PROPOSAL TUGAS AKHIR 2019

PENGAMANAN PERANGKAT IOT DENGAN ENCRYPTED PORT KNOCKING

MARIA OKTA SAFIRA

NIM. 1608561055

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

BUKIT JIMBARAN

2019

LEMBAR PENGESAHAN PROPOSAL

Judul	: Pengamanan Perangkat Iot Dengan Encrypted Port Knocking
Kompetensi	: Jaringan Sensor Nirkabel
Nama	: Maria Okta Safira
NIM	: 1608561055
Tanggal Disetujui	: 25 Oktober 2019

Disetujui Oleh

Ketua Penguji

I Dewa Made Bayu Atmaja Darmawan,S.Kom .,M.Cs .
NIP. 198901272012121001

Sekretari Penguji

I Gede Arta Wibawa, S.T., M.Kom
NIP. 198310222008121001

Anggota Penguji

Dr. Anak Agung Istri Ngurah Eka Karyawati, S.Si.,M.Eng.
NIP. 197404071998022001

Anggota Penguji

Dr. I Ketut Gede Suhartana, S.Kom., M.Kom
NIP. 197201102008121001

Anggota Penguji

Komang Ari Mogi, S.Kom., M.Cs.
NIP. 198409242008011007

Mengetahui,
Komisi Seminar dan Tugas Akhir
Jurusan Ilmu Komputer
Ketua,

I Gusti Ngurah Anom Cahyadi Putra, S.T., M.Cs.
NIP. 198403172019031005

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala limpahan rahmat sehingga penulis dapat menyelesaikan proposal seminar tugas akhir dengan judul Pengamanan Perangkat IoT Dengan Encrypted Port Knocking

Terimakasih kami ucapkan kepada setiap pihak yang telah berkontribusi dalam membantu proses penyelesaian laporan ini, diantaranya:

1. Bapak Dr. I Ketut Gede Suhartana, S.Kom., M.Kom sebagai pembimbing 1 yang telah banyak membantu menyempurnakan proposal ini.
2. Bapak I Komang Ari Mogi, S.Kom., M.Kom sebagai pembimbing 2 yang telah banyak membantu menyempurnakan proposal ini.
3. Para dosen pengajar di program studi Teknik Informatika fakultas MIPA Universitas Udayana yang telah memberikan saran dan masukan dalam menyempurnakan proposal ini.
4. Serta semua pihak yang telah terlibat dan banyak membantu sehinnga proposal ini dapat terselesaikan.

Penulis menyadari bahwa laporan ini jauh dari kesempurnaan, oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun dari semua pihak, guna menyempurnakan isi dari laporan ini. Akhir kata penulis sampaikan terimakasih dan semoga laporan ini dapat bermanfaat bagi pembaca.

Bukit Jimbaran, 5 September 2019

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

1. Latar Belakang

Internet of things atau yang biasa disebut IoT merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet dari perangkat – perangkat fisik disekitarnya dan dapat bertukar informasi antara satu perangkat dengan perangkat lainnya. Contoh dari IoT adalah kulkas dan televisi dimana benda – benda ini telah tertanam sebuah sensor yang digunakan untuk berkomunikasi dan berinteraksi dengan orang lain melalui internet dan dapat dikendalikan dari jarak jauh. Pada era ini, teknologi IoT lebih banyak dimanfaatkan dalam konsep smart house atau bahkan yang lebih besar lagi yaitu smart city. Namun terdapat ancaman serius terhadap teknologi IoT yaitu masalah privasi dan keamanan data dari penggunaan perangkat yang terhubung dengan internet. Salah satu contoh serangan yang dapat mengancam keamanan data adalah Distributed Denial of Service (DDoS). DDoS adalah sebuah percobaan penyerangan dari beberapa sistem komputer yang menargetkan sebuah server agar jumlah traffic menjadi terlalu tinggi sehingga server tidak dapat menghandle requestnya. Serangan ini memanfaatkan internet untuk mengambil alih komputer. Mirai merupakan malware yang mampu mengubah sistem komputer yang menjalankan linux menjadi pengendali bot dari jarak jauh. Bot dapat melakukan berbagai tugas seperti memindai kerentanan perangkat lain, mengirim pesan email spam, atau melakukan berbagai jenis serangan. Target utama dari malware ini yaitu perangkat IoT online seperti cctv dan router rumahan. Mirai mampu melakukan beberapa jenis serangan DDoS antara lain SYN-flooding, UDP flooding, HTTP GET attacks, HTTP POST attack, dan serangan lainnya.

Contoh kasus penyerangan terhadap port IoT adalah mirai botnet yang menyerang pengguna internet di eropa. Mirai merupakan salah satu jenis worm yang sebelumnya berhasil menginfeksi jutaan CCTV dan perangkat IoT di seluruh dunia. Malware ini menyerang router rumahan yaitu router buatan Zyxel dan Speedport dengan port 7547 yang terbuka. Biasanya, *port* ini dipakai oleh penyedia internet untuk mengatur dan memperbaiki masalah *router* dari jarak jauh. Kode yang digunakan untuk menyerang router rumahan ini

didapatkan dari versi modifikasi mirai sebelumnya. Oleh karena itu perlu dilakukan pengamanan pada port IoT. Banyak serangan dilakukan melalui port yang terbuka menjadi salah satu ancaman bagi keamanan data dalam sistem jaringan komputer. Jika orang– orang yang tidak memiliki hak akses dapat mengendalikan port yang dimasuki maka hal ini dapat menjadi ancaman. Oleh karena itu firewall sangat dibutuhkan di dalam jaringan tersebut. Dalam firewall semua komunikasi yang keluar dan masuk dikontrol. Port yang tidak penting dapat diblokir dan port yang penting dan berbahaya juga dapat diblokir, sehingga hanya pihak yang diijinkan yang boleh mengakses port tersebut. Akan tetapi dalam beberapa kondisi pemblokiran yang dilakukan sering menjadi tidak fleksibel karena pada saat dibutuhkan untuk menjalin sebuah komunikasi firewall tidak mengijinkannya karena berada pada area yang tidak diijinkan. Untuk menghindari hal seperti ini digunakan sebuah metode yaitu metode port knocking. Kelebihan dari metode ini adalah meskipun semua port yang ada telah ditutup, tetapi user yang memiliki hak akses dan mengetahui knocking untuk membuka suatu port dapat menggunakan port yang telah ia buka.

Namun port knocking masih memiliki kelemahan dalam pengamanan port yaitu penyerang akan sangat mudah mengetahui ketukan rahasia yang valid dengan melakukan scanning maupun sniffing karena format port number berupa plain text, adanya DOS-Knocking yaitu kondisi ketika penyerang mengirim paket secara terus-menerus dengan random fake network address kepada server yang menyebabkan meningkatnya penggunaan memori secara signifikan dan dapat mengakibatkan server overload. Karena metode ini masih memiliki kelemahan, maka diperlukan keamanan ganda pada port IoT yaitu dengan kriptografi. Kriptografi merupakan sebuah ilmu yang mempelajari cara mengamankan dan menjaga suatu data. Dalam kriptografi terdapat empat aspek penting dalam keamanan informasi antara lain kerahasiaan, integritas data, autentikasi, dan non-repudiation. Aspek – aspek keamanan inilah menjadi dasar bahwa perlu dilakukannya pengamanan dengan melakukan enkripsi.

Dalam penelitian ini, penulis akan menggunakan metode port knocking dalam pengamanan port IoT. Untuk mengastasi kelemahan dari metode ini, digunakan kriptografi sebagai pengamanan ganda pada port IoT. Metode ini disebut encrypted port knocking.

2. Rumusan Masalah

Berdasarkan dari latar belakang tersebut, dapat dirumuskan beberapa rumusan masalah yaitu:

a. Bagaimana meningkatkan keamanan port knocking dengan enkripsi?
b. Bagaimana rancangan port pnocking dengan enkripsi?

3. Tujuan Penelitian

Berdasarkan rumusan masalah tersebut, tujuan dari penelitian ini yaitu:

a. Mengetahui peningkatan keamanan port knocking.

4. Batasan Masalah

Terdapat beberapa hal yang digunakan sebagai batasan permasalahan dalam penelitian ini yaitu:

a. Penelitian ini menggunakan metode RSA.
b. Penelitian ini dilakukan dengan simulasi menggunakan komputer.
c. Penelitian ini menggunakan port 1016 sebagai port IoT.
d. Diasumsikan kunci public telah ditukar.

5. Manfaat Penelitian

Beberapa manfaat yang didapatkan dari penelitian ini yaitu:

a. Bagi penulis

   Dari penelitian ini peneliti dapat menerapkan pengetahuan di bidang keamanan data pada IoT dan menjawab keingintahuan penulis mengenai masalah yang ditelitinya.

b. Bagi pihak lain

   Hasil penelitian ini nantinya diharapkan dapat bermanfaat, menjadi referensi untuk menambah wawasan mengenai pengamanan data pada IoT.

6. Tinjauan Pustaka

6.1 Tinjauan Studi

Pada penelitian ini menggunakan beberapa penelitian yang sudah dilakukan megenai pengamanan data sebagai tinjauan pustaka. Berikut beberapa penelitian tersebut adalah:

a. Aplikasi Pengendalian Port dengan Utilitas Port Knocking untuk Optimalisasi Sistem Keamanan Jaringan Komputer (Muzawi, 2016)

   Dijelaskan dalam penelitian ini, penulis melakukan pengamanan port dengan menggunakan metode port knocking. Cara kerja port knocking adalah dengan menutup semua port yang ada dan hanya user tertentu saja yang dapat mengakses port yang telah ditentukan yaitu dengan cara mengetuk terlebih dahulu. Tujuan utama port knocking ini adalah untuk mencegah attacker melakukan scanning port untuk mencari informasi mengenai port yang terbuka pada router/server. Dari hasil pengujian diperoleh hasil dari implementasi sistem terhadap server yaitu aplikasi pengendalian port dengan utilitas port knocking telah berhasil diuji dan menghasilkan kondisi port yang terbuka untuk digunakan. Dapat disimpulkan bahwa optimalisasi sistem keamanan jaringan komputer bisa memanfaatkan aplikasi pengendalian port ini

b. Penerapan Sistem Pengamanan Port pada Layanan Jaringan Menggunakan Port Knocking (Suchendra, 2017)

   Pada penelitian ini dijelaskan mengenai pengamanan port dengan menggunakan IP filter dan port knocking yang di dalamnya menggunakan fitur packet time out. Metode port knocking pada perancangan sistem ini menggunakan protocol TCP untuk melakukan autentikasi. Dari hasil pengujian yang telah dilakukan menggunakan metode port knocking yang dikombinasikan dengan firewall di Mikrotik dapat memberikan sistem keamanan autentikasi pada server layanan jaringan dan dapat mengamankan server dari 3

serangan yaitu Hydra, DoS, dan Telnet yang menggunakan ptotocol TCP.

c. Pengamanan Dokumen Menggunakan Metode RSA (Rivest Shamir Adleman) Berbasis Web (Agustina, 2017)

Pada penelitian ini menggunakan metode RSA sebagai algoritma yang digunakan untuk mengamankan suatu file. Hasil penelitian ini adalah aplikasi yang dibuat dapat mengubah file asli (plaintext) menjadi file terenkripsi(ciphertext) yang tidak dapat dibaca informasi dari filenya kemudian mengembalikannya kembali menjadi file aslinya (plaintext) tanpa merubah ataupun merusak isi file nya. Dengan melakukan pengujian diperoleh hasil yaitu ukuran file yang dienkripsi berpengaruh terhadap kecepatan proses enkripsi maupun proses dekripsi. Aplikasi yang dihasilkan dapat digunakan untuk dokumen office yang berformat doc,docx,txt,xls,ppt,dan pptx.

6.2 Internet of Thing

(Meutia, 2015) Internet of Things atau yang dikenal dengan IoT pertama kali diperkenalkan oleh Kevin Ashton pada tahun 1999 dan mulai terkenal melalui Auto-ID Center di MIT. Meski telah diperkenalkan sejak 15 tahun yang lalu, hingga kini belum ada sebuah konsensus global mengenai definisi IoT. Namun secara umum konsep IoT diartikan sebagai sebuah kemampuan untuk menghubungkan objek-obek cerdas dan memungkinkannya untuk berinteraksi dengan objek lain, lingkungan maupun dengan peralatan komputasi cerdas lainnya melalui jaringan internet. Contoh dari IoT adalah kulkas dan televisi dimana benda – benda ini telah tertanam sebuah sensor yang digunakan untuk berkomunikasi dan berinteraksi dengan orang lain melalui internet dan dapat dikendalikan dari jarak jauh.

6.3 Port

Dalam protokol jaringan TCP/IP, sebuah port adalah mekanisme yang mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi dengan komputer lainnya dan program di dalam jaringan. Port dapat mengidentifikasikan aplikasi dan layanan yang menggunakan koneksi di dalam jaringan TCP/IP. Sehingga, port juga mengidentifikasikan sebuah proses tertentu di mana sebuah server dapat memberikan sebuah layanan kepada klien atau bagaimana sebuah klien dapat mengakses sebuah layanan yang ada dalam server. Port dapat dikenali dengan angka 16-bit (dua byte) yang disebut dengan Port Number dan diklasifikasikan dengan jenis protokol transport apa yang digunakan, ke dalam Port TCP dan Port UDP. Karena memiliki angka 16-bit, maka total maksimum jumlah port untuk setiap protokol transport yang digunakan adalah 65536 buah

6.4 Firewall

Firewall memiliki tugas untuk melakukan pemblokiran terhadap port-port komunikasi yang terbuka bebas dalam sebuah jaringan computer. Dalam firewall semua komunikasi yang keluar dan masuk dikontrol. Port yang tidak penting dapat diblokir (ditutup) dan port yang penting dan berbahaya juga dapat diblokir, sehingga hanya pihak yang diijinkan saja yang boleh masuk melalui port tersebut. Cara ini merupakan sistem pengamanan jaringan komputer yang paling efektif dan banyak digunakan. Akan tetapi terkadang pemblokiran yang dilakukan sering menjadi tidak fleksibel, ketika dibutuhkan untuk menjalin komunikasi dengan apa yang ada di dalam jaringan, firewall tidak mengijinkannya karena berada pada area yang tidak diijinkan (Muzawi, 2016).

6.5 Port Knocking

(Suchendra, 2017) Port Knocking adalah metode yang digunakan untuk membuka akses ke port tertentu yang telah di block oleh firewall pada perangkat jaringan dengan cara mengirimkan paket atau koneksi tertentu. Koneksi yang dikirim dapat berupa protocol TCP dan UDP. Port hanya akan terbuka jika menggunakan sequence of request untuk nomor port yang telah ditentukan. Dengan menggunakan metode ini, maka perangkat jaringan akan lebih aman karena terdapat proses filtering terhadap port – port yang ada. Pada aplikasinya, jika user tidak memiliki akses ke port maka pada saat dilakukannya port scanning terhadap port – port yang ada maka port – port tersebut tidak akan dapat terlihat atau port tersebut terlihat tertutup. Mekanisme metode port knocking ini memiliki beberapa tahap, sebagai berikut:

1. Pada tahap pertama, klien melakukan koneksi ke komputer server ke salah satu port di komputer server, misal port 1016, namun koneksi tersebut di blok oleh firewall komputer server
2. Klien melakukan koneksi ke port – port sequences yang telah didefinisikan dalam file konfigurasi daemon port knocking ke komputer server dengan mengirimkan paket SYN didalamnya. Selama fase ini, klien tidak akan mendapatkan respon apa – apa
3. Selanjutnya Daemon Port Knocking mencatat adanya percobaan koneksi dan kemudian melakukan autentikasi terhadap percobaan tersebut. Apabila autentikasi sesuai dengan yang didefinisikan pada daemon port knocking dalam hal ini adalah port sequences yang didefinisikan, maka daemon port knocking akan melakukan overwrite terhadap rule yang telah didefinisikan didalam firewall agar membuka port yang ingin dituju oleh klien.
4. Setelah melakukan autentikasi klien dapat melakukan koneksi ke port yang dituju menggunakan aplikasi seperti pada umumnya.
5. Setelah selesai, klien memutuskan koneksi dengan port dan kemudian mengirimkan paket SYN kembali agar daemon port

knocking menulis ulang rule pada firewall agar tidak bisa dilakukan koneksi kembali ke port 1016.

6.6 Kriptografi

(Ginting, Isnanto & Windasari, 2015) Kriptografi (bahasa yunani "Cryptos" yaitu rahasia dan "graphein" yaitu tulisan) merupakan tulisan rahasia. Kriptografi merupakan sebuah ilmu yang mempelajari cara mengamankan dan menjaga suatu data. Terdapat Istilah istilah yang umum digunakan dalam kriptografi yaitu plaintext (M) merupakan pesan yang akan dikirim (berupa data asli), ciphertext (C) merupakan pesan tersandi dan hasil dari enkripsi, enkripsi merupakan proses pengubahan plaintext menjadi ciphertext, dekripsi merupakan proses yang mengubah ciphertext menjadi plaintext yang awalnya berupa data yang berbeda bentuk dari data awal menjadi data awal/asli, serta kunci merupakan suatu bilangan yang dirahasiakan dan digunakan dalam proses enkripsi serta dekripsi. Dalam kriptografi terdapat 4 aspek keamanan informasi yaitu :

1. Confidentiality (Kerahasiaan)

   Kerahasiaan adalah layanan yang digunakan untuk menjaga informasi dari setiap pihak yang tidak berwenang untuk mengaksesnya. Dengan demikian informasi hanya akan dapat diakses oleh pihak-pihak yang berhak saja.
2. Integritas Data

   Integritas berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi terjadinya manipulasi data. Manipulasi data yang dimaksud di sini meliputi penyisipan, penghapusan, maupun penggantian data.
3. Authentication

   Authentication berhubungan dengan identifikasi/pengenalan baik secara kesatuan sistem maupun informasi itu sendiri. Informasi yang

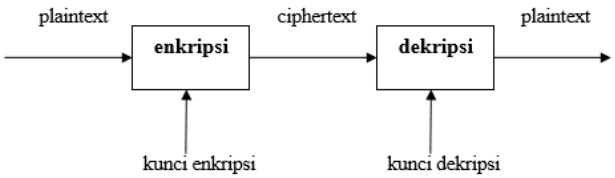
dikirimkan harus diautentikasi keaslian isi data, waktu pengiriman dan lainnya.

4. Non-repudiation

   Non-repudiation adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman suatu informasi oleh yang mengirim atau yang membuat.

Pada dasarnya algoritma kriptografi dibedakan menjadi dua jenis jika dilihat berdasarkan jenis kunci yang digunakan antara lain algoritma simetris dan algoritma asimetris. Algoritma asimetris menggunakan kunci yang berbeda pada saat proses enkripsi dan proses dekripsi. Dalam algoritma simetris terdapat dua jenis kunci yaitu kunci publik dan kunci rahasia. Sedangkan algoritma simetris menggunakan kunci yang sama untuk proses enkripsi serta proses dekripsi.

Dalam kriptografi sendiri terdapat dua hal penting yaitu proses enkripsi dan proses dekripsi. Proses enkripsi merupakan proses dimana informasi atau data yang dikirim diubah menjadi bentuk yang tidak diketahui dengan menerapkan algoritma tertentu. Data yang bisa dimengerti biasa disebut dengan plainteks dan data yang bentuknya berbeda dan tidak dapat dimengerti biasa disebut cipherteks. Sedangkan proses dekripsi adalah proses dimana data atau informasi yang sebelumnya telah diubah menjadi bentuk yang tidak diketahui diubah kembali menjadi data atau informasi awal yang disampaikan. Berikut adalah diagram proses dari enkripsi dan dekripsi.



Gambar 1. Proses Enkripsi dan Dekripsi

6.7 Metode RSA

6.7.1 Proses Pembentukan Kunci

(Agustina, 2017) Metode RSA diciptakan oleh 3 orang peneliti dari Massachussets Institute of Technology (MIT) yaitu Ron Rivest, Adi Shamir dan Leonard Adleman pada tahun 1977. Besaran-besaran yang digunakan pada algoritma RSA antara lain:

1. p dan q bilangan prima (rahasia)
2. $n = p \times q$ (tidak rahasia)
3. $\varphi(n) = (p-1) \times (q-1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci dekripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (chiperteks) (tidak rahasia)

Proses pembangkitan kunci algoritma RSA memiliki dua kunci yang berbeda untuk proses enkripsi dan dekripsi.

Adapun langkah - langkah terkait perhitungan penbentukan kunci yaitu:

1. Menentukan 2 bilangan prima, dengan nama p dan q.
2. Menghitung nilai modulus (n) dengan rumus $n = p \times q$.
3. Menghitung nilai totient atau phi(φ) dari n dengan rumus $\varphi(n) = (p - 1)(q - 1)$
4. Menentukan nilai e dimana nilai e ini merupakan bilangan prima dan nilai e harus sesuai dengan syarat $1 < e < \varphi(n)$. Untuk pembuktian terhadap nilai e dapat dilakukan perhitungan dengan rumus gcd $(e, \varphi(n)) = 1$.
5. Mencari nilai deciphering exponent (d) dengan rumus $d = (1 + (k \times \varphi(n))/e$
6. Setelah menemukan nilai n, e, dan d maka didapatkan pasangan kunci yaitu pasangan kunci publik dan pasangan

kunci rahasia. Pasangan kunci publik (n,e) dan pasangan kunci rahasia (n,d).

6.7.2 Proses Enkripsi

Pada proses ini, dilakukan konversi pesan dari plainteks ke dalam kode ASCII dengan menggunakan tabel ASCII untuk melihat kode sesuai dengan planteks yang ada. Selanjutnya adalah mencari nilai C dengan menggunakan rumus $c = m_i{}^e \ mod \ n$. Kunci yang digunakan pada proses ini yaitu kunci publik (n,e). Dari rumus perhitungan tersebut maka akan ditemukan nilai dari c (cipertext). Berikut ini adalah gambar dari tabel ASCII.

ASCII table

Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(nul)	0	0000	0x00	(sp)	32	0040	0x20	@	64	0100	0x40	`	96	0140	0x60
(soh)	1	0001	0x01	!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
(stx)	2	0002	0x02	"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
(etx)	3	0003	0x03	#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
(eot)	4	0004	0x04	$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
(enq)	5	0005	0x05	%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
(ack)	6	0006	0x06	&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
(bel)	7	0007	0x07	'	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(bs)	8	0010	0x08	(	40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
(ht)	9	0011	0x09	)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
(nl)	10	0012	0x0a	*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
(vt)	11	0013	0x0b	+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
(np)	12	0014	0x0c	,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
(cr)	13	0015	0x0d	-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
(so)	14	0016	0x0e	.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
(si)	15	0017	0x0f	/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
(dle)	16	0020	0x10	0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
(dc1)	17	0021	0x11	1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
(dc2)	18	0022	0x12	2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
(dc3)	19	0023	0x13	3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73
(dc4)	20	0024	0x14	4	52	0064	0x34	T	84	0124	0x54	t	116	0164	0x74
(nak)	21	0025	0x15	5	53	0065	0x35	U	85	0125	0x55	u	117	0165	0x75
(syn)	22	0026	0x16	6	54	0066	0x36	V	86	0126	0x56	v	118	0166	0x76
(etb)	23	0027	0x17	7	55	0067	0x37	W	87	0127	0x57	w	119	0167	0x77
(can)	24	0030	0x18	8	56	0070	0x38	X	88	0130	0x58	x	120	0170	0x78
(em)	25	0031	0x19	9	57	0071	0x39	Y	89	0131	0x59	y	121	0171	0x79
(sub)	26	0032	0x1a	:	58	0072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
(esc)	27	0033	0x1b	;	59	0073	0x3b	[	91	0133	0x5b	{	123	0173	0x7b
(fs)	28	0034	0x1c	<	60	0074	0x3c	\	92	0134	0x5c	|	124	0174	0x7c
(gs)	29	0035	0x1d	=	61	0075	0x3d	]	93	0135	0x5d	}	125	0175	0x7d
(rs)	30	0036	0x1e	>	62	0076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
(us)	31	0037	0x1f	?	63	0077	0x3f	_	95	0137	0x5f	(del)	127	0177	0x7f

Gambar 2. Tabel ASCII

6.7.3 Proses Dekripsi

Pada proses ini akan mengubah ciphertext yang telah diperoleh sebelumnya menjadi data awal yaitu berupa plaintext. Proses dekripsi dilakukan dengan menggunakan rumus $m = m_i{}^d \ mod \ n$. Setelah ditemukan nilai m dengan menggunakan

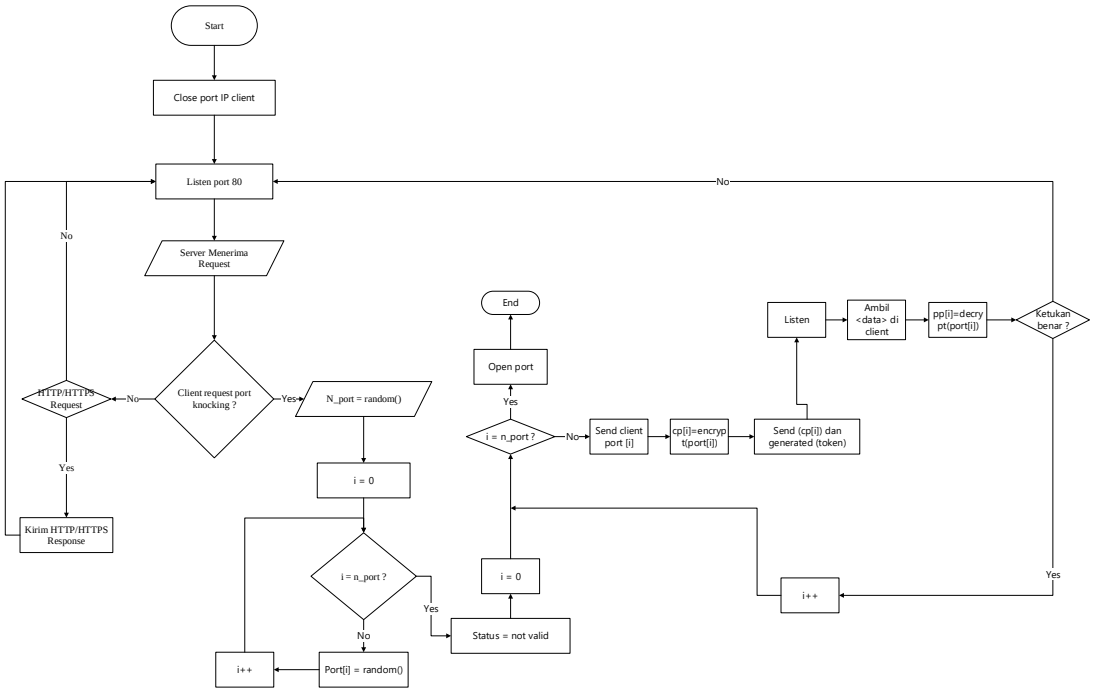
rumus perhitungan untuk dekripsi maka akan ditemukan pula plainteks dari pesan yang dikirim.

7. Metodologi Penelitian

Pada metodologi penelitian ini akan dijelaskan langkah – langkah dalam penelitian ini. Subbab bahasan yang akan dijelaskan meliputi desain penelitian, metode yang digunakan, serta pengujian dan evaluasi.

7.1 Desain Penelitian

Dalam permasalahan mengenai keamanan port, penulis akan melakukan peningkatan keamanan pada port perangkat IoT dengan menggunakan metode port knocking serta kriptografi menggunakan metode RSA. Sebelum *client* dapat mengakses perangkat IoT, maka akan dilakukan pembangkitan random port pada perangkat IoT kemudian port akan dikirimkan ke *client* agar port tersebut dapat diketuk oleh *client* untuk membuka *port* tujuan. Lalu port akan di enkripsi dan cipher port akan dikirim bersamaan dengan kunci yang dibangkitkan. Setelah itu port akan di dekripsi dimana data yang dienkripsi diambil di client. Contoh *port* tujuan yang akan di akses adalah port 1016. Berikut adalah proses membuka port pada server



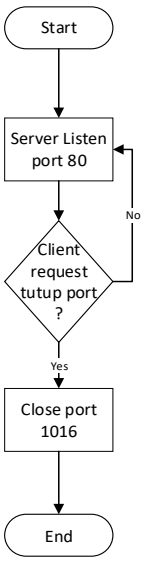
Gambar 3. Flowchart port knocking pada server

Berikut adalah penjelasasn flowchart server:

1. Pertama server akan menutup ip client tertentu untuk membuka port.
2. Server akan berada pada kondisi listening untuk siap menerima input dari client.
3. Server menerima request dari client dan diuji apakah request yang diterima adalah untuk port knocking atau HTTP/HTTPS request.
4. Jika HTTP/HTTPS request yang diterima maka HTTP/HTTPS response yang dikirimkan oleh server ke client dan kembali ke keadaan listen.
5. Jika iya maka server menginisialisasi n_port = random()
6. Lalu memproses i = 0.
7. Lalu masuk ke dalam proses generated random port, apa i = n_port, jika tidak maka random disimpan pada port array di index i dan terus berulang sampai array yang ditentukan
8. Status pada server masih not valid karena masih belum terjadi proses pengetukan
9. Proses i = 0

10. Masuk ke kondisi pengetukan port, apa i = n_port ?, jika tidak maka port pada array di index i dikirimkan ke client
11. Kemudian port akan dienkripsi menjadi bentuk cipher
12. Cipher port dan generated token akan dikirimkan ke client
13. Server kembali ke keadaan listen
14. Server mengambil data di client
15. Server akan melakukan dekripsi port menjadi bentuk plain port
16. Jika ketukan benar maka proses pengetukan akan berulang sebanyak port yang ada pada array di index i.
17. Jika tidak maka server mengirimkan status not valid ke client dan kembali ke keadaan listen port 80
18. Jika iya i = n_port maka buka port tujuan (setting firewall)

Berikut flowchart proses tutup port pada server



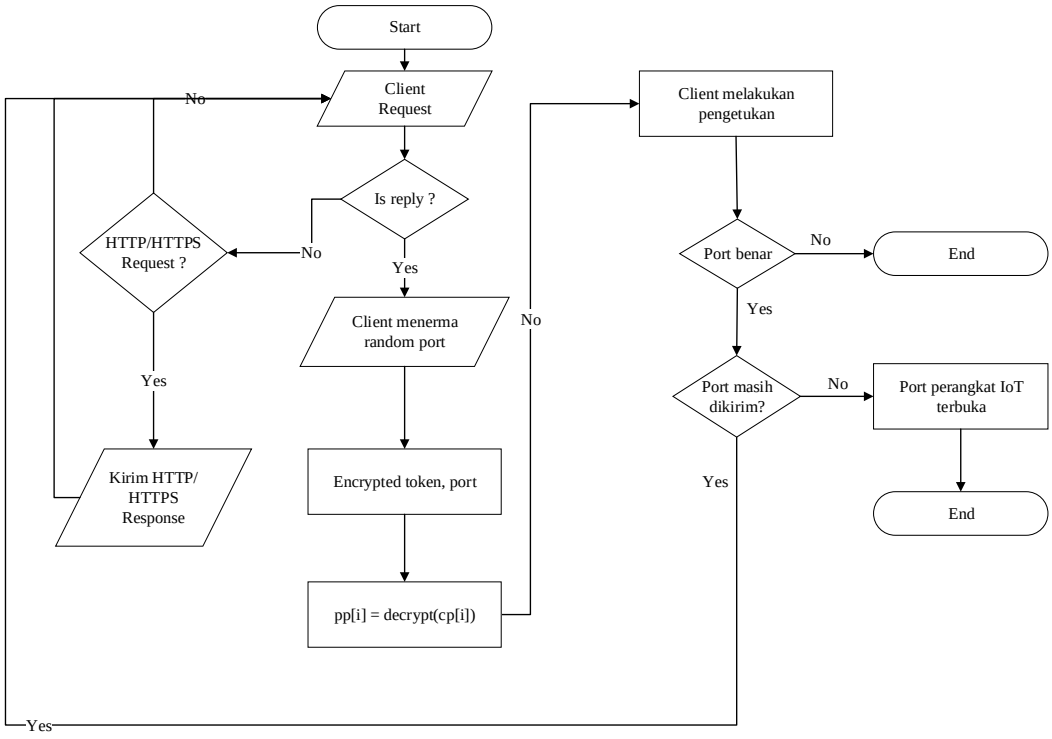
Gambar 4. Flowchart menutup port pada server

Penjelasan flowchart proses menutup port pada server

1. Pada saat menutup port 1016, Server dalam keadaan listening
2. Kemudian server menerima request untuk menutup port dari client

3. Jika server menerima request untuk menutup maka port 1016 akan ditutup. Namun jika tidak maka akan kembali dalam keadaan listening.

Flowchart selanjutnya adalah flowchart proses membuka pada client



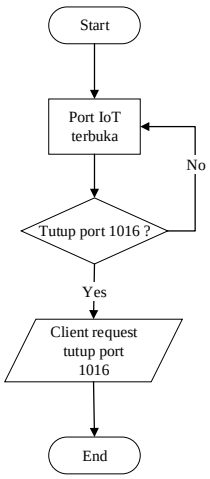
Gambar 5. Flowchart pengetukan oleh client

Berikut adalah penjelasan flowchart diatas:

1. Pada proses pertama client akan merequest menuju ke server
2. Jika iya client menerima generated random port
3. Jika tidak apa HTTP/HTTPS request ?
4. Jika iya kirim HTTP/HTTPS response dan kembali ke client request.
5. Jika tidak maka langsung kembali ke client request
6. Client melakukan encrypted generated port
7. Port di dekripsi menjadi bentuk plain port
8. Client melakukan melakukan proses pengetukan
9. Jika ketukan benar maka client akan lanjut mengetuk port, jika tidak maka proses pengetukan selesai.

10. Jika port masih dikirim maka proses request port akan terus berulang sampai port yang dikirim oleh server habis dan jika tidak maka port IoT terbuka.

Flowchart selanjutnya adalah flowchart proses membuka pada client



Gambar 6. Flowchart menutup port pada client

Berikut adalah penjelasan dari flowchat penutupan port:

1. Client ingin menutup port IoT yang terbuka yaitu port 1016
2. Jika client ingin menutup maka client akan merequest untuk menutup port 1016 (dilakukan pada setting firewall)
3. Port 1016 tertutup.

7.2 Metode yang Digunakan

7.2.1 Metode Port Knocking

Port Knocking adalah metode yang digunakan untuk membuka akses ke port tertentu yang telah di block oleh firewall pada perangkat jaringan dengan cara mengirimkan paket atau koneksi tertentu. Port yang ada tidak akan dapat dilihat oleh pengguna lain yang tidak memiliki akses ke port tersebut. Cara kerja dari port knocking sendiri adalah dengan cara mengetuk port yang ingin diakses. Format ketukan pada pembuatan system

ini adalah 3 port untuk ketukannya. Port tujuan yang ingin diakses adalah port 1016.

7.2.2 Metode RSA

a. Proses Pembentukan Kunci

Dalam mengamankan suatu data diperlukan kunci untuk digunakan pada proses enkripsi dan proses dekripsi. Pada proses pembangkitan kunci ini menggunakan metode RSA dimana nantinya akan ditemukan dua buah kunci yaitu kunci publik dan kunci rahasia. Kunci rahasia akan disimpan oleh pengirim data atau informasi sedangkan kunci publik dikirimkan kepada orang lain (penerima).

b. Enkripsi

Pada proses enkripsi ini menggunakan metode RSA. Data yang dimasukkan akan diolah menjadi data dalam bentuk cipherteks. Fungsi dari proses enkripsi ini adalah untuk memastikan data yang dikirim aman dan tidak mudah tersebar ke orang lain kecuali penerima pesan.

c. Dekripsi

Pada proses ini data yang telah diolah menjadi data dalam bentuk cipherteks akan diubah kembali menjadi data dalam bentuk plainteks agar penerima dapat memahami isi dari data tersebut. Metode yang digunakan pada proses ini adalah metode RSA.

8. Jadwal Penelitian

Pelaksanaan dari kegiatan penelitian yang penulis lakukan menghabiskan waktu selama tiga bulan. Rincian dari kegiatan yang dilakukan dapat dilihat dari tabel di bawah ini.

Tabel 1. Jadwal Pelaksanaan Penelitian

No.	Kegiatan	Minggu ke-											
		1	2	3	4	5	6	7	8	9	10	11	12
1	Studi literatur												
2	Pengumpulan data												
3	Perancangan sistem												
4	Pembuatan sistem												
5	Pengujian sistem												
6	Penulisan laporan penelitian												

DAFTAR PUSTAKA

Agung, H., & Prasta, I. Implementasi Algoritma Rivest, Shamir, Adleman Untuk File Sharing Pada PT. Sumber Makmur Pangan Sejahtera Berbasis Web.

Agustina, A. N. (2017). Pengamanan Dokumen Menggunakan Metode RSA (Rivest Shamir Adleman) Berbasis Web.

Amarudin, A., & Ulum, F. (2018). Desain Keamanan Jaringan Pada Mikrotik Router OS Menggunakan Metode Port Knocking. *Jurnal Teknoinfo*, *12*(2), 72-75.

Aminudin, A., Helmi, A. F., & Arifianto, S. (2018). Analisa Kombinasi Algoritma Merkle-Hellman Knapscak dan Logaritma Diskrit pada Aplikasi Chat. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *5*(3), 325-334.

Arief, M., Fitriyani, F., & Ikhsan, N. (2015). Kriptografi Rsa Pada Aplikasi File Transfer Client-Server Based. *Jurnal Ilmiah Teknologi Informasi Terapan*, *1*(3).

Ginting, A., Isnanto, R. R., & Windasari, I. P. (2015). Implementasi Algoritma Kriptografi RSA untuk Enkripsi dan Dekripsi Email. *Jurnal Teknologi dan Sistem Komputer*, *3*(2), 253-258.

Haryanto E., Widyawan, & Adhipta D. (2016). Meningkatkan keamanan port knocking dengan kombinasi special features icmp, source port, dan tunneling. 187–194.

Meutia, E. D. (2015). Internet of things–Keamanan dan Privasi. In *Seminar Nasional dan Expo Teknik Elektro* (pp. 85-89).

Muzawi, R. (2016). Aplikasi Pengendalian Port dengan Utilitas Port Knocking untuk Optimalisasi Sistem Keamanan Jaringan Komputer. *SATIN-Sains dan Teknologi Informasi*, *2*(1), 52-58.

Stallings, W. (2017). *Cryptography & Network Security GE*. Pearson Australia Pty Ltd.

Suchendra, D. R., Rahman, A. F., & Ismail, S. J. I. (2017). PENERAPAN SISTEM PENGAMANAN PORT PADA LAYANAN JARINGAN MENGGUNAKAN PORT KNOCKING. *Jurnal Komputer Bisnis*, *10*(2).

Wahyuni, A. (2011). Keamanan Pertukaran Kunci Kriptografi dengan Algoritma Hybrid: Diffie-Hellman dan RSA. *Majalah Ilmiah Informatika*, *2*(2).

Yalisa, N., Arhami, M., & Azhar, A. (2019, January). Algoritma Elgamal dengan Pertukaran Kunci Diffie Hellman pada Aplikasi Keamanan Citra Sidik Jari Berbasis Android. In *Prosiding Seminar Nasional Politeknik Negeri Lhkseumawe* (Vol. 2, No. 1).