

Teknik Pencegahan Penetrasi SQL Injeksi Dengan Pengaturan Input Type Number dan Batasan Input Pada Form Login Website

Sahat Parulian Sitorus¹, Rahmad Aditiya²

¹Universitas Labuhanbatu, Rantauprapat, Indonesia

²Universitas Alwashliyah Labuhanbatu, Rantauprapat, Indonesia

Info Artikel

Article history:

Received: 14 08 2020

Revised: 13 09 2020

Accepted: 22 09 2020

Kata Kunci:

Implementasi *Maxlength*

SQL Injeksi

Penetrasi

Website

Penulis Korespondensi:

sahatparuliansitorus4@gmail.com

Abstract

In the world of IT, websites are very vulnerable to hacker attacks in various types of ways so that they can break into the security of the target website. SQL injection attacks are often carried out on breaching websites from the login form by entering a special injection username and password so that the website can be hacked easily. In securing a website from injection attacks, there are various ways, one of which is by using the max length technique and input type number. The *maxlength* technique and input type number are made in the form of php or html source code which is inserted into the login form source code in the username and password input section. One of the advantages of this *maxlength* and input type technique is that it limits the input of the username and changes the format of the password input to only numbers, which means it will prevent hackers from forcibly penetrating the SQL injection attack on the website.

Abstrak

Dalam dunia IT pada *website* sangat rentan akan serangan *hacker* dengan berbagai jenis cara agar mereka dapat membobol keamanan *website* target. Serangan SQL Injeksi sering dilakukan pada pembobolan *website* dari *form login* dengan menginputkan *username* dan *password* khusus injeksi sehingga *website* dapat dibobol dengan mudah. Dalam mengamankan sebuah *website* dari serangan injeksi beragam caranya salah satunya dengan menggunakan teknik *maxlength* dan input type number. Teknik *maxlength* dan input type number ini dibuat dalam bentuk *source code* php atau html yang disisipkan kedalam *source code form login* pada bagian input *username* dan *password*. Salah satu keunggulan teknik *maxlength* dan input type ini akan membuat batasan inputan *username* dan mengubah format inputan *password* hanya bertipekan angka saja yang artinya akan mencegah *hacker* dalam melakukan penetrasi secara paksa dalam serangan SQL Injeksi pada *website*.

1. PENDAHULUAN

Teknologi dibidang IT yaitu pada website saat ini sangat berkembang pesat tidak dapat dibendung seiring perkembangan zaman. Website sudah banyak yang berkembang dari mulai platform php native, cms bahkan platform frame work yang saat ini sudah tersebar dan banyak digunakan oleh hamper seluruh lini aktivitas kegiatan. Dalam pembuatan website hal yang paling rentan menjadi masalah adalah sisi keamanan sistem website tersebut. Saat ini banyak terjadi pembobolan situs website dari mulai situs website perusahaan sampai dengan situs website pemerintahan. Beragam macam jenis-jenis serangan terhadap keamanan website yakni dengan membobol login admin dengan menggunakan banyak teknik yakni ada menggunakan serangan brute force dan yang paling familiar menggunakan serangan SQL injeksi.

Mengamankan website dengan menggunakan teknik *maxlength* dan input type number ini akan mengubah inputan form login pada username terbatas hanya 4-8 karakter saja dan mengubah inputan password yang seharusnya type text atau type password diubah menjadi type number. Mengubah kedua parameter ini dengan batasan input karakter dan type inputan number membuat kinerja serangan SQL Injeksi

tidak akan berpengaruh dan tidak dapat berjalan sehingga website akan aman dari bentuk serangan SQL Injeksi Secara Penetrasi atau secara paksa.

Metode untuk menghindari SQL Injection dapat dilakukan kedalam dua cara yaitu secara client-side dan server-side. Pada metode client-Side yaitu menerima 'Shadow SQL Query' dari server-side dan melakukan pengecekan terhadap deviasi yang terjadi antara shadow query dengan query dinamis yang dibentuk oleh masukan dari pengguna. Jika ditemukan adanya deviasi maka dapat dipastikan bahwa masukannya tidak benar (Malicious). Dengan demikian diperlukan sebuah metode yang mampu memberikan solusi tepat. Untuk dapat menentukan metode yang tepat maka dianggap perlu mengenal karakter dan cara kerja dari kegiatan penyerangan tersebut. Dalam penelitian ini diperkenalkan jenis penyerangan dengan cara SQL injection secara server side scripting untuk kemudian diberikan solusi menghindarinya. Dalam penelitian ini dianggap perlu mengangkat jenis serangan SQL Injection dikarenakan bahwa total serangan terhadap situs-situs yang ada di Indonesia adalah 28.430.843 dan jenis serangan paling besar adalah melalui SQL.

SQL Injeksi Merupakan sebuah kerentanan yang menyebabkan seorang penyerang memiliki kemampuan untuk mempengaruhi query SQL yang dikirimkan melalui aplikasi ke database. Dengan kemampuan tersebut seorang penyerang dapat mempengaruhi syntax SQL, kekuatan, fleksibilitas dari database pendukung fungsional dan mempengaruhi fungsi sistem operasi yang dialokasikan untuk database. SQL Injection tidak hanya mempengaruhi aplikasi web tapi juga semua program lain yang menggunakan kalimat SQL. Semua program yang menggunakan input dinamis dari luar (untrusted) dapat terserang oleh SQL. Dalam mengantisipasi serangan SQL Injeksi ini beragam pula dari mulai memberikan filter dari source code untuk query sampai dengan menggunakan verifikasi robot. Dalam pengamanan website dari serangan SQL Injeksi sebenarnya ada yang sederhana yang jarang diketahui oleh banyak programmer dan developer website yaitu dengan cara menggunakan teknik maxlength dan input type number yang artinya adalah pertama membatasi jumlah karkater inputan username dan mengubah type inputan text atau password menjadi jenis number. Pada prinsipnya serangan SQL Injeksi ini melalui form login admin dengan melakukan injeksi menggunakan software khusus sampai denegan menggunakan cara paksa atau disebut penetrasi yang mana hacker menerobos masuk dengan menginputkan username dan password khusus injeksi dengan kolaborasi angka dan huruf dengan parameter karakter yang ditentukan panjangnya.

Beberapa hal yang membuka kesempatan bagi penyerang untuk melakukan SQL Injection yaitu Memanfaatkan keuntungan dari sebuah aplikasi yang tidak terlindungi pada fungsi autentikasi pengguna karena tidak adanya validasi. Umumnya seorang penyerang membajak login field yang tidak terlindungi untuk memperoleh akses database. SQL Interpreter tidak dapat membedakan antara perintah yang dimaksud dengan kode yang di-inject oleh penyerang yang kemudian dieksekusi dan mengakibatkan tereksposnya database. Penyerang kemudian dapat mengakali aplikasi untuk mengekstrak semua data, menanamkan malicious script.

2. PEMBAHASAN

Pada penelitian ini penulis menggunakan metode SDLC Waterfall. Waterfall atau Classic Life Cycle merupakan metode yang banyak digunakan pada Software Engineering, metode ini melakukan pendekatan secara sistematis dan terurut dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, implementasi, dan pengujian sistem. Disebut Waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Adapun tahapan-tahapan pada penelitian ini adalah:

2.1 Pengumpulan Data

Tahapan ini dilakukan untuk memperoleh data yang diperlukan dalam pembuatan dan pengembangan *system website* dari mulai pengumpulan gambar, isi konten yang dimuat, *source code* dan pendukung data lainnya.

2.2 Analisis

Tahapan analisis dilakukan untuk menganalisis permasalahan dan menentukan kebutuhan yang diperlukan dalam pembuatan sistem. Hasil analisis tersebut kemudian dijadikan dasar dalam membuat perancangan desain sistem.

2.3 Desain

Tahapan desain dilakukan untuk mengetahui alur data dan proses yang terjadi sistem. Perancangan desain sistem dilakukan menggunakan *Unified Modeling Language (UML)*.

2.4 Implementasi Sistem

Tahapan implementasi dilakukan untuk menerjemahkan desain yang telah dibuat menggunakan bahasa pemrograman agar dapat dikembangkan menjadi sebuah sistem atau perangkat lunak. Sistem akan dibuat dan dikembangkan berbasis web menggunakan *framework* php native untuk mencoba melihat dua perbandingan *system* keamanan yang digunakan pada salah satu *website* yang dibuat sebagai bahan uji coba.

2.5 Pengujian Sistem

Tahapan pengujian sistem dilakukan untuk mengetahui apakah sistem yang dibuat telah sesuai dengan tujuan yang direncanakan. Pengujian sistem dilakukan menggunakan metode *black-box testing*. Berikut merupakan gambar ilustrasi dari metode penelitian yang dilakukan:



Gambar 1. Metode Penelitian

3. PEMBAHASAN

Pada bagian ini penulis hanya mengumpulkan data berupa dua rancangan *website* yang rentan akan serangan SQL Injeksi tanpa menggunakan teknik *maxlength* dan input type number. Kemudian rancangan *website* yang kedua merupakan pengembangan dengan menggunakan teknik *maxlength* dan input type number untuk mencegah penetrasi paksa dalam SQL Injeksi dengan pengujian dari server penyedia layanan hosting serta pengumpulan data sintax SQL Injeksi yang biasa digunakan hacker untuk meretas *website* dapat dilihat pada table berikut.

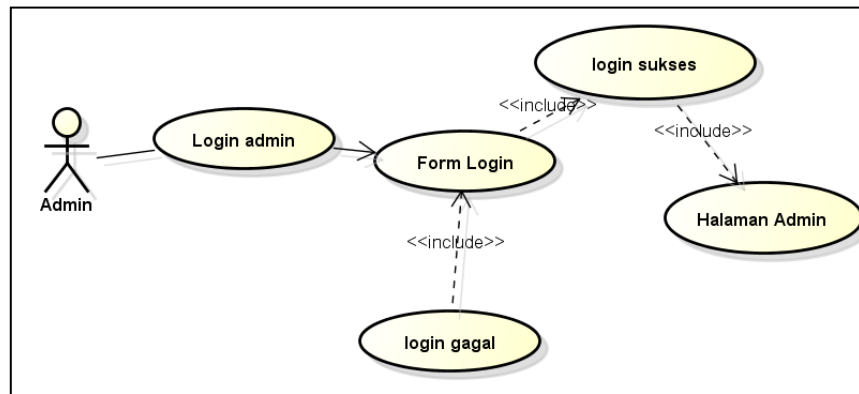
Tabel 1. Data sintax SQL Injeksi yang umum digunakan *hacker* untuk membobol *website* secara paksa.

No	Sintax SQL Injeksi pada <i>username</i>	Sintax SQL Injeksi pada <i>password</i>
1	' or 1=10 limit 10'	' or 100=15 limit 10'
2	''' OR 1=1	or '1'='1
3	admin	' or '1'='1
4	''' OR 1=1	' or 'x'='x
5	''' OR 1=1	' or 0=0 --
6	''' OR 1=1	" or 0=0 --
7	''' OR 1=1	or 0=0 --
8	''' OR 1=1	' or 0=0 #
9	''' OR 1=1	" or 0=0 #
10	''' OR 1=1	or 0=0 #
11	''' OR 1=1	' or 'x'='x
12	''' OR 1=1	" or "x"="x
13	''' OR 1=1	) or ('x'='x
14	''' OR 1=1	' or 1=1--
15	admin	1') and '1'='1--

Pada tabel 1. diatas merupakan sintax yang biasanya digunakan oleh para hacker untuk melakukan peretasan *website* dengan metode SQL Injeksi menarget form login.

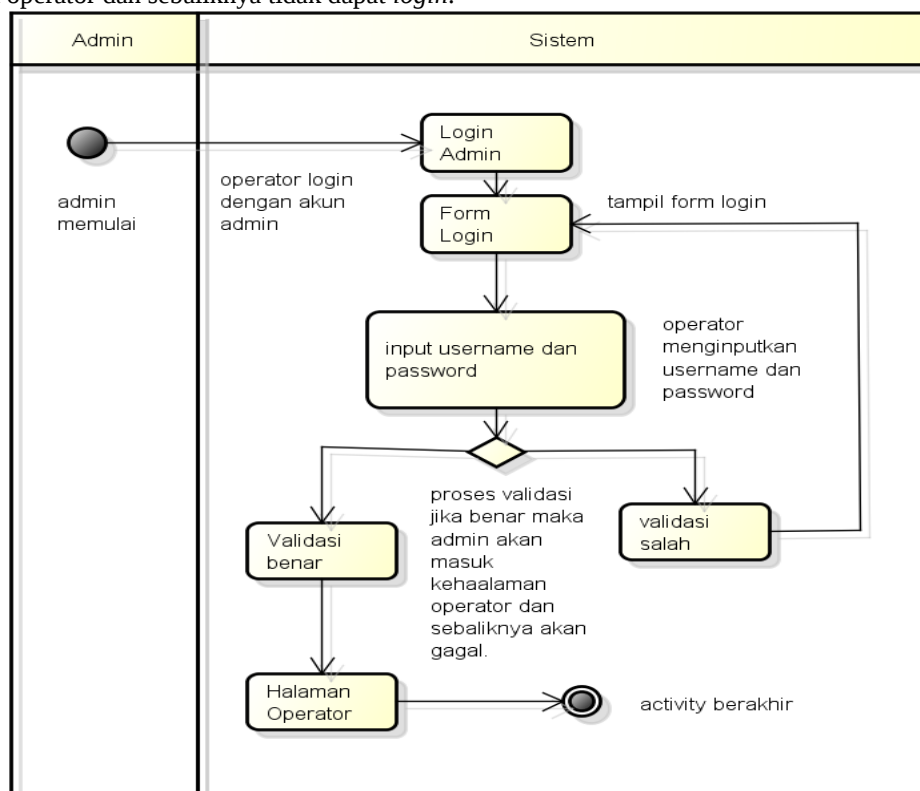
3.1 Analisis dan Desain Sistem

Untuk menjelaskan alur dan konsep dari penelitian, penulisan menggunakan *Unified Modeling Language* (UML) untuk memodelkan alur penelitian yang dilakukan. UML digambarkan mulai dari *use case* dan *activity diagram*. UML dari *system website* dapat dilihat pada gambar.



Gambar 2. Use Case Login Admin

pada gambar 2 *use case* terdapat skenario admin yang melakukan aktivitas *login* admin dengan memasukkan *username* serta *password*. Jika validasi benar maka admin akan dapat *login* dan masuk kehalaman operator dan sebaliknya tidak dapat *login*.

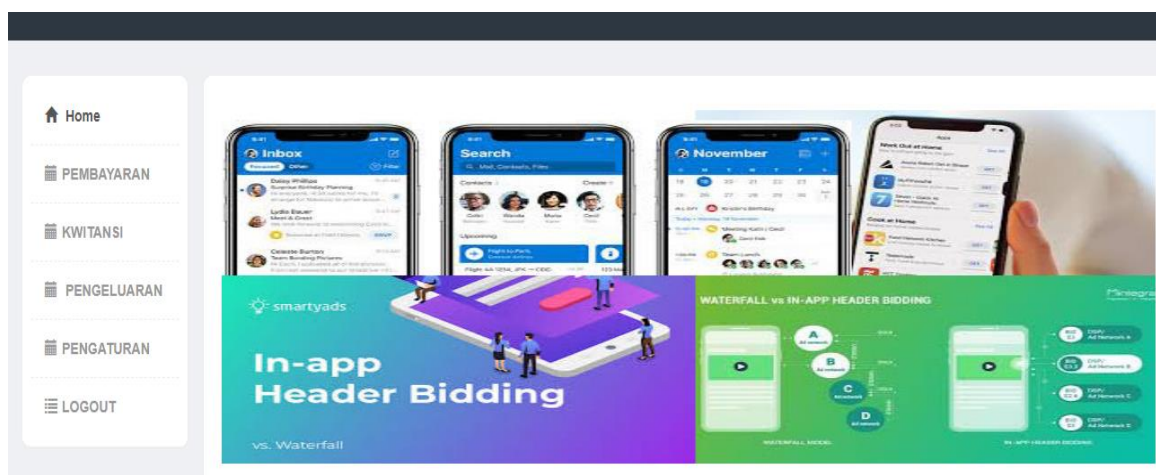


Gambar 3. Activity Diagram

Pada *activity diagram* merupakan aktivitas admin untuk masuk kedalam halaman operator dengan memasukkan akun admin yang resmi secara normal. Jika akun admin benar maka *system* akan melakukan validasi untuk memproses masuk kedalam halaman admin jika akun admin tidak resmi maka akan gagal *login* kecuali melakukan prosedur diluar validasi seperti aktivitas *hacking* dengan cara illegal.

3.2 Implementasi

Pada bagian ini akan dijelaskan tentang implemntasi sebuah *website* publikasi RPS atau disebut *Website* menampilkan informasi Rencana pembelajaran semester yang terdapat menu *home* untuk menampilkan informasi beranda, menu daftar mata kuliah untuk melihat daftar semua mata kuliah yang ada dan menu *login* untuk admin melakukan *login* ke halaman operator.



Gambar 4. Tampilan website

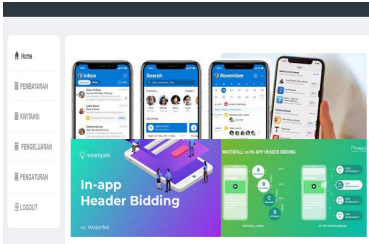
3.3 Implementasi SQL Injeksi

Pada bagian ini akan dijelaskan proses pembobolan *website* dari *login* admin dengan cara SQL Injeksi dengan dua model yaitu SQL Injeksi pada *website* tanpa menggunakan teknik penggunaan *maxlength* dan *input type* number dan *website* yang menggunakan teknik penggunaan *maxlength* dan *input type* number. Untuk *detail* penjelasan dapat dilihat pada pembahasan berikut ini.

a. *System website* tanpa penggunaan teknik *maxlength* dan *input type* number

Pada bagian ini akan ditampilkan *source code* login.php dan tampilan *form login* dengan pembahasan sebagai berikut :

Tabel 2. Tampilan dan *source code* login.php tanpa teknik *maxlength* dan *input type* number

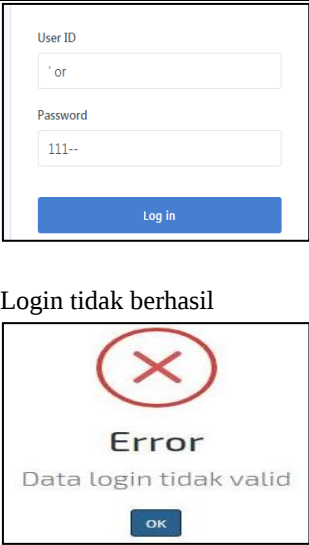
Gambar form login	Source code form login
 Hasil dapat login ke admin 	<pre> <div class="form-group"> <label class="form-label">User ID</label> <input type="text" class="form-control" name="userid" placeholder="Enter UserID" required> </div> <div class="form-group"> <label class="form-label"> Password </label> <input type="password" class="form-control" name="password" placeholder="Password" required> </div> </pre>

Pada tabel 2 form *login* contoh A dapat kita lihat pada bagian *source code* warna merah tidak terdapat *maxlength* untuk pembatasan inputan *username* atau *userid* sehingga *hacker* dapat menginputkan *sintax* SQL Injeksi di inputan *username* tersebut secara bebas. Kemudian lihat *source code* yang berwarna hijau juga tidak menggunakan input *type* number tetapi memakai *type password* yang artinya pada inputan *password* dengan *type password* ini tidak efektif mencegah serangan SQL Injeksi karena *hacker* bisa bebas memasukkan *sintax* Injeksi dengan secara paksa apalagi inputan tidak dibatasi dengan *maxlength*. Untuk melihat perbandingan lainnya dapat dilihat pada bagian B berikut.

b. *System website* penggunaan teknik *maxlength* dan input *type* number

Pada bagian ini akan ditampilkan *source code* login.php dan tampilan form login dengan pembahasan sebagai berikut:

Tabel 3. Tampilan dan *source code* login.php menggunakan teknik *maxlength* dan input *type* number

Gambar form login	Source code form login
	<pre> <div class="form-group"> <label class="form-label">User ID</label> <input type="text" class="form- control" name="userid" maxlength="5" placeholder="Enter UserID" required> </div> <div class="form-group"> <label class="form-label"> Password </label> <input type="number" class="form- control" name="password" maxlength="10" placeholder="Password" required> </div> </pre>

Pada tabel 3 diatas merupakan tampilan *form login* dan *source code* merupakan penggunaan teknik *maxlength* dan input *type* number. Dapat dilihat perbedaanya dimana saat *hacker* menggunakan *sintax* SQL Injeksi pakai *username* menggunakan (' or 1=1 limit 1 -- -+) dengan karakter *text* 21 maka tidak berhasil karena tidak lengkap diakibatkan karena sudah dibatasi *maxlength* menjadi 5 karakter saja sehingga tampilan pada *form login* inputan *username* hanya dapat menampung teks ('or) saja. Kemudian lihat pada bagian inputan *password* sebab sudah di ubah *input type* menjadi angka atau number maka *sintax* SQL Injeksi yang seharusnya (' or 1=1 limit 1 -- -+) berubah menjadi (111--) yang artinya *sintax* itu berubah mengikuti format angka dan panjang arakter inputan hanya menjadi 5 karakter saja hal ini membuat tidak befungsinya proses pembobolan *website* dengan teknik penetrasi Paksa SQL Injeksi.

3.3 Pengujian Sistem

Pada penelitian ini digunakan metode pengujian *black box testing*. Pengujian dilakukan dengan menguji bagian antarmuka dari sistem informasi, setiap bagian dari antarmuka tersebut diuji agar dapat ditentukan apakah sistem sudah berjalan sesuai dengan fungsi yang diharapkan [8]. Tujuan dari pengujian ini adalah untuk mengetahui kesalahan pada sistem yang dibuat. Berikut adalah hasil pengujian menggunakan metode *black box testing* yang ditampilkan dalam bentuk tabel:

Tabel 4. Hasil pengujian sistem

No	Kelas Uji	Skenario Pengujian	Hasil Pengujian
1	Halaman <i>Home</i>	admin dan <i>user</i> dapat mengakses menu <i>home</i> sehingga menampilkan informasi halaman <i>home</i> pada beranda	Sesuai
2	Menu Daftar Mata kuliah	Halaman yang tampil adalah informasi daftar mata kuliah yang sudah ada dalam database.	Sesuai
3	Menu <i>Login</i>	Halaman ini menampilkan form login untuk admin masuk ke halaman operator.	Sesuai

4. SIMPULAN

Adapun kesimpulan yang terdapat dalam penelitian ini adalah sebagai berikut :

1. SQL Injeksi merupakan salah satu teknik dari sekian banyaknya cara untuk membobol *website* yang digunakan oleh *hacker*.
2. SQL Injeksi akan mudah dioperasikan pada *website* yang tidak menggunakan proteksi atau keamanan *website*.
3. SQL Injeksi dapat juga dicegah dengan cara menerapkan keamanan *website* melalui penggunaan captcha, OTP, anti injeksi dibagian *syntax query* database pada *source code* dan menerapkan validasi tertentu seperti membatasi jumlah inputan karakter serta mengubah jenis inputan pada *form login*.

DAFTAR PUSTAKA

- [1] S. Zulkifli, "Implementasi Sistem Keamanan SQL Injection Dalam berbasis web," *Ejurnal.Univalabuhanbatu.Ac.Id*, vol. 04, no. 01, pp. 13–17, 2020, [Online]. Available: <https://ejurnal.univalabuhanbatu.ac.id/index.php/u-net/article/download/164/130>.
- [2] U. Verawardina, F. Edi, and R. Watrionthos, "Analisis Sentimen Pembelajaran Daring Pada Twitter di Masa Pandemi COVID-19 Menggunakan Metode Naïve Bayes," vol. 5, pp. 157–163, 2021, doi: 10.30865/mib.v5i1.2604.
- [3] J. H. P. Sitorus *et al.*, "Perancangan pengontrol lampu rumah miniatur dengan menggunakan micro controler arduino berbasis android 1," vol. 4, no. 1, pp. 1–11, 2020.
- [4] Samsir, D. I. G. Hts, and S. Z. Harahap, "SPK Untuk Pemilihan Kepala Sekolah Menggunakan Metode Saw dan Profile Matching," *U-NET J. Tek. Inform.*, 2020.
- [5] M. V. B. Net, "PADA TOKO URIP MOTOR," no. September, pp. 1–6, 2020.
- [6] W. Fahrozi, P. T. Informatika, T. Informatika, F. U. A. Labuhanbatu, T. Mulia, and K. Medan, "U-NET : Jurnal Teknik Informatika LPPM – Universitas Al Washliyah Labuhanbatu 23 | P a g e U-NET : Jurnal Teknik Informatika Sebagai langkah awal yang dilakukan supaya dapat mengetahui gambaran permasalahan yang dihadapi dalam menentukan rasa yam serama a," vol. 3, no. 5, pp. 23–27, 2019.
- [7] Samsir and Syaiful Zuhri Harahap, "Application Design Resume Medical By Using Microsoft Visual Basic. Net 2010 At the Health Center Appointments," *Int. J. Sci. Technol. Manag.*, vol. 1, no. 1, pp. 14–20, 2020, doi: 10.46729/ijstm.v1i1.5.
- [8] M. Siddik and S. Samsir, "Rancang Bangun Sistem Informasi Pos (Point of Sale) Untuk Kasir Menggunakan Konsep Bahasa Pemrograman Orientasi Objek," *JOISIE (Journal Inf. Syst. Informatics Eng.*, vol. 4, no. 1, p. 43, 2020, doi: 10.35145/joisie.v4i1.607.
- [9] P. T. Informatika and F. U. A. Labuhanbatu, "U-NET : Jurnal Teknik Informatika LPPM – Universitas Al Washliyah Labuhanbatu 18 | P a g e U-NET : Jurnal Teknik Informatika Sebagai langkah awal yang dilakukan supaya dapat mengetahui gambaran permasalahan yang dihadapi oleh bagian kesiswaan adalah denga," vol. 3, no. 4, pp. 18–22, 2019.
- [10] Samsir, "Klasifikasi Penyakit Tenggorokan Hidung Telinga (THT) Menggunakan Jaringan Syaraf Tiruan Dengan Metode Learning Vektor Quantization (THT) Di RSUD Rantauprpat Labuhanbatu Klasifikasi penyakit Tenggorokan Hidung Telinga (THT) Menggunakan," vol. 05, no. 01, pp. 38–47, 2019.
- [11] R. A. Purba, S. Samsir, M. Siddik, S. Sondang, and M. F. Nasir, "The optimalization of

-
- backpropagation neural networks to simplify decision making,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 830, no. 2, 2020, doi: 10.1088/1757-899X/830/2/022091.
- [12] S. Samsir, S. Suparno, and M. Giatman, “Predicting the loan risk towards new customer applying data mining using nearest neighbor algorithm,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 830, no. 3, 2020, doi: 10.1088/1757-899X/830/3/032004.
- [13] D. I. G. H. Wirhan Fahrozi, Samsir, “Penerapan E-Commerce Pada Toko Bunga Underwear,” *J. Tek. Inform.*, vol. 04, no. 01, pp. 1–6, 2020.
- [14] F. Edi, P. T. Informatika, and F. U. A. Labuhanbatu, “UNET | Jurnal Ilmiah Teknik Informatika LPPM Universitas Al Washliyah Labuhanbatu UNET | Jurnal Ilmiah Teknik Informatika ISSN . 2460-3694 , Vol . 2 No . 1 Februari 2018,” vol. 2, no. 1, pp. 2–5, 2018.