SKRIPSI

PENERAPAN SOFTWARE DEFINED NETWORK BERBASIS PEMROGRAMAN P4 LANGUAGES DALAM MENCEGAH SERANGAN DOS

Disusun dan diajukan oleh:

IKHSAN JIHADI
D121 17 1507

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024

LEMBAR PENGESAHAN SKRIPSI

PENERAPAN *SOFTWARE-DEFINED NETWORK* BERBASIS PEMROGRAMAN P4 *LANGUAGES* DALAM MENCEGAH SERANGAN DOS

Disusun dan diajukan oleh

IKHSAN JIHADI
D121171507

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Pada Tanggal 30 Juli 2024
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing,

Dr. Eng. Ir. Muhammad Niswar, S.T., M.IT.
NIP 19730922199931001

Ketua Program Studi,

Prof. Dr. Ir. Indrabayu, ST., MT., M.Bus.Sys., IPM, ASEAN. Eng.
NIP 197507162002121004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;
Nama : Ikhsan Jihadi
NIM : D121171507
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Penerapan *Software Defined Network* Berbasis Pemrograman P4 *Languages* dalam Mencegah Serangan DoS

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab temuan dalam skripsi ini, maka penulisan siap untuk diklarifikasi dan mempertanggungjawabkan segala risiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasikan oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 30 Juli 2024

Yang Menyatakan

Ikhsan Jihadi

ABSTRAK

IKHSAN JIHADI. *Penerapan Software Defined Network Berbasis Pemrograman P4 Language dalam Mencegah Serangan DoS* (dibimbing oleh Muhammad Niswar)

Perkembangan *Software Defined Network* (SDN) berbasis bahasa P4 (*Programming Protocol-independent Packet Processors*), yang diimplementasikan pada *switch Behavioral Model version* 2 (BMv2), merupakan kemajuan dalam lingkungan jaringan yang memungkinkan pengaturan penerusan paket pada lapisan jaringan dan *data link*. P4 adalah bahasa pemrograman yang digunakan untuk mendefinisikan perilaku paket serta mengatur penerusan paket di perangkat jaringan. Studi kasus lainnya adalah *Denial of Service* (DoS), sebuah jenis serangan yang bertujuan mengganggu layanan jaringan. DoS populer dengan berbagai variasi dan tujuan, salah satunya adalah *SYN flood*, serangan yang terjadi di lapisan jaringan.

Penelitian ini mengimplementasikan lingkungan jaringan SDN dengan menggunakan bahasa pemrograman P4 untuk melindungi dari serangan DoS, khususnya *SYN flood*. Penelitian ini melibatkan studi tentang kebutuhan perangkat dan integrasi proses lingkungan jaringan SDN dengan bahasa P4, serta metode yang diterapkan dalam bahasa P4 untuk mencegah serangan *SYN flood*.

Serangan SYN flood melibatkan pengiriman sejumlah besar paket SYN ke target serangan tanpa menyelesaikan proses *three-way handshake* pada protokol TCP (*Transmission Control Protocol*). Studi tentang dampak serangan *SYN flood* pada lingkungan jaringan dilakukan untuk mengidentifikasi metode yang dapat diimplementasikan dalam bahasa P4. Kerangka kerja penelitian ini mencakup identifikasi dan klasifikasi paket sebagai paket yang valid atau tidak valid, dengan memanfaatkan fitur dan fungsi dalam bahasa P4 untuk membatasi jumlah paket yang diterima serta memverifikasi proses pengiriman paket TCP yang sah.

Hasil penelitian menunjukkan bahwa metode yang diimplementasikan dalam bahasa P4 berhasil mencegah serangan *SYN flood*, yang terlihat dari grafik yang tidak menunjukkan lonjakan paket ke target serangan. Pengujian performa jaringan setelah penerapan metode menunjukkan tidak ada perubahan signifikan, dengan rata-rata *latency* 0.0132277 detik, *round-trip time* 0.0132276 detik, dan rata-rata *packet loss* 0%.

Kata Kunci: *SDN, P4*, *BMv2, SYN flood, DoS*

ABSTRACT

IKHSAN JIHADI. *Application of Software Defined Network Based on P4 Language Programming to Prevent DoS Attacks* (supervised by Muhammad Niswar)

The development of Software Defined Network (SDN) based on the P4 (Programming Protocol-independent Packet Processors) language, implemented on the Behavioral Model version 2 (BMv2) switch, represents an advancement in network environments that enables packet forwarding management at the network and data link layers. P4 is a programming language used to define packet behaviors and manage packet forwarding in network devices. Another case study involves Denial of Service (DoS), a type of attack aimed at disrupting network services. DoS is popular for its various types and objectives, one of which is SYN flood, an attack that occurs at the network layer.

This research implements an SDN network environment based on the P4 programming language to prevent SYN flood DoS attacks. The study involves exploring the necessary network environment and integrating SDN with P4 programming language, as well as developing methods translated into P4 to prevent SYN flood attacks.

The behavior of SYN flood attacks involves sending a large number of SYN packets to the target without completing the three-way handshake process of the TCP (Transmission Control Protocol) protocol. Studying the impact of SYN flood behavior on network environments helps identify methods translated into P4. The framework of this research identifies and categorizes packets as valid or invalid using features and functions in P4, limiting the number of packets received and verifying the legitimate TCP packet delivery process.

The results of this research show that the method implemented in P4 successfully prevents SYN flood attacks, as evidenced by graphs showing no packet spikes at the attack target. Network performance testing after implementing the method shows no significant changes, with an average latency of 0.013227 seconds, round-trip time of 0.013227 seconds, and average packet loss of 2%.

Keywords: *SDN, P4, BMv2, SYN flood, DoS*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
SDN	*Software Defined Network*
P4	*Programming Protocol-Independent Packet Processors*
DDoS	*Distributed Denial of Service*
DoS	*Denial of Service*
BMv2	*Behavioral Model version 2*
SYN Flood	*Synchronize Flood*
P4C	*P4 Compiler*
JSON	*JavaScript Object Notation*
VM	*Virtual Machine*
IP	*Internet Protocol*
MAC	*Media Access Control*
TCP	*Transmission Control Protocol*
IPv4	*Internet Protocol version 4*
DSL	*Domain-Specific Languages*

DAFTAR LAMPIRAN

KATA PENGANTAR

Segala puji dan syukur kami panjatkan ke hadirat Allah SWT, karena dengan rahmat dan karunia-Nya, kami dapat menyelesaikan skripsi ini yang berjudul ***"Penerapan Software Defined Network Berbasis Pemrograman P4 Languages dalam Mencegah Serangan DoS"***. Salawat dan salam semoga selalu tercurahkan kepada junjungan kita Nabi Muhammad SAW, yang senantiasa menjadi sumber inspirasi dan teladan untuk umat manusia. Penulisan Skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk menyelesaikan jenjang Strata-1 di Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari keterbatasan dan kelemahan yang ada dalam penyelesaian tugas akhir ini sehingga apa yang penulis berikan masih jauh dari kata kesempurnaan. Proses penyelesaian skripsi ini tidak lepas dari berbagai bantuan, dukungan, saran, dan kritik yang telah penulis dapatkan, oleh karena itu dalam kesempatan ini peneliti ingin mengucapkan terima kasih kepada :

1. Allah SWT., yang telah memampukan penulis hingga dapat menyelesaikan tugas akhir ini, karena tanpa izin dan kuasa-Nya penulis tidak akan mampu menyelesaikan tugas akhir ini.
2. Ibu dan Bapak penulis, Ibu Nursyam atas doa yang tiada henti, dukungan dan kesabarannya dalam mendidik penulis, dan Almarhum Bapak Basri yang dalam peristirahatannya mendapatkan tempat disisi Allah SWT.
3. Bapak Dr. Eng. Muhammad Niswar, S.T., M.IT., selaku dosen pembimbing dan kepala Laboratorium Ubiquitous Computing & Networking Lab (UBICON), yang senantiasa menyediakan waktu, tenaga, pikiran dan perhatian dalam mengarahkan penulis dalam penyusunan tugas akhir ini.
4. Bapak Dr. Eng. Ady Wahyudi Paundu, S.T., M.IT. dan Bapak Adnan, S.T., M.IT., Ph.D., selaku dosen penguji dalam mengarahkan penulis untuk melengkapi penyelesaian tugas akhir ini dalam bentuk kritik dan saran kepada penulis.

5. Bapak Prof. Dr. Ir. Indrabayu, ST., MT., M.Bus.Sys., IPM, ASEAN. Eng., selaku Ketua Program Studi Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bantuan, perhatian dan bimbingannya kepada penulis.
6. Ibu dan Bapak Dosen dan segenap Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas perhatian dan bantuan-nya kepada penulis selama pengerjaan tugas akhir.
7. Seluruh keluarga besar penulis yang senantiasa sabar menunggu dan selalu memberikan dukungan kepada penulis selama studi dan penyelesaian tugas akhir jenjang Strata-1.
8. Teman-teman RECOGN17ER atas lingkungan pertemanan yang diberikan yang telah mempengaruhi penulis dalam kepribadian, perilaku dan pengalaman yang dilalui bersama yang berdampak baik pada penyelesaian tugas akhir ini.
9. Seluruh pihak yang tidak sempat disebutkan yang secara langsung maupun tidak langsung membantu penulis dalam menyelesaikan skripsi ini.

Akhir kata, penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna, untuk itu segala bentuk saran, masukan, dan kritik yang membangun penulis harapkan. Penulis berharap semoga skripsi ini dapat menambah wawasan dan pengetahuan bagi pembaca.

Gowa, Juli 2024

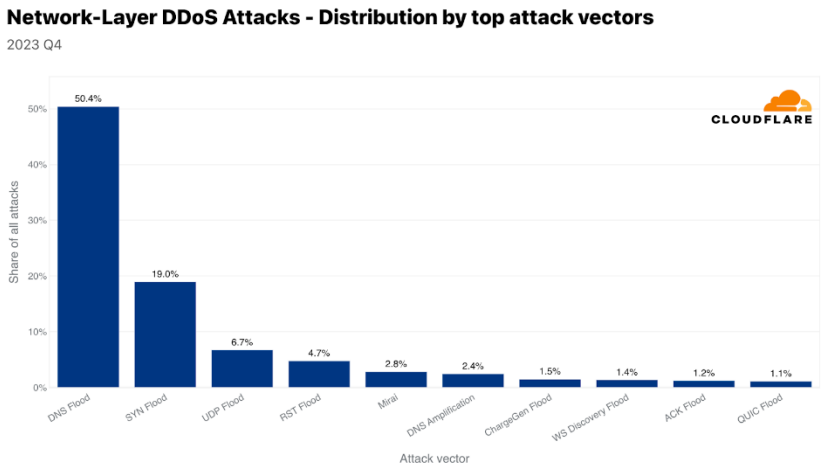
Penulis,

Ikhsan Jihadi

BAB I
PENDAHULUAN

1.1 Latar Belakang

Keamanan jaringan merupakan aspek sangat penting dalam memastikan kelancaran operasi sistem dan layanan. Salah satu bentuk ancaman keamanan jaringan dengan tujuan untuk mengganggu ketersediaan layanan dengan membanjiri jaringan atau sistem target dengan lalu lintas yang berlebihan yang disebut juga *Denial of Service* (DoS). Dari gambar vektor di bawa terdapat beberapa jenis serangan jaringan DDoS yang umum adalah *UDP flood, TCP flood, SYN flood, DNS flood* dan lain-lain, di mana penyerang mengirimkan sejumlah besar paket ke target, menyebabkan *overload* pada sistem dan menyebabkan layanan menjadi tidak tersedia bagi pengguna yang sah. (Yoachimik & Pacheco, 2024)



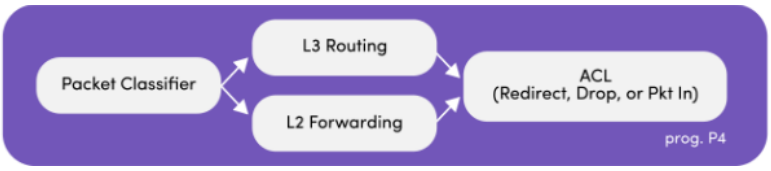
Gambar 1. 1 Vektor Jenis Serangan DDoS pada Q4 2023

Dari berbagai ancaman keamanan jaringan berkembang pula paradigma jaringan yang saat di sebut *Software Defined Network* (SDN). SDN merupakan sebuah jaringan di mana bidang kendali secara fisik terpisah dari bidang penerusan, dan satu bidang kendali mengendalikan beberapa perangkat penerusan. SDN adalah teknologi paling menonjol yang digunakan oleh para peneliti untuk membuat jaringan dapat di program. Secara umum SDN terdiri dari API *Controller*, *Southbound*, dan *Northbound*. Dalam SDN kontrol jaringan (*network control*) dan bidang data (*data plane*) terpisah secara fisik, dan bidang kontrol (*control plane*) dapat mengontrol beberapa perangkat. Bidang kontrol dapat program dengan SDN yang terpasang di dalamnya, tetapi data plane tidak fleksibel karena kemampuan

program belum mencapainya. SDN telah diadopsi secara luas di Amazon, Facebook, dan pusat data Google.(Sidiq dkk., 2020)

Salah satu perkembangan paradigma SDN yang menarik adalah penggunaan bahasa pemrograman P4 *(Programming Protocol-Independent Packet Processors)* untuk mengontrol dan mengatur perilaku *switch* jaringan. P4 adalah bahasa pemrograman khusus yang digunakan untuk memprogram *switch* dan mengontrol paket yang di kirim ke bidang data. Pemrograman bidang data yang dilakukan menggunakan P4 dapat menggambarkan paket yang diproses menggunakan rincian penerusan yang di program sehingga memungkinkan perangkat jaringan di program dengan fitur-fitur baru. Kemampuan program ini membuka peluang untuk fleksibilitas tumpukan jaringan dibandingkan dengan fungsi tradisional.(Goswami dkk., 2023)

Dalam sebuah survei yang diterbitkan IEEE (Goswami dkk., 2023) dijelaskan tentang komponen kunci dalam program P4 di antaranya *header, parser, table, action, deparser* dan *control program*. Dalam penjelasannya komponen-komponen tersebut memiliki fungsi masing dan dapat di program sesuai dengan kebutuhan jaringan. Berfokus pada komponen *control program* dikenal-kan istilah *table match-action* yang berfungsi untuk memproses bidang *header*, ekstraksi *header*, *forwarding,* modifikasi *header*, melakukan perhitungan statistik dan *dropping* paket. (Harkous dkk., 2021)



Gambar 1. 2 Program P4

Melihat kemampuan bahasa P4 dalam perkembangan SDN yang dapat memprogram *data plane* dan dari penjelasan singkat sebelumnya tentang kecenderungan serangan DoS maka dari itu penulis mengusulkan tugas akhir dengan judul "Penerapan *Software Defined Network* Berbasis Pemrograman P4 *Languages* dalam Mencegah Serangan DoS".

1.2 Rumusan Masalah

Berdasarkan latar belakang yang diuraikan, maka rumusan masalah pada penelitian ini antara lain:

1. Bagaimana penerapan paradigma jaringan *Software Defined Network* yang berbasis pemrograman P4 kedalam lingkup jaringan
2. Bagaimana merancang sistem pencegahan *Distributed Denial of Service* (DoS) berbasis pemrograman P4
3. Bagaimana mengukur keberhasilan sistem pencegahan DoS sekaligus menganalisis jaringan SDN yang berbasis pemrograman P4

1.3 Tujuan Penelitian

Tujuan penelitian ini antara lain:

1. Untuk mengimplementasikan paradigma jaringan *Software Defined Network* yang berbasis pemrograman P4
2. Merancang dan implementasi sistem pencegahan DoS berbasis pemrograman bahasa P4, serta menganalisis hasil implementasi sistem.

1.4 Manfaat Penelitian

Manfaat penelitian ini di harapkan dapat jadi acuan dalam pengembangan paradigma jaringan SDN yang berbasis bahasa P4

1.5 Batasan Masalah

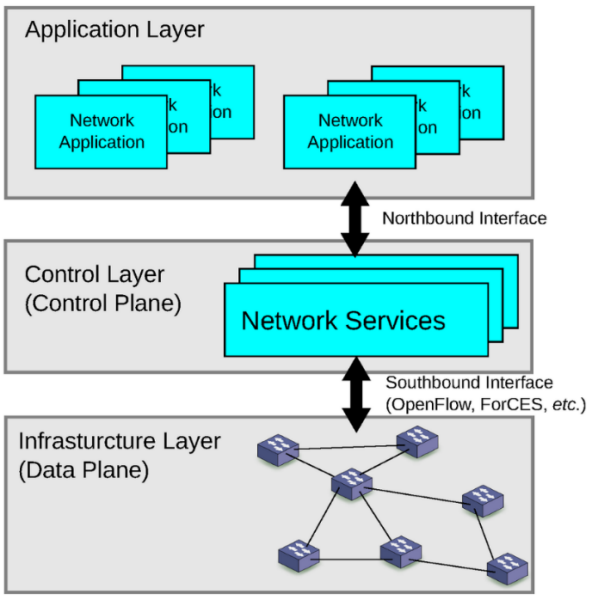
Batasan masalah yang ditentukan dalam sistem ini adalah:

1. Uji coba dan implementasi dilakukan dalam model simulasi yang menghasilkan virtual lab jaringan sebagai *sandbox*
2. Bentuk dari virtual lab adalah *DoS-script* dengan bentuk file .p4, SDN *tools* dengan tambahan instalasi pemrograman P4
3. Implementasi simulasi jaringan SDN menggunakan sistem operasi ubuntu yang instalasi dalam VM VirtualBox

BAB II
TINJAUAN PUSTAKA

2.1 Konsep Dasar Softwire Defined Network

Software-Defined Network (SDN) dengan jelas memisahkan bidang data (*data plane*) dari bidang kendali (*kontrol plane*) dan memfasilitasi implementasi perangkat lunak dari aplikasi jaringan yang kompleks di atasnya.



Gambar 2. 1 *A three-layer SDN Architecture*

Gambar 2.1 mengilustrasikan *framework* SDN yang terdiri dari tiga lapisan. Lapisan paling bawah adalah lapisan infrastruktur, disebut juga *data plane*. Ini terdiri dari elemen jaringan penerusan. Tanggung jawab pesawat penerus terutama meneruskan data, serta memantau informasi lokal dan mengumpulkan statistik. Lapisan kendali, disebut juga bidang kendali. Ia bertanggung jawab untuk memprogram dan mengelola pesawat penerusan. Untuk itu, ia memanfaatkan informasi yang disediakan oleh bidang penerusan dan menentukan operasi dan perutean jaringan. Ini terdiri dari satu atau lebih pengontrol perangkat lunak yang berkomunikasi dengan elemen jaringan penerusan melalui antarmuka standar, yang disebut sebagai antarmuka arah selatan. *OpenFlow*, yang merupakan salah satu antarmuka arah selatan yang paling banyak digunakan, terutama mempertimbangkan *switch*, sedangkan pendekatan SDN lainnya mempertimbangkan elemen jaringan lain, seperti *router*.

Lapisan aplikasi berisi aplikasi jaringan yang dapat memperkenalkan fitur-fitur jaringan baru, seperti keamanan dan pengelolaan, skema penerusan, atau membantu lapisan kontrol dalam konfigurasi jaringan. Lapisan aplikasi dapat menerima tampilan jaringan yang abstrak dan global dari pengontrol dan menggunakan informasi tersebut untuk memberikan panduan yang tepat ke lapisan kontrol. Antarmuka antara lapisan aplikasi dan lapisan kontrol disebut sebagai antarmuka arah utara.

2.2 Programming Protocol-independent Packet Processors (P4)

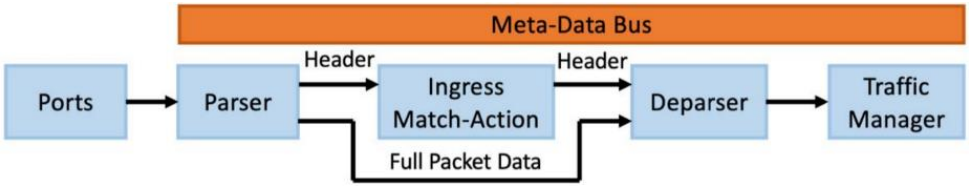
Software-Defined Network mendorong inovasi *Domain-Specifik Languages* (DSL) dan arsitektur *switch. Programming Protocol-independent Packet Processors* (P4) adalah *Domain-Specific Languages* untuk memprogram *data plane* pada *switch* yang dapat di program. P4 mendefinisikan perilaku *data plane* dari *switch* yang dapat di program dengan primitif, terlepas dari arsitektur yang mendasarinya. (Hang dkk., 2019)

Berdasarkan sebuah tesis ada 3 tujuan yang harus dipenuhi oleh bahasa P4 (Beausencourt, 2023), yaitu:

1. *Reconfigurability* artinya perilaku unit tentang bagaimana paket diteruskan dan dapat diubah bahkan setelah program diterapkan.
2. *Protocol Independent* artinya *instance* yang di eksekusi tidak boleh terikat pada format tertentu, namun harus bekerja berdasarkan *header* yang di ekstraksi oleh *parser* dan mendefinisikan *match-action tables.*
3. *Target Independent* artinya seperti dengan bahasa pemrograman C, pemrograman P4 tidak harus berurusan dengan detail spesifik perangkat keras dari perangkat target.

Arsitektur P4 bekerja dengan *Programmable Blocks*. Ini termasuk *Parser, Deparser, Ingress Control Flow* dan *Egress Control Flow.* Setelah sebuah paket memasuki target, paket tersebut diinisialisasi dengan *intristic Metadata.* Ini adalah informasi yang bergantung pada perangkat keras seperti port tempat paket diterima atau *timestamp.* Ada juga kemungkinan tambahan *user-defined metadata* ke paket saat diproses oleh target. Hal ini dapat berguna dalam menentukan cara meneruskan paket.(Beausencourt, 2023)

Jalur umum pada saat pengolahan data suatu paket dapat dilihat pada gambar 2.3. *Payload* atau *Full Packet Data* pada Gambar 2.3 tidak diproses dalam *Match-Action*. Hanya *header* yang berperan dalam pemrosesan dan penerusan *control block*. (Beausencourt, 2023)



Gambar 2. 2 Alur paket di P4

Tabel 2.1 berikut mencantumkan tipe data penting dan *keywords* P4. (Beausencourt, 2023)

Nama	**Code-P4**	**Definisi**
Boolean	Bool	Nilai : true atau false
Integer	int	32-bit Integer
Bit	bit<x>	Data Type, yang berisi x bit. X harus berupa bilangan bulat positif
Struct	struct	Pembuatan tipe data yang ditentukan pengguna
Error	error	Kesalahan sinyal
Definition	#define	Seperti di C: instruksi pra-prosedur yang sebanding dengan *Alias* untuk Linux
Type definition	typedef	Membuat tipe penting seperti alamat IPv4
Header	header	Untuk menentukan header protokol
Constant	const	*Constant* diikuti berdasarkan tipe data, nama dan nilai
Control block	control	Semua *control block* P4 kecuali Parser
Parser	parser	Deklarasi Parser
Tables	table	Deklarasi sebuah Table
Function	action	Function sebanding dengan fungsi bahasa lain

Tabel 2. 1 tipe data P4

Salah tujuan dari P4 disebutkan tentang independensi protokol, tidak ada protokol jaringan yang diimplementasikan secara *default.* Harus dibuat dalam kode menggunakan *Request for Comments* (RFC) yang sesuai atau standar lainnya. Dan seperti yang dijelaskan sebelumnya, P4 menawarkan kemungkinan untuk merancang dan mengimplementasikan pemrosesan dan penerusan paket yang sepenuhnya ditentukan sendiri. Sumber daya infrastruktur jaringan bisa digunakan lebih efisien, karena hanya protokol, *action* atau hal serupa yang diperlukan yang disertakan dan *table* dikonfigurasi dengan sangat tepat. Karena program P4 dapat ditulis sebagian besar secara independen dari perangkat keras dan kompilator menangani penghubungan kode dan perangkat keras, pengembangan lebih lanjut yang mandiri dari bahasa pemrograman dan perangkat keras juga memungkinkan.(Beausencourt, 2023)

2.3 Behavioral Model version 2 (BMv2)

BMv2 (*Behavioral Model version 2*) adalah *software-defined switch* dari *P4 Language Consortium. Software switch* ditulis dalam C++11. Dibutuhkan sebagai masukan file JSON yang dihasilkan dari program P4 oleh P4 *Compiler* dan menafsirkannya untuk mengimplementasikan perilaku *packet-processing* yang ditentukan dari program P4. Pelaku pemrosesan di dalamnya ditentukan oleh program P4. BMv2 tidak dirancang untuk digunakan dalam lingkungan produksi. Hal ini dimaksudkan untuk digunakan sebagai alat untuk pengembangan, pengujian dan debugging. Dengan demikian kinerja BMv2 dalam throughput dan latency jauh lebih rendah dengan perangkat lunak tingkat produksi seperti *Open vSwitch* (*p4lang/behavioral-model: The reference P4 software switch*, t.t.). Dalam sebuah repositori di GitHub BMv2 memiliki enam *control block* di antaranya *Parser, Checksum Verification, Ingress Processing, Egress Processing, Checksum Computation,* dan *Deparser.*

```
169     V1Switch(
170     MyParser(),
171     MyVerifyChecksum(),
172     MyIngress(),
173     MyEgress(),
174     MyComputeChecksum(),
175     MyDeparser()
176     ) main;
```

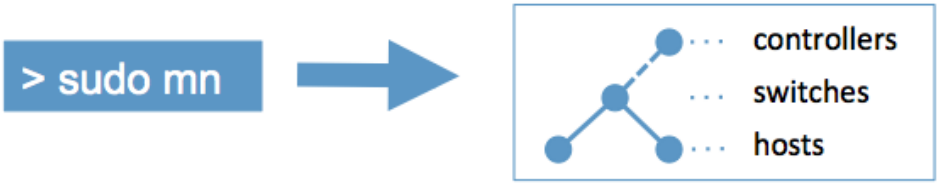
Gambar 2. 3 Control Block BMv2

Berikut penjelasan dari ke-enam *control block*:

1. Parser akan menerima paket mentah yang masuk dan mengurai *header-header* yang ada di dalam paket tersebut.
2. Checksum Verification akan menverifikasi paket dengan memeriksa checksum agar memastikan tidak ada korupsi data selama transmisi.
3. Ingress Processing akan mengelola logika *forwarding* dan pengolahan paket masuk dapat diarahkan ke *port* tertentu, dimodifikasi, atau diproses lebih lanjut berdasarkan logika ingress.
4. Egress Processing akan mengelola logika *forwarding* dan pengolahan paket keluar yang mampu menambahkan tak, modifikasi header, atau penentuan port keluaran dilakukan di sini.
5. Checksum Computation akan menghitung checksum untuk memastikan integrasi paket yang memastikan data saat transmisi ke penerima.
6. Deparser akan menggabungkan kembali header-header dan playload menjadi satu paket lengkap yang siap untuk dikirimkan ke jaringan.

2.4 Mininet

Mininet adalah emulator jaringan yang membuat jaringan host virtual, sakelar, pengontrol, dan tautan. Host Mininet menjalankan perangkat lunak jaringan Linux standar, dan *switch*-nya mendukung OpenFlow untuk peruteran khusus yang sangat fleksibel dan Jaringan Buatan Perangkat Lunak. Mininet mendukung penelitian, pengembangan, pembelajaran, pembuatan prototipe, pengujian, *debugging*, dan tugas lainnya yang dapat memperoleh manfaat dari memiliki jaringan eksperimental lengkap di laptop atau PC lainnya.



Gambar 2. 4 Gambaran umum integrasi mininet

Ada banyak hal yang bisa dilakukan mininet, di antaranya:

1. Memungkinkan beberapa pengembang secara bersamaan untuk bekerja secara independen pada *topology* yang sama
2. Mendukung pengujian regresi tingkat sistem , yang dapat diulang dan dikemas dengan mudah
3. Memungkinkan pengujian *topology* yang kompleks , tanpa perlu menyambungkan jaringan fisik
4. Termasuk CLI yang sadar *topology* dan sadar OpenFlow, untuk melakukan debug atau menjalankan pengujian di seluruh jaringan
5. Mendukung *topology* khusus yang sewenang-wenang , dan mencakup serangkaian *topology* berparametri dasar dapat digunakan langsung tanpa pemrograman
6. Menyediakan API Python yang mudah dan dapat diperluas untuk pembuatan dan eksperimen jaringan
7. Mininet menyediakan cara mudah untuk mendapatkan perilaku sistem yang benar (dan, sejauh didukung oleh perangkat keras Anda, kinerjanya) dan untuk ber-eksperimen dengan *topology*

Jaringan Mininet menjalankan kode asli termasuk aplikasi jaringan Unix/Linux standar serta kernel Linux dan tumpukan jaringan asli (termasuk ekstensi kernel apa pun yang mungkin Anda miliki, selama ekstensi tersebut kompatibel dengan namespace jaringan). Oleh karena itu, kode yang Anda kembang-kan dan uji di Mininet, untuk pengontrol OpenFlow, *switch*, atau host yang dimodifikasi, dapat berpindah ke sistem nyata dengan sedikit perubahan , untuk pengujian dunia nyata, evaluasi kinerja, dan penerapan. Yang penting, ini berarti bahwa desain yang berfungsi di Mininet biasanya dapat berpindah langsung ke sakelar perangkat keras untuk penerusan paket dengan laju saluran. (*Mininet Overview - Mininet*, 2022)

2.5 Oracle VM VirtualBox

Gambar 2. 5 Oracle VM VirtualBox

VirtualBox adalah *virtualizer* lengkap untuk keperluan umum untuk perangkat keras x86, ditargetkan untuk penggunaan *server, desktop* dan *embedded.* VirtualBox adalah produk virtualisasi x86 dan AMD64/Intel64 yang merupakan produk yang kaya fitur dan berkinerja tinggi yang tersedia bebas sebagai *Open Source Software.* Saat ini, VirtualBox berjalan pada *host* Windows, Linux, macOS, dan Solaris dan sejumlah besar *guest operating systems* termasuk OpenSolaris dan lainnya. VirtualBox sedang dikembangkan secara aktif dengan rilis yang sering dan memiliki daftar fitur yang terus bertambah, *guest operating systems* yang didukung, dan platform yang dijalankannya. Salah satu fitur dan keunggulan VirtualBox ykni mampu membentuk lingkungan virtaul yang dapat dijadikan sebagai SandBox yang dalam hal ini merupakan lingkungan virtual yang terisolasi dari jaringan, sistem dan program lain. (*Oracle VM VirtualBox*, 2023)

Beberapa kegunaan dalam menggunakan Oracle VirtualBox diantaranya:

1. Menjalankan beberapa sistem operasi secara bersamaan
2. Instalasi perangkat lunak lebih mudah
3. Dapat melakukan pengujian dan pemulihan bencana
4. Membangun dan menguji layanan jaringan multi-node
5. Dengan fitur snapshot depat menyimpan status tertentu dari mesin virtual dan kembali ke status tersebut
6. Virtualisasi dapat secara signifikan mengurangi biaya perangkat keras dan listrik

2.6 XTerm

XTerm adalah emulator terminal untuk Sistem X window. XTerm dirancang sebagai " Proyek Bahasa dan Kegiatan Produktif" tetapi tidak terbatas pada itu. Bahkan dapat dengan mudah disesuaikan dengan proyek terminologis yang berbeda karena bukan hanya *termbase*, tapi dapat digunakan sebagai Term Base Management System (TBMS). (Piccioni & Zanchetta, t.t.)

Seluruh sistem terdiri dari 4 komponen utama :

1. Mesin basis data (mySQL, Oracle, Access) yang menangani penanganan data mentah tingkat rendah;
2. Xterm.NET, lingkungan grafis untuk penyisipan data, manajemen *termbase*, permintaan dan visualisasi;
3. Satu atau lebih file konfigurasi XML yang mendefinisikan data struktur *termbase* (jumlah yang hampir tidak terbatas *termbase* dengan struktur berbeda dapat di *hosting* mesin yang sama);
4. XTerm.portal, aplikasi web yang menyediakan layanan umum akses ke *termbase(s)* melalui mesin *query* komprehensif.

2.7 Denial of Service (DoS) dan Distributed Denial 0f Service (DDoS)

Masalah keamanan serangan DoS atau varian terdistribusinya (DDoS) merupakan ancaman paling besar bagi jaringan berbeda dengan serangan lainnya. Ini menyiratkan bahwa serangan DoS atau DDoS berhasil memiliki kecenderungan atau kemampuan untuk mengganggu jaringan sepenuhnya dengan menonaktifkan *controller* atau *switch*. Dalam serangan-serangan ini, *switch* tidak dapat mengirimkan paket dengan tepat seperti yang diperlukan sehingga mengakibatkan jaringan gagal dan *subsequently*. (Tamakloe dkk., 2023)

DoS atau DDoS attack adalah serangan *cyber* di mana lalu lintas palsu terus dikirim ke server atau sistem, server tidak dapat menangani semua lalu lintas dan server atau sistem menjadi *down*, serta tujuan dari serangan *Denial 0f Service* adalah untuk membuat sistem target tidak dapat diakses atau untuk meminta layanan. Serangan *Denial of Service* telah sangat mengganggu ketersediaan jaringan selama beberapa dekade dan masih belum ada mekanisme pertahanan yang efektif untuk melawannya(Ariyadi dkk., 2022)

DoS atau DDoS merupakan bentuk serangan dengan cara membanjiri data (*flooding*) yang berusaha membuat suatu host atau *service* menjadi tak dapat diakses oleh user yang berhak. Salah satu jenis serangan yang umum adalah UDP flood merupakan serangan yang bersifat *connectionless*, yaitu tidak memperhatikan apakah paket yang dikirim diterima atau tidak. flood attack akan menempel pada servis UDP *chargen* di salah satu mesin, yang untuk keperluan "percobaan" akan mengirimkan sekelompok karakter ke mesin lain, yang di program untuk meng-*echo* setiap kiriman karakter yang di terima melalui servis *chargen*. Karena paket UDP tersebut di *spoofing* antara ke dua mesin tersebut, maka yang terjadi adalah banjir tanpa henti kiriman karakter yang tidak berguna antara ke dua mesin tersebut.(Serangan & Yasin, 2020)

SYN flood adalah salah satu jenis paket dalam protokol *Transmission Control Protocol* (TCP) yang dapat digunakan untuk membuat koneksi antara dua host dan dikirimkan oleh host yang hendak membuat koneksi, sebagai langkah pertama pembuatan koneksi dalam proses "TCP *Three-way Handshake*". Dalam cara kerja dari *three-why-handshake* dalam protokol TCP, *client* yang melakukan usaha agar bisa terhubung dengan server berarti mengirimkan sebuah pesan request kepada server agar menjadi tanda bahwa client dan server telah terhubung, pesan yang dimaksud ialah SYN. Dan server otomatis menerima kemudian mengonfirmasi *request* dari *client* dan akan mengirimkan ulang SYN-ACK pada *client*. *Client* yang sudah mendapatkan SYN-ACK dari server kemudian merespon dengan pesan ACK, maka saat itu bisa disebut bahwa koneksi dari *client* ke server telah terjadi. (Kalabo & Dwi Setiawan Sumadi, 2022)

2.8 Wireshark

Gambar 2. 6 Logo Wireshark

Wireshark adalah penganalisis protokol jaringan yang memungkinkan melihat apa yang terjadi di jaringan pada tingkat mikroskopis. Ada banyak fitur dalam Wireshark di antaranya sebagai berikut (*Wireshark*, 2024):

1. Inspeksi mendalam terhadap ratusan protokol, dan lebih banyak lagi yang ditambahkan setiap saat
2. Pengambilan langsung dan analisis *offline*
3. *Browser* paket tiga panel standar
4. *Multi-platform*
5. Filter tampilan
6. Dukungan dekripsi untuk banyak protokol

Wireshark dapat digunakan untuk mengidentifikasi dan memecahkan masalah terkait dengan tabel aliran pada *switch*. Penggunaan aplikasi wireshark telah menjadi standar *de facto* dibanyak industri dan institusi pendidikan.

2.9 Metode Keamanan

Serangan Dos (*Denial of Service*) adalah jenis *cyber attack* yang bertujuan untuk mengganggu fungsi normal jaringan dengan membanjiri servernya dengan permintaan dalam jumlah besar. Dalam lingkungan SDN berbasis pemrograman P4, dengan adanya bahasa P4 memungkinkan untuk memprogram *switch* untuk mengatasi banjir serangan pada lingkungan jaringan. (Beausencourt, 2023)

Pemrograman P4 dalam pengembangannya bisa mengatur cara, jenis dan seperti apa paket akan diterima dan diteruskan. Jika jenis serangan DoS diklasifikasikan serangan DoS bisa dikelompokkan berdasarkan OSI Layer (*Open Systems Interconection Layer*). Dari tujuh OSI layer yang diperkenalkan dalam penelitian ini mefokuskan pada serangan yang terdapat pada lapisan jaringan (network layer) dan lapisan tranportasi (transport layer), dimana kedua lapisan ini

sangat berhubungan jika terkait dengan jenis serangan DoS. Salah satu jenis serangan yang terjadi pada lapisan ini dan yang akan menjadi bahan uji untuk penelitian ini adalah SYN flood atau yang menggunakan kelemahan protokol TCP. (Gupta & Dahiya, 2021; Nisa & Ramadona, 2023; *What is OSI Model? - Layers of OSI Model*, 2024)

Dari perilaku DoS yang membanjiri sebuah lingkungan jaringan dan perilaku paket yang bisa dideteksi, perlu ada batasan dan aturan penerusan paket yang diimplementasikan menggunakan pemrograman P4. Struktur *code* pemrograman P4 memungkinkan membuat aturan yang membuat sebuah paket diterima atau tidak diterima. Salah satu metode yang bisa diimplementasikan dengan mengatur agar paket yang berasal dari penyerang tidak diterima atau diteruskan ke tujuan. Metode yang diimplementasikan kali ini adalah metode yang mampu untuk mengetahui dan bisa memverifikasi paket mana saja yang bisa dikategorikan untuk diterima. Terkait ini *Bloom Filter* memungkinkan untuk diimplementasikan.

Bloom filter merupakan sebuah metode atau struktur data probabilistik yang digunakan untuk menguji keanggotaan suatu elemen dalam sebuah set. Hal ini menyediakan metode efisien untuk memeriksa apakah suatu elemen mungkin ada dalam set, berfungsi juga sebagai teknik atau pendekatan untuk memverifikasi keanggotaan. *Bloom filter* ada kemungkinan kegagalan (*false positive*). *Bloom filter* menggunakan memori relatif kecil dikarenakan mampu mengeksploitasi teknik *hashing* dengan bisa lebih dalam membagi lagi keanggotaan suatu elemen. Dalam hal *false positive* pada *bloom filter* memungkinkan diatasi dengan *hashing.* (Malchiodi dkk., 2024)

Metode pencegahan yang digunakan pada penelitian kali ini adalah mendefinisikan metode *Bloom Filter* menggunakan bahasa P4 yang diprogram di *switch* BMv2 dalam sebuah lingkungan jaringan SDN. Implementasi metode tersebut ditujukan untuk pencegahan terhadap jenis serangan DoS *SYN flood*, yang mempertimbangkan besaran paket TCP dan pencocokan paket yang bisa diteruskan.

2.10 Metode Analisis

Metode analisis pada penelitian kali ini dilakukan dengan beberapa cara dan parameter dalam menganalisis lingkungan jaringan. Metode analisis yang digunakan berdasarkan perilaku serangan DoS yang di implementasikan. Maka dari itu metode analisis merupakan skenario pengujian dengan tujuan membandingkan perilaku dari lingkungan jaringan sebelum implementasi metode keamanan dan setelah implementasi metode keamanan. Berdasarkan tujuan tersebut yang dijadikan parameter pengujian adalah *throughput, latency (delay), Round-trip Time* dan *packet loss.*