IJEIS, Vol.5, No.2, October 2015, pp. 211~220
ISSN: 2088-3714

Purwarupa Perangkat Keras untuk Eksekusi Algoritma AES Berbasis FPGA

Nia Gella Augoestien*[1], Agfianto Eko putra[2]
[1,2]Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta
e-mail: ***[1]nia.gella@mail.ugm.ac.id**, [2]agfi68@gmail.com,

Abstrak

Algoritma kriptografi AES merupakan algoritma yang sering digunakan dalam menjaga kerahasiaan data. Kerahasiaan data merupakan parameter utama pengamanan data di berbagai sistem. Keamanan data dapat dicapai dengan mengkolaborasikan algoritma AES dengan algoritma kriptosistem lainnya. Oleh karena itu,perangkat keras pengeksekusi algoritma AES dengan sumber daya terbatas menjadi sangat penting.

Penelitian ini mengusulkan rancang bangun purwarupaperangkat keras untuk eksekusi algorima AES yang mengutamakan pemakaian sumber daya optimalmenggunakan FPGA tanpa mengorbankan kecepatan eksekusi. Pengoptimalan sumber daya ditempuh dengan merancang perangkat keras untuk enkripsi dengan dekripsi yang saling berbagi sumber daya, menggunakan arsitektur iteratif pada level putaran, arsitektur pipeline pada level transformasi, dan lebar data 32 bit.

Purwarupa perangkat keras pada penelitian ini menggunakan FPGA Xilinx Spartan®-6 Seri (XC6LX16-CS324) hasil pemodelan telah berhasil melakukan proses enkripsi dan dekripsi. Efisiensi perangkat keras yang dicapai adalah 1,94Mbps/Slice, sedangkan lewatan yang diperoleh adalah 308,96Mbps. Dengan pemakaian sumber daya hanya 6% dari yang tersedia pada FPGA.

Kata kunci*—Algoritma AES, FPGA, resource sharing, iteratif, pipeline*

Abstract

AES cryptography algorithm is a tool which often using to protect confidentiality of data.Confidentiality of data is principle parameter of data security in various system. Data security achieve by collaborated AES algorithm with another cryptosystem tools. Therefore, limited resource hardware to excecuteAES algorithm is very important.

This research proposed hardware prototype for excecuting AES algorithm based on FPGA. Optimumresource utilizing become basic priority in this design. So that, we are using resource sharing between hardware for encryption and decryption, iteratif architecture on round level, pipeline architecture on transformation level with 32-bit architecture at design to attain optimum resource utilizing.

Hardware prototype in this research use FPGA Xilinx Spartan®-6 (XC6LX16-CS324), encryption and decryption have been done in this hardware prototype. This prototype have 1,94Mbps/Slice hardware efficiency, 308,96Mbps throughput with only using 6% resource that available on this FPGA.

Keywords— *AES Algorithm, FPGA, resource sharing, iterative, pipeline*

Received December 24th,2014; Revised February 20th, 2015; Accepted October 1th, 2015

1. PENDAHULUAN

Pengamanan data menjadi kebutuhan yang tidak dapat diabaikan pada berbagai sistem. Terdapat beberapa parameter keamanan data, yakni: kerahasian data, integritas data, keaslian sumber data, bebas dari penyangkalan dan keaslian entitas. Kerahasiaan data merupakan parameter layanan utama yang dibutuhkan kebanyakan sistem. Dalam implementasinya, pengamanan data memperhatikan beberapa aspek, seperti; ketepatan sarana/ algoritma yang digunakan, kekuatan proteksi dan biaya implementasi. Pemilihan dan kolaborasi *tools* yang tepat mampu mencapai keamanan data yang efisien untuk sebuah sistem [1]**.**

Algoritma AES merupakan salah satu algoritma kriptografi standar yang masih sering dikolaborasikan dalam beberapa sistem dengan beberapa *tools* kriptografi lainnya[2]. Pilihan rekonfigurasi perangkat keras seperti FPGA sebagai target platform untuk implementasi algoritma kriptografi muncul menjadi solusi praktis untuk sistem *embedded* dan aplikasi kecepatan tinggi[3]. Walaupun demikian terdapat 2 tujuan utama yang saling berkebalikan dalam penggunaan FPGA sebagai *platform* komputasi yaitu; penghematan pemakaian sumber daya dan pemaksimalkan kecepatan pemrosesan, maka suatu keharusan untuk memilih tujuan yang sesuai untuk sasaran yang ingin dicapai [4].

Optimalisasi dari segi pemakaian sumber daya pada *chip* tanpa mengorbankan kecepatan komputasi diperlukan untuk mewujudkan komputasi sebuah sistem pengamanan data yang dapat dieksekusi dalam satu *chip* saja. Pengurangan pemakaian sumber daya pada FPGA sesuai dengan piranti *embedded*/ *portable* yang otomatis dapat mengurangi biaya dan konsumsi daya[5]. Pengurangan pemakaian sumber daya merupakan tantangan utama perancangan akhir-akhir ini dan dapat ditempuh dengan mempersempit lebar data[6]. Pasar piranti *embedded* saat ini cendrung pada mikroprosessor 32 bit karena memiliki kinerja yang baik dalam kaitannya dengan power, area dan biaya jika dibandingkan dengan prosessor 8 atau 16 bit [7]. Agar implementasi yang akan dibuat lebih optimal, maka lebar data disesuaikan dengan lebar data mikroprosessor yang berkembang saat ini.

Pada penelitian ini akan dikembangkan purwarupa perangkat keras untuk eksekusi algoritma AES berbasis FPGA dengan lebar data 32-bit yang kompatibel dengan prosessor *embedded system* yang berkembang saat ini. Implementasi dioptimalkan dari segi pemakaian sumber daya dengan kecepatan yang masih dapat diterima, sehingga sasaran pengamanan data pada lingkungan sumber daya terbatas dapat dicapai. Selain itu, untuk mencapai ke 5 parameter keamanan data, algoritma AES perlu dikolaborasikan dengan sarana-sarana kriptografi lainnya. Penghematan pemakaian sumber daya FPGA menjadi penting agar dapat mengimplementasikan kolaborasi algoritma-algoritma lain dengan algoritma AES pada 1 *chip* FPGA.

2. METODE PENELITIAN

2.1 Analisis Sistem

Penelitian ini bertujuan untuk merancang dan membuat purwarupa perangkat keras untuk mengeksekusi algoritma kriptografi AES berbasis FPGA untuk pengamanan data pada sistem dengan sumber daya terbatas. Berdasarkan tinjauan pustaka yang telah dilakukan, maka dibutuhkan spesifikasi sistem berikut:

- Optimasi pemakaian area.

Optimasi pemakaian area pada FPGA menjadi penting mengingat bahwa purwarupa yang akan dibuat digunakan pada sumber daya terbatas. Selain itu, algoritma AES sendiri sering dikolaborasikan dengan beberapa algoritma lain untuk mencapai keamanan yang efisien pada suatu sistem. Pemakaian sumber daya pada FPGA dengan mengoptimalkan akan memberi peluang untuk beberapa algoritma kriptografilain diimplementasikan pada *chip* FPGA yang sama.

- Arsitektur dengan lebar data 32 bit.

Kecendrungan pasar mikroprosessor untuk piranti embedded mengacu pada lebar data 32-bit saat ini, untuk menjadikan purwarupa perangkat keras yang akan dibuat lebih mudah digunakan maka dirancang arsitektur dengan lebar data 32 bit.

- Kerahasiaan kunci

AES merupakan algoritma kunci simetri dimana kerahasiaan kunci menjadi sangat penting. Disisi lain, penggunaan *embedded system* pada beberapa aplikasi dengan sumber daya terbatas seperti Wireless Sensor Network (WSN), kemungkinan penggantian kunci sangat jarang dilakukan.

Agar arsitektur perangkat keras yang akan dibuat sesuai dengan spesifikasinya, maka dilakukan penyusunan strategi supaya sistem dapat bekerja sesuai dengan spesifikasinya. Spesifikasi optimasi pemakaian area dipenuhi dengan menggunakan strategi berikut:

- Mengabungkan arsitektur perangkat keras untuk proses enkripsi dan dekripsi.
- Menggunakan arsitektur iteratif pada level putaran.
- Menggunakan arsitektur pipeline pada level transformasi.

Arsitektur lebar data 32-bit dicapai dengan membagi 128-bit blok data pada AES menjadi 32-bit (1 kolom pada *state*) yang dieksekusi dengan cara pipeline pada level transformasi. Cara ini dapat mengurangi pemakaian pin Input dan keluaran pada arsitektur perangkat keras, sehingga otomatis dapat mengurangi pemakaian sumber daya pada FPGA. Selain itu, dengan arsitektur pipeline untuk 32-bit paket data pada level transformasi diperkirakan dapat mengurangi latensi eksekusi dan pewaktuan.

Spesifikasi kerahasiaan kunci yang digunakan untuk proses enkripsi dan dekripsi dapat dicapai dengan penggunaan kunci spesifik yang diimplementasikan langsung pada rangkaian transformasi *Add Roundkey*, sehingga tidak dibutuhkan lagi perangkat keras yang didedikasikan khusus untuk perluasan kunci atau penyimpan kunci internal setiap putaran. Hal ini diharapkan dapat menjaga kerahasian kunci dan sekaligus menghemat pemakaian sumber daya pada FPGA.

Arsitektur perangkat keras yang akan dibuat menggabungkan antara perangkat keras untuk eksekusi proses enkripsi dan dekripsi, karena terdapat beberapa transformasi pada kedua proses yang dapat saling berbagi sumber daya yang digunakan. Adapun transformasi-transformasi yang dapat saling berbagi sumber daya tersebut diantaranya adalah transformasi Addroundkey pada proses enkripsi dan dekripsi, transformasi Shitf row dengan tranformasi INV*Shift Row*, dan transformasi *Mix Columns* dengan tranformasi INV*Mix Columns* Pemakaian sumber daya yang saling berbagi tidak dapat dilakukan untuk transformasi *Substitute Byte* dan INV*Substitute Byte*, karena walaupun mekanisme kedua transformasi ini sama, namun menggunakan SBOX dan INV SBOX yang berbeda.

Arsitektur iteratif akan digunakan pada level putaran sesuai dengan spesifikasi sistem. Diharapkan dengan arsitektur iteratif ini diharapkan dapat mengoptimalkan pemakaian sumber daya pada FPGA, karena setiap putaran akan dieksekusi dengan menggunakan sumber daya yang sama. Arsitektur iteratif juga mendukung mode feedback, sehingga penggabungan arsitektur perangkat keras untuk proses enkripsi dan dekripsi tidak akan menjadi masalah. Selain itu, penggunaan arsitektur iteratif juga dapat mengurangi kondisi diam (*idle*) bagian arsitektur perangkat keras yang akan dibuat pada saat eksekusi.

Cara lain untuk mengurangi kondisi *idle* bagian arsitektur perangkat keras tertentu adalah dengan menggunakan arsitektur pipeline pada level putaran. Arsitektur pipeline digunakan agar beberapa tahapan eksekusi untuk data yang berbeda dapat dijalankan pada waktu yang bersamaan. Tahapan eksekusi yang dimaksud disini adalah transformasi-transformasi yang terlibat dalam setiap putaran. Penggunaan arsitektur ini diharapkan mampu mengoptimalkan pemakaian sumber daya FPGA tanpa harus mengorbankan kecepatan eksekusi.

Berdasarkan spesifikasi juga akan digunakan lebar data 32 bit pada arsitektur ini. Ada beberapa hal yang perlu diperhatikan ketika menggunakan perangkat keras dengan lebar data 32 bit untuk mengeksekusi algoritma AES, karena pada dasarnya algoritma AES memiliki ukuran blok 128 bit. Adapun hal-hal yang dimaksud adalah sebagai berikut.

- Menurut [8] mengerjakan transformasi *Shift Row* sebelum transformasi *Substitute Byte* tidak memiliki efek pada algoritma enkripsi AES, begitu juga dengan dekripsi.

- Transformasi shiftrow dan INV*Shift Row* dapat dilakukan setelah tersedia 32 bit data dalam 1 baris *state*.
- Transformasi *Mix Columns* dan INVMix Column dapat dilakukan setelah 32 bit data dalam 1 kolom *state* telah tersedia.

2.2 *Perancangan Sistem*

Berdasarkan analisis sistem, 128 bit blok data pada algoritma AES akan dibagi menjadi 4 paket data 32 bit dalam setiap tahapan eksekusi. Adapun contoh pembagiannya dilustrasikan pada Gambar 1.

S0,0	S1,0	S2,0	S3,0	S0,1	S1,1	S2,1	S3,1	S0,2	S1,2	S2,2	S3,2	S0,3	S1,3	S2,3	S3,3
32	43	f6	a8	88	5a	30	8d	31	31	98	a3	e0	37	07	34

(a)

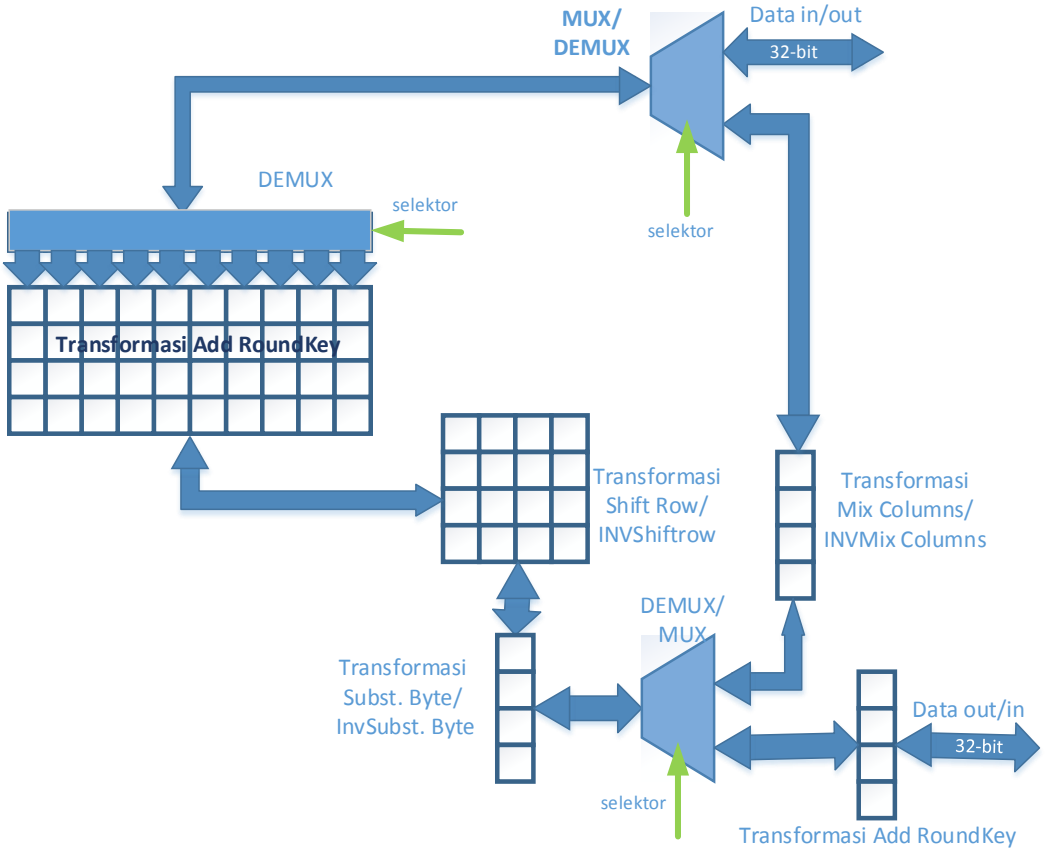
Paket 0	Paket 1	Paket 2	Paket 3
S0,0	S0,1	S0,2	S0,3
S1,0	S1,1	S1,2	S1,3
S2,0	S2,1	S2,2	S3,2
S3,0	S3,1	S3,2	S3,3

(b)

Paket 0	Paket 1	Paket 2	Paket 3
32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

(c)

Gambar 1 Ilustrasi pembagian 128 bit blok data AES

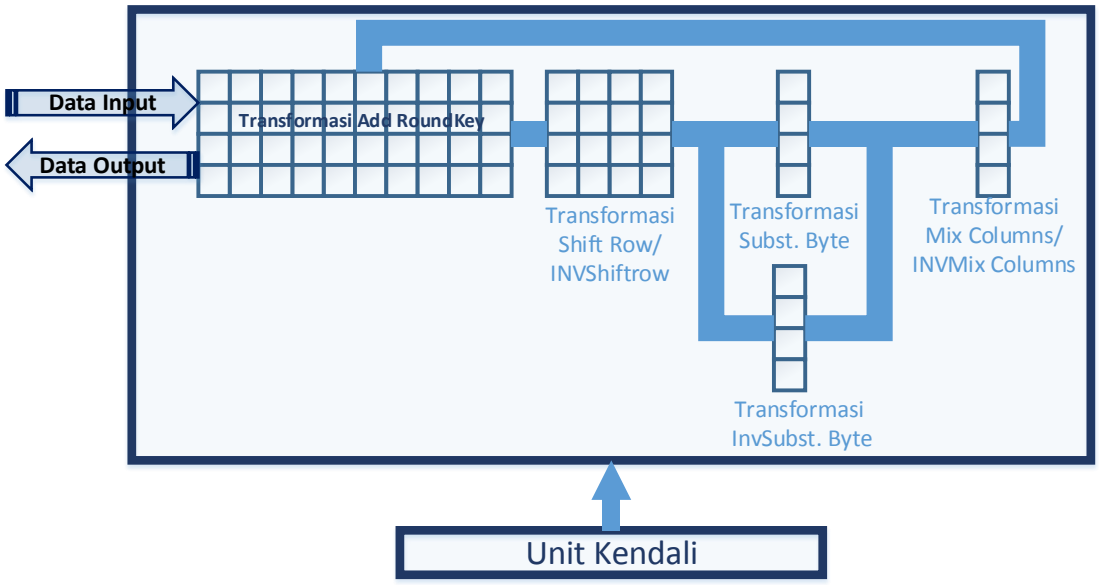


Gambar 2 Aliran data arsitektur perangkat keras

Sesuai dengan aliran enkripsi dan dekripsi pada algoritma AES pada diketahui bahwa

- Tranformasi *Add Roundkey* adalah transformasi pertama yang dilewati data baik untuk proses enkripsi maupun dekripsi.
- Pada putaran 0 aliran enkripsi hanya terdapat transformasi *Add Roundkey*, sedangkan pada putaran 1-9 terdapat 4 transformasi, dan pada putaran terakhir hanya tidak dilaksanakan transformasi mix columns.
- Pada aliran dekripsi putaran 0 tidak dikerjakan tansformasi INV*Mix Columns* pada putaran 1-9 terdapat 4 transformasi dan pada putaran terakhir hanya dikerjakan transformasi *Add Roundkey*.

Purwarupa perangkat keras yang akan dibuat memiliki aliran data seperti pada Gambar 2. Hal ini berdasarkan pada pertimbangan-pertimbangan yang telah dijabarkan diatas. Berdasarkan aliran data pada Gambar 2, maka dapat dibuat Arsitektur level teratas dari perangkat keras seperti yang ditunjukan Gambar 3.



Gambar 3 Arsitektur level teratas perangkat keras

Sesuai dengan Gambar 3, diketahui bahwa arsitektur hadware dapat dibagi menjadi 4 unit berdasarkan transformasinya. Namun, karena transformasi *Substitute Byte* dan INV*Substitute Byte* tidak bisa dirancang untuk saling berbagi sumber daya, maka pada penelitian ini purwarupa perangkat keras yang akan dibuat terdiri dari 5 unit, yaitu; unit tranformasi *Add Roundkey*, unit transformasi *Shift Row*, unit transformasi substitute byet, unit transfromasi INV*Substitute Byte* dan unit transformasi *Mix Columns*.

Proses integrasi dibutuhkan untuk menggabungkan ke 5 unit diatas. Penelitian ini menggunakan arsitektur pipeline pada level transformasi dan iteratif pada level putaran, maka diperlukan perancangan eksekusi setiap siklus detak untuk menjamin semua unit-unit fungsional diatas mampu bekerja sesuai dengan standar proses enkripsi ataupun dekripsi algoritma kriptografi AES. Gambar 4 memperlihatkan perancangan eksekusi setiap siklus detak algoritma AES pada putaran 0 - 2. Pada siklus 0 dilakukan transformasi *Add Roundkey* untuk paket data 0. Saat siklus 1 dilakukan transformasi *Shift Row* untuk paket data 0 dan transformasi *Add Roundkey* untuk paket data 1. Transformasi *Shift Row* dan *Add Roundkey* juga berjalan paralel untuk paket data berbeda pada siklus 2 dan 3. Setelah hasil transformasi *Shift Row* di dapat untuk paket data 0, pada siklus 6 dilakukan transformasi *Substitute Byte* terhadap paket data tersebut. Tranformasi *Substitute Byte* dan *Mix Columns* berjalan paralel untuk siklus 7, sedangkan di siklus 8 dan 9 terdapat 3 dan 4 transformasi berjalan paralel untuk paket data yang

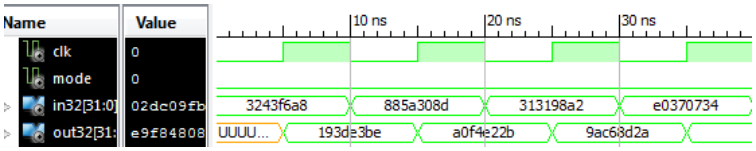
berbeda. Dibutuhkan 82 siklus untuk menyelesaikan eksekusi algoritma AES dengan perancangan eksekusi ini.

siklus	Add RoundKey	Shift Row	Substitusi Byte	Mix Column	
0	paket 0				round 0
1	paket 1	paket 0			
2	paket 2	paket 1			
3	paket 3	paket 2			r
4		paket 3			o
5					u
6			paket 0		n
7			paket 1	paket 0	d
8	paket 0		paket 2	paket 1	
9	paket 1	paket 0	paket 3	paket 2	1
10	paket 2	paket 1		paket 3	
11	paket 3	paket 2			r
12		paket 3			o
13					u
14			paket 0		n
15			paket 1	paket 0	d
16	paket 0		paket 2	paket 1	
17	paket 1	paket 0	paket 3	paket 2	2
18	paket 2	paket 1		paket 3	
19	paket 3	paket 2			r

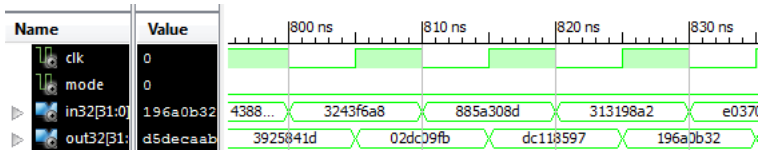
Gambar 4 Perancangan eksekusi setiap siklus detak

3. HASIL DAN PEMBAHASAN

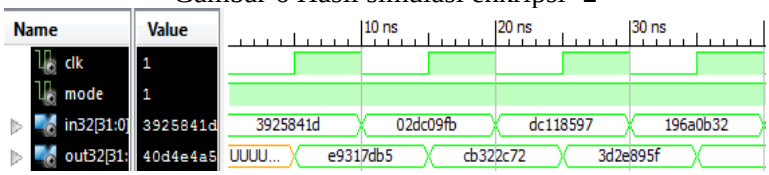
Pengujian dilakukan dalam beberapa tahap, yaitu verifikasi fungsional setiap unit perangkat keras, verifikasi keseluruhan perangkat keras dan pengujian setelah implementasi fisik. Setelah pengujian dinyatajan berhasil, maka dilakukan pengukuran kinerja perangkat keras yang didapat untuk dibandingkan dengan perangkat keras yang dihasilkan pada penelitian lain. Verifikasi fungsional unit dan verifikasi fisik dilakukan menggunakan menggunakan simulator yang disediakan tools perancangan Xilinx ISE Design Suite 14.5. Gambar 5 dan Gambar 6 menunjukkan hasil verifikasi fungsional keseluruhan sistem pada simulator untuk operasi enkripsi, sedangkan Gambar 7 dan Gambar 8 menunjukkan hasil simulasi untuk operasi dekripsi.



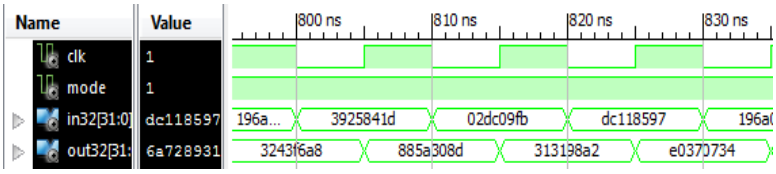
Gambar 5 Hasil simulasi enkripsi-1



Gambar 6 Hasil simulasi enkripsi -2

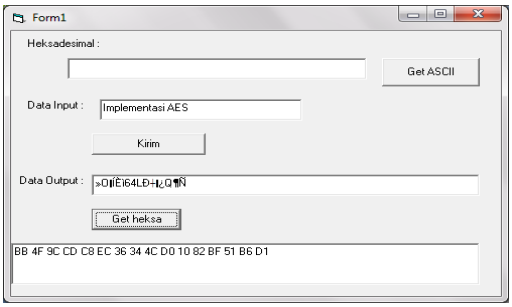


Gambar 7 Hasil simulasi dekripsi -1

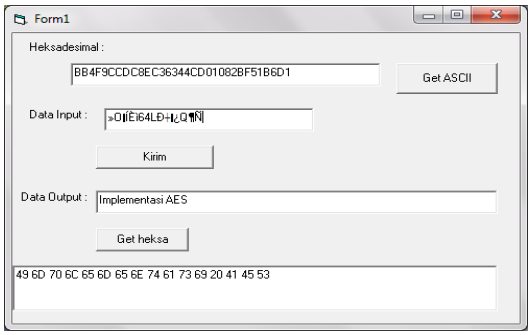


Gambar 8 Hasil simulasi dekripsi -2

Verifikasi fisik dilakukan dengan menggunakan tambahan modul UART untuk memasukkan data ke unit AES pada FPGA dan mengetahui keluaran hasil operasi yang dilakukan Unit AES**.** Gambar 9 dan Gambar 10 juga menunjukkan bahwa hasil pengujian terhadap perangkat keras telah berhasil. Dimana data masukkan "Implementasi AES" yang dienkripsi pada Gambar 9 menghasilkan cipher teks "»OœÍÈì64LÐ,¿Q¶Ñ". Dan ketika cipher teks tersebut didekripsi kembali pada Gambar 10 menghasilkan ouput data "Implementasi AES".



Gambar 9 Contoh hasil enkripsi data



Gambar 10 Contoh hasil dekripsi data

Setelah diketahui bahwa perangkat keras unit AES mampu bekerja dengan baik, maka selanjutnya dilakukan pengukuran kinerja perangkat keras. Tabel 1 Memperlihatkan kinerja perangkat keras yang didapat pada penelitian ini pada 4 tipe FPGA yang dimaksud.

Tabel 1 Kinerja perangkat keras pada beberapa tipe FPGA

Jenis FPGA	**Spartan®-6**	**Spartan®-3**	**Virtex ®-4**	**Virtex®-5**
Timing (ns)	4,992	8,578	4,133	3,315
Frek. Mak. (MHz)	200,341	116.579	241,952	301,618
Latensi (Siklus)	83	83	83	83
Lewatan (Mbps)	308,96	179,784	373,131	465,146
Slice	159	502	475	176
Efisiensi(Mbps/Slice)	1,94	0,36	0,79	2,64

Tabel 1 memperlihatkan bahwa hasil pemodelan dan implementasi perangkat keras paling efisien terdapat pada FPGA keluarga Xilinx Virtex®-5 Seri (xc5vfx70t-2ff1136). Hal ini bisa disebabkan FPGA ini memiliki Slice yang sesuai dengan pemodelan yang dilakukan pada penelitian ini dibandingkan Slice pada FPGA tipe lain atau teknologi yang terdapat pada FPGA

tipe ini lebih canggih. Jika dilihat dari parameter kinerja keseluruhan, hal ini cendrung disebabkan oleh teknologi virtex 5 yang lebih canggih.Dapat dilihat dari frekuensi maksimal pada FPGA virtex jauh lebih tinggi dibandingkan pada FPGA tipe lain yang mengakibatkan lewatan yang tinggi. Namun demikian slice yang digunakan pada FPGA virtex lebih banyak dibandingkan pada FPGA Spartan 6. Karena pemodelan pada penelitian ini memang ditujukan untuk FPGA Xilinx Spartan®-6 Seri (XC6LX16-CS324), dimana efisiensinya mencapai 1,94Mbps/Slice.

Tabel 2 menampilkan data-data kinerja perangkat keras AES pada FPGA yang dilakukan pada penelitian sebelumnya. Data-data yang dimuat pada Tabel 2 hanya untuk penelitian yang mencantumkan parameter-parameter kinerja perangkat keras dengan lengkap. Berdasarkan tabel untuk implementasi menggunakan FPGA tipe Spartan 3 efisiensi yang paling optimal berhasil dilakukan oleh [9] sebesar 0,395. Jika dibandingkan dengan efisiensi pada penelitian ini, efisiensi perangkat keras hampir sama yaitu 0,36. Jika dilihat dari pemodelan yang dilakukan pada kedua perangkat keras ini, perangkat keras ini sama-sama mengimplementasikan pipeline walaupun pada level yang berbeda.

Tabel 2 Perbandingan kinerja perangkat keras dengan penelitian lain

Jenis FPGA	Peneliti	Mode	Lebar Data	Organisasi level putaran	Organisasi level transformasi	Latensi	Timing (ns)	Lewatan (Mbps)	Jumlah Slice	Efisiensi (Mbps/Slice)
Spar-tan 3	[9]	Enk & Dek	128	Iteratif Kombinasional Pipeline Hybrid	Paralel	23 1 12 97	12,8 85,0 11,1 8,9	415 1428 11015 142	6590 36450 27890 1934	0,063 0,039 0,395 0,073
	[6]	Enkripsi	8	Iteratif	Paralel	160	21,92	36.5	184	*0,198*
	Peneliti	**Enk & Dek**	**32**	**Iteratif**	**Pipeline**	**83**	**8,578**	**179,784**	**502**	***0,36***
Spar-tan 6	[6]	Enkripsi	8	Iteratif	Paralel	160	13,76	58,13	80	*0.727*
	Peneliti	**Enk & Dek**	**32**	**Iteratif**	**Pipeline**	**83**	**4,922**	**308,96**	**159**	***1,94***
Virt-ex 4	[5]	Enkripsi	128	Iteratif	Sekuensial	44	6	484	298	*1,624*
	[2]	Enkripsi Dekripsi	128	Non Pipeline	CTM- Enkripsi CTM-Dekripsi	11 11	5,2 6,4	2249 1829	4582 5554	*0,49* *0,33*
	Peneliti	**Enk & Dek**	**32**	**Iteratif**	**Pipeline**	**83**	**4,133**	**373,131**	**475**	***0,79***
Virt-ex 5	[7]	Enkripsi	32	Iteratif	Paralel	15	3,3	2588,706	375	*6,90*
	[10]	Enkripsi	128	Iteratif	Sekuensial	44	3,85	755,99	333	*2,27*
	Peneliti	**Enk & Dek**	**32**	**Iteratif**	**Pipeline**	**83**	**3,315**	**465,16**	**176**	***2,64***

Tabel 2 menampilkan data-data kinerja perangkat keras AES pada FPGA yang dilakukan pada penelitian sebelumnya. Data-data yang dimuat pada Tabel 2 hanya untuk penelitian yang mencantumkan parameter-parameter kinerja perangkat keras dengan lengkap. Berdasarkan tabel untuk implementasi menggunakan FPGA tipe Spartan 3 efisiensi yang paling optimal berhasil dilakukan oleh [9] sebesar 0,395. Jika dibandingkan dengan efisiensi pada penelitian ini, efisiensi perangkat keras hampir sama yaitu 0,36. Jika dilihat dari pemodelan yang dilakukan pada kedua perangkat keras ini, perangkat keras ini sama-sama mengimplementasikan pipeline walaupun pada level yang berbeda.

Implementasi pada FPGA tipe Spartan 6 pada penelitian [6] efisiensi pada penelitian ini jauh lebih besar, yaitu 1,94 : 0.727. Lebar data 8-bit yang digunakan pada penelitian [6] menyebabkan latensi yang besar, sehingga berpengaruh terhadap lewatan dari perangkat keras. Pada penelitian ini memiliki latensi ½ dari latensi pada penelitian [6] yang didapat dengan menggunakan lebar data 32-bit. Walaupun lebar data yang digunakan pada penelitian ini 4 kali lebih besar dari penelitian [6], namun slice yang digunakan pada penelitian ini hanya 2 kali slice yang digunakan pada penelitian [6] dan mampu melakukan operasi enkripsi dan dekripsi.

Penggunaan lebar data 32 bit juga lebih efisien dibandingkan dengan lebar data 8 bit menggunakan FPGA Spartan 3 seperti yang ditunjukkan Tabel 2.Efisiensi perangkat keras

dengan lebar data 8 bit adalah 0,198 dilihat pada Tabel 2, sedangkan efisiensi dengan lebar data 32 bit adalah 0,36. Berdasarkan data ini dapat disimpulkan bahwa lebar data 32 bit lebih efisien dibandingkan lebar data 8 bit.

Implementasi menggunakan perangkat keras Virtex 4 pada penelitian [5] memiliki efisiensi 1,624, sedangkan pada penelitian ini hanya memiliki efisiensi 0,79. Selain disebabkan pada penelitian [5] hanya dirancang perangkat keras untuk enkripsi, sedangkan pada penelitian ini perangkat keras yang dirancang dapat digunakan baik untuk enkripsi maupun dekripsi, arsitektur slice pada Virtex 4 tidak sesuai dengan pemodelan pada penelitian ini. Disisi lain, [5] menyesuaikan pemodelan yang dilakukkannya dengan struktur dari CLB/ Slice yang terdapat pada FPGA Virtex 4, sehingga salah satu langkah yang dapat ditempuh untuk memaksimalkan efisiensi adalah dengan menyesuaikan pemodelan dengan struktur CLB/Slice pada FPGA yang digunakan.

Langkah lain untuk memaksimalkan efisiensi kinerja perangkat keras untuk eksekusi algoritma AES dapat dilakukan dengan menggabungkan arsitektur perangkat keras untuk enkripsi dan dekripsi yang dikenal dengan *resource sharing*. Hal ini ditunjukkan pada Tabel 2, dimana penelitian [2] menggunakan arsitektur yang berbeda untuk kedua operasi tersebut. Efisiensi perangkat keras untuk operasi enkripsi adalah 0,49 dan untuk operasi dekripsi adalah 0,33. Penelitian ini menggabungkan arsitektur untuk kedua operasi dan memiliki efisiensi 0,79.

Pada implementasi menggunakan Virtex 5, penelitian [7] memiliki efiensi lebih tinggi yaitu 6,9, sedangkan penelitian ini hanya memiliki efisiensi 2,64. Hal ini dikarenakan penelitian [7] hanya mendukung operasi enkripsi saja karena menggabungkan ke 4 transformasi menjadi 1 sehingga dapat diselesaikan dalam 1 siklus. Selain itu penelitian [7] memiliki latensi rendah sehingga mampu memiliki lewatan yang sangat besar. Berbeda dengan penelitian ini dengan latensi yang 83. Jika dibandingkan dengan efisiensi pada penelitian [10], efisiensi penelitian ini lebih tinggi seperti yang ditunjukkan pada Tabel 2, walaupun memiliki lebar data-128 bit dan hanya mendukung operasi enkripsi saja.

Dilihat dari variasi organisasi level transformasi yang terdapat pada Tabel 2, terdapat 4 jenis organisasi yang ada, yaitu; paralel, pipeline, sekuensial dan CTM. Berdasarkan Tabel 2, diketahui bahwa organisasi pipeline pada penelitian ini memiliki pewaktuan yang lebih singkat dibandingkan dengan beberapa penelitian lainnya pada berbagai jenis FPGA yang digunakan, kecuali untuk penelitian [7]. Penyebab anomali ini tidak dapat dianalisis, karena pada paper [7] tidak memaparkan secara jelas bagaimana organisasi paralel pada level transformasi memiliki pewaktuan yang singkat.

Alasan yang menyebabkan organisasi pipeline pada umumnya memiliki efisiensi yang lebih baik pada level transformasi dikarenakan setiap transformasi diperlakukan sebagai tahapan yang berbeda sehingga mengurangi jalur kritis dari implementasi yang dilakukan. Pengurangan jalur kritis pada implementasi menyebabkan perambatan delay (penundaan) antara satu siklus ke siklus yang lain tidak terlalu panjang, sehingga dapat mengurangi pewaktuan yang secara langsung meningkatkan lewatan yang diperoleh.

4. KESIMPULAN

Dari penelitian yang telah dilakukan, diperoleh kesimpulan sebagai berikut:

1. Implementasi rancangan pada FPGA Xilinx Spartan®-6 Seri (XC6LX16-CS324) memanfaat sumber daya 6% dari sumber daya yang tersedia dengan rincian 159 Slice dan 70 Pin I/O.
2. Penghematan sumber daya dicapai dengan menggabungkan arsitektur yang saling berbagi sumber daya (*resource sharing)* dan pemodelan yang menyesuaikan dengan struktur FPGA yang digunakan
3. Perangkat keras yang didapat memiliki kinerja
    - Latensi : 83
    - Pewaktuan : 4,992 ns

- Lewatan : 308,96 Mbps
- Efisiensi : 1,94 Mbps/Slice

4. Organisasi pipeline pada level transformasi pada umumnya mampu mengurangi pewaktuan dibandingkan sekuensial dan paralel namun tidak optimal dalam mengurangi latensi jika menggunakan lebar data 32 bit

5. SARAN

Saran-saran yang diperlukan untuk perbaikan penelitian ini adalah:

1. Sumber daya yang belum digunakan pada *chip* FPGA dapat dimanfaatkan untuk mengimplementasikan sarana kriptosistem lainnya untuk mencapai keamanan data dalam 1 chip.
2. Diperlukan teknik baru untuk perancangan transformasi *shift row* agar dapat meminimalkan latensi yang dibutuhkan untuk operasi enkripsi dan dekripsi.
3. Pemetaan dan pemilihan lintasan sumber daya yang digunakan pada FPGA sebaiknya dilakukan secara manual tidak otomatis menggunakan Xilinx ISE Design Suite, untuk meminimalkan pemakaian sumber daya

DAFTAR PUSTAKA

[1]Martin, K., 2012, *Everyday Cryptography Fundamental Principles and Applications*, Oxford University Press, London.

[2]Jamal, H. dan Hussain, U., 2012, An Efficient High Throughput FPGA Implementation of AES for Multi-Gigabit Protocols, *International Conference on Frontiers of Information Technology*, 12, pp. 215-218.

[3]Henriquez, F. R., Saqib, N. A., Perez, A. D. dan Koc, C. K., 2006, *Cryptographic Algorithms on Reconfigurable Hardware*, Springer, New York.

[4]Deshpande, A. M, Deshpande, M. S, dan Kayatanavar, D. N., 2009, FPGA Implementation of AES Encryption and Decryption, *International Conference on "Control, Automation Communication and Energy Conservation"*, 6, pp 1-6.

[5]Ghaznazi, S., Gebotys, C., dan Elbaz., R., 2009, Efficient Technique for the FPGA Implementation of the AES MixColumns Transformation, *International Conference on Reconfigurable Computing and FPGAs*, 52, pp. 219-224.

[6] Benaissa, M., dan Chu, J., 2012, Low area memory-free FPGA implementation of AES algorithm, pp. 623-626.

[7] Benhadjyoussef, N., Machhout, M., Elhadjyoussef, W., dan Tourki, R., 2012, A compact 32-Bit AES design for embedded system, *International Conference on Design & Technology of Integrated Systems in Nanoscale Era*, pp 1-4

[8] Kshirsagar, R. V., dan Vyawahare, M. V., 2012, FPGA Implementation of High speed VLSI Architectures for AES Algorithm, *International Conference on Emerging Trends in Engineering and Technology*, 5, pp. 239-242.

[9]Krukowski, L dan Sugier, J., 2008, Organization of AES Cryptographic Unit for Low Cost FPGA Implementation, Third International Conference on Dependability of Computer Systems DepCoS-RELCOMEX 2008, 36,pp 347-354.

[10]Matsuoka, S. dan Ichikawa, S., 2012, Reduction of power consumption in key-specific AES circuits, *Third International Conference on Networking and Computing*, 3, pp. 323-325.