

Pengembangan Algoritma *auto mitigasi channel* untuk Mitigasi Interferensi menggunakan *Raspberry Pi* sebagai *controller* pada link *point to point WiFi IEEE 802.11n*.

Syafrizal¹, Teuku Yuliar Arif², Ramzi Adriman³

Magister Teknik Elektro, Universitas Syiah Kuala
Jl.T.Nyak Arief Kampus Darussalam Banda Aceh Provinsi Aceh

1syafrizal@unsyiah.ac.id

2yuliar@unsyiah.ac.id

3ramzi.adriman@unsyiah.ac.id

Abstrak— IEEE 802.11 (*Institute of Electrical and Electronics Engineers*) telah mengembangkan beberapa standard yang dikenal dengan 802.11 a/b/g/n, penggunaan standard wlan adanya control dapat menyebabkan *interference channel* ini permasalahan utama dalam teknologi *wireless*, beberapa *wireless* tipe n telah memiliki pilihan *autochannel* untuk memberikan opsi *channel* secara *auto*, permasalahannya adalah pada saat terjadi *interference* fungsi *autochannel* pada wlan tersebut tidak me mitigasi *channel* ke lebih baik, ini mempengaruhi *throughput* pada *client*. Dalam penelitian ini akan dikembangkan algoritma baru *auto mitigasi channel* untuk proses mitigasi ke *channel* yang lebih aman dari *interference*, hasil yang di harapkan dari penelitian ini dapat memberikan kontribusi dari algoritma sebelumnya untuk mitigasi *channel* secara *auto* jika mengalami *interference*, dengan mengembangkan modul API (*Application Programing Interface*) pada raspberry pi terhadap wireless mikrotik sebagai *access point* untuk melakukan *scanning channel* interferensi berdasarkan parameter *channel* dan *power transmit*, maka akan dilakukan proses mitigasi ke *channel* yang lain oleh *access point*.

Kata Kunci— IEEE 802.11, *interferensi*, *Raspberry Pi*, *API*, *wlan*

I. Latar Belakang

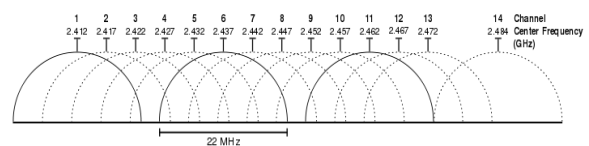
Perkembangan teknologi *wireless* berdasarkan keputusan IEEE 802.11 dengan standard yang dikenal 802.11 a/b/g/n/ac dengan berbagai macam fitur yang mutakhir, terakhir di kembangkan *autochannel* untuk mempermudah dalam konfigurasi *wireless*, kendala yang di hadapi adalah terlalu banyak *wireless* yang berekerja di frekwensi yang sama (*interferensi*), dengan pengembangan algoritma *auto mitigasi interferensi* yang di kembangkan oleh peneliti merupakan salah satu opsi untuk mengatasi permasalahan yaitu dengan pemanfaatan teknologi IEEE 802.11.n yang memiliki spesifikasi yang handal kita dapat memodifikasi dengan penambahan *interface* sebagai media *scanning* yang akan melakukan *scanning* terhadap gangguan *interference*, penyebab utama *interference* adalah karena ketersediaan *channel* yang terbatas sehingga kemungkinan terjadinya tumbukan antar *channel* akan semakin besar perlu penanganan

(mitigasi) secara *auto* untuk mengetahui *channel* yang *transmit power* rendah diantara *access point* dan *client* sehingga dapat menjaga ke stabilan *throughput* perangkat tersebut, untuk itu akan di kembangkan algoritma *auto mitigation channel*.

II. Landasan Teori

II.1 Wlan (Wireless lan)

Berdasarkan keputusan IEEE.802.11 mengenai pengaturan *Medium Access Control (MAC)* dan *Physical Layer (PHY)*, di dapatkan pengaturan *channel* yang berkerja pada fekwensi 2,4 GHz, dengan detail gambar sebagai berikut :



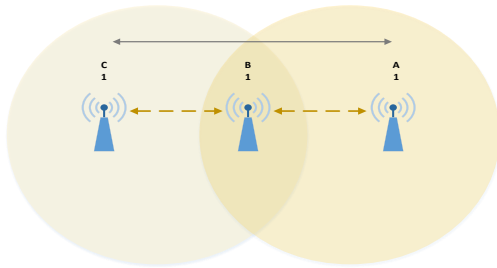
Gambar 1. Standar Wlan Channel

Perkembangan dari standard *channel* yang tersedia untuk memungkinkan lebih banyak nya *carrier* dari *channel* yang tersedia adalah dengan teknologi *multiplexing*, atau yang di kenal dengan OFDM. OFDM merupakan salah satu teknik *multiplexing* yang menggunakan sejumlah besar narrow-band yang paralel dari sub-carrier. Pada teknik ini, sejumlah besar sub-carrier secara orthogonal dengan jarak yang sempit digunakan untuk membawa data. Data dibagi menjadi beberapa berkas data/channel yang paralel, satu untuk setiap sub-carrier. Tiap subcarrier dimodulasikan dengan teknik modulasi tertentu (BPSK, QPKS, 16-QAM atau 64-QAM) pada symbol rate yang rendah

a. Interferensi

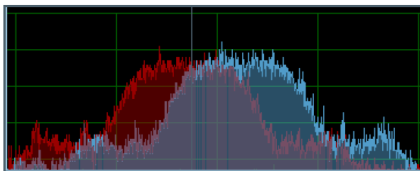
Interferensi merupakan sebuah gangguan yang terjadi akibat korespondensi antara pengirim dan penerima, setiap pengirim hanya berkomunikasi dengan penerima yang sesuai dan setiap penerima hanya menerima setiap paket yang membutuhkannya sesuai dengan otorisasi. Skala gangguan

yang diakibatkan oleh media transmisi antara penerima dan pengirim di analogikan seperti, *link* utama adalah $N = N(N-1)$ link penerima. Hal ini dapat di kurangi dengan mengurangi *transmission power* pada masing-masing perangkat, akan tetapi memiliki konsekuensi yang mengakibatkan terputusnya koneksi dari satu *node* ke lainnya [5].



Gambar 2. Interferensi antar perangkat

Interferensi terjadi akibat adanya *Frequency Reuse*, adalah penggunaan ulang sebuah frekwensi pada suatu sel, dimana frekwensi tersebut sebelumnya sudah digunakan pada satu atau beberapa sel lainnya. Jarak antara 2 sel yang menggunakan frekwensi yang sama ini harus diatur sedemikain rupa sehingga tidak akan mengakibatkan interferensi. Frequency reuse dilakukan untuk meningkatkan efisiensi alokasi frekwensi dan meningkatkan kapasitas sistem [5]. Jika menggunakan Wifi analyzer maka akan di perlihatkan kualitas signal antar *channel* yang saling bertubrukan (interferensi).



Gambar3 . Deteksi menggunakan wifi analyzer

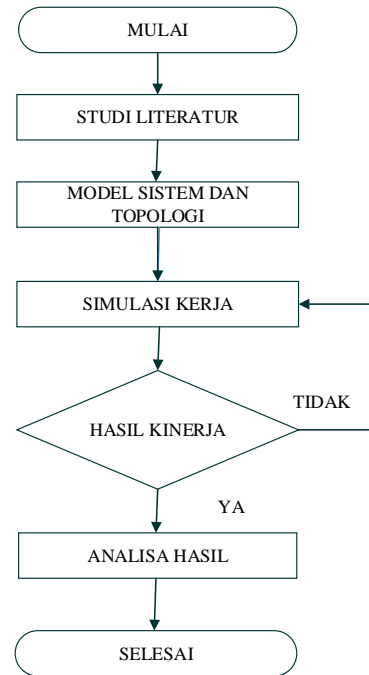
b. Raspberry Pi

Raspberry Pi merupakan mini pc dengan berbagai fitur yang dapat di *embedded* menggunakan *API (Application Programming Interface)*, *Raspberry Pi* memiliki beberapa berbagai fitur, yaitu micro SD yang berfungsi sebagai harddisk, port usb, port. Ethernet, audio output, RCA video, HDMI Video, CPU 400-700 MHz, dan yang paling penting adalah *Raspberry Pi* memiliki pin GPIO yang berfungsi untuk *interface* dengan berbagai perangkat elektronik. Bahasa yang digunakan dalam pengontrolan adalah. bahasa Python. Menurut Dr. Andrew N Harrington (2009), bahasa python adalah bahasa pemrograman yang memiliki banyak fungsi, interaktif, berorientasi objek dan merupakan bahasa pemrograman tingkat tinggi.

III. Metode Penelitian

a. Objek dan Tahapan Penelitian

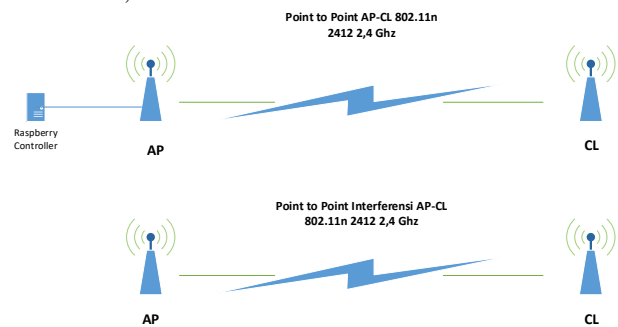
Dalam penelitian ini fokus hal yang di teliti adalah pilihan *autochannel* dan pengaruh pada saat sebelum terjadi interferensi dan saat terjadi interferensi dari perangkat mikrotik *point to point* IEEE 802.11n, serta proses mitigasi yang dikembangkan melalui API (*Aplication Programing Interface*) yang di jalankan melalui raspberry pi. Berikut diagram tahapan penelitian yang dilakukan :



Gambar 4. Diagram Alir Tahapan Penelitian

• Topologi Penelitian

Melakukan perancangan model sistem algoritma *auto* mitigasi *channel* yang yang disebabkan oleh interferensi *channel* dengan topologi *point to point* WLAN IEEE 802.11n frekwensi 2,4 Ghz



Gambar 5. Topologi Penelitian

• Diagram Penelitian

Menganalisa hasil kinerja algoritma baru dengan proses mitigasi *channel* dengan parameter *channel idle* menghasilkan *throughput* yang lebih baik dibandingkan dengan interferensi *autochannel* yang tersedia di perangkat, diagram alur analisis hasil dengan proses *auto* mitigasi adalah sebagai berikut :

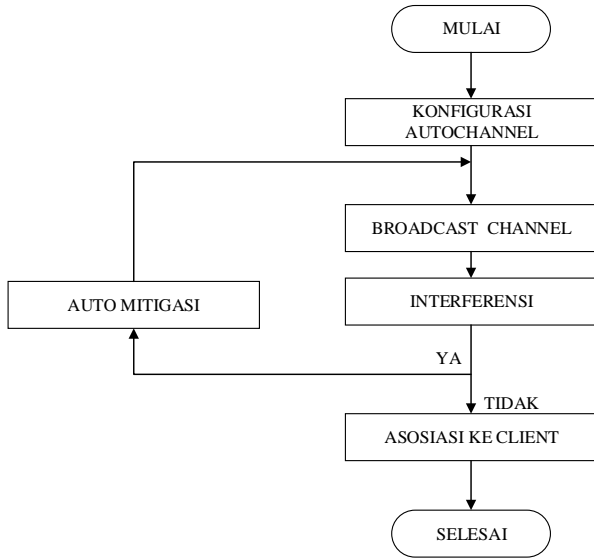


Figure 6 Diagram Alur *auto* mitigasi channel

Pada penelitian ini algoritma baru yang di adaptasi dari algoritma *autochannel* untuk mengurangi *delay* saat proses *auto* mitigasi dengan model *scripting* dan *listing channel* melalui interface yang mengecek interferensi. Pada *flow chart* diatas interferensi terjadi dengan karakteristik *power transmit*, frekwansi, maka akan di *trigger* oleh interface yang bertugas mendeteksi interferensi berdasarkan karakteristik tersebut maka akan di lakukan proses *auto* mitigasi ke signal yang terbaik.

3.2 Peralatan Yang Digunakan

Adapun perangkat/peralatan yang di gunakan dalam penelitian ini adalah sebagai berikut :

Nama Perangkat	Jumlah
Personal Computer	3
Access Point	3
Raspberry pi	1

3.3 Parameter Pengujian

Pengujian dilakukan dengan jarak yang dekat dan terdapat beberapa *access point* lainnya yang sedang aktif dengan beberapa skenario pengujian dengan penyajian *throughput* yang baik dalam waktu yang singkat.

- Pengujian interferensi dan non interferensi *Auto Channel*

Pengujian	<i>Auto Channel</i>		
	2462	2422	2412
Interferensi	V	V	
Non Interferensi			V

- Pengujian statis *channel*

Pengujian	<i>Statis Channel</i>	
	2462	2437
Interferensi	V	V
Non Interferensi	V	V

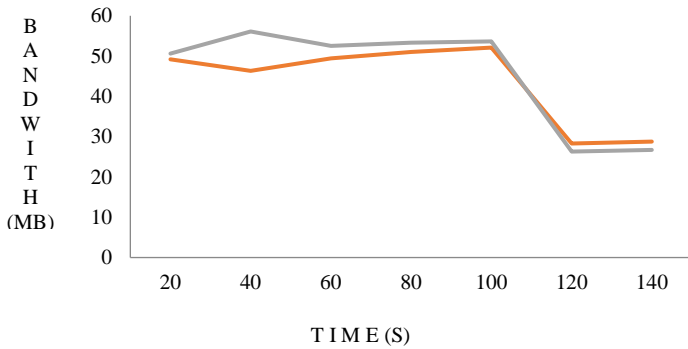
- Pengujian *auto* mitigasi *channel*

Pengujian	<i>Auto Mitigasi Channel</i>				
	2412	2427	2432	2437	2462
1. Interferensi	V				
2. Auto Mitigasi		V			
3. Interferensi				V	
4. Auto Mitigasi			V		
5. Interferensi					V
6. Auto Mitigasi		V			V

IV. Hasil

a. Pengujian *Autochannel* Interferensi

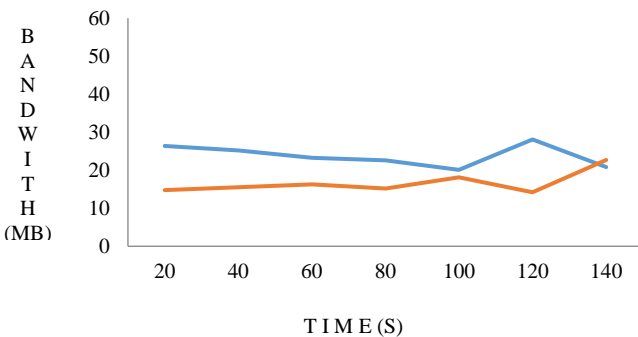
Pengujian *autochannel* di lakukan sebagai perbandingan secara grafik untuk mengetahui kualitas dari *signal* dan pengaruh interferensi yang di dihasilkan dari konfigurasi *autochannel* perangkat tersebut.



b. Pengujian Static Channel 11 Interferensi

Pengujian Statis *Channel* interferensi di lakukan untuk mengetahui *throughput* yang di hasilkan dengankonfigurasi *channel* secara *manual* (statis)

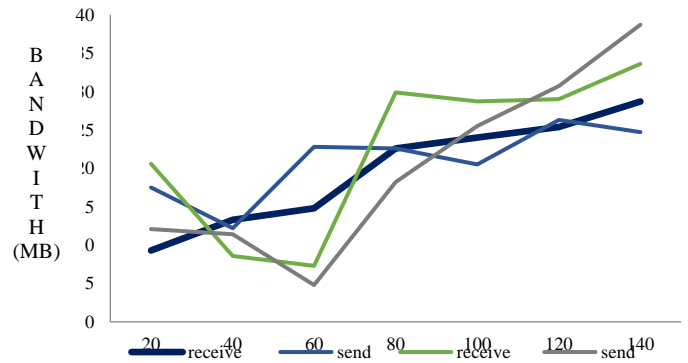
No	time (s)	Troughput (Mbps)		CCQ Tx/Rx Client (%)	Frekwensi 2.4 Ghz (Channel)
		receive	send		
1	20	26.4	14.8	86/84	2462
2	40	25.2	15.5		
3	60	23.3	16.3		
4	80	22.6	15.2		
5	100	20.1	18.1		
6	120	28.1	14.2		
7	140	20.8	22.7		



c. Pengujian Auto Mitigasi Channel

Pengujian ini di lakukan untuk membuktikan pengaruh interferensi *channel* yang memepengaruhi *throughput* dan proses mitigasi secara *auto* jika terjadi interferensi pada *channel* yang di gunakan sebelumnya, hasilnya beberapa saat setelah mitigasi *throughput* yang di hasilkan semakin meningkat.

No	time (s)	Troughput (Mbps)				Frekwensi 2.4 Ghz
		Autochannel Interferensi (2412)		Auto Mitigasi Channel (2412 ke 2427)		
		receive	send	receive	send	
1	20	9.3	17.5	20.6	12.1	2412, 2427
2	40	13.3	12.2	8.6	11.4	
3	60	14.8	22.8	7.3	4.8	
4	80	22.6	22.6	29.9	18.2	
5	100	24	20.5	28.7	25.5	
6	120	25.4	26.3	29	30.7	
7	140	28.7	24.7	33.6	38.7	



V . Kesimpulan dan Saran

5.1. Kesimpulan

Kesimpulan dari penelitian di harapkan dapat memberikan solusi dari permasalahan *loses data* yang disebabkan oleh *interferensi channel* ditambah dengan koneksi yang berjauhan melalui *point to point wireless*, sehingga dengan penerapan algoritma *auto mitigasi channel* dengan media raspberry pi yang dapat mengendalikan akses dari *client* memungkinkan untuk dapat mengontrol interferensii *channel* yang menyebabkan gangguan yang berpengaruh pada *channel* dan *throughput send* dan *receive* dari perangkat baik pada *access point* dan *client* dan untuk dapat memberikan distribusi *bandwith* yar TIME (S)

Untuk dapat memantau kualitas dari signal yang di hasilkan oleh *akses point* pada sisi *client* juga dapat di pantau melalui CCQ (*Client Connection Quality*) dari perangkat tersebut.

5.2. Saran

Pengembangan kedepannya dapat di implementasikan pada wireless IEEE.802.11 ac dengan *throughput* 1300 Mbps.

IV. Daftar Pustaka

- [1] *IEEE 802.11-2007: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.*
- [2] [IEEE 802.11-2012: 16.4.7.5 - Transmit spectrum mask](#). *IEEE*. 2013-05-15.
- [3] IEEE 802.11-2007 Annex J modified by amendments k, y and n.
- [4] 802.11n-2009 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control.
- [5] Vivek P. Mhatre, K. Papagiannaki, Francois Baccelli "Interference Mitigation through Power Control in High Density 802.11 WLANs" , *publication in the IEEE INFOCOM 2007 proceedings.*
- [6] Janis Janson, Toms Dorins, "Analyzing IEEE 802.11n standard: outdoor performanace" 978-1-46373-1107-6/12/\$31.00 © 2012 IEEE.
- [7] Pritesh Chandaliya, Nitin Dhakate, Udaykumar Lokhande, D. D. Ambawade "Interference Analysis of IEEE 802.11n," *International Conference on Communication, Information & Computing Technology (ICCICT)*, Mumbai, india , Oct. 19-20, 2012.
- [8] B. Kauffmann, F. Baccelli, Augustin Chaintreau, Vivek Mhatre, Konstantina Papagiannaki, Christophe Diot, "Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks," *IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM proceedings*, 2007, France.
- [9] Intelligent Appliances Controller Using Raspberry Pi Through Android Application & Browser Mr. Atonu Ghosh Institute of Engineering & Management: Computer Science & Engineering Department Salt Lake, Kolkata, India
- [10] Smart Projectors using Remote Controlled Raspberry Pi *International Journal of Computer Applications (0975 – 8887) Volume 82 – No 16, November 2013.*
- [11] Kalyan Pathapati Subbu, Suchith Soman, "An Interference Mitigation Scheme for IEEE 802.15.4 Networks under IEEE 802.11b/g an Interference Amrita Center for Wireless Networks and Applications," *Kerala 690525, IEEE – 33044, 5th ICCCNT – Hufei, China – July, 11-13, 2014.*
- [12] T.Y. Arif, "Pengantar Protokol IEEE 802.11n", Modul Kuliah Kapita Selekt Telekomunikasi, *Jurusan Teknik Elektro, Fakultas Teknik*, Universitas Syiah Kuala.
- [13] Xiong Cai, Xiaohui Li, Ruiyang Yuan, Yongqiang Hei, "Identification and Mitigation of NLOS Based on Channel State Information for Indoor WiFi Localization," 978-1-4673-7687-7/15 ©2015 IEEE State Key Laboratory of Integrated Service Networks Xidian University Xi'an, Shaanxi 710071, 2015, China.
- [14] P.P del Javier, Choi Sungyun " Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement " 0-7803-7802-4/03/\$17.00 © 2003 IEEE.

Lampiran

PROGRAM Best_channel

DEKLARASI

```

userip, username, password: string
wlan: integer
userip_client, username_client, password_client: string
wlan_client: integer
scan_duration, connection_timeout: integer
data_wireless, mapping_data_wireless: array
variable_login, variable_login_client: object

```

IMPLEMENTASI

```

variable_login <--- login(userip, username, password,
connection_timeout, x: string)
find_current_freq(variable_login)
data_wireless <--- scan(variable_login, wlan, scan_duration)
mapping_data_wireless <--- mapping_freq(data_wireless)
best_freq <--- get_best_freq(mapping_data_wireless)
variable_login_client <--- login(userIp_client,
username_client, password_client,
connection_timeout,"string")
change_ap2(variable_login_client,best_freq)
change_ap1(variable_login,best_freq)

```

END.

FUNCTION

Object login(userip, username, password, connection_timeout, x: string)

```

client <--- new RouterOS\Client(userIp, username, password,
null, false, connection_timeout)
print("Login x OK")
return client

```

String find_current_freq(client)

```

setRequest <--- new
RouterOS\Request('/interface/wireless/print')
query <--- RouterOS\Query::where('name', 'wlan1')
responses <--- client->sendSync(setRequest->setQuery(query))

frequency <--- null
foreach (responses as response) {
    foreach (response as name => value) {
        if(name=='frequency'){
            frequency <--- value
        }
    }
}
print("Frekuensi Access Point saat ini : frequency MHz")
return frequency

```

Object scan(client, wlan, scan_duration)

```

print("Melakukan Scanning...")
setRequest <--- new
RouterOS\Request('/interface/wireless/scan')
responses <--- client->sendSync(setRequest
scan_duration)
    ->setArgument('duration',
    ->setArgument('number', wlan)
)
networks: array
i=0
foreach (responses as response) {
    if (response->getType() ===
RouterOS\Response::TYPE_DATA) {
        mac <--- response->getProperty('address')
        ssid <--- response->getProperty('ssid')
        freq <--- response->getProperty('freq')
        sig <--- response->getProperty('sig')

        if(mac!='00:0C:42:61:B7:21'){
            networks[mac] <--- array(
                'ssid' => ssid,
                'freq' => freq,
                'sig' => sig,
                'mac' => mac
            )
        }
        i++
    }
}
networks = array_sort(networks,'sig',SORT_DESC)
return networks

```

Array mapping_freq(data_freq)

```

group_channel = array(
    '2412'=>array(),
    '2417'=>array(),
    '2422'=>array(),
    '2427'=>array(),
    '2432'=>array(),
    '2437'=>array(),
    '2442'=>array(),
    '2447'=>array(),
    '2452'=>array(),
    '2457'=>array(),
    '2462'=>array()
)
foreach (data_freq as value) {
    array_push(group_channel[value['freq']], value)
}
return group_channel

```

String get_best_freq(data_freq)

```

channel = array(
    '2412',
    '2417',
    '2422',
    '2427',
    '2432',
    '2437',
    '2442',

```

```

    '2447',
    '2452',
    '2457',
    '2462'
)
max <--- 0
i <--- 0
foreach (data_freq as detail_data => value) {
    if(count(value)==0){
        print("Frekuensi yang tidak digunakan :
        ".channel[i]." MHz")
        return channel[i]
        break
    }
    i++
}
return null

```

String change_ap1(client, freq_set)

```

setRequest <--- new RouterOS\Request('/interface wireless set')
responses <--- client->sendSync(setRequest
    ->setArgument('frequency', freq_set)
    ->setArgument('numbers', 0)
)
print( "Mengubah frekuensi radio AP ke freq_set MHz")

```

String change_ap1(client, freq_set)

```

setRequest <--- new RouterOS\Request('/interface wireless set')
responses <--- client->sendSync(setRequest
    ->setArgument('frequency', freq_set)
    ->setArgument('numbers', 1)
)
print( "Mengubah frekuensi radio AP ke freq_set MHz")

```