



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00201947022, 22 Juli 2019

Pencipta

Nama : **Audhy Virabri Kressa, Pantjawati Sudarmaningtyas, Vivine Nurcahyawati**
Alamat : Perum Jatikalang A.8/41 RT 024 RW 005 Jatikalang Krian, Sidoarjo, Jawa Timur, 61262
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Institut Bisnis dan Informatika Stikom Surabaya**
Alamat : Jalan Raya Kedung Baruk 98, Surabaya, Jawa Timur, 60298
Kewarganegaraan : Indonesia
Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **Aplikasi Portal Beasiswa Berbasis Web Menggunakan Algoritma Matching Jaro-Winkler**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 25 Februari 2019, di Surabaya
Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan : 000147234

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001





APLIKASI PORTAL BEASISWA BERBASIS WEB MENGGUNAKAN ALGORITMA *MATCHING* JARO-WINKLER

PENCIPTA :

AUDHY VIRABRI KRESSA (15410100030)

PANTJAWATI SUDARMANINGTYAS (0712066801)

VIVINE NURCAHYAWATI (0723018101)

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2019

1. Requirement

Table 1 Requirement Instalation

No	Requirement
1	Composer
2	PHP >= 7.1.3
3	OpenSSL PHP Extension
4	PDO PHP Extension
5	Mbstring PHP Extension
6	Tokenizer PHP Extension
7	PostgreSQL

2. Install Composer (dapat dilihat pada [HTTPS://GETCOMPOSER.ORG/DOWNLOAD/](https://getcomposer.org/download/))

Table 2 Step Install Composer

No	Step
1	Buka cmd/terminal
2	<code>php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"</code>
3	<code>php -r "if (hash_file('SHA384', 'composer-setup.php') === '544e09ee996cdf60ece3804abc52599c22b1f40f4323403c44d44fdidd586475ca9813a858088ffbc1f233e9b180f061') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"</code>
4	<code>php composer-setup.php</code>
5	<code>php -r "unlink('composer-setup.php');"</code>

3. Deploy Portal Beasiswa

Table 3 Step Deploy Portal Beasiswa

No	Step
1	Ubah timeout php menjadi 90000 ms
2	Buka file .env pada folder portal beasisw. (Jika tidak ada silahkan rename file .env.example)
3	Atur nama database,username dan password pada file.env
4	Buka cmd/Terminal
5	Jalankan "crontab -e"
6	Isi dengan "* * * * * cd path_portalbeasiswa && php artisan schedule:run >> /dev/null 2>&1"
7	Save
8	Arahkan cmd/terminal pada folder aplikasi
9	Composer install
10	Php artisan key:generate
11	Php artisan migrate:refresh --seed

4. Pengaturan Jadwal Scaping dan Broadcast

Tabel 4 Step Pengaturan Jadwal

No	Step
1	Buka file kernel.php pada foldel app/console
2	Ubah jam pada function schedule
3	Save

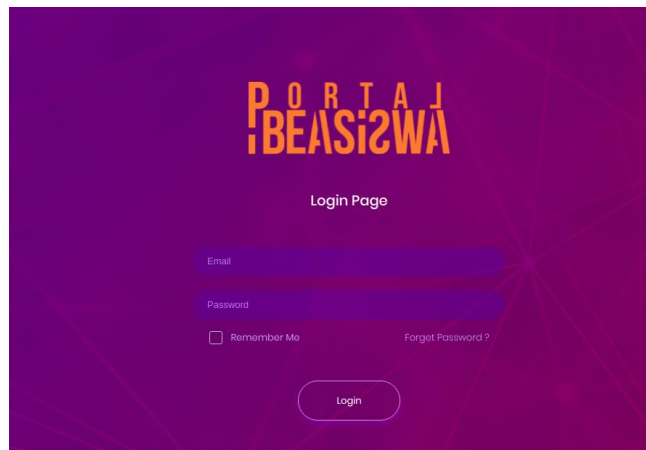
```
protected function schedule(Schedule $schedule)
{
    $schedule->call('App\Http\Controllers\ScaperController@google')
        ->dailyAt('00:10');
    $schedule->call('App\Http\Controllers\ScaperController@website')
        ->dailyAt('00:12');
    $schedule->call('App\Http\Controllers\SubscriberController@subscriberMail')
        ->dailyAt('00:14');
}
```

Gambar 1 Function Schedule

5. Menggunakan Portal Beasiswa Sebagai Administrator

5.1. Login

- 1) Masuk ke url login dengan menambahkan “/login” pada alamat portal beasiswa
- 2) Masukkan email dan password
- 3) Klik Login

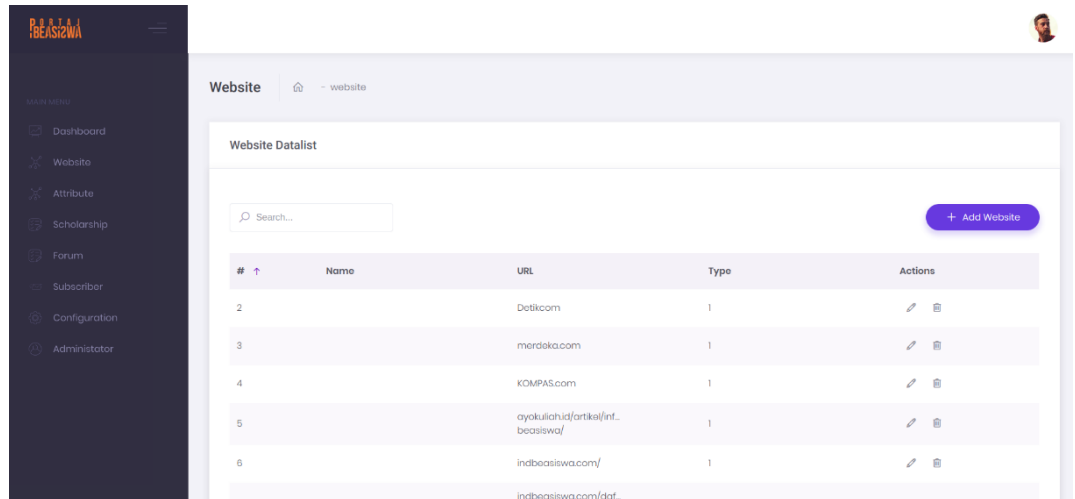


Gambar 2 Login Form

5.2. Mengelola Website

A. Menambah Website

- 1) Login
- 2) Pilih Website pada sidebar



Gambar 3 Tabel Website

- 3) Klik Add Website
- 4) Isi data Website pada form

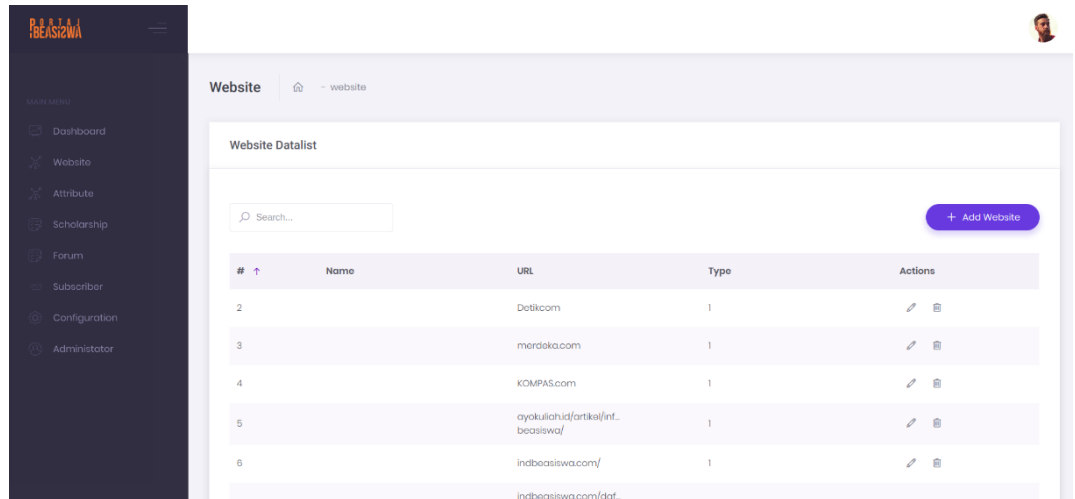
The screenshot shows a 'Website' form with fields for Name, Url, and Type. The Name field is required (indicated by a red asterisk) and has a placeholder text 'Name'. The Url field is also required and has a placeholder text 'Url'. The Type field has two radio button options: 'Portal' and 'Instansi'. At the bottom of the form, there are three buttons: 'Submit' (green), 'Reset' (yellow), and 'Close' (grey).

Gambar 4 Website Form

- 5) Klik Submit

B. Memperbarui Website

- 1) Login
- 2) Pilih Website pada sidebar

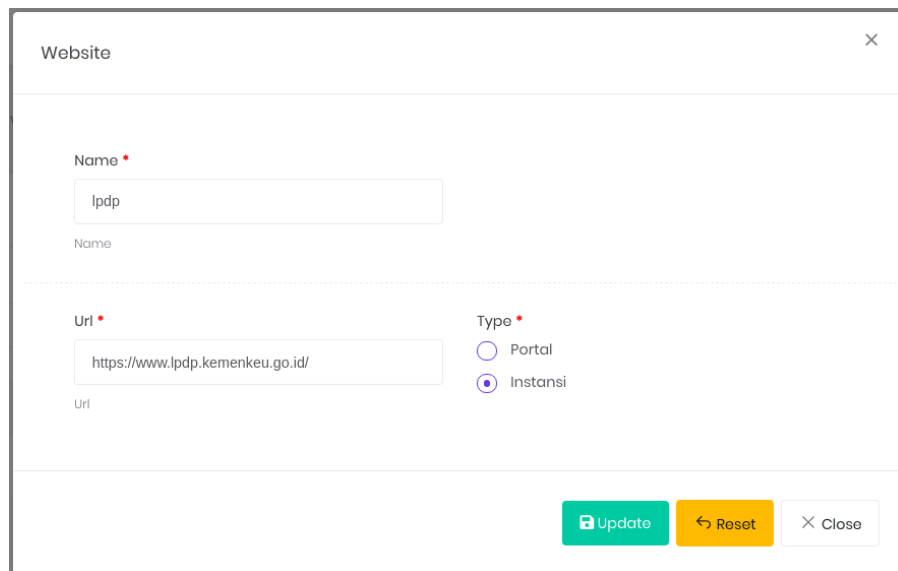


The screenshot shows the REASISWA dashboard with a sidebar menu on the left and a main content area. The sidebar menu includes: Dashboard, Website, Attribute, Scholarship, Forum, Subscriber, Configuration, and Administrator. The main content area is titled 'Website' and contains a 'Website Datalist' table. The table has columns for #, Name, URL, Type, and Actions. There are 6 rows of data, each with an edit icon and a delete icon in the Actions column. A '+ Add Website' button is located at the top right of the table.

#	Name	URL	Type	Actions
2		Detikcom	1	
3		merdeka.com	1	
4		KOMPAS.com	1	
5		ayukuliahid/artika/inf... beasiswa/	1	
6		indbeasiswa.com/	1	
		indbeasiswa.com/daf...		

Gambar 5 Tabel Website

- 3) Klik Ikon Edit pada data table Website
- 4) Isi data Website pada form edit



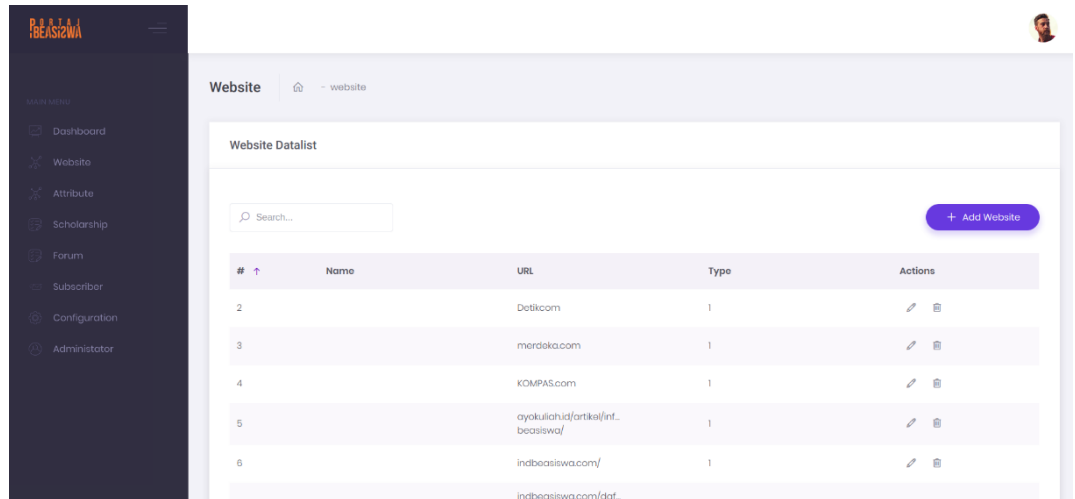
The screenshot shows the 'Website' edit form. It has a title bar with a close button (X). The form contains two main sections: 'Name' and 'Uri'. The 'Name' section has a text input field with the value 'lpdp'. The 'Uri' section has a text input field with the value 'https://www.lpdp.kemenkeu.go.id/'. To the right of the 'Uri' field is a 'Type' section with two radio buttons: 'Portal' and 'Instansi'. The 'Instansi' radio button is selected. At the bottom right of the form are three buttons: 'Update' (green), 'Reset' (yellow), and 'Close' (white with a grey border).

Gambar 6 Form Edit Website

- 5) Klik Update

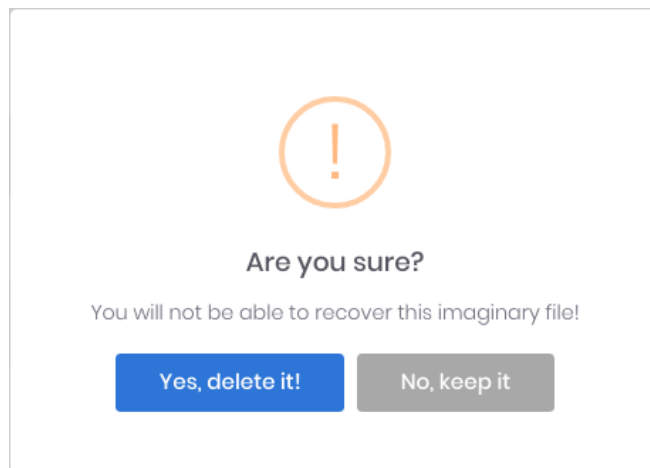
C. Menghapus Website

- 1) Login
- 2) Pilih Website pada sidebar



Gambar 7 Tabel Website

- 3) Klik Ikon Delete pada data table Website
- 4) Klik Yes, delete it

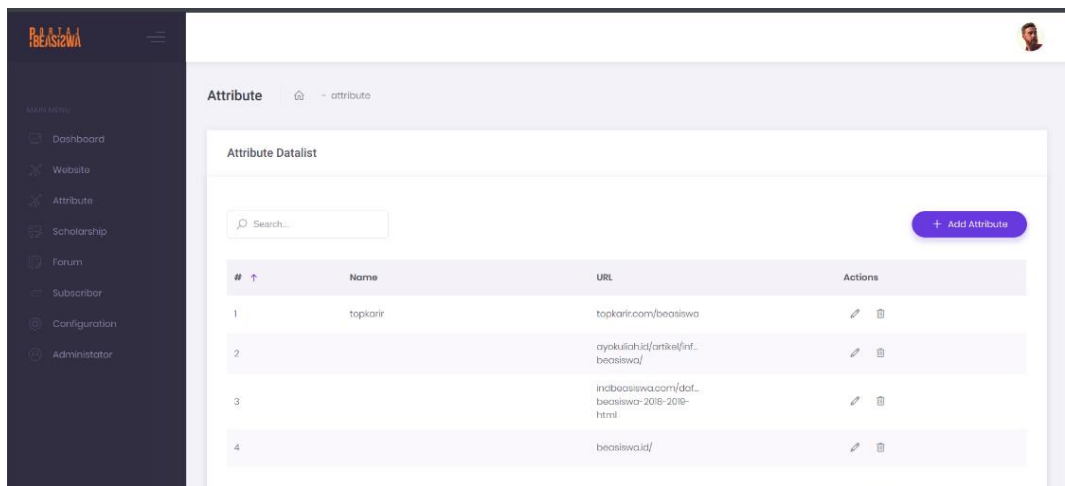


Gambar 8 Delete Confirmation

5.3. Mengelola Attribute Website

A. Menambah Attribute Website

- 1) Login
- 2) Pilih Attribute pada sidebar



Attribute

Attribute Datalist

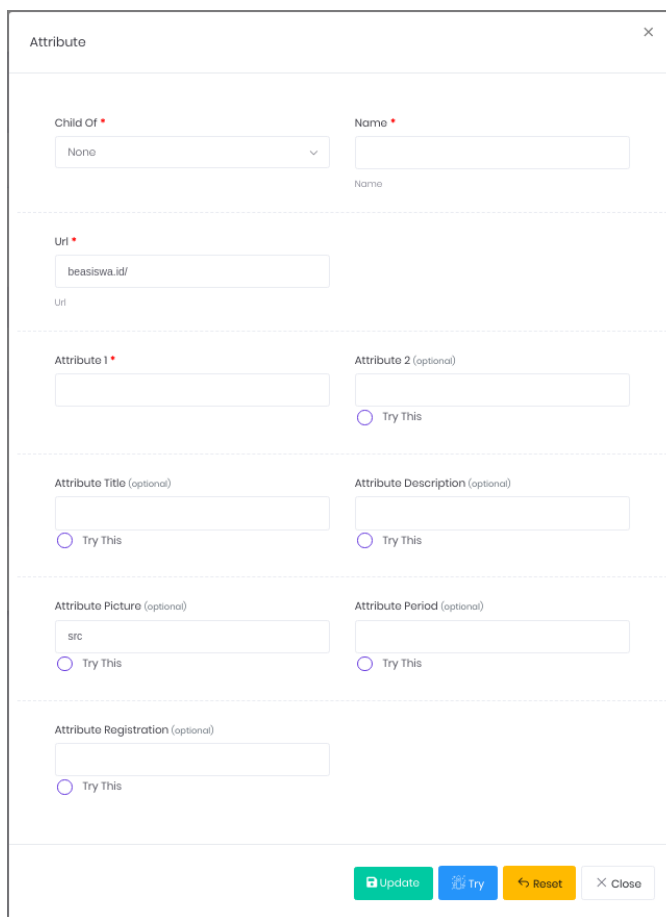
Search...

[+ Add Attribute](#)

#	Name	URL	Actions
1	topkarir	topkarir.com/beasiswa	Edit Delete
2		cyokulichist/artikel/inf... beasiswa/	Edit Delete
3		indobasiswa.com/dof... beasiswa-2018-2019- html	Edit Delete
4		beasiswa.id/	Edit Delete

Gambar 9 Attribute Tabel

- 3) Klik Add Attribute
- 4) Isi data Attribute pada form



Attribute

Child Of *

None

Name *

Name

Url *

beasiswa.id/

Url

Attribute 1 *

Attribute 2 (optional)

Try This

Attribute Title (optional)

Try This

Attribute Description (optional)

Try This

Attribute Picture (optional)

src

Try This

Attribute Period (optional)

Try This

Attribute Registration (optional)

Try This

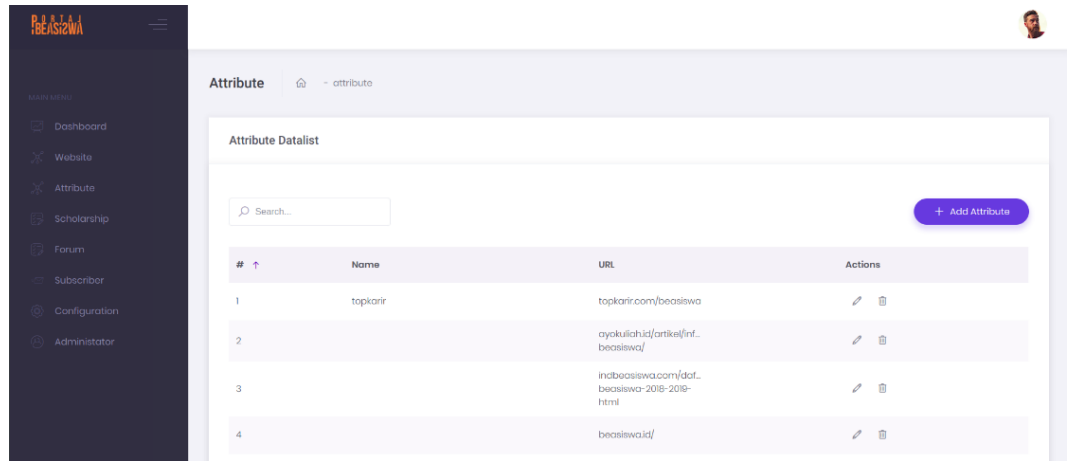
Update Try Reset Close

Gambar 10 Attribute Form

- 5) Klik Submit

B. Memperbarui Attribute Website

- 1) Login
- 2) Pilih Attribute pada sidebar



Gambar 11 Attribute Tabel

- 3) Klik Ikon Edit pada data table Attribute
- 4) Isi data Attribute pada form edit

Attribute

Child Of (optional)

None

Website *

lpdp

Name *

BEASISWA SANTRI

Uri *

https://www.lpd.kemenkeu.go.id/beasiswa-santri/

Attribute (optional)

Attribute Title (optional)

.h2

Try This

Attribute Description (optional)

div .wpb_text_column li

Try This

Attribute Picture (optional)

Try This

Attribute Period (optional)

Try This

Attribute Registration (optional)

Try This

Attribute Additional 1 (optional)

Try This

Attribute Additional 2 (optional)

Try This

Attribute Additional 3 (optional)

Try This

Attribute Additional 4 (optional)

Try This

Attribute Additional 5 (optional)

Try This

Submit

Try

Reset

Close

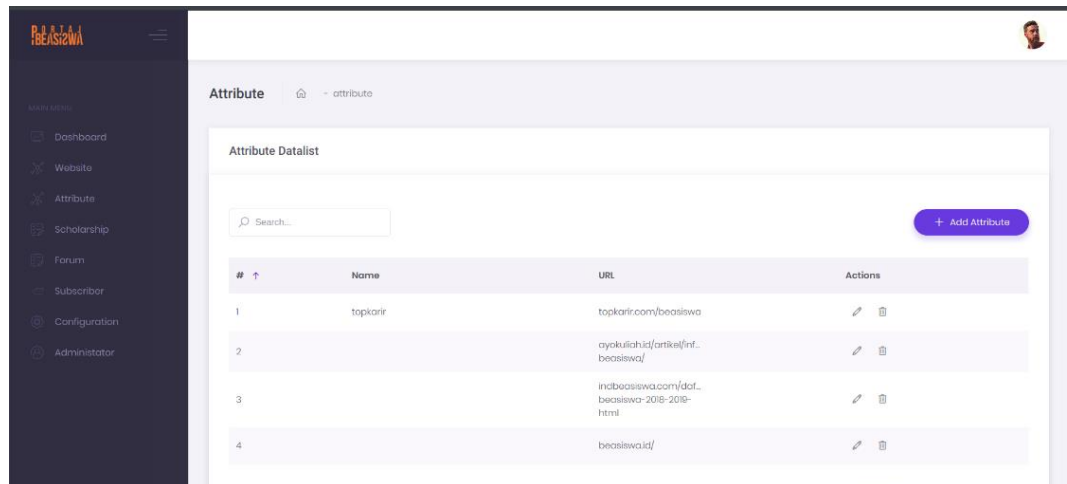
Gambar 12 Form Edit Website

5) Klik Update

C. Menghapus Attribute Website

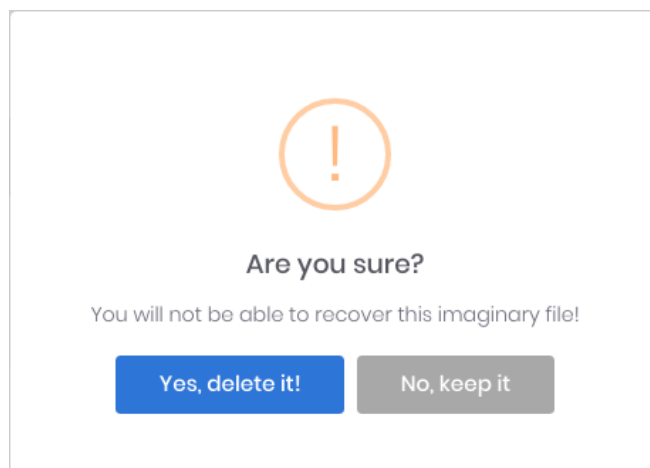
1) Login

2) Pilih Attribute pada sidebar



Gambar 13 Tabel Attribute

- 3) Klik Ikon Delete pada data table Website
- 4) Klik Yes, delete it

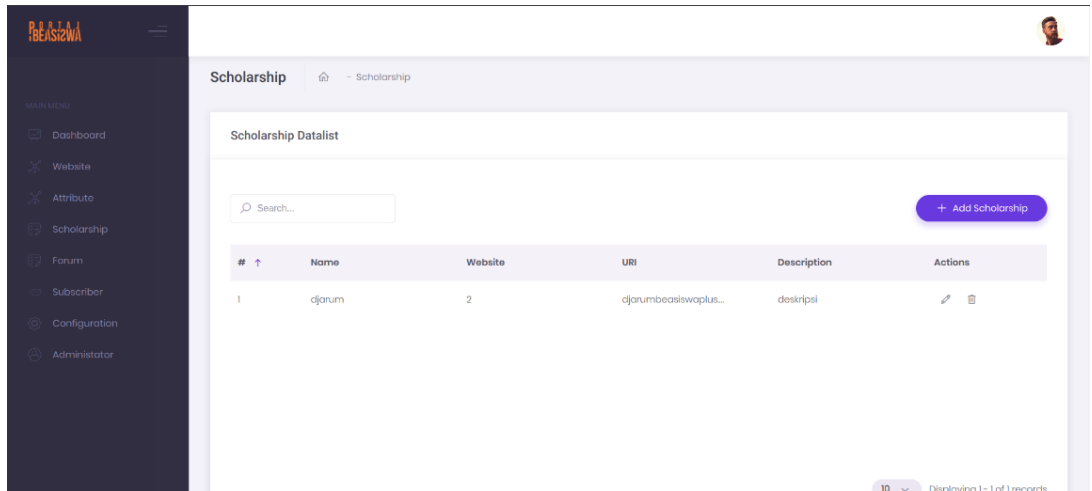


Gambar 14 Delete Confirmation

5.4. Mengelola Beasiswa

A. Menambah Beasiswa

- 1) Login
- 2) Pilih Scholarship pada sidebar



Gambar 15 Tabel Beasiswa

3) Klik Add Scholarship

4) Isi data Scholarship pada form

The screenshot shows the 'Add Scholarship' form. It includes input fields for Name, Website (a dropdown menu currently showing 'null'), URI, and Description (optional). There is also a 'Poster (optional)' field with a 'Browse' button. A '+ Add Website' button is located below the Website field. At the bottom of the form are three buttons: 'Submit' (green), 'Reset' (orange), and 'Close' (grey).

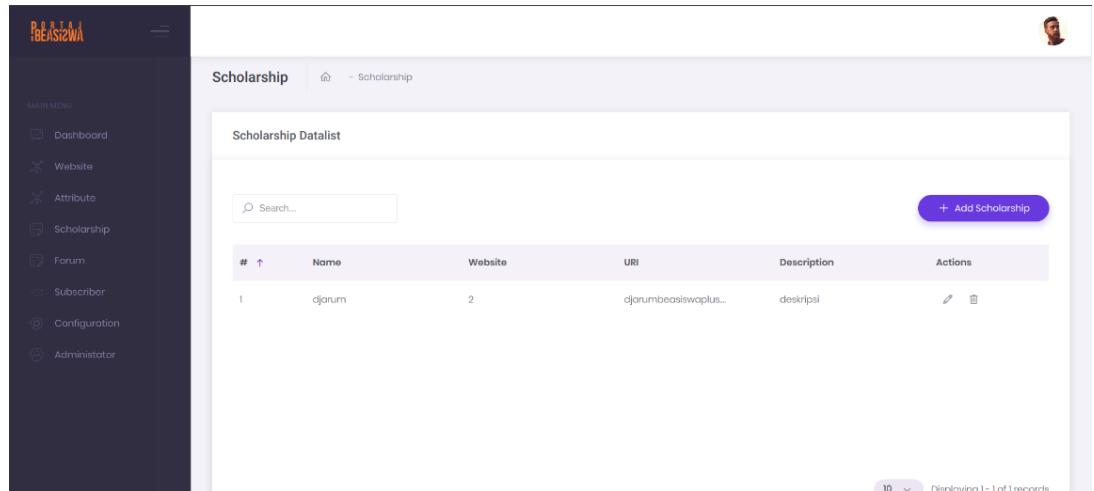
Gambar 16 Beasiswa Form

5) Klik Submit

B. Memperbarui Scholarship

1) Login

2) Pilih Scholarship pada sidebar



Gambar 17 Tabel Beasiswa

- 3) Klik Ikon Edit pada data table Beasiswa
- 4) Isi data Scholarship pada form edit

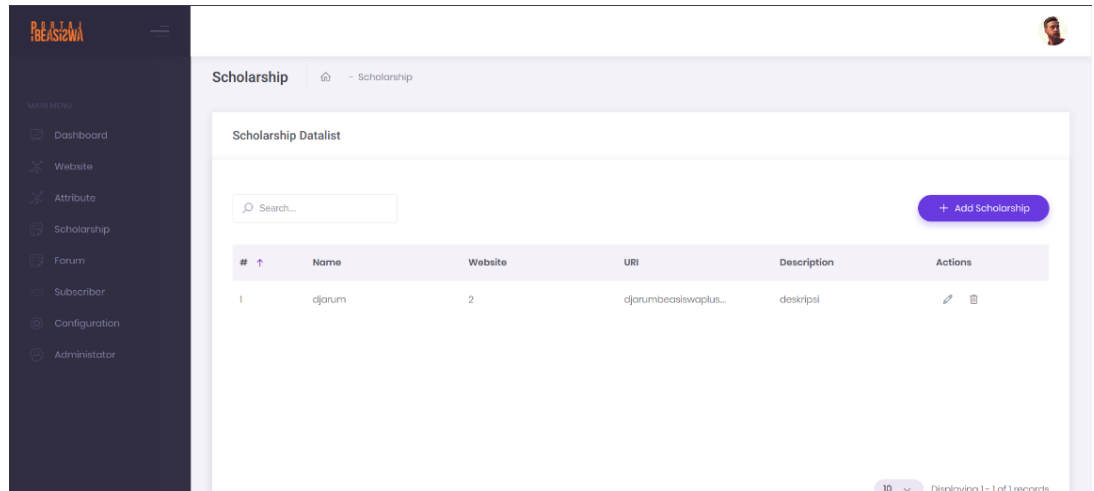
The screenshot shows the 'Scholarship' edit form. It contains several input fields: 'Name' (filled with 'Beasiswa Reguler'), 'Website' (a dropdown menu with 'lpdp' selected), 'Uri' (filled with 'https://www.lpd.kemenkeu.go.id/beasiswa-reguler/'), and 'Period' (filled with '2019'). There are also optional fields for 'Poster' (with a 'Browse' button) and 'Description' (a text area with HTML content). At the bottom right are three buttons: 'Update' (green), 'Reset' (yellow), and 'Close' (white).

Gambar 18 Form Edit Beasiswa

- 5) Klik Update

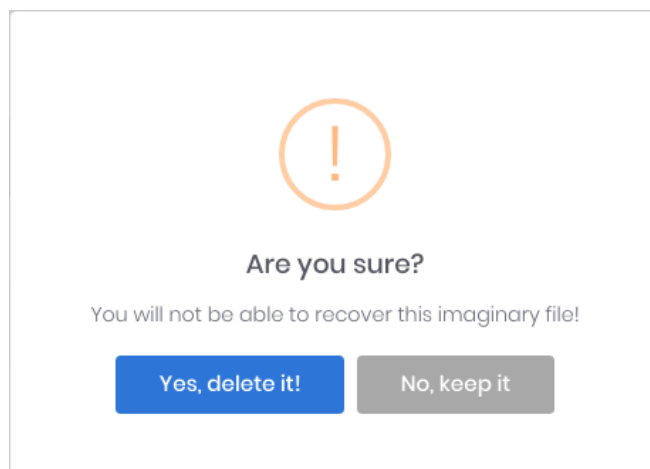
C. Menghapus Scholarship

- 1) Login
- 2) Pilih Scholarship pada sidebar



Gambar 19 Tabel Beasiswa

- 3) Klik Ikon Delete pada data tabel Beasiswa
- 4) Klik Yes, delete it

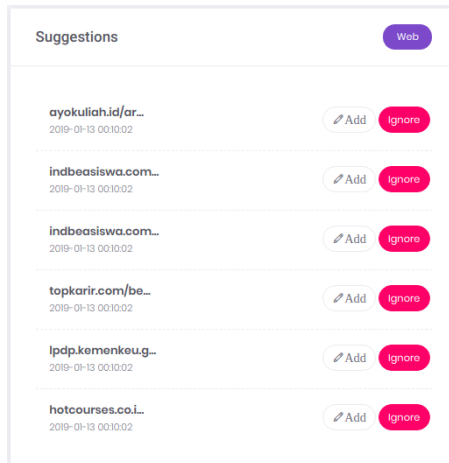


Gambar 20 Delete Confirmation

5.5. Mengelola *Suggestion*

A. Menambahkan Website dari *Suggestion*

- 1) Login
- 2) Klik Add pada Website *Suggestion* di Dashboard



Gambar 21 Website Suggestion

3) Lengkapi data Website

Gambar 22 Website Form Suggestion

4) Klik Submit

B. Mengabaikan *Suggestion*

1) Login

2) Klik Ignore pada Website *Suggestion* di Dashboard

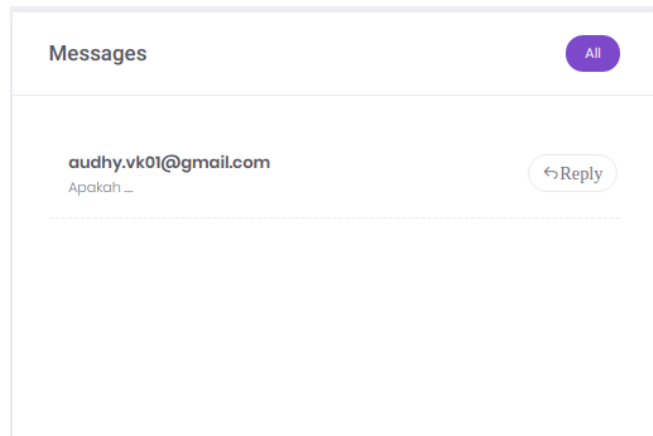


Gambar 23 Tombol abaikan suggestion

5.6. Membalas Pesan

1) Login

2) Klik Reply pada Messages di Dashboard



Gambar 24 Daftar Pesan

3) Isi data Pesan

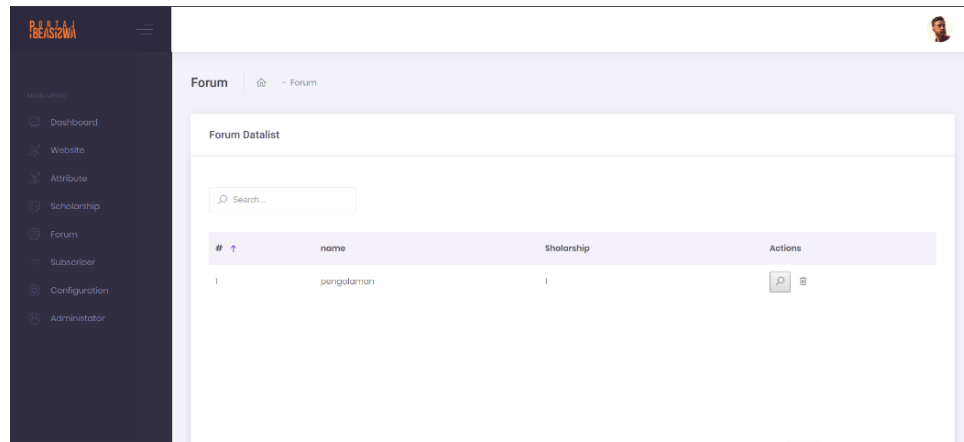
A screenshot of a 'Reply' form in a web application. The form is titled 'Reply' with a close button (X) in the top right corner. It contains three main input areas: 1) An 'Email' field with a red asterisk, containing the text 'audhy.vk01@gmail.com'. 2) A 'Message' field with a red asterisk, containing the text 'saya ingin bertanya mengenai proses beasiswa apa bisa ditampilkan lebih detail'. 3) A 'Reply' field with a red asterisk, which is currently empty. At the bottom right of the form is a green 'Send' button with a paper plane icon.

Gambar 25 Reply Form

4) Klik Sent

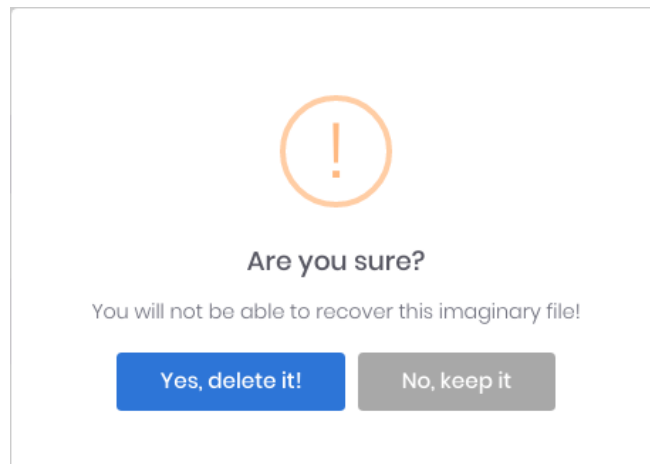
5.7. Menghapus Forum

- 1) Login
- 2) Pilih Forum pada sidebar



Gambar 26 Tabel Forum

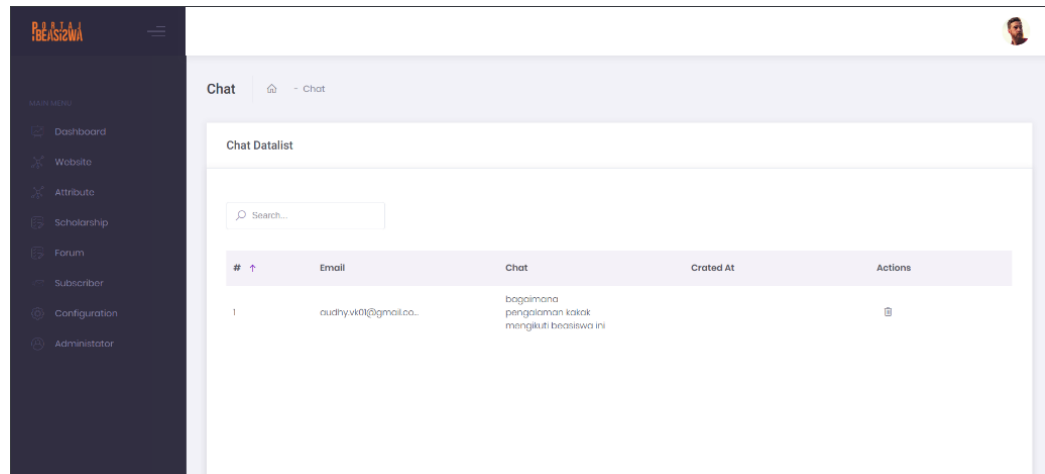
- 3) Klik ikon Delete pada data tabel Forum
- 4) Klik Yes, delete it



Gambar 27 Delete Confirmation

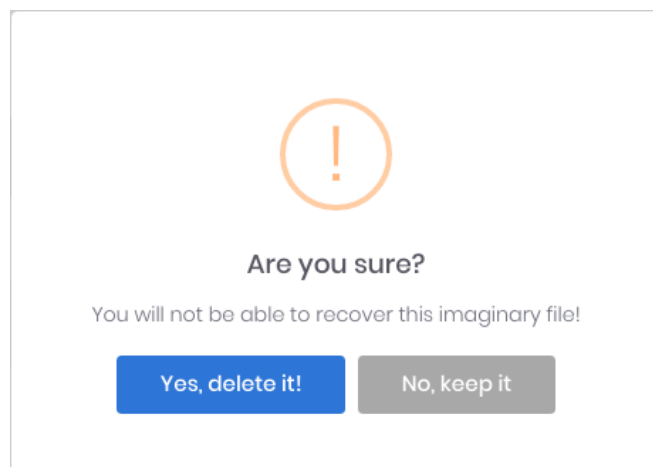
5.8. Menghapus Chat

- 1) Login
- 2) Pilih Forum pada sidebar



Gambar 28 Tabel Chat

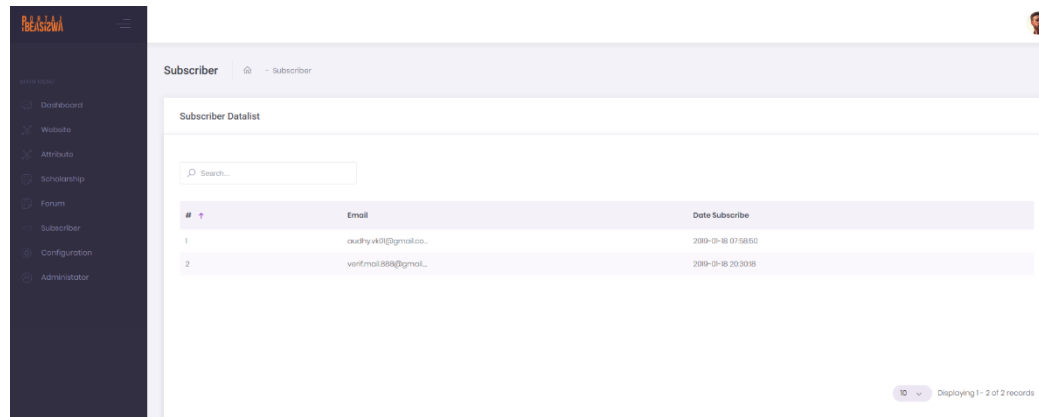
- 5) Pilih ikon detil pada data tabel Forum
- 6) Klik ikon Delete pada data tabel Chat
- 7) Klik Yes, delete it



Gambar 29 Delete Conformation

5.9. Melihat Subscriber

- 1) Login
- 2) Pilih *Subscriber* pada sidebar



Subscriber

Subscriber Datalist

Search...

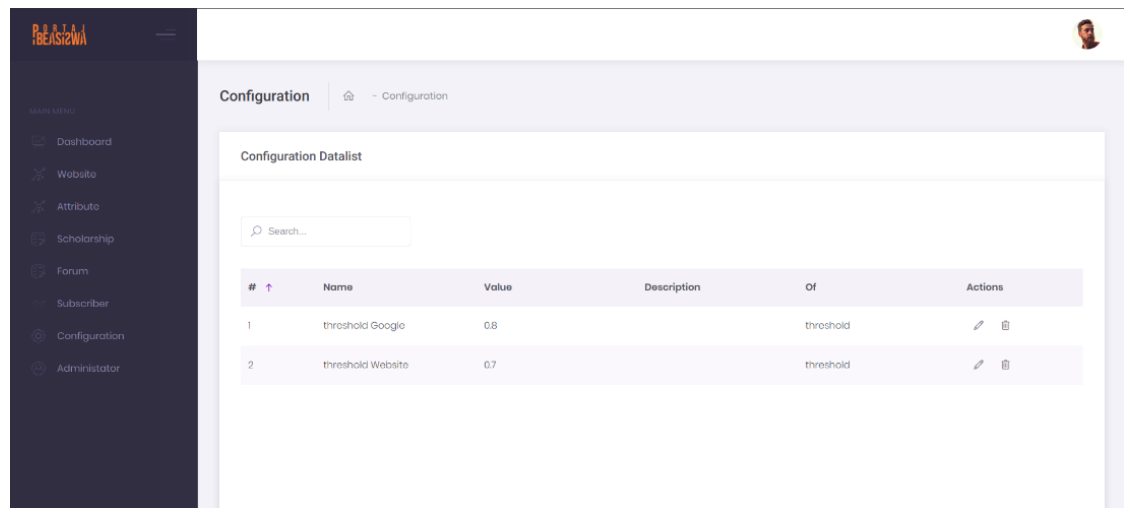
#	Email	Date Subscribe
1	oudhyvki@gmail.com	2019-09-18 07:58:50
2	verfmal88@gmail.com	2019-09-18 20:30:08

10 Displaying 1 - 2 of 2 records

Gambar 30 Tabel Subscriber

5.10. Mengubah *Configuration*

- 1) Login
- 2) Pilih *Configuration* pada sidebar



Configuration

Configuration Datalist

Search...

#	Name	Value	Description	Of	Actions
1	threshold Google	0.8		threshold	 
2	threshold Website	0.7		threshold	 

Gambar 31 Tabel Configuration

- 3) Klik Ikon Edit pada data table *Configuration*
- 4) Perbarui *Configuration*

The image shows a 'Configuration' modal window. It has a title bar with a close button (X). Inside, there are two input fields: 'Name' with the value 'threshold Google' and 'value' with the value '0.8'. Below these are labels 'Name' and 'Value'. There is also a 'Description (optional)' text area. At the bottom right, there are three buttons: 'Update' (green), 'Reset' (yellow), and 'Close' (grey).

Gambar 32 Edit Configuration Form

5) Klik Submit

6. Menggunakan Portal Beasiswa Sebagai Pencari Beasiswa

6.1. Melihat Detil Beasiswa

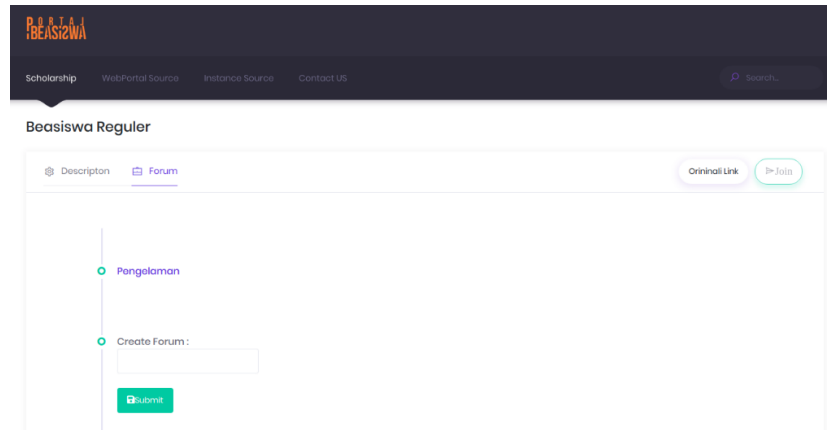
Klik “Read More” pada beasiswa yang diminati

The image shows the 'Beasiswa Regular' page on the BEASISWA portal. The page has a dark header with the BEASISWA logo and navigation links: Scholarship, WebPortal Source, Instance Source, and Contact Us. There is a search bar on the right. Below the header, the title 'Beasiswa Regular' is displayed. The main content area has tabs for 'Description' (selected) and 'Forum'. There is a 'Original Link' button and a 'Juga' button. The text describes the Beasiswa Pendidikan Indonesia Program Regular, which is a scholarship for Indonesian citizens to study abroad. It mentions that the scholarship is for students who have completed high school and are currently studying in a foreign country. The text also mentions that the scholarship is for students who have completed high school and are currently studying in a foreign country.

Gambar 33 Halaman Detil Beasiswa

6.2. Membuat Forum

- 1) Klik “Read More” pada beasiswa yang diminati
- 2) Klik Tab Forum

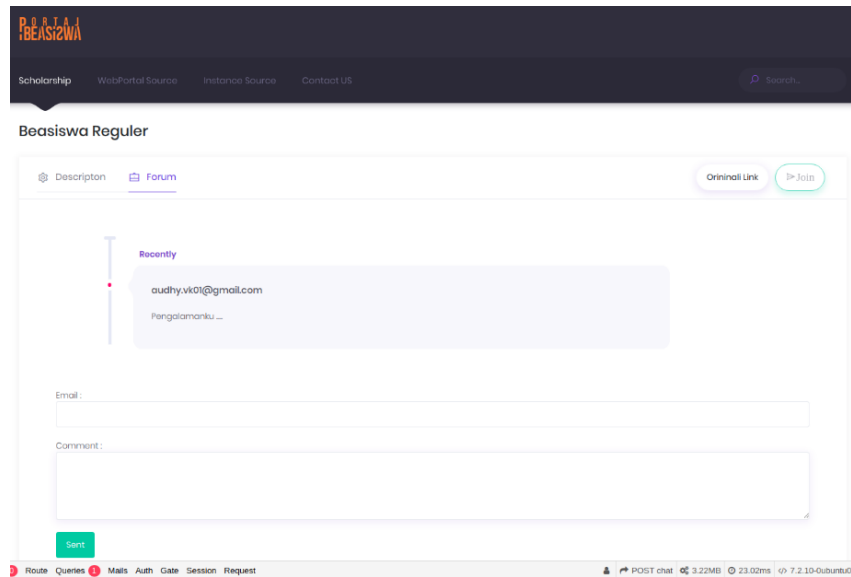


Gambar 34 Forum

- 3) Isi nama Forum
- 4) Klik Submit

6.3. Membuat Chat

- 1) Klik “Read More” pada beasiswa yang diminati
- 2) Klik Tab Forum
- 3) Pilih Forum
- 4) Isi email dan chat yang ingin ditulis



Gambar 35 Chat

- 5) Klik Submit

6.4. Melakukan Subscribe

- 1) *Scroll* kebawah halaman

[About Us](#)

☐ Register your email to follow news information

Email

[Subscribe](#)

Address :
Perum XXX, kecamatan Krian, Sidoarjo

[Facebook](#)

Phone :
081232353834

[Twitter](#)

Gambar 36 Subscribe

- 2) Isi Email
- 3) Klik Subscribe

6.5. Menambah Administrator

- 1) Login
- 2) Pilih Administrator pada sidebar

Administrator

Administrator Datalist

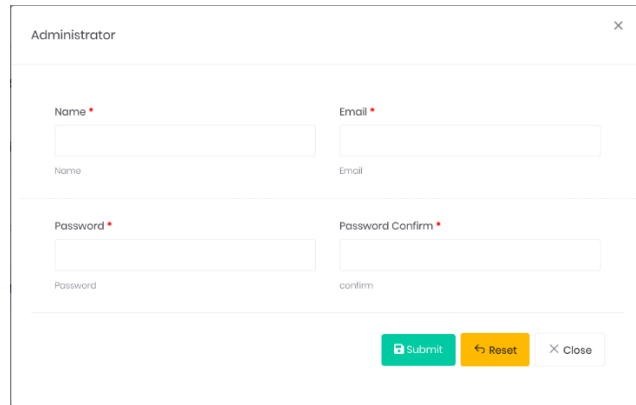
Search...

[+ Add User](#)

#	Name	Email	Actions
1	audhy	audhyxk0@gmail.com	Edit Delete

Gambar 37 Tabel Administrator

- 3) Klik Add Administrator
- 4) Isi data Administrator pada form



Administrator

Name *

Email *

Password *

Password Confirm *

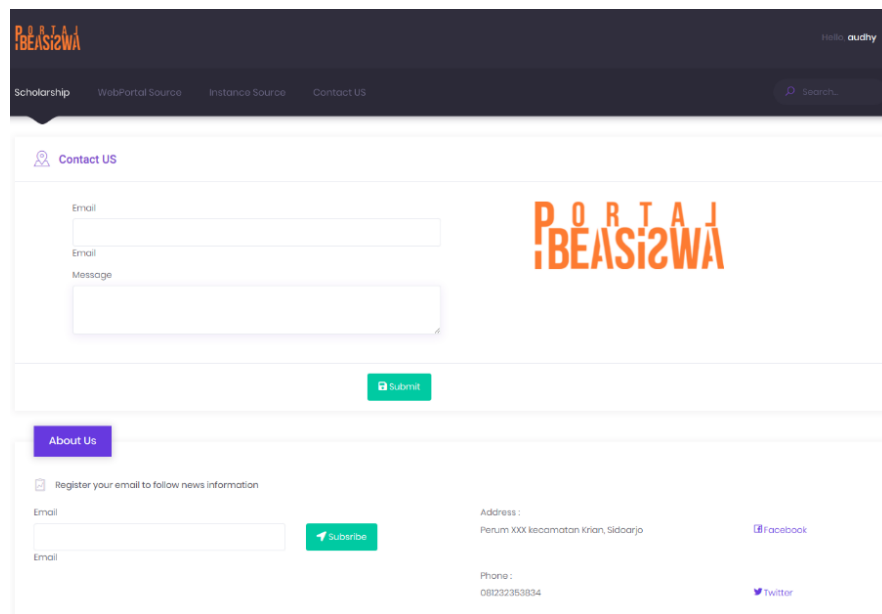
Submit Reset Close

Gambar 38 Adminibstrator Form

5) Klik Submit

6.6. Menghubungi Administrator

1) Klik Contact Us pada navbar



PORTAL BEASISWA

Scholarship WebPortal Source Instance Source Contact Us

Contact US

Email

Email

Message

Submit

About Us

Register your email to follow news information

Email

Subscribe

Address :

Perum XXX Kecamatan Krian, Sidoarjo

Facebook

Phone :

081222353834

Twitter

Gambar 39 Halaman Contact Administrator

2) Isi email dan pesan

3) Klik Submit

1. Souce Code

Model Website

```
<?php

namespace App;

use App\Scholarship;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Website extends Model
{
    use SoftDeletes;

    protected $dates = ['deleted_at'];

    protected $fillable = [
        'name',
        'url',
        'attribute1',
        'attribute2',
        'status',
        'attribute_title',
        'attribute_picture',
        'attribute_description',
        'attribute_registration',
        'attribute_period',
    ];

    /**
     * @return mixed
     */
    public function scholarships()
    {
        // This method must have a second parameter as FK column (website_id),

        return $this->hasMany(Scholarship::class, 'website_id');
    }

    /**
     * One-to-Many
     */
}
```

```

*
* @return mixed
*/
public function scholarship_attributes()
{
    // This method must have a second parameter as FK column (website_id),

    return $this->hasMany(Scholarship_attribute::class, 'website_id');
}

}

```

Model Scholarchip_attribute

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Scholarship_attribute extends Model
{
    use SoftDeletes;

    protected $dates = ['deleted_at'];

    protected $fillable = [
        'url',
        'name',
        'attribute_id',
        'website_id',
        'attribute1',
        'attribute_title',
        'attribute_picture',
        'attribute_description',
        'attribute_registration',
        'attribute_period',
        'attribbute_id',
        'attribute_additional_1',
        'attribute_additional_2',
        'attribute_additional_3',
        'attribute_additional_4',
        'attribute_additional_5',
    ];
}

```

```

/***** RELATIONSHIP
*****/

```

```

/**
 * One-to-Many: A website may have zero or many scholarship.
 *
 * @return mixed
 */
public function website()
{
    return $this->belongsTo(Website::class);
}
}

```

ModelScholarship

```

<?php

namespace App;

use App\Website;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Scholarship extends Model
{
    use SoftDeletes;

    protected $dates = ['deleted_at'];

    protected $fillable = [
        'name',
        'description',
        'uri',
        'picture',
        'reg',
        'additional',
        'website_id',
        'attribute_id',
    ];

/***** RELATIONSHIP
*****/

/**
 * One-to-Many: A website may have zero or many scholarship.

```

```

*
* This function will retrieve the header of a general license.
*
* @return mixed
*/
public function website()
{
    return $this->belongsTo(Website::class);
}

/**
 * One-to-Many
 *
 * @return mixed
 */
public function periods()
{
    // This method must have a second parameter as FK column
    (scholarship_id),

    return $this->hasMany(Period::class, 'scholarship_id');
}
}

```

Model Period

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Period extends Model
{
    use SoftDeletes;

    protected $dates = ['deleted_at'];

    protected $fillable = [
        'period',
        'scholarship_id',
    ];

    /**
     * One-to-Many: A period may have zero or many scholarship.
     */
}

```

```

        *
        * @return mixed
        */
        public function scholarship()
        {
            return $this->belongsTo(Scholarship::class);
        }
    }
}

```

Model Configuration

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Configuration extends Model
{
    protected $fillable = [
        'name',
        'description',
        'value',
        'of',
    ];

}

```

ScraperController

```

<?php

namespace App\Http\Controllers;

use App\Log;
use App\Period;
use App\Configuration;
use App\Scholarship;
use App\Scholarship_attribute;
use App\Website;
use Goutte\Client;
use Illuminate\Http\Request;

class ScaperController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
}

```

```

*/

public function google()
{
    try {

        $Websites = Website::pluck('url');
        if ($Websites->isEmpty()) {
            $client = new Client();
            $crawler = $client->request('GET',
'https://google.com/search?q=beasiswa');
            $crawler->filter('cite')->each(function ($node) {
                $web = str_replace(array('https://', 'www.'), '', $node->text());
                $Website = Website::create([
                    'url' => $web,
                    'status' => '1',
                ]);
            });
        } else {
            $threshold = Configuration::find(1);
            $threshold = $threshold->value;
            $client = new Client();
            $crawler = $client->request('GET',
'https://google.com/search?q=beasiswa');
            $urls = array();
            $matches = array();
            $crawler->filter('cite')->each(function ($node) use (&$threshold,
&$urls, &$matches) {
                $Websites = Website::pluck('url');
                foreach ($Websites as $Website) {
                    $comparison = new
\Atomescrochus\StringSimilarities\Compare();
                    $web = str_replace(array('https://', 'www.'), '', $node->text());
                    $jaroWinkler = $comparison->jaroWinkler($Website, $web);

                    $urls[] = $node->text();
                    if ($jaroWinkler >= $threshold) {
                        $matches[] = $node->text();
                    }
                }
            });

            $MatchArray = array_unique($matches);
            $UrlArray = array_unique($urls);
            $diffs = array_diff($UrlArray, $MatchArray);
            foreach ($diffs as $diff) {
                $web = str_replace(array('https://', 'www.'), '', $diff);
            }
        }
    }
}

```

```

        $Website = Website::create([
            'url' => $web,
            'status' => '1',
        ]);
    }
}

Log::create([
    'type' => 1,
    'status' => 1,
]);
echo 'Done';

} catch (\Exception $e) {

    if (strpos($e, 'timed out') !== false) {
        Log::create([
            'type' => 1,
            'status' => 0,
            'message' => 'Time Out, Maybe Website Down or Your connection
get trouble',
        ]);

        echo 'Timeout ' . '<br>' . $e;
    }
    else{
        Log::create([
            'type' => 1,
            'status' => 0,
            'message' => 'Unknow Error please Check Scrap Website
Manually',
        ]);

        echo 'Error - ' . $e;

    }

}

}

public function website()
{
    try {

        $Scholarship = Scholarship::pluck('description');
        if ($Scholarship->isEmpty()) {

```

```

$attributes = Scholarship_attribute::with('website')->get();
foreach ($attributes as $attribute) {

    if ($attribute->website->status == 2) {
        $client = new Client();

        $attribute_id = $attribute->id;
        $website_id = $attribute->website_id;
        $attribute_url = $attribute->url;
        $attribute_title = $attribute->attribute_title;
        $attribute_desc = $attribute->attribute_description;
        $attribute_picture = $attribute->attribute_picture;
        $attribute_registration = $attribute->attribute_registration;
        $attribute_period = $attribute->attribute_period;
        $attribute_additional1 = $attribute->attribute_additional_1;
        $attribute_additional2 = $attribute->attribute_additional_2;
        $attribute_additional3 = $attribute->attribute_additional_3;
        $attribute_additional4 = $attribute->attribute_additional_4;
        $attribute_additional5 = $attribute->attribute_additional_5;

        $guzzleclient = new \GuzzleHttp\Client(array(
            'timeout' => 90,
            'verify' => false,
        ));
        $client->setClient($guzzleclient);
        $crawler = $client->request('GET', $attribute->url);

        $crawler->filter($attribute->attribute1)->each(function ($node)
        use (&$attribute_desc, &$attribute_title, &$attribute_picture, &$attribute_url,
        &$attribute_website, &$attribute_id, &$website_id, &$attribute_registration,
        &$attribute_additional1, &$attribute_additional2, $attribute_additional3,
        $attribute_additional4, $attribute_additional5) {
            $link = $node->attr('href');
            $client = new Client();
            $crawlerdtl = $client->request('GET', $link);
            $title = "";
            if ($attribute_title != null) {
                $crawlerdtl->filter($attribute_title)->each(function ($node)
                use (&$title, &$attribute_title) {
                    $title = $node->text();
                });
            }

            $desc = "";
            if ($attribute_desc != null) {
                $crawlerdtl->filter($attribute_desc)->each(function ($node)
                use (&$desc, &$attribute_desc) {

```

```

        $desc = $desc . '<p>' . $node->text() . '</p>';
    });
}

$regs = "";
if ($attribute_registration != null) {
    $crawlerdtl->filter("")->each(function ($node) use (&$regs,
&$attribute_registration) {
        $regs = $node->attr('href');
    });
}

$picture = "";
if ($attribute_picture != null) {
    $crawlerdtl->filter($attribute_picture)->each(function
($node) use (&$picture, &$attribute_picture) {
        $picture = $node->attr('src');
    });
}

$additional1 = "";
if ($attribute_additional1 != null) {
    $crawlerdtl->filter($attribute_additional1)->each(function
($node) use (&$additional1, &$attribute_additional1) {
        $additional1 = $additional1 . '<p>' . $node->text() . '</p>';
    });
}

$additional2 = "";
if ($attribute_additional2 != null) {
    $crawlerdtl->filter($attribute_additional2)->each(function
($node) use (&$additional2, &$attribute_additional2) {
        $additional2 = $additional2 . '<p>' . $node->text() . '</p>';
    });
}

$additional3 = "";
if ($attribute_additional3 != null) {
    $crawlerdtl->filter($attribute_additional3)->each(function
($node) use (&$additional3, &$attribute_additional3) {
        $additional3 = $additional3 . '<p>' . $node->text() . '</p>';
    });
}

$additional4 = "";
if ($attribute_additional4 != null) {

```

```

        $crawlerdtl->filter($attribute_additional4)->each(function
($node) use (&$additional4, &$attribute_additional4) {
            $attribute_additional4 = $attribute_additional4 . '<p>' .
$node->text() . '</p>';
        });
    }

    $additional5 = "";
    if ($attribute_additional5 != null) {
        $crawlerdtl->filter($attribute_additional5)->each(function
($node) use (&$additional5, &$attribute_additional5) {
            $additional5 = $additional5 . '<p>' . $node->text() . '</p>';
        });
    }

    $combine = '<p>' . $additional1 . '</p>' . '<p>' . $additional2 .
'</p>' . '<p>' . $additional3 . '</p>' . '<p>' . $additional4 . '</p>' . '<p>' .
$additional5 . '</p>';

    if ($attribute_picture == null or $attribute_picture == "") {
        $path = '/img/no_picture/' . rand(1, 2) . '.jpg';

        $Scholarship = Scholarship::create([
            'name' => $title,
            'uri' => $link,
            'website_id' => $website_id,
            'attribute_id' => $attribute_id,
            'reg' => $regs,
            'description' => $desc,
            'additional' => $combine,

        ]);
    } else {

        $scholarship = Scholarship::create([
            'name' => $title,
            'uri' => $link,
            'picture' => $picture,
            'website_id' => $website_id,
            'attribute_id' => $attribute_id,
            'reg' => $regs,
            'description' => $desc,
            'additional' => $combine,

        ]);
    }
}

```

```

$str = $Scholarship->description . $Scholarship->additional;
if (
    (!empty($str) && preg_match_all('~\b\d{4}\b\+?~', $str,
$matches))
) {
    $today = date("Y");
    $today = substr($today, 0, 2);
    $years = array();
    foreach($matches[0] as $year){
        $test = substr($year, 0, 2);
        if($test == $today){
            $years[] = $year;
        }
    }
    if($years == null ){
        $period = date("Y");
    }
    else{
        $value = max($years);
        $period = $value;
    }
} else {
    $period = date("Y");
}

$period = Period::create([
    'scholarship_id' => $Scholarship->id,
    'period' => $period,
]);

});

} elseif ($Attribute->website->status == 3) {
    if ($Attribute->attribute_id != null) {
        $client = new Client();
        $attribute_id = $Attribute->id;
        $attribute_title = $Attribute->attribute_title;
        $attribute_desc = $Attribute->attribute_description;
        $attribute_picture = $Attribute->attribute_picture;
        $attribute_registration = $Attribute->attribute_registration;
        $attribute_period = $Attribute->attribute_period;
        $attribute_additional1 = $Attribute->attribute_additional_1;
        $attribute_additional2 = $Attribute->attribute_additional_2;
        $attribute_additional3 = $Attribute->attribute_additional_3;
        $attribute_additional4 = $Attribute->attribute_additional_4;
        $attribute_additional5 = $Attribute->attribute_additional_5;
    }
}

```

```

$guzzleclient = new \GuzzleHttp\Client(array(
    'timeout' => 90,
    'verify' => false,
));
$client->setClient($guzzleclient);
$crawler = $client->request('GET', $attribute->url);

if ($attribute_title != null or $attribute_title != "") {
    $crawler->filter($attribute_title)->each(function ($node) use
(&$title, &$attribute_title) {
        $title = $node->text();
    });
    $Scholarship = Scholarship::where('attribute_id', $attribute-
>attribute_id)->first();
    $Scholarship->name = $title;
    $Scholarship->save();
}

if ($attribute_desc != null) {
    $crawler->filter($attribute_desc)->each(function ($node)
use (&$desc, &$attribute_desc) {
        $desc = $desc . '<p>' . $node->text() . '</p>';
    });
    $Scholarship = Scholarship::where('attribute_id', $attribute-
>attribute_id)->first();
    $Scholarship->name = $desc;
    $Scholarship->save();
}

if ($attribute_picture != null) {
    $crawler->filter($attribute_picture)->each(function ($node)
use (&$picture, &$attribute_picture) {
        $picture = $node->attr('src');
    });
    $Scholarship = Scholarship::where('attribute_id', $attribute-
>attribute_id)->first();
    $Scholarship->name = $picture;
    $Scholarship->save();
}

if ($attribute_registration != null) {
    $crawler->filter($attribute_registration)->each(function
($node) use (&$reg, &$attribute_registration) {
        $reg = $node->attr('href');
    });
    $Scholarship = Scholarship::where('attribute_id', $attribute-
>attribute_id)->first();

```

```

        $Scholarship->name = $reg;
        $Scholarship->save();
    }

    $additional1 = "";
    if ($attribute_additional1 != null) {
        $crawler->filter($attribute_additional1)->each(function
($node) use (&$additional1, &$attribute_additional1) {
            $additional1 = $additional1 . '<p>' . $node->text() . '</p>';
        });
    }

    $additional2 = "";
    if ($attribute_additional2 != null) {
        $crawler->filter($attribute_additional2)->each(function
($node) use (&$additional2, &$attribute_additional2) {
            $additional2 = $additional2 . '<p>' . $node->text() . '</p>';
        });
    }

    $additional3 = "";
    if ($attribute_additional3 != null) {
        $crawler->filter($attribute_additional3)->each(function
($node) use (&$additional3, &$attribute_additional3) {
            $additional3 = $additional3 . '<p>' . $node->text() . '</p>';
        });
    }

    $additional4 = "";
    if ($attribute_additional4 != null) {
        $crawler->filter($attribute_additional4)->each(function
($node) use (&$additional4, &$attribute_additional4) {
            $attribute_additional4 = $attribute_additional4 . '<p>' .
$node->text() . '</p>';
        });
    }

    $additional5 = "";
    if ($attribute_additional5 != null) {
        $crawler->filter($attribute_additional5)->each(function
($node) use (&$additional5, &$attribute_additional5) {
            $additional5 = $additional5 . '<p>' . $node->text() . '</p>';
        });
    }

    $Scholarship = Scholarship::where('attribute_id', $attribute-
>attribute_id)->first();

```

```

        $combine = $Scholarship->additional . '<p>' . $additional1 .
'</p>' . '<p>' . $additional2 . '</p>' . '<p>' . $additional3 . '</p>' . '<p>' .
$additional4 . '</p>' . '<p>' . $additional5 . '</p>';

        $Scholarship->additional = $combine;
        $Scholarship->save();

        $str = $Scholarship->description . $Scholarship->additional;
        if (
            (!empty($str) && preg_match_all('~\b\d{4}\b\+?~', $str,
$matches))
        ) {
            $today = date("Y");
            $today = substr($today, 0, 2);
            $years = array();
            foreach($matches[0] as $year){
                $test = substr($year, 0, 2);
                if($test == $today){
                    $years[] = $year;
                }
            }
            if($years == null ){
                $period = date("Y");
            }
            else{
                $value = max($years);
                $period = $value;
            }
        } else {
            $period = date("Y");
        }

        $periods = Period::where('scholarship_id', $Scholarship->id)-
>first();

        if ($period > $periods->period) {
            $periods->period = $period;
            $periods->save();
        }

    } else {
        $client = new Client();
        $attribute_id = $attribute->id;
        $attribute_title = $attribute->attribute_title;
        $attribute_desc = $attribute->attribute_description;
        $attribute_picture = $attribute->attribute_picture;
        $attribute_registration = $attribute->attribute_registration;
        $attribute_period = $attribute->attribute_period;
    }

```

```

$attribute_additional1 = $attribute->attribute_additional_1;
$attribute_additional2 = $attribute->attribute_additional_2;
$attribute_additional3 = $attribute->attribute_additional_3;
$attribute_additional4 = $attribute->attribute_additional_4;
$attribute_additional5 = $attribute->attribute_additional_5;

$guzzleclient = new \GuzzleHttp\Client(array(
    'timeout' => 90,
    'verify' => false,
));
$client->setClient($guzzleclient);
$crawler = $client->request('GET', $attribute->url);

$title = "";
if ($attribute_title != null or $attribute_title != "") {
    $crawler->filter($attribute_title)->each(function ($node) use
(&$title, &$attribute_title) {
        $title = $node->text();
    });
}

$desc = "";
if ($attribute_desc != null) {
    $crawler->filter($attribute_desc)->each(function ($node)
use (&$desc, &$attribute_desc) {
        $desc = $desc . '<p>' . $node->text() . '</p>';
    });
}

$picture = "";
if ($attribute_picture != null) {
    $crawler->filter($attribute_picture)->each(function ($node)
use (&$picture, &$attribute_picture) {
        $picture = $node->attr('src');
    });
}

$reg = "";
if ($attribute_registration != null) {
    $crawler->filter($attribute_registration)->each(function
($node) use (&$reg, &$attribute_registration) {
        $text = $node->text();
        if ($text == "Daftar Online") {
            $reg = $node->attr('href');
        }
    });
}

```

```

$additional1 = "";
if ($attribute_additional1 != null) {
    $crawler->filter($attribute_additional1)->each(function
($node) use (&$additional1, &$attribute_additional1) {
        $additional1 = $additional1 . '<p>' . $node->text() . '</p>';
    });
}

```

```

$additional2 = "";
if ($attribute_additional2 != null) {
    $crawler->filter($attribute_additional2)->each(function
($node) use (&$additional2, &$attribute_additional2) {
        $additional2 = $additional2 . '<p>' . $node->text() . '</p>';
    });
}

```

```

$additional3 = "";
if ($attribute_additional3 != null) {
    $crawler->filter($attribute_additional3)->each(function
($node) use (&$additional3, &$attribute_additional3) {
        $additional3 = $additional3 . '<p>' . $node->text() . '</p>';
    });
}

```

```

$additional4 = "";
if ($attribute_additional4 != null) {
    $crawler->filter($attribute_additional4)->each(function
($node) use (&$additional4, &$attribute_additional4) {
        $attribute_additional4 = $attribute_additional4 . '<p>' .
$node->text() . '</p>';
    });
}

```

```

$additional5 = "";
if ($attribute_additional5 != null) {
    $crawler->filter($attribute_additional5)->each(function
($node) use (&$additional5, &$attribute_additional5) {
        $additional5 = $additional5 . '<p>' . $node->text() . '</p>';
    });
}

```

```

$combine = '<p>' . $additional1 . '</p>' . '<p>' . $additional2 .
'</p>' . '<p>' . $additional3 . '</p>' . '<p>' . $additional4 . '</p>' . '<p>' .
$additional5 . '</p>';

```

```

if ($attribute_picture != null or $attribute_picture == "") {

```

```

$path = '/img/no_picture/' . rand(1, 2) . '.jpg';

$scholarship = Scholarship::create([
    'name' => $title,
    'uri' => $attribute->url,
    'website_id' => $attribute->website_id,
    'attribute_id' => $attribute->id,
    'description' => $desc,
    'reg' => $reg,
    'additional' => $combine,
]);

} else {

    $scholarship = Scholarship::create([
        'name' => $title,
        'uri' => $attribute->url,
        'picture' => $picture,
        'website_id' => $attribute->website_id,
        'attribute_id' => $attribute->id,
        'description' => $desc,
        'reg' => $reg,
        'additional' => $combine,

    ]);

}
$str = $scholarship->description . $scholarship->additional;
if (
    (!empty($str) && preg_match_all('~\b\d{4}\b\+?~', $str,
$matches))
) {
    $today = date("Y");
    $today = substr($today, 0, 2);
    $years = array();
    foreach($matches[0] as $year){
        $test = substr($year, 0, 2);
        if($test == $today){
            $years[] = $year;
        }
    }
}
if($years == null ){
    $period = date("Y");
}
else{
    $value = max($years);
    $period = $value;
}

```

```

    }
    } else {
        $period = date("Y");
    }

    $period = Period::create([
        'scholarship_id' => $scholarship->id,
        'period' => $period,
    ]);

}

}

}

} else {
    $attributes = Scholarship_attribute::with('website')->get();
    foreach ($attributes as $attribute) {

        if ($attribute->website->status == 2) {
            $client = new Client();

            $attribute_id = $attribute->id;
            $website_id = $attribute->website_id;
            $attribute_url = $attribute->url;
            $attribute_title = $attribute->attribute_title;
            $attribute_desc = $attribute->attribute_description;
            $attribute_picture = $attribute->attribute_picture;
            $attribute_registration = $attribute->attribute_registration;
            $attribute_period = $attribute->attribute_period;
            $attribute_additional1 = $attribute->attribute_additional_1;
            $attribute_additional2 = $attribute->attribute_additional_2;
            $attribute_additional3 = $attribute->attribute_additional_3;
            $attribute_additional4 = $attribute->attribute_additional_4;
            $attribute_additional5 = $attribute->attribute_additional_5;
            $threshold = Configuration::find(2);
            $threshold = $threshold->value;

            $guzzleclient = new \GuzzleHttp\Client(array(
                'timeout' => 90,
                'verify' => false,
            ));
            $client->setClient($guzzleclient);
            $crawler = $client->request('GET', $attribute->url);

```

```

        $crawler->filter($attribute->attribute1)->each(function ($node)
use (&$threshold, &$attribute_desc, &$attribute_title, &$attribute_picture,
&$attribute_url, &$attribute_website, &$attribute_id, &$website_id,
&$attribute_registration, &$attribute_additional1, &$attribute_additional2,
$attribute_additional3, $attribute_additional4, $attribute_additional5) {
            $dec_list = array();
            unset($dec_list);
            $dec_list = array();
            $web_matches = array();
            unset($web_matches);
            $web_matches = array();

            $link = $node->attr('href');
            $client = new Client();
            $crawlerdtl = $client->request('GET', $link);
            $title = "";
            if ($attribute_title != null) {
                $crawlerdtl->filter($attribute_title)->each(function ($node)
use (&$title, &$attribute_title) {
                    $title = $node->text();
                });
            }

            $desc = "";
            if ($attribute_desc != null) {
                $crawlerdtl->filter($attribute_desc)->each(function ($node)
use (&$desc, &$attribute_desc) {
                    $desc = $desc . '<p>' . $node->text() . '</p>';
                });
            }

            $regs = "";
            if ($attribute_registration != null) {
                $crawlerdtl->filter("")->each(function ($node) use (&$regs,
&$attribute_registration) {
                    $regs = $node->attr('href');
                });
            }

            $picture = "";
            if ($attribute_picture != null) {
                $crawlerdtl->filter($attribute_picture)->each(function
($node) use (&$picture, &$attribute_picture) {
                    $picture = $node->attr('src');
                });
            }

```

```

$additional1 = "";
if ($attribute_additional1 != null) {
    $crawlerdtl->filter($attribute_additional1)->each(function
($node) use (&$additional1, &$attribute_additional1) {
        $additional1 = $additional1 . '<p>' . $node->text() . '</p>';
    });
}

```

```

$additional2 = "";
if ($attribute_additional2 != null) {
    $crawlerdtl->filter($attribute_additional2)->each(function
($node) use (&$additional2, &$attribute_additional2) {
        $additional2 = $additional2 . '<p>' . $node->text() . '</p>';
    });
}

```

```

$additional3 = "";
if ($attribute_additional3 != null) {
    $crawlerdtl->filter($attribute_additional3)->each(function
($node) use (&$additional3, &$attribute_additional3) {
        $additional3 = $additional3 . '<p>' . $node->text() . '</p>';
    });
}

```

```

$additional4 = "";
if ($attribute_additional4 != null) {
    $crawlerdtl->filter($attribute_additional4)->each(function
($node) use (&$additional4, &$attribute_additional4) {
        $attribute_additional4 = $attribute_additional4 . '<p>' .
$node->text() . '</p>';
    });
}

```

```

$additional5 = "";
if ($attribute_additional5 != null) {
    $crawlerdtl->filter($attribute_additional5)->each(function
($node) use (&$additional5, &$attribute_additional5) {
        $additional5 = $additional5 . '<p>' . $node->text() . '</p>';
    });
}

```

```

$combine = '<p>' . $additional1 . '</p>' . '<p>' . $additional2 .
'</p>' . '<p>' . $additional3 . '</p>' . '<p>' . $additional4 . '</p>' . '<p>' .
$additional5 . '</p>';

```

```

$Scholarship = Scholarship::All();
foreach ($Scholarship as $Scholarship) {

```

```

        $comparison = new
\Atomescrochus\StringSimilarities\Compare();
        $jaroWinkler = $comparison->jaroWinkler($Scholarship-
>description, $desc);

        $dec_list[] = $Scholarship->description;
        if ($jaroWinkler >= $threshold) {
            $scholarship_ID = $Scholarship->id;
            $web_matches[] = $desc;
        }
    }
    $MatchArray = array_unique($web_matches);

    if ($MatchArray == null) {
        if ($attribute_picture == null or $attribute_picture == "") {
            $path = '/img/no_picture/' . rand(1, 2) . '.jpg';

            $Scholarship = Scholarship::create([
                'name' => $title,
                'uri' => $link,
                'website_id' => $website_id,
                'attribute_id' => $attribute_id,
                'reg' => $regs,
                'description' => $desc,
                'additional' => $combine,

            ]);
        } else {
            $scholarship = Scholarship::create([
                'name' => $title,
                'uri' => $link,
                'picture' => $picture,
                'website_id' => $website_id,
                'attribute_id' => $attribute_id,
                'reg' => $regs,
                'description' => $desc,
                'additional' => $combine,

            ]);
        }

        $str = $Scholarship->description . $Scholarship->additional;
        if (
            (!empty($str) && preg_match_all('~\b\d{4}\b\+?~', $str,
$matches))
        ) {
            $today = date("Y");

```

```

$today = substr($today, 0, 2);
$years = array();
foreach($matches[0] as $year){
    $test = substr($year, 0, 2);
    if($test == $today){
        $years[] = $year;
    }
}
if($years == null ){
    $period = date("Y");
}
else{
    $value = max($years);
    $period = $value;
}
} else {
    $period = date("Y");
}

$period = Period::create([
    'scholarship_id' => $Scholarship->id,
    'period' => $period,
]);
} else {

    foreach ($MatchArray as $Match) {
        $scholarship_period = Period::where('scholarship_id',
$scholarship_ID)->first();

        $str = $Match . $combine;
        if (
$matches))
            (!empty($str) && preg_match_all('~\b\d{4}\b\+?~', $str,
$matches))
        ) {
            $today = date("Y");
            $today = substr($today, 0, 2);
            $years = array();
            foreach($matches[0] as $year){
                $test = substr($year, 0, 2);
                if($test == $today){
                    $years[] = $year;
                }
            }
            if($years == null ){
                $period = date("Y");
            }
            else{

```

```

        $value = max($years);
        $period = $value;
    }
} else {
    $period = date("Y");
}

if ($scholarship_period->period < $period) {
    $scholarship_period->period = $period;
    $scholarship_period->save();
}
}

}

});

} elseif ($attribute->website->status == 3) {
    $threshold = Configuration::find(3);
    $threshold = $threshold->value;

    if ($attribute->attribute_id != null) {
        $client = new Client();
        $attribute_id = $attribute->id;
        $attribute_title = $attribute->attribute_title;
        $attribute_desc = $attribute->attribute_description;
        $attribute_picture = $attribute->attribute_picture;
        $attribute_registration = $attribute->attribute_registration;
        $attribute_period = $attribute->attribute_period;
        $attribute_additional1 = $attribute->attribute_additional_1;
        $attribute_additional2 = $attribute->attribute_additional_2;
        $attribute_additional3 = $attribute->attribute_additional_3;
        $attribute_additional4 = $attribute->attribute_additional_4;
        $attribute_additional5 = $attribute->attribute_additional_5;

        $guzzleclient = new \GuzzleHttp\Client(array(
            'timeout' => 90,
            'verify' => false,
        ));
        $client->setClient($guzzleclient);
        $crawler = $client->request('GET', $attribute->url);

        if ($attribute_title != null or $attribute_title != "") {
            $crawler->filter($attribute_title)->each(function ($node) use
(&$title, &$attribute_title) {
                $title = $node->text();
            });
        }
    }
}

```

```

        $Scholarship = Scholarship::where('attribute_id', $attribute-
>attribute_id)->first();
        $Scholarship->name = $title;
        $Scholarship->save();
    }

    if ($attribute_desc != null) {
        $crawler->filter($attribute_desc)->each(function ($node)
use (&$desc, &$attribute_desc) {
            $desc = $desc . '<p>' . $node->text() . '</p>';
        });
        $Scholarship = Scholarship::where('attribute_id', $attribute-
>attribute_id)->first();
        $Scholarship->name = $desc;
        $Scholarship->save();
    }

    if ($attribute_picture != null) {
        $crawler->filter($attribute_picture)->each(function ($node)
use (&$picture, &$attribute_picture) {
            $picture = $node->attr('src');
        });
        $Scholarship = Scholarship::where('attribute_id', $attribute-
>attribute_id)->first();
        $Scholarship->name = $picture;
        $Scholarship->save();
    }

    if ($attribute_registration != null) {
        $crawler->filter($attribute_registration)->each(function
($node) use (&$reg, &$attribute_registration) {
            $reg = $node->attr('href');
        });
        $Scholarship = Scholarship::where('attribute_id', $attribute-
>attribute_id)->first();
        $Scholarship->name = $reg;
        $Scholarship->save();
    }

    $additional1 = "";
    if ($attribute_additional1 != null) {
        $crawler->filter($attribute_additional1)->each(function
($node) use (&$additional1, &$attribute_additional1) {
            $additional1 = $additional1 . '<p>' . $node->text() . '</p>';
        });
    }

```

```

        $additional2 = "";
        if ($attribute_additional2 != null) {
            $crawler->filter($attribute_additional2)->each(function
($node) use (&$additional2, &$attribute_additional2) {
                $additional2 = $additional2 . '<p>' . $node->text() . '</p>';
            });
        }

```

```

        $additional3 = "";
        if ($attribute_additional3 != null) {
            $crawler->filter($attribute_additional3)->each(function
($node) use (&$additional3, &$attribute_additional3) {
                $additional3 = $additional3 . '<p>' . $node->text() . '</p>';
            });
        }

```

```

        $additional4 = "";
        if ($attribute_additional4 != null) {
            $crawler->filter($attribute_additional4)->each(function
($node) use (&$additional4, &$attribute_additional4) {
                $attribute_additional4 = $attribute_additional4 . '<p>' .
$node->text() . '</p>';
            });
        }

```

```

        $additional5 = "";
        if ($attribute_additional5 != null) {
            $crawler->filter($attribute_additional5)->each(function
($node) use (&$additional5, &$attribute_additional5) {
                $additional5 = $additional5 . '<p>' . $node->text() . '</p>';
            });
        }

```

```

        $Scholarship = Scholarship::where('attribute_id', $attribute-
>attribute_id)->first();
        $combine = $Scholarship->additional . '<p>' . $additional1 .
'</p>' . '<p>' . $additional2 . '</p>' . '<p>' . $additional3 . '</p>' . '<p>' .
$additional4 . '</p>' . '<p>' . $additional5 . '</p>';
        $Scholarship->additional = $combine;
        $Scholarship->save();

```

```

        $str = $Scholarship->description . $Scholarship->additional;
        if (
            (!empty($str) && preg_match_all('~\b\d{4}\b\+?~', $str,
$matches))
        ) {
            $today = date("Y");

```

```

$today = substr($today, 0, 2);
$years = array();
foreach($matches[0] as $year){
    $test = substr($year, 0, 2);
    if($test == $today){
        $years[] = $year;
    }
}
if($years == null ){
    $period = date("Y");
}
else{
    $value = max($years);
    $period = $value;
}
} else {
    $period = date("Y");
}
$periods = Period::where('scholarship_id', $Scholarship->id)-
>first();

if ($period > $periods->period) {
    $periods->period = $period;
    $periods->save();
}
} else {
    $dec_list = array();
    $web_matches = array();

    $client = new Client();
    $attribute_id = $attribute->id;
    $attribute_title = $attribute->attribute_title;
    $attribute_desc = $attribute->attribute_description;
    $attribute_picture = $attribute->attribute_picture;
    $attribute_registration = $attribute->attribute_registration;
    $attribute_period = $attribute->attribute_period;
    $attribute_additional1 = $attribute->attribute_additional_1;
    $attribute_additional2 = $attribute->attribute_additional_2;
    $attribute_additional3 = $attribute->attribute_additional_3;
    $attribute_additional4 = $attribute->attribute_additional_4;
    $attribute_additional5 = $attribute->attribute_additional_5;

    $guzzleclient = new \GuzzleHttp\Client(array(
        'timeout' => 90,
        'verify' => false,
    ));
    $client->setClient($guzzleclient);

```

```

$crawler = $client->request('GET', $attribute->url);

$title = "";
if ($attribute_title != null or $attribute_title != "") {
    $crawler->filter($attribute_title)->each(function ($node) use
(&$title, &$attribute_title) {
        $title = $node->text();
    });
}

$desc = "";
if ($attribute_desc != null) {
    $crawler->filter($attribute_desc)->each(function ($node)
use (&$desc, &$attribute_desc) {
        $desc = $desc . '<p>' . $node->text() . '</p>';
    });
}

$picture = "";
if ($attribute_picture != null) {
    $crawler->filter($attribute_picture)->each(function ($node)
use (&$picture, &$attribute_picture) {
        $picture = $node->attr('src');
    });
}

$reg = "";
if ($attribute_registration != null) {
    $crawler->filter($attribute_registration)->each(function
($node) use (&$reg, &$attribute_registration) {
        $text = $node->text();
        if ($text == "Daftar Online") {
            $reg = $node->attr('href');
        }
    });
}

$additional1 = "";
if ($attribute_additional1 != null) {
    $crawler->filter($attribute_additional1)->each(function
($node) use (&$additional1, &$attribute_additional1) {
        $additional1 = $additional1 . '<p>' . $node->text() . '</p>';
    });
}

$additional2 = "";
if ($attribute_additional2 != null) {

```

```

        $crawler->filter($attribute_additional2)->each(function
($node) use (&$additional2, &$attribute_additional2) {
            $additional2 = $additional2 . '<p>' . $node->text() . '</p>';
        });
    }

```

```

    $additional3 = "";
    if ($attribute_additional3 != null) {
        $crawler->filter($attribute_additional3)->each(function
($node) use (&$additional3, &$attribute_additional3) {
            $additional3 = $additional3 . '<p>' . $node->text() . '</p>';
        });
    }

```

```

    $additional4 = "";
    if ($attribute_additional4 != null) {
        $crawler->filter($attribute_additional4)->each(function
($node) use (&$additional4, &$attribute_additional4) {
            $attribute_additional4 = $attribute_additional4 . '<p>' .
$node->text() . '</p>';
        });
    }

```

```

    $additional5 = "";
    if ($attribute_additional5 != null) {
        $crawler->filter($attribute_additional5)->each(function
($node) use (&$additional5, &$attribute_additional5) {
            $additional5 = $additional5 . '<p>' . $node->text() . '</p>';
        });
    }

```

```

    $combine = '<p>' . $additional1 . '</p>' . '<p>' . $additional2 .
'</p>' . '<p>' . $additional3 . '</p>' . '<p>' . $additional4 . '</p>' . '<p>' .
$additional5 . '</p>';

```

```

    $Scholarships = Scholarship::All();
    foreach ($Scholarships as $Scholarship) {
        $comparison = new
\Atomescrochus\StringSimilarities\Compare();
        $jaroWinkler = $comparison->jaroWinkler($Scholarship-
>description, $desc);

```

```

        $dec_list[] = $Scholarship->description;
        if ($jaroWinkler >= $threshold) {
            $scholarship_ID = $Scholarship->id;
            $web_matches[] = $desc;
        }
    }

```

```

    }

    $MatchArray = array_unique($web_matches);

    if ($MatchArray == null) {
        if ($attribute_picture == null or $attribute_picture == "") {
            $path = '/img/no_picture/' . rand(1, 2) . '.jpg';
            $Scholarship = Scholarship::create([
                'name' => $title,
                'uri' => $attribute->url,
                // 'picture' => $path,
                'website_id' => $attribute->website_id,
                'attribute_id' => $attribute->id,
                'description' => $desc,
                'reg' => $reg,
                'additional' => $combine,

            ]);
        } else {
            $Scholarship = Scholarship::create([
                'name' => $title,
                'uri' => $attribute->url,
                'picture' => $picture,
                'website_id' => $attribute->website_id,
                'attribute_id' => $attribute->id,
                'description' => $desc,
                'reg' => $reg,
                'additional' => $combine,

            ]);
        }
    }

    $str = $Scholarship->description . $Scholarship->additional;
    if (
        (!empty($str) && preg_match_all('~\b\d{4}\b\+?~', $str,
$matches))
    ) {
        $today = date("Y");
        $today = substr($today, 0, 2);
        $years = array();
        foreach($matches[0] as $year){
            $test = substr($year, 0, 2);
            if($test == $today){
                $years[] = $year;
            }
        }
        if($years == null ){

```

```

        $period = date("Y");
    }
    else{
        $value = max($years);
        $period = $value;
    }
} else {
    $period = date("Y");
}

$period = Period::create([
    'scholarship_id' => $Scholarship->id,
    'period' => $period,
]);
} else {
    foreach ($MatchArray as $Match) {
        $scholarship_period = Period::where('scholarship_id',
$scholarship_ID)->first();
        $str = $Match . $combine;
        if (
$matches))
            (!empty($str) && preg_match_all('~\b\d{4}\b\+?~', $str,
$matches))
        ) {
            $today = date("Y");
            $today = substr($today, 0, 2);
            $years = array();
            foreach($matches[0] as $year){
                $test = substr($year, 0, 2);
                if($test == $today){
                    $years[] = $year;
                }
            }
            if($years == null ){
                $period = date("Y");
            }
            else{
                $value = max($years);
                $period = $value;
            }
        } else {
            $period = date("Y");
        }

        if ($scholarship_period->period < $period) {
            $scholarship_period->period = $period;
            $scholarship_period->save();
        }
    }
}

```

```

        }
    }
}

}

}

}

}

}

Log::create([
    'type' => 2,
    'status' => 1,
]);

echo 'Done';

} catch (\Exception $e) {
    if (strpos($e, 'timed out') !== false) {
        Log::create([
            'type' => 2,
            'status' => 0,
            'message' => 'Time Out, Maybe Website Down or Your connection
get trouble',
        ]);
        echo 'Timeout '.<br>'.$e;

    }
    else{
        Log::create([
            'type' => 2,
            'status' => 0,
            'message' => 'Unknow Error please Check Scrap Website
Manually',
        ]);

        echo 'Error - '.$e;

    }

}

}

}
}

```