JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

INSTITUT TEKNLOGI SEPULUH NOPEMBER

USULAN TUGAS AKHIR

1. IDENTITAS PENGUSUL

NAMA	**:**	**Ramadhan Satya Putra**
NRP	**:**	**5110 100 005**
DOSEN WALI	**:**	**Prof.Ir. Supeno Djanali, M.Sc, Ph.D**
DOSEN PEMBIMBING	**:**	**1. Waskitho Wibisono, S.Kom., M.Eng., PhD**
		2. Baskoro Adi Pratomo, S.Kom., M.Kom

2. JUDUL TUGAS AKHIR

"**Paralelisasi Algoritma Pencocokan *String* Knuth-Morris-Pratt dengan NVIDIA CUDA pada *Aplikasi Network Intrusion Detection System***"

3. LATAR BELAKANG

Keberadaan teknologi informasi yang terus berkembang dengan pesat menjadikan kebutuhan akan penggunaannya semakin hari semakin meningkat. Transaksi data melalui internet telah menjadi kebutuhan wajib hampir dari semua perangkat lunak yang ada saat ini. Perangkat lunak seperti media sosial, *cloud server*, *online game*, aplikasi layanan pemerintah, aplikasi pengontrol suatu tempat secara *remote*, dsb. Tentu dengan berbagai macam penggunaan internet tersebut dibutuhkan metode untuk mengamankan jaringannya. Sistem pendeteksi serangan atau yang pada umumya disebut IDS (*Intrusion Detection System*) merupakan senjata utama untuk mengamankan suatu jaringan di mana sistem ini nantinya bertugas untuk mengidentifikasi dan mencatat apakah suatu paket merupakan bentuk serangan atau paket biasa [1].

Cara kerja IDS sendiri pada umumnya adalah *sniffing*, pencocokan paket, dan *logging*. Pencocokkannya juga tidak pandang paket, semua paket yang lalu-lalang pada

jaringan yang dilayani akan diperiksa apakah paket tersebut berupa paket normal atau paket yang mengandung elemen sesuai dengan elemen *rule* yang ditetapkan. Berkembangnya *traffic* internet dewasa ini sudah mencapai angka puluhan Gbps (*Gigabit per second)* dan bukan tidak mungkin akan mencapai ratusan atau bahkan ribuan Gbps nantinya [2].

Kemampuan IDS pada umumnya bergantung pada kemampuan CPU*(Central Processing Unit)*. Pada umumnya CPU bukan hanya untuk menjalankan IDS saja namun untuk mengatur jalannya seluruh sistem komputer sehingga tidak jarang suatu CPU kewalahan untuk mengatur IDS dikarenakan banyaknya jumlah data atau paket yang harus diproses. Kekurangan CPU dalam menjalankan fungsi IDS ini dapat disiasati dengan bantuan GPU (*Graphic Processing Unit*).

GPU (*Graphic Processing Unit*) adalah inti dari kartu grafis. Kartu grafis pada umumnya dirancang untuk keperluan *graphic rendering* seperti pada aplikasi editor gambar, video dan game. Seiring berjalannya waktu produsen kartu grafis ternama seperti NVIDIA dan ATI mengembangkan kemampuan inti dari kartu grafis untuk melakukan komputasi selain komputasi grafis. NVIDIA mengembangkan CUDA dan ATI mengembangkan OpenCL, keduanya mempunyai fungsi sama yaitu melakukan komputasi secara umum layaknya sebuah CPU [3]. Kelebihan dari GPU ini antara lain memiliki lebih banyak inti daripada sebuah CPU. Inti yang banyak dari GPU ini dapat dimanfaatkan untuk membantu komputasi dari CPU.

Deteksi serangan pada suatu jaringan yang memiliki *data traffic* besar sudah pasti membutuhkan kemampuan suatu sistem untuk mencocokan banyaknya paket yang ada dengan *rule* yang sudah tersimpan pada aplikasi. Dengan memanfaatkan bantuan GPU proses pencocokan *rule* tersebut dapat dilakukan dengan mudah dan relatif lebih cepat daripada memanfaatkan CPU saja. Akan tetapi proses pendeteksian serangan ini tidak dapat berjalan serta merta pada GPU saja namun tetap memerlukan CPU karena secara utilitas jenis komputasi yang didukung oleh CPU lebih banyak daripada GPU. Seperti contoh proses penangkapan paket dan pembuatan log harus dilakukan pada CPU. Kekurangan lain dari GPU adalah bertambahnya beban listrik yang dibutuhkan.

Tugas Akhir ini bertujuan untuk memanfaatkan kemampuan CPU dan GPU secara optimal pada studi kasus IDS, optimal yang dimaksud adalah dapat mempercepat proses komputasi dengan memanfaatkan GPU. Dengan adanya aplikasi pendeteksi serangan ini, diharapkan kedepannya akan membantu mengamankan suatu jaringan yang memiliki *data traffic* besar tanpa memerlukan sebuah komputer yang memiliki spesifikasi sangat tinggi, cukup hanya dibekali dengan GPU, yang mendukung NVIDIA CUDA, sebagai tambahan untuk memproses pencocokan *string*.

4. RUMUSAN MASALAH

Berikut beberapa hal yang menjadi rumusan masalah dalam tugas akhir ini:

a. Bagaimana menangkap semua paket yang lalu-lalang pada jaringan?
b. Bagaimana menyimpan *rule* untuk dicocokkan dengan paket–paket yang ditangkap?
c. Bagaimana mencocokkan *rule* yang sudah disimpan dengan paket–paket yang ditangkap?
d. Bagaimana mengambil kesimpulan akhir untuk menentukan apakah sebuah paket merupakan paket biasa atau paket berisi serangan?
e. Bagaimana mencatat sebuah log berdasarkan hasil komputasi yang telah dilakukan?

5. BATASAN MASALAH

Dari permasalahan yang telah diuraikan di atas, terdapat beberapa batasan masalah pada tugas akhir ini, yaitu:

a. Jenis protokol paket yang akan diperiksa adalah TCP dan UDP.
b. *Rule* dicatat pada sebuah file teks.
c. GPU yang digunakan adalah merk NVIDIA, sehingga hanya mendukung NVIDIA CUDA.
d. Pencocokan paket dengan *rule* dilakukan dengan algoritma KMP yang terparalelisasi.

6. TUJUAN PEMBUATAN TUGAS AKHIR

Tugas akhir dibuat dengan beberapa tujuan. Berikut beberapa tujuan dari pembuatan tugas akhir:

a. Mampu mendeteksi apakah suatu paket merupakan serangan atau paket biasa.
b. Mampu mempercepat proses komputasi dengan beban kerja yang berat menggunakan bantuan GPU.
c. Mampu Mengurangi beban kerja CPU terhadap IDS.

7. MANFAAT TUGAS AKHIR

Dengan dibangunnya aplikasi ini, diharapkan mampu mempermudah pengawasan keamanan jaringan internet dengan *data traffic* yang besar dengan hanya sebuah komputer yang dibekali GPU.

8. TINJAUAN PUSTAKA

1) Algoritma KMP

Pencocokan *string* Knuth Morris Pratt atau yang lebih terkenal disebut Algoritma KMP. Algoritma ini diajukan oleh Knuth, Morris, dan Pratt untuk problem pencocokan *string*. Algoritma ini berfungsi untuk mencari kemunculan sebuah kata "w" pada *string* "s" dengan mengamati bahwa ketika terjadi ketidakcocokan, kata itu sendiri menjadi informasi yang cukup untuk menentukan di mana pertandingan berikutnya bisa dimulai, sehingga melewati pemeriksaan ulang karakter cocok sebelumnya. Algoritma ini dipublikasikan pada tahun 1977 [4].

2) IDS

IDS (*Intrusion Detection System*) adalah aplikasi perangkat lunak yang digunakan untuk menyiapkan dan menghadapi sebuah serangan. Hal ini dijalankan dengan cara mengoleksi informasi dari berbagai sumber baik dari suatu sistem maupun jaringan, lalu menganalisanya untuk menentukan ada tidaknya ancaman.

IDS dikategorikan menjadi 3, yaitu:

a. NIDS (*Network Intrusion Detection System*), menjalankan analisa terhadap *traffic* dalam sebuah *subnet*. Bekerja secara acak dan mencocokkannya dengan kumpulan pola serangan yang sudah disimpan pada *library*. Ketika NIDS berhasil menedektesi serangan atau perilaku yang abnormal, peringatan akan dikirim ke *administrator* jaringan.

b. NNIDS (*Network Node Intrusion Detection System*), hampir sama dengan NIDS hanya saja NNIDS hanya bekerja pada satu *host* saja tidak untuk satu *subnet.* Contoh penggunaan NNIDS adalah pada VPN, yaitu untuk memeriksa *traffic* ketika sudah terdekripsi. Dengan cara ini dapat diketahui apakah seseorang sedang mencoba merusak sebuah VPN.

c. HIDS (*Host-based Intrusion Detection System*), mengambil *snapshot* dari sistem yang dimiliki dan mencocokkannya dengan *snapshot* yang sudah diambil sebelumnya. Bila sebuah *system files* penting telah termodifikasi atau terhapus, sebuah peringatan akan dikirimkan ke *administrator*. Contoh penggunaanya adalah pada sebuah mesin yang bersifat *mission critical*, sehingga tidak boleh ada perubahan terhadap konfigurasinya [1].

3) GPU

GPU (*Graphic Processing Unit*), pertama kali dikenalkan pada 31 Agustus 1999, merupakan satu *chip processor* yang terintegrasi dengan *transform, lighting, triangle setup/ clipping,* dan *rendering engines* yang memiliki kemampuan untuk memproses minimal 10 juta poligon setiap detiknya.

Dewasa ini teknologi 3D semakin memiliki peran di kehidupan sehingga kebutuhan akan proses komputasi yang lebih cepat semakin meningkat. Berkembangnya GPU juga memungkinkan pemindahan kalkulasi transformasi

dan pencahayaan dari CPU ke GPU sehingga pemrosesan gambar menjadi lebih cepat [5].

4) CUDA

CUDA adalah platform komputasi paralel dan model pemrograman yang memungkinkan peningkatan dramatis pada performa komputasi dengan cara memanfaatkan GPU sebagai sumber daya.

Sejak dikenalkan pada 2006, CUDA sudah dikembangkan secara luas melalui ribuan aplikasi dan *paper* yang terpublikasi. CUDA sudah terpasang pada kurang lebih 300 juta mesin meliputi *notebook, workstation, compute cluster,* dan *supercomputer.* CUDA juga menjadi solusi untuk mengakselerasi aplikasi dalam bidang astronomi, biologi, kimia, fisika, keuangan, dan masih banyak contoh lainnya.

Developer perangkat lunak, saintis, dan peneliti dapat menggunakan bantuan untuk akselerasi via GPU untuk aplikasi buatannya dengan menggunakan tiga jenis pendekatan:

a. Meletakkan *GPU-accelerate library* untuk mengganti atau menambahkan *CPU-only library* seperti MKL, BLAS, dan *library* yang sering digunakan lainnya.

b. Secara otomatis memparalel *loop* pada Fortran atau C menggunakan petunjuk OpenACC untuk mengakselerasi.

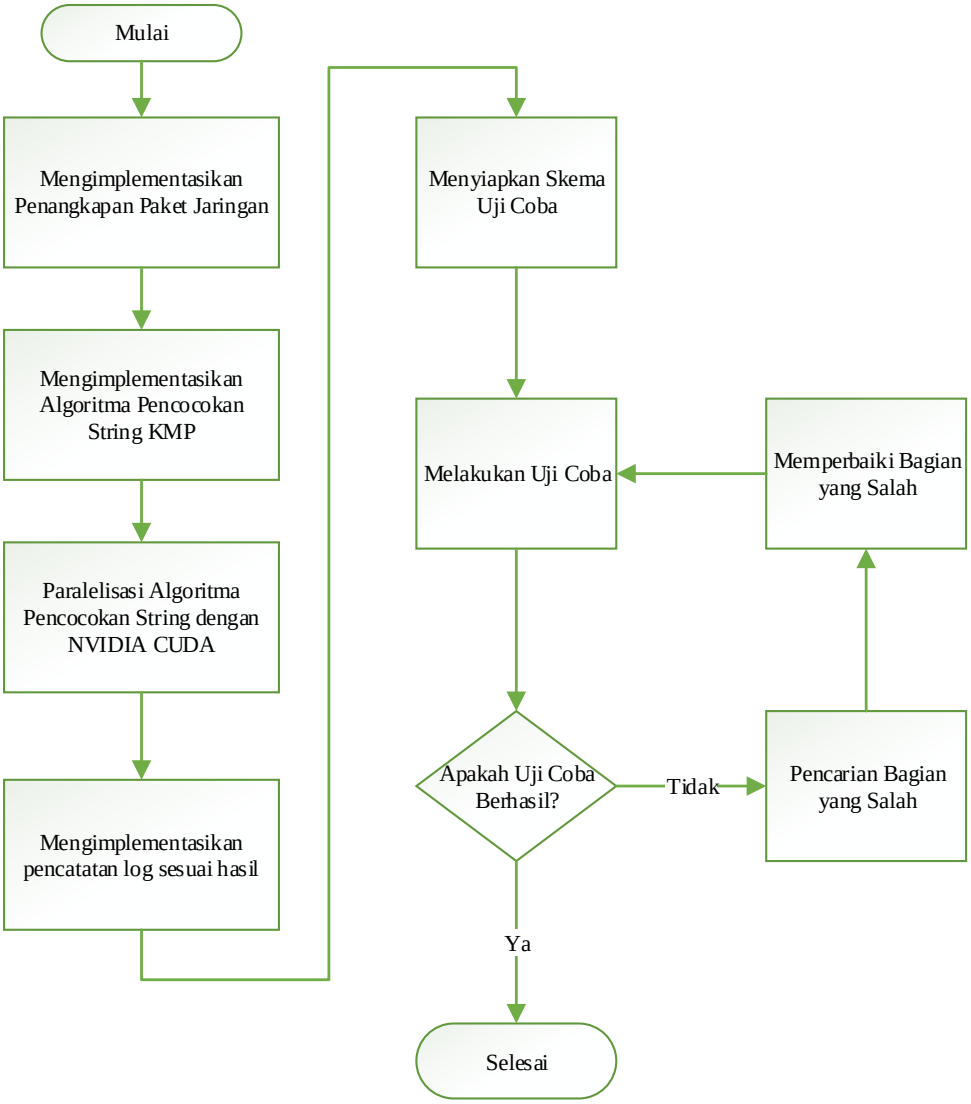
c. Mengembangkan algoritma paralel *custom* dan *library* yang digunakan pada bahasa pemororaman seperti C, C++, C#, Java, dll [3].

9. RINGKASAN ISI TUGAS AKHIR

Keamanan suatu jaringan adalah kebutuhan wajib bagi setiap penggunanya. Potensi–potensi serangan yang dapat merusak sistem pasti ingin dihindari oleh pengguna jaringan. IDS adalah solusi untuk mendeteksi serangan–serangan tersebut. Akan tetapi dengan bertumbuhnya jenis–jenis pemanfaatan internet seperti sekarang ini, jumlah paket yang lalu-lalang pada *traffic* jaringan dapat mencapai angka puluhan *Gigabit* pada setiap detiknya. Besarnya *traffic* data tersebut seringkali membuat suatu IDS kewalahan untuk menjalankan tugasnya. Kewalahan yang dimaksud adalah sistem pendeteksi menjadi tidak *real time* dalam pengerjaannya atau mencapai *system failure* dikarenakan beban kerjanya terlalu berlebihan [6].

Tugas akhir ini ditujukan untuk membuat NIDS (*Network Intrusion Detection System*) yang mampu menangani beban kerja yang berat seperti mengawasi paket data pada *traffic* yang tinggi. NIDS dibangun menggunakan algoritma KMP(Knuth-Morris-Pratt) untuk pencocokan *string*, dalam hal ini *string* yang dimaksud adalah paket–paket yang diterima dan kumpulan *rule* yang dibuat. NIDS akan didesain untuk berjalan pada jaringan yang memiliki *data traffic* yang tinggi, metode yang diterapkan yaitu paralelisasi algoritma KMP dengan NVIDIA CUDA digunakan untuk menjaga

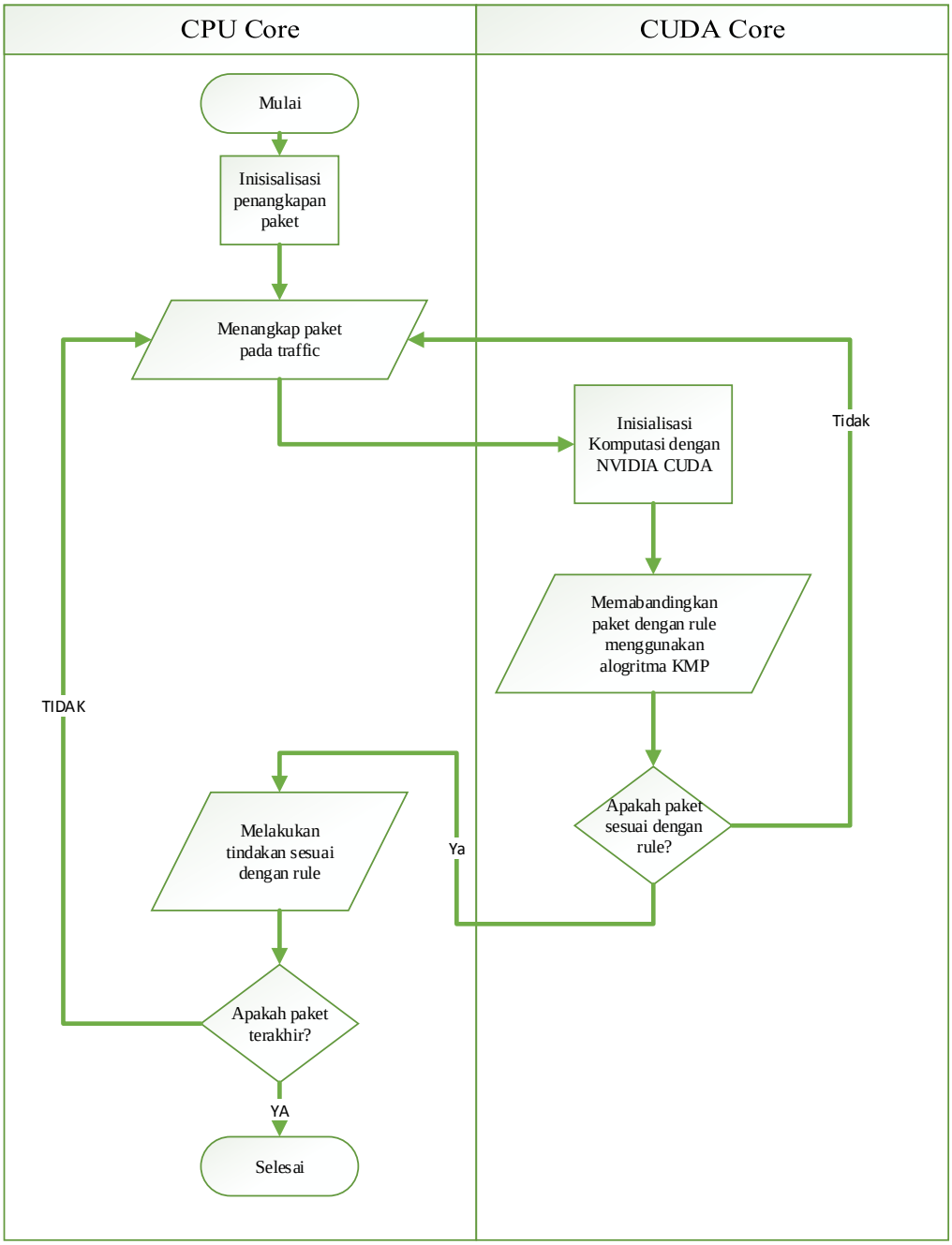
performa aplikasi agar tidak menurun dan memberi beban berat pada CPU. Secara umum adalah membagi kerja antar CPU dan GPU untuk pengawasan paket pada jaringan. Langkah-langkah pengerjaan tugas akhir ini akan dijelaskan pada Gambar 1.



Gambar 1. Langkah-langkah pengerjaan tugas akhir

Dari Gambar 1 ditekankan bahwa inti dari Tugas Akhir ini adalah paralelisasi Algoritma KMP yang digunakan untuk pencocokan *string* pada IDS.

Berikut alur kerja dari aplikasi:



Gambar 2. Alur kerja aplikasi

Pada Gambar 2 dapat dilihat bahwa aplikasi dapat melakukan paralelisasi komputasi algorituma KMP ke GPU, namun penangkapan paket dan pencatatan log tetap dilakukan dengan CPU.

10.METODOLOGI

a. Penyusunan proposal tugas akhir

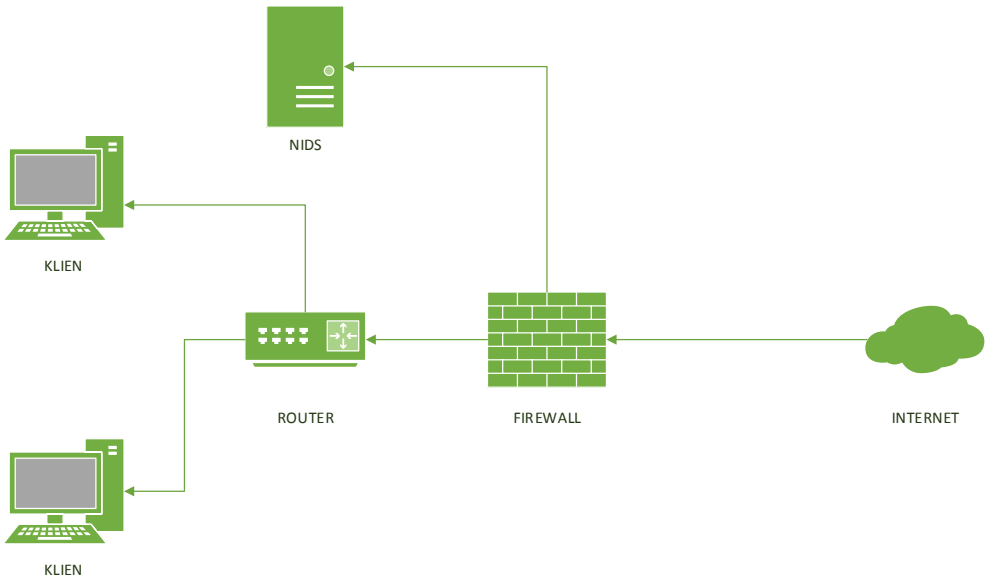
Proposal Tugas Akhir ini berisikan mengenai pembangunan aplikasi yang dibuat untuk mendeteksi serangan yang diarahkan pada jaringan dimana tempat aplikasi berada. Pengawasan ini diharapkan tidak terpengaruh dengan kondisi *traffic* jaringan, meskipun beban kerja berat aplikasi ini masih dapat berjalan dengan bantuan proses pencocokan yang berjalan pada GPU. Apabila serangan terdeteksi aplikasi akan mencatatnya pada *log*.

b. Studi literatur

Tugas Akhir ini menggunakan literatur *paper* beserta artikel dari internet. *Paper* yang menjadi acuan dalam pengerjaan tugas akhir ini adalah *"Gnort: High performance network intrusion detection using graphics processors"*, *"Network Intrusion Detection System Using KMP Algorithm"*, dan *"Parallelization of KMP String Matching Algorithm on Different SIMD architectures: Multi-Core and GPGPU's"*. *Paper* tersebut menjelaskan mengenai penggunaan algoritma KMP pada IDS, paralelisasi algoritma KMP dengan GPU, dan pemindahan komputasi pencocokan pola ke GPU untuk mempercepat proses deteksi serangan [6] [7] [8].

c. Analisis dan Desain Perangkat Lunak

Dalam aplikasi ini IDS akan diletakkan di belakang *firewall*, di mana aplikasi akan berada di tengah-tengah *traffic* data untuk memeriksa paket-paket apa sajakah yang sedang lalu-lalang, sesuai dengan Gambar 3.



Gambar 3. Arsitektur jaringan

d. Implementasi perangkat lunak

Dalam pembuatan aplikasi digunakan beberapa teknologi untuk dapat mengaplikasikan rancangan yang sudah ada, antara lain:

a. Bahasa Pemrograman Aplikasi

Aplikasi ini dibangun dengan menggunakan bahasa pemrograman Java. Penggunaan bahasa pemrograman diharapkan dapat membantu menangani kebutuhan aplikasi terutama kemudahan untuk konektivitas dengan basis data dan kebutuhan lainnya.

b. IDE

Pengembangan aplikasi ini menggunakan Netbeans 7.2 sebagai IDE.

c. Modeling Tools

Beberapa *modeling tools* yang digunakan untuk mengembangkan aplikasi ini Power Designer 15.00, StarUML, dan Microsoft Visio 2010.

e. Pengujian dan Evaluasi

Pengujian dari Tugas Akhir "Paralelisasi Algoritma Pencocokan String Knuth-Morris-Pratt dengan NVIDIA CUDA pada Aplikasi Network Intrusion Detection System" Unit akan diujikan di Laboraturium Arsitektur dan Jaringan Komputer Teknik Informatika ITS dan yang akan diujikan antara lain sebagai berikut:

1. Melakukan uji coba dengan keadaan *traffic* jaringan tidak mengandung serangan.
2. Melakukan uji coba dengan keadaan *traffic* jaringan mengandung serangan.
3. Membandingkan performa aplikasi menggunakan GPU dan tidak menggunakan GPU.

f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
    a. Latar Belakang
    b. Rumusan Masalah
    c. Batasan Tugas Akhir
    d. Tujuan
    e. Metodologi

f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

11. JADWAL KEGIATAN

Tabel 1 merupakan jadwal kegiatan dari pengerjaan Tugas Akhir "Paralelisasi Algoritma Pencocokan *String* Knuth-Morris-Pratt dengan NVIDIA CUDA pada Aplikasi *Network Intrusion Detection System*".

Tabel 1. Jadwal Kegiatan Tugas Akhir

Tahapan	2014																			
	Maret				April				Mei				Juni				Juli			
Penyusunan Proposal																				
Studi Literatur																				
Perancangan system																				
Implementasi																				
Pengujian dan evaluasi																				
Penyusunan buku																				

12. DAFTAR PUSTAKA

[1] SANS Institute, "Understanding Intrusion Detection System," *SANS Institute Reading Room*, pp. 1-9, 2001.

[2] L. Tan and Timothy Sherwood, "A High Throughput String Matching Architecture for Intrusion Detection and Prevention," *IEEE*, pp. 112-122, 2005.

[3] NVIDIA, "What is CUDA | NVIDIA Developer Zone," [Online]. Available: https://developer.nvidia.com/what-cuda. [Accessed 3 March 2014].

[4] T. H. Cormen, C. E. Leiserson , R. L. Rivest and C. Stein, "The Knuth-Morris-Pratt Algorithm," in *Introduction to Algorithm (3rd ed.)*, Massachusetts, MIT Press, 2009, pp. 1002-1013.

[5] NVIDIA, "Graphic Processing Unit (GPU) | NVIDIA," [Online]. Available: http://www.nvidia.com/object/gpu.html. [Accessed 3 March 2014].

[6] G. Vasiliadis, M. Polychronakis, E. P. Markatos and S. Ioannidis, "Gnort: High Performance Network Intrusion Detection System Using Graphic Processor," Institute of Computer Science, Foundation for Research and Technology, Hellas, 2008.

[7] B. Rahu and B. Srinivas, "Network Intrusion Detection System Using KMP Pattern Matching Algorithm," *International Journal of Computer Science and Telecommunications*, vol. 3, no. 1, pp. 33-36, 2012.

[8] A. Rasool and N. Khare, "Parallelization of KMP String Matching Algorithm on Different SIMD architectures: Multi-Core and GPGPU's," *International Journal of Computer Application*, vol. 11, no. 7, pp. 26-28, 2012.