

Penelitian Pengaruh MAC Address Collision Pada Perangkat Yang Dapat Merubah MAC Address

Gabriel Possenti Kheisa Drianasta*¹, Bakhtiar Alldino Ardi Sumbodo²,
Jazi Eko Istiyanto³,

¹Program Studi Elektronika dan Instrumentasi, DIKE, FMIPA, UGM, Yogyakarta, Indonesia

¹Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *¹gabrielkheisa@mail.ugm.ac.id, ²b.alldino.as@ugm.ac.id, ³jazi@ugm.ac.id,

Abstrak

Alamat Media Access Control (MAC) atau MAC address adalah suatu alamat dengan panjang 48 bit yang merupakan alamat unik yang diberikan untuk perangkat Network Interface Card (NIC). Setiap perangkat dapat diidentifikasi secara unik pada suatu jaringan karena memiliki MAC address yang berbeda – beda. MAC address, sebagai alamat fisik, terdapat pada lapisan kedua dari Open Systems Interconnection (OSI) layer atau disebut juga Data Link Layer, yang dimana pada layer ini setiap bingkai atau frame data dikemas dengan MAC address sumber dan MAC address tujuan untuk menandakan pengiriman dan penerimaan data di jaringan, serta memungkinkan perangkat bertukar data dengan benar.

MAC address collision terjadi ketika dua atau lebih perangkat di suatu jaringan menggunakan MAC address yang sama. Beberapa sistem operasi terbaru, seperti Windows 10 dan Android 6.0, mendukung penggantian MAC address secara acak untuk meningkatkan privasi pengguna dengan menyembunyikan MAC address yang sesungguhnya. Jika dua perangkat di suatu jaringan memiliki MAC address yang sama, maka jaringan tidak dapat membedakannya, dan dapat menyebabkan perselisihan dalam transmisi data.

Penelitian Pengaruh MAC address collision dilakukan pada secara langsung pada suatu router fisik dan dilakukan juga pada virtualized environment pada VirtualBox.

Kata kunci—MAC Address collision, VirtualBox, OSI layer, Data Link Layer

Abstract

Media Access Control (MAC) address or MAC address is an address with a length of 48 bits which is a unique address given to Network Interface Card (NIC) devices. Each device can be uniquely identified on a network because it has a different MAC address. MAC addresses, as physical addresses, are at the second layer of the Open Systems Interconnection (OSI) layer or also called the Data Link Layer, where at this layer each data frame is packed with a source MAC address and destination MAC address to indicate sending and receiving data on network, and allows devices to exchange data properly.

MAC address collision occurs when two or more devices on a network use the same MAC address. Some of the latest operating systems, such as Windows 10 and Android 6.0, support changing the MAC address randomly to increase user privacy by hiding the real MAC address. If two devices on a network have the same MAC address, the network cannot tell the difference, and can cause disputes in data transmission.

Research on the effect of MAC address collision was carried out directly on a physical router and also carried out in a virtualized environment on VirtualBox.

Keywords—3-5 MAC Address collision, VirtualBox, OSI layer, Data Link Layer

1. PENDAHULUAN

Dalam suatu jaringan internet, terdapat aturan - aturan atau standar agar suatu perangkat dapat berkomunikasi satu sama lain. Salah satu standar yang paling umum adalah standar IEEE 802 yang umum digunakan dalam jaringan Wi-Fi. (IEEE 802, 2004) Di dalam IEEE 802, terdapat dua lapisan, yakni lapisan Data Link Layer yang terdiri dari Logical Link Layer (LLC) dan Medium Access Control (MAC), dan Physical Layer. MAC adalah lapisan yang mengatur pertanggungjawaban perangkat keras dalam komunikasi antar kabel, antar optik, atau nirkabel. MAC address mengatur pengenalan paket internet, pengalamatan tujuan paket, dan kontrol akses dalam medium fisik. Alamat MAC pada setiap perangkat unik, umumnya tidak dapat diubah dan memiliki panjang data 48-bit, dimana 24-bit pertama adalah kode dari perusahaan penerbit perangkat.

Ketika dua atau lebih perangkat di suatu jaringan memiliki MAC address yang sama, dapat membuat perselisihan pada lapisan Open Systems Interconnection (OSI) kedua, atau Data Link Layer. Kejadian ini disebut juga MAC address collision, dan dapat menyebabkan konflik dalam proses transmisi data dengan pengemasan alamat MAC sumber dan alamat MAC tujuan. Sebelum perangkat ingin mengirim paket data ke perangkat lain pada suatu jaringan, perangkat tersebut harus mengetahui dahulu alamat MAC perangkat yang dituju agar paket data dapat dikirim dengan tujuan yang benar. Proses pemetaan ini dilakukan oleh Address Resolution Protocol (ARP), yaitu sebuah protokol yang digunakan untuk menemukan alamat MAC tujuan dengan mencari terlebih dahulu alamat IP atau Internet Protocol address dari perangkat tujuan.

Ketika ARP menemukan alamat IP perangkat tujuan, maka perangkat pengirim mengirim permintaan ARP ke jaringan tersebut untuk meminta alamat MAC yang sesuai. Hubungan antara alamat MAC dan ARP adalah jika alamat MAC adalah suatu kode untuk mengidentifikasi perangkat pada suatu jaringan, maka ARP merupakan protokol untuk menemukan alamat MAC antara perangkat pengirim dan penerima melalui pencarian alamat IP, dimana alamat IP ini sendiri berjalan pada lapisan yang lebih tinggi pada lapisan OSI ke-3, atau disebut juga Network Layer. Layer ke-3 ini bertanggung jawab dalam pengiriman paket data antar jaringan, dimana setiap perangkat memiliki alamat IP yang unik agar dapat diidentifikasi oleh jaringan, yang memiliki hubungan erat dengan ARP, dimana ARP itu sendiri memetakan alamat IP ke alamat MAC yang berjalan pada lapisan OSI satu tingkat dibawah IP. Network Layer juga menangani proses routing, yaitu proses mengirimkan paket data melalui jalur yang optimal. Network Layer mampu menentukan jalur terbaik untuk mengirimkan data dengan adanya informasi routing yang tersimpan pada routing table. Dalam hal ini, Internet Protocol (IP) juga merupakan elemen yang penting, karena IP, ARP, dan MAC berhubungan erat dan memiliki persamaan untuk mengidentifikasi perangkat pada suatu jaringan.

2. METODE PENELITIAN

Perangkat keras dan perangkat lunak dipastikan dapat diubah MAC Address nya. Percobaan juga dilakukan dalam mesin virtual pada program VirtualBox, dimana virtual NIC dapat dilakukan penggantian MAC Address.

Percobaan pada hasil pra-proposal dilakukan pada router dan AP ZTE F609 dengan versi perangkat lunak V7.0.10P1N14. Perangkat keras yang digunakan pada percobaan ini adalah NodeMCU dengan mikrokontroler ESP8266 yang memiliki spesifikasi CPU 32-bit RISC yang berjalan pada kecepatan 80MHz, memori sebesar 80kB, penyimpanan flash sebesar 4MB, serta mampu berjalan pada protokol Wi-Fi 2.4GHz 802.11/b/g/n. Dukungan untuk mengganti alamat MAC menjadi alasan mengapa mikrokontroler ini digunakan pada penelitian ini. Penggantian

alamat MAC pada ESP8266 dapat dilakukan dengan software development kit (SDK) bawaan atau dengan Arduino ESP8266 Core. Dalam penelitian ini, digunakan Arduino IDE dengan tambahan pustaka ESP8266WiFi. Mengganti alamat MAC pada Arduino IDE dilakukan dengan fungsi dari `wifi_set_macaddr(STATION_IF, new_mac)`.

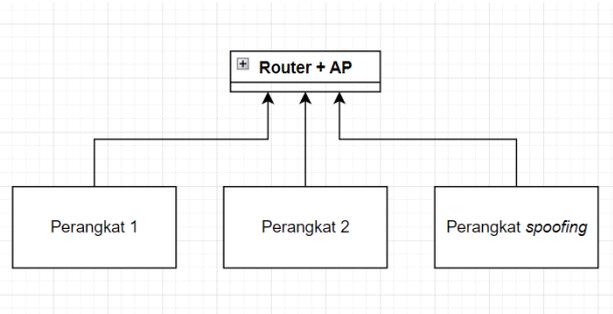
2.1 Rancangan sistem

Penggantian alamat MAC NodeMCU dilakukan sesuai dengan skenario penelitian yang akan diuji.

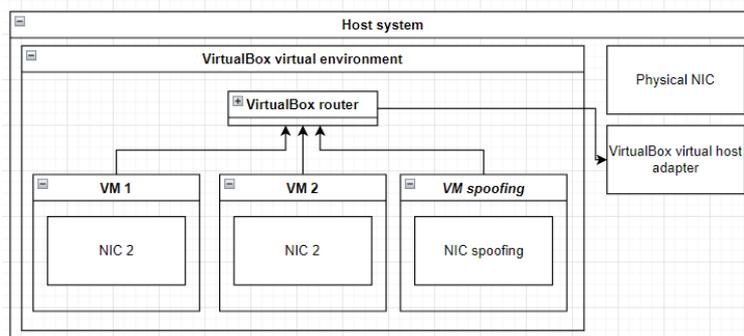
Percobaan dengan traceroute dilakukan dengan menjalankan perintah traceroute pada berbagai OS yang mendukungnya. Pada Windows, traceroute dapat dilakukan dengan cara mengetik “tracert” pada command prompt, diikuti dengan hostname atau IP yang ingin dituju.

Meskipun traceroute hanya merupakan utilitas untuk melakukan pelacakan rute paket, data hasil traceroute digunakan sebagai variabel terikat pembandingan antara kondisi jaringan dalam keadaan ideal dengan kondisi jaringan dalam keadaan dimana terjadi MAC address collision.

Selanjutnya adalah menganalisa pengaruh MAC address collision dengan menggunakan HTTP GET atau cURL untuk mengamati pengaruh MAC address collision pada application layer pada model lapisan OSI. HTTP GET dapat dilakukan melalui browser pada umumnya dengan memasukkan alamat IP atau URL ke address bar. Hal ini setara dengan HTTP GET yang dilakukan oleh cURL GET.



Gambar 1 Topologi pertama (fisik)



Gambar 2 Topologi kedua (virtual environment)

MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	avg Ping variabel bebas ke Router (ms)	avg Ping MAC VM 1 ke Router (ms)	avg Ping MAC VM 2 bebas ke Router (ms)	MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	HTTP GET MAC variabel bebas ke VM Variabel bebas (ms)	HTTP GET MAC VM 1 ke VM Variabel bebas (ms)	HTTP GET MAC VM 2 ke VM Variabel bebas (ms)
08:00:27 :D8:98:53	08:00:27 :6A:5C:2E	08:00:27 :9E:C9:3E	08:00:27 :A0:DF:71				08:00:27 :D8:98:53	08:00:27 :6A:5C:2E	08:00:27 :9E:C9:3E	08:00:27 :A0:DF:71			
MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	avg Ping variabel bebas ke VM 1 (ms)	avg Ping MAC VM 1 ke VM 1 (ms)	avg Ping MAC VM 2 bebas ke VM 1 (ms)	MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	HTTP GET MAC variabel bebas ke VM 2 (ms)	HTTP GET MAC VM 1 ke VM 2 (ms)	HTTP GET MAC VM 2 ke VM 2 (ms)
08:00:27 :D8:98:53	08:00:27 :6A:5C:2E	08:00:27 :9E:C9:3E	08:00:27 :A0:DF:71				08:00:27 :D8:98:53	08:00:27 :6A:5C:2E	08:00:27 :9E:C9:3E	08:00:27 :A0:DF:71			
MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	traceroute MAC variabel bebas ke Router (ms)	traceroute MAC VM 1 ke Router (ms)	traceroute MAC VM 2 bebas ke Router (ms)							
08:00:27 :D8:98:53	08:00:27 :6A:5C:2E	08:00:27 :9E:C9:3E	08:00:27 :A0:DF:71										
MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	traceroute MAC variabel bebas ke VM 1 (ms)	avg Ping variabel bebas ke VM 1 (ms)	traceroute MAC VM 2 bebas ke VM 1 (ms)							
08:00:27 :D8:98:53	08:00:27 :6A:5C:2E	08:00:27 :9E:C9:3E	08:00:27 :A0:DF:71										

Tabel 2 Hasil sementara yang didapat pra-proposal

3. HASIL DAN PEMBAHASAN

3.1 Percobaan sementara pada lingkungan fisik

Dengan konfigurasi berikut, perangkat penguji akan mengambil MAC Address dari perangkat lain yang terhubung. Sebelum tersambung, MAC Address perangkat penguji adalah default atau diacak terlebih dahulu, kemudian setelah tersambung dilakukan scanning untuk mendapatkan MAC Address dari perangkat lain dalam jaringan yang sama.

Hasil sementara yang diperoleh adalah adanya fenomena peningkatan ping, dimana pada keadaan normal, dimana tidak ada MAC Address yang sama, ping berkisar dari dua hingga lima milisekon. Pada keadaan dimana MAC Address NodeMCU disamakan dengan MAC Address dari router, terjadi lonjakan latency, dimana ping berkisar dari 230 hingga 296 milisekon, bahkan respon ping perangkat komputer menuju router menjadi request timed out (RTO).

MAC NodeMCU	MAC Computer	MAC Handphone	MAC Router	avg Ping Computer, Router (ms)	avg Ping Computer, Handphone (ms)	avg Ping Handphone, Router (ms)	avg Ping Handphone, Computer (ms)
34:13:E8:0C:6:4C:42	34:13:E8:0C:6:4C:42	80:35:C1:3E:3B:3C	30:42:40:1E:09:8C	5	4	2	2
30:42:40:1E:09:8C	34:13:E8:0C:6:4C:43	80:35:C1:3E:3B:3C	30:42:40:1E:09:8C	RTO	296	230	
80:35:C1:3E:3B:3C	34:13:E8:0C:6:4C:44	80:35:C1:3E:3B:3C	30:42:40:1E:09:8C	3	RTO		RTO
34:13:E8:0C:6:4C:45	34:13:E8:0C:6:4C:43	80:35:C1:3E:3B:3C	F4:F5:DB:D2:D2:FB	89	103	40	103
F4:F5:DB:D2:D2:FB	34:13:E8:0C:6:4C:44	80:35:C1:3E:3B:3C	F4:F5:DB:D2:D2:FB	80	120		

Tabel 3 Hasil sementara yang didapat pra-proposal

3.2 Percobaan pada lingkungan virtual dengan VirtualBox

Keadaan Normal (tidak ada duplikasi alamat MAC)

Percobaan dengan VirtualBox dalam keadaan ideal, dimana semua VM, termasuk VM OpenWRT, memiliki alamat MAC yang unik. VM OpenWRT yang bertindak sebagai router, sekaligus server DHCP, memberikan alamat IP yang unik untuk kedua VM dengan alamat IP masing – masing 192.168.69.184 dan 192.168.69.137 dengan alamat IP gateway 192.168.69.1, yaitu VM OpenWRT itu sendiri.

Parameter yang diuji adalah ping. VM pertama dan kedua melakukan pengujian ping ke VM OpenWRT yang bertindak sebagai gateway. Hasilnya, ping kedua VM membutuhkan waktu kurang dari 1 milisekon.

Kemudian, VM pertama dengan IP 192.168.69.184 melakukan ping ke VM kedua, dengan IP 192.168.69.137 dan sebaliknya. Hasilnya adalah kedua VM mampu memberikan latency dibawah 1 milisekon, baik dari VM 192.168.69.184 menuju VM 192.168.69.137, maupun dari VM 192.168.69.137 ke VM 192.168.69.184.

Kemudian, dilakukan pengujian HTTP GET ke VM OpenWRT dengan menggunakan browser Mozilla Firefox yang ada pada sistem operasi Lubuntu. Dengan memasukkan alamat host atau IP gateway, maka hal tersebut setara dengan HTTP GET dengan menggunakan cURL. Hasilnya adalah login page gateway dapat diperoleh dengan baik.

MAC VM Variabel bebas	MAC VM 1	MAC VM Pengamat	MAC router	avg Ping MAC VM variabel bebas ke Router (ms)	avg Ping MAC VM variabel bebas ke VM 1 (ms)	avg Ping MAC VM variabel bebas ke VM pengamat (ms)
08:00:27: D8:98:53	08:00:27: 6A:5C:2D	08:00:27: 9E:C9:36	08:00:27: A0:DF:71	0.2	0.2	0.2
MAC VM Variabel bebas	MAC VM 1	MAC VM Pengamat	MAC router	avg HTTP GET VM variabel bebas ke Router (ms)		
08:00:27: D8:98:53	08:00:27: 6A:5C:2D	08:00:27: 9E:C9:36	08:00:27: A0:DF:71	Berhasil		

Tabel 4 Hasil sementara yang didapat pra-proposal

Keadaan dua VM dengan alamat MAC sama dan satu VM pengamat

Percobaan berikutnya dengan VirtualBox dalam keadaan dua buah VM dengan alamat MAC yang sama, hanya MAC VM yang memiliki alamat MAC yang unik. Tampak server DHCP memberikan alamat IP yang sama untuk kedua VM tersebut, sehingga kedua VM memiliki alamat IP 192.168.69.184. Hanya satu VM yang memiliki satu alamat IP yang unik, yaitu 192.168.69.220.

Pada saat VM pengamat melakukan ping ke alamat IP VM dengan alamat MAC yang sama, hasilnya adalah terjadi respon ganda dengan pesan “(DUP!)”, yang menandakan adanya duplicate packets, namun latency tetap lebih kecil dari 1 milidetik.

Kemudian, dilakukan ping dari masing – masing VM dengan alamat MAC duplicate ke VM pengamat. Hasilnya, ping tampak normal dan latency keduanya bernilai kurang dari 1 milidetik.

Selanjutnya, dilakukan percobaan HTTP GET ke webpage OpenWRT dengan alamat 192.168.69.1 dari masing – masing VM yang terdiri dari dua VM dengan alamat MAC yang sama dan satu VM pengamat. Hasilnya adalah hanya VM pengamat yang dapat menampilkan webpage OpenWRT, sedangkan kedua VM duplicate tidak dapat menampilkannya, serta terdapat pesan “The Connection was reset”.

MAC VM Variabel bebas	MAC VM 1	MAC VM Pengamat	MAC router	avg Ping MAC VM variabel bebas ke Router (ms)	avg Ping MAC VM variabel bebas ke VM 1 (ms)	avg Ping MAC VM variabel bebas ke VM pengamat (ms)
08:00:27:D8:98:53	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.2	0.2	0.2
08:00:27:9E:C9:36	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.2	0.2 (DUP!)	0.2
MAC VM Variabel bebas	MAC VM 1	MAC VM Pengamat	MAC router	avg HTTP GET VM variabel bebas ke Router (ms)		
08:00:27:D8:98:53	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	Berhasil		
08:00:27:9E:C9:36	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	Gagal		

Tabel 5 Hasil sementara yang didapat pra-proposal

Keadaan VM duplicate dengan VM OpenWRT

Percobaan berikutnya dengan VirtualBox dalam keadaan salah satu alamat MAC VM memiliki alamat yang sama dengan alamat MAC router OpenWRT. Alamat MAC VM yang akan disamakan dengan alamat MAC OpenWRT adalah 08:00:27:D8:98:53, diganti menjadi alamat MAC OpenWRT dengan nilai 08:00:27:A0:DF:71.

Pengujian ping dilakukan menuju router OpenWRT dengan alamat IP 192.168.69.1. Semua VM berhasil melakukan ping ke VM OpenWRT, namun pada salah satu VM dengan alamat MAC yang sama dengan VM OpenWRT, nilai latency berada 0.909 ms hingga 2.10 ms, berbeda dengan latency kedua VM lain menuju VM OpenWRT dengan alamat MAC yang unik.

Pengujian ping selanjutnya adalah menuju VM dengan alamat MAC unik dengan alamat IP 192.168.69.220. Semua perangkat berhasil melakukan ping dengan latency dibawah 1 ms, namun pada VM duplicate terdapat respon ganda dengan pesan “DUP!”.

Pengujian ping selanjutnya adalah menuju VM dengan alamat MAC duplicate yang sama dengan alamat MAC OpenWRT dengan alamat IP 192.168.69.181. Kedua VM dengan alamat MAC yang unik berhasil melakukan ping dengan latency dibawah 1ms namun terdapat respon ganda “DUP!” dan terdapat pesan “From 192.168.69.1: icmp_seq=2 Redirect Host(New nexthop: 192.168.69.181)” pada paket ICMP dengan nomor sequence 2.

Pengujian ping yang terakhir adalah menuju VM dengan alamat MAC unik kedua dengan alamat IP 192.168.69.184. Sama seperti pada percobaan ping pada VM pertama, semua perangkat berhasil melakukan ping dengan latency dibawah 1 ms, namun pada VM duplicate terdapat respon ganda dengan pesan “DUP!”.

MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	avg Ping variabel bebas ke Router (ms)	avg Ping MAC VM 1 ke Router (ms)	avg Ping MAC VM 2 bebas ke Router (ms)	MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	avg Ping variabel bebas ke VM Variabel bebas (ms)	avg Ping MAC VM 1 ke VM Variabel bebas (ms)	avg Ping MAC VM 2 ke VM Variabel bebas (ms)
08:00:27:D8:98:53	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.2	0.2	0.2	08:00:27:D8:98:53	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.2	0.2	0.2
08:00:27:A0:DF:71	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	1.2	0.2	0.2	08:00:27:A0:DF:71	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.3 (DUP!)	0.3 (DUP!)	

MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	avg Ping variabel bebas ke VM 1 (ms)	avg Ping MAC VM 1 ke VM 1 (ms)	avg Ping MAC VM 2 bebas ke VM 1 (ms)	MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	avg Ping variabel bebas ke VM 2 (ms)	avg Ping MAC VM 1 ke VM 2 (ms)	avg Ping MAC VM 2 ke VM 2 (ms)
08:00:27:D8:98:53	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.2	0.2	0.2	08:00:27:D8:98:53	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.2	0.2	0.2
08:00:27:A0:DF:71	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.2 (DUP!)	0	0.2	08:00:27:A0:DF:71	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.3 (DUP!)	0.2	0

Tabel 6 Pengujian ping pada masing – masing VM

Pengujian selanjutnya adalah traceroute. Traceroute dilakukan menuju alamat IP masing – masing VM yang berbeda. Semua traceroute berhasil dilakukan, namun pada pengujian traceroute yang pertama, dimana masing – masing VM melakukan traceroute menuju VM OpenWRT dengan alamat IP 192.168.69.1, VM yang memiliki alamat MAC yang sama dengan OpenWRT memiliki latency yang relatif lebih besar dibandingkan kedua VM lainnya, dimana VM duplicate memiliki latency sebesar 1.964 – 3.308 ms, berbeda dengan kedua VM lainnya yang memiliki latency sebesar 0.296 – 0.385 ms. Kemudian, pada pengujian traceroute menuju VM OpenWRT, terdapat dua buah hop pada salah satu VM unik, yaitu VM OpenWRT pada hop pertama, dan VM duplicate pada hop kedua. Pada keadaan normal dalam suatu jaringan yang sama, hop hanya terdapat satu buah, dengan alamat IP hop tujuan pada respon traceroute. Lalu, pada saat traceroute dilakukan menuju masing – masing perangkat itu sendiri pada masing – masing perangkat, kedua perangkat unik memiliki latency yang sangat kecil, dengan nilai 0.004 – 0.024 ms, namun pada VM duplicate nilai latency nya bernilai 0.163 – 0.210 ms.

MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	traceroute MAC variabel bebas ke Router (ms)	traceroute MAC VM 1 ke Router (ms)	traceroute MAC VM 2 ke Router (ms)	MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	traceroute MAC variabel bebas ke VM Variabel bebas (ms)	traceroute MAC VM 1 ke VM Variabel bebas (ms)	traceroute MAC VM 2 ke VM Variabel bebas (ms)
08:00:27:D8:98:53	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.3	0.3	0.3	08:00:27:D8:98:53	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0	0.3	0.3
08:00:27:A0:DF:71	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	2.3	0.3	0.3	08:00:27:A0:DF:71	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.2	0.5 (2 hop)	0.3

MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	traceroute MAC variabel bebas ke VM 1 (ms)	avg Ping MAC VM 1 ke VM 1 (ms)	traceroute MAC VM 2 bebas ke VM 1 (ms)	MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	traceroute MAC variabel bebas ke VM 2 (ms)	traceroute MAC VM 1 ke VM 2 (ms)	traceroute MAC VM 2 ke VM 2 (ms)
08:00:27:D8:98:53	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.3	0	0.3	08:00:27:D8:98:53	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.3	0.3	0
08:00:27:A0:DF:71	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.3	0	0.2	08:00:27:A0:DF:71	08:00:27:6A:5C:2D	08:00:27:9E:C9:36	08:00:27:A0:DF:71	0.5	0.4	0.4

Tabel 7 Hasil pengujian traceroute pada masing – masing VM

Pengujian HTTP GET dilakukan dengan adanya server web Nginx yang berjalan pada masing – masing VM pada port TCP 80. Semua permintaan HTTP GET berhasil ke masing – masing VM dengan latency 1ms, baik menuju masing – masing server web Nginx dan webpage OpenWRT.

MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	traceroute MAC variabel bebas ke Router (ms)	traceroute MAC VM 1 ke Router (ms)	traceroute MAC VM 2 ke Router (ms)	MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	traceroute MAC variabel bebas ke VM Variabel bebas (ms)	traceroute MAC VM 1 ke VM Variabel bebas (ms)	traceroute MAC VM 2 ke VM Variabel bebas (ms)
08:00:27: D8:98:53 A0:DF:71	08:00:27: 6A:5C:2D	08:00:27: 9E:C9:36	08:00:27: A0:DF:71	0.3	0.3	0.3	08:00:27: D8:98:53 A0:DF:71	08:00:27: 6A:5C:2D	08:00:27: 9E:C9:36	08:00:27: A0:DF:71	0	0.3	0.3
08:00:27: A0:DF:71	08:00:27: 6A:5C:2D	08:00:27: 9E:C9:36	08:00:27: A0:DF:71	3.3	0.3	0.3	08:00:27: A0:DF:71	08:00:27: 6A:5C:2D	08:00:27: 9E:C9:36	08:00:27: A0:DF:71	0.2	0.5 (2 hop)	0.3
MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	traceroute MAC variabel bebas ke VM 1 (ms)	avg Ping MAC VM 1 ke VM 1 (ms)	traceroute MAC VM 2 bebas ke VM 1 (ms)	MAC VM Variabel bebas	MAC VM 1	MAC VM 2	MAC router	traceroute MAC variabel bebas ke VM 2 (ms)	traceroute MAC VM 1 ke VM 2 (ms)	traceroute MAC VM 2 ke VM 2 (ms)
08:00:27: D8:98:53 A0:DF:71	08:00:27: 6A:5C:2D	08:00:27: 9E:C9:36	08:00:27: A0:DF:71	0.3	0	0.3	08:00:27: D8:98:53 A0:DF:71	08:00:27: 6A:5C:2D	08:00:27: 9E:C9:36	08:00:27: A0:DF:71	0.3	0.3	0
08:00:27: A0:DF:71	08:00:27: 6A:5C:2D	08:00:27: 9E:C9:36	08:00:27: A0:DF:71	0.5	0	0.2	08:00:27: A0:DF:71	08:00:27: 6A:5C:2D	08:00:27: 9E:C9:36	08:00:27: A0:DF:71	0.5	0.4	0.4

Tabel 8 Hasil pengujian HTTP GET pada masing – masing VM

Analisa melalui program Wireshark dilakukan pada VM pengamat dengan alamat IP 192.168.69.184. VM pengamat tidak dapat mengamati traffic ICMP pada VM lain, karena desain internal network pada VirtualBox yang berupa switch, dan bukan hub. Maka, analisa melalui Wireshark hanya dapat dilakukan pada salah satu VM yang akan melakukan pengujian ping dan traceroute. Percobaan pertama dilakukan dengan melakukan ping ke VM OpenWRT dan VM lainnya dengan alamat IP 192.168.69.220. Hasilnya, tampak bahwa dalam satu kali ping terdapat satu kali respon pong, sehingga untuk 3 kali ping dengan nomor icmp_seq=3, total payload yang dikirim dan diterima adalah 6.

Selanjutnya, dilakukan percobaan ping dari VM pengamat ke VM duplicate. Hasilnya, terdapat total 7 buah payload hanya pada percobaan ping kedua, dengan nomor icmp_seq=2. Terdapat dua buah payload pong setelah ping dengan nomor icmp_seq=1, dan satu buah payload pong satu buah payload redirect, dan satu buah payload pong setelah redirect pada ping dengan nomor icmp_seq=2.

Selanjutnya, dilakukan pengamatan traceroute pada Wireshark menuju VM duplicate dengan alamat IP 192.168.69.181. Payload traceroute tampak lebih banyak dibandingkan traceroute ke VM lain dan juga pada keadaan normal, dan juga terdapat payload “Time-to-live exceeded” dari VM OpenWRT.

3.3 Percobaan pada lingkungan fisik

Keadaan Normal (tidak ada duplikasi alamat MAC)

Percobaan dilakukan dengan router Mi Router 4A, dengan client NodeMCU sebagai variabel bebas (perangkat yang dapat mengubah alamat MAC), client kedua yaitu perangkat Android, dan komputer Windows dengan NIC Intel Wi-Fi 6 AX200 yang berjalan pada mode 802.11ac (pada frekuensi 5GHz) melalui percobaan meliputi ping, HTTP GET request, dan traceroute, serta analisis Wireshark.

Pada keadaan normal, ping NodeMCU ke router adalah berkisar antara 3ms – 58ms, ping komputer ke router adalah 1ms, serta ping dari perangkat Android ke router adalah 7 – 105ms.

MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	avg Ping NodeMCU ke Router (ms)	avg Ping komputer ke Router (ms)	avg Ping Android ke Router (ms)	MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	avg Ping NodeMCU ke Komputer (ms)	avg Ping komputer ke Komputer (ms)	avg Ping Android ke Komputer (ms)
84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	8	1	30	84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	17	0	28
MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	avg Ping NodeMCU ke Router (ms)	avg Ping komputer ke Router (ms)	avg Ping Android ke Router (ms)	MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	avg Ping NodeMCU ke Android (ms)	avg Ping komputer ke Android (ms)	avg Ping Android ke Android (ms)
84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69				84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	264	208	0

Tabel 9 Hasil pengujian ping pada masing – masing perangkat

MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	HTTP GET NodeMCU ke Router (ms)	HTTP GET komputer ke Router (ms)	HTTP GET Android ke Router (ms)	MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	HTTP GET NodeMCU ke Komputer (ms)	HTTP GET komputer ke Komputer (ms)	HTTP GET Android ke Komputer (ms)
84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Berhasil	Berhasil	Berhasil	84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Berhasil	Berhasil	Berhasil
MAC Variabel bebas (NodeMCU) <td>MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke NodeMCU (ms)</th> <th>HTTP GET komputer ke NodeMCU (ms)</th> <th>HTTP GET Android ke NodeMCU (ms)</th> <td>MAC Variabel bebas (NodeMCU)</td> <td>MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td></td></td></td></td></td>	MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke NodeMCU (ms)</th> <th>HTTP GET komputer ke NodeMCU (ms)</th> <th>HTTP GET Android ke NodeMCU (ms)</th> <td>MAC Variabel bebas (NodeMCU)</td> <td>MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td></td></td></td></td>	MAC Android <td>MAC router <th>HTTP GET NodeMCU ke NodeMCU (ms)</th> <th>HTTP GET komputer ke NodeMCU (ms)</th> <th>HTTP GET Android ke NodeMCU (ms)</th> <td>MAC Variabel bebas (NodeMCU)</td> <td>MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td></td></td></td>	MAC router <th>HTTP GET NodeMCU ke NodeMCU (ms)</th> <th>HTTP GET komputer ke NodeMCU (ms)</th> <th>HTTP GET Android ke NodeMCU (ms)</th> <td>MAC Variabel bebas (NodeMCU)</td> <td>MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td></td></td>	HTTP GET NodeMCU ke NodeMCU (ms)	HTTP GET komputer ke NodeMCU (ms)	HTTP GET Android ke NodeMCU (ms)	MAC Variabel bebas (NodeMCU)	MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td></td>	MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td>	MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th>	HTTP GET NodeMCU ke Android (ms)	HTTP GET komputer ke Android (ms)	HTTP GET Android ke Android (ms)
84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69		Berhasil	Berhasil	84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Berhasil	Berhasil	Berhasil

Tabel 10 Hasil pengujian HTTP GET pada masing – masing perangkat

Keadaan MAC NodeMCU duplicate dengan router

Pengujian ping pada kondisi MAC NodeMCU duplicate dengan router menunjukkan hasil, dimana NodeMCU tidak dapat melakukan ping dan HTTP GET ke semua perangkat dalam suatu jaringan.

MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	HTTP GET NodeMCU ke Router (ms)	HTTP GET komputer ke Router (ms)	HTTP GET Android ke Router (ms)	MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	HTTP GET NodeMCU ke Komputer (ms)	HTTP GET komputer ke Komputer (ms)	HTTP GET Android ke Komputer (ms)
84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Berhasil	Berhasil	Berhasil	84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Berhasil	Berhasil	Berhasil
MAC Variabel bebas (NodeMCU) <td>MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke NodeMCU (ms)</th> <th>HTTP GET komputer ke NodeMCU (ms)</th> <th>HTTP GET Android ke NodeMCU (ms)</th> <td>MAC Variabel bebas (NodeMCU)</td> <td>MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td></td></td></td></td></td>	MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke NodeMCU (ms)</th> <th>HTTP GET komputer ke NodeMCU (ms)</th> <th>HTTP GET Android ke NodeMCU (ms)</th> <td>MAC Variabel bebas (NodeMCU)</td> <td>MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td></td></td></td></td>	MAC Android <td>MAC router <th>HTTP GET NodeMCU ke NodeMCU (ms)</th> <th>HTTP GET komputer ke NodeMCU (ms)</th> <th>HTTP GET Android ke NodeMCU (ms)</th> <td>MAC Variabel bebas (NodeMCU)</td> <td>MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td></td></td></td>	MAC router <th>HTTP GET NodeMCU ke NodeMCU (ms)</th> <th>HTTP GET komputer ke NodeMCU (ms)</th> <th>HTTP GET Android ke NodeMCU (ms)</th> <td>MAC Variabel bebas (NodeMCU)</td> <td>MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td></td></td>	HTTP GET NodeMCU ke NodeMCU (ms)	HTTP GET komputer ke NodeMCU (ms)	HTTP GET Android ke NodeMCU (ms)	MAC Variabel bebas (NodeMCU)	MAC Komputer <td>MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td></td>	MAC Android <td>MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th> </td>	MAC router <th>HTTP GET NodeMCU ke Android (ms)</th> <th>HTTP GET komputer ke Android (ms)</th> <th>HTTP GET Android ke Android (ms)</th>	HTTP GET NodeMCU ke Android (ms)	HTTP GET komputer ke Android (ms)	HTTP GET Android ke Android (ms)
84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69		Berhasil	Berhasil	84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Berhasil	Berhasil	Berhasil

Tabel 11 Hasil ping pada masing – masing perangkat

Sedangkan ping dari komputer hanya berhasil menuju router dengan latency rata – rata yang lebih besar.

Untuk HTTP GET, tidak seperti pada lingkungan VirtualBox, terjadi kegagalan untuk kasus HTTP GET dari NodeMCU ke semua perangkat, komputer ke NodeMCU dan Android, dan Android ke NodeMCU.

MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	HTTP GET NodeMCU ke Router (ms)	HTTP GET komputer ke Router (ms)	HTTP GET Android ke Router (ms)	MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	HTTP GET NodeMCU ke Komputer (ms)	HTTP GET komputer ke Komputer (ms)	HTTP GET Android ke Komputer (ms)
84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Berhasil	Berhasil	Berhasil	84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Berhasil	Berhasil	Berhasil
50:D2:F5:2F:21:69	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Gagal	Berhasil	Berhasil	50:D2:F5:2F:21:69	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Gagal	Berhasil	Berhasil
MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	HTTP GET NodeMCU ke NodeMCU (ms)	HTTP GET komputer ke NodeMCU (ms)	HTTP GET Android ke NodeMCU (ms)	MAC Variabel bebas (NodeMCU)	MAC Komputer	MAC Android	MAC router	HTTP GET NodeMCU ke Android (ms)	HTTP GET komputer ke Android (ms)	HTTP GET Android ke Android (ms)
84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Berhasil	Berhasil	Berhasil	84:F3:EB:CC:03:EF	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Berhasil	Berhasil	Berhasil
50:D2:F5:2F:21:69	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Gagal	Gagal	Gagal	50:D2:F5:2F:21:69	34:13:EB:C6:4C:42	04:E5:98:32:D0:1E	50:D2:F5:2F:21:69	Gagal	Gagal	Berhasil

Tabel 12 Hasil pengujian HTTP GET pada masing – masing perangkat

4. KESIMPULAN

Penelitian MAC address collision menghasilkan karakteristik anomali yang berbeda pada lingkungan virtual VirtualBox dan percobaan pada lingkungan fisik. Pada pengujian ping di VirtualBox, terdapat pesan "DUP!" pada response dari ping, peningkatan latency yang signifikan, dan bahkan kegagalan ping dengan pesan "request timed out" (RTO) pada lingkungan fisik. Selain itu, pengujian traceroute di VirtualBox menunjukkan peningkatan nilai latency dan melalui dua hop yang seharusnya tidak terjadi pada keadaan normal. Pengujian HTTP GET menunjukkan bahwa semua perangkat di VirtualBox berhasil menampilkan webpage, tetapi hanya beberapa perangkat di lingkungan fisik yang berhasil. Selain itu, terdapat respon HTTP GET yang tidak seharusnya ditampilkan oleh perangkat yang dituju, terutama saat melakukan HTTP GET dari Android ke komputer, di mana respon yang diterima berasal dari NodeMCU.

5. SARAN

Penelitian ini pada lingkungan VirtualBox telah memiliki variabel kontrol yang sama, seperti jenis VM yang identik dan virtual NIC yang sama, namun pada percobaan pada lingkungan fisik, variabel kontrol, seperti jenis koneksi ke AP atau router, perlu disamakan untuk mendapatkan data perbandingan yang lebih jelas.

DAFTAR PUSTAKA

- [1] Alasdair Allan “List of MAC addresses with vendors identities”. [Daring] Tersedia dari: <https://gist.github.com/aallan/b4bb86db86079509e6159810ae9bd3e4> [Diakses:07/04/2022]
- [2] Android 6.0 changes. Tersedia dari : <https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html>, 2015. [Diakses:07/04/2022]
- [3] Cardenas, Edgar D. “MAC Spoofing--An Introduction”. GIAC Security Essentials Certification. SANS Institute 8 Februari 2013.
- [4] Cisco. “Configuring Port Security”. 14 November 2015.
- [5] F. Baker (June 1995). Baker, F (ed.). Requirements for IP Version 4 Routers. p. 52. RFC 1812.
- [6] Institute of Electrical and Electronics Engineers (September 2004). “Overview and Guide to the IEEE 802 LMSC” (PDF). Tersedia dari : <https://grouper.ieee.org/groups/802/802%20overview.pdf> [Diakses:15/07/2022]
- [7] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S. Cardoso, Frank Piessens, iMinds-Distrinet, KU Leuven, Univ Lyon, INSA Lyon, Inria, CITI, France. “Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms”.
- [8] Oracle Corporation. “Changelog for VirtualBox 6.1”. 19 April 2022. Tersedia dari : <https://www.virtualbox.org/wiki/Changelog-6.1> [Diakses:15/07/2022]
- [9] Xerox Corporation. “Xerox System Integration Standard 098404 - Authentication Protocol” (PDF). 1984.
- [10] Junade Ali, Vladimir Dyo “Practical Hash-Based Anonymity for MAC Addresses”. Cloudflare Inc, London, UK. University of Bedfordshire, Luton, UK. 2020
- [11] Jean-François Determe, Sophia Azzagnuni, François Horlin, Philippe De Doncker “MAC Address Anonymization for Crowd Counting”. BEAMS-EE, Université Libre de Bruxelles, 1050 Brussels, Belgium. OPERA Wireless Communications Group, Université Libre de Bruxelles, 1050 Brussels, Belgium. 2020