BAB II

LANDASAN TEORI

2.1 Kajian Penelitian Terdahulu

Tabel berikut akan merangkum beberapa penelitian sebelumnya yang menggunakan algortima Naïve Bayes dan penggunaan teknologi Honeypot dalam mendeskripsikan dan mengklasifikasikan serangan siber.

Tabel 2. 1 Tinjauan Referensi Penelitian Terdahulu

No	Penulis	Tahun	Judul	Metode	Jurnal
1	Desi Kurnia Nurilahi, Rizal Munadi, Syahrizal, dan Al Bahri (Universitas Syiah Kuala, Indonesia)	2021	Penerapan Metode *Naïve Bayes* pada *Honeypot Dionaea* dalam Mendeteksi Serangan	Algoritma : *Naïve Bayes* Honeypot : *Dionaea*	ELKOMIKA : Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika Vol. 10 No. 2 April (2022) – ISSN(p):2338-8323 | ISSN(e):2459-9638 (Sinta 2)
	Hasil :				

	Tujuan dari penelitian ini adalah untuk mengevaluasi sistem Honeypot Dionaea dalam menangani serangan port scanning yang dilancarakan. Dengan menggunakan tools Nmap, hasil pengujian penyerangan akan dievaluasi berdasarkan data log yang berisi aktivitas serangan yang kemudian akan diklasifikasikan dengan metode Naïve Bayes menggunakan perangkat lunak WEKA. Hasil analisis yang diperoleh, dapat digunakan sebagai pedoman dalam meningkatkan sistem keamanan jaringan. Lalu, hasil dari penelitian tersebut, ditemukan port terbuka sebanyak 359 data. Dari data tersebut, peneliti melakukan uji klasifikasi denga perangkat lunak WEKA dan penerapan metode Naïve Bayes. Hasil uji klasifikasi diperoleh nilai akurasi sebesar 86,2% dengan nlai rata-rata Precision sebesar 0,885%, Recall sebesar 0,862%, dan F-Measure sebesar 0,849%. Hasil ini menunjukkan penerapan metode Naïve Bayes berhasil mengklasifikasikan potensi serangan yang dilakukan berdasarkan Teknik Port Scanning. **Kekurangan/Keterbatasan** : Penulis menyebutkan bahwa hasil uji klasifikasi pada model yang sudah dibangun memang sudah menunjukkan angak persentase yang baik, namun penulis menyebutkan bahwa apabila memiliki persentase pembagian antara data testing dengan data training lebih dari yang diujikan, maka akan menghasilkan output yang berbeda dan memungkinkan menaikkan nilai persentasi akurasi dan beberapa parameter uji lainnya.

No	Penulis	Tahun	Judul	Metode	Jurnal
2	Bayu Setyanto Pamungkas dan Dr. Irwan Sembiring, ST., M.Kom. (Universitas Kristen Satya Wacana)	2023	Analisis Serangan Cyber Menggunakan Honeypot Pada Web Berbasis Cloud	T-Pot Honeypot	Jurnal Indonesia : Manajemen Informatika dan Komunikasi Vol 4 No.3, September (2023) – E-ISSN:2723-7079, P-ISSN:2776-8074 (Sinta 4)
	Hasil : Penelitian ini dilakukan implementasi sebuah sistem keamanan untuk mendeteksi, mencegah serangan *cyber* yang terjadi ada web berbasis *cloud* dengan *intrusion detection and prevention system* (*idps*) dalam *T-Pot Honeypot* yang dijalankan menggunakan *virtual machine* pada layanan *cloud Microsoft Azure*, serta menganalisis tingkat keberhasilan dari sistem yang dibuat untuk mendeteksi serangan *cyber* yang terjadi. Sehingga penelitian ini akan dilalukan untuk menguji sistem *T-Pot Honeypot* berbasis pada *Cloud*, serangan yang diuji coba adalah *Port Scanning, Brute Force*				

	Attack, Malware Attack, DoS, dan *DdoS Attack.* Data hasil serangan ditampilkan dalam *kibana dashboard* secara visual berbentuk diagram. Dalam *kibana dashboard* tercatat *username* dan *password* yang digunakan ketika terdapat anomali, segala aktivitas tercatat oleh *honeypot* dan di visualkan dalam *Kibana Dashboard*. Hasil penelitian, pengujian dan analisis deteksi serangan siber menggunakan *Honeypot*, disimpulkan bahwa sistem *T-Pot Honeypot* dapat mendeteksi secara cepat akurat anomali yang terjadi sehingga *cyberattack* dapat diantisipasi terlebih dahulu, dengan cara memberikan penyerang akses masuk kedalam sistem *T-Pot Honeypot* melalui *port* yang terbuka sehingga penyerang berpikir telah berhasil masuk kedalam sistem utama. *Tools* yang disediakan oleh *T-Pot Honeypot* sangat membantu mendeteksi, menganalisa serangan dengan total serangan terdeteksi oleh sistem *T-Pot Honeypot* sebanyak 64017 serangan dengan perincian terdeteksi 39637 *Cowrie Attack*, 6797 *Ipphoney Attack*, 6483 *Citrix Honeypot Attack*, 5232 *Tanner Attack*, 5363 *Honeytrap Attack*, 359 *Dionaea Attack*, 111 *Ciscoasa Attack*, 19 *Conpot Attack*, 10 *AdbHoneyAttack*, 10 *Elasticpot Attack.* Namun kelemahan yang dimiliki dari sistem *T-Pot Honeypot* ini adalah terdapat 1904 atau 3% hit yang tidak terdeteksi oleh *T-Pot Honeypot*, yang dimana ini dapat menjadikan celah untuk *attacker* mengetahui bahwa mereka terjebak di dalam sistem *Honeypot* dan mencari data untuk bisa keluar dari jebakan *Honeypot*. **Kekurangan / Keterbatasan :**

	Penulis menyebutkan bahwa terdapat 1904 atau setidaknya 3% hit atau traffic yang masuk, dan tidak terdeteksi sistem T-Pot Honeypot, yang dimana ini akan menjadikan celah bagi attacker di masa mendatang.

No	Penulis	Tahun	Judul	Metode	Jurnal
3	Toriyansa Natanegara, Yusuf Muhyidin dan Dayan Singasatia (STT Wastukencana Purwakarta)	2023	Implementasi *Honeypot Cowrie* dan *Snort* Sebagai Alat Deteksi Serangan Pada Server	Honeypot Cowrie dan Snort	JATI (Jurnal Mahasiswa Teknik Informatika) – : JATI : Vol. 7 No. 3 (2023) (Sinta 1)
	Hasil : Penelitian ini menggunakan metode *Network Development Life Cycle (NDLC)* yang merupakan metode pengembangan dan perancangan sistem jaringan computer dan memungkinkan sistem yang dirancang atau dikembangkan untuk dimonitor agar bisa ditentukan keefektifiannya. Metoda ini sendiri memiliki 6 tahapan, yaitu Analisa, Desain, Simulasi Prototipe, Implementasi, Monitoring, dan Management. Berdasarkan penelitian yang dilakukan, disimpulkan bahwa Honeypot Cowrie dan IDS				

	Snort efektif dalam mendeteksi serangan yang masuk ke dalam sistem server pada sistem operasi Ubuntu 20.04.2 LTS. Kedua sistem ini mampu mendeteksi serangan berupa Dos/DdoS, serta serangan Bruteforce Attack untuk Snort dan Serangan Bruteforce SSH pada Honeypot Cowrie. Lebih detailnya, dari hasil simulasi penyerangan untuk menguji Honeypot Cowrie, sistem ini tidak dapat mendeteksi jenis serangan DDOS Attack yang masuk kedalam sistem dan mempelajari serangan yang ada. Sedangkan, Snort dapat mendeteksi jenis serangan DDOS Attack maupun Brtueforce Attack, tetapi Snort tidak dapat menindaklanjuti serangan yang masuk. **Keterbatasan / Kekurangan :** Penulis menyebutkan belum menemukan suatu paket sistem yang bisa diuji keabsahan fungsinya dalam mengenali, mendeteksi dan menindaklanjuti sebuah serangan siber dengan jenis yang bermacam-macam.

2.2 Tinjauan Pustaka

2.2.1 Serangan Siber

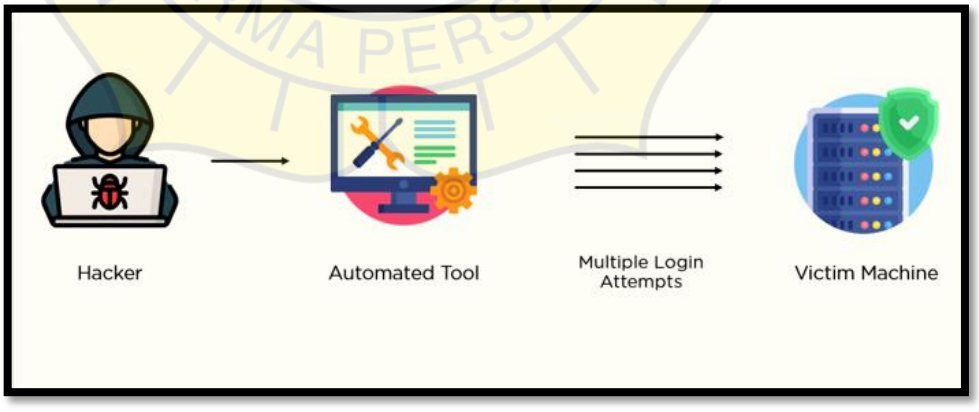
Menurut (Dermawan et al., 2023) Serangan siber dapat didefinisikan sebagai serangan yang didorong oleh suatu penyerang dengan bantuan lebih dari satu komputer atau jaringan. Serangan semacam ini dapat dengan disengaja untuk mempengaruhi suatu sistem tertentu untuk dapat mencuri data. Penjahat dunia maya biasa melakukan berbagai pendekatan dalam melancarkan serangan dunia maya atau biasa disebut serangan siber (*cyber attack*). (Hapsari & Pambayun, 2023) menjelaskan serangan siber merupakan salah satu *cyber crime* atau kejahatan dunia

maya yang dilakukan dengan menggunakan perangkat teknologi informasi dan komunikasi sebagai alat atau target dari kejahatan tersebut. Hal ini diperkuat dengan pernyataan (Razzaq et al., 2022) terkait Serangan Siber atau *Cyber Attack* merupakan jenis kejahatan siber yang menyerang dan mengganggu informasi yang ada dalam suatu komputer dengan sengaja. Tindakan ini biasanya memiliki tujuan untuk mengganggu baik secara fisik bahkan sistem dan perangkat lunak atau software yang ada di dalam suatu komputer yang menjadi target penyerangan. *Cyber crime* terjadi disebabkan beberapa faktor, seperti anonimitas di dunia digital, teknologi yang semakin mutakhir, kesenjangan sosial, insentif finansial, dan kurangnya regulasi serta penegakan hukum yang memadai di banyak negara. Selain itu, penegakan hukum terhadap tindakan kejahatan siber juga masih terbatas oleh keterbatasan sumber daya dan kemampuan teknologi yang dimiliki oleh pihak penegak hukum. Kejahatan siber atau *cybercrime* di Indonesia yang sering terjadi antara lain *brute force attack*, *malware*, *phising*, *DdoS (Denial Distributed of Service)*, *cyberstalking*, *anonymous*, *cyberbullying*, *financial crime*, dan *core infrastructure attack*.

2.2.2 Brute Force Attack

Menurut (Indra Gunawan, 2016), Brute Force adalah sebuah algoritma yang memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang sangat jelas. Penyelesaian permasalahan kode cracking dengan menggunakan algoritma brute force akan menempatkan dan mencari semua kemungkinan kode dengan masukan karakter dan panjang kode tertentu dengan banyak kombinasi inputan. Hal ini diperkuat dengan pernyataan (Pramaditya, n.d.), bahwa Brute Force Attack merupakan salah satu jenis serangan siber yang menggunakan percobaan

terhadap setiap kemungkinan kunci yang ada. Pendekatan ini pada awalnya merujuk pada perangkat lunak atau software yang mengandalkan kekuatan pemrosesan komputer dibandingkan kecerdasarn manusia. Sebagai contoh, untuk menyelesaikan sebuah persamaan X+2X-4=7, dimana nilai X adalah sebuah integer, dengan menggunakan Teknik algoritma brute force, penggunanya hanya dituntut untuk membuat suatu program perangkat lunak yang mencoba setiap kemungkinan nilai integer yang ada sehingga memungkinkan persamaan diatas akan bernilai valid. Jadi bisa disimpulkan bahwa Teknik ini merupakan suatu metoda yang bisa digunakan dalam kejahatan siber, dengan menggunakan metoda menjebol kode rahasia dengan mencoba semua kemungkinan kunci yang ada. Kelayakan dari sebuah brute force attack ini sangat bergantung pada panjangnya cipher (teks enkripsi) yang diinginkan, dan juga jumlah komputasi yang tersedia dari sisi penyerang. Untuk dapat lebih memahami pemahaman terkait Brute Force Attack, berikut penulis jelaskan dalam bentuk visual dan deskriptif terkait serangan siber terkait.



Gambar 2. 1 Brute Force Attack

Pada dasarnya cara kerja *brute force attack* ini memiliki 4 fase yang akan selalu mengalami proses ‘loop’ hingga suatu kondisi terpenuhi atau masuk kedalam fase *terminate*.

a. Attacker akan mencoba membuat sebuah semacam ‘List’ terkait kombinasi dari ‘Username’ dan ‘Password’ dari suatu service ynag akan menjadi objek serangan siber ini.
b. Lalu, proses selanjutnya adalah Attacker akan mencoba membuat sebuah ‘script automation’ dimana proses dasarnya adalah layaknya normal user melakukan proses otentikasi, yaitu melakukan action Login.
c. Proses ini akan terus terjadi hingga suatu kondisi terpenuhi, yaitu attacker berhasil ‘menebak’ kombinasi dari username dan password yang sesuai dengan service yang dituju.
d. Kondisi Tercapai, atau *Terminate* (terpenuhi atau tidak terpenuhi dengan suatu kondisi tertentu).

2.2.3 Lanskap Keamanan Siber Indonesia

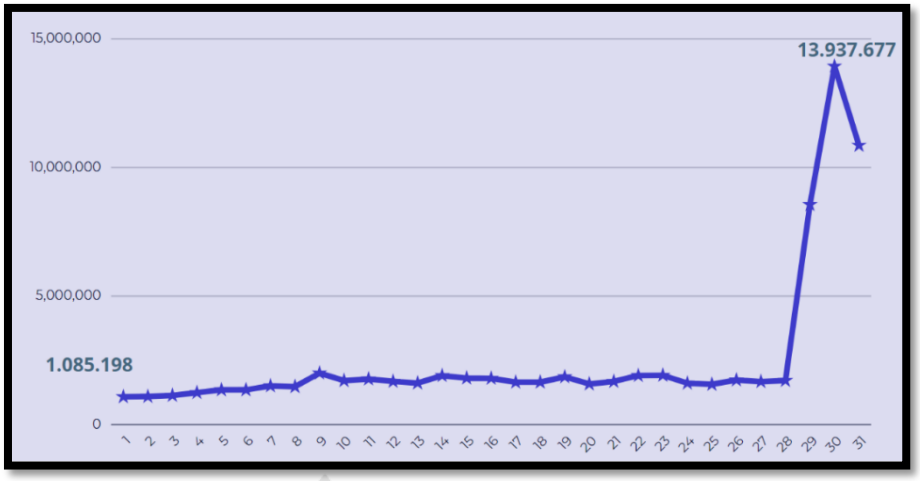
Berdasarkan data statistik yang penulis riset, menurut Badan Siber dan Sandi Negara, lewat (LANSKAP KEAMANAN SIBER INDONESIA, n.d.), menyebutkan bahwa total trafik anomali di Indonesia selama tahun 2023 adalah sebesar **403.990.813 juta anomali.**

Gambar 2. 2 Trafik Anomali Serangan Siber di Indonesia Tahun 2023

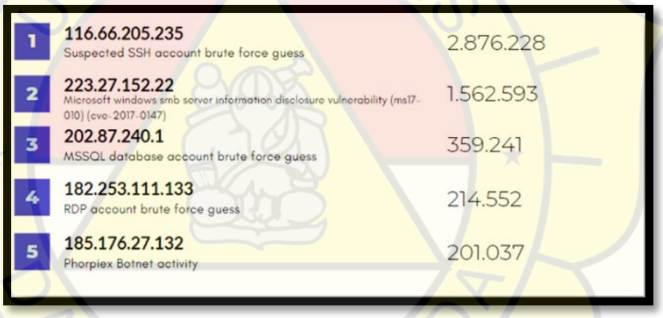
Anomali trafik tertinggi terjadi pada bulan Agustus 2023 dengan jumlah **78.464.385**, sedangkan anomali terendah terjadi pada bulan November 2023 dengan jumlah **19.296.439**. Dari aktivitas anomali trafik ini dapat berdampak pada penurunan performa perangkat dan jaringan, pencurian data sensitif, hingga perusakan reputasi dan penurunan kepercayaan terhadap suatu organisasi tertentu.

2.2.4 Laporan Bulanan Publik

Lalu berdasarkan Laporan Bulanan Agustus 2023 (Badan Siber dan Sandi Negara (BSSN), 2023), 2 jenis serangan siber terbesar yang menyumbang angka tertinggi dari sumber anomali yang ada adalah serangan SSH Account Brute Force Attack dengan jumlah **2.876.228**, lalu ada MSSQL database account Brute Force Attack dengan total data anomali sebesar **359.241.**

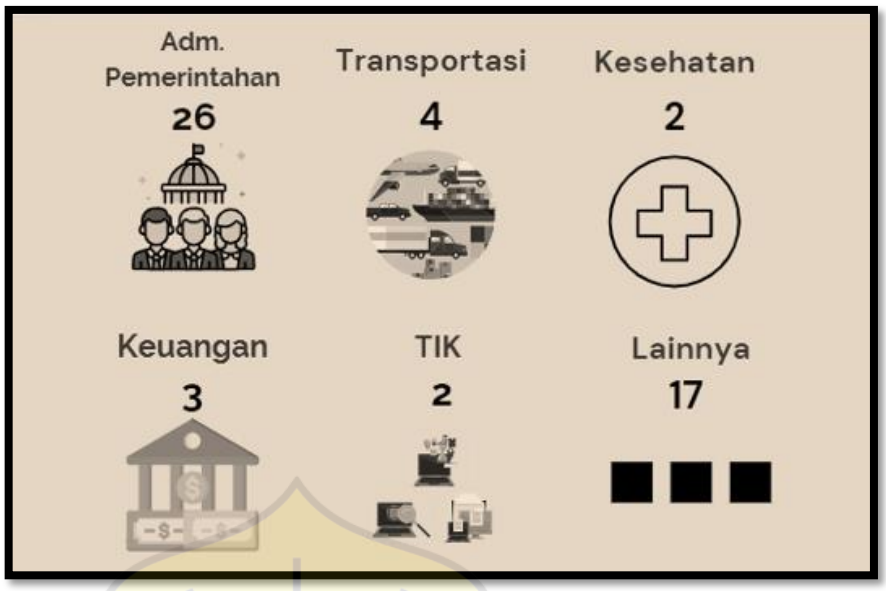


Gambar 2. 3 Laporan Bulanan Publik Bulan Agustus 2023



Gambar 2. 4 Top 5 Sumber IP Anomali

Lalu, dari beberapa pelaporan diatas, (Badan Siber dan Sandi Negara (BSSN), 2023) juga menyebuatkan sebaran kasus per-sektor yang paling banyak terjadi peretasan selama bulan agustus ini adalah sektor Administrasi Pemerintahan dengan jumlah kasus pelaporan sebanyak 26 kasus. Namun tidak menutup kemungkinan pada sektor lain untuk tetap perlu mendapatkan perhatian terkait keamanan perangkat teknologi informasi yang ada.



Gambar 2. 5 Sebaran Jenis Sektor Terindikasi Mengalami Dugaan Indikasi Insiden Keamanan Siber

Pada tanggal 12 Maret 2024, lewat platform Online-Message (WhatsApp), penulis mengajukan permohonan untuk melakukan penelitian terkait sebuah topik berdasarkan beberapa penjabaran diatas, pada salah satu sektor industry, di PT. Riwtech Workshop Indonesia. Sebuah industri yang bergerak di bidang jasa Control Panel dan alat-alat Lift Elevator dan Dumbwaiter. Dimana perusahaan ini baru saja ingin mengaplikasikan atau melakukan proses digitalisasi proses bisnisnya dimulai dari pembuatan Website Company Profile, hingga perencanaan pembuatan sistem internal ERP. Dalam kesempatan ini, penulis meminta izin untuk melakukan penelitian dalam proses migrasi dal hal penggunaan perangkat teknologi informasi sebagai media bisnis. Berdasarkan beberapa paparan diatas, penulis bermaksud untuk menawarkan simulasi penerapan Honeypot Cowrie dalam hal pencegahan

penerobosan masuk kedalam sistem yang baru dibangun, lalu dari sana akan di analisa data log keluaran dari Honeypot Cowrie yang berisikan beberapa informasi terkait event penyerangan tersebut, dengan algoritma naïve bayes. Diharapkan dengan melakukan simulasi ini, sektor-sektor industri pastinya masuk kedalam kategori 'lainnya' pada report yang disajikan oleh BSSN, bisa lebih 'aware' dalam hal penggunaan teknologi informasi, terutama dalam mengimplementasikan sistem informasi sebagai sarana media bisnis.

2.2.5 Secure Shell (SSH)

SSH yang merupakan singkatan dari Secure Shell adalah sebuah protokol keamanan berbasis pada lapisan aplikasi yang dikembangkan oleh IETF's Network Working Group. Hal ini diperkuat dengan pernyataan (Heni Jusuf, 2015), SSH merupakan protokol jaringan yang berada di lapisan aplikasi pada protokol TCP/IP, memfasilitasi sistem komunikasi yang aman antara dua sistem yang menggunakan arsitektur *client server* dengan menyediakan kerahasian dan integritas data melalui Teknik enkripsi dan dekripsi yang dilakukan secara otomatis di dalam setiap koneksinya, untuk menggunakan SSH dibutuhkan proses otentikasi user berupa kunci umum atau password yang terenkripsi. Berdasarkan data report sebelumnya, serangan siber SSH Brute Force Attack, semakin meningkat dalam kompleksitas skema penyerangannya. Hal ini menimbulkan ancaman serius terhadap integritas dan kerahasian data. Brute force attack merupakan salah satu jenis serangan siber yang paling umum dan berbahaya, dimana penyerang akan mencoba untuk mendapatkan hak akses ke sistem dengan mencoba berbagai kombinasi kata sandi dan username secara berulang hingga menemukan hak akses yang benar.

2.2.6 Honeypot

Menurut (Wastumirad & Darmawan, 2021), *Honeypot* adalah suatu sistem atau perangkat lunak yang dibuat dan dirancang mirip dengan server sungguhan sehingga dapat berpura-pura menjadi sasaran nyata serangan. Fungsi utama dari *Honeypot* ini bertujuan untuk menjadi pengalih perhatian atau bisa dibilang sebagai umpan bagi para penyerang dan didalamnya sudah disisipkan seperangkat *script* untuk menangkap setiap *event* yang terjadi pada *service* tersebut. Pada implementasinya Honeypot akan dijadikan sebagai umpan dengan menjalankan beberapa service yang dibuat seperti sungguhan, yang ditujukan untuk menarik perhatian, mendeteksi dan memeriksa serangan yang terjadi dan dilakukan oleh penyerang. Hal ini didukung oleh pernyataan (Ubaidillah et al., 2023) yang mengatakan bahwa Honeypot merupakan sistem tiruan yang dibuat untuk menirukan keaslian sebuah layanan yang berjalan pada server fisik maupun cloud, sehingga dapat mengelabui *attacker* yang mencoba menyerang. Honeypot merupakan salah satu alternatif pengamanan server yang gratis karena tool ini dapat diberdayakan tanpa dipungut biaya apapun, atau bisa disebut *Open Source*. Sebagai gambaran umum, menurut (Muh Masruri Mustofa et al., 2013), Honeypot tidak akan mencatat trafik yang legal. Sehingga dapat dilihat bahwa yang berinteraksi dengan Honeypot adalah user yang menggunakan sumber daya sistem yang digunakan secara illegal. Hal ini diperkuat dengan konsep Honeypot yang memang sengaja disediakan untuk kebutuhan tersebut, dimana akan dibuat sebuah environtment semirip mungkin dengan keadaan asli dari server terkait, namun dengan beberapa informasi yang berbeda untuk membedakan service asli dan service umpan.

Menurut (Wahyu Adi Sulaksono & Cosmas Eko Suharyanto, 2020), terdapat 3 jenis layanan *honeypot* yang dapat disesuaikan dengan potensi ancaman yang diterima *server*, dan pemakai diberikan fleksibilitas untuk memilih salah satu dari ketiga layanan ini untuk dipergunakan didalam sistem internal jaringannya. Ketiga jenis layanan honeypot tersebut adalah sebagai berikut.

2.2.7 Low-Interaction Honeypot

Low interaction honeypot merupakan layanan pertama dalam honeypot dimana Honeypot akan menciptakan server tiruan dan pengelola jaringan selaku pemilik server masih memiliki kendali penuh untuk mengawasi kegiatan penyusupan yang terjadi.

2.2.8 Medium-Interaction Honeypot

Medium interaction honeypot merupakan layanan kedua dalam jenis honeypot dimana sebuah sistem operasi palsu dibuat untuk menjebak attacker. Pada layanan ini, beberapa perintah honeypot akan dilewatkan oleh sistem, sebagai gantinya setiap informasi dari attacker akan direkam dan dapat dievaluasi oleh pihak pengelola jaringan. Salah satu yang menyediakan layanan honeypot ini adalah Cowrie.

2.2.9 High-Interaction Honeypot

High interaction honeypot merupakan jenis ketiga dari layanan honeypot dimana pengelola jaringan tidak perlu lagi mengawasi kegiatan penyusupan karena server asli telah direplikasi secara keseluruhan, sehingga attacker dipersilakan menyerang server replikasi yang diisikan informasi palsu, sehingga attacker merasa

puas telah mendapatkan informasi illegal, padahal server yang sebenarnya masih aman tanpa tersentuh sedikitpun.

2.2.10 Honeypot Cowrie (Medium-Interaction Honyepot)

Menurut (Toriyansa Natanegara et al., 2023), *Cowrie* adalah perangkat lunak yang berfungsi sebagai alat bantu untuk mengimplementasikan *honeypot* dan digunakan untuk menyamarkan layanan di server *openssh*. Perangkat ini termasuk dalam kategori honeypot *Medium-Interaction Honeypot* yang digunakan untuk mendeteksi dan mencatat serangan brute force attack pada service *SSH*, *Telnet*, dan *OpenSSH*. *Honeypot Cowrie* beroperasi dengan menggunakan konsep *redirection*, yang berarti setelah serangan *openssh* berhasil, *cowrie* mengarahkan penyerang ke layanan honeypot palsu. Hal ini akan membuat penyerang mengira bahwa serangannya berhasil, padahal yang sebenarnya terjadi adalah mereka hanya terperangkap di dalam *service honeypot* palsu. Hal ini didukung oleh pernyataan (Wahyu Adi Sulaksono & Cosmas Eko Suharyanto, 2020), bahwa Cowrie adalah sebuah software pendukung yang berguna untuk mempermudah inisialisasi pada honeypot dan dimanfaatkan untuk melakukan penyamaran layanan pada openssh server. Cowrie masuk kedalam tipe honeypot jenis interaksi sedang, yang dipakai untuk hal mendeteksi dan mencatat setiap serangan brute force attack yang menyerang service ssh, telent, dan openssh server. Konsep yang digunakan cowrie adalah pengalihan atau redirection, dimana ketika terjadi penyerangan kedalam service yang disebutkan diatas, cowrie akan mengarahkan penyerang untuk masuk kedalam layanan palsu honeypot. Sehingga attacker akan mengira bahwa dia sudah berada di dalam sistem utama server, padahal attacker hanya sampai kedalam

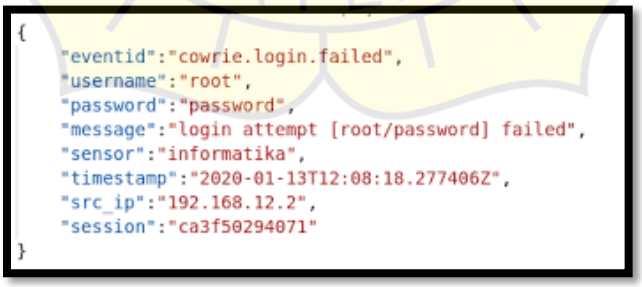
sistem palsu yang honeypot cowrie sediakan dengan sengaja, untuk dicatat segala aktivitas yang ada kedalam sebuah log file.

2.2.11 Data Log

Data log dari *Honeypot Cowrie* dikumpulkan secara otomatis menggunakan skrip pengambilan data kustom yang dijalankan pada server Honeypot. Skrip ini akan mengakses file log cowrie yang terus-menerus diperbarui oleh sistem dan menyimpannya dalam format teks ataupun *JSON* untuk di analisa lebih lanjut.

Data log yang dihasilkan oleh Honeypot Cowrie mencatat berbagai kejadian yang terjadi dalam lingkungan simulasi, termasuk upaya login yang gagal, dan beberapa event lainnya.

Data log yang dihasilkan oleh Honeypot Cowrie disimpan dalam format teks dan JSON Form dengan struktur yang sudah ditentukan berdasarkan jenis filenya. Setiap baris log akan mencatat beberapa informasi dari setiap event yang terjadi, dengan contoh informasi seperti timestamp, jenis aktivitas yang masuk, alamat IP penyerang, alamat IP korban, serta deskripsi singkat tentang aktivitas yang terjadi.



Gambar 2. 6 Cowrie JSON Log Sample

2.2.12 Algoritma Naïve Bayes

Menurut (Aditya et al., 2023), algoritma *Naïve Bayes* termasuk kedalam jenis metoda *Supervised Learning*, bagian Klasifikasi. Ini berarti dalam pemodelan program yang akan dibangun, kita perlu 'mengenalkan' pola-pola dari *'expected-result'* yang kita harapkan. Sedangkan, Klasifikasi disini merupakan suatu cara mendapatkan model atau sebuah fungsi yang mampu mendeskripsikan serta menyeleksi kelas data atau sebuah konsep, yang ditujukan agar model tersebut mampu diaplikasikan dalam memproyeksikan kelas yang tidak pasti dari sebuah entitas yang sedang dianalisa. Hal ini didukung dengar pernyataan (Farid Rifai et al., 2019), Algoritma Naïve Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output. Dengan kata lain, jika diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas individu. Dalam pemahaman sederhana, seperti yang disampaikan (Safii et al., 2022), Algoritma Naïve Bayes merupakan metode yang mengklasifikasi suatu data dengan cara efektif dengan mengoptimalkan pengawasan perkiraan dalam probabilitas akurat dengan asumsi penyederhanaan nilai atribut kondisional yang saling bebas jika diberikan nilai output.

$$\mathbf{P(H|X) = P(X|H)\ P(H) / P(X)} \qquad \mathbf{(1)}$$

Keterangan :

X : Data Sampel dengan label yang tidak diketahui
H : Hipotesis bahwa X adalah data dengan sebuah label, Y
P(H|X) : Peluang bahwa hipotesis benar untuk data sampel X yang dianalisa
P(X|H) : Peluang data sampel X, bila diasumsikan hipotesa benar
P(H) : Peluang dari Hipotesis H

P(X) : Peluang data sampel yang dianalisa

Adapun langkah atau tahapan dari Algoritma Naïve Bayes tersebut adalah :

1. Menghitung jumlah kelas / label “P(H)”
2. Menghitung jumlah kelas per kelas “P(X|H)”
3. Kalilkan semua variable kelas 1(yes) dan 0(no) “P(X|H) * P(H)”
4. Bandingkan hasil per kelas 1(yes) dan 0(no)

2.2.13 Kelebihan dan Kekurangan Naïve Bayes

Berikut adalah uraian terkait keunggulan dan kekurangan dari Algoritma Naïve Bayes.

2.2.13.1 Kelebihan Algoritma Naïve Bayes

Kelebihan algoritma Naïve Bayes antara lain :

1. Algoritma ini cukup memerlukan sejumlah data pelatihan (Data Train) dalam menaksir perkiraan skala yang dibutuhkan dalam proses pengelompokkan. Naïve bayes kerap beroperasi jauh lebih baik dalam mayoritas suasana dunia nyata yang kompleks daripada yang diperlukan (Widodo et al., 2021 dalam Aditya et al., 2023).
2. Cepat dalam perhitungan, algoritma simple dan presisi. Naïve bayes lebih akurat diaplikasikan dalam data yang besar dan dapat mengerjakan data yang tidak utuh (missing value) serta kuat atas atribut yang tidak bermakna dan noise pada sebuah data (Arifin & Ariesta, 2019 dalam Aditya et al., 2023).

3. Mudah diimplementasikan dan dalam banyak kasus memberikan hasil yang baik (Suprianto, 2020 dalam Aditya et al., 2023).

2.2.13.2 Kekurangan Algoritma Naïve Bayes

Kekurangan Algoritma Naives Bayes antara lain :

1. Probabilitasnya tidak dapat menilai seberapa presisi suatu pengelompokkan. Algoritma naïve bayes mempunyai kelemahan pada penetapan atribut yang terdapat di dalam data sehingga dapat mempengaruhi produk akhir berupa tingkat presisi (Arifin & Ariesta, 2019 dalam Aditya et al., 2023).
2. Amat rentan pada fitur yang berlebih, sehingga membuat tingkat presisi pengelompokkan menjadi rendah (Aditya et al., 2023).
3. Tidak valid jika probabilitas sementaranya adalah kosong, jika kosong, maka probabilitas prakiraan akan bernilai kosong juga. Serta memperhitungkan variable independent (Suprianto, 2020 dalam Aditya et al., 2023).

2.2.14 Tipe Algoritma Naïve Bayes

Ada beberapa tipe Algoritma Naïve Bayes yang bisa dijadikan acuan dalam membuat suatu model fungsi tertentu, diantaranya :

2.2.14.1 Naïve Bayes Classifier

Naïve Bayes Classifier merupakan suatu aturan klasifikasi yang menggunakan prinsip kalkulasi peluang (Amrullah et al., 2020 dalam Aditya et al., 2023) dan memperhitungkan seluruh atribut independent atau tidak saling keterikatan yang dibagikan oleh nilai pada variable kelas (Dita et al., 2021 dalam Aditya et al., 2023). Naïve bayes classifier memberikan manfaat, yakni cukup memerlukan sejumlah data pelatihan (Data Train) untuk menaksir perkiraan skala yang dibutuhkan dalam proses pengelompokkan (Affandi et al., 2020 dalam Aditya et al., 2023).

2.2.14.2 Multinomial Naïve Bayes

Multinomial Naïve Bayes memperhitungkan jumlah kemunculan kata dalam dokumen (Purwiantono & Aditya, 2020 dalam Aditya et al., 2023), sehingga mengasumsikan independensi kehadiran kata pada manuskrip dengan tidak memperkirakan urutan kata ataupun konteks informasi. (Haditira et al., 2022 dalam Aditya et al., 2023) , Manuskrip Multinomial Naïve Bayes dapat juga dikenal sebagai “bags of words” yang rentetan kejadian hadirnya kata dalam manuskrip diabaikan, sehingga setiap kata diproses menerapkan distribusi multinomial.

2.2.14.3 Gaussian Naïve Bayes Classifier

Metode Gaussian Naïve Bayes Classifier ialah satu dari algoritma berbasis nilai berkelanjutan dengan rancangan probabilitas yang mampu diaplikasikan pada penetapan kelas dari manuskrip dan dapat mengerjakan data dalam jumlah besar dengan presisi (Mujahidin et al., dalam Aditya et al., 2023).

Gaussian Naïve Bayes ialah satu dari Naïve Bayes, dimana metode ini beroperasi dengan probabilitas. Pada metode ini, data yang diaplikasikan pada fitur yang dipakai yakni bertipe angka (Cahyaningrum et al., dalam Aditya et al., 2023).

2.2.15 Evaluasi Kinerja Model

2.2.15.1 Confusion Matrix

Ada beberapa metoda yang umum digunakan untuk mengevaluasi sebuah model AI, hal ini didasari dari jenis masalah yang coba diselesaikan, dan juga algoritma yang digunakan. Penulis memutuskan untuk menggunakan metoda Confusion Matrix dalam mengevaluasi model AI yang akan dibangun dengan algoritma Naïve Bayes. Hal ini didasari oleh pernyataan (Aditya et al., 2023), Confusion Matrix adalah tabel yang digunakan untuk menggambarkan kinerja model klasifikasi. Hal ini diperkuat dengan penelitian yang dilakukan (Nurhidayat & Dewi, 2023), Confusion Matrix adalah sebuah tabel yang digunakan untuk menampilkan jumlah data uji yang diklasifikasikan dengan benar dan salah, sehingga memudahkan proses penelitian dalam mengevaluasi akurasi suatu sistem klasifikasi.

Menurut (Luthfy Romadloni Pristian et al., 2022), pada konsep dasarnya, confusion matrix merupakan suatu pengukuran yang digunakan untuk memberikan informasi perbandingan dari hasil klasifikasi yang dilakukan oleh suatu algoritma yang akan digunakan dengan hasil klasifikasi sebenarnya. Sebagai contoh, akan disajikan dalam tabel dibawah ini,

Tabel 2. 2 Confusion Matrix Table

Confusion Matrix		**P R E D I C T**	
		Predicted Positive	Predicted Negative
ACTUAL	Actual Positive	True Positive [TP]	False Negative [FN]
	Actual Negative	False Positive [FP]	True Negative [TN]

Keterangan :

1. ***True Positive*** : ini berarti seberapa banyak data actual yang memiliki kelas Positive, dan model juga memprediksi sebagai kelas Positive.
2. ***True Negative*** : ini berarti seberapa banyak data actual yang memiliki kelas Negative, dan model juga memprediksi sebagai kelas Negative
3. ***False Positive*** : ini berarti seberapa banyak data actual yang memiliki kelas Negative, namun model memprediksi sebagai kelas Positive
4. ***False Negative*** : ini berarti seberapa banyak data actual yang memiliki kelas Positive, namun model memprediksi sebagai kelas Negative

Melalui 4 variable tersebut, dapat diperoleh data-data lainnya yang sangat berguna untuk mengukur dan mengevaluasi kinerja Model AI yang akan dibangun, diantaranya :

2.2.15.2 Accuracy

Accuracy merupakan total keseluruhan data, terkait seberapa sering model mengklasifikasikan dengan Benar. Formula accuracy biasa ditulis seperti gambar dibawah ini.

Accuracy = TP + TN / Total **(2)**

2.2.15.3 Precision

Precision merupakan ketika suatu model memprediksi positive, dan seberapa sering prediksi itu benar. Formula precision biasa ditulis seperti gambar dibawah ini.

Precision = TP / FP + TP **(3)**

2.2.15.4 Recall

Recall (Sensitivity / True Positive Rate) merupakan suatu data ketika kelas aktualnya positive, dan seberapa sering model memprediksi sebagai kelas positive. Formula recall dapat ditulis menggunakan persamaan seperti gambar dibawah ini.

Recall = TP / FN + TP **(4)**

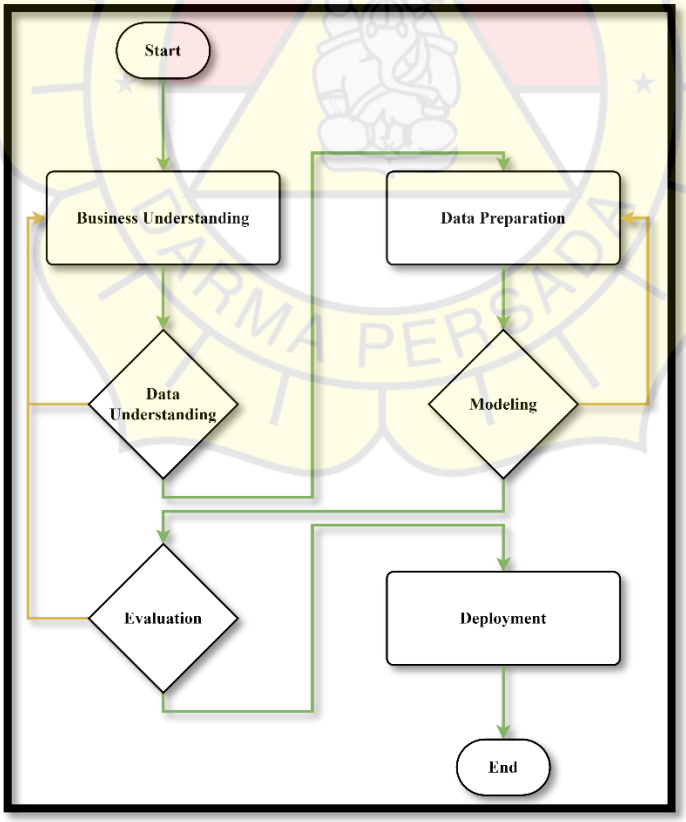
2.2.15.5 F-1 Score

F1-Score merupakan rata-rata harmonik dari Precision dan Recall. Formula F-1 Score dapat ditulis menggunakan persamaan seperti gambar dibawah ini.

F-1 Score = 2* (Precision * Recall) / Precision + Recall **(5)**

2.2.16 CRISP-DM (Cross-Industry Standard Process for Data Mining)

Menurut (Feblian & Daihani, 2017), Cross-Industry Standard Process for Data Mining (CRISP-DM) adalah suatu standarisasi dalam proses data mining sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian. Hal ini didukung oleh pernyataan (Hasanah et al., 2021), CRISP-DM merupakan suatu standarisasi pemrosesan data mining yang telah dikembangkan dimana data yang ada akan melewati setiap fase terstruktur dan terdefinisi dengan jelas dan efisien. Metodologi ini terdiri dari 6 fase tahapan yang perlu dilakukan secara sekuensial , yaitu : Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, dan Deployment. Atau bisa digambarkan secara visual flowchart seperti gambar dibawah ini :



Gambar 2. 7 CRISP-DM

Referensi menurut (Pambudi & Abidin, 2023).

Detail dari keenam tahapan diatas akan dijabarkan sebagai berikut :

1. **Business Understanding**

   Pada tahapan ini akan berfokus untuk memahami tujuan bisnis dari proyek yang akan dikerjakan. Ini akan melibatkan beberapa *stakeholder* atau *enduser* dalam mengidentifikasikan sebauh permasalahan dan peluang yang bisa dimanfaatkan.

2. **Data Understanding**

   Pada tahap ini, akan dilakukan pengumpulan data, mengevaluasi jumlah dan nilai, serta memahami karakteristik dari struktur data yang akan di analisa.

3. **Data Preparation**

   Pada tahapan ini, akan disiapkan kumpulan data yang sudah didapatkan dari proses sebelumnya, untuk dianalisa lebih lanjut. Pada proses ini akan mencakup beberapa proses seperti pembersihan data, penggabungan data, pemilihan atribut yang relevan, serta beberapa proses lainnya yang relevan dengan proses terkait.

4. **Modeling**

   Di tahap ini, akan dibangun sebuah model AI menggunakan algoritma *Naïve Bayes* . Proses ini akan melibatkan data yang sudah diproses pada tahapan CRISP-DM sebelumnya.

5. **Evaluation**

   Tahapan evaluasi akan digunakan untuk mengevaluasi model yang sudah dibangun pada proses sebelumnya. Akan dilakukan beberapa skema pengujian model terkait akurasi , presisi, recall, serta F1-Score atau bisa disebut *Confussion Matrix*.

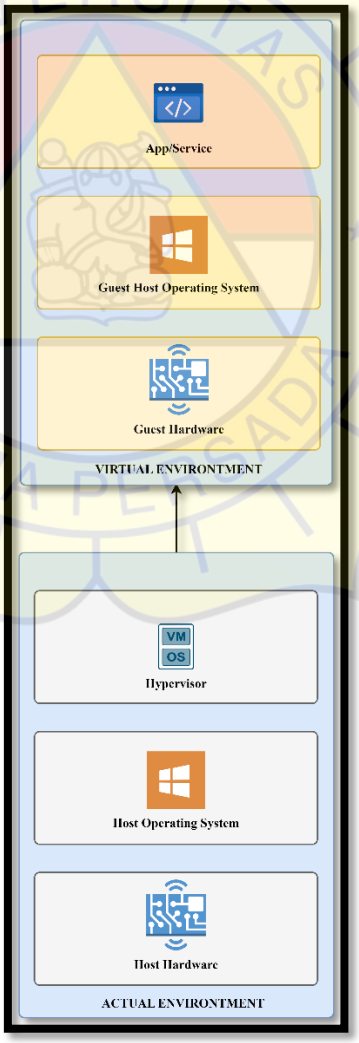
6. **Deployment**

   Setelah model AI yang dibangun sudah memenuhi *goal* dari proyek yang sedang dikerjakan, akan dilakukan tahapan deployment model AI pada platform yang relevan seperti *Cloud Native*, atau media lainnya yang relevan dengan topik akhir.

2.2.17 Virtual Environtment – Virtualisasi Server

Menurut (Afriandi, 2012 dalam Prasandy, 2015), Virtualisasi Server adalah penggunaan suatu software tertentu yang memungkinkan seperangkat hardware menjalankan beberapa sistem operasi dan service secara bersamaan. Dalam terminology yang lain, menurut (Prasetyo et al., 2022), Teknik Virtualisasi adalah suatu cara untuk membuat sesuatu dalam bentuk virtual, tidak nyata. Virtualisasi digunakan juga untuk mengemulasikan perangkat fisik komputer , dengan cara membuatnya seolah-olah perangkat tersebut tidak ada (disembunyikan) atau bahkan menciptakan perangkat yang tidak ada menjadi ada. Lebih jelasnya, menurut (Khasanah et al., n.d.), Pengertian virtualisasi dalam konteks komputasi mengacu pada abstraksi dari komponen fisik menjadi objek logis. Dengan

virtualisasi, dapat diperoleh utilitas yang lebih besar dari komponen fisik menjadi objek logis. Teknologi virtualiasi mengemulasikan sumber daya komputasi fisik, seperti komputer desktop atau server, processor, dan memori, sistem penyimpanan dan jaringan. Virtualisasi server bisa menciptakan sebuah Virtual Environtment yang memungkinkan beberapa beban aplikasi atau server bisa berjalan di satu komputer, namun seolah-olah bisa berjalan di komputer dengan resource yang berbeda. Dibawah ini disajikan gambaran umum terkait penggambaran Virtual Server dan Virtual Environtment dalam konteks yang sudah dijabarkan sebelumnya.



Gambar 2. 8 Virtual Environtment

2.2.18 Penetration Testing Technic

Menurut (Ghanem & Chen, 2020 dalam Fachri et al., 2021), Penetration Testing (Pengujian Penetrasi) adalah suatu praktik umum untuk secara aktif menilai pertahanan jaringan komputer atau Web Server dengan merencanakan dan mengeksekusi semua kemungkinan serangan untuk menemukan dan mengeksploitasi kerentanan yang ada. Hal ini diperkuat dengan penjelasan (Mulyadi dalam Hidayatulloh & Saptadiaji, 2021), menjelaskan bahwa Penetration Testing merupakan serangkaian proses berisi prosedur dan Teknik mengevaluasi keamanan terhadap sistem komputer atau jaringan dengan melakukan simulasi penyerangan untuk mengetahui letak celah-celah kerawanan pada sistem agar kemudian celah tersebut ditutup atau diperbaiki. Penetration Testing dilakukan sebagai langkah Preventive untuk mengatasi terjadinya peretasan pada suatu sistem.

Menurut (Kurniawan & Nugroho, 2019), ada beberapa tahapan yang harus dilakukan dalam melakukan Penetration Testing, diantaranya :

1. Footprinting = umumnya digunakan untuk mengumpulkan informasi sebanyak-banyaknya mengenai target yang akan diserang, baik itu dari sisi Hardware, maupun Software. Hal-hal umum seperti IP Address, Name Server, Server Acrhitecture merupakan target dari tahapan ini.
2. Scanning = merupakan tahapan dimana attacker akan mengumpulkan portlist yang terbuka dari sutau server yang menjadi target penyerangan siber. Tahapan ini seperti mencari sebuah ‘hole’ dari sebuah server yang menjadi target.

3. Enumeration = merupakan sebuah tahapan dimana attacker akan mencari user account yang active atau valid dari target.
4. Gaining Access = merupakan tahapan dimana attacker akan mencoba masuk ke dalam server atau mengakses server tersebut
5. Privilege Escalation = tahapan dimana attacker akan mencari cara untuk mendapatkan hak full-access pada server tersebut
6. Covering Tracks = setelah mendapatkan full-access, attacker akan menghapus ‘data logs’ yang ada agar tidak dicurigai
7. Backdooring = terakhir, user akna mencoba membuat user baru dengan privilege tertentu dengan tujuan ketika attacker kembali akan masuk ke dalam server, maka bisa menggunakan user tersebut.
8. Denial of Service = merupakan langkah akhir apabila cara-cara diatas tidak tercapai, yaitu dengan ‘merusak’ service yang berjalan di server tersebut.

2.2.19 Penetration Tools

Penetration Tools (Alat Penetrasi) pada dasarnya merupakan sebuah perangkat lunak atau software, yang dibuat dan digunakan oleh professional keamanan komputer dan peneliti keamanan suatu sistem komputer atau jaringan. Alat-alat ini dirancang untuk mengidentifikasi dan mengeksploitasi kerentanan dalam Infrastruktur Jaringan IT dengan tujuan menguji ketahanan suatu sistem terhadap suatu jenis serangan siber tertentu.

2.2.19.1 Hydra

Hydra merupakan salah satu tools yang disediakan Kali Linux, untuk melakukan action password-cracking sesuai dengan wordlist user dan password yang ada.

2.2.19.2 Medusa

Medusa adalah software serupa dengan Hydra yang dirancang untuk menangani serangan brute force attack pada berbagai protokol autentikasi , termasuk FTP, SSH, dan Telnet.

2.2.19.3 Ncrack

Merupakan alat serangan otomatis yang digunakan untuk menguji keamanan jaringan dengan schema penyerangan Dictionary Attack dan Brute Force Attack.

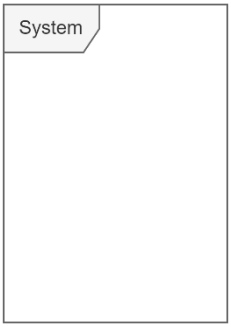
2.2.20 Pemodelan UML

Menurut (Sundaramoorthy Suriya, 2022),UML (Unified Modeling Language) adalah sebuah bahasa visual yang direpresentasikan dalam bentuk diagram-diagram tertentu, yang digunakna untuk menggambarkan , memvisualisasikan, membangun dan juga mendokumentasikan tiap komponen-komponen dari suatu perangkat lunak atau software. UML bertujuan untuk membantu penelitian dalam memahami suatu permasalahan dan cara penyelesaian dalam bentuk visual diagram abstrak.

2.2.20.1 UML Use Case Diagram

Use Case Diagram akan berfokus pada proses identifikasi dari suatu fungsional kebutuhan dari sebuah sistem yang sedang dibangun.

Tabel 2. 3 UML Use Case Diagram

No	Nama Komponen	Notasi UML	Tujuan
1	System Boundary		Digunakan sebagai representasi dari ‘Scope’ dari sebuah Software | Sistem. Notasi ini membungkus hampir secara keseluruhan fungsi yang ada didalamnya.
2	Actors		Menggambarkan sebuah ‘user’ yang bisa berinteraksi sistem. ‘Actor’ disini dapat berupa stakeholder, organisasi, atau sesama aplikasi

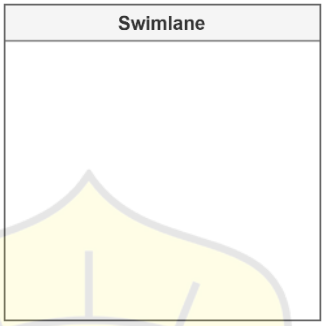
			dengan scope yang berbeda. 'Actor' biasanya berinteraksi dengan sebuah notasi 'usecase'.
3	Usecases		Menggambarkan fungsi bisnis yang dijadikan topik permasalahan.

2.2.20.2 UML Activity Diagram

Activity Diagram berfokus pada sekuensial dan aktivitas parallel yang digambarkan dalam setiap fungsi yang dibutuhkan sistem.

Tabel 2. 4 UML Activity Diagram

No	Nama Komponen	Notasi UML	Tujuan
1	Initial State		Digunakan sebagai notasi 'Start' dari suatu System.

2	Final State		Menggambarkan sebuah notas ‘Akhir’ dari suatu process system.
3	Swimlane		Terbagi menjadi 2 jenis, yaitu Horizontal dan Vertical. Digunakan sebagai representais action yang dilakukan dari Actor terhadap sebuah process usecase atau aktivitas yang terlibat dalam sistem.
4	Action State		Merepresentasikan sebuah proses atau aktivitas dari sistem.

5	Decision		Menggambarkan suatu proses kondisi dengan logic tertentu.
6	Flow Final		Menggambarkan suatu pemberhentian proses dalam suatu logic tertentu. Semacam exception handling di dalam sistem.