Jurnal CyberTech

Vol.4. No.5, Mei 2021, pp. xx~xx
P-ISSN : 9800-3456
E-ISSN : 2675-9802

Implementasi Pendeteksi Serangan *SQL (Structured Query Language)* Dengan Metode *Randomization of Query*

Hengky Pranata[1], Puji Sari Ramadhan[2], Tugiono[3]

[1,2,3] Program Studi Sistem Informasi, STMIK Triguna Dharma

Article Info

Article history:
Received May 12th, 2021
Revised May 20th, 2021
Accepted May 29th, 2021

Keyword:
SQL Injection
Website
Keamanan
Randomization of Query

ABSTRACT

Saat ini pada sistem aplikasi website banyak terjadinya pencurian data yang terjadi. Pada sistem aplikasi website terdapat banyak data data penting yang disimpan dalam sebuah database. Salah satu ancaman dari pencurian data tersebut yaitu serangan SQL Injection. Serangan SQL Injection memungkinkan penyerang dapat masuk kedalam sistem aplikasi website tanpa menggunakan otoritas yang sah. Sehingga penyerang dapat melakukan tindak kejahatan seperti pencurian data. Beberapa solusi yang dapat digunakan dalam mengatasi masalah tersebut yaitu dengan menerapkan pendeteksian terjadinya serangan SQL Injection dengan metode Randomization of Query. Dengan adanya metode ini sistem dapat mencegah terjadinya sebuah serangan SQL Injection. Dengan berhasil dibangunnya sistem pendeteksian terjadinya SQL injection dengan metode Randomization of Query sangat terbantu mecegah terjadinya pencurian data pada sebuah sistem aplikasi website.

Corresponding Author:
Nama : Hengky Pranata
Program Studi : Sistem Informasi
Kampus : STMIK Triguna Dharma
Email : hengkypranata13@gmail.com

1. PENDAHULUAN

Pada zaman sekarang ini banyak kegiatan ataupun aktivitas yang sudah menggunakan sistem informasi online. Salah satunya yang sering di temukan yaitu penggunaan website. Banyak kalangan yang sudah mulai mempergunakan website sebagai wadah untuk memberikan informasi kepada calon pengunjungnya. Mulai dari perusahaan, komunitas, bahkan perorangan sudah mulai menciptakan website sendiri.

Banyak sekali kejahatan yang dapat di temukan dalam sistem online pada website, atau sering disebut dengan Cyber Crime. Beberapa diantaranya seperti pencurian data, defacing, malware, cyber terrorism, konten ilegal, dan masih banyak yang lainnya. Diantara kejahatan dalam internet diatas, salah satu kejahatan yang paling merugikan dan sering terjadi yaitu pencurian data[1].

Banyak metode pencurian data yang digunakan peretas untuk menembus sistem keamanan pada website. Salah satu yang paling populer yaitu serangan SQL Injection. Bahkan menurut Akamai pada www.cbronline.com pada 2017 serangan SQL Injection menjadi serangan cyber yang paling banyak terjadi yaitu hampir mencapai angka 44% kejadian [2].

Serangan SQL Injection itu sendiri memungkinkan penyerang dapat masuk kedalam sistem database website yang sudah di bangun tanpa harus menggunakan akses yang sah. Serangan SQL Injection sendiri memiliki dampak yang cukup besar, Penyerang dapat mencuri data yang tersimpan pada website. Bahkan penyerang dapat merubah isi data (insert/update/delete) yang ada pada database website [3].

2. METODE PENELITIAN

2.1 Metode Penelitian

Teknik pengumpulan data dengan melakukan tinjauan langsung ketempat studi kasus dimana akan dilakukan penelitian. Dalam hal ini dilakukan observasi di toko Berkah Ponsel. Pada toko tersebut dilakukan analisis masalah yang dihadapi kemudian diberikan sebuah *resume* atau kesimpulan Berikut metode dalam penelitian ini adalah sebagai berikut :

1. Observasi

Observasi dilakukan dengan melakukan pengamatan pada website target untuk melakukan identifikasi terhadap masalah yang ada pada website. Dilakukan pengamatan pada beberapa bagian website yang rentan akan serangan SQL Injection.

2. Studi Pustaka

Dilakukan studi pustaka terkait dengan masalah yang sudah ada untuk menentukan metode terbaik yang dapat digunakan sebagai solusi untuk menyelesaikan masalah. Metode yang digunakan adalah metode Randomization of Query sebagai pendeteksi serangan SQL Injection pada website yang akan menjadi target.

3. Eksperimen

Pada tahap eksperimen akan dilakukan sebuah uji coba terhadap solusi dengan menyusun beberapa rangkaian kode dengan menggunakan bahasa pemrograman PHP. Sebuah kode PHP akan dibuat sebagai uji coba dalam pendeteksian serangan SQL Injection pada website target.

4. Analisa Hasil

Proses analisa terhadap hasil eksperimen yang telah dilakukan untuk mendapatkan kesimpulan dari uji coba yang dilakukan untuk memperoleh hasil yang sesuai dengan tujuan yang akan dicapai. Pada tahap ini akan dilakukan analisa terhadap kode program yang telah dibuat menggunakan bahasa PHP.

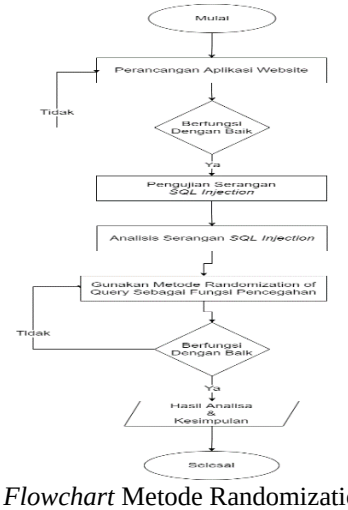
2.2 Algoritma Sistem

Algoritma sistem adalah suatu proses logis tertentu yang digunakan untuk dapat menyelesaikan suatu masalah yang ada. Masalah yang dimaksud yaitu masalah dalam mendeteksi adanya serangan SQL Injection yang terjadi dalam sebuah website dan pada kasus ini metode yang akan digunakan untuk mendeteksi terjadinya serangan SQL Injection pada suatu website yaitu Randomization of Query.

2.3 Metode Perancangan Sistem

1. *Flowchart* Metode Randomization of Query

Flowchart ini menggambar urutan dari suatu program kerja secara keseluruhan menggunakan metode *Metode Randomization of Query* mulai dari awal sampai akhir prosesnya.



Gambar 2.1 *Flowchart* Metode Randomization of Query

2. Perancangan Aplikasi Website

Tahap pertama dalam merancang aplikasi website adalah membuat sebuah database baru yang akan digunakan pada sebuah website. Jenis database yang akan digunakan adalah database MySQL, karena database MySQL cukup popular digunakan oleh semua kalangan.

3. Pengujian Serangan SQL Injection

Pada proses ini akan dilakukan pengujian dengan melakukan uji coba serangan mengikuti pola kejahatan SQL Injection yang sering terjadi yaitu salah satunya masuk melalui halam login secara ilegal.

Pada tahap selanjutnya akan dilakukan uji coba serangan *SQL Injection* pada *form login.* Pada *field username* akan dimasukkan "*admin' or 1=1#"* dan pada *field password* akan dimasukkan "1234", Maka halaman akan tetap beralih ke *dashboard.php* meskipun *username* dan *password* yang dimasukkan tidak sesuai dengan data yang terdapat pada tabel *user.* Hal tersebut dikarenakan jika menggunakan username : admin' or 1=1# dan password : 1234, Maka query yang terbentuk yaitu :

SELECT user_id, username FROM user WHERE username = 'admin' or 1=1 #' and password = '21232f297a57a5a743894a0e4a801fc3' AND is_admin = 1 AND is_active = 1 LIMIT 1

4. Pencegahan Menggunakan Metode Randomization of Query

Pada tahap ini akan dimasukkan sebuah fungsi yang dapat mendeteksi terjadinya sebuah serangan SQL injection. Memasukkan fungsi Randomization of Query memerlukan sedikit usaha untuk memahami bahasa SQL.

Untuk membuat sebuah fungsi Randomization of Query diperlukan sebuah program yang dapat membaca pernyataan SQL dan menulis ulang semua kata kunci dengan kunci acak seperti ini :

SELECT username FROM user WHERE username = '' AND password = ''

Program akan mengidentifikasi kata kunci dalam *query* dan menambahkan kata kunci kedalam setiap *query* (misalnya dengan menggunakan kata kunci '999') :

SELECT999 username FROM999 user WHERE999 username = '' AND999 password = ''

Selanjutnya memasukkan fungsi Randomization of Query kedalam program PHP seperti berikut ini :

```
<?php
If(isset($_POST)['submit'])){
$dbHost = "localhost";
$dbUser = "root";
$dbPass = ""
$dbDatabase = "tgd";
$db = new
$sql = $db->prepare("SELECT * FROM user WHERE username = ? AND password = ? LIMIT 1");
$sql->bindValue(1,$_POST["username"]);
$sql->bindValue(2,$_POST["password"]);
$sql->execute();
If($sql->rowCount()==1){
$row = $sql->fetch($sql);
session_start();
header("Location : login.php");
}else{
Echo"Salah";
Exit;
}
}else{
header("location: index.php);
exit
}
```

3. ANALISA DAN HASIL

3.1 Kebutuhan Sistem

Dalam implementasi data *mining* untuk menganalisa pola pembelian dengan menggunakan algoritma FP-Growth pada Berkah Ponsel membutuhkan dua perangkat yaitu, perangkat lunak (*software*) dan perangkat keras (*hardware*) untuk menguji sistem yang telah dirancang.

3.1.1 Perangkat Lunak (*Software*)

Berikut ini spesifikasi perangkat lunak yang dibutuhkan untuk menjalankan sistem yaitu sebagai berikut :

1. Xampp Control Panel v3.3.0
2. MySQL Database Server 7.4.20
3. Sublime Text Editor 3
4. Googlc Chrome
5. Sistem Operasi Windows 10

3.1.2 Perangkat Keras (Hardware)

Spesifikasi *hardware* yang digunakan dalam implementasi sistem agar berjalan dengan baik dan lancar adalah sebagai berikut :

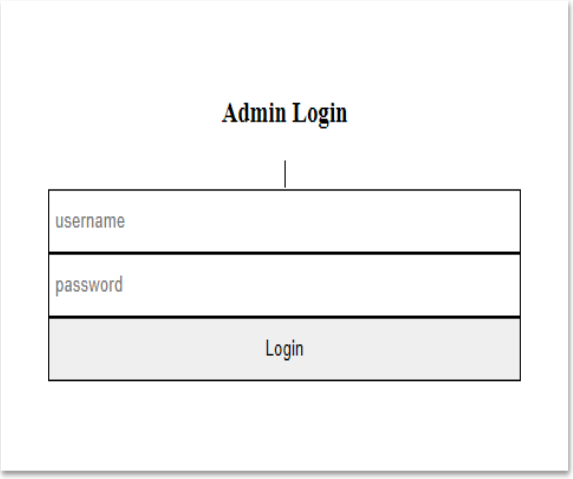
1. CPU (Central Processing Unit)
2. Ram 2 GB
3. Monitor
4. Keyboard

3.2 Hasil Tampilan Antarmuka

Hasil tampilan antarmuka ini merupakan *form* atau menu-menu beserta fungsinya yang ada pada aplikasi *visual basic* yang digunakan untuk mempermudah admin. *Form* ini akan memberikan *input* dan menampilkan *output* dari aplikasi.

1. Tampilan Halaman Login

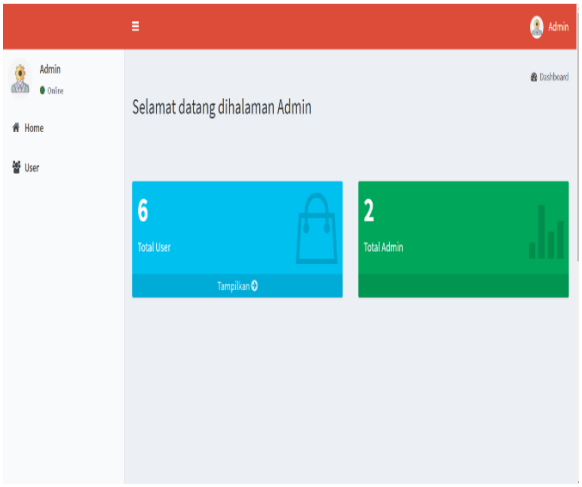
Pada tampilan *login* ini, admin akan memasukkan *username* dan password. Menu *login* ini bertujuan agar tidak sembarang orang bisa mengakses menu- menu yang ada pada aplikasi.



Gambar 3.1 Tampilan Login

2. Tampilan Halaman Dashboard

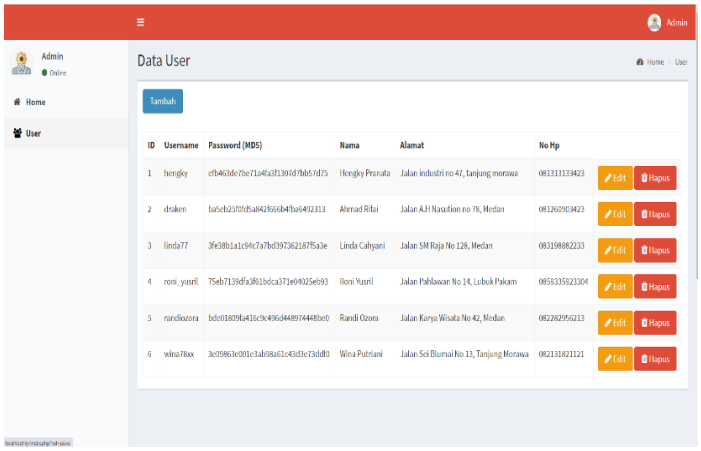
Setelah *login* berhasil, kemudian admin akan memasuki menu utama. Pada menu utama ini terdapat beberapa menu yang dapat digunakan oleh admin, antara lain : menu input data, menu proses *FP-Growth*, menu Laporan dan *exit*.



Gambar 3.2 Tampilan Halaman Dashboard

3. Tampilan Menu User

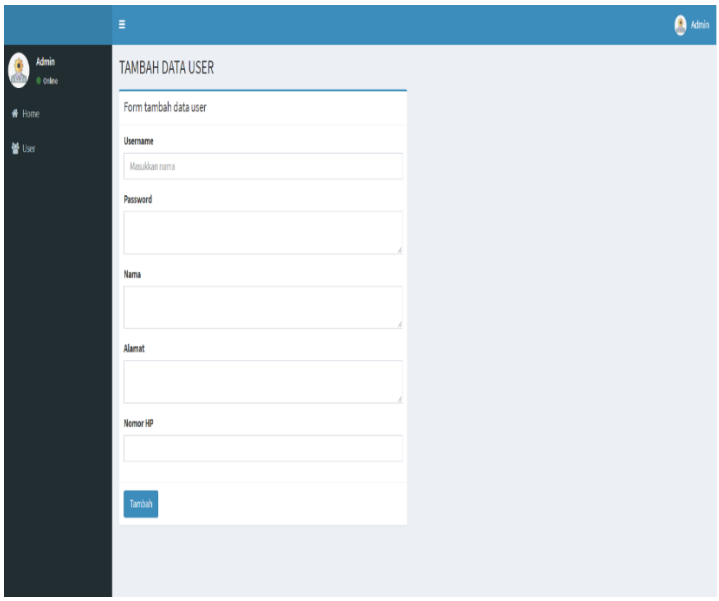
Pada menu user sistem akan menampilkan data user yang terdapat dalam *database*



Gambar 3.3 Tampilan Menu User

4. Tampilan Halaman Tambah User

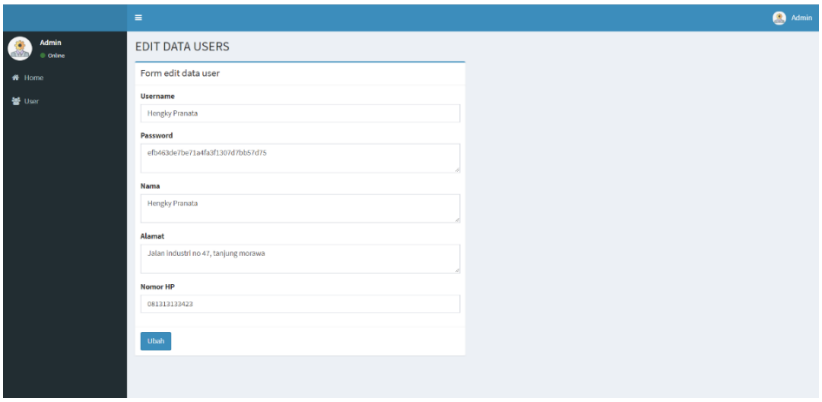
Terdapat form yang dapat digunakan untuk menambah data user kedalam *database*.



Gambar 3.4 Tampilan Halaman Tambah User

5. Tampilan Halaman Edit User

Terdapat form yang digunakan untuk melakukan edit data user yang akan disimpan kedalam database.



Gambar 3.5 Tampilan Halaman Edit User

3.3 Pengujian

Pengujian dilakukan dengan tujuan untuk mengetahui sistem aplikasi website yang telah dirancang sesuai dengan kebutuhan untuk mendeteksi terjadinya serangan SQL Injection pada sebuah website. Setelah dilakukan pengujian maka dapat diberikan sebuah keputusan untuk menentukan ada atau tidaknya sebuah serangan SQL Injection.

4. KESIMPULAN

4.1 Kesimpulan

Adapun kesimpulan yang dapat diperoleh dari penelitian ini adalah sebagai berikut :

1. Berdasarkan hasil analisa pada sistem aplikasi website yang telah dirancang, sebuah website memerlukan keamanan tambahan yang diperlukan saat melakukan aksi login untuk mendeteksi adanya serangan yang terjadi pada sebuah website.
2. Metode Randomization of Query merupakan salah satu metode yang tepat digunakan untuk digunakan dalam pencegahan terjadinya serangan SQL Injection.
3. Sistem aplikasi website yang telah dirancang dapat memudahkan admin untuk melakukan aksi perubahan atau penambahan data yang terdapat dalam database.

4.2 Saran

Untuk mendapatkan hasil penelitian yang lebih baik kedepannya, berikut merupakan beberapa saran yang diharapkan yaitu :

1. Sistem aplikasi website yang dirancang perlu dikembangkan lebih luas dengan menambah fitur fitur lainnya agar website yang dirancang tampak lebih menarik untuk digunakan.
2. Metode pendeteksian dan pencegahan yang digunakan yaitu menggunakan metode Randomization of Query.
3. Pengembangan pada data yang terdapat dalam database sesuai dengan kebutuhan website.
4. Perancangan aplikasi website dapat menggunakan Laravel, Bootsrap dan Codeigniter.

UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan karunia-Nya sehingga dapat menyelesaikan jurnal ilmiah ini. Pada kesempatan ini diucapkan terima kasih yang sebesar-besarnya kepada Ibunda tercinta yang selama ini memberikan do'a dan dorongan baik secara moril mau pun materi sehingga dapat terselesaikan pendidikan dari tingkat dasar sampai bangku perkuliahan dan terselesaikannya jurnal ini. Di dalam penyusunan jurnal ini, banyak sekali bimbingan yang didapatkan serta arahan dan bantuan dari pihak yang sangat mendukung. Oleh karena itu dengan segala kerendahan hati, diucapkan terima kasih yang sebesar-besarnya kepada kepada ketua yayasan STMIK Triguna Dharma, kepada Bapak Puji Sari Ramadhan S.Kom., M.Kom selaku dosen pembimbing 1, kepada Bapak Tugiono S.Kom., M.Kom selaku dosen pembimbing 2, kepada kedua orang tua saya yang selalu memberikan dukungan dan doa kepada saya serta tidak lupa kepada teman-teman saya seperjuangan.

REFERENSI

[1] D. Aulia, "Studi keamanan sistem informasi berbasis," pp. 26–37, 2017.

[2] A. M. I. Zulkifli, I. Riadi, and Y. Sugiantoro, "Live Forensics Method For Analysis Denial of Service (DoS) Attack on Routerboard," vol. 180, no. 35, 2018.

[3] D. RANGGA, "IMPLEMENTASI SISTEM PENDETEKSI SERANGAN SQL INJECTION DENGAN MENGGUNAKAN ALGORITMEK- NEAREST NEIGHBOR," pp. 1–53, 2020.

[4] F. Yudha and A. Muryandi, "Aplikasi Pengujian Celah Keamanan Pada Aplikasi Berbasis Web," CyberSecurity dan Forensik Digit., vol. 1, no. 1, pp. 1–6, 2018, [Online]. Available: http://ejournal.uin-suka.ac.id/saintek/cybersecurity/article/view/1101/1153.

[5] D. Hariyadi, D. P. I. Kusuma, N. H. Maulida, and M. Ma'rifat, "Evaluasi Potensi Celah Keamanan SQL Injection Menggunakan Nearest Neighbor pada Security-Software Development Life Cycle," J. Repos., vol. 2, no. 9, pp. 1273–1280, 2020, doi: 10.22219/repositor.v2i9.999.

[6] G. M. Barek, K. E. Nurnawati, and M. SHOLEH, "RANCANG BANGUN APLIKASI PENCARIAN PERGURUAN TINGGI," J. Scr., vol. 7, no. 2, pp. 232–238, 2019.

[7] E. Gunadhi and A. P. Nugraha, "Penerapan Kriptografi Base64 Untuk Keamanan URL (Uniform Resource Locator) Website Dari Serangan SQL Injection," J. Algoritm. Sekol. Tinggi Teknol. Garut, vol. 13, no. 2, pp. 391–398, 2017, doi: 10.33364/algoritma/v.13-2.391.

[8] O. C. Abikoye, A. Abubakar, A. H. Dokoro, O. N. Akande, and A. A. Kayode, "A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm," Eurasip J. Inf. Secur., vol. 2020, no. 1, 2020, doi: 10.1186/s13635-020-00113-y.

[9] H. Gaddam, M. Maheshwari, and G. Harshavardhan, "SQL Injection-Biggest Vulnerability of the Era," EasyChair Prepr., no. 4175, 2020.

[10] S. Raut, A. Nikhare, Y. Punde, S. Manerao, and S. Choudhary, "A Review on Methods for Prevention of SQL Injection Attack," Int. J. Sci. Res. Sci. Technol., vol. 6, no. 2, pp. 463–470, 2019, doi: 10.32628/ijsrst196258.

[11] M. Thiyab, M. Ali, and F. Basil, "The Impact of SQL Injection Attacks on the Security of Databases," Proc. 6th Int. Conf. Comput. Informatics, no. 080, pp. 323–331, 2017, [Online]. Available: https://www.researchgate.net/publication/316609616_THE_IMPACT_OF_SQL_INJECTION_ATTACKS_ON_THE_SECURITY_OF_DATABASES.

[12] Y. Yulianingsih, "Menangkal Serangan SQL Injection Dengan Parameterized Query," J. Edukasi dan Penelit. Inform., vol. 2, no. 1, pp. 46–49, 2016, doi: 10.26418/jp.v2i1.

[13] M. R. Efendi, L. V. Yovita, and Hafidudin, "Analisis Penanganan SQL Injection pada Basis Data MySQL dengan Framework Code Igniter dan PHP," 2016.

[14] D. Chen, Q. Yan, C. Wu, and J. Zhao, "SQL Injection Attack Detection and Prevention Techniques Using Machine Learning," J. Phys. Conf. Ser., vol. 1757, no. 1, p. 012055, 2021, doi: 10.1088/1742-6596/1757/1/012055.

[15] S. Raj and E. Sherly, "SQL Injection Attack Prevention by Direct Reverse Resemblance Technique," Int. J. Pure Appl. vol. 118, no. 16, pp. 599–614, 2018, [Online]. Available: https://acadpubl.eu/jsi/2018-118-16-17/articles/16/39.pdf.

[16] A. . Ridhlo, PANDUAN PEMBUATAN FLOWCHART. academi.edu, 2017.[Online].Available:https://www.academia.edu/34767055/Pedoman_P embuatan_Flowchart

[17] Rosa A.s and M.Shalahuddin, Rekayasa Perangkat Lunak. Bandung: Informatika Bandung, 2018.

[18] Muntihana. Vimila, "Analisis Dan Perancangan Sistem Informasi Berbasis Web Dan Android Pada Klinik Gigi Lisda Medica Di Kabupaten Bulukumba Sulawesi Selatan" Jurnal Sistem Informasi, Teknologi Informatika dan Komputer, Vol.9, No.1, Sept 2018.

[19] R. Setiawan and R. Gunawan, "Perancangan Aplikasi Pengelolaan Dana Bantuan Operasional Sekolah di Sekolah Menengah Atas," J. Algoritm., vol. 14, no. 2, pp. 154–163, 2017, [Online]. Available: http://juornals.sttgarut.ac.id.

[20] Munawar, Analisis Perancangan Sistem Berorientasi Objek dengan UML (Unified Modeling Language), Informatika Bandung, Bandung, 2018.

BIBLIOGRAFI PENULIS

Nama	:	Hengky Pranata
NIDN	:	2017020779
Program Studi	:	Sistem Informasi STMIK Triguna Dharma
Deskripsi	:	Mahasiswa STMIK Triguna Dharma Stambuk 2017 Program Studi Sistem Informasi.

Nama	:	Puji Sari Ramadhan S.Kom., M.Kom
NIDN	:	0126039201
Program Studi	:	Sistem Informasi
Deskripsi	:	Dosen tetap STMIK Triguna Dharma yang aktif mengajar pada bidang keilmuan kecerdasan buatan dan data sains. Telah menulis 1 buku dibidang ilmu computer. Memiliki sebanyak 2 Hak Kekayaan Intelektual (HKI). Menjabat sebagai Ketua Program Studi Sistem Informasi
Prestasi	:	Dosen Terbaik Tahun 2018, Pemenang PDP 2018 dan 2019.

Nama	:	Tugioono S.Kom., M.Kom
NIDN	:	0111068302
Program Studi	:	Sistem Informasi
Deskripsi	:	Dosen tetap STMIK Triguna Dharma yang aktif mengajar dan fokus pada bidang keilmuan Pemrograman Visual, Sistem Pendukung Keputusan dan Sistem Manajemen Basis Data.