, password, mac_address, dan comment.
        */
        $API->write('/ip/hotspot/user/set', false);
        $API->write('=.id='.$id, false);
        $API->write('=name='.$nama, false);
        $API->write('=password='.$sandi, false);
        $API->write('=mac-address='.$alamat_mac, false);
        $API->write('=comment='.$komentar);
        $API->read();

        // Memutuskan koneksi dari router mikrotik

```

```

        $API->disconnect();

        return true;
    }
    else
    {
        return false;
    }
}

// Menghapus user hotspot
function remove_ip_hotspot_user($id)
{
    // Deklarasi variable dengan scope global
    global $hostname, $username, $password;

    // Membuat instance (objek) dari class routeros_api
    $API = new routeros_api();

    // Membuat koneksi ke router Mikrotik
    if ($API->connect($hostname, $username, $password))
    {
        // Memanggil method "write" dari objek yang
        digunakan untuk mengeksekusi perintah2 dari router Mikrotik
        // Parameter "/ip/hotspot/user/remove" pada
        method "write" berfungsi untuk menghapus user tertentu
        // yang terdapat di router Mikrotik

        $API->write('/ip/hotspot/user/remove', false);
        $API->write('=.id=.'.$id);
        $API->read();

        // Memutuskan koneksi dari router mikrotik
        $API->disconnect();

        return true;
    }
    else
    {
        return false;
    }
}

// Mengaktifkan user hotspot
function enable_ip_hotspot_user($id)
{
    // Deklarasi variable dengan scope global
    global $hostname, $username, $password;

    // Membuat instance (objek) dari class routeros_api
    $API = new routeros_api();

    // Membuat koneksi ke router Mikrotik
    if ($API->connect($hostname, $username, $password))
    {

```

```

        // Memanggil method "write" dari objek yang
        digunakan untuk mengeksekusi perintah2 dari router Mikrotik
        // Parameter "/ip/hotspot/user/enable" pada
        method "write" berfungsi untuk mengaktifkan user tertentu
        // yang terdapat di router Mikrotik

        $API->write('/ip/hotspot/user/enable', false);
        $API->write('=.id=.'.$id);
        $API->read();

        // Memutuskan koneksi dari router mikrotik
        $API->disconnect();

        return true;
    }
    else
    {
        return false;
    }
}

// Menonaktifkan user hotspot
function disable_ip_hotspot_user($id)
{
    // Deklarasi variable dengan scope global
    global $hostname, $username, $password;

    // Membuat instance (objek) dari class routers_api
    $API = new routers_api();

    // Membuat koneksi ke router Mikrotik
    if ($API->connect($hostname, $username, $password))
    {
        // Memanggil method "write" dari objek yang
        digunakan untuk mengeksekusi perintah2 dari router Mikrotik
        // Parameter "/ip/hotspot/user/disable" pada
        method "write" berfungsi untuk menonaktifkan user tertentu
        // yang terdapat di router Mikrotik

        $API->write('/ip/hotspot/user/disable', false);
        $API->write('=.id=.'.$id);
        $API->read();

        // Memutuskan koneksi dari router mikrotik
        $API->disconnect();

        return true;
    }
    else
    {
        return false;
    }
}
?>

```

2. Buat file dengan nama "ip_hotspot_user.php" yang digunakan untuk menampilkan informasi user hotspot yang terdapat di router mikrotik, dengan kode program sebagai berikut:

```
<?php
// Menyisipkan Mikrotik.php yang menampung function-
function untuk manajemen mikrotik
include('mikrotik.php');

// Memanggil function getHotspotUsers dan menampung hasil
pemanggilan pada variable $hotspot_users
$hotspot_users=getHotspotUsers();

// Membuat tabel
echo "<table border='1' cellpadding='2'
cellspacing='2'>";
// Membuat baris (row) sebagai header (judul kolom) dari
tabel
echo "<tr>";
echo "<th>.id</th>";
echo "<th>Server</th>";
echo "<th>Nama Login</th>";
echo "<th>Sandi Login</th>";
echo "<th>Alamat MAC</th>";
echo "<th>Comment</th>";
echo "<th>Profile</th>";
echo "<th>Disabled</th>";
echo "<th>Aksi</th>";
echo "</tr>";

// Membuat perulangan untuk menampilkan data yang
ditampung pada variable "hotspot_users"
foreach ($hotspot_users as $key => $data)
{
    // Membuat baris (row) dari tabel
    echo "<tr>";
    echo "<td bgcolor='white'>". $data['.id'] . "</td>";
    echo "<td bgcolor='white'>". $data['server']
    . "</td>";
    echo "<td bgcolor='white'>". $data['name'] . "</td>";
    echo "<td bgcolor='white'>". $data['password']
    . "</td>";
    echo "<td bgcolor='white'>". $data['mac-address']
    . "</td>";
    echo "<td bgcolor='white'>". $data['comment']
    . "</td>";
    echo "<td bgcolor='white'>". $data['profile']
    . "</td>";
    echo "<td bgcolor='white'>". $data['disabled']
    . "</td>";
    echo "<td bgcolor='white'>";
    echo "<a href=\"detail_ip_hotspot_user.php?id=".
    $data['.id'] . "\">Detail</a>";
```

```

        echo " | <a href=\"edit_ip_hotspot_user.php?id=".
$data['.id'] ."\">Ubah</a>";
        echo " | <a href=\"remove_ip_hotspot_user.php?id=".
$data['.id'] ."\">Hapus</a>";

        // Mengatur penampilan link Disable dan Enable
        // Jika status disabled user hotspot adalah false,
        maka tampilkan link disable
        // sebaliknya tampilkan link enable
        if ($data['disabled'] == 'true')
        {
            echo " | <a
href=\"enable_ip_hotspot_user.php?id=". $data['.id']
.\">Enable</a></td>";
        }
        else
        {
            echo " | <a
href=\"disable_ip_hotspot_user.php?id=". $data['.id']
.\">Disable</a></td>";
        }
        echo "</tr>";
    }
    echo "</table>";
    echo "<a href='add_ip_hotspot_user.php'>Tambah IP Hotspot
User</a>";
?>

```

3. Buat file dengan nama "add_ip_hotspot_user.php" yang digunakan untuk memasukkan data user hotspot baru, dengan kode program sebagai berikut:

```

<?php
    ob_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Add User Hotspot</title>
</head>

<body>
<?php
    if (isset($_POST['btnsubmit'])) &&
$_POST['btnsubmit']=='Submit')
    {
        if (empty($_POST['txtnama']))
        {
            echo "Nama harus diisi!";
        }
        elseif (empty($_POST['txtsandi']))
        {
            echo "Sandi harus diisi!";
        }
    }
}

```

```

    }
    elseif (empty($_POST['txtulangisandi']))
    {
        echo "Ulangi Sandi harus diisi!";

    }
    elseif (($_POST['txtsandi'] !=
$_POST['txtulangisandi']))
    {
        echo "Sandi dan Ulangi Sandi harus sama!";

    }
    elseif (empty($_POST['txtalamat_mac']))
    {
        echo "Alamat MAC harus diisi!";

    }
    else
    {
        include 'mikrotik.php';

        $nama = $_POST['txtnama'];
        $sandik = $_POST['txtsandi'];
        $alamat_mac = $_POST['txtalamat_mac'];
        $komentar = $_POST['txtkomentar'];

        if (add_ip_hotspot_user($nama, $sandik,
$alamat_mac, $komentar))
        {
            header('location: ip_hotspot_user.php');
        }
        else
        {
            echo "Penambahan data user hotspot gagal
dilakukan!";
        }
    }
}

?>
<form id="frmAddHotspotUser" name="frmAddHotspotUser"
method="post" action="add_ip_hotspot_user.php">
    <table width="36%" border="0">
        <tr>
            <th width="36%" scope="row">Name</th>
            <td width="2%">:</td>
            <td width="62%"><label>
                <input type="text" name="txtnama" id="txtnama" />
            </label></td>
        </tr>
        <tr>
            <th scope="row">Password</th>
            <td>:</td>

```

```

        <td><input          type="password"          name="txtsandi"
id="txtsandi" /></td>
    </tr>
    <tr>
        <th scope="row">Retype Password</th>
        <td>:</td>
        <td><input          type="password"          name="txtulangisandi"
id="txtulangisandi" /></td>
    </tr>
    <tr>
        <th scope="row">MAC Address</th>
        <td>:</td>
        <td><input          type="text"          name="txtalamat_mac"
id="txtalamat_mac" /></td>
    </tr>
    <tr>
        <th scope="row">Comment</th>
        <td>:</td>
        <td><input          type="text"          name="txtkomentar"
id="txtkomentar" /></td>
    </tr>
    <tr>
        <th scope="row">&nbsp;</th>
        <td>&nbsp;</td>
        <td><label>
            <input type="submit" name="btnsubmit" id="btnsubmit"
value="Submit" />
            <input type="reset" name="btnreset" id="btnreset"
value="Reset" />
        </label></td>
    </tr>
</table>
</form>
<p><a href="ip_hotspot_user.php">Tampil Data User
Hotspot</a></p>
</body>
</html>

```

4. Buat file dengan nama "edit_ip_hotspot_user.php" yang digunakan untuk mengubah data user hotspot tertentu, dengan kode program sebagai berikut:

```

<?php
    ob_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Edit User Hotspot</title>
</head>

<body>
<?php

```

```

include 'mikrotik.php';

if (isset($_POST['btnsubmit'])) &&
$_POST['btnsubmit']=='Submit')
{
    if (empty($_POST['txtnama']))
    {
        echo "Nama harus diisi!";
    }
    elseif (empty($_POST['txtsandi']))
    {
        echo "Sandi harus diisi!";
    }
    elseif (empty($_POST['txtulangisandi']))
    {
        echo "Ulangi Sandi harus diisi!";
    }
    elseif ($_POST['txtsandi'] !=
$_POST['txtulangisandi'])
    {
        echo "Sandi dan Ulangi Sandi harus sama!";
    }
    elseif (empty($_POST['txtalamat_mac']))
    {
        echo "Alamat MAC harus diisi!";
    }
    else
    {
        $id = $_POST['id'];
        $nama = $_POST['txtnama'];
        $sandi = $_POST['txtsandi'];
        $alamat_mac = $_POST['txtalamat_mac'];
        $komentar = $_POST['txtkomentar'];

        if (edit_ip_hotspot_user($id, $nama, $sandi,
$alamat_mac, $komentar))
        {
            header('location: ip_hotspot_user.php');
        }
        else
        {
            echo "Perubahan data user hotspot gagal
dilakukan!";
        }
    }
}

// Mengambil data user hotspot tertentu dan
menampilkannya di form
if (isset($_GET['id']) && !empty($_GET['id']))

```

```

{
    // Mengambil nilai parameter querystring dengan nama
    id
    $id = $_GET['id'];

    // Memanggil function untuk mengambil data user
    hotspot tertentu
    $hotspot_users = detail_ip_hotspot_user($id);
    foreach ($hotspot_users as $key => $data)
    {
        $id = $data['.id'];
        $nama = $data['name'];
        $sandi = $data['password'];
        $ulangisandi = $data['password'];
        $alamat_mac = $data['mac-address'];
        $komentar = $data['comment'];
    }
}
?>
<form id="frmAddHotspotUser" name="frmAddHotspotUser"
method="post" action="edit_ip_hotspot_user.php">
    <table width="36%" border="0">
        <tr>
            <th width="36%" scope="row">Name</th>
            <td width="2%">:</td>
            <td width="62%"><label>
                <input type="text" name="txtnama" id="txtnama"
value="<?php if (!empty($nama)) { echo $nama; } else { echo
$_POST['txtnama']; }?>" />
            </label><input type="hidden" name="id" value="<?php
echo $id; ?>" /></td>
        </tr>
        <tr>
            <th scope="row">Password</th>
            <td>:</td>
            <td><input type="password" name="txtsandi"
id="txtsandi" value="<?php if (!empty($sandi)) { echo
$sandi; } else { echo $_POST['txtsandi']; }?>" /></td>
        </tr>
        <tr>
            <th scope="row">Retype Password</th>
            <td>:</td>
            <td><input type="password" name="txtulangisandi"
id="txtulangisandi" value="<?php if (!empty($ulangisandi)) {
echo $ulangisandi; } else { echo $_POST['txtulangisandi'];
}?>" /></td>
        </tr>
        <tr>
            <th scope="row">MAC Address</th>
            <td>:</td>
            <td><input type="text" name="txtalamat_mac"
id="txtalamat_mac" value="<?php if (!empty($alamat_mac)) {
echo $alamat_mac; } else { echo $_POST['txtalamat_mac'];
}?>" /></td>
        </tr>
    </table>

```

```

<tr>
  <th scope="row">Comment</th>
  <td>:</td>
  <td><input type="text" name="txtkomentar"
id="txtkomentar" value="<?php if (!empty($komentar)) { echo
$komentar; } else { echo $_POST['txtkomentar']; }?>" /></td>
</tr>
<tr>
  <th scope="row">&nbsp;</th>
  <td>&nbsp;</td>
  <td><label>
    <input type="submit" name="btnsubmit" id="btnsubmit"
value="Submit" />
    <input type="reset" name="btnreset" id="btnreset"
value="Reset" />
  </label></td>
</tr>
</table>
</form>
<p><a href="ip_hotspot_user.php">Tampil Data User
Hotspot</a></p>
</body>
</html>

```

5. Buat file dengan nama "remove_ip_hotspot_user.php" yang digunakan untuk menghapus data user hotspot tertentu, dengan kode program sebagai berikut:

```

<?php
  ob_start();

  include 'mikrotik.php';

  $id = $_GET['id'];

  if (remove_ip_hotspot_user($id))
  {
    header('location: ip_hotspot_user.php');
  }
  else
  {
    echo "Penghapusan user hotspot gagal dilakukan!<br
/><a href=ip_hotspot_user.php>Tampil Data User Hotspot</a>";
  }
?>

```

6. Buat file dengan nama "disable_ip_hotspot_user.php" yang digunakan untuk menonaktifkan data user hotspot tertentu, dengan kode program sebagai berikut:

```

<?php
  ob_start();

  include 'mikrotik.php';

  $id = $_GET['id'];

  if (disable_ip_hotspot_user($id))
  {

```

```

        header('location: ip_hotspot_user.php');
    }
    else
    {
        echo "Penonaktifan user hotspot gagal dilakukan!<br
/><a href=ip_hotspot_user.php>Tampil Data User Hotspot</a>";
    }
?>

```

7. Buat file dengan nama "enable_ip_hotspot_user.php" yang digunakan untuk mengaktifkan data user hotspot tertentu, dengan kode program sebagai berikut:

```

<?php
    ob_start();

    include 'mikrotik.php';

    $id = $_GET['id'];

    if (enable_ip_hotspot_user($id))
    {
        header('location: ip_hotspot_user.php');
    }
    else
    {
        echo "Pengaktifkan user hotspot gagal dilakukan!<br
/><a href=ip_hotspot_user.php>Tampil Data User Hotspot</a>";
    }
?>

```

8. Buat file dengan nama "detail_ip_hotspot_user.php" yang digunakan untuk menampilkan detail informasi dari user hotspot tertentu, dengan kode program sebagai berikut:

```

<?php
    // Menyisipkan Mikrotik.php yang menampung function-
function untuk manajemen mikrotik
    include('mikrotik.php');

    // Mengambil nilai dari querystring untuk parameter id
    $id = $_GET['id'];

    // Memanggil function getHotspotUsers dan menampung hasil
panggilan pada variable $hotspot_users
    $hotspot_users=detail_ip_hotspot_user($id);

    // Membuat tabel
    echo "<table          border='1'          cellpadding='2'
cellspacing='2'>";
    // Membuat baris (row) sebagai header (judul kolom) dari
tabel
    echo "<tr>";
    echo "<th>.id</th>";
    echo "<th>Server</th>";
    echo "<th>Nama Login</th>";
    echo "<th>Sandi Login</th>";

```

```

echo "<th>Alamat MAC</th>";
echo "<th>Comment</th>";
echo "<th>Profile</th>";
echo "<th>Disabled</th>";
echo "</tr>";

// Membuat perulangan untuk menampilkan data yang
ditampung pada variable "hotspot_users"
foreach ($hotspot_users as $key => $data)
{
    // Membuat baris (row) dari tabel
    echo "<tr>";
    echo "<td bgcolor='white'>". $data['.id'] . "</td>";
    echo "    <td bgcolor='white'>". $data['server']
    . "</td>";
    echo "<td bgcolor='white'>". $data['name'] . "</td>";
    echo "    <td bgcolor='white'>". $data['password']
    . "</td>";
    echo "    <td bgcolor='white'>". $data['mac-address']
    . "</td>";
    echo "    <td bgcolor='white'>". $data['comment']
    . "</td>";
    echo "    <td bgcolor='white'>". $data['profile']
    . "</td>";
    echo "    <td bgcolor='white'>". $data['disabled']
    . "</td>";
    echo "</tr>";
}
echo "</table>";
echo "<a href='ip_hotspot_user.php'>Kembali ke Tampil IP
Hotspot User</a>";
?>

```

BAB 7**Mikrotik PHP API PADA FIREWALL FILTER****Tujuan:**

Mahasiswa memahami mengenai Mikrotik PHP API untuk manajemen firewall.

Materi:**A. Pengenalan IP Firewall Filter.**

Menurut manual Mikrotik, IP Firewall Filter merupakan sebuah fitur pada Mikrotik yang dapat digunakan untuk membangun firewall dengan cara mengimplementasikan pemfilteran pada paket, sehingga aliran paket data ke, dari, dan melewati router dapat dimanajemen.

B. Perintah-perintah Mikrotik CLI untuk melakukan packet filtering menggunakan ip firewall filter.**1. Menambahkan aturan pada ip firewall filter**

```
ip firewall filter add chain=forward action=drop src-address=192.168.171.0/24 comment="blokir hotspot"
```

2. Mengubah aturan pada ip firewall filter

```
ip firewall filter edit 1  
value-name=comment
```

3. Menghapus aturan pada ip firewall filter

```
ip firewall filter remove 1
```

4. Menampilkan informasi seluruh aturan ip firewall filter

```
ip firewall filter print
```

5. Menonaktifkan ip firewall filter tertentu

```
ip firewall filter disable 1
```

6. Mengaktifkan ip firewall filter tertentu

```
ip firewall filter enable 1
```

C. Perintah-perintah Mikrotik PHP API untuk manajemen ip firewall filter.**1. Menambahkan aturan pada ip firewall filter**

```
/ip/firewall/filter/add  
=chain=?
```

```
=action=?


```

2. Mengubah aturan pada ip firewall filter

```
/ip/firewall/filter/set
=.id=?
=chain=?
=action=?


```

3. Menghapus aturan pada ip firewall filter

```
/ip/firewall/filter/remove
=.id=?
```

4. Menampilkan informasi seluruh aturan ip firewall filter

```
/ip/firewall/filter/getall
```

5. Menonaktifkan ip firewall filter tertentu

```
/ip/firewall/filter/disable
=.id=?
```

6. Mengaktifkan ip firewall filter tertentu

```
/ip/firewall/filter/enable
=.id=?
```

D. Studi Kasus aplikasi manajemen firewall menggunakan Mikrotik PHP API.

Berdasarkan contoh aplikasi manajemen *user sistem*, dan *ip hotspot* pada bab sebelumnya, serta pembahasan perintah-perintah Mikrotik PHP API untuk memanajemen IP Firewall Filter, buatlah aplikasi manajemen firewall dengan antar muka sebagai berikut:

1. Halaman yang menampilkan informasi seluruh aturan pada IP Firewall Filter, seperti terlihat pada gambar berikut:

*ID	ACTION	CHAIN	SOURCE ADDRESS	DESTINATION ADDRESS	PROTOCOL	DESTINATION PORT	DISABLED	ACTION
*2	jump	forward					false	Update Delete Disable
*3	jump	forward					false	Update Delete Disable
*4	jump	input					false	Update Delete Disable
*5	drop	input			tcp	64872-64875	false	Update Delete Disable
*6	jump	hs-input					false	Update Delete Disable
*7	accept	hs-input			udp	64872	false	Update Delete Disable
*8	accept	hs-input			tcp	64872-64875	false	Update Delete Disable
*9	jump	hs-input					false	Update Delete Disable
*A	reject	hs-unauth			tcp		false	Update Delete Disable
*B	reject	hs-unauth					false	Update Delete Disable
*C	reject	hs-unauth-to					false	Update Delete Disable
*1	passthrough	unused-hs-chain					true	Update Delete Enable
*D	reject	input	192.168.19.0/24	192.168.17.0/24	udp	8000	true	Update Delete Enable

[Tambah IP Firewall Filter](#)

2. Halaman yang digunakan untuk menambahkan aturan IP Firewall Filter baru, seperti terlihat pada gambar berikut:

Tampil IP Firewall Filter

FORM TAMBAH IP FIREWALL FILTER

Chain	:	Choose ...
Source Address	:	
Destination Address	:	
Protocol	:	Choose ...
Destination Port	:	
Action	:	Choose
Comment	:	
Status	:	☐ Enable ☐ Disable
		Submit Reset

Berikut adalah beberapa contoh kode program dalam bentuk fungsi yang dapat digunakan untuk manajemen IP Firewall Filter:

```
<?php
// Menambahkan aturan IP Firewall Filter
function tambah_ip_firewall_filter($src_address,
    $comment)
{
    // Deklarasi variable dengan scope global
    global $hostname, $username, $password;

    // Membuat instance (objek) dari class
    routeros_api
    $API = new routeros_api();

    $chain="forward";
    $action="drop";
    // Membuat koneksi ke router Mikrotik
    if ($API->connect($hostname, $username,
        $password))
    {
        /*
        * Memanggil method "write" dari objek yang
        digunakan untuk mengeksekusi perintah2 dari router
        Mikrotik
        */
    }
}
```

```

        *   Parameter   "/ip/firewall/filter/add"
digunakan untuk menambahkan ip firewall filter baru di
router Mikrotik
        *   dengan   argumen   chain,   action,
src_address, dan comment.
        */
        $API->write('/ip/firewall/filter/add',
false);

        $API->write('=chain='.$chain, false);
        $API->write('=action='.$action, false);
        $API->write('=src-address='.$src_address,
false);

        $API->write('=comment='.$comment);
        $API->read();

        // Memutuskan koneksi dari router mikrotik
        $API->disconnect();

        return true;
    }
    else
    {
        return false;
    }
}

// Menghapus aturan IP Firewall Filter
function hapus_ip_firewall_filter($comment)
{
    // Deklarasi variable dengan scope global
    global $hostname, $username, $password;

    // Membuat instance (objek) dari class
    routers_api
    $API = new routers_api();

    // Membuat koneksi ke router Mikrotik
    if ($API->connect($hostname, $username,
$password))
    {
        /*
        * Memanggil method "write" dari objek yang
digunakan untuk mengeksekusi perintah2 dari router
Mikrotik

        * Parameter "/ip/firewall/filter/remove"
pada method "write" berfungsi untuk menghapus ip
firewall filter tertentu
        * yang terdapat di router Mikrotik.
        */

```

```

        $API-
>write('/ip/firewall/filter/remove',false);
        $API->write('=.id=.'.$comment, true);
        $API->read();

        // Memutuskan koneksi dari router mikrotik
        $API->disconnect();

        return true;
    }
    else
    {
        return false;
    }
}

// Mengubah aturan IP Firewall Filter
function      ubah_ip_firewall_filter($comment_old,
$comment_new, $src_address_new)
{
    // Deklarasi variable dengan scope global
    global $hostname, $username, $password;

    // Membuat instance (objek) dari class
    routers_api
    $API = new routers_api();

    // Membuat koneksi ke router Mikrotik
    if      ($API->connect($hostname,      $username,
$password))
    {
        // Memanggil method "write" dari objek
yang digunakan untuk mengeksekusi perintah2 dari router
Mikrotik
        // Parameter "/ip/firewall/filter/print"
pada method "write" berfungsi untuk memperoleh ip
firewall filter tertentu yang terdapat di router
Mikrotik
        $API->write('/ip/firewall/filter/print',
false);
        $API->write('=.proplist=.id', false);
        $API->write('?comment=.'.$comment_old,
true);

        $ip_firewall_filter = $API->read();

        foreach ($ip_firewall_filter[0] as $key =>
$data)
        {
            $id = $data;
        }
    }
}

```

```

        // Memanggil method "write" dari objek
yang digunakan untuk mengeksekusi perintah2 dari router
Mikrotik
        // Parameter "/ip/firewall/filter/set"
pada method "write" berfungsi untuk mengubah nilai dari
        // property "comment" dan "src-address"
dari ip firewall filter
        $API-
>write('/ip/firewall/filter/set',false);
        $API->write('=.id=',$id, false);
        $API->write('=.comment=',$comment_new,
false);
        $API->write('=.src-
address=',$src_address_new, true);
        $API->read();
        // Memutuskan koneksi ke router Mikrotik
        $API->disconnect();
        return true;
    }
    else
    {
        return false;
    }
}
?>

```

BAB 8

Koneksi PHP ke database MySQL menggunakan MySQLi Extension

Tujuan:

Mahasiswa memahami mengenai koneksi PHP ke database MySQL.

Materi:**1. Fungsi-fungsi PHP untuk mengakses MySQL.**

PHP menyediakan fungsi-fungsi untuk mengakses database MySQL. Berikut ini ditampilkan fungsi-fungsi yang sering digunakan untuk mengakses MySQL melalui PHP.

Tabel Fungsi-fungsi akses MySQL melalui PHP

Fungsi mysql	Keterangan
<code>mysqli_connect(string server, string username, string password, string database)</code>	Digunakan untuk membuka koneksi ke server MySQL dan memilih database yang akan digunakan.
<code>mysqli_query(resource link_identifier, string query)</code>	Digunakan untuk mengirimkan perintah SQL ke server MySQL.
<code>mysqli_num_rows(resource result)</code>	Digunakan untuk mengambil jumlah baris pada hasil.
<code>mysqli_fetch_array(resource result)</code>	Digunakan untuk mengambil baris hasil sebagai array asosiatif (menggunakan referensi key berupa nama field) atau array satu per satu (menggunakan referensi key berupa nomor urutan field).

<code>mysqli_fetch_row(resource result)</code>	Digunakan untuk menampilkan baris hasil array satu per satu menggunakan referensi key berupa nomor urutan field.
<code>mysqli_error(resource_link_identifier)</code>	Digunakan untuk menampilkan pesan kesalahan dari server MySQL.

Secara umum untuk mengakses database pada server MySQL dilakukan melalui 3 tahapan yaitu **membuka koneksi**, melakukan **query**, dan **menutup koneksi**. Berikut ini diberikan contoh kode program untuk menampilkan data yang tersimpan pada tabel “**users**” di database MySQL dengan nama “**mikrotik_php_api_modul**”:

Contoh:

```
<?php
    // Mengatur variable koneksi
    $hostname_database = 'localhost';
    $username_database = 'root';
    $password_database = '';
    $database_name = 'mikrotik_php_api_modul';

    // Koneksi ke database & memilih database aktif
    $koneksi = mysqli_connect($hostname_database,
    $username_database, $password_database, $database_name);

    // Mengecek apakah berhasil terkoneksi atau tidak
    if (!$koneksi)
    {
        echo "Koneksi ke MySQL Server gagal dilakukan!";
    }
    else
    {

```

```

// Membuat perintah query

$sql = "SELECT * FROM users";

// Mengeksekusi query dan
// menampilkan pesan kesalahan jika gagal

$rset = mysqli_query($koneksi, $sql) or die("Query gagal:
". mysqli_error($koneksi));

// Menghitung jumlah baris hasil eksekusi query
$jumlahrecord = mysqli_num_rows($rset);

// Mengecek nilai jumlah baris hasil eksekusi query
// Jika nilainya 0 maka tidak ada data,
// sebaliknya tampilkan data hasil query
if ($jumlahrecord == 0)
{
    echo "Data users tidak ditemukan!";
}
else
{
    echo "<h1>DATA USERS </h1>";
    // melakukan perulangan untuk menampilkan data
    while ($hasil = mysqli_fetch_array($rset))
    {
        echo "ID : ". $hasil['id'] . "<br />";
        echo "Username: ". $hasil['username'] . "<br
/>";
        echo "Password: ". $hasil['password'] . "<br
/>";
        echo "Access Level: ". $hasil['access_level']
. "<br />";
        echo "<hr />";
    }
}

```

```

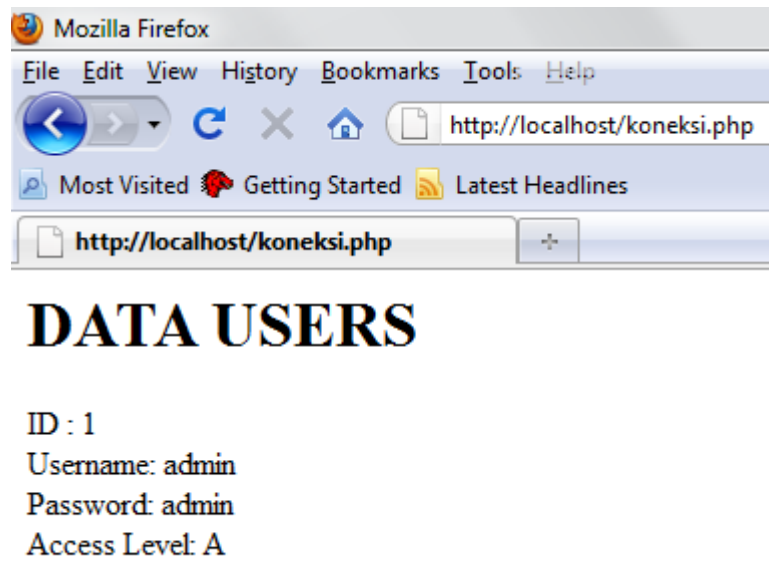
    }

}

?>

```

Hasilnya dari eksekusi skrip tersebut adalah sebagai berikut:



2. Studi kasus aplikasi **Create, Read, Update, Delete (CRUD)** dari PHP ke database MySQL

a) Melalui browser akses aplikasi **PhpMyAdmin** pada alamat "<http://localhost/phpmyadmin>", dan buat database dengan nama "*mikrotik_php_api_modul*", serta tabel "*users*" dengan nama-nama field dan tipe data sebagai berikut:

```

CREATE TABLE IF NOT EXISTS `users` (
  `id` smallint(6) NOT NULL AUTO_INCREMENT,
  `username` varchar(20) NOT NULL,
  `password` varchar(255) NOT NULL,
  `access_level` enum('A','E','O') NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin;

```

b) Buat file dengan nama "**config.php**" yang digunakan untuk melakukan koneksi dari PHP ke MySQL dengan kode program berikut:

```

<?php
    // mendeklarasikan variable untuk koneksi dari PHP ke
    MySQL Server
    $hostname = "localhost"; // cantumkan alamat ip atau nama
    komputer MySQL server

```

```

$username = "root"; // cantumkan nama user yg digunakan
untuk login
$password = ""; // cantumkan sandi user yg digunakan
untuk login
$dbase = "mikrotik_php_api_modul"; // cantumkan nama
database yg ingin diakses

```

```

// membuat koneksi ke MySQL Server
$conn = mysqli_connect($hostname, $username,
$password, $dbase);
// mengecek status koneksinya
if (!$conn)
{
    die("Gagal terkoneksi ke MySQL Server!");
}
?>

```

- c) Buat file dengan nama **"index.php"** yang digunakan untuk menampilkan data user yang terdapat pada tabel **"users"** di database **"mikrotik_php_api_modul"** dengan kode program berikut:

```

<?php
// menyisipkan file config.php untuk mengkoneksikan ke
MySQL Server
include("config.php");

// membuat perintah SQL
$sql = "SELECT * FROM users";

// mengeksekusi perintah tersebut
$result = mysqli_query($conn, $sql);

// menghitung jumlah baris hasil query
$numrecords = mysqli_num_rows($result);

// mengecek isian variable numrecords
if ($numrecords == 0)
{
    echo "Data user tidak ditemukan!<br>";
}
else
{
    echo "Terdapat $numrecords data user!<br>";

    echo "<table border='1'>";
    echo "<tr>";
    echo "<th>No.</th>";
    echo "<th>ID</th>";
    echo "<th>Username</th>";
    echo "<th>Password</th>";
    echo "<th>Access Level</th>";
    echo "<th>Action</th>";
    echo "</tr>";
    $no = 1;
    while ($row = mysqli_fetch_array($result))

```

```

{
    switch ($row['access_level'])
    {
        case 'A':
            $access_level='Administrator';
            break;
        case 'E':
            $access_level='Editor';
            break;
        case 'O':
            $access_level='Operator';
            break;
    }

    echo "<tr>";
    echo "<td>$no.</td>";
    echo "<td>".$row['id']."</td>";
    echo "<td>".$row['username']."</td>";
    echo "<td>".$row['password']."</td>";
    echo "<td>$access_level</td>";
    echo "<td><a
href=\"ubah.php?id=".$row['id']."\">Ubah</a> | <a
href=\"hapus.php?id=".$row['id']."\">Hapus</a></td>";
    echo "</tr>";
    $no++; // $no = $no + 1;

}
echo "</table>";
}

?>
<a href="tambah.php">Tambah User</a>

```

- d) Buat file dengan nama "tambah.php" yang digunakan untuk dengan kode program berikut:

```

<?php
// mengecek apakah terjadi penekanan tombol submit
if (isset($_POST['btnsubmit']))
{
    // mengambil nilai isian form
    $nama = $_POST['txtusername'];
    $sandi = $_POST['txtpassword'];
    $hak_akses = $_POST['lstaccess_level'];

    // menyisipkan file config.php
    include("config.php");
    // membuat perintah SQL
    $sql = "INSERT INTO users (username, password,
access_level)
        VALUES ('$nama','$sandi','$hak_akses')
        ";
    // mengeksekusi perintah
    $hasil = mysqli_query($koneksi, $sql);
    //mengecek hasil
    if ($hasil)

```

```

        {
            echo "Data user berhasil disimpan!<br>";
        }
        else
        {
            echo "Data user gagal disimpan!<br>";
        }
    }
?>
<a href="index.php">Tampil Data User</a>
<form name="frmusers" method="post" action="tambah.php">
Username : <input type="text" name="txtusername"><br>
Password : <input type="password" name="txtpassword"><br>
Access Level :
<select name="lstaccess_level">
<option value="">Choose...</option>
<option value="A">Administrator</option>
<option value="E">Editor</option>
<option value="O">Operator</option>
</select><br>
<input type="submit" name="btnsubmit" value="Submit">
<input type="reset" name="btnreset" value="Reset">
</form>

```

- e) Buat file dengan nama **"ubah.php"** yang digunakan untuk mengubah data user yang terdapat di tabel **"users"** dengan kode program berikut:

```

<?php
// menyisipkan file config.php
include("config.php");

// mengecek apakah terjadi penekanan tombol submit
if (isset($_POST['btnsubmit']))
{
    // mengambil nilai isian form
    $id = $_POST['id'];
    $nama = $_POST['txtusername'];
    $sandi = $_POST['txtpassword'];
    $hak_akses = $_POST['lstaccess_level'];

    // membuat perintah SQL
    $sql = "UPDATE users SET username='$nama',
password='$sandi', access_level='$hak_akses'
WHERE id=$id
";

    // mengeksekusi perintah
    $hasil = mysqli_query($koneksi, $sql);
    //mengecek hasil
    if ($hasil)
    {
        echo "Data user berhasil diubah!<br>";
    }
    else
    {
        echo "Data user gagal diubah!<br>";
    }
}

```

```

    }
    else
    {
        // mengambil nilai querystring id
        $id = $_GET['id'];

        // membuat perintah SQL
        $sql = "SELECT * FROM users WHERE id=$id";

        // mengeksekusi perintah tersebut
        $recordset = mysqli_query($koneksi, $sql);
        $row = mysqli_fetch_array($recordset);
        $nama = $row['username'];
        $hak_akses = $row['access_level'];
    }
?>
<a href="index.php">Tampil Data User</a>
<form name="frmusers" method="post" action="ubah.php">
Username : <input type="text" name="txtusername" value="<?php
if      (isset($_POST['txtusername']))      {      echo
$_POST['txtusername']; } else { echo $nama; } ?>"><br>
Password : <input type="password" name="txtpassword"><br>
Access Level :
<select name="lstaccess_level">
<option value="">Choose...</option>
<option value="A" <?php if (isset($_POST['lstaccess_level'])
&& ($_POST['lstaccess_level'] == 'A')) { echo "selected"; }
else { if ($hak_akses=='A') { echo "selected"; } }
?>>Administrator</option>
<option value="E" <?php if (isset($_POST['lstaccess_level'])
&& ($_POST['lstaccess_level'] == 'E')) { echo "selected"; }
else { if ($hak_akses=='E') { echo "selected"; } }
?>>Editor</option>
<option value="O" <?php if (isset($_POST['lstaccess_level'])
&& ($_POST['lstaccess_level'] == 'O')) { echo "selected"; }
else { if ($hak_akses=='O') { echo "selected"; } }
?>>Operator</option>
</select><br>
<input type="submit" name="btnsubmit" value="Submit">
<input type="reset" name="btnreset" value="Reset">
<input type="hidden" name="id" value="<?php echo $id; ?>">
</form>

```

- f) Buat file dengan nama **"hapus.php"** yang digunakan untuk menghapus user di tabel **"users"** dengan kode program berikut:

```

<?php
// menyisipkan file config.php
include("config.php");

// mengambil nilai querystring id
$id = $_GET['id'];

// membuat perintah SQL untuk menghapus record

```

```
$sql = "DELETE FROM users WHERE id=$id";

// mengeksekusi perintah tersebut
$hasil = mysqli_query($koneksi, $sql);

// mengecek variable hasil
if ($hasil)
{
    echo "Data user berhasil dihapus!<br>";
}
else
{
    echo "Data user gagal dihapus!<br>";
}
?>
<a href="index.php">Tampil Data User</a>
```

DAFTAR REFERENSI

Mikrotik Manual, wiki.mikrotik.com

Modul Workshop Mikrotik PHP API, I Putu Hariyadi, STMIK Bumigora, Mataram, 2010

Panduan Lengkap Menguasai Router Masa Depan Menggunakan Mikrotik RouterOS, Moch. Linto Herlambang, Azis Catur L., Andi Publisher, Yogyakarta, 2008

PHP5 Pemrograman Berorientasi Objek Konsep & Implementasi, Stendy B. Sakur, Andi Publisher, Yogyakarta, 2011

STMIK BUMIGORA MATARAM