PERBANDINGAN KINERJA ALGORITMA KRIPTOGRAFI RSA DAN AES UNTUK KEAMANAN INFORMASI PERANGKAT KOMUNIKASI ZIGBEE

Murowe Hakiki, Lalu A. Syamsul Irfan Akbar, Misbahuddin Misbahuddin

Teknik Elektro Fakultas Teknik Universitas Mataram

Email : misbahuddin@unram.ac.id

ABSTRAK

Data atau informasi yang sangat penting membutuhkan keamanan, sehingga akan terjaga dari pihak yang tidak bertanggung jawab. Internet of Things (IoT) merupakan teknologi yang sedang berkembang, namun keamanan yang rentan. Arduino merupakan salah satu komputer kecil yang digunakan pada perangkat IoT, sehingga kapasitas memori arduino merupakan memori pada perangkat IoT. Kriptografi merupakan sebuah cara penyandian data yang akan menyembunyikan data yang asli sekalipun sudah ditemukan oleh orang lain. Penelitian ini bertujuan membandingkan kinerja kriptografi Rivest Shamir Adleman dan Advanced Encryption Standard pada perangkat komunikasi Zigbee. Pengujian dilakukan dengan mengenkripsi plaintext dengan ukuran yang sama dan akan dibandingkan waktu komputasi, penggunaan memori antara kedua algoritma. Hasil dari penelitian ini menunjukkan algoritma yang lebih baik digunakan pada perangkat komunikasi ZigBee adalah algoritma Advanced Encryption Standard dengan waktu komputasi lebih cepat dan penggunaan memori lebih sedikit dibanding algoritma Rivest Shamir Adleman.

Kata Kunci :Zigbee, Kriptografi, Advanced Encryption Standard, Rivest Shamir Adleman, Internet of Things

ABSTRACT

Important data need a security, it will maintain from irresponsible parties. Internet of Things is a developing technology, but it is not save technology. Arduino is one of small computer that used in IoT device, Arduino memory capacity is memory in IoT device. Cryptography is an encoding data methode that will hide original data even someone found it. This study purpose is compare cryptography Rivest Shamir Adleman and Advanced Encryption Standard performance to Zigbee communication device. Testing by encrypt same size plaintexts and compare computing time, memory usage between two algorithms. In this research, the best algorithm for Zigbee communication device is Advanced Encryption Standard algorithm with faster computation time and less memory usage than Rivest Shamir Adleman.

Keywords : Zigbee ,Crypthography, Advanced Encryption Standard, Rivest Shamir Adleman, Internet of Things..

I. PENDAHULUAN

Internet of Thing (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang dari konvergensi teknologi nirkabel, *micro-electromechanical systems* (MEMS), dan Internet.

IoT merupakan teknologi yang terus berkembang saat ini, tapi seiring dengan perkembanagannya timbul masalah yang berbahaya bagi perangkat IoT. Sekalipun perangkat IoT terus berkembang semakin canggih namun dibagian keamanan perangkat IoT sangat lemah dan rentan di retas, apabila perangkat IoT di ambil alih oleh orang lain maka semua yang tersambung dengan IoT akan di akses dan dikendalikan. Perangkat IoT perangkat dengan kapasitas memori yang kecil sehingga dibutuhkan sebuah pengaman yang dirancang untuk penggunaan memori yang kecil untuk mendapatkan kecepatan proses yang maksimal, hal ini karena IoT juga termasuk perangkat dengan proses yang lambat.

Masalah keamanan untuk perangkat IoT merupakan suatu hal yang membutuhkan perhatian bagi pengembang IoT, beberapa penelitian tentang keamanan perangkat IoT seperti penelitian yang membahas masalah keamanan dan privasi dalam perangkat IoT yang dapat mengancam entitas IoT serta dapat merugikan dan membahayakan perangkat, sehingga dibutuhkan teknik mitigasi yang ampuh dan berbagai metode kriptografi untuk mengatasi kelemahan keamanan dan privasi tersebut(Meutia, 2015). Adapun pada penelitian yang lain menggunakan algoritma RSA untuk keamanan data pada perangkat IoT yang berfokus pada pengiriman data dari mukrikontroler ke server, dimana pada proses transmisis ini sangat rentan terhadap gangguan sehingga digunakan algoritma ini untuk enkripsi data yang dikirim dari mikrokontroller ke server (Aseng dkk,2017). Sebuah penelitian menyebutkan tiga bagian perangkat IoT yang rentan keamanannya sehingga dibuat sebuah metode keamanan yang memberikan perlindungan khusus terhadap tiga bagian ini, karena bagian tersebut merupakan bagian yang paling pital apabila tidak diberikan keaman yang baik. Sistem keamanan tersebut dinamakan CIA *(Confidentiality Integrity Availability) Triad*(Farooq dkk, 2015). Terakhir yaitu penelitian untuk mengoreksi 10 celah kelemahan yang berbahaya untuk keamanan perangkat IoT dengan menggunakan OWASP *Top Ten*. OWASP *Top Ten* sendiri adalah sebuah dokumen yang merangkum 10 celah keamanan paling berbahaya pada suatu aplikasi, sehingga dengan menggunakan OWASP *Top Ten* kita bisa mengetahui bagian mana yang harus mendapatkan keamanan yang baik (Giantara, 2016).

Dari berbagai penelitian yang dilakukan menyatakan bahwa keamanan perangkat IoT sangat rentan ditembus sehingga hal ini perlu diatasi dengan mengembangkan sistem keamanan yang tepat untuk perangkat IoT. Sehingga pada penelitian ini diharapkan bisa memberikan sebuah solusi keamanan yang tepat dan baik untuk perangkat IoT

II. Kriptografi

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data serta otentikasi. Kata "seni" didalam definisi di atas berasal dari fakta sejarah bahwa pada masa masa awal sejarah kriptografi, setiap orang mungkin mempunyai cara yang unik untuk merahasiakan pesan. Cara-cara unik tersebut mungkin berbeda-beda pada setiap pelaku kriptografi sehingga setiap cara menulis pesan rahasia itu mempunyai nilai estetika tersendiri sehingga kriptografi berkembang menjadi sebuah seni merahasiakan pesan (kata "*graphy'* didalam *"cryptography''* itu sendiri sudah menyiratkan sebuah seni).

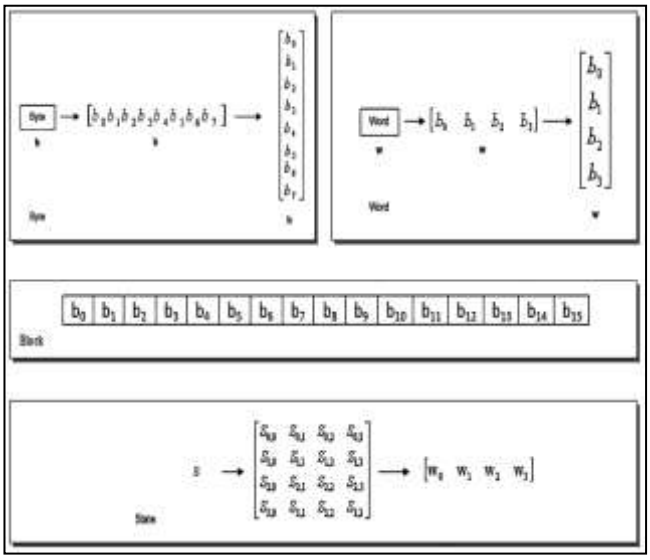
Pada perkembangan selanjutnya, kriptografi berkembang menjadi sebuah disiplin ilmu sendiri karena teknik-teknik kriptografi dapat diformulasikan secara matematik sehingga menjadi sebuah metode yang formal.

II.1 Advanced Encryption Standard (AES) – Rijndael

AES menggunakan 5 unit ukuran untuk data: *bit*, *byte*, *word*, *blocks*, dan *state*. Gambar 2.2 menunjukkan unit data selain *bit*.

- *Bit*

  Dalam AES *bit* adalah bilangan biner yang bernilai 0 atau 1. *Bit* adalah unit data terkecil. *Bit* disimbolkan dengan b



Gambar 1 Unit Data dalam AES

- *Byte*

  Byte adalah kumpulan delapan *bit* yang dapat dianggap sebagai sebuah entitas, contohnya: satu baris matriks (1x8) delapan *bit* atau satu kolom matriks (8x1) delapan *bit*. *Byte* disimbolkan dengan **b** (b tebal).Untuk lebih jelasnya byte bisa dilihat pada gambar 2.2(a) dibawah ini.

- *Word*

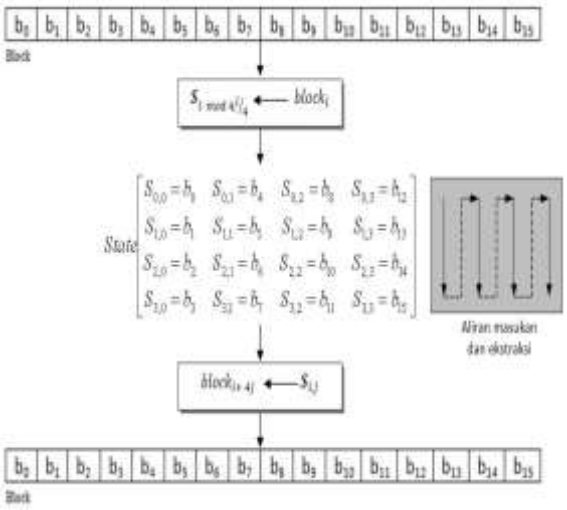
  Word adalah kumpulan 32-bit yang dapat dianggap sebagai sebuah entitas, contohnya: satu baris matriks 4-byte atau satu kolom matriks 4-byte. Jika sebuah baris matriks, maka *byte* di-*input*-kan dari kiri ke kanan, jika sebuah kolom matriks, maka *byte* di-*input*-kan dari atas ke bawah, *word* di simbolkan dengan **w** (w tebal). Untuk lebih jelasnya tentang word bisa dilihat pada gambar 2.2(b) diatas, dimana dengan gambar itu sudah menjelaskan lambang dan tampilan dari *word,* bisa juga dilihat satu word terdiri dari 4 byte atau 32 bit.

- *Block*

  AES mengenkripsi dan mendekripsi *block* data. *Block* dalam AES adalah kumpulan dari 128-bit. *Block* juga dapat direpresentsikan sebagai baris dalam matriks 16-byte. Gambar 2 merupakan illustrasi yang menggambarkan sebuah blok.

- *State*

  gambar 2 merupakan *state* disimbolkan dengan **S** (S tebal). Sama seperti *block*, *state* berukuran 16-byte, tetapi biasanya dalam matriks 4x4-byte. Jadi setiap elemen dari *state* dinotasikan dengan $S_{r,c}$, dimana *r* (0-3) indeks dari baris dan *c* (0-3) indeks dari kolom. Pada awal proses enkripsi, *byte* dalam *block* data dimasukkan ke perkolom dari *state*, dari atas ke bawah. Pada akhir proses enkripsi, *byte* dalam *state* dikembalikan ke *block* dengan cara yang sama seperti ditunjukkan pada Gambar 2.3.



Gambar 2 Transformasi *Block* ke *State* dan *State* ke *Block*

II.2 (Rivest Shamir Adleman)RSA

Algoritma RSA diperkenalkan oleh tiga orang peneliti dari MIT (*Massachussets Institute of Technology*), yaitu Ron Rivest, Adi Shamir, dan Len Adleman, pada tahun 1976. RSA mendasarkan proses enkripsi dan deskripsinya pada konsep bilangan prima dan aritmatika modulo. Baik kunci enkripsi maupun kunci deskripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diketahui umum (sehingga dinamakan juga kunci publik), namun kunci

untuk deskripsi bersifat rahasia. Kunci deskripsi dibangkitkan oleh beberapa buah bilangan prima bersama-sama dengan kunci enkripsi. Untuk menemukan kunci enkripsi, seseorang harus memfaktorkan suatu bilangan non prima menjadi faktor primanya. Kenyataannya, memfaktorkan bilangan nonprime menjadi faktor primanya bukanlah pekerjaan yang mudah. Belum ada algoritma yang efisien yang ditemukan untuk pemfaktoran itu. Semakin besar bilangan non-primanya tentuakan semakin sulit menemukan faktor primanya. Semakin sulit pemfaktorannya, semakin kuat pula algoritma RSA.

II.3 Zigbee

Spesifikasi untuk jaringan protokol komunikasi tingka tinggi, dengan daya rendah, untuk jaringan personal nirkabel tingkat rendah, seperti saklar lampu nirkabel dengan lampu, alat pengukur listrik dengan inovasi *In-Home Display* (IHD), serta perangkat-perangkat elektronik konsumen lainnya yang menggunakan jaringan radio jarak dekat dengan daya transfer data tingkat rendah. Teknologi yang memenuhi spesifikasi dari *ZigBee* adalah perangkat dengan pengoperasian yang mudah, sederhana, membutuhkan daya sangat rendah serta biaya yang murah jika dibandingkan dengan WPANs lainnya, yakni *Bluetooth. ZigBee* fokus pada aplikasi Radio *Frequency* (RF) yang membutuhkan data tingkat rendah, baterai tahan lama, serta jaringan yang aman. *ZigBee* adalah jaringan mesh nirkabel dengan daya rendah dan biaya yang murah. Pertama, biayanya yang murah memungkinkan teknologi ini banyak digunakan sebagai pengendali jaringan nirkabel dan aplikasi pemantauan.Kedua, penggunaannya dengan daya yang rendah membuatnya dapat bertahan lama bahkan dengan baterai berukuran lebih kecil.Ketiga, jaringan mesh memberikan realibilitas yang tinggi serta jangkauan yang lebih luas.

III. Pengujian

Pengujian dilakukan dengan tujuan untuk menemukan data performa kecepatan dan kebutuhan memori yang digunakan pada saat enkripsi maupun dekripsi. Pengujian ini dilakukan dengan menggunakan dua buah komputer sebagai *interface*, perangkat Arduino sebagai media penyimpan program enkripsi maupun dekripsi yang akan dijalankan dan di lengkapi dengan perangkat Xbee yang sudah di konfigurasi dengan menggunakan *software* XCTU dan dihubungkan dimasing-masing Arduino. Untuk mengetahui waktu yang dibutuhkan

pada proses enkripsi bisa dilihat pada tampilan setelah proses pengiriman, sedangkan untuk mengetahui berapa memori Arduino yang digunakan pada saat melakukan enkripsi maupun dekripsi, pada penelitian ini menggunakan WINAVR. Dengan WINAVR ini kita bisa mengetahui *flash* memori yang digunakan maupun SRAM yang digunakan

Tabel 1 Pengujian Algoritma AES

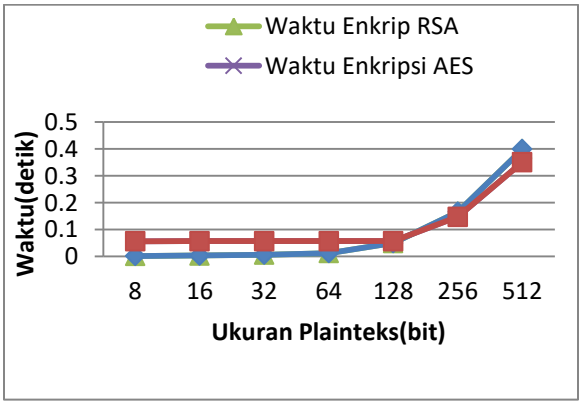
Plaintexts	Kunci	Enkripsi			Dekripsi		
		Waktu(s)	Memori		Waktu(s)	Memori	
			bytes	%		bytes	%
8 bit	128 bit	0.056108	1256	15.7	0.055648	2149	26.86
16 bit		0.056928	1256	15.7	0.055644	2149	26.86
32 bit		0.056972	1262	15.78	0.055656	2149	26.86
64 bit		0.056972	1266	15.82	0.055660	2149	26.86
128 bit		0.056880	1274	15.92	0.055648	2149	26.86
256 bit		0.147156	1290	16.1	0.099040	2165	27
512 bit		0.351048	1322	16.5	0.185780	2497	31.3

Tabel 1 merupkan hasil pengujian algoritma AES, dimana semakin besar ukuran *Plaintexts* yang kita gunakan maka waktu yang dibutuhkn dalam proses enkripsi semakin lama, begitu juga dengan kebutuhan memori yang digunakan pada saat enkripsi, ukuran *plaintexts* dan kunci yang digunakan mempengaruhi kecepatan pada proses enkripsi maupun dekripsi, kecepatan proses enkripsi lebih lambat dari pada proses dekripsi. Sesuai dengan penelitian terdahulu tentang algoritma AES yaitu analisis algoritma

tabel 2 merupakan hasil pengujian algoritma RSA, didapatkan bahwa semakin besar ukuran *Plaintexts* yang kita gunakan maka waktu yang dibutuhkan dalam proses enkripsi semakin lama, begitu juga dengan kebutuhan memori yang digunakan pada saat enkripsi, semakin besar *Plaintexts* maka memori yang digunakan semakin besar juga. Begitu juga dengan kecepatan dan memori pada proses dekripsi dipengaruhi oleh besarnya *plaintexts* dan kunci yang digunakan.

Tabel 2 Pengujian Algoritma RSA

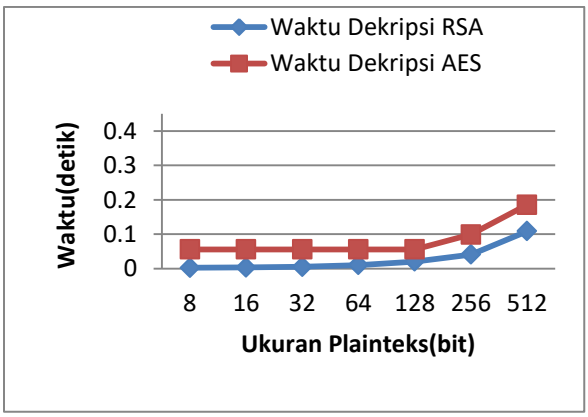
Plaintexts	Kunci	Enkripsi			Dekripsi		
		Waktu(s)	Memori		Waktu(s)	Memori	
			bytes	%		bytes	%
8 bit	8 bit	0.0008	351	4.4	0.001724	373	4.7
	16 bit	0.001528	351	4.4	0.002088	373	4.7
16 bit	8 bit	0.001732	351	4.4	0.002624	373	4.7
	16 bit	0.00306	351	4.4	0.003292	373	4.7
32 bit	8 bit	0.003296	351	4.4	0.004632	373	4.7
	16 bit	0.006112	351	4.4	0.005160	373	4.7
64 bit	8 bit	0.00652	359	4.5	0.007812	541	6.8
	16 bit	0.012	359	4.5	0.010	841	10.5
128 bit	8 bit	0.015636	367	4.6	0.015264	849	10.6
	16 bit	0.049672	367	4.6	0.020544	849	10.6
256 bit	8 bit	0.096512	383	4.9	0.030168	865	10.8
	16 bit	0.16562	383	4.9	0.040808	865	10.8
512 bit	8 bit	0.2614	415	5.2	0.087408	897	11.2
	16 bit	0.3996	415	5.2	0.108656	897	11.2



Gambar 3 Waktu Proses Enkripsi RSA dan AES

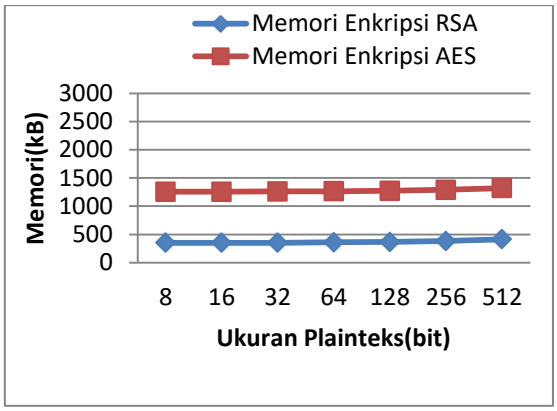
Gambar 3 menunjukkan hasil dari penelitian kecepatan komputasi pada proses enkripsi pada algoritma RSA dan AES, sesuai grafik diatas waktu komputasi proses enkripsi dengan algoritma RSA dan AES hampir sama, sekalipun pada rentang plaintext 8 bit hingga 128 bit menunjukkan algoritma RSA lebih cepat dan pada bagian akhir plainteks 256 bit dan 512 bit menunjukkan waktu ekripsi dengan algoritma RSA lebih lambat daripada waktu

enkripsi dengan algoritma AES. Untuk rata-rata kecepatan pada kedua algoritma pada proses enkripsi ini menunjukkan rata-rata kecepatan pada algoritma RSA adalah 0,25 detik sedangkan rata-rata kecepatan dari algoritma AES adalah 0,23 detik. Dari hasil rata-rata kecepatan setiap algoritma menunjukkan bahwa proses enkripsi dengan algoritma AES lebih cepat, walaupun pada algoritma AES menggunakan kunci yang jauh lebih besar dapripada algoritma RSA.



Gambar 4 Waktu Prosesi Dekripsi RSA dan AES

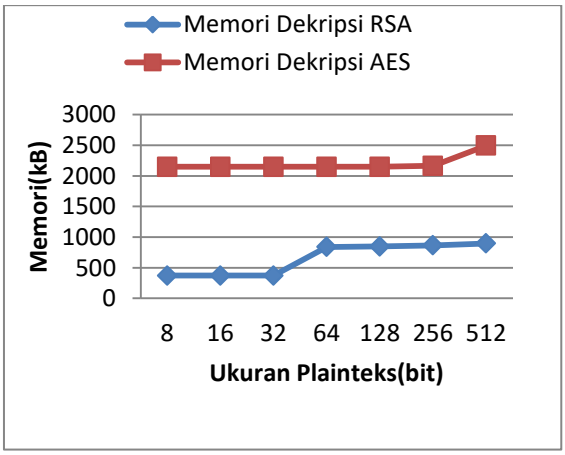
Gambar 4 menunjukkan hasil dari penelitian kecepatan komputasi pada proses dekripsi dengan menggunakan algoritma RSA dan AES, sesuai grafik diatas proses dekripsi pada algoritma AES membutuhkan waktu yang lebih banyak daripada pada algoritma RSA, sehingga untuk dekripsi algoritma RSA lebih cepat, namun hal ini sesuai dengan kunci yang digunakan, pada penelitian ini algoritma RSA menggunakan kunci 16 bit dan algoritma AES menggunkan kunci 128 bit dengan perbedaan yang cukup besar menghasilkan kecepatan yang tidak terlalu jauh beda. Jadi dengan kunci yang lebih kecil algoritma RSA sedikit lebih cepat daripada algoritma AES dengan kunci yang jauh lebih besar.



Gambar 5 Penggunaan memori Proses Enkripsi RSA dan AES

Gambar 5 menunjukkan memori data atau SRAM arduino yang digunakan pada proses enkripsi, terlihat kebutuhan memori lebih besar digunakan pada saat proses enkripsi dengan algoritma AES. Perbedaan alokasi memori pada sebuah program yang dijalankan pada perangkat Arduino tergantung pada program yang dibuat, karena besarnya SRAM yang digunakan tergantung seberapa banyak variabel, *array* ataupun sejenisnya yang digunakan pada sebuah program. Sedangkan untuk penyimpanan program pada perangkat Arduino terdapat memori *flash*, dimana pada

bagian ini akan menyimpan secara permanen program yang kita buat pada perangkat Arduino. Namun pada grafik diatas memori yang dijelaskan adalah SRAM yang digunakan pada proses enkripsi dengan algoritma RSA dan AES



Gambar 6 Penggunaan memori Proses Dekripsi RSA dan AES

IV. Kesimpulan.

1. Pada algoritma RSA dengan kunci 16 bit rata-rata kecepatan untuk proses enkripsi adalah 0,25 detik sedangkan dengan algoritma AES 128 kecepatan rata-rata enkripsinya adalah 0,22 detik, jadi dengan perbedaan kunci yang cukup jauh, kecepatan proses yang di hasilkan hampir sama sehingga pada perangkat IoT algoritma AES lebih baik
2. Pada proses dekripsi rata-rata kecepatan dengan algoritma RSA 0,07 detik lebih cepat daripada algoritma AES dengan kecepatan rata-rata 0,13 detik. Dengan kunci algoritma AES 128 bit dan algoritma RSA 16 bit, selisih kecepatan yang didapatkan sangat kecil.
3. Kebutuhan memori pada algoritma RSA untuk proses enkripsi mencapai 5.2% dan 11.2% pada proses dekripsi dari kapasitas memori SRAM Arduino. Sedangkan pada algoritma AES kebutuhan memori pada proses enkripsi mencapai 16.5% dan 31.3% pada proses dekripsi. Kebutuhan memori pada

Gambar 6 diatas menunjukkan memori data atau SRAM arduino yang digunakan pada proses dekripsi RSA dan AES, terlihat kebutuhan memori lebih besar digunakan pada saat proses dekripsi AES. Perbedaan alokasi memori pada sebuah program yang dijalankan pada perangkat Arduino tergantung pada program yang dibuat, karena besarnya SRAM yang digunakan tergantung seberapa banyak variabel, array ataupun sejenisnya yang digunakan pada sebuah program. Sedangkan untuk penyimpanan program pada perangkat Arduino terdapat memori flash, dimana pada bagian ini akan menyimpan secara permanen program yang kita buat pada perangkat Arduino (Wardana,2015). pada grafik diatas memori yang dijelaskan adalah SRAM.

algoritma AES lebih banyak dari RSA sesuai dengan perbedaan kunci yang digunakan.

DAFTAR PUSTAKA

Ariyus, D. (2008). Pengantar Ilmu Kriptografi. Yogyakarta: Penerbit Andi.

Farooq, M., Waseem, M., Khairi, A., & Mazhar, S. (2015). A Critical Analysis on the Security Concerns of Internet of Things (IoT). *I*nternational Journal of Computer Applications.

Giantara, G. (2016). *OWASP* Internet of Things Top Ten.

Luqman, M., Septama, M. P., Clara, V., Wlw, T., & Hamidan, R. (2018). Analisis Perbandingan Mekanisme Enkripsi Data Pada Teknologi Low Power Wide Area (Lpwa)

Marsel Sampe Asang & Irwan Sembiring. (2017). Keamanan Data Pada Perangkat Internet Of Things Menggunakan Metode Public-Key Cryptography. Teknologi Informasi.

Menezes, a. J., Van Oorschot, P. C., & Vanstone, S. a. (1997). Appendix from Handbook of applied cryptography. Annals of Physics.

Meutia, E. D. (2015). Internet of Things – Keamanan dan Privasi. Seminar Nasional Dan Expo Teknik Elektro 2015.

Ordinary, A. (2016). Pengertian Internet of Things dan Implementasi ioT. Retrieved

Rijmen, V., & Daemen, J. (2001). Advanced Encryption Standard (AES) (FIPS PUB 197)Technology Laboratory, National Institute of Standards

Tampubolon, N. B., Isnanto, R. R., & Sinuraya, E. W. (2015). Implementasi Dan Analisis Algoritma Advanced Encryption Standard (Aes) Pada Tiga Variasi Panjang Kunci Untuk Berkas Multimedia.

Wardana, K (2015). Perbedaan Jenis Memori Pada Arduino. http://Tutorkeren.com

Azam, M (2019). Pengertian Tipe Data Beserta Fungsi dan Jenis-Jenis Tipe Data dalam Pemrogaman. http://www.nesabamedia.com

Yohanes (2019) Padding dalam Kriptografi. Blog.compactbyte.com

Moccworld (2014) Pembangkitan (Ekpansi) Kunci AES 256 bit.lingkarantekno.wordpress.com

Munir,R (2007) Strategi Algoritma. Penerbit Informatika.