Edu Komputika 7 (1) (2020)

Edu Komputika Journal

http://journal.unnes.ac.id/sju/index.php/edukom

Analisis Validasi *Image PNG File Upload* menggunakan Metadata pada Aplikasi Berbasis *Web*

Fahmi Anwar[1)]✉, Abdul Fadlil[2)], dan Imam Riadi[3)]

[1]Program Studi Teknik Informatika, Universitas Ahmad Dahlan, Yogyakarta

[2]Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Yogyakarta

[3]Program Studi Sistem Informasi, Universitas Ahmad Dahlan, Yogyakarta

Info Artikel

Sejarah Artikel:
Diterima: Mei 2020
Direvisi: Juni 2020
Disetujui: Juni 2020

Keywords:
Keamanan, Metadata, PNG, Unggah, Validasi

Abstrak

Penggunaan *website* berkembang pesat dengan banyaknya penggunaan berbagai aplikasi atau layanan yang terhubung ke Internet. Faktor keamanan sangatlah penting dikarenakan pada zaman yang semakin berkembang banyak sistem yang berupa digital. Ancaman serangan terhadap berbagai jenis sistem yang berbentuk digital atau server maka diperlukan sebuah penanganan terhadap ancaman atau serangan pada *server* dengan celah fitur unggahan. File Upload pada website yang diproses oleh sistem melakukan validasi dengan menyaring jenis berkas objek digital di *server side (backend)* atau dalam halaman *website* pada *web browser* dalam bentuk HTML atau Javascript (*frontend*). Biasanya teknik penyaringan hanya melihat File Extention atau Magic Number dari sebuah berkas yang diunggah akan tetapi penyaringan tersebut bisa dimanipulasi, masalah tersebut dapat diatasi dengan menambahkan teknik penyaringan untuk mengecek validasi dari sebuah berkas dengan Metadata. Penelitian ini dikembangkan dengan menggunakan metode Extreme Programming (XP) karena lebih fleksibel dalam penggunaan secara eksperimental dengan tim dalam skala kecil hingga medium juga sesuai jika dihadapkan dengan kebutuhan atau requirement yang terjadi perubahan–perubahan dari kebutuhan-kebutuhan yang dinamis atau sangat cepat maupun yang tidak jelas sejak awal.. Pengujian keamanan dilakukan untuk mengetahui perbedaan antara sebelum dan sesudah kondisi diterapkan dengan skenario uji yang telah ditentukan. Berdasarkan hasil pengujian yang dilakukan dengan dari 8 skenario yang dipersiapkan menghasilkan 100% sesuai dengan harapan atau penggunaan teknik penyaringan Metadata yang berisi dimensi dan nilai RGBA lebih efektif dalam menyaring berkas.

Abstract

The use of websites is growing rapidly with the use of various applications or services that are connected to the Internet. The safety factor is very important because in an increasingly modern era many systems are digital. The threat of attacks on various types of digital or server systems requires a handling of threats or attacks on the server with upload feature loopholes. File uploads on websites that are processed by the system validate by filtering file types of digital objects on the server side (backend) or on web pages in web browsers in the form of HTML or Javascript (frontend). Usually filtering techniques only see the File Extention or Magic Number of an uploaded file but the filtering can be manipulated, the problem can be overcome by adding filtering techniques to check the validation of a file with Metadata. This research was developed using the Extreme Programming (XP) method because it is more flexible in experimental use with teams from small to medium scale is also appropriate when faced with the needs or requirements that occur changes from the needs of dynamic or very fast or not clear from the start. Safety testing is carried out to determine the difference between before and after the conditions are applied with the specified test scenarios. Based on the results of tests conducted with 8 scenarios prepared to produce 100% in accordance with expectations or the use of metadata filtering techniques that contain dimensions and values of RGBA are more effective in filtering files.

✉ Alamat korespondensi:
Kampus 3 UAD, Jl Prof. Dr. Soepomo S.H., Yogyakarta 55164
E-mail: fahwar95@gmail.com

ISSN 2252-6811
E-ISSN 2599-297X

PENDAHULUAN

Teknologi sangat memberikan manfaat dan membantu kegiatan manusia sehari-hari misalnya pemanfaatan teknologi internet seperti website banyak diimplementasikan diberbagai sistem (Umar et al., 2019). *Website* merupakan layanan pada Internet yang dapat diakses oleh berbagai orang di dunia, layanan website memiliki berbagai fitur salahsatunya fitur unggahan. Fitur unggah berkas adalah teknik yang biasanya dibutuhkan secara fungsional pada aplikasi untuk para pengguna (Chen et al., 2015) yang dapat digunakan untuk dokumen, gambar, unggah data dan penyimpanan oleh klien (Umar et al., 2019). Namun tanpa metode keamanan dan penyaringan yang tepat, pemilihan berkas dan proses validasi selama mengungah dapat memberikan resiko keamanan yang signifikan misalnya teknik keamanan aplikasi *website* (Li & Xue, 2011) Kolaborasi antara *server* dan klien untuk meningkatkan keamanan dengan memberikan mekanisme untuk mencapai keamanan *end-to-end* seperti pada pemanfaatan teknologi *Web Push Notification* yang megirim pesan ke *web browser* dari *server* (Rahmatulloh et al., 2019).

Beberapa kerentanan terkait dengan lapisan pada aplikasi berbasis *web* (Sajjad et al., 2015). Penelitian ini juga memberikan ulasan tentang teknik, tahapan, pendekatan dan alat untuk mendeteksi kerentanan. Pentingnya *web server* bersama dengan ancaman yang ditimbulkan oleh peretas (Almi, 2014). Penelitian ini mengambil masalah berdasarkan OWASP (OWASP, 2013).

Beberapa teknik unggahan *file* dapat dieksploitasi seperti tidak ada validasi yang dilakukan pada klien atau *server*, validasi yang diterapkan di sisi klien dapat dilewati menggunakan opsi pengembang, tidak ada validasi yang dilakukan untuk memeriksa konten *file* yang diunggah oleh pengguna akhir, tidak ada validasi yang dilakukan untuk memeriksa ukuran *file* yang diunggah oleh pengguna akhir, ketika validasi didasarkan hanya pada tipe konten, serangan dapat dilakukan dengan memanipulasi tipe konten *file* yang menentukan sifat data, diizinkan menggunakan lebih dari satu jenis ekstensi *file* dan beberapa kondisi dapat menggunakan ekstensi *file* terlarang bersama dengan ekstensi *file* yang tidak diizinkan oleh aplikasi (Pooj & Patil, 2016). Para penyerang dapat melakukan *XSS* yang disimpan menggunakan fitur unggah *file* seperti berkas gambar.(W et al., 2018).

Gambar merupakan susunan angka yang bernilai intensitas cahaya juga memiliki deskripsi numerik mengambil bentuk bernama piksel (Poornima & Iswarya, 2013). *GD Graphic Library* adalah pustaka kode *opensource* untuk pembuatan gambar secara dinamis oleh pemrograman. *GD* ditulis dalam bahasa *C* yang tersedia untuk bahasa *Perl, PHP*, dan bahasa lainnya. *GD* menciptakan gambar *PNG, JPEG, GIF, WebP, XPM, BMP* dan format lainnya. *GD* umumnya digunakan untuk menghasilkan grafik, gambar, *thumbnail*, dan sebagian besar hal lainnya dengan cepat meskipun tidak terbatas untuk digunakan hanya pada *website*, aplikasi *GD* yang paling umum melibatkan pengembangan pada *website*. *Library* ini awalnya dikembangkan oleh Thomas Boutell dan sekarang dikelola oleh Pierre Joye di bawah naungan PHP.net. (Boutell & Joye, n.d.)

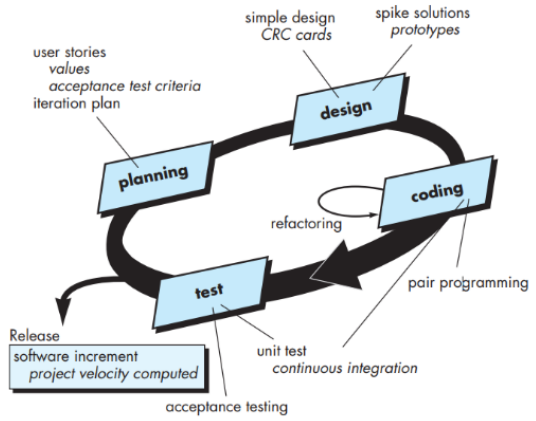
Metadata merupakan informasi yang berisi pada objek digital seperti pembuat, tanggal dan waktu pembuatan dan yang lainnya. Metadata merupakan informasi yang dapat digunakan untuk mengidentifikasi sebuah objek digital. (Sari et al., 2016). Gambar juga dapat dideteksi keaslian gambar digital dengan memverifikasi menggunakan alat pendeteksi. (Riadi et al., 2017)

Penelitian (Sari et al., 2016) menjelaskan teknik forensik yang dapat mendeteksi berkas gambar yang direkayasa menggunakan *Error Level Analysis (ELA)* dan teknik *ELA (Error Level Analysis)* pada Forensicallybeta dapat digunakan untuk mendeteksi keaslian suatu citra (Sulistyo et al., 2018), citra juga digunakan untuk mendeteksi tekstur dengan memanfaatkan *Gray Level Coocurrence Matrix (GLCM)* dengan klasifikasi jarak menggunakan *Euclidean* (Saifudin & Fadlil, 2015).

Penelitian terdahulu yang meneliti tentang teknik validasi dengan memanfaatkan Metadata dan nilai *RGB* (*Red, Green, Blue*) pada *Image JPEG File* menggunakan *Image Processing* (Anwar et al., 2019). Pada penelitian ini teknik penyaringan pada validasi Image PNG dengan memanfaatkan *metadata* dan nilai *RGBA* (*Red, Green, Blue, Alpha*) pada fitur *File Upload* pada *website*.

METODE PENELITIAN

Extreme Programming (XP) merupakan metode yang digunakan dalam membangun aplikasi. *XP* biasanya menggunakan pendekatan pemrograman berorientasi objek *(object oriented programming)* dan sasaran dari metode ini merupakan tim dalam skala kecil hingga *medium* juga sesuai jika dihadapkan dengan kebutuhan atau *requirement* yang terjadi perubahan–perubahan dari kebutuhan-kebutuhan yang dinamis atau sangat cepat maupun yang tidak jelas sejak awal. (Pressman & Maxim, 2014)



Gambar 1. Alur Metode *Extreme Programming* (*XP*)

Gambar 1 merupakan tahapan alur pembangunan aplikasi *website* menggunakan metode *Extreme Programming* terdiri dari Perencanaan (*Planning*) yang berisi kumpulan kebutuhan aktifitas dan berbagai proses bisnis yang akan dibuat, Perancangan (*Design*) yang berisi kumpulan dari *Unified Modelling Language (UML)* yang diperlukan, Pengkodean (*Coding*) yang berisi kumpulan kode yang dibuat menggunakan PHP, Pengujian (*Testing*) yang berisi pengujian menggunakan *Black Box Testing* dan Peningkatan Perangkat Lunak (*Software Increment*) yang berisi informasi yang diperlukan untuk pengembangan berikutnya.

HASIL DAN PEMBAHASAN

Penelitian ini menggunakan Extreme Programming sebagai metode pengembangan implementasi teknik penyaringan untuk validasi *Image PNG File Upload* menggunakan Metadata memanfaatkan *GD Graphic Library* dengan beberapa tahapan sebagai berikut :

A. Perencanaan *(Planning)*

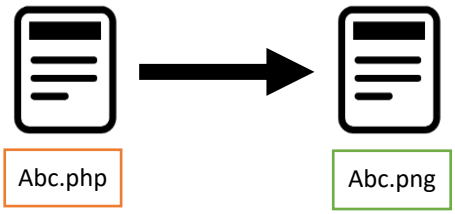
Tahap Perencanaan berisi tahapan yang dilakukan menggunakan sampel yang dibuat seperti yang tersaji pada Tabel 1 dengan 8 skenario dengan menggunakan *Remove Properties* pada sistem operasi *Windows* dibantu dengan aplikasi *Hex Editor* untuk menghilangkan nilai *metadata* (*width, height* dan nilai *RGBA*).

Gambar 2. Sampel *Image PNG*

Tabel 1. Skenario Gambar yang Digunakan

No.	*File Extention*	*Magic Number*	*Metadata*
1	✕	✕	✕
2	✓	✕	✕
3	✕	✓	✕
4	✕	✕	✓
5	✓	✓	✕
6	✓	✕	✓
7	✕	✓	✓
8	✓	✓	✓

Tabel 1 berisi skenario gambar yang dimanipulasi menggunakan Hex Editor seperti hanya mengandung parameter yang diberi tanda ceklis (✓) sedangkan tanda cakra (✕) dihapus dengan Hex Editor. File Extention Abc.php diganti dengan nama Abc.php untuk mengecek teknik penyaringan validasi File Extention seperti pada Gambar 3.



Gambar 3. Perubahan nama *file extention*

Data heksadesimal pada Berkas yang bisa disebut pula dengan *segments* beserta *fields marker* tersebut merupakan awalan nilai pada heksadesimal atau *Start Of Image* bernilai 89 50 4E 47 0D 0A 1A 0A diakhiri sampai panjang IHDR bernilai 00 00 00 0D dan IHDR bernilai 49 48 44 52 atau informasi *metadata* yang berisi nilai (*Width, Height dan Bits*) seperti pada Gambar 4.

```
0000000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 ; ‰PNG........IHDR
0000010h: 00 00 02 00 00 00 02 00 08 06 00 00 00 F4 78 D4 ; .............ôxÔ
0000020h: FA 00 00 00 04 73 42 49 54 08 08 08 08 7C 08 64 ; ú....sBIT....|.d
0000030h: 88 00 00 00 09 70 48 59 73 00 00 0B 13 00 00 0E ; ˆ....pHYs.......
```

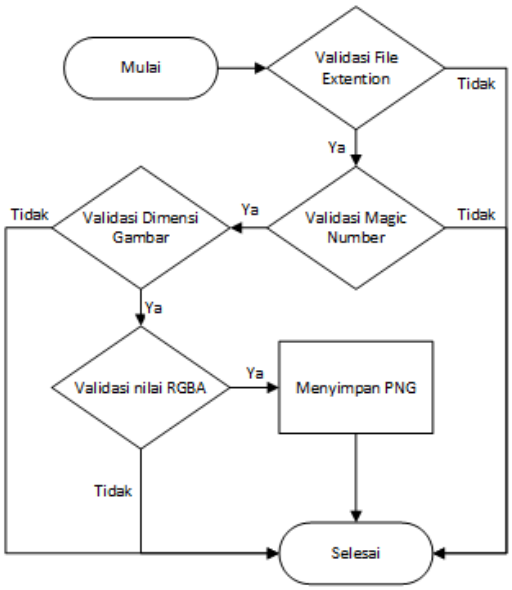
Gambar 4. Heksadesimal Metadata Berisi MIME (*Header File*) Pada Berkas Gambar PNG

Data nilai *RGBA* juga dihilangkan untuk mengecek teknik penyaringan validasi menggunakan Metadata dengan memanfaatkan GD Graphic Library sebagai *image processing*, seperti pada Gambar 5.

```
0000000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 ; ‰PNG........IHDR
0000010h: 00 00 02 00 00 00 02 00 08 06 00 00 00 F4 78 D4 ; .............ôxÔ
0000020h: FA 00 00 00 04 73 42 49 54 08 08 08 08 7C 08 64 ; ú....sBIT....|.d
0000030h: 88 00 00 00 09 70 48 59 73 00 00 0B 13 00 00 0E ; ˆ....pHYs.......
```

Gambar 5. Heksadesimal Metadata

B. Perancangan *(Design)*



Gambar 6. Alur Validasi Image PNG

Gambar 6 merupakan *Flowchart* teknik penyaringan validasi *Image PNG* menggunakan Metadata memanfaatkan *GD Graphic Library* pada PHP dengan membandingkan File Extention, Magic Number dan Metadata yang berisi nilai dimensi dan nilai RGBA. Nilai *red*, *green* dan *blue* memiliki range nilai 0 - 255 sedangkan nilai *alpha* memiliki range nilai 0 -127. Alur validasi ini diimplementasikan ke dalam tahap Pengkodean (*Coding*) dengan menggunakana pemrograman PHP.

C. Pengkodean *(Coding)*

Tabel 2. Algoritma Validasi Image PNG

```
<?php
$filePath = "Abc.png";
$errors = array();
$fileExtentionWhitelist = array("PNG");
$imageFileExtention  =  pathinfo($filename,
PATHINFO_EXTENSION);

if
(!in_array(strtoupper($imageFileExtention)
, $fileExtentionWhitelist)){
    array_push($errors,"File  Extention  is
not allowed");
}

function magicNumber($filename) {
    if(file_exists($filename)){
        $handle = fopen($filename, 'r');
        $bytes                          =
strtoupper(bin2hex(fread($handle, 2)));
        fclose($handle);
        return $bytes;
    }else{
        return false;
    }
}
$magicNumberWhitelist = array("8950");
if       (!in_array(magicNumber($filePath),
$magicNumberWhitelist)){
    array_push($errors,"Magic Number is not
allowed");
  }
$im = imagecreatefrompng($filePath);

if(empty(getimagesize($filePath))){
    array_push($errors,  "PNG  Dimension  is
not valid");
}

list($width,           $height)           =
getimagesize($filePath);

for ($x=0; $x < $width; $x++) {
    for ($y=0; $y < $height; $y++) {
    $rgba = imagecolorat($im, $x, $y);
    $colors    =    imagecolorsforindex($im,
$rgba);
    if(is_int($colors["red"])){
        if($colors["red"]     <     0     ||
$colors["red"] > 255){
            array_push($errors,  "Red  Color
is not valid");
        }
    }
    if(is_int($colors["green"])){
        if($colors["green"]    <    0    ||
$colors["green"] > 255){
            array_push($errors,      "Green
Color is not valid");
        }
    }
    if(is_int($colors["blue"])){
        if($colors["blue"]     <     0     ||
$colors["blue"] > 255){
            array_push($errors, "Blue Color
is not valid");
        }
    }
    if(is_int($colors["alpha"])){
        if($colors["alpha"]    <    0    ||
$colors["alpha"] > 127){
            array_push($errors,      "Alpha
Color is not valid");
        }
    }
    }
}

if(empty($errors)){
    echo "PNG is valid";
}else{
    foreach ($errors as $data) {
        echo $data."<br />";
    }
}

?>
```

Algoritma pemrograman pada Tabel 2 berfungsi untuk mengambil nilai dari dimensi berkas gambar yang telah diunggah oleh pengguna jika berisi sesuai nilai warnanya maka proses penyimpanan akan dilakukan sedangkan jika tidak berisi nilai maka akan menampilkan pesan kesalahan. Contoh informasi yang ada pada *getimagesize()* adalah sebagai berikut :

```
Array ( [0] => 512 [1] => 512 [2] => 3 [3] => width="512" height="512" [bits] => 32 [mime] => image/png )
```

Fungsi getimagesize() menghasilkan 6 elemen *array* seperti elemen [0] bernilai 512, elemen [1] bernilai 512, elemen [2] bernilai 3, elemen [3] bernilai width="512" height="512", elemen [*bits*] bernilai 32, elemen mime bernilan image/png.

D. Pengujian *(Testing)*

Tahapan Pengujian pada penelitian ini menggunakan 8 skenario kemudian di unggah sesuai dengan skenario uji yang telah ditentukan pada Tabel 1 hdan hasil pengujian tersaji pada Tabel 3.

Tabel 3. Skenario Uji pada Algoritma

No	Validasi			Hasil yang diharapkan	Kesimpulan
	File Extention	Magic Number	Metadata		
1	×	×	×	Gagal menggunggah	[✓] Berhasil [ ] Tidak Berhasil
2	✓	×	×	Gagal menggunggah	[✓] Berhasil [ ] Tidak Berhasil
3	×	✓	×	Gagal menggunggah	[✓] Berhasil [ ] Tidak Berhasil
4	×	×	✓	Gagal menggunggah	[✓] Berhasil [ ] Tidak Berhasil
5	✓	✓	×	Gagal menggunggah	[✓] Berhasil [ ] Tidak Berhasil
6	✓	×	✓	Gagal menggunggah	[✓] Berhasil [ ] Tidak Berhasil
7	×	✓	✓	Gagal menggunggah	[✓] Berhasil [ ] Tidak Berhasil
8	✓	✓	✓	Berhasil mengunggah	[✓] Berhasil [ ] Tidak Berhasil

Pengujian Skenario Uji dengan hasil pada Tabel 3 semua skenario uji telah melakukan ujicoba dengan membandingkan hasil yang diharapkan dengan hasil setelah implementasi kemudian kesimpulan dari hasil yang diharapkan telah berhasil atau sesuai dengan akurasi 100% dari 8 skenario uji.

E. Peningkatan Perangkat Lunak *(Software Increment)*

Penelitian ini dapat diterapkan dalam Algoritma *Image PNG File Upload* setelah validasi lainnya misalnya seperti validasi *File Extention*, *Magic Number* dan yang lainnya.

SIMPULAN

Penelitian ini menghasilkan algoritma yang dapat memberikan keamanan dalam proses pengunggahan berkas pada aplikasi berbasis *web* khususnya berjenis *PNG File*. Metode *Extreme Programming (XP)* digunakan untuk mengembangkan atau membangun perangkat lunak karena banyaknya perubahan secara cepat atau dinamis dalam proses pembuatan.

Teknik validasi keamanan penanganan *Image PNG File Upload* menggunakan *Metadata* lebih baik dibanding hanya menggunakan *File Extention* dan *Magic Number* saja yang dapat menyaring berkas *Image PNG* hingga dapat memilah berkas yang berstatus *corrupt* atau tidaknya sebuah *Image PNG* dengan mengecek validasi secara khusus dengan menyaring secara khusus sesuai keunikan atau karakteristik dari *Image File* yaitu mempunyai metadata (*Width, Height, Bits, Mime*) serta *RGBA (Red, Green, Blue, Alpha)* pada sebuah gambar PNG. Berdasarkan hasil pengujian yang dilakukan dari 8 skenario yang dipersiapkan menghasilkan keberhasilan 100%.

DAFTAR PUSTAKA

Almi, S. B. (2014). Web Server Security and Survey on Web Application Security. *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)*, *2*(1), 114–119. http://ijritcc.org/IJRITCC Vol_2 Issue_1/Web Server Security and Survey on Web Application Security.pdf

Anwar, F., Fadlil, A., & Riadi, I. (2019). ANALISA KEAMANAN IMAGE JPEG FILE UPLOAD MENGGUNAKAN METADATA DAN GD GRAPHIC LIBRARY PADA APLIKASI BERBASIS WEB. In U. Krisnadwipayana (Ed.), *Seminar Nasional Teknologi Fakultas Teknik Universitas Krisnadwipayana* (pp. 479–487). Universitas Krisnadwipayana.

Boutell, T., & Joye, P. (n.d.). *About*. Retrieved May 14, 2019, from https://libgd.github.io/pages/about.html

Chen, H., Zhang, L. J., Hu, B., Long, S. Z., & Luo, L. H. (2015). On Developing and Deploying Large-File Upload Services of Personal Cloud Storage. *Proceedings - 2015 IEEE International Conference on Services Computing, SCC 2015*, 371–378.

https://doi.org/10.1109/SCC.2015.58

Li, X., & Xue, Y. (2011). A survey on web application security. *Nashville, TN USA*. http://isis.vanderbilt.edu/sites/default/files/main_0.pdf

OWASP. (2013). *OWASP Top 10 - The Ten Most Critical Web Application Security Risks*. OWASP Top 10. https://www.owasp.org/images/f/f8/OWASP_Top_10_-_2013.pdf

Pooj, K., & Patil, S. (2016). Understanding File Upload Security for Web Applications. *International Journal of Engineering Trends and Technology*, *42*(7), 342–347. https://doi.org/10.14445/22315381/ijett-v42p261

Poornima, R., & Iswarya, R. J. (2013). An Overview Of Image Steganography. *International Journal of Computer Science & Engineering Survey*, *4*(1), 23–31.

Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering : a practitioner's approach* (8th ed.). McGraw-Hill Education.

Rahmatulloh, A., Rachman, A. N., & Anwar, F. (2019). Implementasi Web Push Notification pada Sistem Informasi Manajemen Arsip Menggunakan PUSHJS. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *6*(3), 327–334. https://doi.org/10.25126/jtiik.20196936

Riadi, I., Fadlil, A., & Sari, T. (2017). Image Forensic for detecting Splicing Image with Distance Function. *International Journal of Computer Applications*, *169*(5), 6–10. https://doi.org/10.5120/ijca2017914729

Saifudin, S., & Fadlil, A. (2015). Sistem Identifikasi Citra Kayu Berdasarkan Tekstur Menggunakan Gray Level Coocurrence Matrix (GLCM) Dengan Klasifikasi Jarak Euclidean. *Sinergi*, *19*(3), 181. https://doi.org/10.22441/sinergi.2015.3.003

Sajjad, R., Mamoona, H., Zartasha, G., Ansar, A., & Hasan, J. (2015). Systematic Review of Web Application Security Vulnerabilities Detection Methods. *Journal of Computer and Communications*, *3*, 28–40. https://doi.org/10.1007/s10462-012-9375-6

Sari, T., Riadi, I., & Fadlil, A. (2016). Forensik Citra untuk Deteksi Rekayasa File Menggunakan Error Level Analysis. *Annual Research Seminar 2016*, *2*(1), 133–138. http://ars.ilkom.unsri.ac.id

Sulistyo, W. Y., Riadi, I., & Yudhana, A. (2018). *Analisis Deteksi Keaslian Citra Menggunakan Teknik*. *2018*(November), 154–159.

Umar, R., Hadi, A., Widiandana, P., Anwar, F., Jundullah, M., & Ikrom, A. (2019). Perancangan Database Point of Sales Apotek Dengan Menerapkan Model Data Relasional. *Query: Jurnal Sistem Informasi*, *03*(02), 33–41.

W, Y., Riadi, I., & Yudhana, A. (2018). Analisis Deteksi Vulnerability pada Webserver Open Jurnal System menggunakan OWASP Scanner. *JURTI*, *2*(1), 1–8.