

Manajemen User Mikrotik Berbasis Telegram Bot

Ananda Adhe Suqma¹, Agus Prihanto²

^{1,2}Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

¹anandasuqma16051204002@mhs.unesa.ac.id

²agusprihanto@unesa.ac.id

Abstrak—*Smartphone* merupakan alat yang penting untuk menunjang aktivitas manusia dalam melakukan pekerjaan, hal itu dikarenakan *smartphone* dapat mengakses informasi dari satu tempat ke tempat lain dengan mudah dan cepat walaupun jaraknya jauh. Salah satu contoh layanan *smartphone* adalah sosial media yang berfungsi sebagai media telekomunikasi dan informasi. Berbagai aplikasi sosial media tersedia pada *smartphone*, salah satunya adalah telegram. Di dalam telegram terdapat fitur *telegram bot*. Saat ini telegram bot mulai dikembangkan untuk dapat monitoring mikrotik dan melakukan perintah untuk manajemen *user* mikrotik. Untuk melakukan hal tersebut, administrator harus terkoneksi dengan jaringan router untuk dapat melakukan manajemen *user hotspot*. Berdasarkan masalah tersebut, administrator yang mengelola *user hotspot* dapat menggunakan sebuah *Telegram Bot* tanpa melalui 1 jaringan yang sama dengan *router mikrotik*. Ketika *user login*, *Telegram Bot* menampilkan informasi *IP*, *Mac Address*, dan *username user* yang melakukan *login* ke *hotspot*. Begitu juga apabila administrator melakukan manajemen *user*, langkah dan waktu yang diperlukan untuk melakukan perintah manajemen *user hotspot* pada *Telegram bot* memerlukan langkah dan waktu yang lebih sedikit dibandingkan dengan melalui *Winbox* dan juga administrator bisa melakukan manajemen *user* dan monitoring dari jarak jauh. Hasil penelitian menunjukkan bahwa menjalankan perintah manajemen *user hotspot* Mikrotik dengan Telegram lebih efektif dan efisien jika dibandingkan dengan menggunakan *Winbox*.

Kata Kunci— *sosial media, telegram, hotspot, mikrotik*

I. PENDAHULUAN

Smartphone merupakan alat yang penting untuk menunjang aktivitas manusia dalam melakukan pekerjaan, hal itu dikarenakan *smartphone* dapat mengakses informasi dari satu tempat ke tempat yang lain dengan mudah dan cepat walaupun dengan jarak yang jauh. Selain praktis dan dapat dibawa kemana - mana *smartphone* juga mampu memberikan kemudahan bagi penggunanya. Salah satu contoh layanan *smartphone* adalah sosial media yang berfungsi sebagai media telekomunikasi dan informasi. Salah satunya adalah telegram. Di dalam telegram itu sendiri memiliki fitur *telegram bot*, yang dimana dapat dikembangkan oleh pengguna.

Telegram merupakan aplikasi yang membuat pengguna atau user bisa melakukan pengiriman pesan secara cepat dan aman, disamping itu *Telegram* bisa menyajikan kemudahan pelayanan dan bersifat gratis. *Telegram* bisa dipakai pada *smartphone*, tablet dan juga komputer. *Telegram* untuk platform *iOS* telah dirilis pada tanggal 14 Agustus 2013[1]

Sedangkan *Bot Telegram* merupakan aplikasi perpesanan instan yang tidak membutuhkan nomor telepon untuk mendaftarkan ke *server Telegram*. Akun ini yang akan

difungsikan sebagai antarmuka untuk berkomunikasi dengan *server Telegram*. Sehingga pengguna dapat membuat semacam aplikasi *chatting*. [2]

Pada saat ini, aplikasi media sosial telegram tidak hanya dapat digunakan untuk berkomunikasi antar perseorangan, tetapi komunikasi juga dapat digunakan dari mesin ke personal dan bisa bermanfaat untuk berbagai hal, satu diantaranya adalah bermanfaat sebagai alternatif pengantar pesan untuk memberikan informasi terkait status koneksi internet kepada admin dan teknisi jika koneksi jaringan terputus. Dengan menggunakan teknologi *Telegram Bot API (Application Programming Interface)*, *server* mampu melakukan komunikasi dengan administrator dengan cara memanfaatkan program *client* pada telegram yang berada pada *smartphone* android administrator. [3]

Saat ini administrator masih menggunakan *Winbox* untuk melakukan konfigurasi manajemen *user* dan melakukan monitoring *router* Mikrotik. Hal tersebut masih terkendala karena terkadang administrator masih perlu menggunakan laptop kemudian memakai *Winbox* atau alamat *IP router*. Dan diwajibkan juga ketika konfigurasi menggunakan satu jaringan yang sama dengan *router* Mikrotik. Permasalahan yang lain juga ketika banyak *user* yang mendaftar untuk mendapatkan *hotspot* agar segera menggunakan jaringan, harus menunggu terlalu lama karena bergantian administrator melakukan registrasi *user hotspot* untuk diberikan kepada *user*. Masalah lain yang muncul adalah ketika administrator tidak sedang berada dekat dengan *router*, kemudian ada *user* yang ingin segera menikmati internet. Sehingga administrator perlu untuk berada di dekat *router*, atau harus menggunakan jaringan yang sama untuk membuat konfigurasi *user hotspot*. Berdasarkan masalah tersebut, administrator yang mengelola *user hotspot* dapat menggunakan sebuah *telegram bot* tanpa melalui jaringan yang sama dengan *router* Mikrotik dan juga mempersingkat waktu dan langkah untuk melakukan pengelolaan *user hotspot*.

Penelitian sebelumnya telah dijalankan oleh Dimara Kusuma Hakim dan Septian Adi Nugroho, 2019 "Implementasi *Telegram Bot* untuk *Monitoring* Mikrotik Router" yang menghasilkan sebuah simpulan bahwa menggunakan *telegram bot* memudahkan admin atau teknisi dalam memperoleh informasi ketika user melakukan *login* maupun *logout* ke dalam jaringan.[4]. Mengacu pada penelitian tersebut, *telegram bot* masih dapat dikembangkan yakni pengembangan untuk dapat melakukan manajemen *user* mikrotik melalui *telegram bot*. Pengembangan tersebut menjadi pembeda dari penelitian sebelumnya yang hanya berfokus pada *monitoring user*.

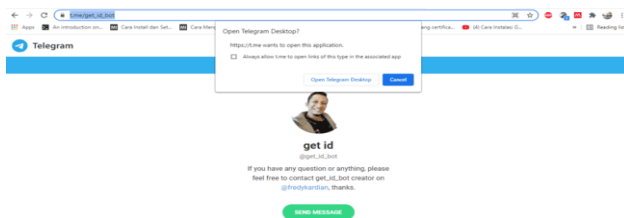
Hasil akhir penelitian ini diharapkan dapat menambah fitur baru pada *telegram bot* dan mempermudah administrator dalam mengelola penambahan *user hotspot*, penghapusan *user hotspot*, penghapusan *session user hotspot*, pemindahan *user profile*, penggantian *password user hotspot* dan juga melakukan *monitoring router* yang bisa dilakukan oleh administrator melalui *telegram bot* dan tidak berada di jaringan yang sama dengan *router Mikrotik* sehingga lebih efektif dan efisien.

II. KAJIAN PUSTAKA

A. Telegram API

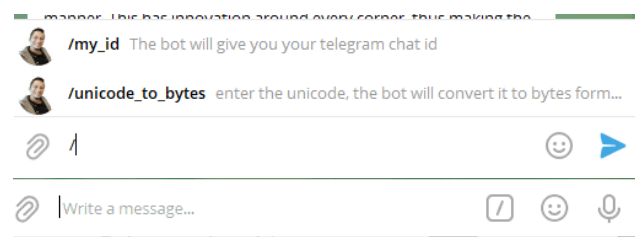
Telegram adalah sebuah aplikasi perpesanan instan yang berbasis *cloud*, yang memberikan kemudahan terhadap pemakainya dalam mendapatkan akses satu account Telegram melalui perangkat lain diwaktu yang sama. Dan juga bisa membagikan jumlah berkas yang hingga 1,5 GB. [5]. Didalam Telegram terdapat fitur Bot API. Telegram Bot API (*Application Programming Interface*) merupakan sebuah fitur yang telah dibuat oleh telegram yang diperuntukkan untuk para pengembang atau developer melakukan pengembangan BOT telegram secara bebas karena bersifat open source. Bot API ini mempunyai interface yang basisnya ialah HTTP yang mana fungsinya agar bias menjadi penghubung antara pengembang dan system telegram[6]. Telegram telah menciptakan 2 jenis API untuk para developer bias mengembangkan system berbasis telegram yakni *Bot API (Bot Engine)* dan *TDLib (Telegram Database Library)*[7]. *Bot Engine* ialah sebuah aplikasi pihak ketiga yang bisa melakukan interaksi secara langsung melalui token dengan Bot Telegram dengan pengiriman pesan, perintah (*command*) dan *inline request*. *Bot Engine* bisa dipakai untuk berita dan notifikasi khusus, bisa berintegrasi dengan layanan lain, bisa mendapat pembayaran dari user telegram lain, bisa dipakai sebagai alat khusus, bisa dipakai sebagai Game (Single maupun Multiplayer), bisa dipakai sebagai layanan media sosial, bisa dipakai sebagai layanan virtual lain[7]. Disetiap akun telegram yang telah terdaftar pasti memiliki Chat ID. User ID digunakan untuk berkomunikasi dengan Bot. Dengan Chat ID kita maka Bot akan mengirimkan pesan ke Chat ID. Berikut langkah-langkah untuk mendapatkan Chat ID :

- 1) Buka https://t.me/get_id_bot. (Bila di desktop, maka pertama-tama membutuhkan penginstallan telegram versi dektop. Jika pengembang diminta agar launch telegram desktop maka memilih untuk allow supaya masuk ke telegram kita ke kotak chat get id



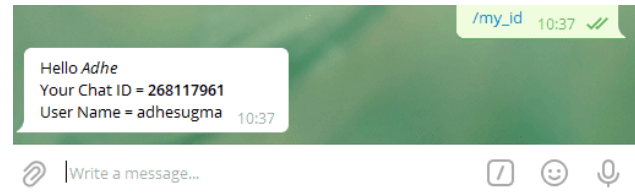
Gbr. 1 Langkah pertama mendapatkan chat ID

- 2) Klik ikon *command bot* “/” kemudian pilih `/my_id`



Gbr. 2 Langkah Kedua Mendapatkan Chat ID

- 3) Kemudian kita akan memperoleh token *API* dari *Bot Father*, untuk bisa memakai token tersebut maka *copy*-lah token tersebut agar bisa dipakai pada *script Mikrotik*



Gbr. 3 Langkah ketiga mendapatkan chat ID

Bot Telegram ialah akun telegram yang memang telah dikhususkan proses pelayanannya dengan tidak butuh bantuan nomor telepon dalam menjalankannya. Akun ini memiliki fungsi untuk menjadi interface dalam menjalankan code yang telah dibuat. Semua yang berhubungan dengan enkripsi dan komunikasi semuanya di tangani oleh telegram dengan tujuan keamanan, yang akhirnya para pengembang tidak butuh untuk tahu tentang bagaimana protokol enkripsi *MTPProto* pada server Telegram bekerja[8]. Bot Telegram juga dapat mengirimkan pesan, gambar, video dan file jenis apapun. Ada beberapa metode yang dipakai untuk melakukan perancangan terhadap sebuah Bot di Telegram Messenger. Berikut ialah beberapa metode tersebut yakni *sendMessage*, *sendPhoto*, *sendDocument*, *sendVideo*, *sendLocation*, *editMessageText*. Semua query ke Telegram Bot API harus melalui HTTPS dan disajikan dalam bentuk `https://api.telegram.org/bot/NAMA_METODE`. Contohnya:
`https://api.telegram.org/bot123456:ABC-DEF1234ghIklzxyS7W2v1u123ew11/getMe`. [8]

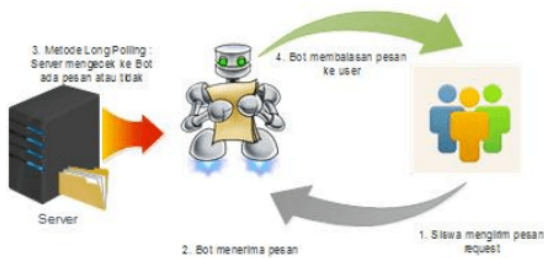
B. Komunikasi dengan Bot Server

Ada 2 cara untuk membuat *bot telegram*, yakni dengan memakai metode *long-polling* dan metode *webhook*. *Long-polling* itu sendiri ialah metode *default* dari telegram. Akan tetapi *webhook* merupakan metode yang memerlukan pemasangan pada *server hosting*.

C. Long Pooling

Long Polling merupakan sebuah metode yang dipakai untuk memperoleh update pada *bot*. Dengan menggunakan metode *long-polling*, maka *server* akan melakukan pengecekan secara periodik ke *Bot*. Jika telegram menerima pesan, maka server akan bereaksi yang disesuaikan pesan

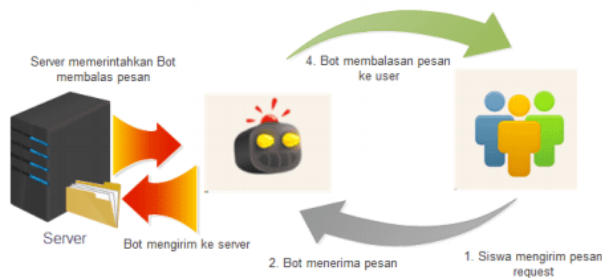
yang didapat. Jika tidak ada pesan yang masuk, maka kondisi server *idle*. Alur penggunaan telegram *bot* dengan metode *long-polling* ialah seperti pada gambar 4.[9].



Gbr. 4 Bot telegram dengan metode *long-polling*
(sumber : H. Soeroso *et al.*, 2017 pp. 46)

D. Webhook

Webhook ialah sebuah metode yang sistemnya ialah jika bot tersebut mendapat pesan di telegram dan akan meneruskan ke alamat website tertentu. *Web server* akan berada pada hosting dan wajib menggunakan *https*. Yang akhirnya bot yang berada di server bisa terakses oleh pengguna lain. Hal tersebut dikarenakan *API Telegram Bot* yang memberi izin pada pengembang untuk menciptakan sebuah bot yang didasarkan pada 'bot' yang pengembang inginkan. Sehingga terdapat **Input-Proses-Output** yang terjadi pada bot telegram. Alur penggunaan *telegram bot* dengan metode *webhook* ialah seperti pada gambar 5 [9].



Gbr. 5 Bot telegram dengan metode *webhook*
(sumber : H. Soeroso *et al.*, 2017 pp. 46)

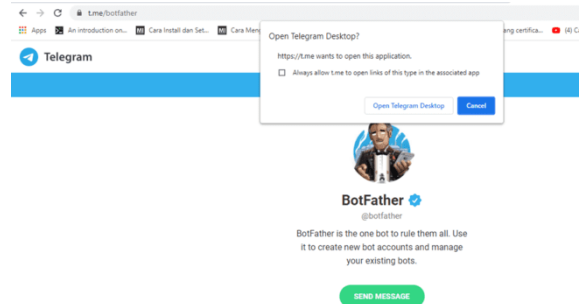
Penelitian ini menggunakan metode Long-Polling untuk *Bot API*. Penggunaan long-polling memudahkan penulis dalam melakukan proses pembuatan, pengujian, dan menemukan kesalahan kode program pada *chatbot* yang dibuat. Di sisi lain, penggunaan metode long-polling tidak mengharuskan router mikrotik memiliki *IP Public* dan membutuhkan *hosting server*. *Interval* untuk melakukan *update* juga dapat diatur pada *scheduler* Mikrotik.

E. BotFather

Sebelum menggunakan Bot Telegram, hal pertama yang wajib dilakukan ialah mendapat token *API* dengan cara membuat bot baru dengan mengakses Bot Father dengan link <https://t.me/botfather>. Token *API* merupakan hal utama yang

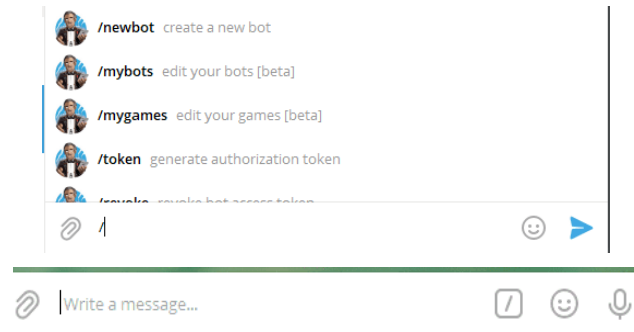
diperlukan untuk mengakses Bot. Berikut langkah-langkah untuk mendapatkan Token *API* [10]:

- 1) Buka <https://t.me/botfather> pada browser. Jika pada desktop, maka diawal membutuhkan proses instalasi telegram versi dulu. Jika pengembang diminta agar launch telegram desktop maka memilih untuk allow supaya masuk ke telegram kita ke kotak chat Bot Father



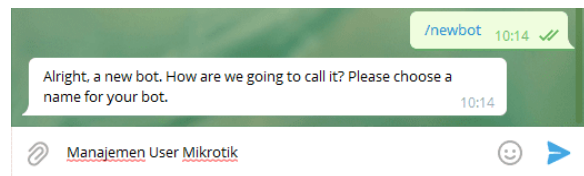
Gbr. 6 Langkah pertama membuat bot

- 2) Klik ikon *command bot* "/" kemudian pilih /newbot



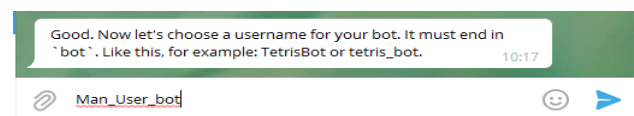
Gbr. 7 Langkah kedua membuat bot

- 3) Tuliskan judul *bot* yang akan dibuat



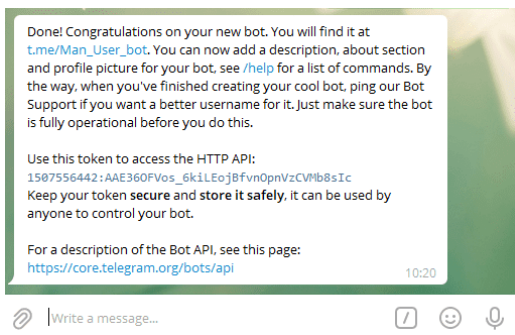
Gambar 8. Langkah Ketiga Membuat Bot

- 4) Lalu tulis username yang mana username itu akan dijadikan sebagai penamaan unit dari bot yang akan dijalankan. Terdapat aturan tertentu untuk membuat username ini yakni username haruslah diakhiri dengan memakai 'bot' di akhir judul. Misalkan Man_User_bot.



Gbr. 9 Langkah keempat membuat bot

- 5) Lalu kita akan memperoleh token API dari Bot Father, untuk bisa memakai token tersebut maka *copy*-lah token tersebut agar bisa dipakai pada script Mikrotik

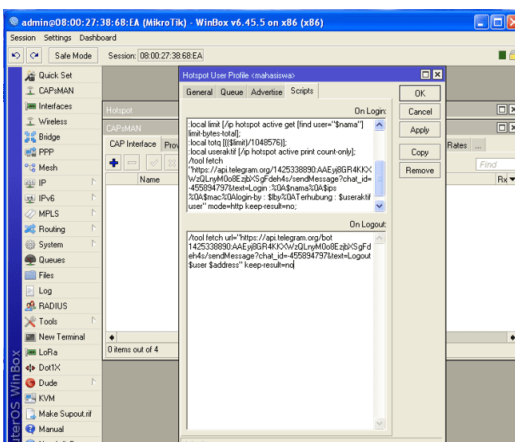


Gbr. 10 Langkah kelima membuat bot

F. Mikrotik Router

Mikrotik ialah router yang memiliki basis operasi jenis linux dan memiliki beberapa kemudahan dan beberapa fitur canggih didalamnya. Mikrotik dipakai sebagai sarana untuk membantu manusia dalam hal jaringan computer, yang memiliki beberapa fungsi seperti halnya ruting dinamis, hotspot, firewall, VPN, DHCP, DNS Cache, Web Proxy, dan lain-lain. Mikrotik menyajikan banyak kemudahan didalamnya, hal tersebutlah yang membuat mikrotik dijuluki sebagai surganya para admin. Mikrotik menyajikan beberapa varian produk seperti halnya router indoor, wireless router indoor/outdoor, embedded 2,4 GHz atau 5,x GHz, antena indoor/outdoor, dan lain sebagainya [4]. Terdapat beberapa fitur yang ada pada Mikrotik, sebagai contohnya adalah fitur *Script* dan *Scheduler*.

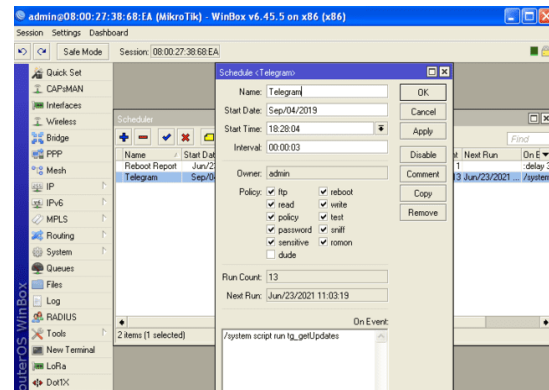
Script adalah jenis program yang tersusun dari himpunan instruksi untuk program aplikasi. Biasanya tersusun dari beberapa instruksi yang disimbolkan melalui utility dan syntax, yang pengembang kombinasikan dengan struktur kontrol yang sederhana. Pada penelitian ini, script digunakan untuk menempatkan instruksi dari Telegram bot dengan cara berkomunikasi menggunakan Token API dan Chat ID seperti pada gambar 11.



Gbr. 11 Penggunaan script

Pada dasarnya, *script* tersebut menggunakan *tool Fetch* yang ada di dalam mikrotik untuk mengakses ke halaman *web* tertentu. *tool fetch* digunakan pada penelitian ini untuk mengakses Telegram API dengan token, chatID dan isi pesan sesuai yang kita butuhkan. Bahkan *URL* ini juga dapat diakses pada *Browser* dengan pesan teks yang berbeda-beda.

Scheduler ialah service yang dipakai untuk membuat jadwal dan mengeksekusi suatu script setelah interval waktu tertentu[11]. Pada penelitian ini, *scheduler* digunakan untuk mengatur interval waktu dalam berkomunikasi dengan Telegram Bot seperti pada gambar 12 berikut.



Gbr. 12 Penggunaan scheduler

III. METODE PENELITIAN

Jenis penelitian ini merupakan pengembangan sistem yang memiliki tujuan supaya bias memperoleh hasil yang efektif dan efisien. Metode penelitian metode air terjun (*Waterfall method*) [12]. Seluruh program dan konfigurasi dijalankan secara virtual menggunakan GNS3.

A. Analisis kebutuhan

Peneliti dalam menentukan analisis kebutuhan menggunakan metode observasi. Nasution (1998) dalam Sugiyono memaparkan bahwasannya, observasi merupakan sumber dari segala pengetahuan yang ada. Hal tersebut dikarenakan pengetahuan yang bisa dipertanggung jawabkan merupakan pengetahuan yang diketahui secara langsung yakni melalui proses observasi. [13]. Observasi dilakukan di salah satu warung kopi dengan objek pelaku adalah administrator jaringan pada warung kopi tersebut. Dalam hal ini, administrator memerlukan suatu sistem yang dapat mendukung untuk melakukan manajemen user pada *hotspot* secara cepat dan bisa dilakukan secara jarak jauh.

Analisis kebutuhan sistem bisa divisualkan untuk menunjukkan spesifikasi yang disesuaikan dengan kebutuhan. Berikut hasil analisis spesifikasi kebutuhan pada table 1

TABEL 1.

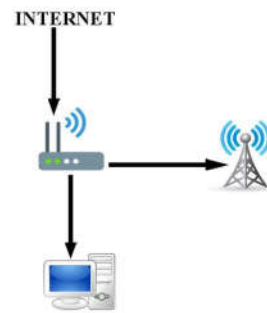
ANALISIS KEBUTUHAN

No	Deskripsi
1.	Sistem wajib untuk memberikan informasi apabila terdapat user hotspot yang telah terdaftar pada mikrotik sedang login/logout pada Admin atau Teknisi. Spesifikasi kebutuhan :

No	Deskripsi
	<ul style="list-style-type: none"> ● Sistem wajib memberikan informasi mengenai alamat ip dan nama user yang login pada jaringan hotspot. ● Sistem diharuskan menyajikan informasi alamat ip dan nama user yang logout pada jaringan hotspot.
2.	<p>Sistem wajib menyajikan informasi kuantitas user dan detail user yang sedang aktif kepada Admin atau Teknisi. Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> ● Sistem harus menyajikan informasi terkait jumlah user yang sedang aktif ● 2. Sistem harus menyediakan informasi detail user yang sedang aktif mulai dari username sampai uptime (tanpa menampilkan password user)
3.	<p>Sistem harus bisa melakukan pencabutan session user oleh Admin atau Teknisi. Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> ● Sistem harus melakukan pencabutan session perangkat berdasarkan username ● Sistem harus melakukan pencabutan session perangkat berdasarkan alamat IP ● Sistem harus melakukan pencabutan session perangkat berdasarkan MAC-Address
4.	<p>Sistem harus bisa mengaktifkan dan menonaktifkan seluruh fungsi hotspot melalui Admin atau Teknisi. Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> ● Sistem bisa mengaktifkan seluruh fungsi hotspot ● Sistem bisa menonaktifkan seluruh fungsi hotspot
5.	<p>Sistem harus bisa membuat dan menghapus user melalui Admin atau Teknisi. Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> ● Sistem bisa membuat user dengan username dan password ● Sistem bisa menghapus user berdasarkan username secara permanen
6.	<p>Sistem harus bisa mengaktifkan dan menonaktifkan user melalui Admin atau Teknisi. Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> ● Sistem bisa menonaktifkan user menggunakan username ● Sistem bisa mengaktifkan kembali user yang dinonaktifkan menggunakan username
7.	<p>Sistem harus bisa melakukan perubahan password user melalui Admin atau Teknisi. Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> ● Sistem bisa mengubah password user menggunakan username
8.	<p>Sistem harus bisa mengubah profil user melalui Admin atau teknisi. Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> ● Sistem bisa melakukan pemindahan profil user

B. Rancangan Topologi jaringan

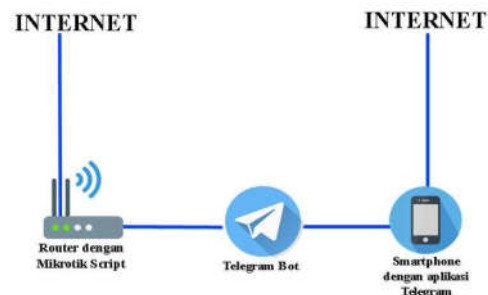
Di dalam penelitian ini, penulis menggunakan 3 komponen hardware. Komponen yang dipakai adalah 2 PC dan 1 router Mikrotik yang telah terhubung dengan internet. Peran sebagai admin adalah PC 1 yang terhubung dengan Mikrotik. Dan PC 2 memiliki peran untuk menjadi *client* dari router. Sedangkan untuk menghubungkan PC dan internet merupakan peran dari Router Mikrotik. Disisi lain, Router Mikrotik juga berfungsi sebagai hotspot dan juga menjadi tempat untuk berjalannya program yang akan di bangun, seperti pada gambar 13 :



Gbr. 13 Gambaran topologi jaringan

C. Desain Struktur Program

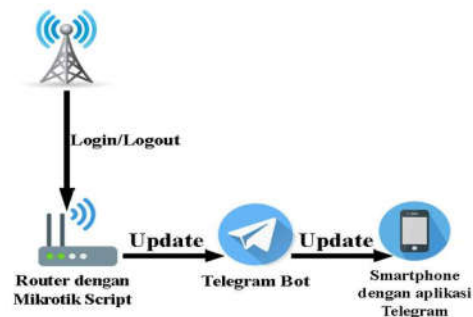
Peran sebagai jembatan penghubung antara smartphone dan Mikrotik diambil oleh telegram bot server. Sehingga, supaya smartphone mendapatkan notifikasi maka diharuskan smartphone tersebut terhubung dengan internet. Sama halnya pada bagian Router Mikrotik, untuk mengirim pesan notifikasi harus terhubung ke internet seperti pada Gambar 14 :



Gbr. 14 Desain dasar struktur program

D. Desain Struktur Bot Notifikasi user login dan logout

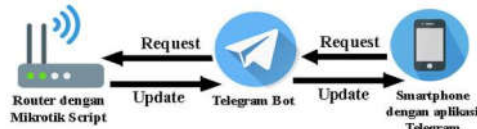
Telegram bot server sebagai jembatan penghubung antara mikrotik dan *smartphone* untuk mengirim notifikasi ketika ada *user hotspot* yang sudah terdaftar di Mikrotik meluan *login* ke jaringan dan juga ketika *user hotspot* *logout* dari jaringan *hotspot* seperti pada Gambar 15.



Gbr. 15 Desain struktur bot notifikasi

E. Desain Struktur Interaksi Mikrotik dengan Telegram Bot

Telegram bot server berfungsi untuk menjembatani antara *smartphone* dan Mikrotik. Request melalui Telegram Server dijalankan oleh *smartphone*. Sehingga telegram server bisa menjalankan script yang berada di dalam mikrotik kemudian bot server memberikan update dari hasil request tersebut seperti pada gambar 16.



Gbr. 16 Desain struktur interaksi mikrotik dengan Telegram Bot

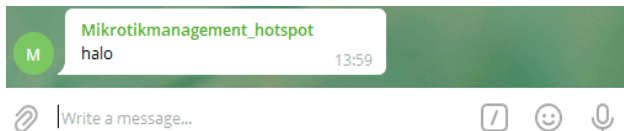
F. Menguji botAPI Telegram

Untuk menguji *botAPI* telegram apakah bisa digunakan untuk mengirim pesan yaitu dengan membuka *browser* kemudian masuk ke alamat : https://api.telegram.org/bot1425338890:AAEyj8GR4KKXWzQLNyM0o8EzjbXSgFdeh4s/sendmessage?chat_id=-455894797&text=halo, seperti pada gambar 17.



Gambar 17. Menguji BotAPI

Apabila berhasil mengirim pesan melalui bot *API*, maka akan menerima pesan yang sama dari bot telegram seperti pada gambar 18.



Gbr. 18 Hasil mengirim pesan dengan botapi

G. Script Notifikasi via Telegram Bot

Dalam pembuatan *script* untuk notifikasi ketika user *login*, *script* dipasang pada *user profile* dan di bagian *script*.

1). Script login

Pada tab *login*, penulis isi dengan *script* sebagai berikut :

```
{
:local username "$pengguna";
:local internet-service-provider [/ip hotspot
active get [find user="$username"] address];
:local media-access-control [/ip hotspot active get
[find user="$username"] mac-address];
:local lby [/ip hotspot active get [find
user="$username"] login-by];
:local batas [/ip hotspot active get [find
user="$username"] limit-bytes-total];
:local totalq [(($batas)/1048576)];
:local pengguna-aktif [/ip hotspot active print
count-only];
/tool fetch
"https://api.telegram.org/bot1425338890:AAEyj8GR4
```

```
KKXWzQLNyM0o8EzjbXSgFdeh4s/sendMessage?chat_id=-
455894797&text=Login
:%0A$username%0A$internet-service-provider
%0A$media-access-control%0Alogin-by :
$lby%0ATerhubung : $pengguna-aktif user" mode=http
keep-result=no;
}
```

Script tersebut mengambil data *username*, *IP* dan *Mac-address* dan kemudian dikirim ke administrator menggunakan *telegram Bot API*.

2). Script logout

Pada tab *logout*, penulis isi dengan *script* sebagai berikut :

```
/tool fetch
url="https://api.telegram.org/bot1425338890:AAEyj
8GR4KKXWzQLNyM0o8EzjbXSgFdeh4s/sendMessage?chat_i
d=-455894797&text=Logout $pengguna $alamat" keep-
result=no
```

Script tersebut hanya mengambil *username* dan *IP* yang kemudian dikirim ke *chat ID* administrator melalui bot *API* telegram.

H. Script Manajemen User

Dalam pembuatan *script* untuk manajemen *user*, terdapat unsur-unsur yang perlu untuk mendapat perhatian. Unsur-unsur tersebut antara lain :

1). Telegram config

Script ini berisi bot *API* dan *Chat ID* administrator. *Script* ini digunakan ketika *script* lain memanggil bot *API* dan *Chat ID* untuk melakukan eksekusi. Pada tab *config*, penulis isi dengan *script* sebagai berikut :

```
:log info "telegram configuration file has been
loaded";

# to use config insert next lines:
#:local fconfig [:parse [/system script get
tg_config source]]
#:local config [$fconfig]
#:put $config

#####
# Common parameters
#####

:local config {
"Command"="telegram";
"botAPI"="1425338890:AAEyj8GR4KKXWzQLNyM0o8E
zjbXSgFdeh4s";
"defaultChatID"="-455894797";
"trusted"="-455894797";
"storage"="";
"timeout"=5;
"refresh_active"=15;
"refresh_standby"=300;
}
return $config
```

Script tersebut menyimpan bot *API* telegram dan juga *Chat ID* administrator yang akan dipanggil oleh *script* yang lain.

2). botAPI getUpdates

Script ini berfungsi untuk melakukan *update* dan menerima pesan masuk dari :

<https://api.telegram.org/bot1425338890:AAEyj8GR4KKXWzQLnyM0o8EzjbXSgFdeh4s/getUpdates>
yang kemudian akan dilakukan pengambilan *command* dan parameter-parameter terkait. Pada tab ini, penulis isi dengan *script* sebagai berikut :

```
:global TGTERAKHIRIDPESAN
:global TGTERAKHIRIDUPDATE

:local pengaturanf [:parse [/system script get
tg_config source]]
:local http [:parse [/system script get
func_fetch source]]
:local dapatkey [:parse [/system script get
tg_getkey source]]
:local kirim [:parse [/system script get tg
kirimPesan source]]

:local config [$pengaturanf]
:local terpercaya [:toarray ($config->
"terpercaya")]
:local IDbot ($config->"botAPI")
:local penyimpanan ($config->"penyimpanan")
:local waktuhabis ($config->"waktuhabis")

:put "config=$config"
:put "terpercaya=$terpercaya"
:put "IDbot=$IDbot"
:put "penyimpanan=$penyimpanan"
:put "waktuhabis=$waktuhabis"

:local berkas ($penyimpanan."tg_get_updates.txt")
:local logberkas ($penyimpanan."tg_fetch_log.txt")
#get 1 Pesan per time
:local url ("https://api.telegram.org/bot".$IDbot."/getUpdates?waktuhabis=$waktuhabis&limit=1")
:if ([:len $TGTERAKHIRIDUPDATE]>0) do={
: set url "$url&offset=${$TGTERAKHIRIDUPDATE+1}"
}

:put "Membaca Update..."
:local res [$http dst-path=$berkas url=$url
resberkas=$logberkas]
:if ($res!="berhasil") do={
:put "Gagal mendapat update"
return "Gagal mendapat update"
}
:put "Selesai membaca update"

:local konten [/berkas get [/berkas find
name=$berkas] kontens]
```

```
:local idpesan [$dapatkey key="Pesan_id"
text=$konten]
:if ($idpesan="") do={
:put "No new updates"
:return 0
}
:set TGTERAKHIRIDPESAN $idpesan

:local updid [$dapatkey key="update_id"
text=$konten]
:set TGTERAKHIRIDUPDATE $updid

:local fromid [$dapatkey block="from" key="id"
text=$konten]
:local pengguna [$dapatkey block="from"
key="pengguna" text=$konten]
:local namaawal [$dapatkey block="from"
key="first_name" text=$konten]
:local namaakhir [$dapatkey block="from"
key="last_name" text=$konten]
:local idpesan [$dapatkey block="chat" key="id"
text=$konten]
:local textpesan [$dapatkey block="chat"
key="text" text=$konten]

:put "Pesan id=$idpesan"
:put "update id=$updid"
:put "from id=$fromid"
:put "first name=$namaawal"
:put "last name=$namaakhir"
:put "pengguna=$penggunaaname"
:local name "$namaawal $namaakhir"
:if ([:len $name]<2) do {
: set name $penggunaaname
}

:put "in chat=$idpesan"
:put "command=$textpesan"

:local izinkan ( [:type [:find $terpercaya
$fromid]]!="nil" or [:type [:find $terpercaya
$idpesan]]!="nil")
:if (!$izinkan) do={
:put "Pengirim tidak diketahui tetap diam"
:return -1
}

:local cmd ""
:local parameter ""
:local ltext [:len $textpesan]

:local pos [:find $textpesan " "]
:if ([:type $pos]="nil") do={
: set cmd [:pick $textpesan 1 $ltext]
} else={
: set cmd [:pick $textpesan 1 $pos]
: set parameter [:pick $textpesan ($pos+1)
$ltext]
```

```

}

:local pos [:find $cmd "@"]
:if ([:type $pos]!="nil") do={
:set cmd [:pick $cmd 0 $pos]
}

:put "cmd=<$cmd>"
:put "parameter=<$parameter>"

:global TGLASTCMD $cmd

:put "Coba aktifkan skrip eksternal tg_cmd_$cmd"
:local script [:parse [/system script get
"tg_cmd_$cmd" source]]
$script parameter=$parameter idpesan=$idpesan
from=$name

```

Script tersebut digunakan untuk mengecek terus menerus sesuai interval yang sudah ditentukan dan membaca pesan yang diterima oleh *botAPI* yang kemudian mengambil *command* dan parameter perintah, seperti *username* yang akan di eksekusi oleh *script* lain.

3) Command manajemen hotspot

Pada *script* ini berisi perintah dan variabel yang digunakan untuk mengeksekusi perintah dari *botAPI*. *Script* ini juga mengambil parameter yang didapat dari *getUpdates* kemudian dieksekusi pada mikrotik. Pada tab ini, penulis isi *script* seperti berikut:

```

:local kirim [:parse [/system script get
tg kirimPesan source]]
:local parameter1 [:pick $parameter 0
[:find $parameter " "]]
:local parameter2 [:pick $parameter
[:find $parameter " "]+1] [:len $parameter]]
:local parameter3 [:pick [:pick
$parameter ([:find $parameter " "]+1) [:len
$parameter]] ([:find [:pick $parameter
[:find $parameter " "]+1) [:len $parameter]]
" "]+1) [:len [:pick $parameter ([:find
$parameter " "]+1) [:len $parameter]]]]]
:if ([:len [:find $parameter2 " "]]>0)
do={
:set parameter2 [:pick [:pick
$parameter ([:find $parameter " "]+1) [:len
$parameter]] 0 [:find [:pick $parameter
[:find $parameter " "]+1) [:len $parameter]]
" "]]
} else={:set parameter3 ""}

:put $parameter
:put $parameter1
:put $parameter2
:put $parameter3
:put $idpesan
:put $from

:if ($parameter1="session") do={
:if ($parameter2="count") do={
:local keluaran
:local hotspot [:len [/ip hotspot active
find]]

```

```

:local text "Router ID:* $[/system
identity get name] * %0A\
Hotspot users: $hotspot online"

:kirim chat=$idpesan text=$text
mode="Markdown"
}
:if ($parameter2="showall") do={
:local keluaran
:foreach activeIndex in=[/ip hotspot
active find] do={
:local Useraktif ("pengguna: ".[/ip
hotspot active get value-name="user"
$activeIndex]."%0A")
:local Addressaktif ("IP: ".[/ip hotspot
active get value-name="address"
$activeIndex]."%0A")
:local MACAddraktif ("MAC: ".[/ip hotspot
active get value-name="mac-address"
$activeIndex]."%0A")
:local LoginByactive ("Login Method:
".[/ip hotspot active get value-name="login-
by" $activeIndex]."%0A")
:local Uptimeactive ("Uptime: ".[/ip
hotspot active get value-name="uptime"
$activeIndex]."%0A")
:local timeidle ("Idle Time: ".[/ip
hotspot active get value-name="idle-time"
$activeIndex]."%0A")
:local Inserver ("Server: ".[/ip hotspot
active get value-name="server"
$activeIndex]."%0A")
:set keluaran
($keluaran.$Useraktif.$Addressaktif.$MACAddr
aktif.$Uptimeactive.$timeidle.$LoginByactive.$
Inserver."%0A")
}
:kirim chat=$idpesan text=("$keluaran")
mode="Markdown"
}
:if ($parameter2="deauth-by-user") do={
/ip hotspot active remove [find
user="$parameter3"]
:kirim chat=$idpesan text=("Sesi User
$parameter3 berhasil dihapus")
mode="Markdown"
}
:if ($parameter2="deauth-by-mac") do={
/ip hotspot active remove [find mac-
address="$parameter3"]
:kirim chat=$idpesan text=("Sesi MAC
$parameter3 berhasil dihapus")
mode="Markdown"
}
:if ($parameter2="deauth-by-ip") do={
/ip hotspot active remove [find
address="$parameter3"]
:kirim chat=$idpesan text=("Sesi IP
$parameter3 berhasil dihapus")
mode="Markdown"
}
}
:if ($parameter1="add") do={
/ip hotspot user add name=$parameter2
password=$parameter3 proberkas=default
:kirim chat=$idpesan text=("Berhasil
membuat user baru. Masuk ke hotspot
dengan:%0A%0A*pengguna:*
$parameter2%0A*Password:* $parameter3")
mode="Markdown"
}
:if ($parameter1="delete") do={

```



```

/ip hotspot user remove [find
name=$parameter2]
  kirim chat=$idpesan text=("Berhasil
menghapus user $parameter2.") mode="Markdown"
}
:if ($parameter1="disable") do={
/ip hotspot user disable [find
name=$parameter2]
  kirim chat=$idpesan text=("$parameter2
kini telah dinonaktifkan") mode="Markdown"
}
:if ($parameter1="enable") do={
/ip hotspot user enable [find
name=$parameter2]
  kirim chat=$idpesan text=("$parameter2
kini telah diaktifkan") mode="Markdown"
}
:if ($parameter1="setproberkas") do={
/ip hotspot user set
proberkas=$parameter3 [find name=$parameter2]
/ip hotspot active remove [find
name=$parameter2]
  kirim chat=$idpesan text=("Berhasil
mengganti proberkas menjadi $parameter3.")
mode="Markdown"
}
:if ($parameter1="change-password") do={
/ip hotspot user set password=$parameter3
[find name=$parameter2]
/ip hotspot active remove [find
name=$parameter2]
  kirim chat=$idpesan text=("Berhasil
mengganti password untuk $parameter2.")
mode="Markdown"
}
  kirim chat=$idpesan text=("$keluaran")
mode="Markdown"
}
}

```

Pada *script* tersebut, untuk melakukan eksekusi mengambil parameter yang diterima oleh bot *API*. salah satu contohnya ketika menambah *user* baru parameter pertama adalah *add*, kemudian parameter kedua merupakan *username* dan parameter ketiga adalah *password*.

4) sendMessage telegram

Script ini digunakan untuk mengirim pemberitahuan kepada administrator bahwa perintah yang diminta untuk Mikrotik sudah berhasil dijalankan. Pada tab ini, penulis isi dengan *script* sebagai berikut :

```

:local pengaturanf [:parse [/system
script get tg_config source]]

:local config [$pengaturanf]
:local IDpesan ($config-
>"defaultIDpesan")
:local IDbot ($config->"botAPI")
:local penyimpanan ($config-
>"penyimpanan")

:if ([:len $chat]>0) do={:set IDpesan
$chat}

:local url
"https://api.telegram.org/bot$IDbot
kirimPesan?chat_id=$IDpesan&text=$text"
:if ([:len $mode]>0) do={:set url
($url."&parse_mode=$mode")
}

```

```

:local berkas
($tgpenyimpanan."tg_get_updates.txt")
:local logberkas
($tgpenyimpanan."tg_fetch_log.txt")

/tool fetch url=$url keep-result=no

```

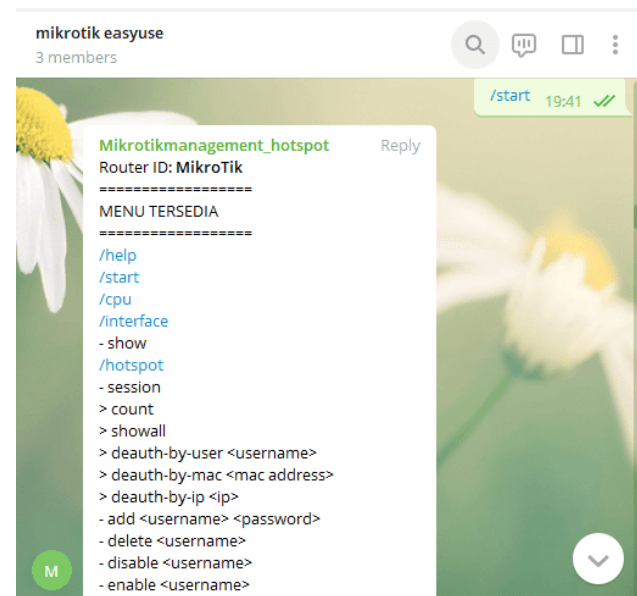
Pada *script* tersebut, mikrotik akan mengirim pesan bahwa perintah yang diminta oleh administrator berhasil di jalankan.

IV. HASIL DAN PEMBAHASAN

A. Pengujian Menjalankan Perintah

1). Perintah /Start

Apabila *API Token* dan juga *UserID* Administrator telah dimasukkan dengan benar, maka ketika mengetikkan */start* ataupun akan muncul perintah-perintah untuk dilanjutkan administrator dapat melakukan manajemen *user*. Hasil ketika melakukan pengiriman */start* kepada *Telegram Bot*, ialah seperti pada gambar 19.

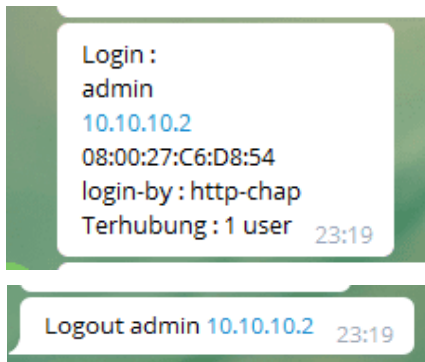


Gbr. 19 Menjalankan perintah /start

Untuk selanjutnya pengiriman perintah harus diawali dengan tanda "/" (Garis Miring)

2) Bot Monitoring User Hotspot

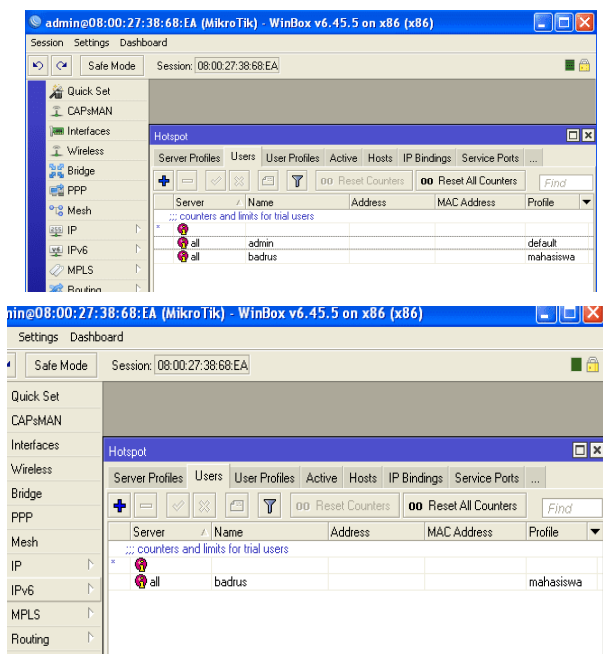
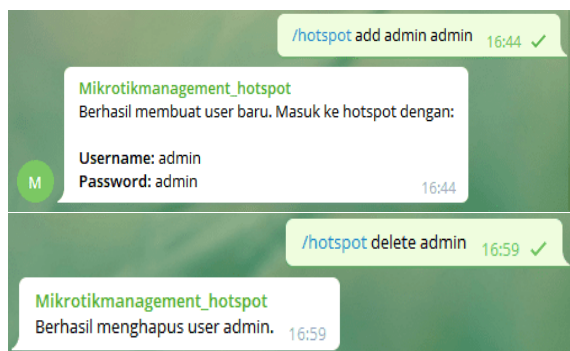
Dalam implementasinya, pembuatan bot monitoring ini hanya memasukkan *Script* yang sudah disusun kedalam mikrotik pada bagian *user profile*. Bila terdapat *user hotspot* yang telah mendaftarkan diri maka bisa langsung *login*, *bot* akan secara langsung menjalankan dan dan memberi notifikasi kepada administrator yang isinya ialah pesan nama *user* dan *ip address* menjalankan *login*. Apabila ada *user hotspot logout*, *bot* akan otomatis berjalan dan mengirimkan notifikasi kepada administrator yang berisikan pesan nama *user* dan *ip address* menjalankan *logout* seperti yang ada di Gambar 20.



Gbr. 20 Bot monitoring user hotspot

3) Bot Menambah dan Menghapus User

Bot menambah dan menghapus user hotspot oleh administrator mempunyai beberapa command seperti pada gambar 21.

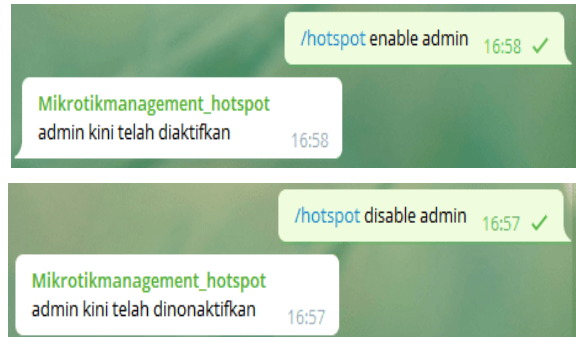


Gbr. 21 Bot menambah dan menghapus user

- /hotspot add ini merupakan perintah membuat user baru
- /hotspot delete ini merupakan perintah menghapus user baru

4). Bot Mengaktifkan dan Nonaktifkan User

Bot untuk mengaktifkan dan nonaktifkan user oleh administrator memiliki beberapa command sesuai gambar 22.

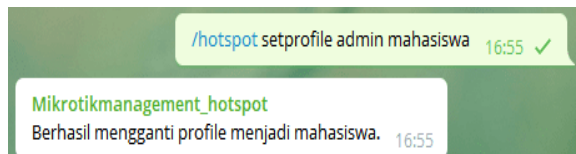


Gbr. 22 Bot mengaktifkan dan nonaktifkan user

- /hotspot enable ini merupakan perintah untuk mengaktifkan user sesuai username
- /hotspot disable ini merupakan perintah untuk menonaktifkan user sesuai username

5). Memindah User Profile User

Bot untuk memindahkan user profil yang sudah disediakan sebelumnya oleh administrator dengan perbedaan kecepatan upload dan download data. Ada beberapa command sesuai pada gambar 23.

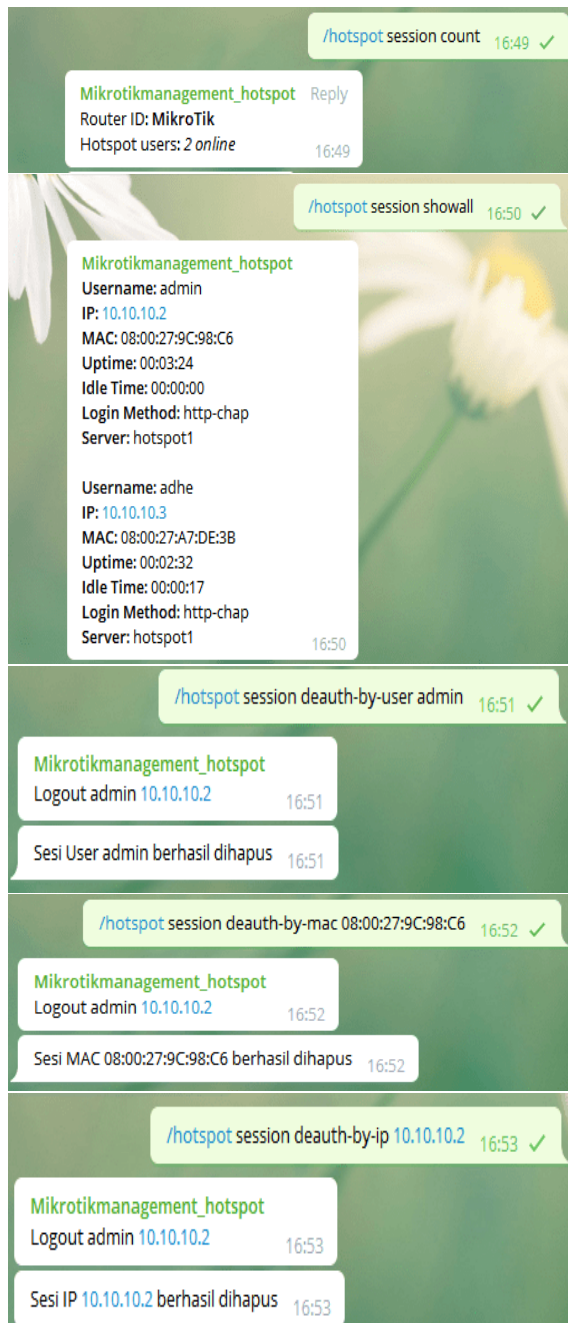


Gbr. 23 Bot memindahkan user profile

- /hotspot setprofile ini merupakan perintah untuk memindahkan user ke profil lain yang telah disediakan dengan pembatasan upload dan download

6). Bot Monitoring dan Pencabutan Session User

Bot untuk monitoring berapa user yang sedang online dan melakukan pencabutan session oleh administrator. Ada beberapa command seperti pada gambar 24.



Gbr. 24 Hasil *command Bot Monitoring* dan *Pencabutan Session User*

- */hotspot session count* ini merupakan perintah untuk menampilkan jumlah *user* yang sedang aktif
- */hotspot session showall* ini merupakan perintah untuk menampilkan detail seluruh *user* yang terhubung tanpa menampilkan *password*
- */hotspot session deauth-by-user* ini merupakan perintah untuk mencabut *session* user berdasarkan *username*
- */hotspot session deauth-by-mac* ini merupakan perintah untuk mencabut *session* user berdasarkan *Mac-Address*

- */hotspot session deauth-by-ip* ini merupakan perintah untuk mencabut *session* user berdasarkan *IP address*.

B. Pengujian Konfigurasi

1). Kondisi Pengujian

Pengujian dilakukan dengan mempertimbangkan kondisi peralatan yang dipakai supaya mendapatkan hasil yang maksimal. Berikut peralatan yang dipakai ialah seperti pada tabel 2.

TABEL 2

TABEL KONDISI PERANGKAT YANG DIPAKAI

Perangkat	Keterangan
GNS3	2.1.19
VirtualBox	6.1.12
Virtual Windows	Windows XP
Mikrotik	6.33
Winbox	6.45.5 en x86
Telegram	Telegram Desktop 2.9.2
Jaringan	WiFi

2). Langkah Konfigurasi

Peneliti melakukan pengujian untuk memperoleh jumlah langkah yang diperlukan untuk menjalankan masing-masing perintah pada Winbox maupun *telegram*. Untuk pengujian *telegram bot*, dimulai ketika peneliti melakukan pengetikan perintah. Sedangkan pada Winbox dimulai ketika peneliti melakukan *login* ke dalam *router* Mikrotik terlebih dahulu. Hasil pengujiannya ialah seperti pada tabel 3.

TABEL 3

TABEL PERBANDINGAN LANGKAH KONFIGURASI

Winbox	Telegram Bot	Keterangan
4 langkah	2 langkah	Membuat user
4 langkah	2 langkah	Menghapus user
4 langkah	2 langkah	Mengubah password user
4 langkah	2 langkah	Mengubah Profil
4 langkah	2 langkah	Mengaktifkan user
4 langkah	2 langkah	Menonaktifkan user
5 langkah	3 langkah	Mencabut IP session user

Dari table 3 di atas diperoleh informasi bahwa langkah konfigurasi menggunakan Telegram lebih pendek daripada menggunakan Winbox sehingga penggunaan Telegram langkahnya lebih efektif.

3) Waktu Konfigurasi

Peneliti melakukan pengujian untuk memperoleh jumlah waktu yang dibutuhkan untuk melakukan konfigurasi pada

masing-masing perintah. Dalam pengujian ini, peneliti menggunakan *stopwatch* untuk mengukur waktu. Hal lain yang diperhatikan dalam melakukan pengujian ini adalah peneliti melakukan 3 kali pengujian untuk mengambil rata-rata waktu yang dibutuhkan dan menggunakan 1(satu) buah *router* Mikrotik. Alasan peneliti menggunakan 1(satu) buah *router* adalah karena 1(satu) Telegram Bot hanya dapat digunakan untuk konfigurasi 1(satu) *router* saja. Penghitungan waktu pada *telegram bot* dimulai saat peneliti mulai mengetik *command* pada *telegram bot*. Sedangkan pada Winbox, mulai dihitung saat peneliti *login* kedalam *router* Mikrotik ialah seperti pada tabel 4.

TABEL 4

TABEL PERBANDINGAN WAKTU KONFIGURASI

Perintah	Winbox	Telegram Bot
Menambah user	28 detik	12 detik
Menghapus user	12 detik	8 detik
Mengubah password user	18 detik	13 detik
Mengubah profile user	21 detik	10 detik
Menonaktifkan user	16 detik	8 detik
Mengaktifkan user	20 detik	8 detik
Mencabut session user	23 detik	16 detik

Dari table 4 di atas diperoleh informasi bahwa waktu konfigurasi menggunakan Telegram lebih cepat daripada menggunakan Winbox sehingga penggunaan Telegram lebih efisien waktu.

V. KESIMPULAN

Berdasarkan hasil uji untuk melakukan perintah manajemen user oleh administrator pada *Telegram Bot* adalah sebagai berikut:

- 1) Berdasarkan hasil pengujian menjalankan perintah administrasi dan monitoring user hotspot dapat dilakukan cukup mudah oleh administrator jaringan menggunakan chatbot Telegram.
- 2) Berdasarkan Pengujian Langkah dan Waktu Konfigurasi diperoleh bahwa penggunaan Telegram lebih efektif dan Efisien jika dibandingkan menggunakan Winbox.

UCAPAN TERIMA KASIH

Dengan selesainya penelitian ini penulis panjatkan Puji syukur kepada Allah SWT yang telah memberi rahmat dan hidayah-Nya sehingga penulis bisa menyelesaikan penelitian dan menulis artikel ilmiah dengan baik. Tidak lupa juga penulis mengucapkan terimakasih kepada dirinya sendiri yang telah berusaha sampai saat ini, kedua orang tua serta adik saya yang selalu memberikan do'a dan dukungan, dosen pembimbing, teman dan semua pihak yang telah mendukung, membantu, serta membimbing penulis dalam mengerjakan dan menyelesaikan penelitian ini.

REFERENSI

- [1] "Telegram Bot API." <https://core.telegram.org/Bots/api> (accessed Jun. 16, 2021).
- [2] "Telegram FAQ." <https://telegram.org/faq#q-what-is-telegram-what-do-i-do-her> (accessed Jun. 16, 2021).
- [3] T. FARIDA, "Implementasi Notifikasi Dengan Sms Pada the Dude Network Monitoring (Implementasi Notifikasi Dengan Sms Pada the Dude Network Monitoring)," *J. Manaj. Inform.*, vol. 5, no. 2, pp. 42–49, 2016.
- [4] D. K. Hakim and S. A. Nugroho, "Implementasi Telegram Bot untuk Monitoring Mikrotik Router," *Sainteks*, vol. 16, no. 2, pp. 151–157, 2020, doi: 10.30595/st.v16i2.7132.
- [5] F. Fifit, "Penggunaan Telegram Sebagai Media Komunikasi Dalam Pembelajaran Online," *Cakrawala-Jurnal Hum.*, vol. 20, no. 2, p. 113, 2020, [Online]. Available: <https://ejournal.bsi.ac.id/ejurnal/index.php/cakrawala/article/view/8935>.
- [6] R. D. Risanty and A. Sopiyan, "Pembuatan Aplikasi Kuesioner Evaluasi Belajar Mengajar Menggunakan Bot Telegram Pada Fakultas Teknik Universitas Muhammadiyah Jakarta (Ft-Umj) Dengan Metode Polling," *Semin. Nas. Sains dan Teknol.*, no. November, pp. 1–9, 2017, [Online]. Available: <https://jurnal.umj.ac.id/index.php/semnastek/article/view/2071/1712>.
- [7] M. A. A. Widya and P. Airlangga, "Pengembangan Telegram Bot Engine Menggunakan Metode Webhook Dalam Rangka Peningkatan Waktu Layanan E-Government," *Saintekbu*, vol. 12, no. 2, pp. 13–22, 2020, [Online]. Available: <http://ejournal.unwaha.ac.id/index.php/saintek/article/view/884>.
- [8] B. A. B. Ii, "2 bab ii landasan teori 2.1," pp. 10–44, 2017.
- [9] H. Soeroso *et al.*, "Penggunaan Bot Telegram Sebagai Announcement System pada Intansi Pendidikan," *Semin. MASTER 2017 PPNs*, vol. 1509, pp. 45–48, 2017.
- [10] A. D. Mulyanto, "Pemanfaatan Bot Telegram Untuk Media Informasi Penelitian," *Matics*, vol. 12, no. 1, p. 49, 2020, doi: 10.18860/mat.v12i1.8847.
- [11] M. S. Natsir and S. Alam, "Generate User Menggunakan Script Berbasis SMS Gateway Pada Hotspot Mikrotik."
- [12] P. S. Ganney, S. Pisharody, and E. Claridge, *Software Engineering*. 2013.
- [13] "UNTUK MEMENUHI TUGAS MATAKULIAH Metode Penelitian Pengembangan Yang dibina oleh Bapak Prof. Dr. M. E. Winarno, M. Pd Awal Akbar Jamaluddin PROGRAM PASCASARJANA PROGRAM STUDI PENDIDIKAN OLAHRAGA FEBRUARI 2017 ABSTRAK," 2017.