

# ULTIMA Computing

## Jurnal Sistem Komputer

**DANIEL HERO FERSIL PUNUINDOONG, HIRA MEIDIA**

Rancang Bangun Aplikasi Pembelajaran Chord Dan Melodi Gitar Berbasis Augmented Reality (01-06)

**FRANSISCUS ATI HALIM**

Distribusi Pengiriman File Multimedia Secara Realtime Dengan Jaringan WAN Frame Relay (07-11)

**ADHI KUSNADI, ABRAHAM KHRISNANDI WICAKSONO**

Perbandingan Algoritma Horspool Dan Algoritma Zhu-Takaoka dalam Pencarian String Berbasis Desktop (12-16)

**IVAN JONATHAN, ALBERT YEUSIAWAN HARYONO, KEVIN LEONARDI**

Penelitian Mengenai Metode Steganografi Least Significant Bit (17-20)

**ENDANG TIRTANA, EKO BUDI SETIAWAN**

Aplikasi Pengendali Kamera DSLR Nirkabel Tipe Low End Berbasis Android (21-29)

**SUNDERI PRANATA, HARGYO TRI NUGROHO, HIROFUMI YAMAKI**

Analisis Dan Implementasi Protokol Otentikasi Fido U2F (30-35)

**HARGYO TRI NUGROHO**

Pengaruh Algoritma Stemming Nazief-Adriani Terhadap Kinerja Algoritma Winnowing Untuk Mendeteksi Plagiarisme Bahasa Indonesia (36-40)

**SAMUEL, CAHYO EKO SAMUDERA**

Rancang Bangun Mekanisme Load Sharing Pada Link Aggregation Menggunakan Software Defined Networking (41-47)



**UMN**

## SUSUNAN REDAKSI

### Pelindung

Dr. Ninok Leksono

### Penanggung Jawab

Dr. Ir. P.M. Winarno, M.Kom.

### Pemimpin Umum

Hargyo Tri Nugroho Ignatius, S.Kom., M.Sc.

### Mitra Bestari

Hira Meidia, B.Eng., Ph.D. (UMN)

Dr. Ranga Winantyo, Ph.D. (UMN)

Dr. Hugeng, S.T., M.T. (UMN)

Filbert Hilman Juwono, S.T., M.T.

(Universitas Indonesia)

Nur Afny Catur Andryani, M.Sc.

(Tanri Abeng University)

Kanisius Karyono, S.T., M.T. (UMN)

Felix Lokananta, S.Kom., M.Eng.Sc. (UMN)

Samuel, M.T.I. (UMN)

Adhi Kusnadi, S.T., M.Si. (UMN)

Wolfgang Xaverius D. J. N., S.T., M.Sc. (UMN)

Nabila Husna Shabrina, S.T., M.T. (UMN)

### Ketua Dewan Redaksi

Ni Made Satvika Iswari, S.T., M.T.

### Dewan Redaksi

Felix Lokananta, S.Kom., M.Eng.Sc.

Samuel, M.T.I.

### Desainer & Layouter

Wella, S.Kom, M.MSI.

### Sirkulasi dan Distribusi

Sularmin

### Keuangan

I Made Gede Suteja

## ALAMAT REDAKSI

Universitas Multimedia Nusantara (UMN)

Jl. Scientia Boulevard, Gading Serpong

Tangerang, Banten, 15811

Tlp. (021) 5422 0808

Faks. (021) 5422 0800

Email: ultimacomputing@umn.ac.id



**Jurnal ULTIMA Computing** merupakan Jurnal Program Studi Sistem Komputer Universitas Multimedia Nusantara yang menyajikan artikel-artikel penelitian ilmiah dalam bidang Sistem Komputer serta isu-isu teoritis dan praktis yang terkini, mencakup komputasi, organisasi dan arsitektur komputer, *programming*, *embedded system*, sistem operasi, jaringan dan internet, integrasi sistem, serta topik lainnya di bidang Sistem Komputer.

# Call for Paper



**International Journal of New Media Technology (IJNMT)** is a scholarly open access, peer-reviewed, and interdisciplinary journal focusing on theories, methods and implementations of new media technology. IJNMT is published annually by Faculty of Engineering and Informatics, Universitas Multimedia Nusantara in cooperation with UMN Press. Topics include, but not limited to digital technology for creative industry, infrastructure technology, computing communication and networking, signal and image processing, intelligent system, control and embedded system, mobile and web based system, robotics

## Important Dates

- October 31<sup>st</sup>, 2017**  
Deadline for submission of papers
- November 30<sup>th</sup>, 2017**  
Announcement for Acceptance
- December 15<sup>th</sup>, 2017**  
Deadline for submission of final papers



**Jurnal ULTIMATICS** merupakan Jurnal Program Studi Teknik Informatika Universitas Multimedia Nusantara yang menyajikan artikel-artikel penelitian ilmiah dalam bidang analisis dan desain sistem, *programming*, algoritma, rekayasa perangkat lunak, serta isu-isu teoritis dan praktis yang terkini, mencakup komputasi, kecerdasan buatan, pemrograman sistem *mobile*, serta topik lainnya di bidang Teknik Informatika.



**Jurnal ULTIMA InfoSys** merupakan Jurnal Program Studi Sistem Informasi Universitas Multimedia Nusantara yang menyajikan artikel-artikel penelitian ilmiah dalam bidang Sistem Informasi, serta isu-isu teoritis dan praktis yang terkini, mencakup sistem basis data, sistem informasi manajemen, analisis dan pengembangan sistem, manajemen proyek sistem informasi, *programming*, *mobile information system*, dan topik lainnya terkait Sistem Informasi.



**Jurnal ULTIMA Computing** merupakan Jurnal Program Studi Sistem Komputer Universitas Multimedia Nusantara yang menyajikan artikel-artikel penelitian ilmiah dalam bidang Sistem Komputer serta isu-isu teoritis dan praktis yang terkini, mencakup komputasi, organisasi dan arsitektur komputer, *programming*, *embedded system*, sistem operasi, jaringan dan internet, integrasi sistem, serta topik lainnya di bidang Sistem Komputer.

## DAFTAR ISI

<b>Rancang Bangun Aplikasi Pembelajaran Chord dan Melodi Gitar Berbasis Augmented Reality</b>	
Daniel Hero Fersil Punuindoong, Hira Meidia	1-6
<b>Distribusi Pengiriman File Multimedia Secara RealTime dengan Jaringan WAN Frame Relay</b>	
Fransiscus Ati Halim	7-11
<b>Perbandingan Algoritma Horspool dan Algoritma Zhu-Takaoka dalam Pencarian String Berbasis Desktop</b>	
Adhi Kusnadi, Abraham Khrisnandi Wicaksono	12-16
<b>Penelitian Mengenai Metode Steganografi Least Significant Bit</b>	
Ivan Jonathan, Albert Yeusiawan Haryono, Kevin Leonardi	17-20
<b>Aplikasi Pengendali Kamera DSLR Nirkabel Tipe Low End Berbasis Android</b>	
Endang Tirtana, Eko Budi Setiawan	21-29
<b>Analisis dan Implementasi Protokol Otentikasi FIDO U2F</b>	
Sunderi Pranata, Hargyo Tri Nugroho, Hirofumi Yamaki	30-35
<b>Pengaruh Algoritma Stemming Nazief-Adriani Terhadap Kinerja Algoritma Winnowing untuk Mendeteksi Plagiarisme Bahasa Indonesia</b>	
Hargyo Tri Nugroho	36-40
<b>Rancang Bangun Mekanisme Load Sharing Pada Link Aggregation Menggunakan Software Defined Networking</b>	
Samuel, Cahyo Eko Samudera	41-47

## KATA PENGANTAR

Salam ULTIMA!

ULTIMA Computing – Jurnal Sistem Komputer UMN kembali menjumpai para pembaca dalam terbitan saat ini Edisi Juni 2017, Volume IX, No. 1. Jurnal ini menyajikan artikel-artikel ilmiah hasil penelitian mengenai komputasi, organisasi dan arsitektur komputer, programming, embedded system, sistem operasi, jaringan dan internet, integrasi sistem, serta topik lainnya di bidang Sistem Komputer.

Pada ULTIMA Computing Edisi Juni 2017 ini, terdapat delapan artikel ilmiah yang berasal dari para peneliti, akademisi, dan praktisi di bidang Sistem Komputer, yang mengangkat beragam topik, antara lain: Rancang Bangun Aplikasi Pembelajaran *Chord* dan Melodi Gitar Berbasis *Augmented Reality*; Distribusi Pengiriman *File Multimedia* Secara *RealTime* dengan Jaringan *WAN Frame Relay*; Perbandingan Algoritma Horspool dan Algoritma Zhu-Takaoka dalam Pencarian String Berbasis *Desktop*; Penelitian Mengenai Metode Steganografi *Least Significant Bit*; Aplikasi Pengendali Kamera DSLR Nirkabel Tipe *Low End* Berbasis Android; Analisis dan Implementasi Protokol Otentikasi FIDO U2F; Pengaruh Algoritma *Stemming* Nazief-Adriani Terhadap Kinerja Algoritma *Winnowing* untuk Mendeteksi Plagiarisme Bahasa Indonesia; Rancang Bangun Mekanisme *Load Sharing* Pada Link Aggregation Menggunakan *Software Defined Networking*.

Pada kesempatan kali ini juga kami ingin mengundang partisipasi para pembaca yang budiman, para peneliti, akademisi, maupun praktisi, di bidang Teknik dan Informatika, untuk mengirimkan karya ilmiah yang berkualitas pada: *International Journal of New Media Technology* (IJNMT), ULTIMATICS, ULTIMA InfoSys, ULTIMA Computing. Informasi mengenai pedoman dan *template* penulisan, serta informasi terkait lainnya dapat diperoleh melalui alamat surel [ultimacomputing@umn.ac.id](mailto:ultimacomputing@umn.ac.id).

Akhir kata, kami mengucapkan terima kasih kepada seluruh kontributor dalam ULTIMA Computing Edisi Juni 2017 ini. Kami berharap artikel-artikel ilmiah hasil penelitian dalam jurnal ini dapat bermanfaat dan memberikan sumbangsih terhadap perkembangan penelitian dan keilmuan di Indonesia.

Juni 2017,

**Ni Made Satvika Iswari, S.T., M.T.**  
Ketua Dewan Redaksi

# Rancang Bangun Aplikasi Pembelajaran Chord dan Melodi Gitar Berbasis *Augmented Reality*

Daniel Hero Fersil Punuindoong<sup>1</sup>, H Meidia<sup>2</sup>.

Program Studi Sistem Komputer, Universitas Multimedia Nusantara, Tangerang, Indonesia  
dherofp@gmail.com<sup>1</sup>, hira.meidia@umn.ac.id<sup>2</sup>

Diterima 15 Maret 2017

Disetujui 5 Juni 2017

**Abstract** - In this study, the application of the basic concepts of chord and melody guitar is made by using marker-based Augmented Reality method. Unity 3D is used to build the application and Vuforia is used as a library of Augmented Reality, where the application can recognize the marker as the basis of the virtual 3D objects rendering chord and melody guitar by using Android smartphone. There are several colours and numbers to make it easier to know the type and position of the fingers used in forming a chord or a melody.

**Index Terms**- Augmented Reality, Guitar 3D Chord, Guitar 3D Melodi, Unity 3D, Vuforia, Android

## I. PENDAHULUAN

Saat ini, telah banyak orang yang memanfaatkan kemajuan teknologi komputer untuk belajar musik mulai dari drum, gitar, piano ataupun alat musik lainnya. Alat musik yang cukup sering ditemui dan mudah untuk dimainkan serta didukung dengan harga yang terjangkau adalah gitar. Selain itu, gitar juga memiliki range nada yang cukup lebar, sehingga alunan musik yang dihasilkan cenderung lembut dan disukai oleh banyak orang[1].

Salah satu teknologi yang dipakai dalam kegiatan pembelajaran adalah teknologi *Augmented Reality* (AR). AR merupakan teknologi yang menggabungkan dunia nyata dan dunia *virtual* melalui penambahan obyek-obyek tiga dimensi buatan komputer pada tampilan dunia nyata yang mampu memberikan pengalaman dan pemahaman yang mendalam bagi aktifitas pembelajaran[2][3].

Mengacu pada hal tersebut, teknologi AR dapat dijadikan alat untuk metode pembelajaran musik gitar yang lebih menarik dibandingkan metode pembelajaran standar dengan adanya tampilan obyek *virtual* 3D, sehingga pengguna bisa melihat dan mempelajari secara tiga dimensi bagaimana susunan bentuk *chord* dan melodi pada alat musik gitar.

Pada penelitian ini, dikembangkan sebuah aplikasi untuk media pembelajaran bentuk *chord* dan melodi pada alat musik gitar secara *mobile* berbasis

*Augmented Reality* menggunakan metode *marker based system* dengan *Vuforia* SDK sebagai *library* AR. Terdapat bentuk dasar *chord* mayor, minor seperti yang dapat dilihat pada tabel 1 dan melodi C, F, A beserta bentuk suara yang dapat dimunculkan, sehingga aplikasi ini dapat membantu proses pembelajaran bentuk *chord* dan melodi pada alat musik gitar.

## II. LANDASAN TEORI

### A. Augmented Reality

*Augmented Reality* (AR) merupakan teknologi yang menggabungkan benda-benda nyata dan maya pada lingkungan nyata dan dapat berjalan secara interaktif dalam waktu nyata[4]. Motokawa et.al [5] membuat sistem pendukung untuk memainkan gitar menggunakan AR.

AR mempunyai tiga karakteristik yang menjadi dasar diantaranya adalah kombinasi pada dunia nyata dan *virtual*, kemudian interaksi yang berjalan secara *real-time* dan yang terakhir adalah bentuk obyek yang berupa model 3 dimensi atau yang biasa disebut 3D di mana semuanya mencakup data lokasi, *audio*, *video* ataupun dalam bentuk data model 3D[3].

*Augmented Reality* mempunyai dua metode, yaitu sebagai berikut:

#### 1. Marker - Based Augmented Reality

Metode ini bersifat fiducial di mana pengguna harus mempunyai sebuah *marker* untuk menampilkan obyek *virtual* 3D menggunakan kamera. *Marker* biasanya merupakan ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. Komputer akan mengenali posisi dan orientasi *marker* kemudian menciptakan dunia *virtual* 3D yaitu titik (0,0,0) dan tiga sumbu yaitu X, Y, dan Z[6].

#### 2. Markerless Augmented Reality

Metode ini tidak menggunakan fiducial *marker* dengan pola unik, tetapi menggunakan bagian apapun

dari dunia nyata sebagai target dari *tracking* untuk penempatan obyek *virtual* [2].

## B. Alat Musik Gitar

Alat musik ini umumnya mempunyai enam senar yang terbuat dari bahan nilon maupun kawat dan dimainkan dengan jari-jemari tangan atau sebuah plektrum (alat petik gitar). Suara gitar dihasilkan dari getaran senar gitar yang dialirkan melalui sadel dan jembatan tempat pengikat senar ke dalam ruang suara[7].

### 1. Chord Gitar

*Chord* atau yang biasa dikenal sebagai kunci gitar merupakan sebuah bentuk rangkaian nada yang dibunyikan secara serentak untuk membangun keharmonisan pada musik[8]. *Chord* dasar gitar baik dalam mayor maupun minor dapat dilihat pada tabel 1.

Tabel 1. *Chord* Dasar Gitar

Chord Dasar Gitar	
Mayor	Minor
C	Cm
D	Dm
E	Em
F	Fm
G	Gm
A	Am
B	Bm

### 2. Melodi Gitar

Melodi adalah runtutan nada-nada yang dipetik dan dimainkan secara harmonis. Oleh sebab itu suatu permainan musik gitar akan terdengar lebih indah dengan adanya melodi. Variasi dari melodi gitar terbilang cukup banyak dikarenakan untuk mendapatkan kenikmatan dari setiap bunyi yang dimainkan. Namun, setiap variasi tersebut didasari dari tiga bentuk melodi yaitu C, F dan A yang dapat dilihat pada gambar 1, 2 dan 3[1][7].



Gambar 1. Melodi C, F, dan A

## C. Unity 3D

*Unity 3D* adalah sebuah *game engine* yang banyak digunakan untuk membuat game dengan obyek 3D ataupun untuk konteks interaktif lain seperti animasi 3D dan visualisasi arsitektur[3]. Keunggulan dari *Unity 3D* adalah mendukung pembuatan *game* berbasis *cross-platform* dan dapat dijalankan pada *Mac*, *Android*, *iOS*, *XBox*, *Playstation 3* dan *Wii*. Untuk membangun sebuah karya pada *Unity 3D*, disediakan berbagai pilihan bahasa pemrograman untuk mengembangkan *game*, antara lain *JavaScript*, *C#*, dan *Boo*[9].

## D. Vuforia

*Vuforia* merupakan *Software Development Kit* (SDK) yang dikembangkan oleh *Qualcomm* untuk membantu pengembang dalam menciptakan aplikasi atau *game* yang menggunakan teknologi *Augmented Reality*, sehingga membuat aplikasi atau *game* yang dibuat dengan teknologi ini akan terlihat lebih interaktif dan hidup. Pengembang dapat membuat obyek *virtual 3D* yang dapat berinteraksi dengan pengguna aplikasinya baik *game*, aplikasi pembelajaran, *video*, aplikasi dongeng, dan masih banyak lagi[10].

*Vuforia SDK* ini menggunakan teknologi *Computer Vision* untuk mengenali dan melacak gambar target dan obyek 3D secara *real-time*. Kemampuan registrasi citra ini memungkinkan pengembang untuk mengatur posisi dan orientasi obyek *virtual* seperti model 3D dan media lain dalam kaitannya dengan gambar dunia nyata ketika hal ini dilihat melalui kamera dari perangkat *mobile* dengan melacak posisi dan orientasi dari gambar secara *real-time* yang membuat perspektif pengguna pada obyek dapat sesuai dengan perspektif yang ada pada *image target*, sehingga tampak bahwa obyek *virtual* adalah bagian dari suatu adegan yang ada di dunia nyata. *Vuforia SDK* mendukung berbagai macam pengembangan aplikasi *Augmented Reality* pada *Unity 3D* dan kompatibel dengan berbagai perangkat *mobile* seperti *iPhone* dan *Android* [11].

## E. Android Studio

*Android Studio* adalah sebuah *Integrated Development Environment* (IDE) yang dikembangkan oleh *google* dan dipakai untuk pengembangan aplikasi *Android*. *Android Studio* merupakan pengembangan dari *Eclipse IDE*, dan dibuat berdasarkan IDE java populer, yaitu *intelliJ IDEA*[12].

## F. Adobe Photoshop CS6

*Adobe Photoshop CS6* adalah *software digital imaging* versi ke tiga belas buatan *Adobe Systems* yang dikhususkan untuk pengeditan gambar atau foto serta pembuatan efek dengan performa dan fitur *editing* gambar yang kuat serta *interface* yang *modern* dan



menarik[13]. Aplikasi ini digunakan untuk membuat gambar *marker*.

### G. Autodesk 3Ds Max

*Autodesk 3Ds Max* adalah sebuah perangkat lunak untuk pemodelan 3D dan animasi yang dikembangkan oleh *Autodesk Media and Entertainment*, di mana aplikasi ini sering digunakan untuk membuat *video* animasi dan visualisasi arsitektur 3D [14]. Aplikasi ini digunakan untuk membuat obyek 3D *chord* dan melodi gitar.

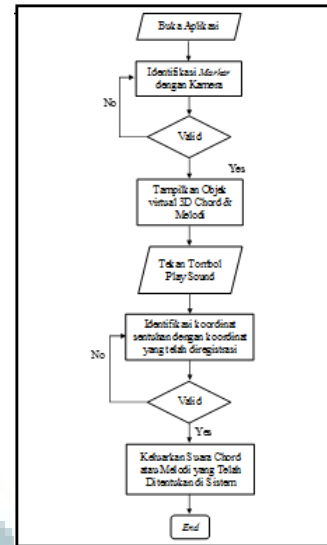
## III. METODOLOGI PENELITIAN

Penelitian ini dilakukan dalam beberapa tahapan yaitu perancangan system, pembuatan aplikasi dan implementasi dari aplikasi ini.

### A. Perancangan Sistem

Perancangan dari sistem pembelajaran *chord* dan melodi gitar berbasis *Augmented Reality* ini dideskripsikan dalam *flowchart* sistem pada gambar 4. Pada saat aplikasi dijalankan, terdapat pilihan *chord* mayor, minor dan melodi yang siap untuk dimainkan. Pengguna kemudian mengaktifkan kamera *smartphone Android* dan mengarahkannya pada sebuah *marker* yang telah diregistrasi dengan *Vuforia*, sehingga sistem akan mengidentifikasi koordinat *marker* untuk menyesuaikan posisi orientasi obyek yang ditangkap kamera. Apabila koordinat telah sesuai dengan yang telah ditentukan, maka sistem akan menampilkan bentuk obyek *virtual 3D chord* dan melodi gitar, sebaliknya jika tidak sesuai, maka tidak akan ada obyek *virtual 3D* yang akan ditampilkan oleh sistem.

Sebuah tombol *virtual 3D "Play Sound"* akan ditampilkan dalam waktu yang sama dengan obyek *virtual 3D chord* dan melodi. Tombol ini dapat disentuh oleh pengguna untuk memunculkan bentuk suara dari *chord* dan melodi gitar. Setelah tombol tersebut disentuh, sistem akan mengidentifikasi dan koordinat sentuhan dengan koordinat yang telah diregistrasi. Apabila hasil yang didapat sesuai, maka sistem akan memunculkan suara, tetapi sebaliknya apabila tidak sesuai dengan koordinat yang telah diregistrasi, maka sistem tidak akan memunculkan suara.



Gambar 4. *Flowchart* Sistem

### B. Metode Penelitian

Sistem yang telah selesai dirancang pada *Unity 3D*, kemudian diuji satu per satu untuk memastikan bahwa setiap modul yang telah dirancang dan diimplementasikan pada obyek 3D baik *chord* maupun melodi, suara dan antarmuka telah berjalan sesuai dengan fungsi masing-masing.

Selanjutnya, aplikasi dipasang pada sebuah *smartphone* dengan sistem operasi *Android Jelly Bean* untuk menguji apakah kamera dari *smartphone* dapat mengidentifikasi koordinat sebuah gambar *marker* yang telah diregistrasi dengan *Vuforia* untuk menampilkan obyek *virtual 3D chord* dan melodi dan memunculkan suara sesuai koordinat yang disentuh. Perbaikan dilakukan jika ada bagian-bagian yang tidak sesuai dengan harapan dalam perancangan awal. Realisasi aplikasi dianggap selesai apabila obyek *virtual chord* dan melodi berhasil ditampilkan, suara dari bentuk *chord* dan melodi berhasil dimunculkan dan fungsionalitas pada antarmuka telah berjalan dengan baik.

## IV. IMPLEMENTASI DAN PENGUJIAN

### A. Implementasi Obyek 3D Chord dan Melodi

Pada tahap implementasi ini, langkah dimulai dengan melakukan *import file .fbx* obyek *chord* dan melodi yang sudah didesain menggunakan *Autodesk 3DS Max*. *File* tersebut diletakkan pada *scene* yang sudah memiliki *frame marker*. Pada gambar 5 menunjukkan obyek F mayor diatur sebagai *child* dari *ImageTarget* yang ada di *Hierarchy*. Proses selanjutnya adalah mengatur posisi, rotasi dan skala dari obyek di atas *marker*. Terdapat beberapa warna yang diimplementasikan dalam penyusunan bentuk *chord* seperti merah untuk posisi jari telunjuk, biru untuk jari tengah, hijau untuk jari manis, kuning untuk

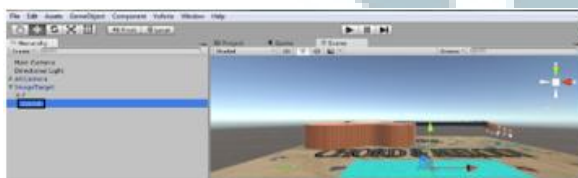
kelingking dan putih untuk *open string*. Khusus untuk *open string* dipakai dalam membentuk melodi.



Gambar 5. Import Obyek ke Unity 3D

B. Implementasi Suara *Chord* dan Melodi

Terdapat dua Game Object yang diimplementasikan untuk menginterpretasikan suara *chord* dan melodi. Pada gambar 6 merupakan *Prefab VirtualButton Sound*, diletakan pada *Hierarchy VirtualButton Sound*, dan dijadikan *child* dari *ImageTarget* kemudian dikonfigurasi dengan menambahkan komponen *Audio Source* pada *inspector* yang berfungsi untuk mengakses *file* suara seperti yang dapat dilihat pada gambar 8. Selanjutnya, pada gambar 7 menampilkan penambahan *GameObject cube* yang di-*rename* menjadi *Click* dan dijadikan *child ImageTarget* sebagai info yang merepresentasikan *VirtualButton Sound*. Pada *GameObject Click*, ditambahkan sebuah *script VirtualButtonEventHandler* untuk mendeklarasikan *variable Audio Source* agar suara bisa muncul ketika ada sentuhan tangan pada koordinat bidang *VirtualButton Sound* seperti yang dapat dilihat pada 9.



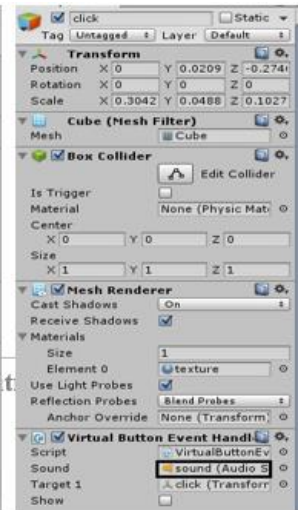
Gambar 6. Prefab *VirtualButton Sound* di *ImageTarget*



Gambar 7. *GameObject Click* di *ImageTarget*



Gambar 8. *Component Audio Source*



Gambar 9. Deklarasi *Variabel Sound*

C. Implementasi Antarmuka

1. Antarmuka Halaman Awal

Pada saat *launcher* selesai, maka aplikasi akan masuk ke halaman awal atau *menu* awal. Terdapat tiga tombol yang diimplementasikan pada *menu* awal yaitu “PLAY” “TUTORIAL” dan “EXIT” seperti yang dapat dilihat pada gambar 10.



Gambar 10. Antarmuka *Menu* Awal Aplikasi

2. Antarmuka *Menu Tutorial*

*Menu tutorial* ini menampilkan cara penggunaan aplikasi. *Menu* ini dapat diakses dengan menekan tombol *tutorial* yang terdapat pada halaman awal. Gambar 11 menampilkan *menu tutorial* aplikasi.



Gambar 11. Antarmuka *Tutorial* Aplikasi

### 3. Antarmuka Menu Play

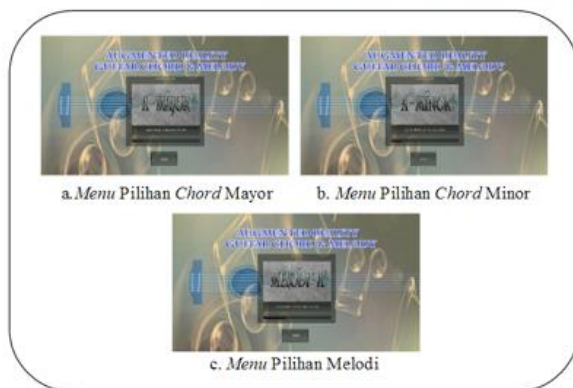
Terdapat empat buah tombol pada *menu play* ini, yaitu “CHORD-MAYOR”, “CHORD-MINOR”, “MELODI” dan “BACK”. *Menu* ini dapat diakses dengan menekan tombol “PLAY” pada *menu* awal. Gambar 12 menampilkan antarmuka dari *menu play*.



Gambar 12. Antarmuka Menu Play

### 4. Antarmuka Menu Pilihan Chord dan Melodi

Terdapat sebuah tombol *scroll* pada *menu* ini yang berfungsi untuk memberikan kemudahan kepada pengguna dalam menentukan *chord* atau melodi yang akan di-*play*. *Menu* pilihan *chord* mayor menampilkan tujuh pilihan *chord* mayor, *menu* pilihan *chord* minor menampilkan tujuh pilihan *chord* minor dan *menu* pilihan melodi menampilkan tiga pilihan melodi. Gambar 13 menampilkan antarmuka *menu* pilihan *chord* mayor, minor dan melodi. Terdapat tombol “Click Here To Play” pada setiap pilihan *chord* dan melodi yang berfungsi untuk mengaktifkan kamera yang siap untuk melakukan *tracking* pada *marker*.



Gambar 13. Antarmuka Menu Pilihan Chord dan Melodi

### D. Pengujian Sistem

Pengujian pertama dilakukan pada setiap obyek *chord* dan melodi. Sebanyak tujuh belas obyek yang terdiri dari tujuh obyek *chord* mayor, tujuh *chord* minor dan tiga *chord* mayor berhasil dimunculkan dalam bentuk *Augmented Reality* pada saat kamera mendeteksi sebuah *Marker* yang telah diregistrasi dengan *Vuforia*. Gambar 14 Menampilkan obyek

*virtual 3D* gitar *full body* yang muncul di atas *marker* dan gambar 15 menampilkan kamera *smartphone* yang digeser ke arah *neck* gitar untuk melihat bentuk *chord* secara jelas.

Pengujian kedua dilakukan pada setiap bentuk suara *chord* dan melodi. Muncul atau tidaknya bentuk suara *chord* dan melodi diuji dengan menekan tombol “Play Sound” yang muncul pada layar *smartphone* atau di atas *marker* ketika kamera berhasil mengenali koordinat yang telah ditentukan pada *marker*. Hasil Pengujian didapati bahwa sebanyak tujuh belas bentuk suara yang diimplementasikan untuk mewakili setiap bentuk *chord* dan melodi telah berhasil dimunculkan dan tidak terjadi kesalahan, artinya jenis suara muncul sesuai dengan bentuk *chord* atau melodi yang ditampilkan.

Pengujian ketiga dilakukan untuk menguji setiap fungsionalitas tombol-tombol untuk mengakses setiap *menu* yang telah berjalan sesuai dengan desain pada implementasi antarmuka.



Gambar 14. Kamera Mendeteksi Marker



Gambar 15. Smartphone digeser ke arah Neck Gitar

## V. KESIMPULAN

Melalui penelitian ini, telah dibuat sebuah aplikasi yang dapat dijalankan pada *smartphone* dengan sistem operasi *Android* minimum *Jelly Bean 4.1*. Aplikasi ini merupakan pengenalan dasar *chord* dan melodi gitar dengan memanfaatkan teknologi *Augmented Reality* yang sudah berjalan dengan baik, di mana pengguna dapat melihat obyek *chord* dan melodi gitar yang ditampilkan secara *virtual 3D* dengan menggunakan kamera *smartphone* untuk mendeteksi sebuah benda nyata *marker* yang telah teregistrasi dengan *Vuforia* SDK, sehingga obyek *virtual 3D chord* dan melodi gitar akan tampil pada layar *smartphone*. Terdapat juga

suara dari setiap *chord* dan melodi yang bisa didengarkan oleh pengguna dengan menyentuh sebuah tombol yang tersedia pada saat obyek 3D *chord* dan melodi dimunculkan.

Pada aplikasi ini total menampilkan tujuh belas bentuk obyek 3D dan suara yang terdiri dari tujuh bentuk *chord* mayor, tujuh bentuk *chord* minor dan tiga bentuk melodi di mana sebuah obyek *virtual* 3D gitar *full body* mewakili setiap bentuk *chord* dan melodi yang ditampilkan, sehingga pengguna harus melakukan mengarahkan kamera ke arah *neck* gitar untuk melihat dengan jelas bentuk *chord* dan melodi yang ditampilkan.

Terdapat beberapa warna dan nomor yang bertujuan untuk memudahkan penyusunan jari – jari dalam membentuk sebuah *chord* dan melodi, di mana warna merah untuk jari telunjuk, biru jari tengah, hijau jari manis, kuning jari kelingking dan putih untuk *open string* artinya senar dipetik bebas yang hanya dapat digunakan pada bentuk melodi.

#### SUMBER REFERENSI

- [1] Arya Adi Prakasa Bangun. Bandung. 2010. Deteksi Melodi pada Gitar Menggunakan Transformasi Wavelet. Universitas Telkom Bandung.
- [2] Halim Dareen. 2014. Skripsi. Tangerang. Rancang Bangun Aplikasi Ensiklopedia Berbasis Augmented Reality Menggunakan Intel Creative Senz 3D Perceptual Camera. Universitas Multimedia Nusantara.
- [3] Setya Iwan Nugraha. 2014. Semarang. Pemanfaatan Augmented Reality Untuk Pembelajaran Pengenalan Alat Musik Piano. Universitas Diponegoro.
- [4] Azuma Ronald T. 1997. A Survey of Augmented Reality. Hughes Research Laboratories 301 1 Malibu Canyon Road, MS RL96 Malibu, CA 90265.
- [5] Yoichi Motokawa, Hideo Saito, Support System for guitar playing using augmented reality display, Fifth IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR 2006, Santa Barbara, CA, USA, October 22-25, 2006.
- [6] Pengertian Augmented Reality (AR). [Online] Tersedia: <http://www.it-jurnal.com/2014/05/Pengertian-Augmented-Reality-AR.html>.
- [7] True Jesus Church. Akustik Gitar Tahap Pemula & Menengah. [Online]. Tersedia: <http://romaric.mywapblog.com/files/tutorial-gitar.pdf>.
- [8] Wisnudasitra Elghar. 2010. Bogor. Pengenalan Chord pada Alat Musik Gitar Menggunakan CodeBook dengan Teknik Ekstraksi Ciri MFCC. Jurnal Ilmiah Ilmu Komputer, 2010.
- [9] Gurusakti. Unity, Game Engine Sakti Untuk Berbagai Platform. [Online]. Tersedia: <https://gurusakti.wordpress.com/2012/02/15/unity-game-engine-sakti-untuk-berbagai-platform/>.
- [10] Vuforia – SDK canggih untuk wujudkan aplikasi dan game dengan teknologi augmented reality. [Online]. Tersedia : <http://teknojurnal.com/vuforia/>.
- [11] Apa itu Vuforia?. [Online]. Tersedia: <http://kecerdasanbuatangame.blogspot.co.id/2014/03/apa-itu-vuforia.html>.
- [12] Herdi Hafizh. Belajar membuat aplikasi Android menggunakan android studio. [Online]. Tersedia: <http://www.twoh.co/2014/09/belajar-membuat-aplikasiandroid-menggunakan-android-studio/>.
- [13] Riyanto Slamet. Adobe® Photoshop CS6. [Online]. Tersedia: <http://brainmatics.com/adobe-photoshop-cs6/>.
- [14] Sejarah 3Ds Max. Sejarah, Pengertian 3D Max dan Kelebihan Feature 3D Animation Dengan 3ds Max. [Online]: Tersedia :

<http://ubudbelajarkomputer.blogspot.co.id/2014/11/pengertian-3d-max-dan-keunggulan.html>

# Distribusi Pengiriman File Multimedia Secara RealTime dengan Jaringan WAN Frame Relay

Fransiscus Ati Halim

Program Studi Sistem Komputer, Universitas Multimedia Nusantara, Tangerang, Indonesia  
fransiscus.atil@lecturer.umn.ac.id

Diterima 25 Maret 2017

Disetujui 5 Juni 2017

**Abstract**—Today's communications networks are built using high-speed digital trunks that inherently provide high throughput, minimal relay, and a very low error rate. Such transmission networks supply highly reliable service without the overhead of error control functions. Frame relay is a packet-mode transmission network service that exploits these network characteristics by minimizing the amount of error detection and recovery performed inside the network [1]. In addition, real time network based systems can also minimize the possibility of employees committing fraud resulting in losses for the company. The research was carried out in a service provider video ad impressions company, which has fifteen branches in the islands of Java and Sumatra. The problem is the revenue reporting is not real-time and non-standard video file format ads to each branch. This is because the distribution process with the hard disk media are still using courier services. Based on user demand, it was decided to use the computer network using Frame Relay technology. With the computer network that connects all the branches to the head office, the data updating process can be done in a shorter time than the previous system and will reduce fraud forms of each branch and enable the achievement of a better quality of service to customers.

**Index Terms**—Computer Network, Frame Relay

## I. PENDAHULUAN

Teknologi Informasi dan Komunikasi sangat mendukung aktifitas dunia bisnis yang semakin padat dengan pertukaran data yang bersifat aktual, memiliki integritas yang sangat tinggi terutama bagi perusahaan yang memiliki banyak cabang [2]. Latar belakang perusahaan yang bergerak di bidang jasa penayangan video iklan adalah belum didukung oleh penggunaan Teknologi Jaringan Komputer terkoneksi seluruh cabang, sehingga seringkali terjadi masalah keterlambatan pengiriman data. Hal ini menjadi penghambat utama bagi kantor pusat untuk memonitor aktivitas yang telah dilakukan oleh masing-masing cabang. Untuk itu haruslah terdapat sebuah jaringan komputer yang dapat menghubungkan setiap cabang ke kantor pusat dengan waktu pengiriman data yang singkat dan terpercaya agar masalah yang dihadapi perusahaan yaitu keterlambatan distribusi pengiriman file iklan video yang tidak serentak antar cabang dapat dihilangkan atau diminimalisir. Selama ini

distribusinya menggunakan jasa kurir ke setiap cabang. Masalah berikutnya adalah pelaporan keuangan dari masing-masing cabang ke kantor pusat belum terkoneksi secara online sampai saat ini. Hal ini tentu saja rentan terhadap kesalahan manusia (*human error*) dan manipulasi oleh pegawai cabang

Agar dapat mencapai tujuan tersebut diperlukan dukungan jaringan komputer berbasis *Wide Area Network (WAN)* untuk meningkatkan pelayanan yang dapat memenuhi harapan pelanggan dan secara simultan meningkatkan kinerja perusahaan sesuai target.

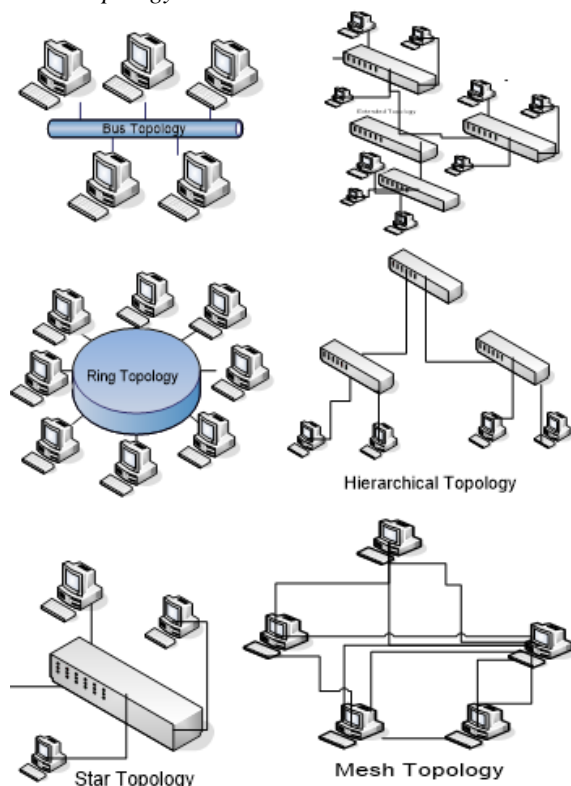
## II. LANDASAN TEORI

### A. Dasar-dasar Jaringan Komputer

Jaringan komputer adalah sebuah sistem komunikasi antara dua atau lebih komputer. Jaringan ini dapat bersifat permanen (terhubung melalui kabel) atau sementara (terhubung melalui *modem* atau *null modem*). Tujuan utama dari sistem komunikasi adalah pertukaran data antara dua pihak. Elemen-elemen utama dari model komunikasi adalah sebagai berikut: a) Sumber (*source*): Alat ini menghasilkan data untuk dikirim contohnya adalah telepon dan *Personal Computer (PC)*, b) *Transmitter*: berfungsi mengubah dan mengkodekan informasi dengan suatu cara tertentu untuk menghasilkan sinyal elektromagnetik yang dapat dikirimkan melalui suatu sistem transmisi. Sebagai contoh, *modem (modulator-demodulator)* mengambil aliran bit digital dari suatu alat yang terhubung seperti PC dan mengubah aliran bit tersebut menjadi sinyal analog yang dapat diterima oleh jaringan telepon., c) Sistem Transmisi: dapat berupa saluran transmisi tunggal atau jaringan kompleks yang menghubungkan sumber dan tujuan. d) *Receiver*: Menerima sinyal dari sistem transmisi dan mengubahnya ke dalam bentuk yang dapat diterima oleh tujuan. Sebagai contoh, sebuah modem akan menerima sinyal analog yang datang dari jaringan atau *line* transmisi dan mengubahnya ke dalam aliran bit digital. e) Tujuan (*destination*): mengambil data yang datang dari penerima. [3]

## B. Topologi Jaringan Komputer

Jenis-jenis topologi dalam LAN: a) *Physical Topology*: Bentuk jaringan komputer sebenarnya yang dihubungkan secara langsung. *Physical Topology* yang sering dipakai adalah sebagai berikut: *Bus Topology*, *Ring Topology*, *Star Topology*, *Extended Star Topology*, *Hierarchical Topology*, *Mesh Topology*



Gambar 1 *Physical Topology* LAN

### b) *Logical Topology*

Bentuk jaringan komputer yang menjelaskan bagaimana sebuah *host* berkomunikasi melalui media perantara. Dua tipe *logical topology* yang sering digunakan adalah: *Broadcast Topology*, *Token-passing*

## C. Jaringan Wide Area Network (WAN)

Jenis-jenis jaringan komputer : a) *Local Area Network (LAN)* : LAN adalah sebuah jaringan komputer yang dibatasi oleh area geografis yang relatif kecil dan umumnya dibatasi oleh area lingkungan seperti perkantoran atau sebuah sekolah dan biasanya ruang lingkup yang dicakupnya tidak lebih dari 2 km<sup>2</sup>. LAN didesain untuk: 1) beroperasi pada wilayah geografi yang terbatas, 2) memungkinkan banyak *user* untuk mengakses media dengan kecepatan tinggi, 3) menyediakan koneksi ke layanan lokal setiap saat (seperti printer dan *file* di *server*), menghubungkan peralatan yang berdekatan. b) *Metropolitan Area Network (MAN)* : Sebuah MAN mencakup area yang lebih besar dari

LAN, misalnya antarwilayah dalam satu propinsi. MAN juga dapat menghubungkan beberapa LAN menjadi suatu bagian jaringan yang lebih besar. Cakupan geografis dari MAN terbatas pada area geografis yang sama. c) WAN adalah jaringan yang ruang lingkungannya sudah terpisahkan oleh batas geografis. WAN memiliki sejumlah *switching nodes* yang saling terhubung. Transmisi dari salah satu peralatan diteruskan melalui *node* internal ke peralatan tujuan. *Node* ini tidak berhubungan dengan isi data. Tujuan *node* adalah untuk menyediakan fasilitas *switching* yang akan memindahkan data dari satu *node* ke *node* lainnya hingga sampai ke tujuan [3]

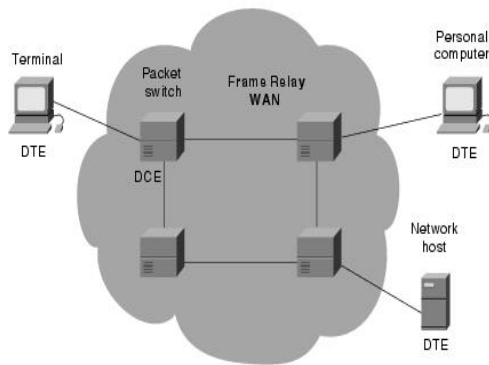
## D. Teknologi Frame Relay

Frame Relay adalah salah satu protokol WAN yang memiliki performa tinggi yang beroperasi pada *physical* dan *data link layer* pada model OSI (*Open System Interconnection*). Pada awalnya Frame Relay didesain untuk penggunaan pada *interface Integrated Services Digital Network (ISDN)*. Sekarang, Frame Relay juga digunakan pada *interface* jaringan lainnya. Frame Relay adalah satu contoh teknologi *packet-switched*. Jaringan *packet-switched* memungkinkan *end stations* untuk berbagi media jaringan dan *bandwidth* yang tersedia secara dinamis.[4]

Dua teknik yang digunakan pada teknologi *packet-switching*: a) *Variable-length Packets* : digunakan untuk transfer data yang lebih efisien dan fleksibel. Paket-paket ini diedarkan antara segmen-segmen yang berbeda dalam jaringan hingga tiba di tujuan. b) *Statistical Multiplexing*: Teknik ini mengendalikan akses jaringan dalam suatu jaringan *packet-switched*. Keunggulan teknik ini adalah dapat mengakomodasi penggunaan *bandwidth* sehingga lebih fleksibel dan efisien. Contoh yang menggunakan teknik tersebut saat ini adalah *Ethernet* dan *Token Ring* [5].

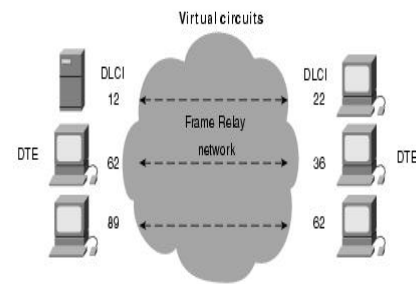
Peralatan yang terhubung pada WAN Frame Relay terbagi atas dua kategori umum: a) *Data Terminal Equipment (DTE)*: DTE secara umum dianggap sebagai *terminating equipment* untuk suatu jaringan khusus dan biasanya terletak pada sisi *client*. Contoh peralatan DTE adalah terminal, PC, IPPhone, Server, Printer b) *Data Circuit-terminating Equipment (DCE)* : DCE adalah peralatan antar jaringan *carrier*. Contoh peralatan DCE adalah Modem, Router, Switch. Tujuan dari DCE adalah menyediakan layanan *clocking* dan *switching* dalam suatu jaringan. [6]

Gambar berikut ini menunjukkan hubungan antara dua kategori peralatan:



Gambar 2. Peralatan DCE Umumnya Berada Dalam Carrier-Operated WAN

*Virtual circuit* menyediakan jalur komunikasi dua arah dari satu peralatan DTE ke peralatan DTE lainnya dan diidentifikasi secara unik oleh *data-link connection identifier* (DLCI). *Frame Relay virtual circuit* dibagi menjadi dua kategori: a) *Switched Virtual Circuit* (SVC) : Adalah koneksi sementara yang digunakan pada situasi yang hanya memerlukan transfer data sporadis antara peralatan DTE dalam jaringan *Frame Relay*. Sebuah sesi komunikasi dalam SVC terdiri dari empat tahap operasional: 1) *Call Setup* -- terciptanya virtual circuit antara dua peralatan DTE *Frame Relay*, 2) *Data transfer* -- data ditransmisi antara peralatan DTE melalui *virtual circuit*, 3) *Idle* -- koneksi antara peralatan DTE tetap terjalin tetapi tidak ada transfer data. Jika sebuah SVC berada pada tahap *idle* selama periode tertentu, panggilan dapat diputuskan, 4) *Call termination* -- *virtual circuit* antara peralatan DTE diputuskan. Setelah *virtual circuit* diputuskan, peralatan DTE harus membangun sebuah SVC baru jika ada tambahan pertukaran data. Hanya beberapa produsen peralatan DCE *Frame Relay* mendukung koneksi *switched virtual circuit*. Maka dari itu, penggunaannya sangat minim dalam jaringan *Frame Relay* saat ini. Perusahaan telah menyadari bahwa SVC menghemat biaya karena sirkuit tidak selalu terbuka. b) *Permanent Virtual Circuit* (PVC) : PVC adalah koneksi permanen yang digunakan untuk transfer data antarperalatan DTE melalui jaringan *Frame Relay* secara konsisten. Komunikasi melalui PVC tidak membutuhkan tahap *call setup* dan *termination* yang digunakan oleh SVC. PVC selalu beroperasi dalam salah satu tahap di bawah ini: 1) *Data transfer* -- data ditransmisi antara peralatan DTE melalui virtual circuit, 2) *Idle*-- koneksi antara peralatan DTE masih terjalin tetapi tidak ada transfer data. Tidak seperti SVC, koneksi PVC tidak akan terputus walaupun di dalam tahap *idle*. Peralatan DTE dapat mulai mentransfer data kapanpun karena sirkuit terhubung secara permanen [6].



Gambar 3. *Frame Relay Virtual Circuit Tunggal Dengan DLCI Berbeda Pada Tiap Virtual Circuit.*

Gambar 3 mengilustrasikan bagaimana dua peralatan DTE yang berbeda dapat diberi nilai DLCI yang sama di dalam sebuah *Frame Relay WAN* [6].

### III. ANALISIS SISTEM BERJALAN DAN USULAN SISTEM

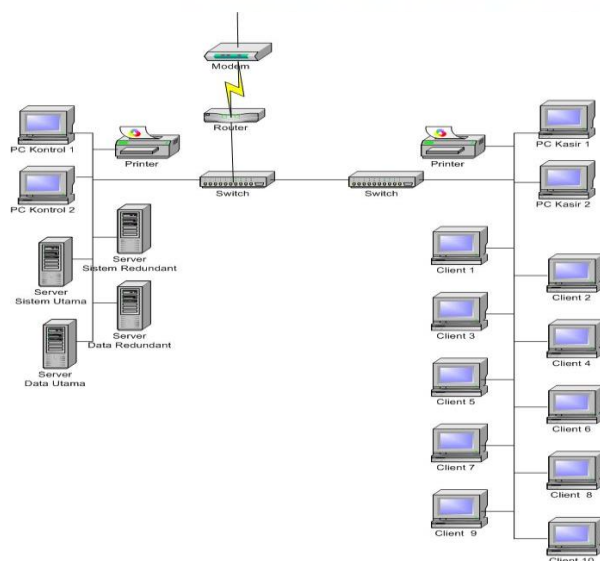
Untuk menayangkan video iklan dari suatu iklan baru yang mulai populer tidaklah mudah. Pertama-tama, perusahaan harus menunggu video iklan resmi keluar di pasaran, sedangkan biasanya video tersebut baru dirilis saat sebuah iklan memasuki akhir kepopulerannya. Tentu saja hal ini tidak menguntungkan bagi perusahaan karena video iklan-iklan merupakan modal utama bagi perusahaan yang bergerak di bidang penayangan video iklan. Bayangkan penghasilan yang hilang selama proses menunggu tersebut. Namun, masih ada kendala lainnya yaitu, periode pendistribusian video yang cukup lama. Prosedur pendistribusian yang menggunakan jasa kurir mengakibatkan setiap master video iklan yang selesai diproduksi tidak mungkin langsung dikirim dan disebar ke semua cabang karena akan memakan biaya yang sangat banyak. Dan berdasarkan dengan tahapan diatas, dapat dilihat bahwa periode pengiriman iklan ditentukan oleh jumlah master video iklan yang sudah selesai diproduksi. Hal ini mengakibatkan pelanggan merasa bosan menunggu dan saat akhirnya mereka dapat menikmati iklan-iklan tersebut, sebagian besar sudah tidak populer lagi dan sudah muncul iklan-iklan baru yang lebih menarik dan populer. Selain itu, juga mengakibatkan tidak serentak pembaharuan koleksi iklan baru di setiap cabang. Dan kadangkala, ada cabang yang malah terlupakan. Dari prosedur penambahan video iklan baru dan transaksi keuangan terjadi hal-hal sebagai berikut: a) Tidak serentak video iklan baru diterima oleh tiap cabang, b) Menurut hasil survei di beberapa cabang di luar daerah, koleksi video iklan selalu terlambat diperbaharui bahkan terkadang terlupakan, c) *Update* video iklan baru dapat didistribusikan ke setiap cabang setelah satu minggu pengumpulan video-video iklan baru sehingga terdapat rentang waktu yang cukup lama untuk penambahan video iklan baru. d) Laporan omset per hari yang dikirimkan melalui SMS

rentan terhadap manipulasi, e) Inspeksi terhadap laporan keuangan yang menyita waktu dan tenaga.

Dari hasil analisis di atas dapat dilihat bahwa perusahaan menghadapi masalah yang cukup kritis, yaitu: a) Proses distribusi video iklan baru dari kantor pusat ke masing-masing cabang, b) Proses pelaporan omset dan keuangan. Dari permasalahan yang ditemukan, diusulkan untuk membuat sebuah jaringan komputer yang dapat menghubungkan antar cabang sehingga informasi yang dibutuhkan dapat sampai dengan lebih cepat dan akurat. Ukuran file yang akan di transfer dari pusat ke cabang cukup besar yakni sekitar 41.325 KB. Setelah dilakukan wawancara dengan pihak perusahaan, ingin diciptakannya sebuah jaringan komputer yang memiliki lebar pita (*bandwith*) yang besar dan koneksi internet yang bersifat *connection-oriented*. Dengan pertimbangan diatas, maka akan dirancang sebuah jaringan komputer dengan teknologi Frame Relay.

Dengan adanya teknologi Frame Relay, diharapkan kedua masalah utama yang telah disebutkan di atas dapat diatasi dengan cara-cara sebagai berikut: a) Proses distribusi video iklan baru dari kantor pusat ke masing-masing cabang dapat diatasi dengan mentransfer setiap file video iklan baru ke setiap cabang di seluruh Indonesia pada hari video iklan tersebut selesai diproduksi, b) Proses pelaporan omset dan keuangan dapat diatasi dengan memprogram setiap komputer kasir di setiap cabang agar dapat mengirimkan setiap transaksi keuangan yang terjadi secara otomatis ke kantor pusat.

Berikut ini diberikan usulan implementasi jaringan komputer secara online dengan topologi jaringan komputer seperti pada gambar 4 di bawah ini



Gambar 4. Usulan Topologi Jaringan Komputer Dengan Sistem Baru Pada Setiap Cabang

Pada Gambar 4, diajukan *hierarchical topology* sebagai usulan topologi pertama yang terstruktur secara regional. Struktur topologi dapat dijelaskan sebagai berikut: a) Kantor pusat di Jakarta membawahi cabang-cabang yang ditunjuk sebagai cabang pusat distribusi video iklan pada tingkat Propinsi b) Setiap cabang pusat tingkat Propinsi membawahi cabang-cabang pusat tingkat Kabupaten c) Setiap cabang pusat tingkat Kabupaten membawahi cabang-cabang pusat tingkat kotamadya di propinsi tersebut, d) Setiap cabang pusat tingkat kotamadya membawahi cabang-cabang lainnya di kabupaten/kotamadya tersebut, e) Untuk Jakarta, cabang pusat tingkat kotamadya tidak berada di bawah cabang pusat tingkat propinsi tetapi langsung di bawah kantor pusat. Pengecualian ini dibuat mengingat Jakarta sebagai ibukota dan juga pusat hiburan negara. Oleh karena itu, Jakarta dipastikan memiliki jumlah cabang yang cukup banyak.

#### IV. EVALUASI PENGUJIAN SISTEM

Aturan proses pendistribusian pada setiap titik pada usulan topologi: a) Distribusi dilakukan secara otomatis dengan bantuan aplikasi yang akan diinstalasikan pada setiap titik sehingga setiap pengiriman tidak perlu didahului oleh konfirmasi manual. Hal ini akan mempersingkat waktu, b) Setiap *leaf* (titik yang tidak memiliki anak) memiliki nilai *flag* 1. Setiap titik yang bukan *leaf* akan memiliki nilai *flag* sebesar jumlah total nilai *flag* yang dimiliki oleh anak-anaknya ditambah 1, c) Setiap titik pada suatu tingkat akan mengirimkan file video iklan ke titik pada tingkat selanjutnya berdasarkan urutan nilai *flag* mulai dari nilai yang paling besar (*descending*).

Tabel distribusi berikut menunjukkan keadaan setiap cabang dalam lima siklus distribusi video iklan. Satu siklus distribusi adalah waktu yang diperlukan suatu titik untuk mengirimkan satu file video iklan dengan ukuran rata-rata 41.325 KB. Tanda “√” pada Tabel siklus distribusi video iklan baru menunjukkan bahwa cabang telah menerima video iklan baru pada siklus yang dimaksud. Dari tabel terlihat bahwa semua cabang dapat memperoleh file yang dikirim oleh kantor pusat dalam lima kali pengiriman.

Tabel 1 Siklus distribusi video iklan baru

Cabang		Siklus distribusi				
		I	II	III	IV	V
DKI Jakarta	Kebayoran baru				√	√
	Melawai		√	√	√	√
	Tebet			√	√	√
	Sunter	√	√	√	√	√
	Tanah Abang			√	√	√
	Cileduk				√	√
	Kebon Jeruk					√
	Kosambi			√	√	√
Jawa	Bandung		√	√	√	√



Cabang		Siklus distribusi				
		I	II	III	IV	V
	Semarang			√	√	√
	Cirebon				√	√
	Malang				√	√
	Surabaya				√	√
Sumatra	Medan			√	√	√
	Padang				√	√

Berdasarkan informasi di atas, diperoleh persamaan dengan mengabaikan waktu propagasi:

$$t = s / B \quad (1)$$

Keterangan:

t = waktu yang diperlukan per distribusi satu video iklan (detik),

s = ukuran rata-rata satu file video iklan (KB),

B = kecepatan akses minimum (*Committed Information Rate / CIR*) (Kbps).

Tabel 2 Usulan kecepatan akses

V. TITIK	Kecepatan Akses (Kbps)	CIR (Kbps)
Kantor pusat	128	64
Cabang	64	32

Perhitungan:

$$\begin{aligned} t &= 41.325 * 8 / 32 \\ &= 10331,25 \text{ detik} \\ &= 2 \text{ jam } 52 \text{ menit atau dibulatkan } \approx 3 \text{ jam} \end{aligned}$$

Jadi dengan menggunakan Tabel 2 setiap cabang memerlukan waktu lebih kurang 3 jam untuk mengirimkan satu file video iklan ke cabang lainnya.

Perhitungan waktu total:

$$t_{\text{total}} = 5 * 3 \text{ jam} = 15 \text{ jam}$$

Tabel 2 menunjukkan juga waktu yang dibutuhkan seluruh jaringan untuk menyebarkan file video iklan ke setiap titik adalah 15 jam. Keunggulan yang dimiliki topologi di atas adalah: a) Memiliki struktur kewenangan yang jelas -- Dengan bentuk topologi yang terstruktur secara regional dapat diterapkan suatu struktur kewenangan yang jelas. Struktur kewenangan ini dapat mendefinisikan dengan jelas tanggung jawab masing-masing cabang dalam distribusi video iklan baru sehingga dapat dilakukan pengendalian aliran data., b) Mendukung pengembangan jaringan -- Bila ada penambahan cabang baru di masa yang akan datang, posisi cabang tersebut dapat dengan mudah ditentukan. Dengan demikian, tanggung jawab cabang tersebut dalam

distribusi video iklan secara mudah dapat didefinisikan. Selain itu, penambahan cabang baru tidak akan menyebabkan perubahan topologi secara signifikan.

## V. SIMPULAN

Kesimpulan dari penelitian ini antara lain: a) Jaringan komputer menggunakan Frame Relay lebih dapat dipercaya karena telah menggunakan teknologi *packet-switched* sehingga jika koneksi terputus, tidak perlu mengirim ulang bagian dari file yang telah terkirim. Selain itu *bandwidth* pada koneksi Frame Relay terjamin dengan adanya *Committed Information Rate (CIR)* yang stabil oleh Penyedia layanan internet (*Internet Service Provider*) b) Setelah dievaluasi dengan suatu software simulasi jaringan, terbukti bahwa pengembangan jaringan komputer antarcabang pada perusahaan dapat mengatasi masalah waktu distribusi file video iklan dan mempermudah pelaporan omset dan keuangan.

Untuk pengembangan selanjutnya, penelitian dapat dikembangkan dengan beberapa saran berikut ini: a) Dalam pengembangan jaringan selanjutnya, dapat dirancang sebuah aplikasi untuk mengotomatisasi pengiriman file video iklan antarcabang. Hal ini dapat mengurangi tenaga dan biaya untuk operator, juga dapat meminimalisasi *delay time*, b) merancang suatu software untuk mengirimkan rincian transaksi yang terjadi di setiap cabang ke kantor pusat secara *real time* serta database di kantor pusat untuk menampung baik semua koleksi file video iklan yang dimiliki perusahaan maupun data transaksi yang terjadi setiap harinya, c) Untuk keamanan jaringan yang telah dirancang sebaiknya dipasang *firewall* yang diletakkan di antara *router* Jakarta dan *modem Frame Relay* untuk melindungi dari serangan yang berasal dari luar.

## DAFTAR PUSTAKA

- [1] Roden, Robert J. and Deborah Tayler, "Frame Relay Networks", Digital Technical Journal Vol. 5 No. 1, Winter 1993
- [2] Pressman, Roger S. And Bruce R Maxim., *Software Engineering: A Practitioner's Approach, 8th edition.* McGraw-Hill, New York, 2015
- [3] M.Ikhsan dan Y.Syahfitri, "Memahami Jaringan Komputer untuk Membangun Local Area Network (LAN)", Jurnal SAINTIKOM Vol 7/No. 2, hal. 336, Agustus 2009
- [4] Purbo, Onno W., 1997, *Frame Relay Service*, <http://onno.vlsm.org/v11/ref-ind-1/physical/frame-relay-1997.rtf> diakses 20 Mei 2011
- [5] Zwayen, Shahad H. And Mustapha B.Ibrahim, "Evaluating the Performance of MPLS and Frame-Relay using OPNET Modeler", International Journal of Computer Applications (0975-887) Vol 108 – No 12, December 2014
- [6] Stallings, William., "Data and Computer Communication, 10th edition. Pearson USA, 2014

# Perbandingan Algoritma Horspool dan Algoritma Zhu-Takaoka dalam Pencarian *String* Berbasis Desktop

Adhi Kusnadi<sup>1</sup>, Abraham Khrisnandi Wicaksono<sup>2</sup>

Fakultas Teknologi Informasi dan Komunikasi, Universitas Multimedia Nusantara, Gading Serpong, Indonesia  
adhi.kusnadi@umn.ac.id<sup>1</sup>, lincoln.abra92@gmail.com<sup>2</sup>

Diterima 14 Maret 2017

Disetujui 12 Juni 2017

**Abstract**— *String searching is the search process by using the index to find text that can help in information retrieval systems. Continuing previous research, this study uses an algorithm Horspool and Zhu-Takaoka to find the performance of each of these algorithms in the search for a pattern in the text. So they make a useful desktop-based application to measure performance of both algorithm, particularly the time required to perform string searching process. Using the method of prototyping and Microsoft Visual Studio with C# programming language implementation. Result obtained from this application is the number of words found, and the processing time of each algorithm. From this study, the Horspool algorithm is 19,82845 percent faster in first test with the pattern "swan" in a text file 50 multiples 1000 words and 15.9442 percent in the second trial using text files 70000 words with different pattern than the number of characters Zhu algorithm -Takaoka in the process of searching string.*

**Index Terms**-*String searching, Horspool, Zhu-Takaoka, Microsoft Visual Studio, application, processing time.*

## I. PENDAHULUAN

Pada zaman sekarang ini penggunaan komputer sudah merakyat dan hampir selalu digunakan untuk menjalankan berbagai aktivitas manusia. Diantara banyak fungsi komputer yang digunakan oleh manusia adalah pencarian data serta pengurutan data [1]. Perkembangan teknologi informasi dan komunikasi berdampak luas dalam kehidupan manusia. Semenjak dikembangkannya komputer pada pertengahan abad ke-20, peradaban manusia memulai babak yang baru. Teknologi Informasi dan Komunikasi membawa manusia pada era baru, era dimana percepatan perubahan kebudayaan sangatlah dramatis. Suatu era dimana informasi menjadi hal yang sangat penting bagi kehidupan manusia. Era ini kemudian dinamakan Era Informasi [2]. Oleh karena itu teknologi informasi sangat penting dan berguna di setiap aspek kehidupan bagi manusia.

*String searching* adalah komponen yang penting dari berbagai macam masalah, seperti *text editing*, data *retrieval* dan *symbol manipulation*. Walaupun penggunaan indeks digunakan untuk mencari teks,

*string searching* dapat membantu dalam sistem pencarian informasi. Sebagai contoh, digunakan untuk mencari persamaan yang potensial dalam pencarian teks atau untuk mencari istilah yang akan ditampilkan melalui perangkat output [3]. *String searching* merupakan sarana untuk mengembangkan sistem pencarian informasi.

Permasalahan pencocokan *string* (*string matching*) merupakan permasalahan yang sangat terkenal dalam dunia informatika. Contoh implementasi dari permasalahan pencocokan *string* adalah pada pencocokan sebuah *string* pada Microsoft Word, *string editor*, atau dalam kasus yang lebih besar lagi, yaitu pencocokan *website* dengan memasukkan kata-kata kunci sebagaimana yang telah diimplementasikan pada *search engine*, seperti Yahoo atau Google [4]. Pencocokan *string* (*string matching*) telah menjadi kebutuhan yang digunakan untuk mencari informasi melalui berbagai aplikasi.

Pada penelitian sebelumnya yang dilakukan oleh Vina Sagita disimpulkan bahwa algoritma Boyer-Moore adalah algoritma yang paling cepat dalam pencarian *string* [5] dan memberikan saran untuk dapat dilakukan penelitian kembali untuk mencari kecepatan dari tiap-tiap algoritma pencarian *string* dan menemukan algoritma yang dapat melakukan pencarian *string* paling cepat.

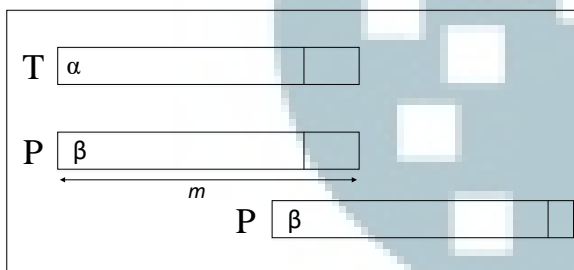
Dengan mengetahui pentingnya pencarian *string* di dalam pencarian informasi, penelitian ini bertujuan untuk mengetahui algoritma manakah yang paling cepat dalam mencari *string* yang berguna untuk mengoptimalkan sistem pencarian informasi melalui sarana berupa teks secara lebih cepat dan akurat. Dalam penelitian ini, dilakukan percobaan untuk membandingkan kecepatan dari algoritma Horspool dan Zhu-Takaoka dalam kemampuan pencarian *string* yang diimplementasikan ke dalam sebuah aplikasi berbasis desktop.

II. ALGORITMA DASAR

A. Algoritma Boyer-Moore

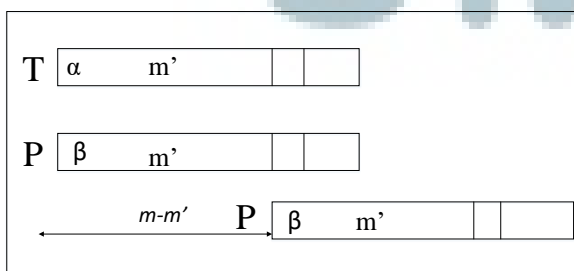
Pada tahun 1977, R.S Boyer dan J.S Moore merancang algoritma pencarian *string* linear. Proses komparasi pattern dengan *text* yang dicari dilakukan dari kanan ke kiri. Algoritma Boyer-Moore sangat populer dan paling sering digunakan dalam algoritma pencocokan *string*. Algoritma Boyer-Moore membangun dua tabel *preprocessing* yaitu *Boyer-Moore bad characters* (BMbc) yang sering disebut pergeseran ketidakcocokan dan *Boyer-Moore good suffix* (BMgs) yang disebut pergeseran kecocokan. Pergeseran pattern dilakukan berdasarkan nilai dari tabel BMbc dan BMgs [6].

Pergeseran BMbc memiliki dua kasus pergeseran. Kasus pertama adalah jika ketidakcocokan ada pada karakter pertama dari pattern P (posisi karakter ke  $m$ th) dengan karakter dari *text* T yang dicocokkan. Selain itu ketidakcocokan karakter dari *text* T juga tidak muncul di pattern P sehingga pergeseran pattern sebanyak panjang pattern  $m$  yang ditunjukkan pada Gambar 1 [6].



Gambar 1 Pergeseran BMbc Kasus Pertama

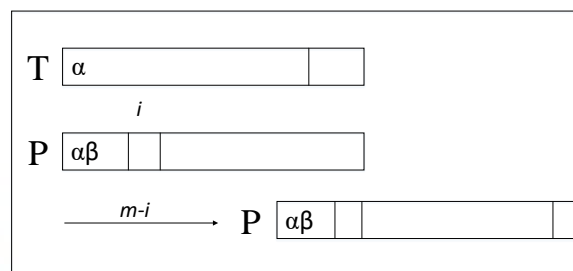
Sedangkan untuk kasus kedua jika karakter terakhir  $m'$  dari pattern P sesuai dengan karakter  $m'$  yang berkorespondensi *text* T kemudian terjadi ketidakcocokan setelah karakter  $m'$  kemudian pattern digeser sebanyak  $(m - m')$  yang ditunjukkan pada Gambar 2 [6].



Gambar 2 Pergeseran BMbc Kasus Kedua

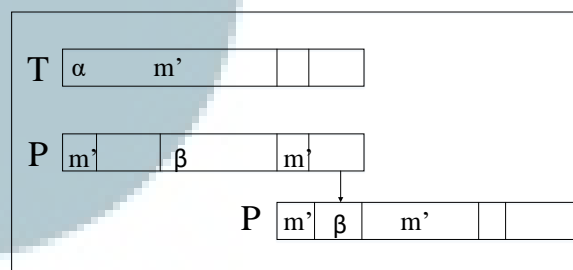
Untuk pergeseran BMgs memiliki dua kasus dalam melakukan pergeseran. Kasus pertama adalah jika ketidakcocokan ada di karakter pertama dari pattern P (posisi ke  $m$ th) dengan karakter yang

berkorespondensi dengan *text* T tetapi ketidakcocokan karakter dari *text* T juga muncul di pattern P pada posisi ke  $i$ th kemudian pergeseran pattern dilakukan sebanyak  $(m-i)$ th. Di mana,  $i$  adalah posisi dari karakter yang cocok di pattern P yang ditunjukkan pada Gambar 3 [6].



Gambar 3 Pergeseran BMgs Kasus Pertama

Sedangkan untuk kasus kedua dalam pergeseran BMgs adalah jika karakter terakhir  $m'$  dari pattern P tidak cocok dengan karakter ke  $m'$  dari *text* T tetapi karakter ke  $m'$  terulang kembali di pattern P maka pergeseran pattern dilakukan dengan mensejajarkan karakter yang sesuai dengan *text* dan pattern yang ditunjukkan pada Gambar 4 [6].



Gambar 4 Pergeseran BMgs Kasus Kedua

B. Algoritma Horspool

Algoritma Horspool adalah modifikasi dari algoritma Boyer-Moore dengan sedikit perubahan. Tidak seperti algoritma Boyer-Moore, algoritma Horspool hanya menggunakan satu tabel (*bad character shift*) dimana algoritma Boyer-Moore menggunakan dua tabel: *bad character shift* dan *good suffix shift* [7].

Algoritma Horspool mencari pattern dari kiri ke kanan dan untuk *shift value* berdasarkan ukuran dari pattern yang dicari dalam *bad character shift* tabel. Algoritma Horspool tidak efisien untuk kata yang pendek. Tetapi, jika *text* yang panjang dibandingkan dengan pattern yang dicari seperti yang terjadi dengan tabel ASCII dan pencarian kata yang ada di *text* editor, algoritma Horspool menjadi sangat berguna. Algoritma Horspool hanya menggunakan pergeseran *bad-character* pada bagian kanan karakter [8].

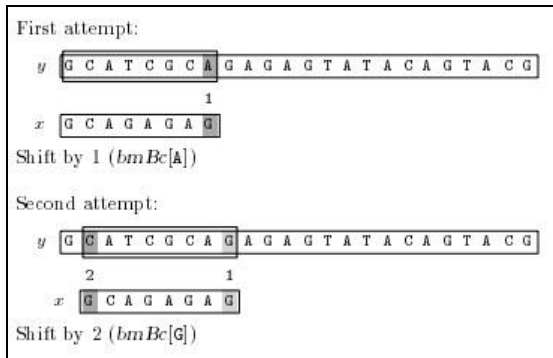
Sebagai contoh terdapat *text* dengan karakter GCATCGCAGAGAGTATACAGTACG kemudian

dengan pattern yaitu GCAGAGAG. Pada algoritma horspool dilakukan subproses dengan menghitung tabel *bad character shift* dari algoritma Boyer-Moore.

<i>a</i>	A	C	G	T
<i>bmBc[a]</i>	1	6	2	8

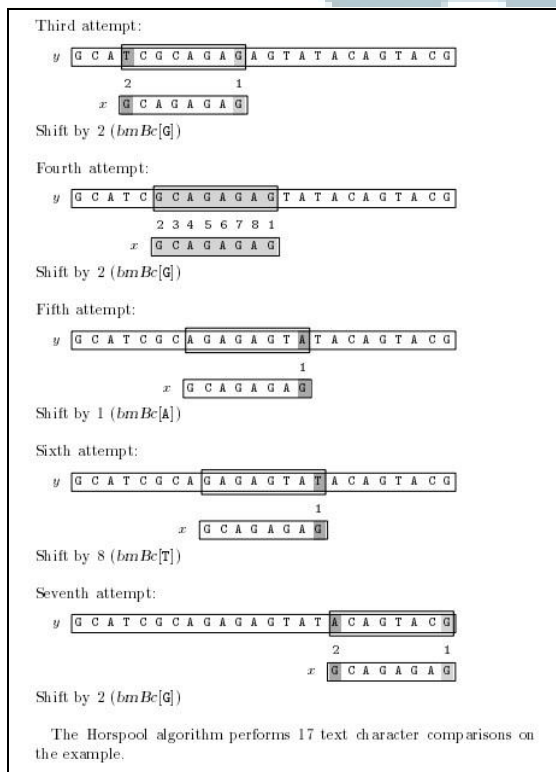
Gambar 5. *Bad character shift* Algoritma Horspool[8]

Langkah-langkah yang dilakukan seperti pada gambar berikut :



Gambar 6 *Searching Phase* Algoritma Horspool [8]

Kemudian dilanjutkan seperti pada gambar dibawah berikut :



Gambar 7. *Searching Phase* Algoritma Horspool [8]

Pada Gambar 6 dan 7 menunjukkan langkah-langkah pencocokan *string* dari algoritma Horspool pergeseran dilakukan menggunakan *bad character shift* dari karakter paling kanan dari *text* y yang dicocokkan dengan pattern x.

C. Algoritma Zhu-Takaoka

Algoritma Zhu-Takaoka diciptakan oleh R.F.Zhu dan T.Takaoka pada tahun 1989. Algoritma ini menggunakan dua *text* karakter untuk menghitung pergeseran karakter berdasarkan *bad-character*. Untuk mencari ketidakcocokan atau mencari seluruh *text* menggunakan *preprocessing hashing*. Hal ini efektif untuk pencarian *string* dua dimensi [9].

Algoritma Zhu-Takaoka menjelaskan bahwa pencarian pattern dilakukan mulai dari tiap baris mulai dari baris ke 0 dan berakhir pada baris ke  $n1 - m1$ . Zhu dan Takaoka merancang sebuah algoritma yang dapat bekerja menggunakan pergeseran dengan *bad-character* untuk dua karakter *text* secara berturut-turut. Selama fase pencarian pencocokan karakter dilakukan dari kanan ke kiri dan ketika proses berada di posisi  $y[j..j + m-1]$  dan ketidakcocokan terjadi diantara  $x[m-k]$  dan  $y[j+m-k]$  ketika  $x[m - k+1..m-1] = y[j+m-k+1..j+m-1]$  pergeseran dilakukan dengan pergeseran *bad-character* untuk *text* karakter  $y[j+m-2]$  dan  $y[j+m-1]$ . Tabel good-suffix dari algoritma Boyer-Moore juga digunakan untuk menghitung pergeseran [8].

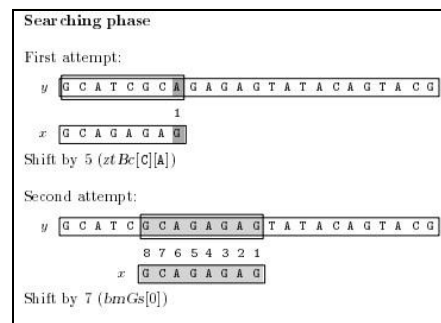
Pada awal proses algoritma Zhu-Takaoka melakukan *preprocessing* untuk menghitung *bad character shift* dan *good suffix* dari algoritma Boyer-Moore.

<i>zBc</i>	A	C	G	T
A	8	8	2	8
C	5	8	7	8
G	1	6	7	8
T	8	8	7	8

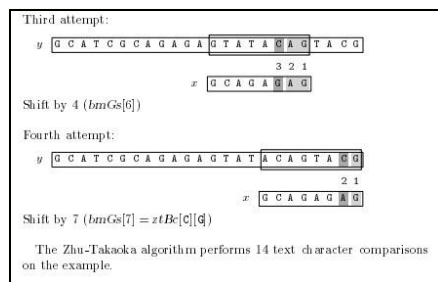
  

<i>i</i>	0	1	2	3	4	5	6	7
<i>x[i]</i>	G	C	A	G	A	G	A	G
<i>bmGs[i]</i>	7	7	7	2	7	4	7	1

Gambar 8. *Bad character shift* dan *Good suffix Shift* Zhu-Takaoka



Gambar 9. Searching Phase Algoritma Zhu-Takaoka



Gambar 10. Searching Phase Algoritma Zhu-Takaoka

Pada Gambar 9 dan Gambar 10 merupakan langkah-langkah pencarian *string* dari algoritma Zhu-Takaoka jika terjadi ketidakcocokan maka akan dilakukan pergeseran berdasarkan nilai terbesar antara  $bmGs[i]$  dengan  $ztBc[y[j+m-2]][y[j+m-1]]$ . Sedangkan jika ditemukan kecocokan maka pergeseran dilakukan dengan nilai  $i$  dari  $bmGs$  pada posisi 0 atau nilai  $i=0$  [8].

### III. PENELITIAN DAN HASIL UJI COBA

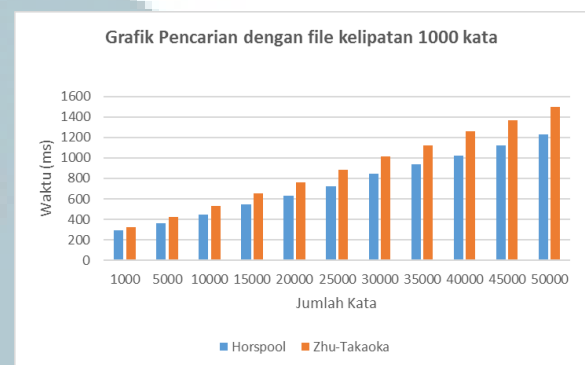
Aplikasi yang telah dibuat mulai dijalankan dengan berbagai *text file* untuk membandingkan kecepatan dari tiap algoritma. Data-data yang digunakan dalam proses pengujian ini adalah *text file* dengan ekstensi (\*.txt). *Text file* tersebut berisi kutipan kalimat dari novel fiksi *Twilight* versi bahasa inggris karangan dari Stephenie Meyer. Proses pengujian disediakan 50 file yang terdapat kalimat novel *Twilight*. Panjang kalimat tiap *text file* tersebut adalah kelipatan dari 1000 kata yang dimulai dari file yang terdiri 1000 kata hingga 50000 kata. Pattern yang dicari adalah kata "Swan" (gambar 11). Selain itu disediakan data yang terdiri dari 70000 kata untuk melakukan ujicoba dengan pattern yang lebih panjang dengan kata "Isabella Swan" (gambar 12).

Setelah data telah disiapkan, proses pengujian dilanjutkan dengan membuka aplikasi. Setelah aplikasi terbuka, *priority* aplikasi ini diubah menjadi *high* melalui Windows Task Manager. Tujuan mengubah *priority* yaitu untuk menyediakan *resource* yang lebih besar untuk aplikasi sehingga dapat melakukan proses pencarian secara cepat dan stabil.

Proses pengujian dilanjutkan dengan membuka *file text* dengan submenu *Open* yang ada di dalam menu File pada aplikasi. Setelah file dibuka dan pattern berupa kata "Swan", proses pencarian dilakukan dengan klik button "Cari Kata". Setiap klik button tersebut proses pencarian dari tiap algoritma dilakukan sebanyak 500 kali dengan tujuan untuk memperoleh hasil pencarian yang stabil. Pada gambar 11 dapat dilihat grafik pencarian dengan file kelipatan 1000 kata dengan pattern "swan". Setiap kelipatan kata dengan pattern "swan" dengan masing-masing algoritma diuji kecepatannya dalam mencari, kemudian hasilnya dirata-rata, didapat bahwa

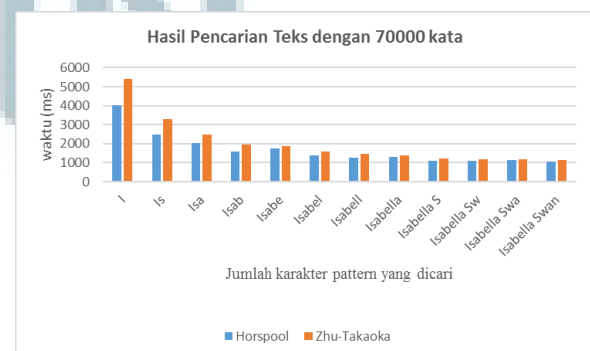
algoritma Horspool lebih cepat 19.82845 % pada uji coba pertama menggunakan 50 file *text file* kelipatan 1000 kata dengan pattern yang sama. Kemudian dilakukan percobaan dengan pattern yang berbeda beda (gambar 12). Dengan *text file* 7000 kata dan 12 pattern dihasilkan rata-rata kecepatan 15.9442 %.

Setelah proses pengujian selesai, proses validasi dilakukan untuk mengetahui apakah algoritma Horspool maupun algoritma Zhu-Takaoka valid dalam melakukan proses pencarian *string*. Proses validasi yang dilakukan adalah membandingkan jumlah kata yang ditemukan oleh aplikasi ini dapat menggunakan aplikasi lain yang sejenis atau yang berfungsi sama atau juga dapat menggunakan fitur *find* yang ada di aplikasi microsoft word. Dengan hasil yang sama antara aplikasi perbandingan dan microsoft word.



Gambar 11. Grafik Pencarian Dengan File Kelipatan 1000 Kata Dengan Pattern "Swan"

Perbandingan waktu pencarian algoritma Horspool dan algoritma Zhu-Takaoka yang telah di uji coba. Grafik tersebut merupakan perbandingan waktu proses dengan jumlah kata yang ada dalam *text file*.



Gambar 12. Grafik Pencarian Dengan File 70000 Kata Dengan Berbagai Jumlah Pattern

Dari gambar tersebut menunjukkan grafik perbandingan antara waktu dengan jumlah pattern yang dicari. Dalam grafik tersebut algoritma Horspool lebih cepat daripada algoritma Zhu-Takaoka dalam

melakukan proses pencarian *string*. Perlu ada penelitian lanjutan mengapa algoritma Horspool lebih cepat dari algoritma Zhu-Takaoka dan menganalisa hasil percobaan yang telah dilakukan.

#### IV. SIMPULAN

Berdasarkan tujuan penelitian untuk mengetahui perbandingan performa antara algoritma Horspool dan algoritma Zhu-Takaoka, dibuatlah aplikasi perbandingan algoritma Horspool dan algoritma Zhu-Takaoka yang dapat mengukur performa antara kedua algoritma tersebut. Performa yang diukur adalah waktu dari algoritma Horspool dan algoritma Zhu-Takaoka dalam melakukan pencarian pattern di dalam suatu *text* file.

Dari hasil pengujian aplikasi, bahwa algoritma Horspool lebih cepat 19.82845 persen pada uji coba pertama menggunakan 50 file *text* file kelipatan 1000 kata dengan pattern yang sama dan 15.9442 persen pada ujicoba kedua dengan file *text* 7000 kata. Sehingga dapat disimpulkan algoritma Horspool lebih cepat daripada algoritma Zhu-Takaoka.

#### DAFTAR PUSTAKA

- [1] Handoyo, Ria Arini. 2004. "Perbandingan Waktu Proses Pencarian Data Antara Algoritma Raita dengan Algoritma Zhu-Takaoka" Dalam <http://karyailmiah.tarumanagara.ac.id/index.php/S1TI/article/view/2315> Diakses tanggal 29 Agustus 2014
- [2] Munggaran, Abdi Halim. 2009. "Perilaku Pencarian Informasi Mahasiswa yang Memanfaatkan Layanan Search Engines dalam Menyusun Skripsi: Studi Kasus Mahasiswa S1 Program Studi Ilmu Perpustakaan Universitas Indonesia" Dalam <http://lib.ui.ac.id/file?file=digital/123786-RB13A44p-Perilaku%20pencarian-Pendahuluan.pdf> Diakses tanggal 26 Agustus 2014.
- [3] Baeza-Yates, Ricardo A. Tanpa Tahun. "String Searching Algorithms" Dalam <http://orion.lcg.ufrj.br/Dr.Dobbs/books/book5/chap10.htm> Diakses tanggal 25 Agustus 2014
- [4] Hartoyo, Eko Gunocipto, Yus Gias Vembrina, dkk. 2010. "Analisis Algoritma Pencarian *String* (*String Matching*)" Dalam <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/Makalah/MakalahStmik10.pdf> Diakses tanggal 25 November 2014
- [5] Sagita, Vina. 2012. "Studi Perbandingan Implementasi Algoritma Boyer-Moore, Turbo Boyer-Moore, dan Tuned Boyer-Moore dalam Pencarian *String*" skripsi UMN.
- [6] Bhandari, Jamnua dan Anil Kumar. 2014. "String Matching Rules Used By Variants of Boyer-Moore Algorithm" Dalam <http://www.rroij.com/open-access/string-matching-rules-used-by-variants-of-boyermooore-algorithm-8-11.pdf?aid=37835> Diakses tanggal 17 Agustus 2015
- [7] Lovis, C dan R.H.Baud. 2000. "Fast Exact *String* Pattern-matching Algorithms Adapted to the Characteristics of the Medical Language"
- [8] Charras, Christian dan Thierry Lecroq. 2004. "Handbook of Exact *String*-Matching Algorithms" Dalam <http://www-igm.univ-mlv.fr/~lecroq/string.pdf> Diakses tanggal : 8 September 2014
- [9] Prasad J.C1 & K.S.M.Panicker. 2010. "String Searching Algorithm Implementation-Performance Study with Two Cluster Configuration" Dalam <http://www.csjournals.com/IJCS/2/56.pdf> Diakses tanggal 8 September 2014

# Penelitian Mengenai Metode Steganografi Least Significant Bit

Ivan Jonathan<sup>1</sup>, Albert Yeusiawan Haryono, Kevin Leonardi

Program Studi Teknik Informatika, Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia

Diterima 14 Maret 2017

Disetujui 5 Juni 2017

**Abstract** - In today's technological era, the concealment of sensitive information is the concern of many people. Because the information is often shared and discussed through a very commonly used communication medium. Steganography is one technique to hide a secret message into a file that has a larger size. In this paper, we will discuss the methods that can be used in steganography, especially the method of Least Significant Bit.

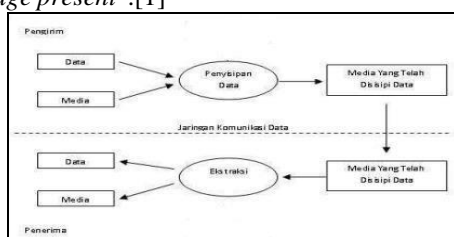
**Keywords:** Steganography, Data Hiding, Steganography Algorithm, Least Significant Bit.

## I. PENDAHULUAN

Pada era sekarang ini, hampir semua hal dikomunikasikan menggunakan teknologi. Mulai dari percakapan sehari-hari, sampai dengan informasi yang sensitif. Maka dari itu, masalah keamanan dalam transfer data menjadi perhatian banyak orang. Sampai saat ini teknik pengamanan informasi yang dipakai adalah enkripsi dan steganografi.

Steganografi adalah seni penyembunyian informasi atau pesan rahasia pada suatu media sehingga tidak terdeteksi oleh pihak lain. Kata steganografi berasal dari bahasa Yunani yaitu *steganos* yang berarti "tersembunyi" dan *graphein* yang berarti "menulis".

Markus Kahn mendefinisikan steganografi sebagai berikut, "Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to Cryptography, where the enemy is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of Steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second message present". [1]



Gambar 1: Proses Steganografi [2]

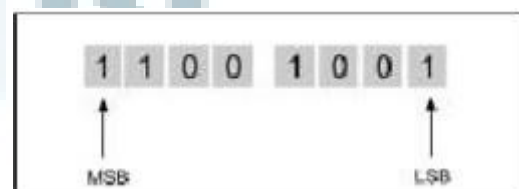
“Steganografi mempunyai dua proses utama yaitu embed/penyisipan dan ekstrak/pengungkapan seperti pada gambar 1. Proses penyisipan merupakan proses menyisipkan hidden object atau informasi/pesan yang akan disisipkan, ke dalam sebuah cover object atau media penampung, sehingga menghasilkan file baru yang telah tersisipi pesan didalamnya yang disebut dengan stego file. Sedangkan proses ekstrak merupakan proses pengembalian hidden object secara utuh setelah disisipkan ke dalam cover object.” [2]

## II. METODE *LEAST SIGNIFICANT BIT* (LSB)

LSB adalah algoritma sederhana yang menukar bit terkecil ke dalam beberapa byte media penyembunyiannya secara berurutan. [3]

*Digital image* pada komputer merupakan kumpulan dari angka-angka yang merepresentasikan *grid* dan titik-titik yang disebut *pixel*. Karena itu, digital image dapat dipakai menjadi media dalam penyembunyian pesan dengan metode LSB.

Berikut ini adalah contoh sebuah deretan angka biner dari angka decimal 201:



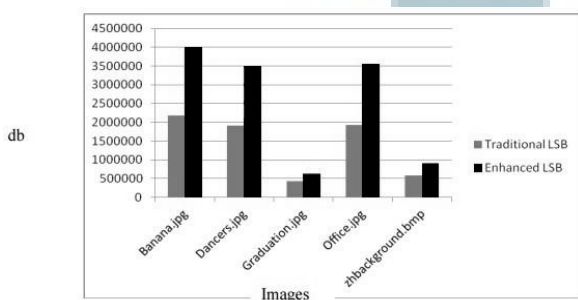
Gambar 2: Deretan Angka Biner dengan Digit LSB ([https://www.academia.edu/11764630/Steganografi\\_Metode\\_LSB](https://www.academia.edu/11764630/Steganografi_Metode_LSB))

Bit paling kanan dari sebuah deretan angka biner merupakan bit yang disebut Least Significant Bit (LSB). Apabila kita mengganti nilai bit LSB pada deret tersebut, maka akan dihasilkan deret "11001000". Deret tersebut bila dikonversikan ke dalam decimal bernilai 200. Perbedaan yang sangat kecil ini menyebabkan file media yang disisipi pesan di dalamnya tidak akan kelihatan jelas perbedaannya.

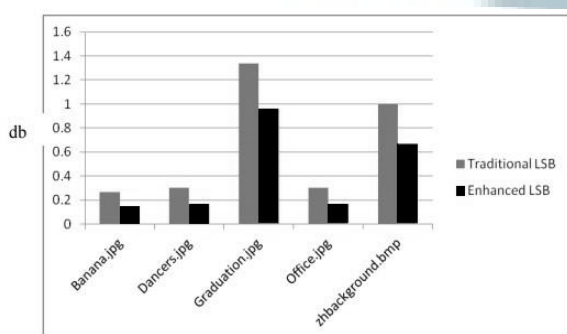
Dengan cara ini, LSB memanfaatkan fakta bahwa tingkat presisi dalam digital image jauh lebih tinggi daripada yang bisa dilihat oleh mata manusia.

### III. TINJAUAN PUSTAKA

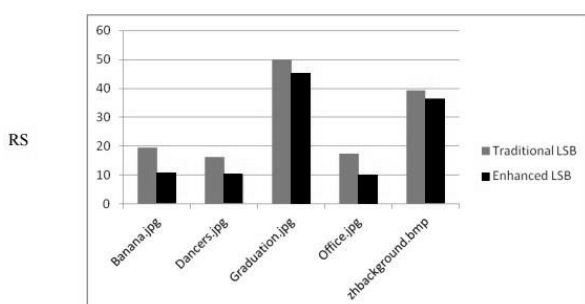
Gabriel dkk [3] mengajukan sebuah metode yang meningkatkan kualitas dari LSB. Metode tersebut menggunakan *linear congruential pseudo random number generator (LCG)* dengan sebuah *stego key* yang di-hash untuk memilih secara acak bit dari *cover image* yang akan ditukar dengan bit dari informasi rahasia. Hasil dari percobaan menunjukkan bahwa metode ini menghasilkan PSNR yang lebih tinggi, MSE yang lebih rendah, serta RS yang lebih rendah daripada metode LSB tradisional. Hal ini menunjukkan bahwa metode ini menghasilkan kualitas *stego image* yang lebih baik dengan penyimpangan dan noise yang lebih sedikit.



Gambar 3: Hasil PSNR (Db) Dari Percobaan Menggunakan Metode LSB Tradisional Dan *Enhanced LSB*



Gambar 4: Hasil MSE Dari Percobaan Menggunakan Metode LSB Tradisional Dan *Enhanced LSB*



Gambar 5: Hasil RS Dari Percobaan Menggunakan Metode LSB Tradisional Dan *Enhanced LSB*

Shahim dan Kattamanchi [4] mengajukan sebuah metode yang menggabungkan kriptografi dan steganografi untuk menyelesaikan masalah akses data yang tidak terotorisasi. Metode menggabungkan kriptografi dan steganografi ini diajukan karena penyembunyian data yang hanya menggunakan metode LSB masih tidak terlalu aman. Jadi, untuk menambah keamanan data digunakan kriptografi. Mula-mula pesan dienkripsi dengan menggunakan metode *transposition cipher*, kemudian pesan yang telah terenkripsi tersebut ditanamkan ke dalam sebuah digital image menggunakan metode LSB. Dengan metode ini, apabila seorang *attacker* berhasil mengekstrak data dari *stego image*, ia masih membutuhkan *decoding key* untuk mendekripsi data tersebut.

Cover image	Stego Image	Amount of data embedded	MSE %	PSNR (dB)	Amount of data extracted
clover (35 KB)	stegclover (35 KB)	4267 bytes	0.48	51.28	4267 bytes
flower (43 KB)	stegflower (43 KB)	4513 bytes	0.41	51.93	4513 bytes
bud (47 KB)	stegbud (47 KB)	5075 bytes	0.43	51.69	5075 bytes

Gambar 6: Tabel Nilai MSE Dan PSNR Dari Cover Image Dan Stego Image.

Champakamala dkk [5] mengajukan sebuah teknik baru dalam steganografi LSB yang merupakan improvisasi dari teknik *one bit LSB*. Pada teknik LSB Tradisional, kita membutuhkan 8 byte pixel untuk menyimpan 1 byte data, sedangkan pada teknik ini hanya membutuhkan 4 bytes pixel untuk menyimpan 1 byte data. Dalam teknik ini, data disisipkan pada 2 bit terakhir dari setiap pixel. Simulasi dari teknik ini dilakukan menggunakan MATLAB dengan empat langkah utama:

- Konversi gambar ke matrix
- Proses penyisipan data
- Konversi matrix kembali ke gambar
- Proses ekstraksi data

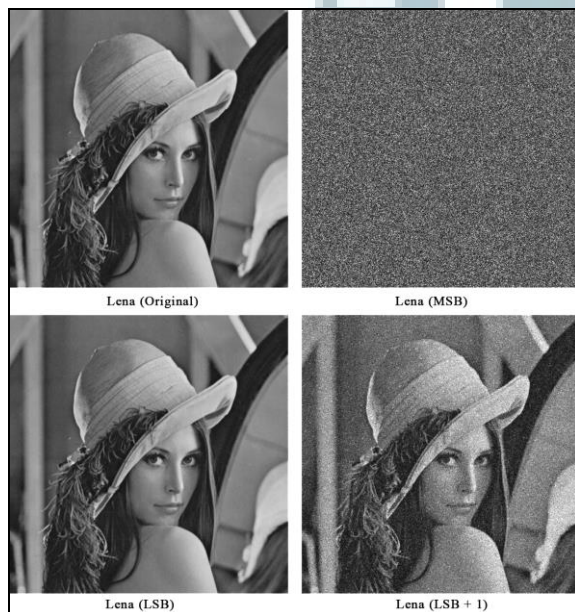
Setelah melakukan beberapa percobaan, dapat dikatakan bahwa teknik ini membantu dalam menyembunyikan data rahasia ke dalam *cover image* tanpa adanya distorsi.

Gies Masita dan Tri Ismardiko [6] mengajukan penggunaan metode LSB dalam pemrograman berbasis web menggunakan bahasa pemrograman PHP. Pemanfaatan metode LSB pada PHP dapat dilakukan dengan menggunakan *Stegger*, sebuah *class library open source* berlisensi. *Stegger* menukar bit terakhir dari setiap warna primer pada pixel. Jadi dengan *stegger* dapat dilakukan penyimpanan 3 bit



data di setiap pixelnya. Gies dan Tri juga menambahkan enkripsi pada data rahasia sebelum data tersebut disisipkan ke dalam *cover image* agar tercipta keamanan data yang lebih baik. Untuk melakukan enkripsi pada data, digunakan fungsi *secript* pada *stegger* yaitu dengan cara mengkombinasikan antar informasi rahasia dengan sebuah key.

Yudhi Andrian [3] memaparkan perbandingan antara metode LSB, LSB+1, dan MSB. Perbedaan terdasar dari ketiga metode ini adalah letak bit yang ditukar dengan bit dari pesan rahasia. Pada LSB bit yang disisipi data adalah bit terakhir (bit ke-8), pada LSB+1 bit yang disisipi data adalah bit ke-7, sedangkan pada MSB bit yang disisipi data adalah bit ke-1. Selain memaparkan perbedaan ketiga metode ini secara teori, Yudhi juga menambahkan perbedaan ketiganya dilihat dari hasil stego image dan nilai dari PSNR-nya.



Gambar 7: Citra Lena Original Dan Setelah Disisipkan Data.

Citra	MSE	PSNR
Lena	0,479	51,327

Gambar 8: Tabel MSE Dan PSNR Pada Metode LSB.

Citra	MSE	PSNR
Lena	1,905	45,332

Gambar 9: Tabel MSE Dan PSNR Pada Metode LSB+1.

Citra	MSE	PSNR
Lena	8185,701	9

Gambar 10: Tabel MSE Dan PSNR Pada Metode MSB.

Dari hasil stego image dan nilai PSNR-nya dapat dikatakan bahwa metode LSB lebih baik dari LSB+1 dan MSB. Tetapi, berdasarkan percobaan penghancuran pesan yang dilakukan oleh Yudhi dapat dikatakan bahwa metode LSB+1 dan MSB lebih tahan terhadap penghancuran pesan.

Metode	Pesan yang disisipkan	Hasil ekstraksi pesan
LSB	"Hancur"	yyyyyyyy
LSB+1	"Hancur"	"Hancur"
MSB	"Hancur"	"Hancur"

Gambar 11: Tabel Pengujian Tingkat Ketahanan Pesan Terhadap Penghancuran Pesan Menggunakan Metode LSB.

#### IV. SIMPULAN

Pada paper ini, dipaparkan hasil penelitian tentang metode steganografi LSB yang dilakukan dengan membandingkan beberapa paper yang telah dipublikasi. Masing-masing penulis memberikan metode mereka yang berbeda dengan yang lainnya. Beberapa mencoba meningkatkan keamanan data dan yang lainnya membandingkan beberapa metode.

Dari keseluruhan referensi, ditarik kesimpulan bahwa metode LSB pada steganografi menghasilkan *stego image* yang sangat mirip dengan *image original* yang digunakan sebagai *cover image*. Hal ini terjadi karena penyisipan data dengan metode LSB hanya mengganti bit terkecil dari pixel gambar, sehingga perbedaan *original image* dan *stego image* tidak dapat diketahui oleh mata manusia. Metode LSB juga memiliki kekurangan yaitu keamanan data masih kurang terjamin. Maka dari itu, sebaiknya dalam melakukan penyembunyian data rahasia kita sebaiknya steganografi dan kriptografi untuk meningkatkan keamanan data.

#### DAFTAR PUSTAKA

- [1] Aditi dan Sujit, "A Survey on Spread Spectrum Image Steganography Hiding Text in Digital Data", *International Journal for Scientific Research & Development*, Vol 3, No.5, 2015.
- [2] M. Maha Andar Pasaribu, "Perancangan Program Berbasis Mobile Dengan Menggunakan Metode Pixel Value Differencing dan Algoritma Rijndael", *Binus University*, 2014.
- [3] Gabriel, Stephen, dan Waweru, "An enhanced Least Significant Bit Steganographic Method for Information Hiding", *Journal of Information Engineering and Applications*, Vol 2, No.9, 2012.
- [4] Shahim dan Kattamanchi, "High Capacity data hiding using LSB Steganography and Encryption", *International Journal of Database Management Systems*, Vol 4, No.6, 2012.
- [5] Champakamala, Padmini, dan Radhika, "Least Significant Bit algorithm for image steganography", *International*

- Journal of Advanced Computer Technology, Vol 3, No.4, 2013.
- [6] Gies Masita dan Tri Ismardiko, "Pengamanan Pesan Steganografi dengan metode LSB Berlapis Enkripsi dalam PHP", Universitas Budi Luhur, [https://www.academia.edu/5306496/Pengamanan\\_Pesan\\_Steganografi\\_dengan\\_Metode\\_LSB\\_Berlapis\\_Enkripsi\\_dalam\\_PHP](https://www.academia.edu/5306496/Pengamanan_Pesan_Steganografi_dengan_Metode_LSB_Berlapis_Enkripsi_dalam_PHP)
- [7] Yudhi Andrian, "Perbandingan Metode LSB, LSB+1, dan MSB pada Steganografi Citra Digital", STMik Potensi Utama, [https://www.academia.edu/6024132/PERBANDINGAN\\_METODE\\_LSB\\_LSB\\_1\\_DAN\\_MSB\\_PADA\\_STEGANOGR\\_AFI\\_CITRA\\_DIGITAL](https://www.academia.edu/6024132/PERBANDINGAN_METODE_LSB_LSB_1_DAN_MSB_PADA_STEGANOGR_AFI_CITRA_DIGITAL).
- [8] Fahrijal, "Steganografi", Dilihat 12 April 2016 <https://www.academia.edu/6977142/Steganografi>.



# Aplikasi Pengendali Kamera DSLR Nirkabel Tipe *Low End* Berbasis Android

Endang Tirtana<sup>1</sup>, Eko Budi Setiawan<sup>2</sup>

Program Studi Teknik Informatika, Universitas Komputer Indonesia, Bandung, Indonesia  
endang7tirtana@gmail.com<sup>1</sup>, eko@email.unikom.ac.id<sup>2</sup>

Diterima 21 April 2016

Disetujui 5 Juni 2017

**Abstract**— *One of those beneficial innovations in DSLR camera it is able to connect with other mobile devices using wireless network. Its primary function is to control camera from distance and to instantly upload photos to social medial. But, not all DSLR cameras are equipped with this feature. Only the latest mid-end and high-end DSLR camera have the feature. In order to solve the previously mentioned issue, the researcher conducted a trial using Arduino and Bluetooth as the camera's wireless network. The result showed that Arduino did not have its own module and, as a result, users would need to add new modules if they wanted to add new features. This was considered ineffective and inefficient. Moreover, Bluetooth connection is rarely used nowadays due to the limited transfer rate and area coverage compare to using WiFi connection. In line with the aforementioned explanation, a research was held with the main objective to access and operate low-end DLSR camera using Raspberry Pi. The Raspberry Pi was operated using access point and controlled by smartphone.*

**Index Terms**— *Camera, DSLR, wireless, Raspberry Pi, Android*

## I. PENDAHULUAN

Perkembangan teknologi yang sangat dinamis pada kamera DSLR (*Digital Single Lens Reflex*) memicu para vendor untuk melakukan inovasi-inovasi yang berkaitan dengan kamera digital [1]. Salah satu fitur yang berguna saat ini adalah DSLR bisa terhubung dengan perangkat bergerak lainnya menggunakan jaringan nirkabel. Tujuannya agar bisa mengendalikan kamera dari jarak jauh dan bisa secara instan mengunggah foto-foto tersebut ke jejaring sosial [2]. Namun tidak semua kamera DSLR mempunyai fitur tersebut, hanya kamera DSLR *mid-end* sampai *high-end* edisi terbaru yang memilikinya [1].

Berdasarkan kuesioner yang peneliti berikan di bulan Oktober 2016 kepada komunitas fotografer di Kota Bandung yang memiliki kamera DSLR *low-end*, ditemukan beberapa kendala dalam pengoperasian dan pengambilan berkas-berkas di dalam kartu memori yang masih menggunakan kabel data atau melepas kartu memori. Permasalahan lainnya adalah ketika mengendalikan kamera DSLR menggunakan kabel data jaraknya hanya terbatas beberapa meter saja, sehingga mobilitas pengguna sangat terbatas. Masalah

lainnya ketika mengoperasikan kamera DSLR secara langsung sering terjadi guncangan sehingga menghasilkan gambar yang tidak jelas. Hal tersebut juga sering terjadi ketika mengambil obyek dengan *shutter speed* rendah meskipun menggunakan tripod [2].

Lusiana Citra Dewi [3] pernah melakukan penelitian mengenai perkembangan teknologi jaringan nirkabel. Lusiana membandingkan teknologi infrared, *bluetooth*, dan WiFi. Kesimpulan dari penelitian tersebut menerangkan bahwa perkembangan teknologi jaringan nirkabel yang sangat pesat menyebabkan teknologi transmisi nirkabel *infrared* dan *bluetooth* mulai ditinggalkan. Salah satu alasannya karena keterbatasan kecepatan transmisi [3]. Penelitian lainnya juga pernah dilakukan oleh Yuris Octa Primariadi menggunakan Arduino dan *bluetooth* berbasis Android. Android dapat mengendalikan kamera DSLR dan menggerakkan tripod sesuai dengan keinginan pengguna. Koneksi kamera dan tripod menggunakan dua buah modul bluetooth RN-42 dan HC-05 [4], sehingga kurang efisien karena harus berganti-ganti pengaturan. Modul tersebut merupakan alat tambahan karena tidak tertanam langsung di Arduino.

Raspberry Pi adalah salah satu *Single Board Computer* (SBC) yang cukup populer meskipun ada beberapa SBC yang lain seperti BeagleBone, Intel Galileo, PandaBoard, CubieBoard, dan lain-lain [5]. Generasi terbaru Raspberry Pi adalah Raspberry Pi 3 yang baru dirilis pada Februari 2016. Raspberry Pi 3 mengusung bentuk board dan interface yang sama dengan Raspberry Pi 2 dan Raspberry Pi 1, tetapi mengalami peningkatan pada *System on Chip* (SoC) yaitu 10 x lebih cepat dari Raspberry Pi 1 dan sudah terintegrasi dengan *wifi module* dan *low power bluetooth* [5].

Dengan demikian, aplikasi yang akan dibangun dalam penelitian ini diharapkan untuk dapat mengatasi permasalahan para fotografer yang memiliki kamera DSLR *Low End* sehingga dapat membantu untuk mempermudah pengambilan data secara manual serta dapat mengatasi permasalahan lain terkait adanya

keterbatasan jarak pengendalian kamera DSLR tipe *Low End*.

## II. TINJAUAN PUSTAKA

### A. Fotografi

Fotografi adalah media komunikasi antara fotografer dan penikmat foto. Supaya komunikasi tersebut dapat disampaikan dengan baik, *fotografer* harus bisa mengkomposisikan foto dengan baik. Komposisi dalam fotografi adalah cara menempelkan elemen-elemen visual dalam sebuah foto. *Fotografer* adalah suatu profesi yang menggeluti bidang fotografi.

### B. Android

Proyek Android dimulai pada awal tahun 2003 oleh sebuah perusahaan kecil bernama Android, Inc., sebelum istilah *smartphone* dikenal luas oleh pengguna saat ini yang memiliki layar sentuh besar, kecepatan internet yang tinggi, GPS, dan hal menyenangkan lainnya. Pada tahun 2007 perusahaan Android, Inc., diakuisisi oleh Google [6]. Pada bulan Oktober 2012, terdapat 700.000 aplikasi yang tersedia pada Android, dan sekitar 25 juta aplikasi telah diunduh dari *Google Play*. Sebuah survey pada bulan Mei-Juni 2013 menemukan bahwa Android adalah *platform* paling populer bagi pengembang, digunakan oleh 71% pengembang aplikasi perangkat bergerak dan pada 1 September 2013, 1 milyar perangkat telah diaktifkan [7].

Google menginginkan Android menjadi terbuka dan bebas, oleh karena itu sebagian besar kode Android dirilis di bawah sumber terbuka *Apache License*. Artinya siapapun yang akan menggunakan Android dapat melakukannya dengan mengunduh seluruh *source code* Android [8]. Para produsen *smartphone* yang menggunakan sistem operasi Android dapat menambahkan berbagai aplikasi tambahan agar dapat membedakan produk mereka dari produsen lainnya. Karena sifat Android yang terbuka banyak komunitas pengembang yang membangun aplikasi untuk Android. Tersedianya berbagai jenis aplikasi gratis di *Google Play Store* membuat masyarakat banyak yang beralih menggunakan *smartphone* berbasis Android.

### C. DSLR

Kamera *Digital Single Lens Reflex* (DSLR) adalah kamera digital yang menggunakan sistem cermin otomatis dan pentaprisma atau pentamirror untuk meneruskan cahaya dari lensa menuju *viewfinder*. *Viewfinder* adalah lubang kecil dibelakang kamera untuk melihat obyek foto [4]. Kamera DSLR memiliki kelebihan dalam hal kebebasan pengguna untuk merubah pengaturan kamera sesuai dengan keinginan. Fitur-fitur dan teknologi yang diterapkan pada kamera DSLR berbeda-beda pada setiap tipenya. Perbedaan teknologi yang diterapkan dalam setiap kamera DSLR dibedakan berdasarkan kelas kamera [1].

### D. Flask API

Flask API merupakan implementasi dari web API yang dapat dijelajahi menggunakan kerangka yang sudah disediakan Django REST. Django REST merupakan web *framework* sumber terbuka berbasis Python. Aplikasi web yang dibuat dengan Flask disimpan dalam satu berkas *.py*. Flask merupakan web *framework* yang sederhana namun dapat diperluas dengan beragam pustaka tambahan yang sesuai dengan kebutuhan penggunaannya. Flask API menyimpan perintah-perintah dari *gphoto library* dan *piggyphoto library*.

Pendefinisian abreviasi dan akronim dilakukan di awal naskah baik pada bab maupun pada abstraksi. Selanjutnya abreviasi dan abstraksi tidak perlu didefinisikan ulang. Abreviasi umum seperti IEEE tidak perlu didefinisikan. Sebaiknya tidak menggunakan bentuk abreviasi dalam penulisan judul naskah.

### E. Raspberry Pi

Raspberry Pi (RPI) adalah mini PC berukuran seperti kartu kredit yang dikembangkan oleh *Raspberry Pi Foundation*. Pada awalnya dibuat untuk proses pembelajaran komputer bagi para siswa di sekolah. Seiring perkembangan teknologi, sekarang RPI dapat digunakan untuk berbagai aplikasi, seperti *home automation*, web *server*, DNS *Server*, dan lain-lain. RPI pertama kali dirilis pada bulan Februari tahun 2012 dengan model B Rev. 1 dengan RAM sebesar 256 MB. Pada September 2012 keluar model B Rev. 2 dengan penambahan RAM menjadi 512 MB. Pada bulan Februari 2013 keluar model A tetapi dengan RAM 256 MB dan pada April 2014 keluar varian *Compute Module* dari Raspberry Pi [5].



Gambar 1. *Raspberry Pi 3*

### F. Java

Sejak pertama kali dirilis pada tahun 1996, Java menjadi bahasa pemrograman populer bagi pengembang di seluruh dunia [9]. Sekitar tahun 2012 9 juta pengembang aktif menggunakan bahasa pemrograman Java [9]. Java telah diterima secara luas oleh para pengembang sebagai bahasa alternatif untuk pemrograman. Bagi sebagian perusahaan banyak yang

menggunakan bahasa pemrograman Java untuk memenuhi kebutuhan pemrograman. Java banyak digunakan untuk mengimplementasikan aplikasi berbasis internet dan perangkat lunak yang digunakan untuk berkomunikasi melalui jaringan [10].

Salah satu alasan Java menjadi bahasa pemrograman yang terkenal karena independensi *platform*. Yang berarti program Java dapat berjalan di banyak jenis komputer. Sebuah program Java dapat berjalan pada setiap komputer dengan terlebih dahulu sudah terinstall *Java Runtime Environment (JRE)*. Program Java dapat berjalan di *Windows*, komputer *Macintosh*, *Unix*, dan komputer *Linux*, bahkan program Java dapat berjalan di telepon seluler [11].

### G. Python

*Python* merupakan bahasa pemrograman *open source* yang banyak digunakan untuk menangani beberapa jenis masalah dalam pemrograman. *Python* banyak digunakan untuk meningkatkan kualitas perangkat lunak, produktivitas pengembang, portabilitas program, dan integrasi komponen. *Python* digunakan oleh setidaknya ratusan ribu pengembang di seluruh dunia dalam bidang-bidang seperti *scripting internet*, pemrograman sistem, *user interface*, kustomisasi produk, pemrograman numerik, dan banyak lagi [12].

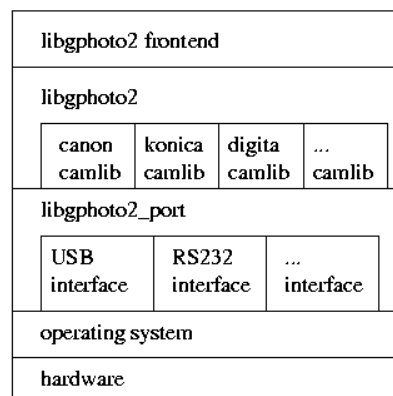
*Python* awalnya dikembangkan pada akhir tahun 1980-an di *National Research Institute for Mathematics and Computer Science* oleh Guido van Rossum sebagai penerus bahasa ABC [13]. *Python* dikembangkan dengan fokus untuk memastikan bahwa kode *Python* mudah dibaca. *Python* termasuk bahasa pemrograman tingkat tinggi yang berarti kode *Python* ditulis menggunakan bahasa Inggris yang sebagian besar sudah dikenali. Berbeda dengan bahasa pemrograman tingkat rendah seperti *Assembler* susah dipahami oleh manusia.

### H. Gphoto2 Library

*Gphoto2* merupakan perangkat lunak berbasis *Linux* yang digunakan untuk mengendalikan kamera digital. Saat ini *gphoto2* sudah memiliki ratusan *driver* kamera digital yang terdapat dipasaran, sehingga dapat terdeteksi dengan baik di *Linux* [14]. *Gphoto2* merupakan bagian dari perangkat sumber terbuka *Linux* untuk digunakan pada kamera digital. *Gphoto2* dikembangkan di bawah lisensi GNU LGPL, karena itu tersedia juga untuk penggunaan komersial [15].

### I. LibGphoto2

*Libgphoto2* (pustaka *gphoto2*) berisi mengenai perintah-perintah yang bisa digunakan untuk mengendalikan kamera digital. *Libgphoto2* merupakan pustaka inti yang dirancang untuk memungkinkan akses ke kamera digital dengan program eksternal. Gambar 2 menunjukkan arsitektur global dari *Libgphoto2*.



Gambar 2. Arsitektur *LibGphoto 2*

*Port* komunikasi dan protokol kamera memungkinkan untuk membuat modularitas yang lengkap. Untuk mendukung lapisan fisik (*physical layer*) seperti IEEE1394 tinggal menambahkan *port* baru ke *libgphoto2\_port*. Untuk mendukung kamera digital model baru tinggal menyediakan *camlib* baru dengan pemanggilan perintah yang diperlukan. Perintah-perintah dari pustaka *gphoto2* yang berbasis pemrograman *Python* dirangkum dalam satu pustaka baru yang disebut *Piggyphoto*. *Piggyphoto* dapat diunduh dengan mengetikkan perintah *sudo apt-get install piggyphoto* di terminal *Raspberry Pi*. Perintah dasar dari *Piggyphoto* yang digunakan dalam penelitian ini diantaranya:

1. *Preview.py*  
*Preview.py* adalah perintah untuk mengendalikan layar kamera DSLR dari jarak jauh dan akan ditampilkan di layar *Raspberry Pi* jika perintah tersebut diketik di terminal.
2. *Snap.py*  
*Snap.py* merupakan perintah untuk mengambil gambar yang sudah di tampilkan di layar, perintah tersebut diketik di terminal *Raspberry Pi*.
3. *Focus-snap.py*  
*Focus-snap.py* memberikan perintah pada kamera DSLR untuk mengambil gambar secara otomatis dengan terlebih dahulu mengaktifkan *focus* untuk mengunci objek yang tertangkap lensa kamera.

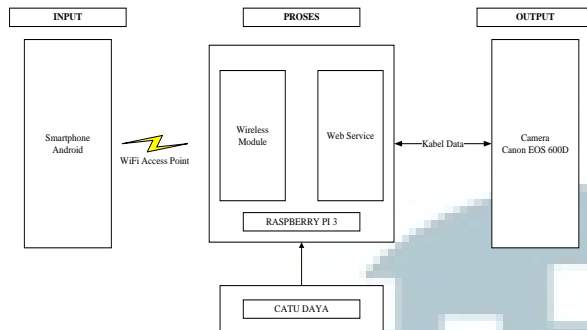
## III. HASIL DAN PEMBAHASAN

Pada tahapan ini terdiri dari tahapan penelitian yang dilakukan. Tahapan tersebut terdiri dari gambaran umum sistem yang akan dijelaskan dalam blok diagram sistem, analisis dan perancangan sistem, analisis dan konfigurasi jaringan yang terdiri dari konfigurasi jaringan berupa konfigurasi *interfaces* jaringan, konfigurasi IP statis, serta konfigurasi dan pengaturan *access point*. Tahapan selanjutnya setelah konfigurasi jaringan yaitu tahapan konfigurasi *gphoto2* library, konfigurasi ISO, konfigurasi *aperture*,

konfigurasi *shutter speed*, konfigurasi *preview.py*, konfigurasi *snap.py* serta tahapan terakhir yaitu konfigurasi *flask API*.

#### A. Gambaran Umum

Gambaran umum dari aplikasi yang akan dibuat dapat dilihat pada Gambar 3.

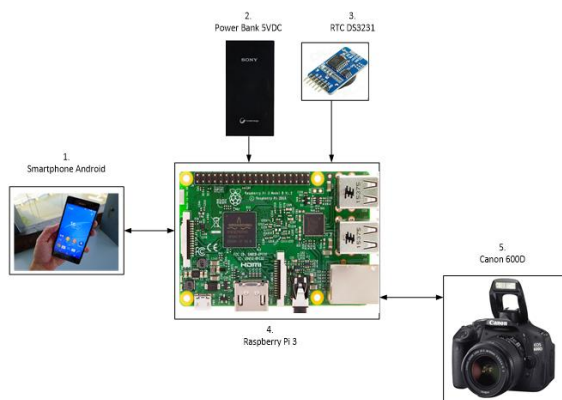


Gambar 3. Blok Diagram Sistem

Secara umum cara kerja sistem perangkat pengendali kamera DSLR nirkabel dapat dijelaskan sebagai berikut: input berasal dari pengguna (*user*) yang menggunakan *smartphone* Android dan terkoneksi *WiFi*. Setelah terhubung dengan *access point Raspberry Pi*, *smartphone* Android berkomunikasi melalui *RESTfull API* dan mengakses perintah-perintah yang sudah disimpan di *Raspberry Pi*. Perintah-perintah tersebut berfungsi untuk mengendalikan *focus*, *ISO*, *aperture*, *shutter speed*, *live view*, *capture*, dan mengunduh berkas-berkas di memori kamera. Setelah mendapat input dari *smartphone* Android, *Raspberry Pi* mengeksekusi perintah untuk menjalankan kamera DSLR.

#### B. Analisis dan Perancangan Sistem

Sistem yang akan dibangun yaitu aplikasi pengendali kamera DSLR nirkabel berbasis Android untuk kamera DSLR *low-end*. Sistem yang akan dibangun dapat dilihat pada Gambar 4 berikut ini.



Gambar 4. Konsep sistem yang akan dibangun

*Raspberry Pi* berfungsi sebagai pengolah perintah-perintah yang dikirimkan oleh *smartphone* Android. *Smartphone* Android memberikan perintah pada *Raspberry Pi* untuk mengaktifkan fitur-fitur yang ada pada kamera DSLR menggunakan jaringan nirkabel *WiFi*. Setelah pengaturan sesuai pengguna aplikasi tinggal menekan tombol *snap* untuk mengambil gambar, jika gambar yang diambil belum sesuai proses tinggal diulang dari awal. Penyimpanan waktu pada *Raspberry Pi* tidak berjalan secara *real time* jika tidak tersambung pada jaringan internet, karena itu ditambahkan modul RTC DS3231.

#### C. Analisis dan Konfigurasi Sistem

Konfigurasi sistem dari aplikasi yang dibuat dibagi ke dalam beberapa tahapan yang akan dijelaskan di bawah ini.

##### 1. Konfigurasi jaringan

Agar perangkat dapat terhubung melalui jaringan *WiFi* tanpa memerlukan koneksi internet diperlukan beberapa tahapan. Tahapan pertama melakukan konfigurasi *DHCP server Raspberry Pi* dan set IP menjadi statis agar ketika terhubung dengan perangkat tidak memerlukan konfigurasi IP ulang. Pada bagian ini melakukan konfigurasi *access point*, tahapan ini diperlukan agar *Raspberry Pi* dapat memancarkan sinyal *WiFi* secara otomatis. Untuk konfigurasi *Access Point* terlebih dahulu merubah pengaturan IP menjadi statis dengan mengetikkan konfigurasi seperti pada Gambar 5.

```
$ sudo nano /etc/network/interfaces
```

Gambar 5. Konfigurasi *Interfaces* Jaringan

Langkah selanjutnya yaitu mengubah IP dari *Raspberry Pi* menjadi statis seperti pada Gambar 6 berikut :

```
iface wlan0 inet static
address 192.168.10.1
netmask 255.255.255.0
```

Gambar 6. Konfigurasi IP Statis pada *Raspberry pi*

Untuk konfigurasi *access point* maka perlu dibuat suatu file dengan masuk ke setingan seperti pada Gambar 7:

```
$ sudo nano /etc/hostapd/hostapd.conf
```

Gambar 7. Konfigurasi *Access Point*

Kemudian selanjutnya membuat file untuk melakukan pengaturan *access point* seperti pada Gambar 8 :

```
interface=wlan0
ssid=RPi_cam
country_code=US
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=raspberr
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP
wpa_group_rekey=86400
ieee80211n=1
wme_enabled=1
```

Gambar 8. Pengaturan *Access Point*

## 2. Konfigurasi *Gphoto2 library*

*Gphoto2 library* dapat diunduh gratis dengan mengetik perintah `sudo apt-get install gphoto2` pada terminal *Raspberry Pi*. Pemilihan *library* harus disesuaikan dengan tipe kamera DSLR karena tidak semua perintah *gphoto2 library* dapat digunakan. Untuk instalasi *gphoto2* maka unduh file tersebut dengan mengetikkan perintah berikut ini:

```
$ sudo apt install gphoto2
```

Gambar 9. Unduh dan *Install Gphoto2*

Beberapa perintah yang ada di *gphoto2 library* dapat dilihat pada Gambar 10. *Gphoto2 library* yang digunakan dalam penelitian ini yaitu untuk pengaturan ISO, *aperture* dan *shutter speed*.

```
pi@raspberrypi:~$ gphoto2
Usage: gphoto2 [-s=serialnumber] [-?] [--help] [--usage] [--debug]
  [--debug-loglevel=STRING] [--debug-logfile=FILENAME] [-q|--quiet]
  [--hook-script=FILENAME] [--stdout] [--stdout-size] [--auto-detect]
  [--show-xfif=STRING] [--show-info=STRING] [--summary] [--manual]
  [--about] [--storage-info] [--shell] [-v|--version]
  [--list-cameras] [--list-ports] [-a|--abilities] [--port=FILENAME]
  [--speed=SPEED] [--camera=MODEL] [--usbid=USBIDS] [--list-config]
  [--list-all-config] [--get-config=STRING] [--set-config=STRING]
  [--set-config-index=STRING] [--set-config-value=STRING] [--reset]
  [--keep] [--keep-ras] [--no-keep]
  [--wait-event=COUNT, SECONDS, MILLISECONDS or MATCHSTRING]
  [--wait-event-and-download=COUNT, SECONDS, MILLISECONDS or MATCHSTRING]
  [--capture-preview] [--show-preview] [-B|--bulb=SECONDS]
  [-F|--frames=COUNT] [-I|--interval=SECONDS] [--reset-interval]
  [--capture-image] [--trigger-capture] [--capture-image-and-download]
  [--capture-movie=COUNT or SECONDS] [--capture-sound]
  [--capture-tethered=COUNT, SECONDS, MILLISECONDS or MATCHSTRING]
  [--trigger-capture] [-l|--list-folders] [-L|--list-files]
```

Gambar 10. *Gphoto2 Library*

## 3. Konfigurasi ISO

Perintah yang digunakan untuk pengaturan ISO ada dua perintah yaitu untuk melihat nilai ISO dan perintah untuk merubah nilai ISO. Untuk melihat nilai ISO dapat mengetikkan perintah: `gphoto2 --get-config iso`. Untuk merubah parameter ISO pengguna tinggal mengetikkan perintah: `gphoto2 --set-config iso=x`, dengan x = nilai parameter ISO pada Gambar 10. Dengan mengetikkan perintah seperti pada Gambar 11, maka nilai ISO pada kamera DSLR akan berubah.

```
pi@raspberrypi:~$ gphoto2 --set-config iso=200
```

Gambar 11. Cara Merubah Parameter Nilai ISO

## 4. Konfigurasi *Aperture*

Untuk merubah nilai *aperture* kamera, perintah yang digunakan sama dengan perintah yang digunakan untuk merubah nilai ISO. Untuk melihat parameter nilai *aperture* pengguna tinggal mengetikkan: `gphoto2 --get-config aperture`, seperti terlihat pada Gambar 12. Untuk merubah nilai *aperture* maka gunakan perintah: `gphoto2 --set-config aperture=x`, dengan x = nilai *aperture*.

```
pi@raspberrypi:~$ gphoto2 --set-config aperture=5.6
```

Gambar 12. Cara Merubah Nilai *Aperture*

## 5. Konfigurasi *Shutter Speed*

Untuk melihat parameter nilai *shutter speed* dengan cara mengetikkan `gphoto2 --get-config shutter speed`. Untuk merubah nilai *shutter speed* maka ketikkan perintah seperti pada Gambar 13.

```
pi@raspberrypi:~$ gphoto2 --set-config shutter speed=1/6
```

Gambar 13. Cara Merubah Nilai *Shutter Speed*

Untuk mendapatkan *piggyphoto* pengguna bisa mendapatkannya di repositori <https://github.com>, dengan cara mengetikkan perintah `git clone`, setelah diunduh akan tersimpan di folder `/home/pi/piggyphoto`. Daftar perintah yang digunakan untuk penelitian ini sudah disediakan oleh *piggyphoto library*. Perintah-perintah tersebut dapat dilihat pada Gambar 14.

```
pi@raspberrypi:~/piggyphoto$ ls
autodetect.py  list-config.py  piggyphoto  sflask.py
bulb.py        list-files.py  preview.jpg  snap.jpg
focus.py      mat            preview.py  snap.py
focus-snap.py my-logfile.txt README.md    templates
```

Gambar 14. Daftar Perintah *Piggyphoto Library*

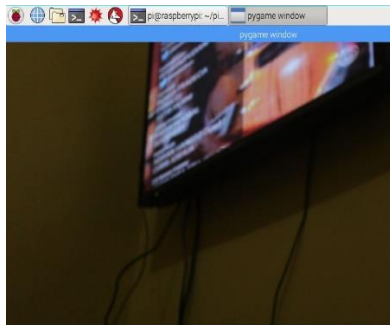
## 6. Konfigurasi *Preview.py*

*Preview.py* adalah perintah untuk mengendalikan *liveview* kamera secara nirkabel. Untuk mengakses perintah ini melalui terminal harus masuk ke direktori *piggyphoto* terlebih dahulu dengan mengetikkan: `~$ cd piggyphoto`. Setelah masuk, ketikkan: `sudo python preview.py` maka *liveview* kamera dapat dilihat pada layar *raspberrypi*.

```
pi@raspberrypi:~$ cd piggyphoto/
pi@raspberrypi:~/piggyphoto$ sudo python preview.py
```

Gambar 15. Perintah *Liveview* Kamera

Tampilan layar *raspberrypi* dapat dilihat pada Gambar 16.



Gambar 16. Tampilan Liveview Kamera

6. Konfigurasi Snap.py

Snap.py adalah perintah untuk mengambil gambar dan menampilkannya pada layar Raspberry Pi. Ketikkan perintah: `sudo python snap.py` pada direktori `piggyphoto`. Konfigurasi `snap.py` perintahnya dapat dilihat pada Gambar 17 berikut :

```
pi@raspberrypi:~/piggyphoto $ sudo python snap.py
Model : Canon EOS 1200D
Status : 0
Port : 4
Operations : 57
File Operations : 10
Folder Operations : 14
USB (vendor/product) : 0x4a9/0x327f
USB class : 0x0/0x0/0x0
Library : /usr/local/lib/libgphoto2/2.5.10/ptp2
Id : FTP
```

Gambar 17. Perintah snap.py

6. Konfigurasi flask API

Flask API berfungsi sebagai jembatan penghubung antara Raspberry Pi dan smartphone Android. Untuk mendapatkan *micro framework Flask*, ketikkan perintah yang ada pada Gambar 18 berikut:

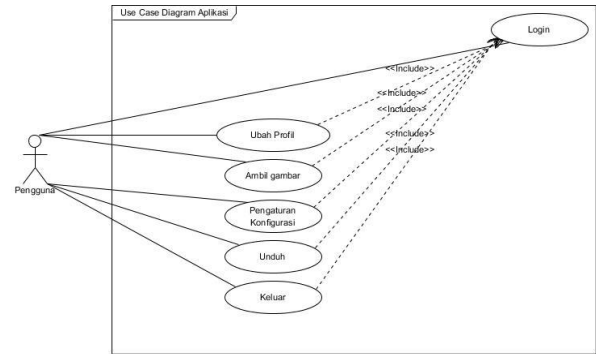
```
pi@raspberrypi:~ $ sudo pip install Flask
```

Gambar 18. Install Flask API

Setelah instalasi *Flask Micro Framework* selesai dilanjutkan dengan mengatur perintah-perintah dari *gphoto2 library* dan *piggyphoto* pada *Flask API* agar dapat diakses melalui *HTTP-Request*.

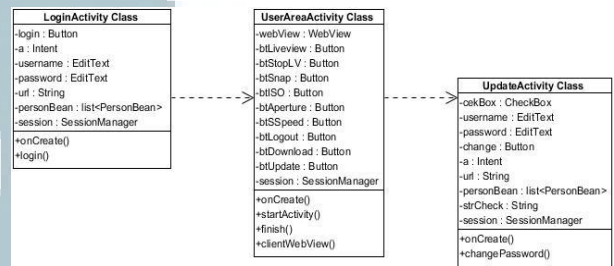
D. Use Case dan Class Diagram Aplikasi Mobile Android

Diagram *usecase* dari aplikasi mobile yang dibangun terdiri dari sepuluh fungsional, yaitu *Login*, *update password*, *liveview*, *stop liveview*, *snap*, *ISO setting*, *aperture setting*, *shutter speed setting*, dan *logout*.



Gambar 19. Diagram Use Case Aplikasi Mobile Android

Sedangkan *class diagram* dari aplikasi mobile android yang dibangun dapat dilihat pada Gambar 20.



Gambar 20. Class Diagram Aplikasi Mobile Android

E. Implementasi Sistem

Tahap implementasi merupakan tahap dimana aplikasi pengendali kamera DSLR nirkabel berbasis android siap dioperasikan. Tahapan ini juga akan menjelaskan mengenai implementasi perangkat keras, implementasi perangkat lunak, dan implementasi program serta antarmuka aplikasi pengendali kamera DSLR nirkabel berbasis android untuk kamera DSLR *low-end*.

E.1. Implementasi Perangkat Keras Pengguna

Perangkat keras komputer dan *hardware* lainnya yang digunakan untuk mengimplementasikan aplikasi pengendali kamera DSLR nirkabel berbasis android dapat dilihat pada Tabel 1 berikut.

Tabel 1. Implementasi Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1.	Processor	Dual Core 1.2 GHz Cortex-A53
2.	RAM	2 GB
3.	Layar	720x1280 pixels (~294 ppi pixel density)
4.	Baterai	Removable Li-Ion 2600
5.	Raspberry Pi	Raspberry Pi 3 Model B
6.	SD Card	16 GB



No.	Perangkat Keras	Spesifikasi
7.	<i>Real Time Clock</i>	RTC DS3231
8.	Kamera DSLR	Canon EOS 1200D
9.	Adaptor	5VDC/2A

### E.2. Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk mengimplementasikan aplikasi ini terbagi menjadi dua tipe, yaitu implementasi perangkat lunak pada komputer dan perangkat lunak pada *Raspberry Pi*. Untuk selengkapnya dapat dilihat pada Tabel 2.

Tabel 2. Implementasi Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	Windows 10 Pro, <i>Raspbian Jessie</i>
2.	Android Studio	Versi 2.2.3
3.	Android SDK	Level 24 ke atas
4.	WinSCP + PuTTY	Versi 5.9.3
5.	Bahasa pemrograman	Java, <i>Python 2.7</i>
6.	Database	<i>SQLite</i>
7.	API	<i>Flask API</i>

### E.3. Implementasi Aplikasi

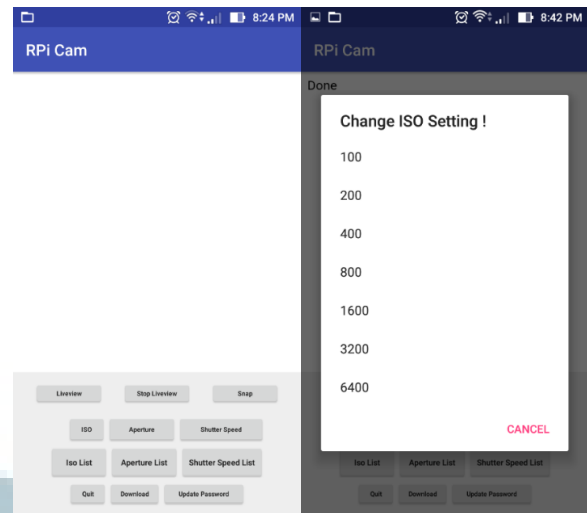
Aplikasi pengendali kamera DSLR nirkabel berbasis Android untuk kamera DSLR *low-end* membutuhkan proses instalasi pada perangkat android. Pengguna harus mempunyai file *rplicam.apk* pada perangkat android yang akan digunakan untuk menjalankan aplikasi tersebut. Setelah aplikasi terinstall, pengguna juga harus mempunyai kamera DSLR *low-end*.

#### E.3.1 Implementasi Antarmuka

Antarmuka merupakan bagian penting pada aplikasi pengendali kamera DSLR nirkabel berbasis android untuk kamera DSLR *low-end*. Implementasi antarmuka dari aplikasi yang dibangun dapat dilihat di bawah ini.

##### 1. Antarmuka menu utama

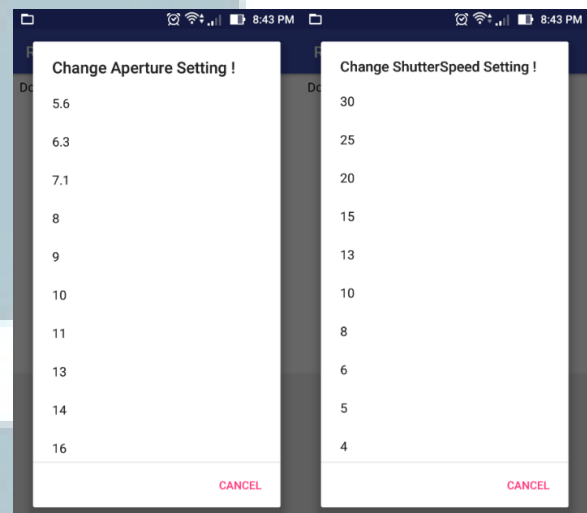
Antarmuka utama terdiri dari tampilan *liveview*, *stop liveview*, *snap*, *ISO Setting*, *Aperture Setting*, *Shutter Speed Setting*, *Update Profile*, dan *Download*.



Gambar 22. Tampilan Menu Utama dan ISO Setting

##### 2. Antarmuka Aperture dan Shutter Speed Setting

Antarmuka untuk *setting aperture* dan *shutter speed* dapat dilihat pada Gambar 23 berikut:



Gambar 23. Tampilan Setting Aperture dan Shutter Speed

##### 3. Antarmuka Download

Antarmuka *download* ini berfungsi untuk menyimpan hasil tampilan gambar kedalam media penyimpanan internal ataupun eksternal pada *smartphone*. Tampilannya dapat dilihat pada Gambar 24.

Gambar 24. Tampilan *Download Gambar*

#### F. Pengujian Sistem

Pengujian dilakukan dengan tujuan untuk menemukan kesalahan dan kekurangan pada perangkat lunak yang diuji. Pengujian tersebut dimaksudkan untuk mengetahui perangkat lunak yang dibuat apakah sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak. Pengujian dilakukan dengan teknik pengujian *black box* yang lebih berfokus pada menemukan kesalahan program secara fungsional.

Pengujian *black box* dilakukan untuk komponen fungsional *login*, pengambilan gambar, pengaturan parameter kamera, *update profile* dan fungsional *download*. Pengujian pengambilan gambar terdiri dari pengujian penekanan tombol *liveview*, *stop liveview* dan tombol *snap*. Pengujian pengaturan parameter kamera terdiri dari pengujian pengaktifan parameter ISO, pengaktifan parameter *aperture* serta pengaktifan parameter shutter *speed*. Dari hasil pengujian *black box* dapat disimpulkan bahwa aplikasi ini sudah berhasil dapat berjalan sesuai yang diharapkan.

Selain itu, pengujian juga dilakukan pada bulan Januari 2017 dengan menyebarkan kuesioner terhadap 46 responden yang merupakan komunitas fotografi di Kota Bandung. Responden tersebut memberikan jawaban yang variatif, namun dari nilai rata-rata yang ada dapat ditarik kesimpulan bahwa responden setuju pengoperasian kamera secara nirkabel dapat memudahkan pengoperasian, responden juga setuju bahwa pengoperasian kamera DSLR secara nirkabel memudahkan akses untuk mengunduh data di kartu memori kamera.

Selain pengujian fungsional kepada pengguna, dilakukan juga pengujian dengan model *smartphone* yang berbeda. Dalam pengujian ini dilakukan dengan menggunakan tiga buah *smartphone* yang berbeda, yaitu LG Nexus 4, Sony Xperia Z1 dan Asus Zenfone Go. Berdasarkan hasil pengujian *smartphone* android terhadap aplikasi yang dilakukan maka dapat

disimpulkan bahwa aplikasi dapat berjalan pada *smartphone* yang berbeda jenis, hanya saja masih mempunyai kelemahan yakni apabila digunakan pada LG Nexus 4 sering terjadi *delay* ketika mengakses *port* kamera setelah menggunakan mode *liveview*, selain kekurangan tersebut didapatkan bahwa semua fungsional aplikasi dapat berjalan dengan baik pada tiga *smartphone* android yang digunakan ketika pengujian.

#### IV. SIMPULAN

Adapun kesimpulan dan saran yang didapatkan dari hasil penelitian ini yaitu :

##### A. Kesimpulan

Berdasarkan hasil penelitian, analisis, perancangan sistem, dan implementasi serta pengujian sistem, dapat diambil kesimpulan sebagai berikut :

1. Aplikasi pengendali kamera DSLR nirkabel berbasis android dapat membantu pengoperasian kamera.
2. Aplikasi pengendali kamera DSLR nirkabel berbasis android membantu memudahkan akses data kamera jika ingin diunduh.

##### B. Saran

Aplikasi pengendali kamera DSLR nirkabel berbasis android untuk kamera DSLR *low-end* ini masih ada kekurangan. Oleh karena itu perlu dilakukan pengembangan dan penyempurnaan lebih lanjut. Adapun beberapa saran yang mungkin dapat diterapkan untuk pengembangan selanjutnya adalah:

1. Penyempurnaan proses *liveview* dan *snap* agar tidak terjadi konflik pada *port* kamera sehingga tidak terjadi lagi *error* saat terkoneksi.
2. Penyempurnaan *liveview* kamera agar kualitas video lebih stabil.
3. Penambaham fitur pengendali terminal *Raspberry Pi* agar bisa dikendalikan langsung dari *smartphone* jika terjadi *error*.

#### DAFTAR PUSTAKA

- [1] P. Nugrahajati and E. Targo, *Buku Pintar Fotografi dengan Kamera DSLR*, 1st ed. Yogyakarta: Indonesia Tera, 2011.
- [2] D. D. Busch, *Mastering Digital SLR Photography*, 3rd ed. Boston: Course Technology, 2012.
- [3] L. C. Dewi, "Wireless Technology Development: History, Now, and Then," *ComTech*, vol. 2, no. 9, pp. 1228–1240, 1238–1240, 2011.
- [4] Y. O. Primariadi, E. Susanto, and U. Sunarya, "Perancangan Kendali pada Tripod dan Kamera DSLR Menggunakan Komunikasi Bluetooth Berbasis Aplikasi Android," *e-Proceeding Eng.*, vol. 2, no. 2, pp. 2, 4–5, 2015.

- [5] D. Kurniawan, *Membangun Aplikasi Elektronika dengan Raspberry Pi 2 dan WhatsApp*, 1st ed. Jakarta: PT. Elex Media Komputindo, 2016.
- [6] G. Andreas and C. R. David, *Professional Android Open Accessory Programming with Arduino*. Indiana: John Wiley & Sons, Inc, 2013.
- [7] S. Salbino, *Buku Pintar Gadget Android untuk Pemula*, 1st ed. Jakarta: Kunci Komunikasi, 2014.
- [8] W.-M. Lee, *Beginning Android 4 Application Development*, 4th ed. Indianapolis: John Wiley & Sons, Inc, 2012.
- [9] P. Sarang, *Java Programming*. New York: Oracle Press, 2012.
- [10] P. Deitel and H. Deitel, *Java How to Program*, 10th ed. New Jersey: Pearson, 2015.
- [11] D. Lowe, *Java All-in-One For Dummies*, 4th ed. Hoboken: John Wiley & Sons, Inc, 2014.
- [12] M. L. T. Cossio, L. F. Giesen, and G. Araya, *Programming Python*, 4th ed., vol. XXXIII, no. 2. Sebastopol: O'Reilly, 2012.
- [13] G. Halfacree and E. Upton, *Raspberry Pi User Guide*, 1st ed. West Sussex: John Wiley & Sons, Inc, 2012.
- [14] I. Linux, "Alternatif," pp. 40–43, 2006.
- [15] T. Koutecký, D. Paloušek, and J. Brandejs, "Method of photogrammetric measurement automation using TRITOP system and industrial robot," *Optik (Stuttg)*, vol. 124, no. 18, pp. 3705–3709, 2013.



# Analisis dan Implementasi Protokol Otentikasi FIDO U2F

Sunderi Pranata<sup>1</sup>, Hargyo Tri Nugroho<sup>1</sup>, Hirofumi Yamaki<sup>2</sup>

<sup>1</sup> Program Studi Sistem Komputer, Universitas Multimedia Nusantara, Tangerang, Indonesia

<sup>2</sup> School of Information Environment, Tokyo Denki University, Tokyo, Japan  
sunderi.pranata@gmail.com<sup>1</sup>

Diterima 5 Mei 2017

Disetujui 16 Juni 2017

**Abstract**— It is known that password itself is not enough for formidable authentication method since it has a lot of vulnerabilities. Multi factor authentication (MFA) is introduced for the next generation for good authentication to address that issue. MFA combines two or more of three principles of good security, “something you know”, “something you have”, and “something you are”. Most MFA mechanisms work as one time passwords (OTP). However, they can still be vulnerable to phishing and MiTM attack. On top of that, OTP can be hard to use as it requires user to input another password given by the device (SMS, token, authenticator). Implemented in small USB U2F device, FIDO U2F delivers easier yet stronger security on authentication process which implements public key cryptography, challenge-response protocol, and phishing and MiTM protection.

**Index Terms**— Authentication protocol, FIDO U2F, Multi factor authentication, OTP

## I. PENDAHULUAN

Kata sandi atau *password* umum digunakan sebagai mekanisme otentikasi di dunia maya. Namun, otentikasi dengan kata sandi sendiri memiliki banyak celah keamanan. Kata sandi yang dirancang dengan sembrono sangat rentan diretas. Bahkan masyarakat masih sering menggunakan kata sandi yang berisi data-data pribadi dan mudah ditebak seperti *username*-nya, nama binatang peliharaan, dan tanggal ulang tahun. Namun kata sandi yang kuat bukan berarti tanpa masalah karena kata sandi yang kuat adalah kata sandi yang umumnya sulit untuk diingat [1].

Masalah kata sandi ini sebenarnya sudah ada solusinya yaitu *Multi Factor Authentication* (MFA). *Multi Factor Authentication* adalah metode otentikasi dengan dua atau lebih dari tiga prinsip dasar otentikasi yaitu *something you know* (SYK), *something you have* (SYH), dan *something you are* (SYA) [2] [3]. Pada umumnya, bentuk MFA yang paling umum digunakan sampai saat makalah ini ditulis adalah SYK dan SYH yang secara berurutan berupa kata sandi dan *token* atau telepon genggam.

MFA bekerja dengan baik untuk menutupi kelemahan dari kata sandi tetapi masih berisiko terhadap serangan *phishing* dan *MiTM*. Selain itu, faktor SYH pada MFA biasanya berupa *One Time*

*Password* (OTP). Sayangnya, bagi pengguna awam, OTP umumnya relatif sulit digunakan. Pengguna harus menggenerasi dan meng-input OTP sendiri hanya untuk melakukan satu kali proses otentikasi. Prosesnya relatif panjang dan tidak praktis. Hal inilah yang mendasari FIDO Alliance untuk membuat protokol otentikasi yang baru dan lebih aman serta lebih mudah digunakan yang kemudian dinamakan protokol FIDO dan diluncurkan pada Februari 2013 [4].

FIDO bekerja di atas konsep MFA dan *public key cryptography*. Salah satu keuntungan dari FIDO adalah bahwa pengguna tidak perlu kata sandi yang kompleks, pengguna bahkan dapat menggantikan kata sandinya dengan kata sandi yang sederhana seperti PIN 4 digit.

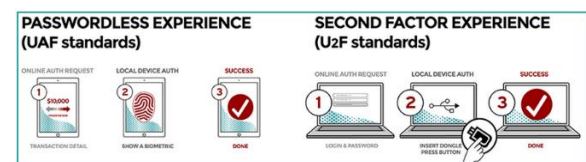
Pada penelitian ini FIDO diimplementasikan pada sebuah sistem yang bersifat *proof-of-concept* baik pada sisi server maupun klien untuk membuktikan kehandalan FIDO sebagai protokol otentikasi.

## II. KAJIAN PUSTAKA

### A. FIDO

FIDO Alliance mengedepankan tiga aspek utama yaitu 1) kemudahan penggunaan, 2) privasi dan sekuriti, dan 3) standardisasi. Tujuan utama adalah untuk membuat layanan *online* dan *website* baik untuk korporasi atau terbuka, untuk mendukung fitur keamanan yang tertanam secara *native* untuk dukungan fitur keamanan yang baik dan mengurangi masalah yang muncul akibat proses pembuatan dan pengingatan kata sandi yang banyak dan berbeda setiap situs [4].

Protokol FIDO terbagi menjadi dua yang mengakomodasi dua pilihan bagi pengguna untuk berinteraksi dengan otentikasi dunia maya. Dua protokol tersebut memiliki banyak basis protokol yang sama tetapi dibedakan hanya untuk kasus penggunaan yang spesifik. Kedua protokol FIDO itu adalah FIDO U2F dan FIDO UAF [5].



Gambar 1. FIDO U2F dan UAF [6]



Gambar 2. Proses Otentikasi FIDO U2F [7]

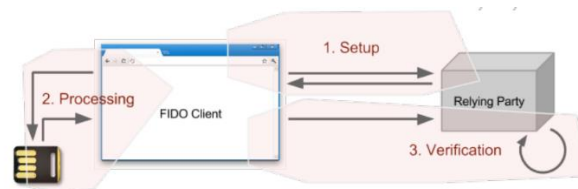
### B. FIDO U2F

Protokol FIDO U2F memungkinkan layanan online untuk tetap menggunakan infrastruktur layanan kata sandi tetapi dengan menambahkan faktor keamanan kedua yang kuat untuk log in. Layanan online juga dapat meminta pengguna untuk mempresentasikan perangkat faktor kedua kapan pun dibutuhkan. Faktor kedua tersebut memungkinkan pengguna untuk menggunakan kata sandi yang sederhana seperti 4 digit PIN tanpa melemahkan aspek keamanan. Pada saat proses registrasi dan otentikasi, pengguna cukup mempresentasikan perangkat faktor kedua ke komputer hanya dengan gerakan menekan tombol pada perangkat USB atau *tap* NFC. Pengguna dapat menggunakan perangkat FIDO U2F terhadap banyak layanan online yang mendukung FIDO U2F tetapi harus disertai dengan web browser yang mendukung [5].

Dari sisi user, layanan online dapat melakukan proses otentikasi seperti biasa yaitu dengan memasukkan username dan kata sandi. Setelah itu, FIDO U2F akan digunakan sebagai faktor kedua yang memperkuat keamanan otentikasi. Server akan meminta pengguna untuk melakukan gerakan sederhana untuk membuktikan bahwa pengguna ada di tempat. Gerakan tersebut dapat berupa penekanan tombol pada perangkat FIDO U2F, *tapping* perangkat FIDO U2F, atau pun *plug in* perangkat FIDO U2F. Proses dari otentikasi faktor kedua dengan FIDO U2F jauh lebih sederhana dibanding dengan proses OTP yang mengharuskan proses generasi dan *input* manual kode ke sistem.

### C. FIDO UAF

Protokol FIDO UAF memungkinkan layanan online untuk menawarkan layanan otentikasi tanpa kata sandi. Pengguna akan meregistrasikan perangkat otentikasi ke layanan online dengan pemilihan mekanisme otentikasi seperti gestur jari pada telepon genggam, melihat ke kamera, berbicara ke mic, memasukkan PIN, dll. Setelah teregistrasikan, pengguna cukup melakukan otentikasi tadi secara lokal kapan pun pengguna membutuhkan untuk otentikasi ke layanan tersebut. Pengguna tidak perlu lagi memasukkan kata sandi apa pun ketika ingin mengotentikasi dari perangkat tersebut. [5]



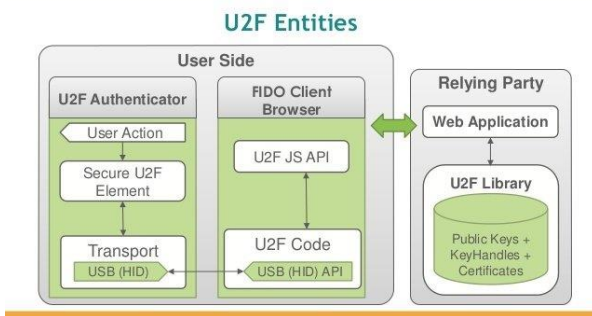
Gambar 3. Tiga Fase Dari Proses Registrasi Dan Otentikasi [8]

### D. Protokol FIDO U2F

*Token* U2F menyediakan kriptografik *assertions* yang dapat diverifikasi oleh server dan kriptografi digunakan sebagai faktor kedua (setelah kata sandi) saat proses otentikasi. Token U2F umumnya kecil dan berupa perangkat dengan tujuan khusus yang tidak terkoneksi dengan internet tetapi dapat berkomunikasi dengan server. Maka dari itu, token FIDO U2F memerlukan FIDO *client* untuk me-*relay* pesan antara token dan server. Pada umumnya FIDO *client* berupa web browser.

Protokol FIDO U2F dibagi menjadi dua proses besar yaitu proses registrasi dan proses otentikasi. Proses registrasi akan mendaftarkan pasangan kunci (*key pair*) yang digenerasi atas kontrol token U2F sedangkan proses otentikasi akan membuktikan kepemilikan dari *key pair* yang telah diregistrasikan sebelumnya ke server. Kedua proses tersebut terdiri dari 3 fase yaitu [8]:

- 1. Setup:** pada fase ini, FIDO *client* akan mendapatkan *challenge* dari server. Dengan *challenge* tersebut, FIDO *client* akan membuat *request message* kepada token U2F
- 2. Processing:** pada fase ini, FIDO *client* akan mengirim *request message* ke token, dan token akan melakukan operasi kriptografi pada *message* tersebut dan membuat *response message*. *Response message* ini kemudian akan dikirim ke FIDO *client*.
- 3. Verification:** pada fase ini, FIDO *client* mengirim *response message* dari token bersamaan dengan data-data lainnya yang diperlukan oleh server untuk melakukan verifikasi token *response* ke server. Server kemudian akan memroses *token response* ini untuk membuktikan kebenarannya. Jika benar pada proses registrasi maka server akan mendaftarkan *public key* yang baru untuk user tersebut sedangkan jika benar pada proses otentikasi maka server akan menerima bahwa klien memiliki *private key* yang dibutuhkan dan dapat melanjutkan ke proses selanjutnya.



Gambar 4. Entitas U2F [9]

E. Entitas FIDO U2F

Pada protokol U2F entitas yang penting terbagi menjadi tiga yaitu U2F authenticator, browser, dan server.

1. **Server:** server umumnya berupa aplikasi web online. Pada saat proses registrasi, entitas ini akan menggenerasi dan mengirimkan challenge ke browser yang meminta dan menerima public key yang digenerasikan oleh U2F authenticator, key handles, dan certificates dan menyimpannya ke dalam basis data. Pada saat proses otentikasi, browser akan membuat challenge, kemudian akan menerima signed-challenge dari browser dan melakukan un-sign-challenge-nya dengan public key yang telah disimpan sebelumnya. Setelah proses otentikasi selesai, server dapat melanjutkan ke tahap selanjutnya seperti mengatur cookie atau session untuk browser.

2. **FIDO Client Browser:** FIDO client browser akan bertindak sebagai jembatan antara U2F authenticator dengan server. Dengan menggunakan library U2F javascript API yang telah dikembangkan, browser dapat berkomunikasi dengan U2F authenticator. Browser akan mengirimkan challenge untuk meminta proses pembuatan key pairs untuk proses registrasi maupun meminta proses sign-the-challenge untuk proses otentikasi.

3. **U2F Authenticator:** U2F authenticator adalah perangkat kecil USB yang dirancang secara khusus untuk otentikasi. Entitas ini akan menggunakan elemen U2F khusus yang dirancang secara aman untuk membuat key pairs dan key handles pada saat proses registrasi dan akan melakukan sign-the-challenge dengan private key yang diregenerasi (akan dijelaskan lebih lanjut pada bagian IV). Semua data tersebut akan dikirim ke browser yang kemudian akan diteruskan ke server.

III. METODE

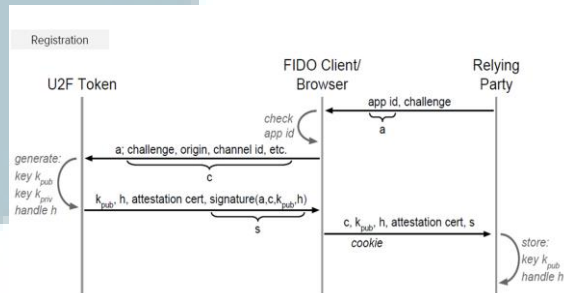
A. Perancangan Sistem

U2F mempunyai dua alir proses yaitu alir proses registrasi dan alir proses otentikasi. Untuk dapat melakukan proses otentikasi dengan U2F, pengguna harus lebih dulu meregistrasikan perangkat U2F

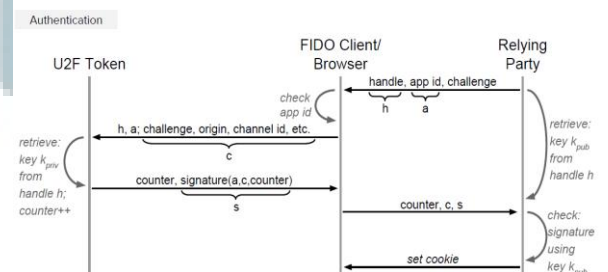
authenticator ke server. Berikut alir proses registrasi dan otentikasi:

Kemudian, proses registrasi akan dimulai dengan server melakukan pengiriman challenge dan appID ke browser, kemudian browser akan mengecek appID dan mengirim data yang diperlukan ke token U2F. Token U2F akan menggenerasi key pairs dan key handles berdasarkan data yang dikirim oleh server, kemudian mengirimkan signed challenge ke browser beserta attestation certificate (optional). Browser kemudian akan mengirim key handle, dan signed challenge ke server. Server akan kemudian menyimpan public key dan key handle yang akan digunakan kembali pada proses otentikasi [10].

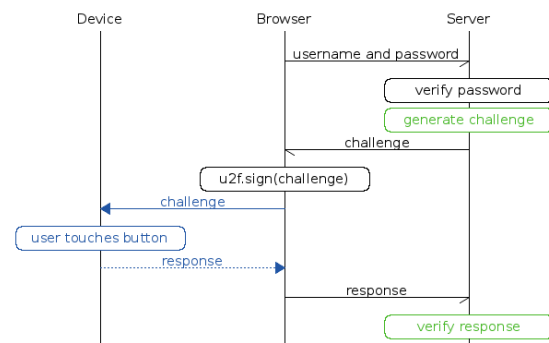
Setelah proses registrasi selesai, pengguna dapat memulai proses otentikasi. Proses otentikasi dimulai dengan pengiriman key handle, appID, dan challenge oleh server ke browser. Browser kemudian akan mengecek appID dan melanjutkan data yang diperlukan ke token U2F. Token U2F akan membuat ulang private key dari key handle dan appID, kemudian sign challenge tersebut dan mengirim hasilnya kembali ke server melalui browser. Server kemudian akan mengecek signature menggunakan public key yang telah disimpan pada saat proses registrasi sebelumnya.



Gambar 5. Proses registrasi U2F [11]



Gambar 6. Proses Otentikasi U2F [11]



Gambar 7. Diagram Sekuensial FIDO U2F [12]

#### IV. ANALISIS DAN IMPLEMENTASI

##### A. Implementasi

Pada makalah ini, penulis mengimplementasikan protokol FIDO U2F dengan membuat web server yang berjalan di atas HTTPS. Server dibuat dengan menggunakan bahasa pemrograman Go. Dikarenakan protokol FIDO U2F *web facet* harus berjalan di atas HTTPS, diimplementasikan juga protokol TLS dengan *self-signed certificate*. Pengimplementasian dari sisi klien menggunakan javascript API yang dibuat Google sebagai salah satu anggota dari FIDO Alliance untuk komunikasi antara perangkat U2F dengan browser.

Ada pun diagram sekuensial sederhana yang menggambarkan keseluruhan sistem ditunjukkan pada gambar 7.

Proses implementasi dibagi menjadi dua bagian besar yaitu implementasi sisi server dan implementasi sisi klien.

Pada **implementasi sisi server**, sebelum mengaplikasikan U2F, server harus berjalan di atas HTTPS. Berikut cuplikan kode yang ditulis dengan bahasa pemrograman Go:

```
s:=u2f.StdServer(&userDB{}, "https://gou2f.com:8079")
http.ListenAndServeTLS(":"8079", "cert.pem", "key.pem", nil)
```

Implementasi server U2F sendiri memiliki 4 fungsi utama yaitu *start registration*, *finish registration*, *start authentication*, *finish authentication*. Pada implementasi sisi server, digunakan library U2F yang mengikuti pseudocode untuk seperti in ditunjukkan pada gambar 8.

**Implementasi sisi klien** pada paper ini berupa web sebagai agen tengah antara server dan perangkat U2F. Perlu dicatat bahwa hingga saat makalah ini ditulis, tidak semua browser mendukung U2F secara *native*, browser yang mendukung U2F secara *native* salah satu contohnya adalah Chrome versi 38 ke atas. Untuk menggunakan U2F pada browser yang tidak didukung secara *native*, diperlukan instalasi plug in tambahan.

```
# handles HTTPS requests to /start_registration
def start_registration(username):
    challenge = u2f_lib.start_registration(APP_ID)
    challenge_store.set(username, challenge)
    return challenge

# handles HTTPS requests to /finish_registration
def finish_registration(username, device_response):
    challenge = challenge_store.pop(username)
    registered_device = u2f_lib.finish_registration(challenge, device_response)
    device_store.set(username, registered_device)
    return "Success!"

# handles HTTPS requests to /start_authentication
def start_authentication(username, password):
    verify_user_pass(username, password)
    registered_device = device_store.get(username)
    challenge = u2f_lib.start_authentication(registered_device, APP_ID)
    challenge_store.set(username, challenge)
    return challenge

# handles HTTPS requests to /finish_authentication
def finish_authentication(username, password, device_response):
    challenge = challenge_store.pop(username)
    u2f_lib.finish_authentication(challenge, device_response, registered_device)
    return "Success!"
```

Gambar 8. Pseudocode Proses Registrasi Dan Otentikasi FIDO U2F [12]

FIDO Alliance telah membuat library pendukung untuk sisi klien dalam bentuk javascript API (*u2f-api.js*) yang pada intinya berisikan dua fungsi utama yaitu: **u2f.register**, yang digunakan pada saat mendaftarkan perangkat U2F; dan **u2f.sign** yang digunakan pada saat mengotentikasi perangkat U2F [13].

Berikut cuplikan kode yang ditulis dengan bahasa pemrograman Go:

```
<script src="https://demo.yubico.com/js/u2f-api.js"></script>
<script>
(function() {
    window.u2fRegister = u2fRegister;
    window.u2fSign = u2fSign;

    function ajax(url, args, cb) {
        var aj = new XMLHttpRequest();
        aj.onreadystatechange = function() {
            if (aj.readyState == 4) {
                if (aj.status == 200) {
                    cb(JSON.parse(aj.responseText));
                } else {
                    msg("failed: " + aj.responseText);
                }
            }
        }
        aj.open('POST', url, true);
        aj.setRequestHeader('Content-type', 'text/json')
        aj.send(JSON.stringify(args));
    }

    function u2fRegister() {
        ajax("/Register", {}, function(r) {
            msg("touch it to register");
            console.log('registerChallenge',r);
            u2f.register([r], [],function(response) {
                console.log('registerSigned',response);
                if (response.errorCode) {
                    msg("failed to enroll:" + response.errorCode);
                    return;
                }
            });
            msg("finalizing/validating registration...");
            u2fRegisterFin(response);
        });
    }
});
```

```

});
}
function u2fRegisterFin(rr) {
  ajax("/RegisterFin", rr, function(r) {
    msg("device registered");
  });
}
function u2fSign() {
  ajax("/Sign", {}, function(r) {
    msg("touch it to sign");
    console.log('authenticationChallenge', r);

    u2f.sign(r, function(response) {
      console.log('authenticationSigned', response);
      if (response.errorCode) {
        msg(response.errorCode);
        return;
      }
      msg("verifying");

      u2fSignFin(response);
    }, 5);
  });
}
function u2fSignFin(verify) {
  ajax("/SignFin", verify, function() {
    msg("logged in");
  });
}
function msg(m) {
  var e = document.getElementById("msg");
  e.innerHTML = 'HI: ' + m;
  e.innerText = m;
}
}());
</script>

```

### B. Analisis Protokol

Protokol U2F berbasis protokol *challenge-response* yang dibuat agar memiliki proteksi phishing dan MitM. Protokol U2F juga dibuat agar memiliki *key* yang unik terhadap situs (*application specific keys*), *device cloning detection*, dan *device attestation*.

1. **Challenge-response:** otentikasi berbasis *challenge-response* ini didasari oleh *public key cryptography*. Perangkat U2F memiliki *key*  $k_{priv}$  dan server memiliki *key*  $k_{pub}$ . *Key pair* ini digenerasi di dalam perangkat dengan operasi yang *tamper-resistant*, di mana  $k_{priv}$  **tidak dapat diekstrak dari perangkat** [10].  $k_{priv}$  hanya berupa fungsi pada hardware dan didesain agar tidak dapat diekstrak dan hanya dapat digunakan langsung oleh proses yang telah ditentukan seperti proses *signing*.

2. **Proteksi dari phishing dan MitM:** konsep ini membutuhkan informasi dari koneksi HTTP yang sedang berlangsung (URI dan TLS channel ID). Informasi ini kemudian di-*sign* oleh perangkat U2F dan kemudian dikirim ke server, yang akan memverifikasi bahwa informasi tersebut adalah benar [10]. Proses ini menambahkan informasi origin (protocol, hostname, dan port number) yang akan menggagalkan phishing serta TLS channel ID yang akan menggagalkan MitM.

3. **Application Specific Keys:** *application specific keys* ditujukan agar server tidak dapat mengetahui bahwa digunakannya perangkat U2F yang sama pada dua akun yang berbeda. Sebagai contoh, example.com tidak akan tahu bahwa user1 dan user2 menggunakan perangkat yang sama. Perangkat U2F akan menggenerasi *key pair* dan *key handle* baru untuk setiap proses registrasi. *Key handle* akan disimpan oleh server dan dikirimkan kembali ke perangkat U2F saat proses otentikasi. Dengan ini, perangkat tahu *key* mana yang harus diotentikasi. Untuk mencapai tujuan ini, dibutuhkan *key handle* dan *appID* (untuk membantu menemukan *key handle*). Ada pun proses pembuatan *key handle* sebagai berikut (perangkat Yubikey karena hingga saat penulisan makalah, belum ada spesifikasi cara menyimpan *keys*).

Ketika pengguna mendaftarkan perangkat U2F, server akan menyediakan AppID (berupa URL situs dan mencegah phishing). Perangkat U2F akan kemudian menggenerasi nonce (N). Yubikey kemudian akan menggunakan AppID dan nonce untuk diproses melalui fungsi HMAC-SHA256, menggunakan *device-specific secret* sebagai *key*. *Device-specific secret key* ini digenerasi *on-chip* pada saat proses manufaktur. Keluaran dari proses tersebut adalah *private key*, dan nonce bersamaan dengan Message Authentication Code (MAC) akan menjadi *key handle*. MAC berfungsi untuk memastikan bahwa *key handle* tersebut hanya valid untuk kombinasi khusus antara perangkat U2F dan appID. Pada saat otentikasi, *key handle* akan dilewatkan pada yubikey lagi dan diverifikasi bahwa *key handle*-nya menggunakan MAC bahwa *key handle*-nya belum diubah dan bahwa *credential*-nya benar sesuai dengan appID. menggunakan MAC [14].

4. **Device cloning detection:** Perangkat U2F memang *tamper-resistant* dan  $k_{priv}$  tidak dapat diekstrak. Namun, untuk menyediakan *cloning detection* pada perangkat U2F yang tidak memiliki perangkat *tamper-resistant* (contoh: implementasi dengan software), protokol ini dirancang dengan penambahan *counter*. Konsepnya sederhana, perangkat akan menambah *counter* setiap kali proses otentikasi dan server akan memverifikasi bahwa *counter* tersebut bernilai lebih tinggi dari yang sebelumnya [10].

5. **Device attestation:** proses pengesahan memungkinkan server untuk memverifikasi properti perangkat U2F, seperti nomor model perangkat U2F. Proses ini diimplementasikan melalui *attestation certificate*, di-*sign* oleh *device vendor*, dan perangkat mengirimkan data tersebut ke server pada saat registrasi. Proses ini tidak berpengaruh pada proses otentikasi dan bersifat optional [10].

### V. SIMPULAN


Implementasi FIDO U2F pada makalah ini masih sebatas *proof of concept* dan masih jauh dari solusi



yang sempurna yang dapat diterapkan. Namun FIDO telah terbukti mudah dan handal untuk digunakan. Ada pun implementasi yang dapat ditambahkan pada masa yang akan datang antara lain penambahan fitur otentikasi pengguna dan kata sandi sebelum otentikasi U2F; penambahan fitur registrasi metode otentikasi pengguna; penambahan fitur penyimpanan data pengguna ke basis data permanen (pada implementasi makalah ini, digunakan memori untuk menyimpan data); maupun pengaturan *cookie* dan *session* untuk klien.

#### DAFTAR PUSTAKA

- [1] K. Chanda, "Password Security: An Analysis of Password," *I. J. Computer Network and Information Security*, 2016.
- [2] J. Anita, L. Kaspars dan I. Saudinis, "Multi factor authentications a necessary solution in the fight with information technology security threats," dalam *10th International Scientific and Practical Conference*, Rezekne, 2015.
- [3] Wikimedia Foundation, "Multi-factor authentication," 8 December 2016. [Online]. Available: [https://en.wikipedia.org/wiki/Multi-factor\\_authentication](https://en.wikipedia.org/wiki/Multi-factor_authentication).
- [4] Wikimedia Foundation, "FIDO Alliance," 10 12 2016. [Online]. Available: [https://en.wikipedia.org/wiki/FIDO\\_Alliance](https://en.wikipedia.org/wiki/FIDO_Alliance).
- [5] FIDO Alliance, "Universal 2nd Factor (U2F) Overview," 2014.
- [6] FIDO Alliance, "Specifications Overview," 2014.
- [7] Yubico, "FIDO U2F (UNIVERSAL 2ND FACTOR)," [Online]. Available: <https://www.yubico.com/about/background/fido/>. [Diakses 05 Mei 2017].
- [8] FIDO Alliance, "FIDO U2F Raw Message Formats," 2014.
- [9] FIDO Alliance, "FIDO U2F Specifications: Overview & Tutorial," 07 April 2016. [Online]. Available: <https://www.slideshare.net/FIDOAlliance/fido-u2f-specifications-overview-tutorial>. [Diakses 05 Mei 2017].
- [10] Yubico, "U2F Technical Overview," [Online]. Available: [https://developers.yubico.com/U2F/Protocol\\_details/Overview.html](https://developers.yubico.com/U2F/Protocol_details/Overview.html).
- [11] FIDO Alliance, "FIDO U2F & UAF Tutorial," 24 Maret 2016. [Online]. Available: <https://www.slideshare.net/FIDOAlliance/fido-u2f-uaf-tutorial>. [Diakses 05 Mei 2017].
- [12] Yubico, "Using a U2F library," [Online]. Available: [https://developers.yubico.com/U2F/Libraries/Using\\_a\\_library.html](https://developers.yubico.com/U2F/Libraries/Using_a_library.html).
- [13] FIDO Alliance, "FIDO U2F JavaScript API," 2016.
- [14] Yubico, "Key generation," [Online]. Available: [https://developers.yubico.com/U2F/Protocol\\_details/Key\\_generation.html](https://developers.yubico.com/U2F/Protocol_details/Key_generation.html).



UMMN

# Pengaruh Algoritma *Stemming* Nazief-Adriani Terhadap Kinerja Algoritma *Winnowing* Untuk Mendeteksi Plagiarisme Bahasa Indonesia

Hargyo Tri Nugroho I.

Program Studi Sistem Komputer, Universitas Multimedia Nusantara, Tangerang, Indonesia  
hargyo@umn.ac.id

Diterima 5 Mei 2017  
Disetujui 16 Juni 2017

**Abstract**—*Winnowing algorithm is one among many algorithms for detecting document similarity and plagiarism. Some studies show that Winnowing algorithm performs quite well. One form of plagiarism is paraphrase plagiarism. Paraphrase plagiarism can be done by changing sentence structure, changing vocabulary, and adding or changing affixes. Based on some of our previous experiments, detecting document resemblances can be enhanced by changing the words containing affixes to their basic words. In computer science, this technique is known as stemming - a technique to extract the basic word from an affixed word. Usually this technique is required in the filtering process to save storage media. For Indonesian, the Nazief-Adriani stemming algorithm is by far the most appropriate. This study examines how the effect of Nazief-Adriani stemming algorithm on Winnowing algorithm's performance against Indonesian texts. The results showed that the stemming process using Bloom-Filter on the Winnowing algorithm tends to decrease the similarity level achieved, but it accelerates processing time by approximately 30%.*

**Keywords**—*Algoritma Nazief-Adriani, Algoritma Winnowing, Bloom-Filter, Plagiat, Plagiat Checker*

## I. PENDAHULUAN

Kemajuan teknologi informasi yang pesat memungkinkan penggunaannya untuk berbagi informasi dengan cepat dalam berbagai format digital. Berbagai artikel berbahasa Indonesia dapat kita baca melalui laman *web* secara daring maupun mengunduhnya dalam format Ms. Word atau PDF secara luring. Dokumen-dokumen tersebut dapat dengan mudah kita temukan melalui mesin pencari maupun mengunjungi repository yang banyak tersedia di internet [1]. Kemudahan mengakses dokumen-dokumen yang tersedia dalam format digital ini tentu saja banyak memberi dampak positif namun juga mempermudah dilakukannya plagiasi.

Rawannya plagiat pada dokumen digital mendorong para peneliti untuk mengembangkan

piranti lunak *plagiarism checker* untuk mendeteksi plagiasi dengan cara mengukur tingkat kemiripan dokumen tersebut dengan dokumen-dokumen lainnya. Hal ini penting karena kredibilitas akademisi maupun penerbit ditentukan oleh originalitas artikel yang diterbitkan [2]. Salah satu algoritma yang banyak digunakan adalah algoritma *Winnowing*. Algoritma ini sederhana namun cukup dapat diandalkan untuk mendeteksi plagiat [3].

Bahasa Indonesia memiliki karakteristik yang berbeda dari bahasa yang lain. Penambahan imbuhan (afiksasi) pada sebuah leksem (morfem dasar) dapat dilakukan dengan berbagai cara; sebagai awalan (prefix), akhiran (sufix), maupun sisipan (infix) [4].

Afiksasi pada leksem tidak hanya merubah bentuknya, namun juga maknanya secara gramatikal, sedangkan maknanya yang semula yaitu makna leksikal sedikit banyak tidak berubah [5].

Plagiat parafrase dapat dilakukan dengan mengubah struktur kalimat, merubah kosakata, serta menambah atau mengubah imbuhan [6]. Berdasarkan beberapa percobaan kami sebelumnya, kemiripan dokumen dapat ditingkatkan dengan mengubah kata-kata yang mengandung imbuhan menjadi kata dasarnya. Pada ilmu komputer, teknik ini dikenal sebagai *stemming* yaitu suatu teknik untuk mengekstraksi kata dasar dari suatu kata [7]. Umumnya teknik ini diperlukan pada proses *filtering* untuk menghemat media penyimpanan. Untuk bahasa Indonesia, algoritma *stemming* Nazief-Adriani sejauh ini dirasa paling sesuai [8]. Paper ini membahas bagaimana pengaruh algoritma *stemming* Nazief-Adriani terhadap kinerja algoritma *Winnowing* terhadap teks-teks berbahasa Indonesia.

## II. RISET SEBELUMNYA

Kinerja Algoritma WInnowing dibandingkan algoritma *fingerpint* oleh [9]. Hasil pengujian menunjukkan bahwa walaupun secara kinerja algoritma WInnowing (91.8%) tidak sebaik algoritma *fingerpint* (92.8%), namun algoritma WInnowing memiliki tingkat relevansi topik yang lebih baik daripada algoritma *fingerpint*.

Algoritma WInnowing diimplementasikan oleh Ryan et al. [10] menggunakan metode K-Gram dipadukan dengan Synonim Analysis untuk meningkatkan akurasi *plagiarism checker*. Hasilnya sangat memuaskan khususnya terhadap artikel jiplakan yang banyak mengalami perubahan kosakata. Selain itu algoritma WInnowing juga menunjukkan kinerja yang baik pada [11] yang mengimplementasikannya pada jurnal *online*.

Teknik *stemming* digunakan oleh Sagala [12] dengan memadukan algoritma Enhanced Confix Stripping bersama dengan algoritma WInnowing namun hasilnya perpaduan ini justru menghasilkan kinerja yang kurang baik. Namun, hasil yang berbeda ditemukan oleh Alfikri et al. [13] yang menyatakan bahwa *stemming* akan memperbaiki kinerja sistem namun tidak signifikan. Perbedaan hasil ini mengarah pada suatu hipotesa bahwa pengaruh *stemming* terhadap kinerja algoritma *wInnowing* akan berbeda-beda tergantung beberapa kondisi yang akan diteliti pada penelitian ini.

## III. TINJAUAN PUSTAKA

### A. Algoritma WInnowing

Algoritma WInnowing [3] adalah algoritma untuk menghasilkan suatu deret bilangan unik (*fingerpint*) yang mewakili suatu dokumen. Dengan *fingerpint* tersebut kita bisa mengetahui tingkat kemiripan satu dokumen dengan dokumen yang lain.

Secara garis besar, algoritma WInnowing bekerja sebagai berikut:

1. Penghapusan karakter-karakter yang tidak relevan (*whitespace insensitivity*).
2. Pembentukan rangkaian gram dengan ukuran k.
3. Penghitungan nilai hash.
4. Membagi ke dalam window tertentu.
5. Pemilihan beberapa nilai hash menjadi *fingerpint* dokumen

### B. Algoritma Stemming Nazief-Adriani

*Stemming* adalah cara yang digunakan untuk meningkatkan performa *Information Retrieval* dengan cara mentransformasikan kata-kata dalam sebuah dokumen teks ke kata dasarnya [8].

Proses *stemming* pada teks Bahasa Indonesia digunakan untuk menghilangkan sufiks, konfiks, dan prefiks. Hal ini berbeda dengan teks Bahasa Inggris,

di mana *stemming* digunakan untuk menghilangkan sufiks [8].

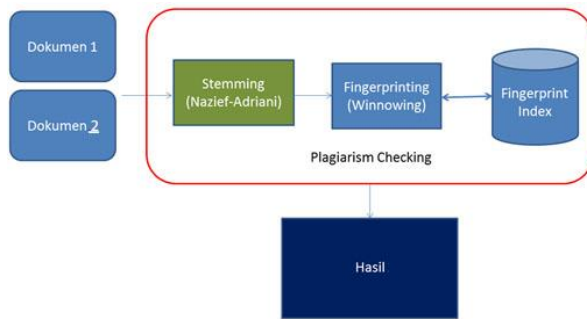
Algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani memiliki tahapan sebagai berikut:

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tersebut adalah root word. Maka algoritma berhenti.
2. Inflection Suffixes (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa particles (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possesive Pronouns (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus Derivation Suffixes (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
  - a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
  - b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus Derivation Prefix. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
  - a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
  - b. For  $i = 1$  to 3, tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Melakukan Recoding.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai root word. Proses selesai.

## IV. ARSITEKTUR SISTEM

Seperti terlihat pada gambar 1, pada penelitian ini algoritma WInnowing diimplementasikan dengan diawali dengan preprocessing. Penelitian ini menggunakan dua jenis preprocessing yaitu *case folding*, dan *stemming*. Pengujian akan menggunakan dua jenis implementasi preprocessing yaitu *case folding* saja, dan *case folding* ditambah dengan *stemming* untuk mengetahui pengaruh *stemming* terhadap hasil akhir algoritma WInnowing.

Proses ini *stemming* dimulai ketika *Nazief-Adriani Stemmer* menerima kalimat dari *case folding*. Kalimat ini merupakan rangkaian kata tanpa huruf kapital. Pada sebagian besar implementasi yang ditemui, kalimat ini kemudian diolah dengan menggunakan aturan pembuangan imbuhan dan pencocokan kata dengan data yang ada di dalam kamus atau basis data. Pencocokan kata pada database ini memiliki kelemahan yaitu lamanya waktu yang dibutuhkan untuk melakukan *string matching*.



Gambar 1. Arsitektur Sistem

Pada penelitian ini diajukan solusi berupa proses tambahan, yaitu dengan menggunakan *Bloom Filter* [14]. Pemanfaatan *Bloom Filter* ini dimulai dengan membuat suatu ruang penyimpanan di dalam memori yang memiliki ukuran sesuai dengan jumlah data dari tabel di database. Kemudian ruang penyimpanan ini diisi dengan nilai hashing dari setiap kata dasar yang ada di database. Setelah itu proses *Bloom Filter* dapat digunakan oleh proses *Nazief-Adriani Stemmer* untuk melakukan pencocokan kata dasar. Dengan demikian proses pencarian kata dasar tidak lagi menggunakan database, namun menggunakan struktur data *Bloom Filter* yang siap pakai pada RAM.

Setelah preprocessing, file teks dibentuk menjadi rangkaian substring senilai  $k$  atau  $k$ -gram. Nilai dari rangkaian substring selanjutnya diproses menjadi rangkaian hash. Hashing adalah proses untuk mengubah string menjadi bilangan integer yang disebut nilai hash. Proses pengubahan menjadi nilai hash menggunakan fungsi rolling hash. Rolling hash merupakan salah satu metode hashing yang menghitung nilai hash dengan tanpa mengulangi semua string.

Setelah rangkaian hash terbentuk, proses selanjutnya adalah pembentukan window dari rangkaian hash. Window merupakan substring dari nilai hash sepanjang  $w$ gram. Proses ini menghasilkan *fingerprint* yang digunakan untuk menentukan tingkat kemiripan dokumen.

## V. HASIL PENGUJIAN

### A. Pengaruh Penggunaan Bloom-Filter

Percobaan dilakukan dengan menggunakan 4 buah artikel yang dipilih secara acak berisikan kalimat yang merupakan rangkaian kata dasar maupun kata berimbuhan. Seperti terlihat pada tabel 1, rata-rata kecepatan pengolahan (*processing speed*) *stemming* menggunakan database adalah 9,15 kata/detik. Sedangkan apabila menggunakan *Bloom-Filter* kecepatan pengolahan meningkat sangat tajam menjadi rata-rata 8039,73 kata/detik. Hal ini membuat perbedaan waktu yang semakin signifikan, berbanding lurus dengan jumlah kata dalam artikel.

*Bloom-Filter* lebih cepat diakses karena berada di RAM. Pencocokan datanya pun tidak dilakukan dengan *string matching* melainkan melalui proses *hashing*. Tentu saja ini sangat jauh berbeda dengan database yang berada di hard disk yang memiliki *access time* yang jauh lebih lambat dibanding RAM.

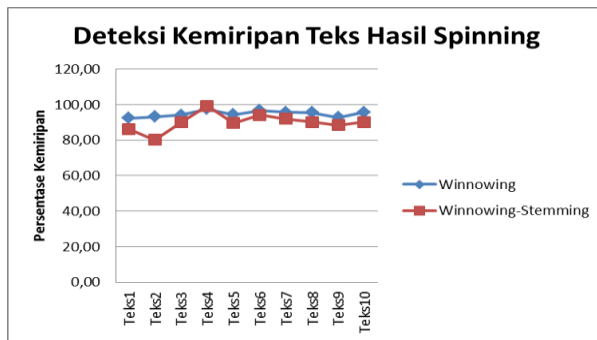
Tabel 1. Perbandingan Kecepatan Stemming Menggunakan Database vs Bloom Filter

### B. Pengaruh Stemming Nazief-Adriani Pada Winnowing

Gambar 2 menunjukkan hasil pengujian algoritma *Winnowing - Stemming* untuk mendeteksi kemiripan artikel yang direkayasa menggunakan teknik *spin*,

File	Jumlah Kata	Database		Bloom Filter	
		Waktu	Kecepatan (kata/detik)	Waktu	Kecepatan (kata/detik)
Teks3	651,00	65,86	9,88	0,09	7233,33
Teks5	712,00	76,69	9,28	0,10	7340,21
Teks6	460,00	54,76	8,40	0,05	10000,00
Teks9	622,00	68,85	9,03	0,08	7585,37
<b>Rata-rata</b>			<b>9,15</b>		<b>8039,73</b>

yaitu mengganti beberapa kata secara acak dengan sinonimnya. Rekayasa ini tanpa merubah struktur kalimat. Dapat dilihat bahwa algoritma *Winnowing* mampu mendeteksi kemiripan artikel yang dibuat dengan teknik *spinning* sebesar rata-rata 94,84% sedangkan algoritma *Winnowing - Stemming* mendeteksi kemiripan artikel sebesar rata-rata 90,03%.



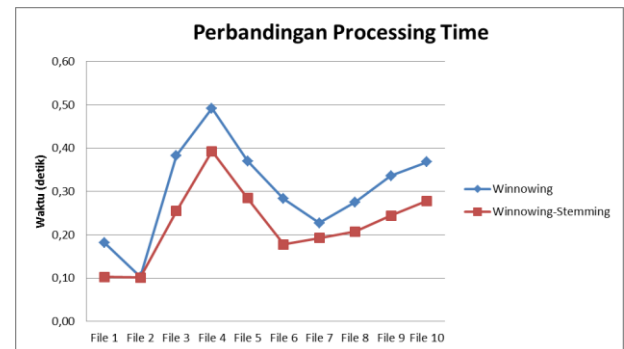
Gambar 2. Deteksi Kemiripan Terhadap Teks Hasil Spinning

Hasil yang berbeda ditunjukkan gambar 3. Pada percobaan ini beberapa artikel diparafrase menggunakan teknik *spin* dan perubahan struktur kalimat. Perubahan struktur kalimat yang dimaksud mencakup mengubah kalimat dari pasif menjadi aktif (dan sebaliknya), melakukan penambahan imbuhan, serta menambahkan kata penghubung.. Algoritma Winnowing-Stemming rata-rata mendeteksi kemiripan dokumen yang telah dirubah struktur kalimatnya sebesar 45,60%. Sedangkan algoritma Winnowing murni mampu mendeteksi kemiripan di atas 60%.



Gambar 3. Deteksi Kemiripan Terhadap Teks Hasil Parafrese

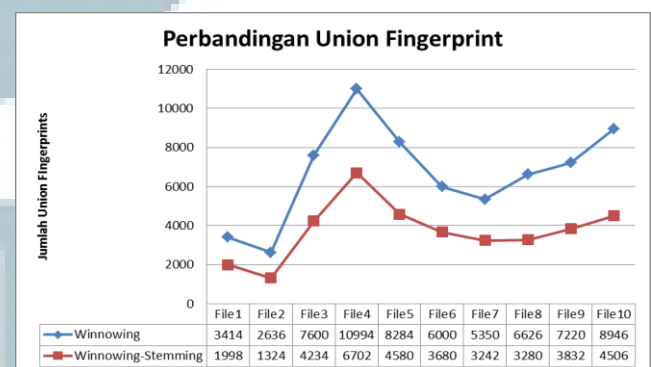
Untuk plagiasi berbasis parafrase yang merubah struktur kalimat, baik algoritma Winnowing murni maupun algoritma Winnowing-Stemming menunjukkan kinerja yang lebih rendah dibandingkan pada saat deteksi kemiripan dokumen berbasis spinning. Hal ini menunjukkan bahwa, dengan merubah struktur kalimat, *n-gram* maupun *fingerprints* yang terbentuk lebih berbeda dibandingkan hanya sekedar mengganti kata-kata dengan sinonimnya saja. Jumlah *fingerprints* yang semakin sedikit, memperkecil peluang munculnya *fingerprints* yang sama. Sehingga hasil kalkulasi koefisien Jaccard akan lebih kecil.



Gambar 4. Perbandingan Processing Time Algoritma Winnowing vs Winnowing - Stemming

Seperti terlihat di gambar 4, *stemming* membuat algoritma Winnowing berjalan lebih cepat. Hal ini disebabkan karena berkurangnya jumlah karakter yang harus diproses lebih lanjut hingga menjadi *fingerprints*. Terlihat pada gambar 5, *stemming* memotong jumlah *union fingerprint* yang linear dengan penurunan processing time apabila dibandingkan dengan algoritma Winnowing murni.

Berdasarkan pengujian di atas maupun analisis algoritma, algoritma Winnowing memiliki potensi perbaikan terutama dalam proses pembentukan *n-gram* yang memakan waktu relatif tinggi.



Gambar 5. Perbandingan Jumlah Union Fingerprints

Seluruh pengujian di atas menggunakan *k-gram* 5 dan *w-gram* 4. Baik *w-gram*, maupun *k-gram* memiliki pengaruh terhadap hasil *similarity* maupun waktu proses (*processing time*). Semakin besar *w-gram*, maupun *k-gram*, semakin lama waktu proses yang diperlukan. Namun, pada penelitian ini *k-gram* lebih menentukan tingkat *similarity* maupun waktu prosesnya. Semakin kecil nilai *k-gram* maka semakin kecil jumlah karakter yang akan dicocokkan dan semakin sering rangkaian karakter tersebut akan ditemukan dalam teks.

## VI. KESIMPULAN

Dari hasil pengujian dan analisis maka dapat disimpulkan beberapa hal sebagai berikut: Algoritma Winnowing sangat efektif untuk mendeteksi plagiarisme dokumen baik dengan teknik *spinning* maupun parafrase yang merubah struktur kalimat. Proses *stemming* pada algoritma Winnowing cenderung menurunkan tingkat *similarity* yang dicapai, namun mempercepat processing time kurang lebih sebesar 30%. Penggunaan Bloom-Filter dalam proses *stemming* terbukti efektif untuk mempercepat processing time sekitar 1000 kali lebih cepat.

## DAFTAR PUSTAKA

- [1] Ali, A.E.T., H.D. Abdulla and V. Snasel, "Survey of plagiarism detection methods", Proceedings of the 5th Asia Modelling Symposium, May 24-26, 2011, Manila, Philippines, pp: 39-42.
- [2] Khan, M.A., A. Aleem, A. Wahab and M.N. Khan, "Copy detection in Urdu language documents using n-grams model", Proceedings of the International Conference on Computer Networks and Information Technology, July 11-13, 2011, Abbottabad, Pakistan, pp: 263-266.
- [3] Schleimer, S., Wilkerson, D. S., & Aiken, A., "Winnowing: local algorithms for document fingerprinting", Proceedings of the 2003 ACM SIGMOD international conference on Management of data (pp. 76-85). ACM 2003.
- [4] Kridalaksana, H., "Pembentukan kata dalam bahasa Indonesia", Gramedia pustaka utama, 1989.
- [5] Purnanto, Dwi, "Kajian morfologi derivasional dan infleksional dalam bahasa Indonesia", Kajian Linguistik dan Sastra, Vol. 18, No. 35, 2006: 136-152
- [6] Clough, P., "Old and New Challenges in Automatic Plagiarism Detection", Sheffield, UK : Department of Information Studies, University of Sheffield, 2003.
- [7] Utomo, M. S., & Winarko, E., "Design And Implementation of Document Similarity Search System For WEB-Based Medical Journal Management", IJCCS-Indonesian Journal of Computing and Cybernetics Systems, 5(1), 2013.
- [8] Agusta, Ledy, "Perbandingan Algoritma *Stemming* Porter dengan Algoritma Nazief & Adriani untuk *Stemming* Dokumen Teks Bahasa Indonesia", Prosiding Konferensi Nasional Sistem dan Informatika, November 14th, 2009, Bali, Indonesia, pp: 196-201
- [9] Wibowo, A. T., Sudarmadi, K. W., & Barmawi, A. M., "Comparison between fingerprint and winnowing algorithm to detect plagiarism fraud on Bahasa Indonesia documents", Information and Communication Technology (ICoICT), 2013 International Conference of (pp. 128-133). IEEE, March 2013.
- [10] Ryan, Mudafiq, E.B. Cahyono, & G.I. Marthasari, "Aplikasi Pendeteksi Duplikasi Dokumen Teks Bahasa Indonesia Menggunakan Algoritma Winnowing Dengan Metode K-Gram Dan Synonym Recognition", Universitas Muhammadiyah, Malang. Tersedia daring: <http://www.mudafiqriyan.net/wp-content/uploads/2012/03/Mudafiq-AplikasiPendeteksiDuplikasiDokumen.pdf>
- [11] Kharisman, O., Susanto, B., & Suwarno, S., "IMPLEMENTASI ALGORITMA WINNOWER UNTUK MENDETEKSI KEMIRIPAN PADA DOKUMEN TEKS", Jurnal Informatika, 9(1), 2013.
- [12] Sagala, A. C. S., "Pendeteksian Kesamaan Pada Dokumen Teks Menggunakan Kombinasi Algoritma Enhanced Confix Stripping Dan Algoritma Winnowing", 2009.
- [13] Alfikri, Z. F., & Purwarianti, A., "The Construction Of Indonesian-English Cross Language Plagiarism Detection System Using Fingerprinting Technique", Jurnal Ilmu Komputer dan Informasi, 2012, 5(1), 16-23.
- [14] Mullin, James K. "A second look at Bloom filters." Communications of the ACM 26.8 (1983): 570-571.



# Rancang Bangun Mekanisme *Load Sharing* Pada *Link Aggregation* Menggunakan *Software Defined Networking*

Samuel<sup>1</sup>, Cahyo Eko Samudera<sup>2</sup>

Computer Engineering, Universitas Multimedia Nusantara, Tangerang, Indonesia  
samuel.hutagalung@umn.ac.id<sup>1</sup>, cahyosamudera@gmail.com<sup>2</sup>

Diterima 14 Mei 2017

Disetujui 16 Juni 2017

**Abstract**— Given the level of activity of Internet users is high on lately. Impact on the growing needs for server throughput can be treated with load sharing on link aggregation. However, it possess a problem for UDP protocol data packets which will experience jitter and packet received out of order high. Therefore, the author tries to create an algorithm to share the package which will be sent to the server with weighted round robin method that focuses UDP Protocol data packets. The author uses the Software Defined Networking (SDN) and OpenFlow protocol that is capable of directly reprogrammed network devices. Host connect by sending packets Internet Control Message Protocol (ICMP), Transport Control Protocol (TCP) and User Datagram Protocol (UDP) to the server, then do an analysis of the bandwidth, jitter, datagrams, and retry. The author has successfully implemented with the emulator Mininet and testing. The results indicate that the average jitter is able to be reduced to 50% and packets received out of order is reduced to 0 compared to standard link aggregation with weighted round robin load sharing method.

**Keywords**— Software Defined Networking, Link Aggregation, Weighted Round Robin, UDP.

## I. PENDAHULUAN

Perangkat jaringan komputer, seperti Router dan Switch memiliki Management Interface, sehingga dapat dikonfigurasi dan dikelola oleh operator. Command Line Interface (CLI) dan Graphical User Interface (GUI) merupakan contoh dari Management Interface [1]. Management Interface merupakan salah satu fungsi dari Sistem Operasi yang terdapat pada perangkat jaringan komputer. Sistem Operasi dan fungsi-fungsi di dalamnya bersifat embedded. Dengan demikian, tidak memungkinkan apabila seseorang ingin mengubah atau mengoptimalkan cara kerja dari suatu fungsi dalam perangkat jaringan komputer.

Dengan melakukan pemisahan terhadap data-plane dan control-plane pada perangkat jaringan komputer, Sistem Operasi dan cara kerja dari fungsi-fungsi di dalamnya dapat diubah sesuai dengan yang diinginkan. Pemisahan inilah yang mendasari terbentuknya paradigma baru dalam jaringan komputer yang disebut dengan Software Defined Networking (SDN) [2].

Dengan cara ini, pengguna SDN dapat membuat sendiri sebuah fungsi pada perangkat jaringan komputer yang akan dijalankan pada Platform Controller. Platform Controller juga memiliki Application Programming Interfaces (APIs) untuk mempermudah pengguna dalam mengimplementasikan fungsi yang dibuat pada perangkat jaringan komputer atau mengubah cara kerja dari fungsi yang sudah ada sebelumnya. Dengan menggunakan SDN, diharapkan ketergantungan terhadap vendor tertentu bisa diminimalisir, memungkinkan adanya fleksibilitas dalam hal modifikasi fungsi pada perangkat jaringan komputer sesuai dengan kebutuhan [16].

Melihat tingkat aktivitas dari pengguna internet yang tinggi, tentu saja hal ini akan berdampak pada penyedia informasi. Kinerja web & database server sebagai penyedia layanan konten selalu diharapkan dapat memenuhi semua kebutuhan dari pengguna. Penerapan Link Aggregation sangat diperlukan untuk meningkatkan throughput dan menyediakan fault tolerance apabila ada Link yang bermasalah. Load Sharing pada Link Aggregation merupakan teknik yang berfungsi untuk membagi beban traffic kepada beberapa Link dengan tujuan agar Link tidak mengalami overload [3].

Salah satu metode Load Sharing yang populer digunakan adalah Round Robin. Metode Round Robin ini merupakan metode Load Sharing yang bersifat statis dan selalu balance. Metode Round Robin memiliki algoritma penjadwalan yang kurang efektif untuk diterapkan pada dunia nyata, karena setiap Link akan memiliki penggunaan yang berbeda. Metode lain yang mampu menutupi kelemahan ini adalah Weighted Round Robin. Algoritma penjadwalan pada Weighted Round Robin hampir sama dengan Round Robin, perbedaannya metode ini dapat memberikan prioritas terhadap Link yang digunakan.

Beberapa contoh jenis paket data yang dapat dikirimkan lewat Link Aggregation dengan cara Load Sharing berupa Internet Control Message Protocol (ICMP) Packet, Transport Control Protocol (TCP)

based Packet dan User Datagram Protocol (UDP) based Packet. Akan tetapi terdapat masalah pada paket yang menggunakan protokol UDP ketika dikirimkan dengan cara Load Sharing. Hal ini disebabkan karena protokol UDP tidak memiliki sequence number sehingga paket yang menggunakan protokol UDP tidak dapat disusun kembali sesuai dengan urutan pengiriman apabila sampai di tujuan dengan urutan acak. Protokol UDP umumnya digunakan oleh aplikasi ringan yang membutuhkan kecepatan komunikasi dan tidak transaksional, seperti Domain Name System (DNS) dan Real-Time Transport Protocol (RTP). Sebagaimana diketahui bahwa DNS digunakan untuk menerjemahkan nama domain menjadi alamat IP, sedangkan RTP digunakan untuk mengirimkan audio dan video melalui jaringan berbasis IP.

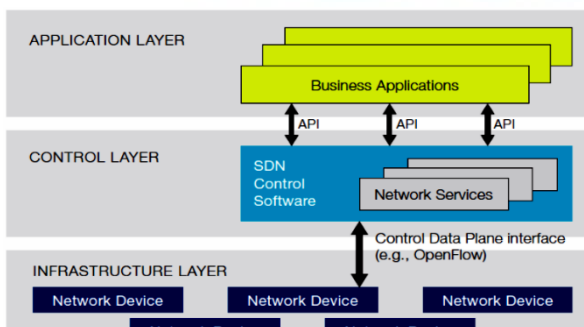
Dengan latar belakang dan permasalahan tersebut, penulis melakukan penelitian untuk merancang mekanisme Load Sharing pada Link Aggregation dengan metode Weighted Round Robin yang memperhatikan UDP based Packet pada Software Defined Networking.

II. TINJAUAN PUSTAKA

Di dalam bab ini, penulis mengambil beberapa tinjauan pustaka yang dijadikan sebagai landasan pengerjaan aplikasi Link Aggregation dengan Load Sharing bermetode Weighted Round Robin pada Software Defined Networking (SDN) dengan menggunakan Ryu Controller.

A. Software Defined Networking

Software Defined Networking (SDN) merupakan sebuah pendekatan arsitektur jaringan komputer yang memisahkan control-plane dari sebuah perangkat jaringan komputer (Switch atau Router) dengan data-plane perangkat jaringan komputer tersebut [4]. Pemisahan data-plane dan control-plane ini memungkinkan untuk memprogram perangkat tersebut sesuai dengan yang diinginkan secara terpusat (SDN Controller), sehingga hal ini memungkinkan untuk mengontrol, memonitor, dan mengatur sebuah jaringan komputer dari sebuah titik (node) terpusat tersebut.

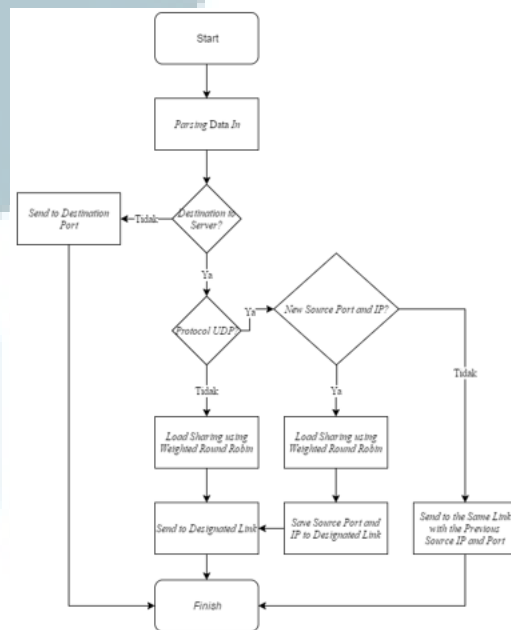


Gambar 1. Ilustrasi dari Arsitektur SDN [2]

Melalui OpenFlow, kita tidak hanya dapat melakukan flow forwarding berbasis network layer

tetapi juga dapat dilakukan pengaturan pergerakan paket data secara terpusat mulai dari layer 2 sampai layer 7 forwarding (flow granularity), sehingga aliran paket data di jaringan dapat diprogram secara independen [5]. Hal ini dapat dilakukan dengan membuat algoritma dan forwarding rules-nya pada controller kemudian aturan tersebut didistribusikan ke switch yang ada di jaringan. Terdapat beberapa OpenFlow Controller yang dapat digunakan seperti NOX (C base), Ryu (python base), dan Floodlight (java base).

Proses aliran paket data pada Switch Openflow [6] yang pertama, Ketika sebuah paket data tiba di switch OpenFlow, bagian header paket diperiksa berdasarkan entri flow table. Lalu, jika entri yang sesuai ditemukan, Switch kemudian menerapkan instruksi-instruksi terkait berdasarkan aliran paket data yang sesuai juga. Terakhir, jika tidak ada yang sesuai dengan prosedur flow table, maka aliran paket data akan diarahkan ke entri table-miss. Entri table-miss adalah entri yang diperlukan yang menentukan set instruksi yang akan diterapkan terhadap paket data yang masuk ketika tidak ada yang cocok atau sesuai dengan prosedur flow table. Instruksi mencakup: Menghapus (dropping) paket; Mengirim paket pada semua interface; dan Meneruskan paket ke controller.



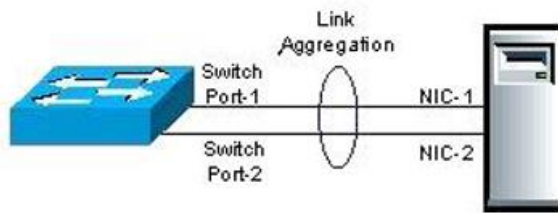
Gambar 2. Flowchart Proses Pengiriman Data [6]

B. Link Aggregation

Link Aggregation (LAG) adalah bahasa standar teknologi jaringan komputer yang menggabungkan beberapa link/trunk/cable/port fisik secara paralel untuk mendapatkan kapasitas Bandwidth yang lebih besar dan untuk memberikan redundancy dengan menggunakan teknologi Ethernet. Beberapa vendor memiliki standar masing-masing untuk penerapan LAG, namun standar internasional yang umum digunakan



untuk *LAG* adalah *IEEE802.3ad*. Istilah lain *LAG* ini adalah *Link Aggregation Control Protocol (LACP)*.



Gambar 3. *Link Aggregation* pada *Single Server* [11]

Penggabungan beberapa *link* menggunakan *LAG* harus dari *link* yang memiliki kapasitas yang sama, seperti sesama 10Mb/s atau sesama 100Mb/s. Kapasitas yang lain yang memungkinkan untuk digabungkan adalah 1Gb/s dan 10Gb/s.

Kombinasi dari berbagai kapasitas tidak dimungkinkan. Peningkatan kapasitas dalam *LAG* sesuai dengan jumlah *link* yang digunakan terhadap kapasitas individu *link*. Misal, penggabungan tiga *link* 10Mb/s akan mendapatkan 30Mb/s. Begitu juga dengan penggabungan lima *link* 100Mb/s akan mendapatkan kapasitas 500Mb/s. Proteksi *link* menggunakan *LAG* tidak dimaksudkan untuk menambah kapasitas tetapi adalah untuk *high availability*.

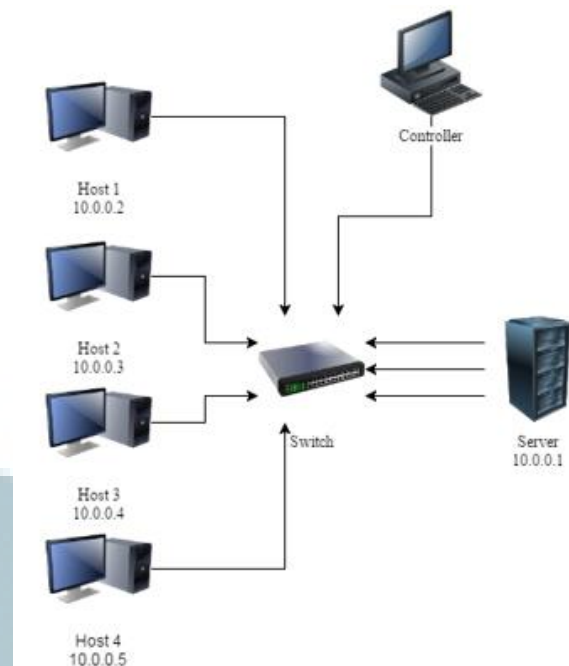
### C. *Weighted Round Robin*

Penjadwalan *Weighted Round Robin* merupakan evolusi dari penjadwalan *Round Robin*. Setiap *Link* dapat diberi bobot berupa *integer* yang mengindikasikan kapasitas performanya. *Link* yang mempunyai bobot lebih tinggi akan mendapatkan jatah terlebih dahulu dari pada *Link* yang mempunyai bobot lebih rendah dan *Link* yang mempunyai bobot lebih besar akan mendapatkan jatah lebih banyak dibandingkan dengan *Link* yang mempunyai bobot lebih kecil [13].

Satu siklus *WRR* dapat diketahui dari total bobot prioritasnya. Contoh, terdapat tiga *Link* yaitu A, B, dan C yang masing-masing mempunyai bobot 4, 3, dan 2. Maka, 1 siklus *WRR* adalah 9 dari penjumlahan  $4 + 3 + 2 = 9$ . Proses distribusi pembagian jatah dimulai dari prioritas tertinggi ke terendah dengan proses giliran *Round Robin*. Penulis menggunakan teknik *smooth service* yang terkenal lebih baik [13]. Oleh karena itu, urutan penjadwalan sesuai dengan *smooth service* untuk kasus ini adalah AABABCABC untuk setiap 1 siklus. Apabila prioritas diatur menjadi sama maka akan menjadi *balanced*. Singkatnya, penjadwalan *Round Robin* adalah penjadwalan *Weighted Round Robin* yang menganggap semua bobot *Link* sama.

## III. PERANCANGAN SISTEM

Pada bab ini akan dijelaskan tentang proses perancangan sistem yang digunakan dalam penelitian.



Gambar 4. Rancangan Topologi Sistem

### A. *Analisis Pemilihan Link*

Topologi yang akan dibuat memiliki jumlah *Link* ditentukan sebanyak empat unit *Link*, satu unit *Server*, satu unit *Switch*, dan satu unit *Controller*. Semua perangkat ini akan disimulasikan dalam emulator *Mininet*. Selanjutnya, analisa dilakukan terhadap paket data dengan protokol *TCP* dan *UDP* pada *Link Aggregation* dengan metode *Load Sharing Weighted Round Robin*.

Pada proses pemilihan *Link*, paket data yang masuk dianalisis dan dikategorikan menjadi *UDP based Packet* atau *non-UDP based Packet*. Berdasarkan topologi pada Gambar 4, ada satu unit *Server* yang terhubung dengan satu unit *Switch* yang dikendalikan oleh satu unit *Controller*. Semua perangkat tersebut akan disimulasikan dalam emulator *Mininet*.

Setiap data yang masuk ke *Switch* akan di-parse untuk bisa didapatkan detail yang lebih jauh. Pada saat proses *parsing* setiap data akan diperiksa tipe protokol, alamat *IP*, dan *MAC* tujuan dan sumbernya. Layaknya tukang pos, setelah diketahui tujuan dan sumbernya, apabila tujuan sebuah paket tersebut tidak menuju ke *Server*, maka paket akan dikirim langsung ke *port* tujuan sesuai dengan *CAM table* yang telah dipelajari oleh *Switch*. Sedangkan apabila tujuan dari suatu paket ke *Server* maka paket harus diproses lebih lanjut.

Semua paket yang menuju *Server* akan diperiksa, apakah menggunakan protokol *UDP* atau tidak. Apabila protokol tersebut menggunakan *UDP* maka sebelum dilakukan *Load Sharing* akan diperiksa sumber *IP* dan *port* dari paket tersebut. Jika paket data tersebut menggunakan *UDP* dan alamat sumber *IP* dan *portnya* belum pernah dikenali oleh *Switch* maka

alamat sumber IP tersebut akan disimpan di *Link* bersangkutan setelah proses *Load Sharing*. Sedangkan apabila paket data tersebut berprotokol *UDP* dan alamat sumber IP dan *port*nya telah dikenali sebelumnya. Maka paket data tersebut langsung dikirimkan ke *Link* yang sama dengan alamat IP dan *port* sumber yang sebelumnya tanpa harus melalui proses *Load Sharing*.

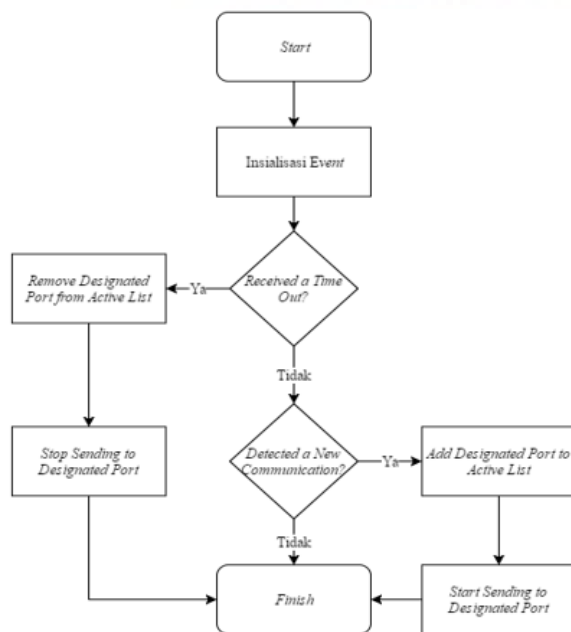
Tetapi jika dari awal paket data tersebut memang bertujuan ke *Switch* sedangkan memiliki tipe protokol yang *non-UDP* maka setiap paket data tersebut harus melalui proses *Load Sharing* sebelum dikirim ke *Link* tujuan.

Tabel 1. Alamat IP dan MAC *Host*

Nama <i>Host</i>	Alamat <i>IP</i>	Alamat <i>MAC</i>
H1	10.0.0.2	00:00:00:00:00:22
H2	10.0.0.3	00:00:00:00:00:23
H3	10.0.0.4	00:00:00:00:00:24
H4	10.0.0.5	00:00:00:00:00:25

B. *Event Handling*

Jika pada situasi ketika suatu *Link* aktif atau mati secara tiba-tiba yang sedang dalam kondisi *Link Aggregation*, maka *Ryu Controller* bisa mengirim suatu *event* berupa “*EventSlaveStateChanged*”. *Event* tersebut akan berisi bagaimana *Ryu Controller* mengatasi masalah tersebut pada kondisi *Link Aggregation* untuk menjamin kelangsungan komunikasi dan meningkatkan *fault tolerance*. Oleh karena itu, Dengan Gambar 5 di bawah ini, penulis memaparkan aplikasi untuk *Ryu Controller*.



Gambar 5. *Flowchart Event Handler*

*Event* bisa terpicu ketika terjadi *event* yang bisa berupa *request time out* atau pun terdeteksi *Link* baru. Pada suatu ketika ada *Link* yang secara tiba-tiba berhenti berkomunikasi dan melewati batas waktu maka secara otomatis *Switch* akan menganggap telah terjadi *event time out* yang memicu *Ryu Controller* mengirim suatu *event* berupa “*EventSlaveStateChanged*”. Lalu *Ryu Controller* menghapus *port* yang dianggap mati tersebut dari *active list* kemudian memerintahkan *Switch* untuk tidak mengirim apapun melalui *port* tersebut. Apabila suatu ketika ada *Link* yang tiba-tiba berkomunikasi kembali, *Switch* akan memberitahu *Controller* bahwa ada *Link* yang mencoba berkomunikasi. *Ryu Controller* akan segera menambahkan *port* dimana *Link* yang sedang mencoba berkomunikasi itu ke *active list* kemudian memerintahkan *Switch* untuk bisa mengirim paket data melalui *port* tersebut.

IV. IMPLEMENTASI DAN PENGUJIAN

Dalam proses implementasi ini, hal yang pertama dilakukan ialah masing-masing *Host* akan melakukan *ping* pada alamat *IP* 10.0.0.1, yang merupakan alamat *Server* untuk memastikan semua *Host* bisa berkomunikasi dengan *Server*. Selanjutnya, pengiriman paket data *TCP* dan *UDP* yang diikuti dengan pengujian *fault tolerance* dan analisis data.

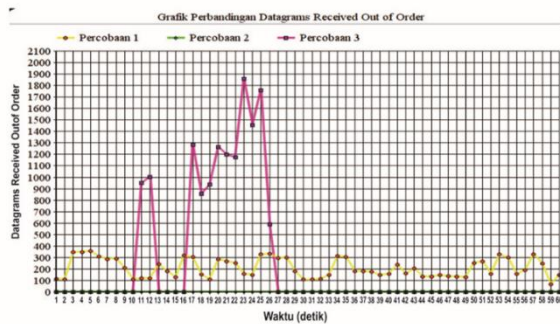
A. *Pengujian Terhadap Paket User Datagram Protocol*

Proses pengujian terhadap paket data *UDP* akan dilakukan dengan tool *Iperf* dari *Host* ke *Server*. *Host* akan mengirim paket *UDP* selama 60 detik melalui port tujuan 3333 dengan bandwidth 10Mbps. Informasi tentang percobaan *UDP* dipaparkan pada Tabel 2 di bawah ini.

Tabel 2. Tabel Percobaan *UDP*

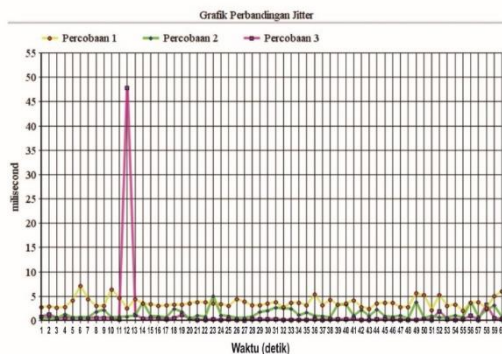
No.	Percobaan
1	Metode <i>Load Sharing</i> dengan <i>Weighted Round Robin</i> tanpa menggunakan algoritma yang penulis buat.
2	Metode <i>Load Sharing</i> dengan <i>Weighted Round Robin</i> dengan menggunakan algoritma yang penulis buat.
3	Sama seperti nomor kedua namun ditambah apabila terjadi ada <i>Link</i> yang mati pada detik ke-5 dan aktif pada detik ke-16.

Hasil dari pengujian tersebut ditampilkan dengan grafik pada gambar 6, 7, dan 8.



Gambar 6. Grafik Perbandingan *Datagrams Received Out of Order*

Dari Gambar 6 di muka terlihat bahwa percobaan pertama menghasilkan semua pengiriman paket UDP sampai ke Server selalu out of order. Pada percobaan kedua semua paket UDP yang sampai ke Server tidak ada yang mengalami out of order. Pada percobaan terakhir, paket UDP mulai mengalami out of order ketika di detik ke-5. Link pertama yang menjadi jalur awal untuk mengirim paket UDP mengalami masalah. Di detik ke-13, *Switch* telah selesai *learning* dan menggunakan *Link* lainnya untuk mengirim paket UDP sehingga tidak ada paket yang mengalami out of order. Kemudian di detik ke-16, *Link* pertama terdeteksi kembali aktif yang menyebabkan *Switch* kembali *learning* untuk memilih jalur untuk pengiriman paket UDP hingga detik ke-27. Mulai detik ke-28, *Switch* telah selesai *learning* sehingga tidak ada paket yang mengalami out of order seperti semula.

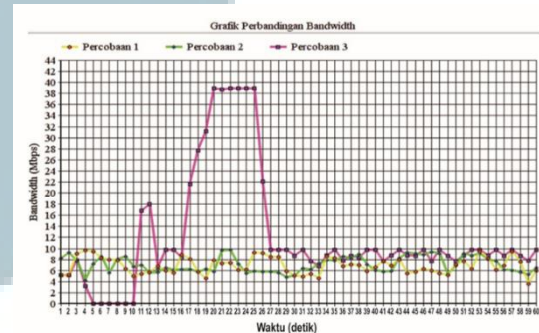


Gambar 7. Grafik Perbandingan *Jitter*

Dari Gambar 7 terlihat bahwa *Jitter* pada percobaan pertama rata-rata lebih tinggi hingga 100% dari percobaan kedua. *Jitter* adalah variasi jeda dalam *milliseconds* antar paket data yang diterima. Pada jaringan yang sehat semakin kecil nilai *Jitter* semakin bagus. Semakin tinggi nilai *Jitter* bisa menyebabkan *packet loss* dan mengurangi *quality of service* untuk protokol yang sensitif terhadap waktu seperti *VoIP*.

Pada percobaan ketiga, di detik ke-11 terlihat ada peningkatan *Jitter* yang tinggi karena ada *Link 1* terdeteksi mati pada detik ke-5. Namun, itu tidak lama karena *Switch* secara cepat beradaptasi pada situasi. Pada detik ke-16 dimana ada *event Link 1* terdeteksi kembali aktif, tidak nampak perubahan *Jitter* yang spesifik karena *Switch* secara otomatis beradaptasi.

Dari Gambar 8 terlihat bahwa perbedaan *Bandwidth* pada percobaan pertama dan kedua tidak memiliki perbedaan yang signifikan. Akan tetapi pada percobaan ketiga terlihat ada gejala perubahan *Bandwidth* hingga 400%. Karena pada detik ke-5, *Link 1* mati yang menyebabkan tidak ada paket UDP yang diterima oleh *Server* sehingga *Server* mendeteksi 0 *Bandwidth*. Pada detik ke-10, *Switch* mulai beradaptasi dengan melakukan *learning* yang menyebabkan *flooding*. Setelah proses *learning* selesai pada detik ke-13, grafik *Bandwidth* kembali normal. Pada detik ke-16 dimana *Link 1* terdeteksi nyala, secara otomatis *Switch* melakukan *learning* kembali lalu *flooding* yang menyebabkan *Bandwidth* yang diterima oleh *Server* meningkat hingga 400%. Setelah *flooding* selesai yang menandakan *learning* selesai *Bandwidth* kembali seperti normal layaknya percobaan pertama dan kedua.



Gambar 8. Grafik Perbandingan *Bandwidth*

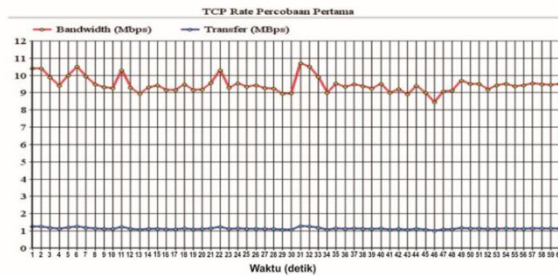
B. Pengujian Terhadap Paket Transport Control Protocol

Proses pengujian terhadap paket *TCP* akan dilakukan dengan *tool Iperf* dari *Host* ke *Server*. *Host* akan mengirim paket *TCP* selama 60 detik melalui *port* tujuan 7777 dengan *Bandwidth* 10Mbps. Informasi tentang percobaan *TCP* dipaparkan pada Tabel 3 di bawah ini

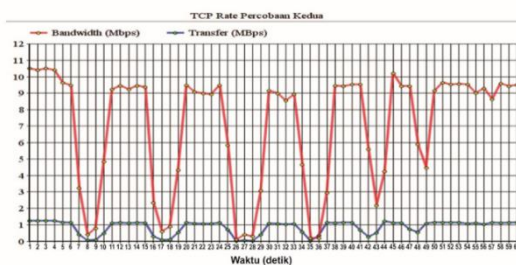
Tabel 3. Tabel Percobaan *TCP*

No.	Percobaan
1	Metode <i>Load Sharing</i> dengan <i>Weighted Round Robin</i> sesuai dengan algoritma yang penulis buat.
2	Sama seperti yang pertama namun ditambah apabila terjadi ada <i>Link</i> yang mati pada detik ke 6, 15, dan 24. <i>Link</i> aktif pada detik ke 33, 41, dan 47.

Hasil dari pengujian tersebut ditampilkan dengan grafik pada gambar 9, 10, dan 11.



Gambar 9. Grafik TCP Rate Percobaan Pertama

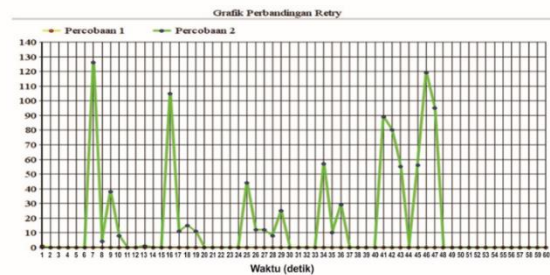


Gambar 10. Grafik TCP Rate Percobaan kedua

Dari Gambar 9 terlihat Bandwidth pengiriman TCP dari Host ke Server secara normal menggunakan Load Sharing dengan metode Weighted Round Robin. Gambar 10 ini merupakan grafik dari percobaan kedua.

Dari Gambar 10 terlihat bahwa ada perubahan yang signifikan yang disebabkan event-event pada Link yang terhubung ke Server. Pada detik ke-6, Switch mendeteksi Link pertama mati yang menyebabkan ada penurunan Bandwidth. Switch selesai learning untuk event pertama pada detik ke-10 yang menyebabkan Bandwidth kembali seperti semula. Pada detik ke-15, Switch mendeteksi Link kedua mati, dan detik ke-24, Switch mendeteksi Link ketiga mati sehingga terjadi penurunan Bandwidth yang signifikan seperti event pertama. Terutama ketika Link yang tersisa tinggal satu, terlihat Bandwidth pada posisi terendah. Event keempat dimulai pada detik ke-33, dimana Link satu terdeteksi aktif yang menyebabkan Bandwidth juga pada posisi terendah. Selanjutnya di detik ke-41, Link ke-2 aktif, dan di detik ke-47 Link ketiga kembali aktif. Jadi dari grafik ini terlihat bagaimana Switch secara otomatis beradaptasi apabila ada kondisi dimana ada Link yang aktif atau mati secara tiba-tiba.

Dari Gambar 11 terlihat bahwa pada saat percobaan pertama hampir tidak ada *retry* sama sekali. *Retry* adalah proses Server meminta kembali mengirimkan paket TCP dari Host. Pada detik ke-6 adalah posisi dimana *retry* tertinggi karena Link pertama adalah Link yang memiliki bobot tertinggi pada penjadwalan *Weighted Round Robin*. Posisi tertinggi kedua pada detik ke-15 dimana Link kedua yang memiliki bobot



tertinggi kedua juga mati. Perubahan yang curam terjadi setiap ada *event* baik ada Link yang hidup atau pun mati.

## V. KESIMPULAN

Penelitian ini menjelaskan hasil pengujian dari algoritma yang dibuat oleh penulis untuk paket data yang menggunakan protokol UDP. Penulis menggunakan *controller Ryu* dalam mengimplementasikan aplikasi dan emulator *Mininet* untuk mensimulasikan topologi. Hasil dari simulasi menunjukkan bahwa algoritma yang penulis buat mampu menurunkan *jiter* hingga 50% dan paket received out of order berhasil diturunkan hingga 0. Sedangkan waktu yang dibutuhkan untuk kembali *online* ketika ada link yang mati adalah 5 detik untuk UDP dan 3 detik untuk TCP. Pada saat pengujian TCP juga terdeteksi 1 *Retry* dan proses learning untuk fault tolerance dapat dilakukan kurang dari 5 detik.

Dengan Software Defined Networking kita dapat membuat mekanisme baru dari proses yang dilakukan oleh perangkat-perangkat jaringan komputer, dimana hal ini tidak dapat dilakukan pada perangkat-perangkat jaringan komputer konvensional.

## DAFTAR PUSTAKA

- [1] T.D. Nadeau dan K. Gray, SDN : *Software Defined Networks*, 2013.
- [2] US: *Open Networking Foundation, Software-defined Networking: The New Norm for Networks*. ONF White Paper. Palo Alto, 2013.
- [3] H. Long, Y. Shen, M. Guo, dan F. Tang, "LABERIO: Dynamic Load Balanced Routing in OpenFlow-enabled Networks". *IEEE 27<sup>th</sup> International Conference on Advanced Information Networking and Applications*, 2013.
- [4] R. Khondoker, A. Zaalouk, R. Marx, dan K. Bayarou, *Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers*, 2014.
- [5] S. Azodomolky, *Software Defined Networking with OpenFlow*.
- [6] US: *Open Networking Foundation*. 2012. *Openflow Switch Specification*, 2013.
- [7] Sdx Central, *What is Ryu Controller?* <https://www.sdxcentral.com/sdn/definitions/sdn-controllers/open-source-sdn-controllers/what-is-ryu-controller>. Diakses pada tanggal 25 Agustus 2016, 2015.
- [8] Ryu SDN Framework Community, *RYU OpenFlow Controller*. <https://osrg.github.io/ryu/index.html>. Diakses pada tanggal 26 Agustus 2016, 2014.
- [9] Mininet Team, *Mininet Overview*. <http://mininet.org/overview>. Diakses pada tanggal 26 Agustus 2016, 2016.

- [10] Bourke, Tony, *Server Load Balancing*. Sebastopol: O'Reilly & Associates, Inc, 2001.
- [11] The Linux Foundation, *Bonding*. <https://wiki.linuxfoundation.org/networking/bonding>. Diakses pada tanggal 20 Desember 2016, 2015.
- [12] Cisco, *Link Aggregation Control Protocol (LACP) (802.3ad) for Gigabit Interfaces*. [http://www.cisco.com/c/en/us/td/docs/ios/12\\_2sb/feature/guide/gigeth.html](http://www.cisco.com/c/en/us/td/docs/ios/12_2sb/feature/guide/gigeth.html). Diakses pada tanggal 25 Desember 2016, 2015.
- [13] Wensong, *Job Scheduling Algorithms in Linux Virtual Server*. <http://www.linuxvirtualserver.org/docs/scheduling.html>. Diakses pada tanggal 19 Desember 2016, 1998.
- [14] M. Charles Kozierok, *Transport Layer (Layer 4)*. [http://www.tcpipguide.com/free/t\\_TransportLayerLayer4.htm](http://www.tcpipguide.com/free/t_TransportLayerLayer4.htm). Diakses pada tanggal 26 Desember 2016, 2005.
- [15] Ryu Project Team, *Ryu SDN Framework Using OpenFlow 1.3*, 2016.
- [16] SDN Central, *Kanazawa University Hospital Reaping the Benefits of a Successful Production SDN Deployment*, 2013.



# PEDOMAN PENULISAN JURNAL ULTIMATICS, ULTIMA INFOSYS, DAN ULTIMA COMPUTING

## 1. Kriteria Naskah

- Naskah belum pernah dipublikasikan atau tidak dalam proses penyuntingan di jurnal berkala lainnya.
- Naskah yang dikirimkan dapat berupa naskah hasil penelitian atau konseptual.

## 2. Pengetikan Naskah

- Naskah diketik dengan jarak spasi antar baris 1 pada halaman ukuran A4 (21 cm x 29,7 cm), margin kiri-atas 3 cm dan kanan-bawah 2 cm, dengan jenis tulisan Times New Roman.
- Naskah dapat ditulis dalam Bahasa Indonesia atau Bahasa Inggris.
- Jumlah halaman untuk tiap naskah dibatasi dengan jumlah minimal 4 halaman dan maksimal 8 halaman.

## 3. Format Naskah

- Komposisi naskah terdiri dari Judul, Abstrak, Kata Kunci, Pendahuluan, Metode, Hasil Penelitian dan Pembahasan, Simpulan, Lampiran, Ucapan Terima Kasih, dan Daftar Pustaka.
- Judul memiliki jumlah kata maksimal 15 kata dalam Bahasa Indonesia atau maksimal 12 kata dalam Bahasa Inggris (termasuk subjudul bila ada).
- Abstrak ditulis dengan Bahasa Inggris paling banyak 200 kata, meskipun bahasa yang digunakan dalam penyusunan naskah adalah Bahasa Indonesia. Isi abstrak sebaiknya mengandung argumentasi logis, pendekatan pemecahan masalah, hasil yang dicapai, dan simpulan singkat.
- Kata Kunci ditulis dengan Bahasa Inggris dalam satu baris, dengan jumlah kata antara 4 sampai 6 kata.
- Pendahuluan berisi latar belakang dan tujuan penelitian.
- Metode dapat diuraikan secara terperinci dan dibedakan menjadi beberapa bab maupun subbab yang terpisah.
- Hasil dan Pembahasan disajikan secara sistematis sesuai dengan tujuan penelitian.
- Simpulan menyajikan intisari hasil penelitian yang telah dilaksanakan. Saran pengembangan untuk penelitian selanjutnya juga dapat diberikan di sini.

- Lampiran dan Ucapan Terima Kasih dapat dijabarkan setelah Simpulan secara singkat dan jelas.
- Daftar Pustaka yang dirujuk dalam naskah harus dituliskan di bagian ini secara kronologis berdasarkan urutan kemunculannya. Cara penulisannya mengikuti cara penulisan jurnal dan transaction IEEE.
- Template naskah telah disediakan dan dapat diminta dengan menghubungi surel redaksi.

## 4. Penulisan Daftar Pustaka

- Artikel Ilmiah:  
N. Penulis, "Judul artikel ilmiah," *Singkatan Nama Jurnal*, vol. x, no. x, hal. xxx-xxx, Sept. 2013.
- Buku  
N. Penulis, "Judul bab di dalam buku," di dalam *Judul dari Buku*, edisi x. Kota atau Negara Penerbit: Singkatan Nama Penerbit, tahun, bab x, subbab x, hal. xxx-xxx.
- Laporan  
N. Penulis, "Judul laporan," *Singkatan Nama Perusahaan, Kota Perusahaan, Singkatan Nama Negara, Laporan xxx*, tahun.
- Buku Manual/ *handbook*  
*Nama dari Buku Manual*, edisi x, *Singkatan Nama Perusahaan, Kota Perusahaan, Singkatan Nama Negara, tahun*, hal. xxx-xxx.
- Prosiding  
N. Penulis, "Judul artikel," di dalam *Nama Konferensi Ilmiah*, Kota Konferensi, *Singkatan Nama Negara (jika ada), tahun*, hal. xxx-xxx.
- Artikel yang Disajikan dalam Konferensi  
N. Penulis, "Judul artikel," disajikan di *Nama Konferensi, Kota Konferensi, Singkatan Nama Negara, tahun*.
- Paten  
N. Penulis, "Judul paten," HKI xxxxxx, 01 Januari 2014.
- Tesis dan Disertasi  
N. Penulis, "Judul tesis," M.Sc. thesis, *Singkatan Departemen, Singkatan*

Universitas, Kota Universitas, Singkatan Nama Negara, tahun.

N. Penulis, "Judul disertasi," Ph.D. dissertation, Singkatan Departemen, Singkatan Universitas, Kota Universitas, Singkatan Nama Negara, tahun.

- Belum Terbit  
N. Penulis, "Judul artikel," belum terbit.

N. Penulis, "Judul artikel," Singkatan Nama Jurnal, proses cetak.

- Sumber online  
N. Penulis. (tahun, bulan tanggal). Judul (edisi) [Media perantara]. Alamat situs: [http://www.\(URL\)](http://www.(URL))

N. Penulis. (tahun, bulan). Judul. Jurnal [Media perantara]. *volume(issue)*, halaman jika ada. Alamat situs: [http://www.\(URL\)](http://www.(URL))

Catatan: media perantara dapat berupa media online, CD-ROM, USB, dan sebagainya.

#### 5. Pengiriman Naskah Awal

- Para penulis dapat mengirimkan naskah hasil penelitiannya dalam bentuk .doc atau .pdf melalui surel ke [umnjurnal@gmail.com](mailto:umnjurnal@gmail.com) dengan subjek sesuai Jurnal yang dipilih.
- Seluruh isi naskah yang dikirimkan harus memenuhi syarat dan ketentuan yang ditentukan.
- Kami akan menjaga segala kerahasiaan dan Hak Cipta karya Anda.
- Sertakan biodata penulis pertama yang lengkap, meliputi nama, alamat kantor, alamat penulis, telpon kantor/ rumah dan hp, serta No NPWP (bagi yang memiliki NPWP).

#### 6. Penilaian Naskah

- Seluruh naskah yang diterima akan melalui serangkaian tahap penilaian yang melibatkan mitra bestari.
- Setiap naskah akan direview oleh minimal 2 orang mitra bestari.
- Rekomendasi dari mitra bestari yang akan menentukan apakah sebuah naskah diterima, diterima dengan revisi minor, diterima dengan revisi major, atau ditolak.

#### 7. Pengiriman Naskah Final

- Naskah yang diterima untuk diterbitkan akan diinformasikan melalui surel redaksi.
- Penulis berkewajiban memperbaiki setiap kesalahan yang ditemukan sesuai saran dari mitra bestari.
- Naskah final yang telah direvisi dapat dikirimkan kembali ke surel redaksi beserta hasil scan Copyright Transfer Form yang telah ditandatangani.

#### 8. Copyright dan Honorarium

- Penulis yang naskahnya dimuat harus membaca dan menyetujui isi Copyright Transfer Form kepada redaksi.
- Copyright Transfer Form harus ditandatangani oleh penulis pertama naskah.
- Naskah yang dimuat akan mendapatkan honorarium sebesar Rp 1.000.000,- per naskah, setelah dipotong pajak 2.5% (bila penulis pertama yang memiliki NPWP) dan 3% (tanpa NPWP).
- Honorarium akan ditransfer ke rekening penulis pertama (tidak dapat diwakilkan) paling lambat 2 minggu setelah jurnal naik cetak dan siap didistribusikan.
- Penulis yang naskahnya dimuat akan mendapatkan copy jurnal sebanyak 2 eksemplar.

#### 9. Biaya Tambahan

- Permintaan tambahan copy jurnal harus dibeli seharga Rp 50.000,- per copy.
- Permintaan penambahan jumlah halaman dalam naskah (maksimal 8 halaman) akan dikenai biaya sebesar Rp 25.000,- per halaman.

#### 10. Alamat Redaksi

d.a. Koordinator Riset  
Fakultas Teknologi Informasi dan Komunikasi  
Universitas Multimedia Nusantara  
Gedung Rektorat Lt.6  
Scientia Garden, Jl. Boulevard Gading Serpong,  
Tangerang, Banten -15333  
Surel: [umnjurnal@gmail.com](mailto:umnjurnal@gmail.com)

# Judul Paper

Sub Judul (jika diperlukan)

Nama Penulis A<sup>1</sup>, Nama Penulis B<sup>2</sup>, Nama Penulis C<sup>2</sup>

<sup>1</sup> Baris pertama (dari afiliasi): nama departemen organisasi, nama organisasi, kota, negara  
Baris kedua: alamat surel jika diinginkan

<sup>2</sup> Baris pertama (dari afiliasi): nama departemen organisasi, nama organisasi, kota, negara  
Baris kedua: alamat surel jika diinginkan

Diterima dd mmmmm yyyy

Disetujui dd mmmmm yyyy

**Abstract**—This electronic document is a “live” template which you can use on preparing your IJNMT paper. Use this document as a template if you are using Microsoft Word 2007 or later. Otherwise, use this document as an instruction set. Do not use symbol, special characters, or Math in Paper Title and Abstract. Do not cite references in the abstract.

**Index Terms**—enter key words or phrases in alphabetical order, separated by commas

## I. PENDAHULUAN

Dokumen ini, dimodifikasi dalam MS Word 2007 dan disimpan sebagai dokumen Word 97-2003, memberikan panduan yang diperlukan oleh penulis untuk mempersiapkan dokumen elektroniknya. Margin, lebar kolom, jarak antar baris, dan jenis-jenis format lainnya telah disisipkan di sini. Penulis berkewajiban untuk memastikan dokumen yang dipersiapkannya telah memenuhi format yang disediakan.

Isi Pendahuluan mengandung latar belakang, tujuan, identifikasi masalah dan metode penelitian yang dipaparkan secara tersirat (implisit). Kecuali bab Pendahuluan dan Simpulan, penulisan judul bab sebaiknya eksplisit sesuai dengan isi yang dijelaskan, tidak harus implisit dinyatakan sebagai Dasar Teori, Perancangan, dan sebagainya.

## II. PENGGUNAAN YANG TEPAT

### A. Memilih Template

Pertama, pastikan Anda memiliki *template* yang tepat untuk artikel Anda. *Template* ini ditujukan untuk Jurnal ULTIMATICS, ULTIMA InfoSys, dan ULTIMA Computing. *Template* ini menggunakan ukuran kertas A4.

### B. Mempertahankan Keutuhan Format

*Template* ini digunakan untuk mem-format artikel dan *style* isi artikel Anda. Seluruh margin, lebar kolom, jarak antar baris, dan jenis tulisan telah diberikan, jangan diubah.

## III. PERSIAPKAN ARTIKEL ANDA

Sebelum Anda mulai mem-format artikel Anda, tulislah terlebih dahulu artikel Anda dan simpan sebagai *text file* lainnya. Setelah selesai baru lakukan pencocokkan *style* dokumen. Jangan tambahkan nomor halaman di bagian manapun dari dokumen ini. Perhatikan pula beberapa hal berikut saat melakukan pengecekan tulisan.

### A. Singkatan

Definisikan singkatan pada saat pertama kali digunakan di dalam isi tulisan, walaupun singkatan tersebut telah didefinisikan di dalam abstrak. Singkatan seperti IEEE, SI, MKS, CGS, sc, dc, dan rms tidak harus didefinisikan. Singkatan yang menggunakan tanda titik tidak boleh diberi spasi, seperti “C.N.R.S.”, bukan “C. N. R. S.” Jangan gunakan singkatan di dalam Judul Artikel atau Judul Bab, kecuali tidak dapat dihindari.

### B. Unit

- Gunakan baik SI (MKS) atau CGS sebagai unit primer.
- Jangan menggabungkan kepanjangan dan singkatan dari unit, yang tepat seperti “Wb/m<sup>2</sup>” atau “webers per meter persegi,” bukan “webers/m<sup>2</sup>.”
- Gunakan angka nol di depan suatu bilangan desimal, seperti “0,25” bukan “.25.”

### C. Persamaan

Format persamaan merupakan suatu pengecualian di dalam spesifikasi *template* ini. Anda harus menentukan apakah akan menggunakan jenis tulisan Times New Roman atau Symbol (jangan jenis tulisan yang lain). Bila Anda membuat beberapa persamaan berbeda, akan lebih baik bila Anda mempersiapkan persamaan tersebut sebagai gambar dan menyisipkannya ke dalam artikel Anda setelah diberi *style*.



Beri penomoran untuk persamaan Anda secara berurutan. Nomor persamaan berada dalam tanda kurung seperti (1), dan diletakkan pada bagian kanan dengan menggunakan suatu *right tab stop*.

$$\int_0^{r_2} F(r, \phi) dr d\phi = [\sigma r_2 / (2\mu_0)] \quad (1)$$

Perhatikan bahwa persamaan di atas diposisikan di bagian tengah dengan menggunakan suatu *center tab stop*. Pastikan bahwa simbol-simbol yang digunakan dalam persamaan Anda didefinisikan sebelum atau sesudah persamaan. Gunakan "(1)," bukan "Persamaan (1)," kecuali pada awal sebuah kalimat, seperti "Persamaan (1) merupakan ...."

#### D. Beberapa Kesalahan Umum

- Perhatikan tata cara penulisan Bahasa Indonesia yang benar, perhatikan penggunaan kata depan dan kata sambung yang tepat, seperti "di depan" dan "disampaikan".
- Kata-kata asing yang belum diserap ke dalam Bahasa Indonesia dapat dicetak miring, atau diberi garis bawah, atau dicetak tebal (pilih salah satu), seperti "*italic*", "underlined", "**bold**".
- Prefiks seperti "non", "sub", "micro", "multi", dan "ultra" bukan kata yang berdiri sendiri, oleh karenanya harus digabung dengan kata yang mengikutinya, biasanya tanpa tanda hubung, seperti "subsistem".

#### IV. MENGGUNAKAN TEMPLATE

Setelah naskah artikel Anda selesai di-*edit*, artikel Anda dapat dipersiapkan untuk *template*. Gandakan template ini dengan menggunakan perintah Save As dan simpan dengan penamaan berikut:

- ULTIMATICS\_namaPenulis1\_judulArtikel.
- ULTIMAInfoSys\_namaPenulis1\_judulArtikel.
- ULTIMAComputing\_namaPenulis1\_judulArtikel.

Selanjutnya Anda dapat meng-*import* artikel Anda dan mempersiapkannya sesuai *template* yang diberikan. Perhatikan beberapa hal berikut pada saat melakukan pengecekan.

##### A. Penulis dan Afiliasi

*Template* ini didesain untuk tiga penulis dengan dua afiliasi yang berbeda. Penamaan afiliasi yang sama tidak perlu berulang, cukup afiliasi yang berbeda yang ditambahkan. Berikan alamat surel resmi afiliasi atau penulis jika diinginkan.

##### B. Penamaan Judul Bab dan Subbab

Bab merupakan suatu perangkat organisatorial yang memandu pembaca untuk membaca isi artikel

Anda. Terdapat dua jenis bab: bab utama (bab) dan subbab.

Bab utama mengidentifikasi komponen-komponen yang berbeda dalam artikel Anda dan tidak memiliki hubungan isi yang erat satu sama lainnya. Sebagai contoh PENDAHULUAN, DAFTAR PUSTAKA, dan UCAPAN TERIMA KASIH. Penulisan judul bab utama menggunakan huruf kapital dan penomoran angka Romawi.

Subbab merupakan isi yang dijabarkan lebih terstruktur dan memiliki relasi yang kuat. Penamaan subbab ditulis dengan menggunakan cara penulisan judul kalimat utama (*Capitalize Each Word*) dan penomorannya menggunakan huruf alfabet kapital secara berurutan. Untuk subsubbab, penamaan dan penomoran mengikuti cara penamaan dan penomoran subbab diikuti angka Arab, seperti "A.1 Penulis", "A.1.1 Afiliasi Penulis".

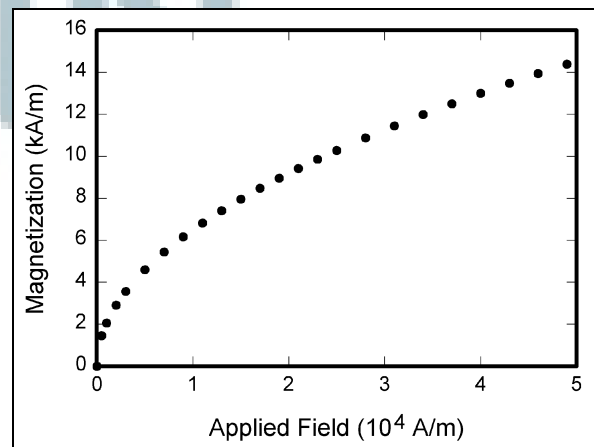
##### C. Gambar dan Tabel

Letakkan gambar dan tabel di atas atau di bawah kolom. Hindari posisi di tengah kolom. Gambar dan tabel yang besar dapat mengambil area dua kolom menjadi satu kolom. Judul gambar harus diletakkan di bawah gambar, sedangkan judul tabel harus diletakkan di atas tabel. Masukkan gambar dan tabel setelah mereka dirujuk di dalam isi artikel.

Tabel 1. Contoh tabel

Table Head	Table Column Head		
	Table column subhead	Subhead	Subhead
copy	More table copy		

Penamaan judul gambar dan tabel menggunakan cara penulisan kalimat biasa (*Sentence case*). Berikan jarak baris sebelum dan sesudah gambar atau tabel dengan kalimat penyertanya.



Gambar 1. Contoh gambar

## V. SIMPULAN

Bagian simpulan bukan merupakan keharusan. Meskipun suatu simpulan dapat memberikan gambaran mengenai intisari artikel Anda, jangan menduplikasi abstrak sebagai simpulan Anda. Sebuah simpulan dapat menekankan pada pentingnya penelitian yang Anda lakukan atau saran pengembangan penelitian selanjutnya yang dapat dikerjakan.

## LAMPIRAN

Jika diperlukan, Anda dapat menyisipkan lampiran-lampiran yang digunakan dalam artikel Anda sebelum UCAPAN TERIMA KASIH.

## UCAPAN TERIMA KASIH

Di bagian ini Anda dapat memberikan pernyataan atau ungkapan terima kasih pada pihak-pihak yang telah membantu Anda dalam pelaksanaan penelitian yang Anda lakukan.

## DAFTAR PUSTAKA

Untuk penamaan daftar pustaka, gunakan tanda kurung siku, seperti [1], secara berurutan dari awal rujukan dilakukan. Untuk merujuknya dalam kalimat, cukup gunakan [2], bukan “Rujukan [3]”, kecuali di awal sebuah kalimat, seperti “Rujukan [3] menggambarkan ....”

Penomoran catatan kaki dilakukan secara terpisah dengan *superscripts*. Letakkan catatan kaki tersebut di

bawah kolom dimana catatan kaki tersebut dirujuk. Jangan letakkan catatan kaki di dalam daftar pustaka.

Kecuali terdapat enam atau lebih penulis, jabarkan nama penulis tersebut satu-satu, jangan gunakan “dkk”. Artikel yang belum diterbitkan, meskipun sudah dikirim untuk diterbitkan, harus ditulis “belum terbit” [4]. Artikel yang sudah dikonfirmasi untuk diterbitkan, namun belum terbit, harus ditulis “proses cetak” [5]. Gunakan cara penulisan kalimat (*Sentence case*) untuk penulisan judul artikel.

Untuk artikel yang diterbitkan dalam jurnal terjemahan, tuliskan terlebih dahulu rujukan hasil terjemahannya, diikuti dengan jurnal aslinya [6].

- [1] G. Eason, B. Noble, dan I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” *Phil. Trans. Roy. Soc. London*, vol. A247, hal. 529-551, April 1955.
- [2] J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, hal.68-73.
- [3] I.S. Jacobs dan C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in *Magnetism*, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, hal. 271-350.
- [4] K. Elissa, “Title of paper if known,” belum terbit.
- [5] R. Nicole, “Title of paper with only first word capitalized,” *J. Name Stand. Abbrev.*, proses cetak.
- [6] Y. Yorozu, M. Hirano, K. Oka, dan Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” *IEEE Transl. J. Magn. Japan*, vol. 2, hal. 740-741, Agustus 1987 [Digests 9th Annual Conf. Magnetism Japan, hal. 301, 1982].
- [7] M. Young, *The Technical Writer’s Handbook*. Mill Valley, CA: University Science, 1989.

UMN

ISSN 2355-3286



9 772355 328009



Universitas Multimedia Nusantara  
Scientia Garden Jl. Boulevard Gading Serpong, Tangerang  
Telp. (021) 5422 0808 | Fax. (021) 5422 0800