



MATAKULIAH

0955531102

MID-TEST

KELAS TI-C – NIM.M3118051

NAMA : Luthfi Puji Ningtyas

Hari/Tanggal :

Senin / 18.10.2020

Ruang :

Dirumah

Dosen : Fiddin Yusfida A'la,
S.T., M.Eng.

A. DASAR TEORI

Login bisa juga di sebut “logon” atau “sign in” adalah istilah untuk keamanan computer yakni pintu masuk atau halaman utama bagi pengguna untuk mengakses system computer. Kegunaan login untuk mengatur proses identifikasi. Saat pengguna sudah masuk ke web atau aplikasi yang di akses, token masuk dapat digunakan untuk melacak tindakan apa yang telah dilakukan pengguna saat terhubung ke situs tersebut. Dalam sebuah layanan website atau aplikasi, biasanya terdapat hak akses atau role saat melakukan login.

Role atau hak ases merupakan key field yang ada saat menambahkan pengguna baru, dengan kata lain role ini dapat mengatur pengguna baru agar tidak dapat mengakses dan mengubah pengaturan data rahasia dari website. Suatu login dan register untuk menjaga system keamanannya di perlukan proses penyandian enkripsi dan dekripsi seperti saat memasukkan password.

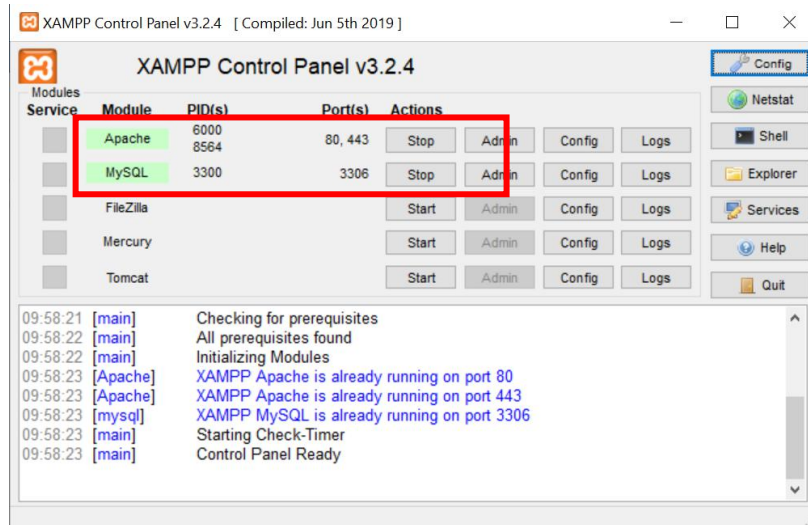
Suatu proses penyandian terhadap suatu data disebut enkripsi dan dekripsi. Enkripsi adalah proses pengubahan data asli (plaintext) menjadi data rahasia (ciphertext) pada saat pengiriman (sending) sehingga kerahasiaan data terjaga. Sedangkan dekripsi adalah proses pengubahan data rahasia (ciphertext) menjadi data asli (plaintext) pada saat penerimaan (receiving) sehingga data sesuai dengan data asli yang dikirimkan. Dengan proses ini, maka selama proses pengiriman data, data bersifat rahasia sehingga keamanannya dapat terjaga.

Caesar chipper merupakan algoritma kriptografi klasik yang sebelumnya digunakan oleh Julius Caesar untuk mengirimkan sebuah pesan rahasia. Caesar chipper bisa di artikan sebagai sandi geser karena huruf-huruf dalam plainteks digantikan oleh huruf lainnya dalam posisi tertentu dalam susunan alphabet. Dibawah ini contoh Caesar chipper menggeser 1 huruf :

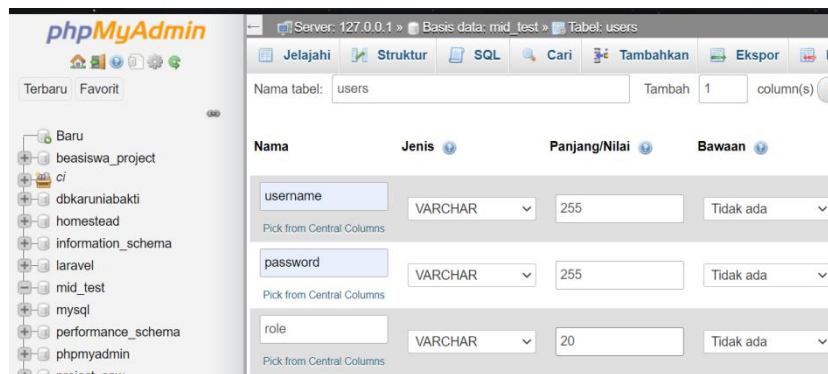
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A

B. LANGKAH-LANGKAH PRAKTIKUM

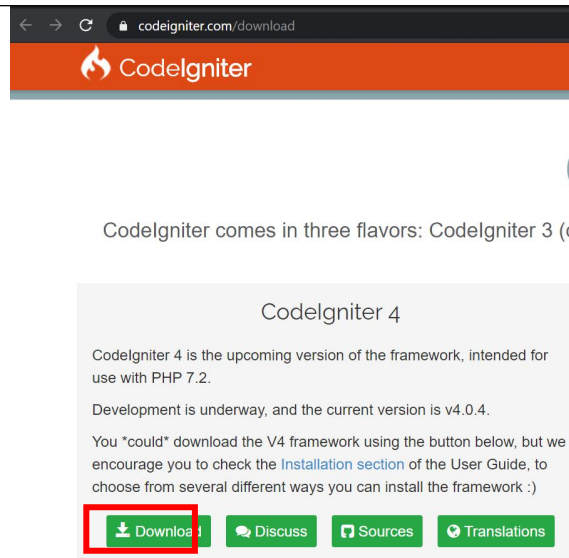
1. Jalankan server XAMPP dan buka aplikasi code editor, disini saya menggunakan Sublime Text



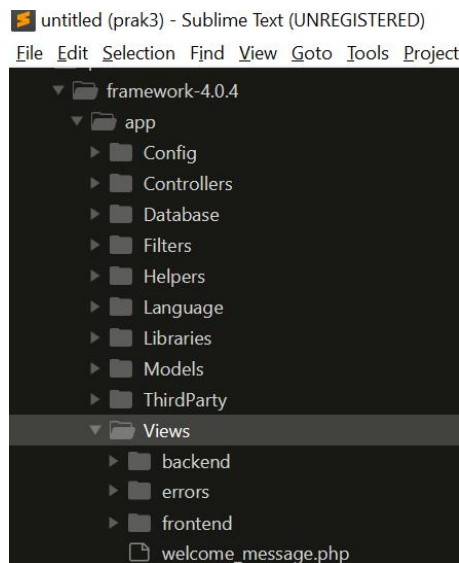
2. Selanjutnya masuk ke php my admin dan buat database bernama “mid-test”, lalu buat table bernama “users”. Jika sudah isikan field seperti dibawah ini



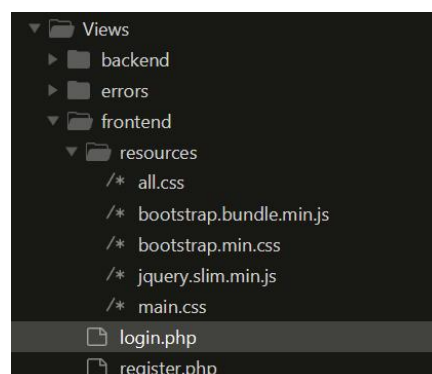
3. Selanjutnya bisa download Codeigniter dengan cara klik url ini <https://codeigniter.com/download> dan pilih “Download”. Taruh folder CI tersebut ke Htdocs dan di extract dengan cara klik kanan pilih extract here.



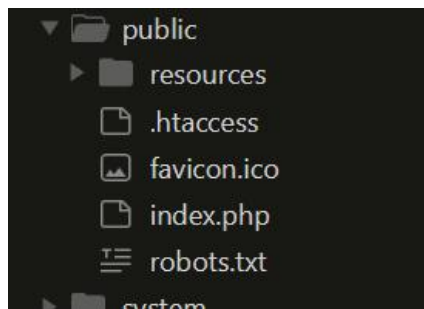
4. Selanjutnya download template login (<https://drive.google.com/file/d/1oytA1SfvmjF7-t8Jq2s0TSxoNLHBs5z/view>) dan masukkan ke htdocs lalu extract. Jika sudah kita buat folder bernama “frontend” dan “backend”.



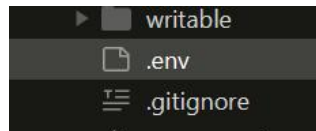
5. Selanjutnya masukkan “login” dan “register” ke dalam frontend dan rename menjadi .php



6. Selanjutnya copy folder resources yang sudah di download tadi dan paste kan ke folder public



7. Buka folder projectnya dan tambahkan titik pada file “env” menjadi “.env”



8. Selanjutnya buka file “.env” lalu ubah CI_ENVIRONMENT = dari production menjadi development dan hentikan tanda “#”.

```
17 CI_ENVIRONMENT = development
18
```

9. Setelah itu jalankan cmd/terminal dengan cara masuk ke file htdocsnya dan ketik perintah “php spark serve”.

```
D:\xampp\htdocs\skd\prak3\project_login>php spark serve

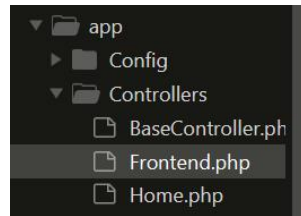
CodeIgniter CLI Tool - Version 4.0.4 - Server-Time: 2020-09-17 11:13:40am

CodeIgniter development server started on http://localhost:8080
Press Control-C to stop.
[Thu Sep 17 23:13:42 2020] PHP 7.4.7 Development Server (http://localhost:8080) started
```

10. Selanjutnya setting database, App -> Config -> Database.php dan sesuaikan dengan database Php MyAdmin yang dimiliki.

```
32 public $default = [
33     'DSN' => '',
34     'hostname' => 'localhost',
35     'username' => 'root',
36     'password' => '',
37     'database' => 'mid_test',
38     'DBDriver' => 'MySQLi',
39     'DBPrefix' => '',
40     'pConnect' => false,
41     'DBDebug' => (ENVIRONMENT !== 'production'),
42     'cacheOn' => false,
43     'cacheDir' => '',
44     'charset' => 'utf8',
45     'DBCollat' => 'utf8_general_ci',
46     'swapPre' => '',
47     'encrypt' => false,
48     'compress' => false,
49     'strictOn' => false,
50     'failover' => [],
51     'port' => 3306,
52 ];
53
```

11. Selanjutnya buat file baru bernama “Frontend.php” di dalam folder App -> Controller.



12. Selanjutnya buka app-config-routes.php dan tambahkan route untuk mengakses login

```
Auth.php x UserModel.php x Routes.php x
1 <?php namespace Config;
2
3 // Create a new instance of our RouteCollection class.
4 $routes = Services::routes();
5
6 // Load the system's routing file first, so that the app an
7 // can override as needed.
8 if (file_exists(SYSTEMPATH . 'Config/Routes.php'))
9 {
10     require SYSTEMPATH . 'Config/Routes.php';
11 }
12
13 /**
14  * -----
15  * Router Setup
16  * -----
17  */
18 $routes->setDefaultNamespace('App\Controllers');
19 $routes->get('/', 'Frontend::index');
20 $routes->get('/register', 'Frontend::register');
21 $routes->setDefaultController('Home');
22 $routes->setDefaultMethod('index');
23 $routes->setTranslateURIDashes(false);
24 $routes->set404Override();
25 $routes->setAutoRoute(true);
26
27 /**
28  * -----
```

Routes ini akan mengenali bahwa terdapat controller dengan nama frontend dan mempunyai function index, sehingga apabila dijalankan akan mengarah ke frontend.

13. Selanjutnya masuk ke app-controllers-Frontend.php dan tambahkan function register.

```
Frontend.php x Auth.php x login.php x
1 <?php
2
3 namespace App\Controllers;
4
5 use CodeIgniter\Controller;
6
7 class Frontend extends Controller
8 {
9     public function index()
10     {
11         return view('frontend/login');
12     }
13
14     public function register()
15     {
16         session();
17         $data = [
18             'validate' => \Config\Services::validation(),
19         ];
20         return view('frontend/register', $data);
21     }
22 }
23
```

Controller ini yang pertama kalinya akan di akses user/rute pertama yaitu index, diindex akan diarahkan ke view file frontend -> login, dan nantinya akan menampilkan halaman login begitu juga dengan register.

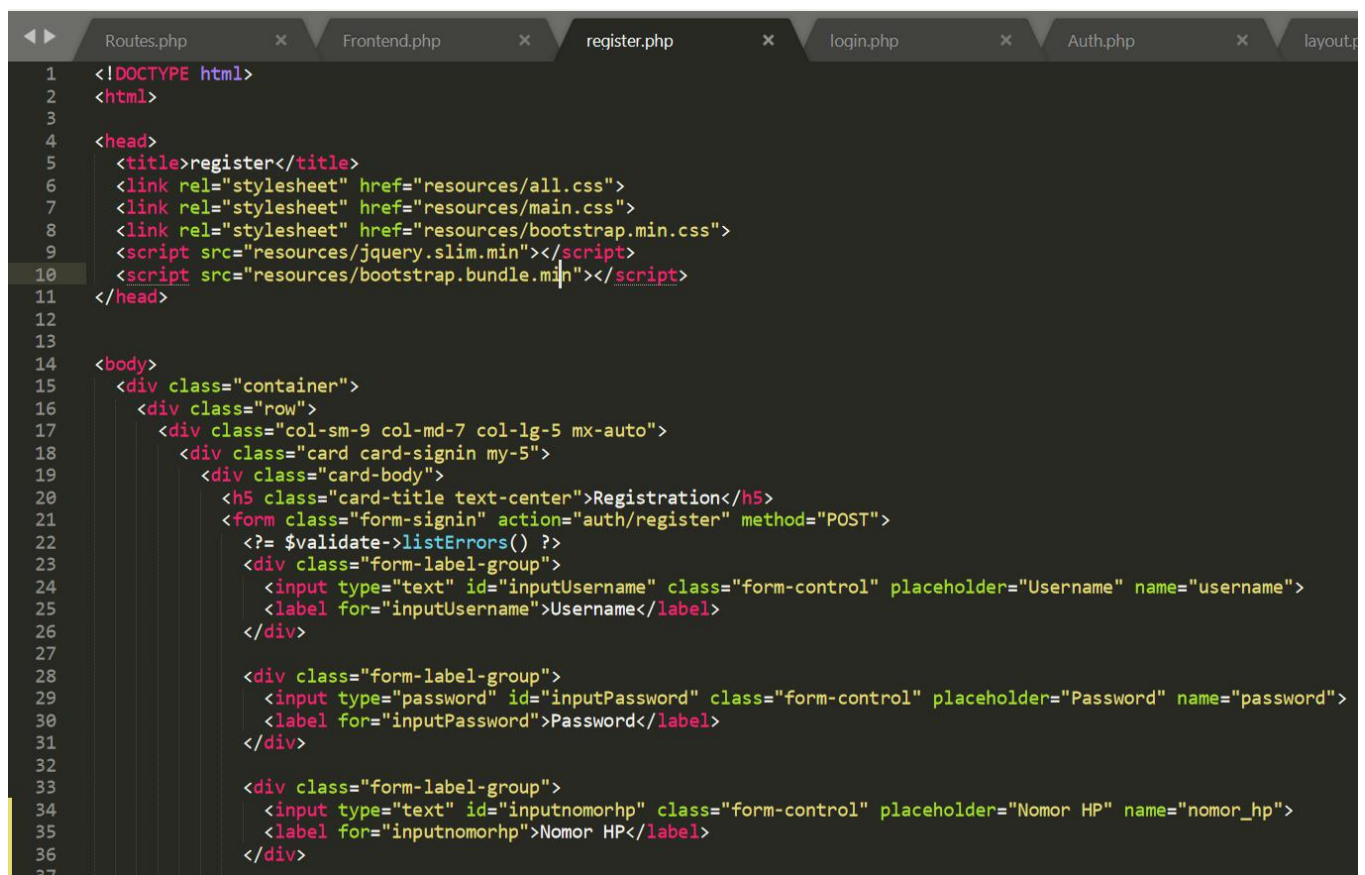
Fungsi ini nanti akan mengubah tampilan yang sebelumnya default nya codeigniter menjadi halaman login. Function register ini nanti berfungsi untuk penambahan data ketika user belum mempunyai akun untuk masuk ke halaman webnya.

14. Selanjutnya daftarkan routes register ke app-config-routes.php

```
$routes->setDefaultNamespace('App\Controllers');
$routes->get('/', 'Frontend::index');
$routes->get('/register', 'Frontend::register');
$routes->setDefaultController('Home');
$routes->setDefaultMethod('index');
$routes->setTranslateURIDashes(false);
$routes->set404Override();
$routes->setAutoRoute(true);
```

- Jika routes mengakses method get, lalu '/' => rute index/awal masuk ke class Frontend function index.
- Yang kedua, Jika routes akses method get => '/register' masuk ke rute awal lalu file register dan mencari class frontend (frontend.php) dan mengecek d bagian function register.

15. Selanjutnya masuk ke view register.php untuk mendaftar akun agar bisa login.



```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>register</title>
6   <link rel="stylesheet" href="resources/all.css">
7   <link rel="stylesheet" href="resources/main.css">
8   <link rel="stylesheet" href="resources/bootstrap.min.css">
9   <script src="resources/jquery.slim.min"></script>
10  <script src="resources/bootstrap.bundle.min"></script>
11 </head>
12
13
14 <body>
15   <div class="container">
16     <div class="row">
17       <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">
18         <div class="card card-signin my-5">
19           <div class="card-body">
20             <h5 class="card-title text-center">Registration</h5>
21             <form class="form-signin" action="auth/register" method="POST">
22               <?=$validate->listErrors() ?>
23               <div class="form-label-group">
24                 <input type="text" id="inputUsername" class="form-control" placeholder="Username" name="username">
25                 <label for="inputUsername">Username</label>
26               </div>
27
28               <div class="form-label-group">
29                 <input type="password" id="inputPassword" class="form-control" placeholder="Password" name="password">
30                 <label for="inputPassword">Password</label>
31               </div>
32
33               <div class="form-label-group">
34                 <input type="text" id="inputnomorhp" class="form-control" placeholder="Nomor HP" name="nomor_hp">
35                 <label for="inputnomorhp">Nomor HP</label>
36               </div>
37             </form>

```

```

37
38     <div class="form-group">
39         <input type="text" class="form-control form-control-user" name="alamat" id="alamat"
40             aria-describedby="alamat" placeholder="Masukkan Alamat...">
41     </div>
42
43     <div class="form-label-group">
44         <select class="form-control" name="role">
45             <option>Admin</option>
46             <option>Staff</option>
47         </select>
48     </div>
49 <div class="form-group mt-4 mb-0">
50     <button class="btn btn-primary btn-block" type="submit">Create Account</button>
51 </div>
52 <div class="form-group mt-4 mb-0 align="text-center">
53     <a href="/" type="submit" >Have an account? Go to login</a>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </body>
61 </html>
62

```

Jadi, jika user mengakses rute ke index lalu masuk ke register akan menampilkan fom pendaftaran.

- user mengisi form, jika sudah “create account” dengan type submit yang nanti akan diarahkan ke action “auth/register” (file auth function register).

```

Auth.php
Routes.php
1  <?php
2
3  namespace App\Controllers;
4
5  use CodeIgniter\Controller;
6  use App\Models\UserModel;
7  use App\Models\AuthModel;
8
9

```

Di auth.php masukkan models dan controller codeigniter.

```

Auth.php x AuthModel.php x UserModel.php x Routes.php x Fro
1 <?php
2
3 namespace App\Controllers;
4
5 use CodeIgniter\Controller;
6 use App\Models\UserModel;
7 use App\Models\AuthModel;
8
9
10 class Auth extends Controller
11 {
12     public function register()
13     { //masuk validasi sesuai dtbase
14         $val = $this->validate(
15             [
16                 'username' => 'required',
17                 'password' => 'required',
18                 'nomor_hp' => 'required',
19                 'alamat' => 'required',
20             ]
21         );
22         //validasi berhasil redirect ke halaman register
23         var_dump($val);
24         if (!$val) {
25             $pesanvalidasi = \Config\Services::validation();
26             return redirect()->to('/register')->withInput()->with(
27                 'validate',
28                 $pesanvalidasi
29             );
30         }
31     }

```

- variabel \$val digunakan untuk validasi data dari database jika validasi berhasil atau sama dengan database maka validasi dengan database berhasil, jika username, password, nomor_hp dan alamat tidak sama dengan database akan di redirect ke halaman register untuk melakukan register kembali.

```

32 //proses enkripsi, ngambil password dari database
33 //proses dekrip (halaman regis) jadi enkrip(database)
34 $kalimat = $this->request->getPost('password');
35 $key = 1 ; //NIM kaliann
36 for ($i = 0; $i < strlen($kalimat); $i++) {
37     $kode[$i] = ord($kalimat[$i]); //rubah ASCII ke desimal
38     $b[$i] = ($kode[$i] + $key) % 256; //proses enkripsi
39     $c[$i] = chr($b[$i]); //rubah desimal ke ASCII, desimal diubah ke huruf
40 }
41
42 for ($i = 0; $i < strlen($kalimat); $i++) {
43     echo $kalimat[$i];
44 }
45 $hsl = '';
46 for ($i = 0; $i < strlen($kalimat); $i++) {
47     echo $c[$i];
48     $hsl = $hsl . $c[$i];
49 }
50 //hasil proses enkripsi
51 $password = $hsl;
52

```

- Jika sudah, masuk untuk proses enkripsi. Variabel \$kalimat akan merequest password yang ada di database, lalu menggunakan key =1 (NIM sesuai angka terakhir), selanjutnya masuk ke perulangan dengan indeks ke 0 dan pengecekan password dimulai dari awal.

- password huruf pertama akan di cek dan dihitung menggunakan kode ascii + 1 (\$key) modulo 256 menggunakan variabel \$b[\$i] selanjutnya hasil akan di tampung menggunakan variabel \$c[\$i] sampai index terakhir atau sampai habis password yang diambil dari database.


```

53 //request ke database, jika berhasil nampilin session terus redirect ke login
54 $data = array(
55     'username' => $this->request->getPost('username'),
56     'password' => $hsl,
57     'nomor_hp' => $this->request->getPost('nomor_hp'),
58     'alamat' => $this->request->getPost('alamat'),
59     'role' => $this->request->getPost('role')
60 );
61 $model = new UserModel();
62 $model->insert($data);
63 session()->setFlashdata('pesan', 'selamat anda berhasil registrasi');
64 return redirect()->to('/');
65 }

```

Jika sudah, hasil dari \$hsl disimpan menggunakan variabel \$password. Lalu data tadi disimpan dalam bentuk array menggunakan variabel \$data, selanjutnya deklarasi variabel baru untuk memanggil class UserModel yang berisi nama field database Users untuk di tambahkan ke database dan jika berhasil muncul session flashdata pesan “selamat anda berhasil registrasi dan akan di redirect ke halaman index/awal untuk melakukan login.

- Selanjutnya “App-Models-UserModel”. UserModel ini untuk memanggil tabel sesuai field yang ada didatabase

```

1 <?php
2 namespace App\Models;
3 use CodeIgniter\Model;
4
5 class UserModel extends Model
6 {
7     //nampung data sesuai tabel dan kolom database
8     protected $table = 'users';
9     protected $allowedFields = ['username', 'password', 'nomor_hp', 'alamat', 'role'];
10 }
11
12 ?>

```

- Selanjutnya kita masuk ke view login.php untuk membuat halaman login

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <title>login</title>
6     <link rel="stylesheet" href="resources/all.css">
7     <link rel="stylesheet" href="resources/main.css">
8     <link rel="stylesheet" href="resources/bootstrap.min.css">
9     <script src="resources/jquery.slim.min"></script>
10    <script src="resources/bootstrap.bundle.min"></script>
11 </head>
12
13
14 <body>
15     <div class="container">
16         <div class="row">
17             <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">
18                 <div class="card card-signin my-5">
19                     <div class="card-body">
20                         <h5 class="card-title text-center">Log In</h5>
21                         <?= session()->get('pesan') ?>
22                         <form class="form-signin" method="POST" action="/auth/login">
23                             <div class="form-group">
24
25                                 <div class="form-group">
26                                     <input type="text" name="username" id="inputUsername" class="form-control" placeholder="Username">
27                                     <label for="inputUsername">Username</label>
28                                 </div>
29
30                                 <div class="form-group">
31                                     <input type="password" name="password" id="inputPassword" class="form-control" placeholder="Password">
32                                     <label for="inputPassword">Password</label>
33                                 </div>

```

```

34 <div class="form-group">
35 <div class="custom-control custom-checkbox">
36 <input class="custom-control-input" id="rememberPasswordCheck" type=
  "checkbox" />
37 <label class="custom-control-label" for="rememberPasswordCheck">
  Remember password</label>
38 </div>
39 </div>
40 <div class="form-group d-flex align-items-center justify-content-between
  mt-4 mb-0">
41 <a class="small" href="password.html">Forgot Password?</a>
42
43
44 <button class="btn btn-primary" type="submit">Sign In</button>
45 </div>
46
47 <form class="form-signin">
48 <hr class="my-4">
49 <div class="card-footer text-center">
50 <a href="/register" class="small" type="submit">Sign up</a>
51 </div>
52 </form>
53 </div>
54 </div>
55 </div>
56 </div>
57 </body>
58
59 </html>

```

Login.php ini berisi form login dan terdapat tombol sign in yang nantinya datanya akan di samakan dengan database dahulu jadi sebelum form login menggunakan method=POST dan action “/auth/login” yaitu ke file auth.php function login.

“App-Models-AuthModel.php”

```

1 <?php
2 namespace App\Models;
3 use CodeIgniter\Model;
4 class AuthModel extends Model
5 {
6     //cek $builder dimana username dari database mengambil data $username
7     function get_data_login($username, $tbl)
8     {
9         $builder = $this->db->table($tbl);
10        $builder->where('username', $username);
11        $log = $builder->get()->getRow();
12        return $log;
13    }
14 }
15 >?

```

Berikut adalah script dari auth.php (App-Controllers-Auth.php)

```

64 public function login()
65 {
66     $model = new AuthModel();
67     $table = 'users';
68     $username = $this->request->getPost('username');
69     $password = $this->request->getPost('password');
70     $row = $model->get_data_login($username, $table);
71     if ($row == NULL) {
72
73         session()->setFlashdata('pesan', 'username anda salah');
74         return redirect()->to('/');
75     }
76

```

- \$tabel = users untuk memanggil tabel users yang ada didatabase setelah itu di cocokkan data mulai dari username dan password dengan cara menyimpan data menggunakan variabel \$username dan \$password, jika sudah \$row untuk mendeklarasikan \$model dan menggenerate login dan mencocokkan \$tabel sesuai database.

- selanjutnya masuk ke percabangan, jika data yang diisikan kosong/NULL maka akan memanggil session setFlashData dan menampilkan pesan “username salah” dan akan mendirect ke halaman index atau awal.

Char	ASCII Code	Binary	Char	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110

```

$isi=$row->password;//enkr
$hs1 = '';
$key = 1; //nim kalian
for($i=0;$i<strlen($isi);$i++)
{
    $kode[$i]=ord($isi[$i]); // rubah ASII ke desimal
    $b[$i]=($kode[$i] - $key ) % 256; // proses dekripsi Caesar
    $c[$i]=chr($b[$i]); //rubah desimal ke ASCII-nilai desimal diubah ke huruf
    $hs1 = $hs1 . $c[$i];
}

```

- selanjutnya proses dekripsi menjadi enkripsi, alurnya :

- variabel \$isi ini memanggil \$row ->password yang ada didatabase, lalu deklarasi \$hs1 ini untuk menyimpan data baru, untuk \$key = 1 itu kuncinya karena NIM terakhir saya 1. contoh jika sata memasukkan password “aaa” dengan kunci 1 maka yang ada di database “bbb”.

- selanjutnya masuk ke perulangan, jadi \$i =0 itu nilai awal/baru dan mengambil data dari \$isi yang berisi password seperti database, lalu \$b ini mengambil \$kode[\$i] => b (enkripsi) => 98 (kode ascii) lalu dikurangi \$key yaitu 1 setelah itu di modulo 256 dan hasilnya di tampung dengan \$c[\$i], lalu \$hs1 akan merubah nilai desimal itu ke huruf.

```

88         if ($password==$hs1) {
89             $data = array(
90                 'log' => TRUE,
91                 'username' => $row->username,
92                 'role' => $row->role,
93             );
94             session()->set($data);
95             session()->setFlashdata('pesan', 'Berhasil login');
96             return redirect()->to('/backend');
97         }
98         else{
99             session()->setFlashdata('pesan', 'Password Salah');
100             return redirect()->to('/');
101         }
102     }
103 }

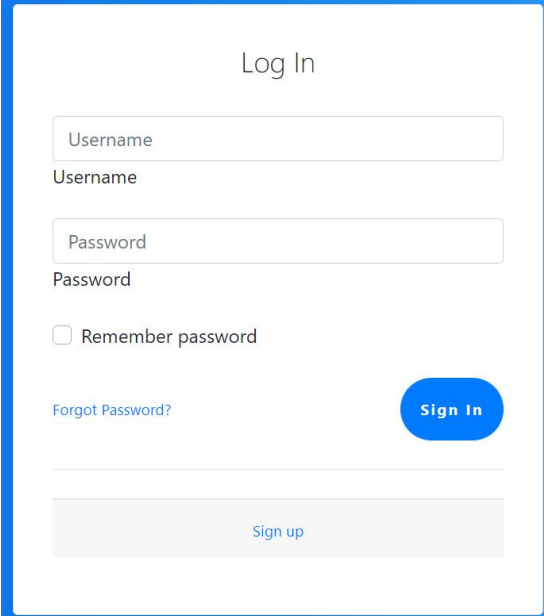
```

Jika sudah di dapatkan huruf enkripsinya, masuk ke pengecekan \$password == \$hs1, jika data username dan role benar sesuai database akan menampilkan halaman backend dan session flashdata “berhasil login” jika salah maka akan kembali ke halaman index/login.php dan menampilkan session

“password salah”.

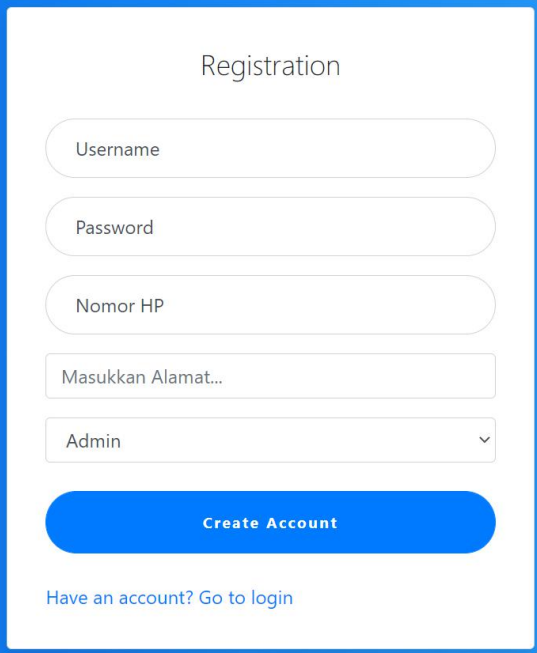
18. Hasil running :

* Halaman Login/utama <http://localhost:8080/>

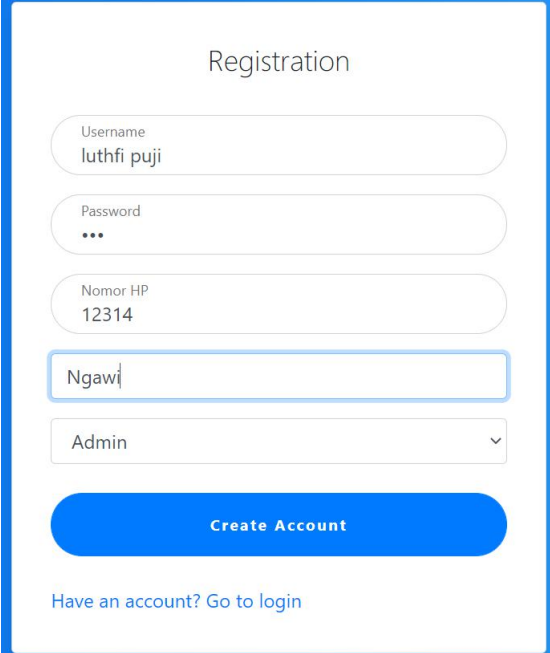


The screenshot shows a 'Log In' form with the following elements: a title 'Log In', a 'Username' input field with a label 'Username' below it, a 'Password' input field with a label 'Password' below it, a checkbox labeled 'Remember password', a link 'Forgot Password?', a blue 'Sign In' button, and a grey 'Sign up' button at the bottom.

* Halaman Register/Sign Up <http://localhost:8080/register>. mendaftar dengan username : luthfi puji dan password : aaa



The screenshot shows an empty 'Registration' form with the following elements: a title 'Registration', input fields for 'Username', 'Password', 'Nomor HP', and 'Masukkan Alamat...', a dropdown menu for 'Admin', a blue 'Create Account' button, and a link 'Have an account? Go to login'.



The screenshot shows the 'Registration' form filled with the following data: 'Username' is 'luthfi puji', 'Password' is masked with '...', 'Nomor HP' is '12314', 'Masukkan Alamat...' is 'Ngawi', 'Admin' is set to 'Admin', the 'Create Account' button is highlighted, and the link 'Have an account? Go to login' is visible.

Klik Create account dan masuk ke halaman login

Log In

selamat anda berhasil registrasi

luthfi puji

Username

...

Password

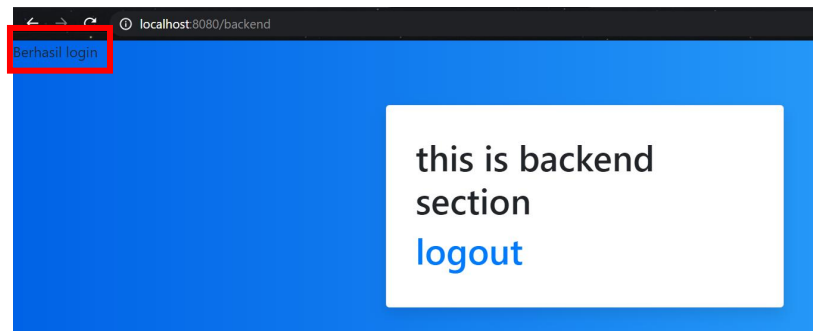
☐ Remember password

[Forgot Password?](#)

[Sign In](#)

[Sign up](#)

Session -> Flashdata



Jika username salah saat menginputkan

Log In

username anda salah

fifi

Username

...

Password

☐ Remember password

[Forgot Password?](#)

[Sign In](#)

[Sign up](#)

Jika password salah saat menginputkan

Log In

Password Salah

fifi

Username

...

Password

☐ Remember password

[Forgot Password?](#)

[Sign In](#)

[Sign up](#)

* Proses Dekripsi saat register Keterangan : Password : aaa

* Proses Enkripsi didatabase

Registration

Username

luthfi puji

Password

...

Nomor HP

12314

Ngawi

Admin

[Create Account](#)

[Have an account? Go to login](#)

+ Opsi

username	password	nomor_hp	alamat	role
www	bbb	42	fdsg	Admin
lll	VVV	5244	DSGEFG	Admin
ning	bbb	3423	fsd	Admin
qqq	bbb	32432	dfsgsf	Admin
fifi	bbb	45435	dfsg	Admin
puji	bbb	3534	dfsgff	Admin
nnn	bbb	2555	fdsg	Admin
luthfi puji	bbb	12314	Ngawi	Admin

C. KESIMPULAN

Dari hasil percobaan yang telah dilakukan dapat disimpulkan bahwa dengan adanya penyandian menggunakan enkripsi dan dekripsi selama proses pengiriman data, data bersifat rahasia sehingga keamanannya dapat terjaga dan penggunaan session berfungsi untuk melakukan aktivitas yang berhubungan dengan interaksi user pada sebuah web server php.

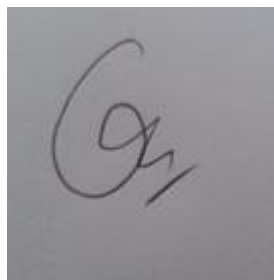
DAFTAR PUSTAKA

Hendro. (2013). *PENYANDIAN DALAM KRIPTOGRAFI*. Retrieved from informatika.stei.itb.ac.id:

<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2006-2007/Makalah/Makalah0607-74.pdf>

Widayan, S. N. (2015, November). *Algoritma Kriptografi Klasik: Caesar Chiper*. Retrieved from Catatan

Instrumatika: <https://www.catataninstrumatika.com/2015/11/algoritma-kriptografi-klasik-caesar.html>



(Luthfi Puji Ningtyas – M3118051)