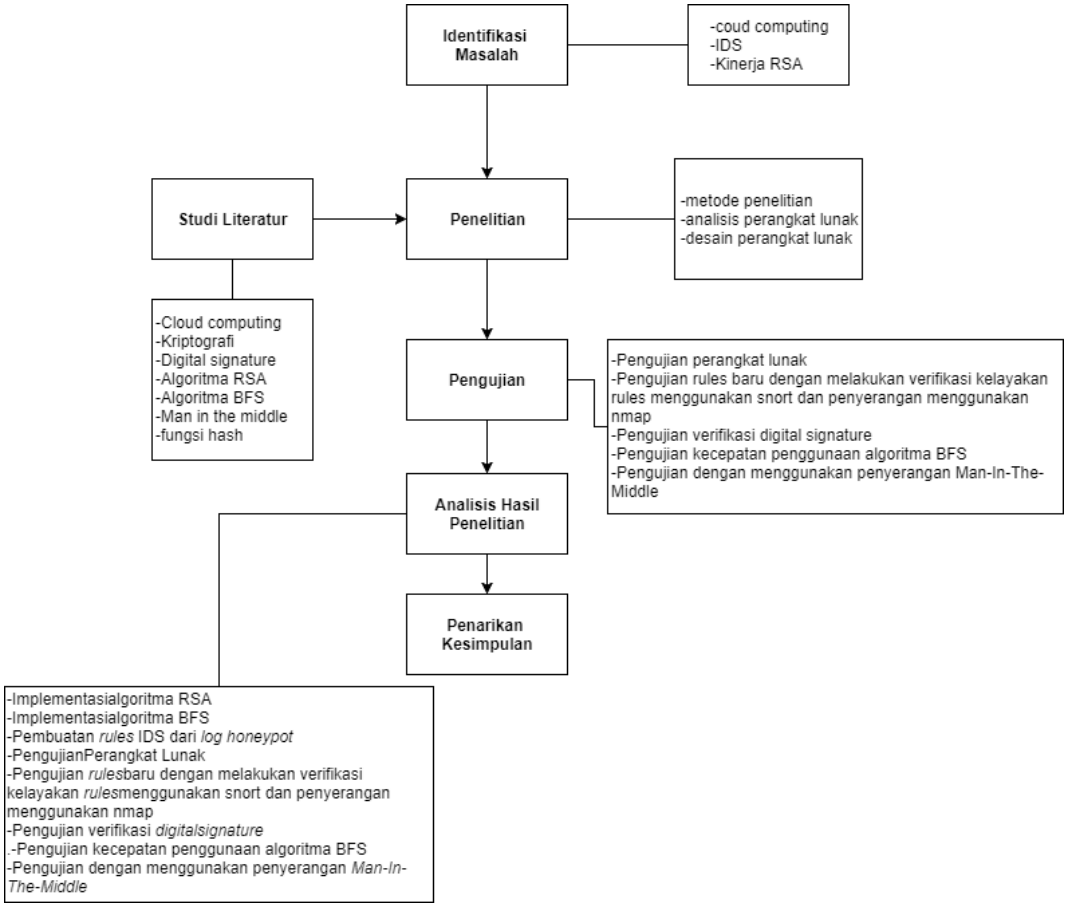
BAB III
METODO*LOG*I PENELITIAN

Pada bab ini akan dijelaskan mengenai metodologi penelitian, mulai dari desain penelitian, alat penelitian, dan metode penelitian.

3.1 Desain Penelitian

Desain penelitian adalah kerangka kerja yang digunakan untuk melakukan penelitian. Pada bagian ini penulis akan memaparkan kerangka kerja dari mulai penelitian sampai selesai. Desain penelitian digambarkan pada gambar 3.1.



Gambar 3.1 Desain penelitian

3.1.1 Identifikasi masalah

Identifikasi masalah merupakan tahapan awal penelitian yang sejalan dengan studi literatur. Pada tahap ini dilakukan identifikasi permasalahan dari penelitian-penelitian yang sudah dilakukan sebelumnya terhadap IDS, *Cloud computing* dan penggunaan *digital signature* RSA, sehingga didapatkan rumusan masalah sebagai landasan untuk dilakukannya penelitian ini.

3.1.1.1 Cloud computing

Cloud computing memiliki banyak permasalahan. Salah satunya adalah masalah keamanan. Seperti yang telah dibahas sebelumnya hapir 80% layanan OSN berbasis *cloud* rentan terhadap XSS. Bukan hanya XSS saja serangan berjenis DDOS juga dapat menjadi ancaman bagi *cloud computing*. Untuk itu dibutuhkan sebuah solusi untuk mencegah hal-hal tersebut. Salah satu solusinya adalah dengan memasang IDS pada lingkungan *cloud computing*.

Sr. no.	Victim	Year	Category of XSS worm
1.	UK Parliament Web Site	2014	Persistent XSS
2.	Video Sharing Website	2014	Non-Persistent XSS
3.	Yahoo! Mail	2013	Persistent XSS
4.	Paypal Website	2012	Persistent XSS
5.	Facebook	2011	Non-Persistent XSS
6.	Hotmail Website	2011	Persistent XSS
7.	eBay Website	2011	Persistent XSS
8.	Twitter Website	2010	Mikeyy [21]
9.	Orkut Website	2009	XSS bug [21]
10.	MySpace	2006	Samy [20]

Gambar 3.2 Insiden serangan XSS terhadap OSN (gupta, 2016).

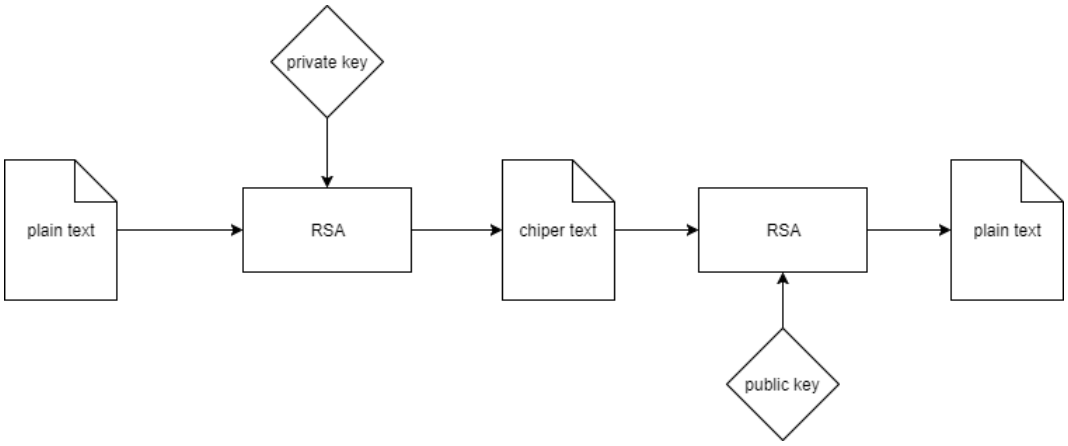
3.1.1.2 Intrusion Detection System

Meskipun IDS dapat menjadi solusi terhadap sebuah masalah, bukan berarti IDS sendiri tidak memiliki masalah. IDS memiliki beberapa kelemahan.

Salah satunya adalah tidak mampunya mendeteksi serangan baru. Karena itu dibutuhkan sebuah mekanisme dimana setiap serangan baru terdeteksi, maka *rules* baru tercipta. Caranya adalah dengan memanfaatkan *honeypot* untuk mendeteksi serangan, kemudian memanfaatkan *log honeypot* yang terkumpul untuk dijadikan *rules* IDS.

3.1.1.3 Kinerja RSA

Pada penelitian ini *rules* yang terbuat dari *log honeypot* kemudian didistribusikan kepada beberapa VM didalam *cloud computing*. Untuk memastikan keaslian dari *rules* yang diterima, algoritma RSA digunakan sebagai *digital signature* untuk melakukan otentifikasi keaslian *rules* tersebut. Namun penggunaan algoritma RSA dapat menurunkan performa VM sehingga menambah waktu kerja. Penelitian ini menggunakan algoritma BFS untuk mempersingkat waktu pembagian, dengan harapan waktu yang dihabiskan oleh RSA dapat tertutupi.



Gambar 3.3 RSA

3.1.2 Studi literatur

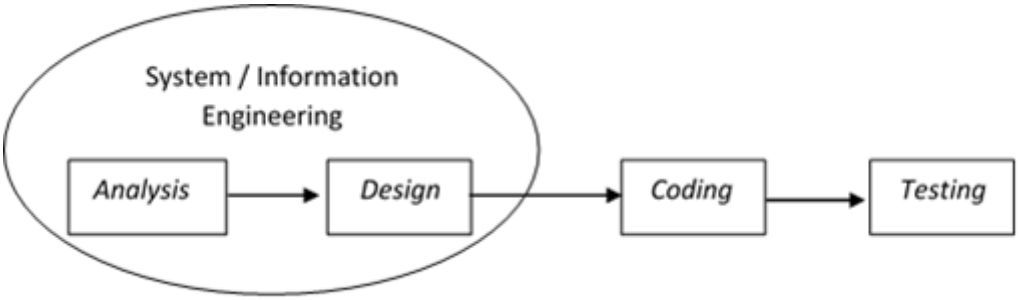
Studi literatur merupakan tahapan studi pendahuluan terhadap penelitian-penelitian sebelumnya yang terkait dengan penelitian yang akan dilakukan. Selain itu tahap ini dilakukan untuk memprlajari hal-hal teoritis mengenai hal-hal yang terkait dengan penelitian yang dilakukan, seperti IDS, kriptografi, *digital signature*,

honeypot, algoritma RSA, algoritma BFS, dan penelitian-penelitian lain yang relevan.

3.1.3 Penelitian

3.1.3.1 Metode Penelitian

Dalam penelitian ini, dilakukan pengembangan perangkat lunak menggunakan model *Linear Sequential*. *Linear Sequential* mengusulkan sebuah pendekatan kepada pengembangan perangkat lunak yang sistematik dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh analis, desain, kode, pengujian, dan pemeliharan. Berikut adalah proses gambaran dari *Linear Sequential Model* gambar 3.2.



Gambar 3.3.4 Model Linear *Sequential Model* (Pressman, 2001)

1. *System / Informartion engineering* Merupakan bagian dari sebuah sistem terbesar yang mana dalam pengerjaannya dimulai dengan menetapkan berbagai kebutuhan dari semua elemen yang diperlukan sistem dan mengalokasikannya ke dalam pembentukan perangkat lunak.
2. Analisis perangkat lunak merupakan tahap menganalisis hal-hal yang diperlukan dalam pembentukan sebuah perangkat lunak.
3. Desain merupakan beberapa langkah proses yang berfokus pada empat buah atribut yang berbeda dari program, yakni struktur data, arsitektur perangkat lunak, representasi antarmuka, dan sebuah algoritma.
4. *Coding* dilakukan untuk menerjemahkan pembuatan disain ke dalam bentuk yang bisa dimengerti oleh mesin. Sehingga komputer bisa

merepresentasikan ke dalam bentuk perangkat lunak. Pada penelitian ini *coding* dijelaskan lebih lanjut pada dokumen teknis

5. *Testing* merupakan langkah paling akhir yang dikerjakan, sebuah pengetesan pada perangkat lunak yang sudah melalui beberapa tahap dan dapat dipakai oleh user, dalam tes juga dapat di lakukan pengecekan apakah perangkat lunak yang dibuat sudah sesuai. Pada penelitian ini hasil dari *testing* dijelaskan lebih lanjut pada dokumen teknis

3.1.3.2 Analisis perangkat lunak

1. Analisis kebutuhan

   Perangkat lunak yang dibuat merupakan sebuah perangkat lunak yang memiliki fungsi untuk membuat dan mendistribusikan *rules* kepada VM lain. Pengembangan perangkat lunak ini membutuhkan beberapa hal seperti, perangkat keras, perangkat lunak dan juga *library*. Perangkat keras yang dibutuhkan pada penelitian ini berupa laptop dan juga beberapa *virtual server* dari *Amazon Web Service* (AWS) *Elastic Cloud Computing*(EC2). Perangkat lunak yang dibutuhkan adalah, linux ubuntu sebagai sistem operasi dimana perangkat lunak berjalan, dan PuTTY untuk mengakses AWS EC2. Sedangkan *library* yang digunakan adalah *library* RSA untuk membuat *digital signature* dan *library* SFTP untuk mentransfer *file*.

2. Analisis *input*

   Input yang dibutuhkan dalam perangkat lunak ini berupa data *log honeypot honeyd* yang telah dikumpulkan. *Log* didapatkan dengan cara menjalankan *honeyd* dan kemudian diserang menggunakan nmap. Setelah melakukan penyerangan, *honeyd* akan mengeluarkan data berupa *log*. Data *log* tersebut berupa kumpulan *string* yang nantinya akan diolah sehingga menjadi *rules* IDS. *Log* tersebut terdiri dari <*date&time*> <*protocol*> <*start/end*> <*source address*> <*source port*> <*destination address*> <*destination port*>. Data tersebut kemudian diseleksi, sehingga yang hanya diambil adalah *protocol*,

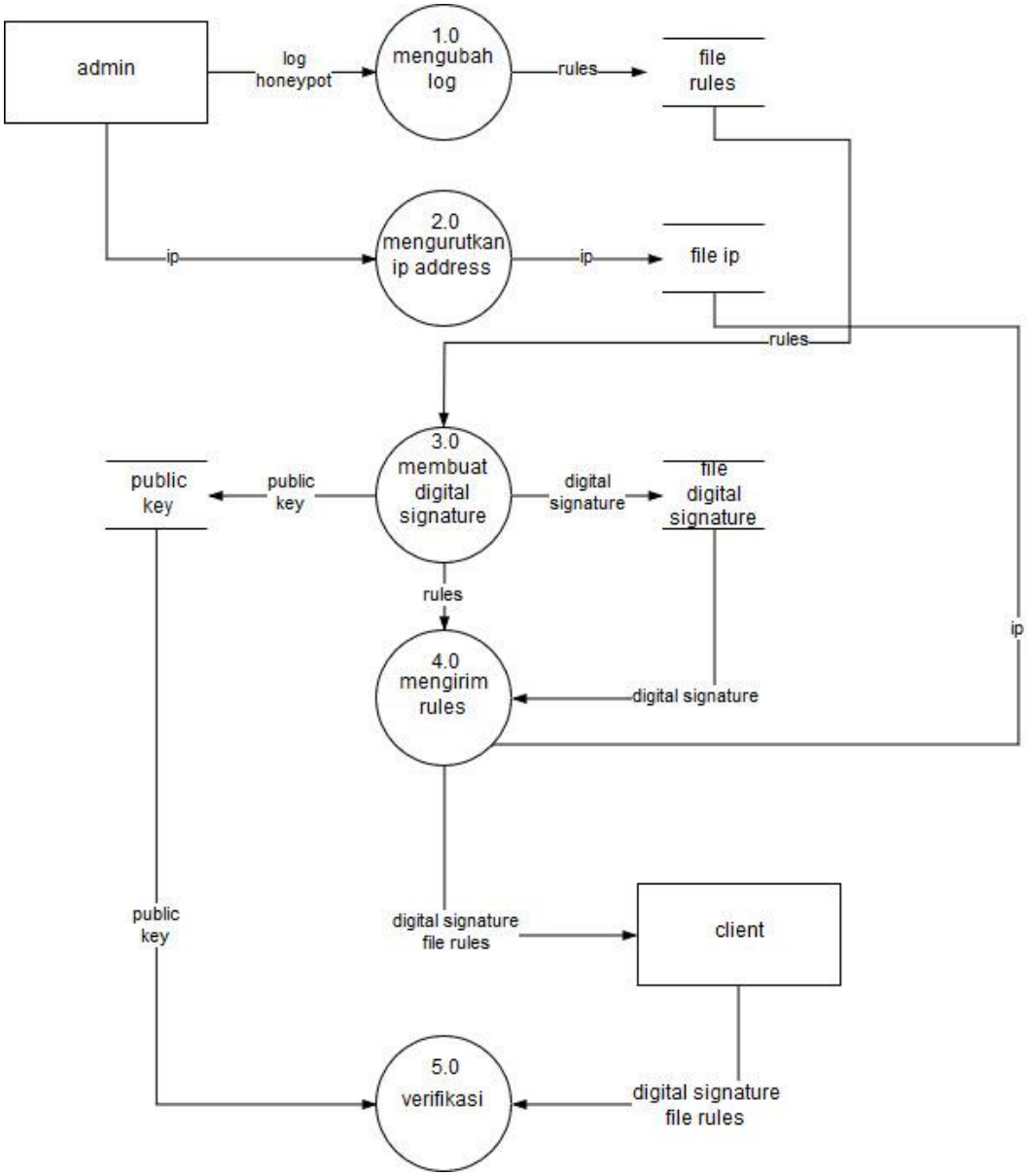
source address, source port dan *destination port*. Selain data *log*, masukan yang digunakan adalah ip address VM yang akan dibagikan file *rules*. Ip address, dimasukkan kedalam file txt, yang kemudian nantinya akan diurutkan dengan bantuan algoritma BFS.

3. Analisis *output*

   Output dari perangkat lunak ini adalah berupa file *rules* yang dibuat dari *log* yang sebelumnya telah dikumpulkan. Selain file *rules* perangkat lunak juga membuat keluaran berupa *digital signature* dan *public key* yang dibuat menggunakan algoritma RSA. Perangkat lunak ini juga membuat keluaran berupa IP yang telah diurutkan oleh algoritma BFS.

3.1.3.3 Desain perangkat lunak

Pada penelitian ini, perangkat lunak memiliki lima proses utama yaitu, mengubah *log*, mengurutkan ip address, membuat *digital signature*, mengirim *rules* dan verifikasi seperti gambar 3.3.



Gambar 3.5 Desain perangkat lunak

Admin adalah pihak yang membuat dan membagikan *rules* sedangkan *client* adalah pihak yang menerima *rules*. Berikut penjelasan dari masing-masing proses.

1. Mengubah log

   Input dari proses ini adalah *log* sedangkan output nya berupa rules. Rules dibuat dari *log honeypot* dengan cara mengubah bentuk dari *log* menjadi *rules* dengan menggunakan template yang sudah ditentukan.
2. Mengurutkan IP *address*

Pada proses ini input merupakan IP address yang belum terurut, dan output merupakan IP yang sudah diurutkan. Algoritma BFS, dimanfaatkan untuk mengurutkan IP dengan menghitung jumlah VM atau anak yang terhubung kepada suatu VM. Jika VM tersebut, setelah dihitung memiliki jumlah anak lebih banyak, maka VM tersebut akan didahulukan saat proses pengiriman *rules*.

3. Membuat *digital signature*

   Proses ini memiliki satu buah masukan berupa file *rules* dan dua buah keluaran yaitu *digital signature* dan *public key*. Proses pembuatan *digital signature*, diawali dengan membuat dua buah kunci atau *key*. *Key* tersebut adalah *private key* dan *public key*. Setelah itu *rules* dilakukan proses *hashing* menggunakan SHA-256 sehingga menghasilkan *digest*. *Digest* tersebut kemudian dienkripsi menggunakan *private key* hasil dari enkripsi tersebut adalah *digital signature*. Pada penelitian ini algoritma enkripsi yang digunakan adalah RSA.

4. Mengirim rules

   Pada proses ini *rules* dikirimkan dari admin kepada client dan *digital signature*. Pengiriman dilakukan berdasarkan IP yang sudah diurutkan.

5. Verifikasi

   Proses ini memiliki *input* berupa *public key*, *digital signature* dan file *rules.* Verifikasi dilakukan dengan cara melakukan hashing terhadap file *rules* yang diterima sehingga menghasilkan *digest*, kemudian digital signature didekripsi menggunakan public key pengirim file. Hasil dari dekripsi kemudian di bandingkan dengan *Digest* file yang diterima.

3.1.4 Pengujian

Pada tahap ini pungujian dilakukan dengan cara, menguji perangkat lunak dengan beberapa skenario. Baik skenario penggunaan perangkat lunak maupun skenario keamanan. Pengujian-pengujian yang dilakukan pada perangkat lunak adalah: Pengujian *rules* baru dengan melakukan verifikasi kelayakan *rules* menggunakan snort dan penyerangan menggunakan nmap, pengujian verifikasi

digital signature, pengujian kecepatan BFS. Sedangkan untuk skenario keamanan pengujian dilakukan dengan menggunakan serangan *Man In The Middle* (MITM).

Pengujian verifikasi kelayakan *rules* baru menggunakan snort dilakukan agar *rules* yang dibuat dapat digunakan oleh snort. Pengujian dilakukan dengan cara menjalankan *command* pada snort untuk melakukan verifikasi *file rules* yang telah dibuat. Sedangkan pengujian menggunakan penyerangan nmap dilakukan untuk menguji apakah *rules* yang dibuat dapat mendeteksi serangan. Pengujian dilakukan dengan melancarkan serangan menggunakan nmap kepada VM yang telah mengaplikasikan *rules* pada snort.

Pengujian verifikasi *digital signature* ditujukan untuk mengetahui apakah *digital signature* berfungsi dengan baik. Faktor yang menunjukkan apakah *digital signature* bekerja dengan semestinya adalah penerima dapat memverifikasi keaslian *digital signature*. Faktor lainnya adalah jika *digital signature* tidak dapat diverifikasi dikarenakan ada modifikasi atau perubahan pada tiga komponen yaitu *digital signature*, *palaintext*, atau *public key*. Pengujian ini dilakukan dengan cara melakukan verifikasi *digital signature* dengan tanpa memodifikasi ketiga komponen. Kemudian pengujian dilanjutkan dengan melakukan verifikasi menggunakan ketiga komponen yang telah dimodifikasi.

Pengujian kecepatan BFS dilakukan untuk mengetahui apakah algoritma BFS dapat membantu mempercepat pendistribusian *file rules*. Pengujian dilakukan dengan cara membandingkan kecepatan distribusi dengan menggunakan algoritma BFS dan tanpa algoritma BFS.

Skenario MITM digunakan untuk menguji apakah pihak yang menerima *file rules* dapat mengetahui jika *file rules* atau *digital signature* telah dimodifikasi pihak ketiga atau MITM. Pengujian dilakukan dengan cara memasukkan MITM antara pengirim dan penerima. MITM kemudian akan memodifikasi atau mengganti *file rules* atau *digital signature* yang dikirim oleh pengirim asli dan kemudian diteruskan kepada penerima.

3.1.5 Analisis hasil penelitian

Pada tahap ini hasil pengujian dibahas lebih lanjut, berikut hasil penelitian yang akan dibahas:

1. Implementasialgoritma RSA
2. Implementasialgoritma BFS
3. Pembuatan rules IDS dari log honeypot
4. PengujianPerangkat Lunak
5. Pengujian rulesbaru dengan melakukan verifikasi kelayakan rulesmenggunakan snort dan penyerangan menggunakan nmap
6. Pengujian verifikasi digitalsignature
7. Pengujian kecepatan penggunaan algoritma BFS
8. Pengujian dengan menggunakan penyerangan Man-In-The-Middle

3.1.6 Penarikan kesimpulan

Penarikan kesimpulan merupakan tahapan untuk menentukan kesimpulan kesimpulan dari hasil penelitian.

3.2 Alat Penelitian

Penelitian ini menggunakan seperangkat komputer yang dilengkapi perangkat lunak pendukung, dengan spesifikasi perangkat keras sebagai berikut:

1. Prosesor intel i7-6700HQ
2. Kartu Grafis NVIDIA Geforce GTX 950M
3. Random Access Memory(RAM) 8 GB
4. Solid State Drive (SSD) 256 GB
5. Hard Disk Drive (HDD) 1 TB
6. Monitor 14 inci dengan resolusi 1920x1080
7. 10 unit AWS EC2 t2.micro 1 vCPU 1 GB RAM

Adapun perangkat lunak yang digunakan adalah:

1. Sistem Operasi Ubuntu Server
2. Honeyd
3. Snort

4. PuTTY
5. WinSCP
6. Notepad++