

DAFTAR ISI

| | |
|--|------|
| HALAMAN PERNYATAAN KEASLIAN | vi |
| LEMBAR PENGESAHAN TUGAS AKHIR | vii |
| HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS | viii |
| KATA PENGANTAR | ix |
| ABSTRAK | xi |
| DAFTAR ISI..... | xii |
| DAFTAR TABEL..... | xiv |
| DAFTAR GAMBAR | xv |
| BAB 1 PENDAHULUAN..... | 16 |
| 1.1 Latar Belakang | 16 |
| 1.2 Identifikasi Masalah | 17 |
| 1.3 Tujuan Tugas Akhir..... | 18 |
| 1.4 Manfaat Tugas Akhir..... | 18 |
| 1.5 Lingkup Tugas Akhir | 18 |
| 1.6 Kerangka Berpikir | 19 |
| 1.7 Sistematika Penulisan Tugas Akhir..... | 21 |
| BAB 2 TINJAUAN PUSTAKA..... | 23 |
| 2.1 Arsip, Dokumen Elektronik, dan Arsip Elektronik..... | 23 |
| 2.1.1 Sistem Manajemen Arsip dan Dokumen Elektronik | 23 |
| 2.1.2 Klasifikasi Arsip Fasilitatif Kementerian Perdagangan..... | 24 |
| 2.2 Multimedia | 25 |
| 2.2.1 Format File | 26 |
| 2.2.2 Kompresi Data..... | 26 |
| 2.3 Analisis PIECES..... | 27 |
| 2.4 Apache Solr | 28 |
| 2.4.1 <i>Inverted Index</i> | 29 |
| 2.5 <i>Extreme Programming (XP)</i> | 29 |
| 2.6 <i>Unified Modeling Language (UML)</i> | 31 |
| 2.7 Teknologi Pendukung Sistem yang Digunakan | 38 |
| 2.7.1 <i>PHP</i> | 38 |

| | | |
|----------------------------|-------------------------------------|----|
| 2.7.2 | <i>Laravel</i> | 38 |
| 2.7.3 | <i>MySQL</i> | 38 |
| 2.8 | <i>Black Box Testing</i> | 39 |
| BAB 3 METODE | | 40 |
| 3.1 | Rencana Penelitian | 40 |
| 3.2 | Gambaran Sistem yang Berjalan | 41 |
| 3.3 | Usulan Sistem..... | 47 |
| 3.4 | Metode Pengembangan Sistem | 48 |
| BAB 4 HASIL DAN PEMBAHASAN | | 50 |
| 4.1 | Analisis Kebutuhan Sistem | 50 |
| 4.1.1 | Perangkat Keras..... | 50 |
| 4.1.2 | Perangkat Lunak..... | 50 |
| 4.1.3 | Kebutuhan Fungsional..... | 50 |
| 4.2 | Pembahasan Apache Solr | 51 |
| 4.3 | Rancangan Sistem | 56 |
| 4.3.1 | <i>Use Case Diagram</i> | 56 |
| 4.3.2 | <i>Class Diagram</i> | 57 |
| 4.3.3 | <i>Activity Diagram</i> | 63 |
| 4.4 | Arsitektur Sistem..... | 69 |
| 4.5 | <i>User Interface</i> | 70 |
| 4.6 | Pengujian Sistem | 76 |
| BAB V KESIMPULAN DAN SARAN | | 81 |
| 5.1 | Kesimpulan..... | 81 |
| 5.2 | Saran..... | 81 |
| DAFTAR PUSTAKA | | 82 |
| LAMPIRAN..... | | 83 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 1 Perbedaan Sistem Manajemen Dokumen Elektronik dan Sistem Manajemen Arsip Elektronik | 24 |
| Tabel 2 Klasifikasi Arsip Keuangan – Pengeluaran Anggaran..... | 25 |
| Tabel 3 Jadwal Perencanaan | 40 |
| Tabel 4 Hasil Analisis PIECES..... | 43 |
| Tabel 6 Data Arsip Keuangan..... | 52 |
| Tabel 7 Deskripsi <i>Class Diagram</i> Arsip | 58 |
| Tabel 8 Deskripsi <i>Class Diagram</i> Klasifikasi..... | 58 |
| Tabel 9 Deskripsi <i>Class Diagram</i> Lokasi | 59 |
| Tabel 10 Deskripsi <i>Class Diagram</i> Users..... | 60 |
| Tabel 11 Deskripsi <i>Class Diagram</i> Roles..... | 60 |
| Tabel 12 Deskripsi <i>Class Diagram</i> Permissions..... | 61 |
| Tabel 13 Deskripsi <i>Class Diagram</i> Bagian..... | 61 |
| Tabel 14 Deskripsi <i>Class Diagram</i> Subbagian | 62 |
| Tabel 15 Deskripsi <i>Class Diagram</i> Password_Resets..... | 63 |
| Tabel 16 Hasil Pengujian Sistem | 76 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 1 Kerangka Berpikir | 19 |
| Gambar 2 <i>Inverted Index</i> | 29 |
| Gambar 3 <i>Extreme Programming</i> | 30 |
| Gambar 4 Model 4+1 View | 32 |
| Gambar 5 <i>Use Case Diagram</i> | 32 |
| Gambar 6 Package Diagram..... | 33 |
| Gambar 7 Class Diagram | 34 |
| Gambar 8 Deployment Diagram | 35 |
| Gambar 9 Activity Diagram..... | 36 |
| Gambar 10 Sequence Diagram..... | 37 |
| Gambar 11 Activity Diagram – Proses Penyimpanan Arsip Elektronik..... | 41 |
| Gambar 12 Activity Diagram Proses Pencarian Arsip Elektronik..... | 42 |
| Gambar 13 Usulan Sistem – Penyimpanan Arsip Elektronik | 47 |
| Gambar 14 Function Post to SOLR | 51 |
| Gambar 15 Hasil Ekstraksi File <i>Function Post to Solr</i> | 53 |
| Gambar 16 <i>Function Query to Solr</i> | 54 |
| Gambar 17 Hasil Pencarian..... | 55 |
| Gambar 18 Konten Arsip Hasil Pencarian | 55 |
| Gambar 19 <i>Use Case Diagram</i> Sistem Usulan..... | 56 |
| Gambar 20 <i>Class Diagram</i> Sistem Usulan | 57 |
| Gambar 21 <i>Activity Diagram Login</i> | 63 |
| Gambar 22 <i>Activity Diagram Logout</i> | 64 |
| Gambar 23 <i>Activity Diagram List Arsip</i> | 65 |
| Gambar 24 <i>Activity Diagram Input Data Arsip</i> | 66 |
| Gambar 25 <i>Activity Diagram Cari Data Arsip</i> | 67 |
| Gambar 26 <i>Activity Diagram Input Data User</i> | 68 |
| Gambar 27 Arsitektur Sistem..... | 69 |
| Gambar 28 <i>User Interface</i> Halaman <i>Login</i> | 70 |
| Gambar 29 <i>User Interface</i> Halaman <i>Dashboard</i> | 70 |
| Gambar 30 <i>User Interface</i> Halaman List Arsip | 71 |
| Gambar 31 <i>User Interface</i> Halaman Input Data Arsip | 72 |
| Gambar 32 <i>User Interface</i> Halaman Laporan..... | 72 |
| Gambar 33 <i>User Interface</i> Halaman Klasifikasi..... | 73 |
| Gambar 34 <i>User Interface</i> Halaman Bagian..... | 74 |
| Gambar 35 <i>User Interface</i> Halaman Subbagian | 75 |
| Gambar 36 <i>User Interface</i> Halaman <i>Manage User</i> | 75 |
| Gambar 37 <i>User Interface</i> Halaman <i>Manage Roles</i> | 76 |

LAMPIRAN

Lampiran 1 Kode Aplikasi

| | |
|---|----------------|
| Nama file | Controller.php |
| <pre><?php namespace App\Http\Controllers; use Illuminate\Foundation\Auth\Access\AuthorizesRequests; use Illuminate\Foundation\Bus\DispatchesJobs; use Illuminate\Foundation\Validation\ValidatesRequests; use Illuminate\Routing\Controller as BaseController; class Controller extends BaseController { use AuthorizesRequests, DispatchesJobs, ValidatesRequests; }</pre> | |

| | |
|--|---------------------------|
| Nama file | FilemanagerController.php |
| <pre><?php namespace App\Http\Controllers; use App\Http\Controllers\Controller; use Illuminate\Http\Request; use App\Models\FileManager; use Auth; use Helpers; use Validator; use Session; class FilemanagerController extends Controller { /** * Create a new controller instance. * * @return void */ public function __construct() { } /** * Show the application dashboard. * * @return \Illuminate\Contracts\Support\Renderable */ public function go(Request \$request) { \$message = ""; \$result_validation = true; \$filename = \$request->filename; \$caption = \$request->caption; \$config = json_decode(\$request->config); \$config_original = \$request->config; \$container = \$config->container; \$container_image = \$config->container_image; \$crop = \$config->crop; \$validation = \$config->validation; \$editor = \$config->editor;</pre> | |

```

$config_name = $config->config_name;
$file_manager_id = $request->file_manager_id;
$file_manager_id_name = $config->file_manager_id_name;
$custom_caption_name = $config->custom_caption_name;
$custom_caption = $config->custom_caption;
$key = $config->key;
$path = $request->path;
$editor_id = "";
if(isset($config->editor_id)){
    $editor_id = $config->editor_id;
}

/*OTHER IMAGE*/
$other_image = false;
$container_caption = "";
if(isset($config->other_files){
    $other_image = true;
}
if(isset($config->container_caption)){
    $container_caption = $config->container_caption;
}

$komponen = array();
$width = array();
$height = array();

//disini lanjut kondisi append

if(!$editor){

    //manual crop
    if(isset($crop->status) && $crop->status){
        foreach($crop->data as $crop_key => $crop_value){
            $crop_value->segment = str_replace("/", "garing", $crop_value-
>segment);

            $komponen[] = $crop_value->segment;
            $width[] = $crop_value->width;
            $height[] = $crop_value->height;
        }
    }

    //validation
    list($validation_width, $validation_height, $type, $attr) =
getimagesize(env('APP_UPLOAD').$path.$filename);
    if(!empty($validation->status)){
        if($validation->status){
            if($validation->data->orientation=="landscape"){
                if($validation_height > $validation_width){
                    $result_validation = false;
                }
            }
            elseif($validation->data->orientation=="portrait"){
                if($validation_width > $validation_height){
                    $result_validation = false;
                }
            }
            elseif($validation->data->orientation=="square"){
                if($validation_width != $validation_height){
                    $result_validation = false;
                }
            }
        }

        if(!$result_validation){
            $message = "Pilih foto dengan orientasi ".$validation-
>data->orientation;
        }

        if(!empty($validation->data->min_resolution)){
            if($validation_width < $validation->data-
>min_resolution->width || $validation_height < $validation->data-
>min_resolution->height){
                $result_validation = false;
            }
        }
    }
}

```

```

                $message .= ", Pilih foto dengan minimal ukuran,
width : ".$validation->data->min_resolution->width." dan height : ".$validation-
>data->min_resolution->height;
            }
        }
    }
}
$type = (isset($config->type))? $config->type:'image';

$result = json_encode(array(
    'container' => $container,
    'container_caption' => $container_caption,
    'result_validation' => $result_validation,
    'editor' => $editor,
    'status' => 'ok',
    'message' => $message,
    'other_image' => $other_image,
    'editor_id' => $editor_id,
    'html' => view('segment.file-manager-
go',compact('config','path','key','custom_caption_name','custom_caption','file_m
anager_id_name','file_manager_id','config_name','config_original','filename','cr
op','validation','komponen','width','height','editor','caption','container_image
','type'))->render(),
    'html_caption' => view('segment.file-manager-go-
caption',compact('custom_caption_name','caption','custom_caption'))->render()
));

echo $result;
}

public function index()
{
    return view('segment.file-manager-all');
}

public function get(Request $request)
{
    $data = FileManager::Data($request->key,$request->type,$request-
>order,$request->page);
    $order = $request->order;
    $key = $request->key;
    $request->config = htmlspecialchars_decode($request->config);
    $config = json_decode($request->config);

    if($request->type=='image'){
        $page = 'segment.file-manager-image';
    }
    elseif($request->type=='video'){
        $page = 'segment.file-manager-video';
    }
    elseif($request->type=='others'){
        $page = 'segment.file-manager-others';
    }

    return view($page ,compact('data','order','key','config'));
}

public function add(Request $request)
{
    $config = json_encode(json_decode($request->config));
    $curent_config = json_decode($config);
    $type = (isset($curent_config->type))? $curent_config->type :
'image';

    return view('segment.file-manager-add-ajax',compact('config', 'type'));
}

public function edit(Request $request)
{
    $id = $request->id;
    $type = $request->type;
    $config = json_encode(json_decode($request->config));
    $data = FileManager::edit($request->id);

```

```

        return view('segment.file-manager-edit-
ajax',compact('data','id','type','config'));
    }

    public function update(Request $request)
    {
        //update data
        $data = FileManager::find($request->id);
        $type = $request->type;

if($type=='image'){ $type=1;}elseif($type=='video'){ $type=2;}elseif($type=='other
s'){ $type=3;}
        $data->type = $type;
        $data->title = $request->title;
        $data->author = $request->author;
        $data->caption = $request->caption;
        $data->note = $request->note;
        $data->share = $request->share;

        $valid = true;
        // if($request->file('upfile') != "")
        // {
        //     $validator = Validator::make($request->all(), [
        //         'upfile' =>
'file|max:'.(config('constant.MAX_SIZE_IMAGE_UPLOAD')*1000),
        //     ]);

        //     if ($validator->fails()) {
        //         $valid = false;
        //     }
        //     else{
        //         //cek resolusi

        //         $path = date('Y').'/'.date('m').'/'.date('d').'/';
        //         $destinationPath =
config('constant.ACTUAL_PATH_ASSETS').'/'. $path;
        //         $filename = md5(time()).'.'. $request->file('upfile')-
>getClientOriginalExtension();
        //         if(!$request->file('upfile')->move($destinationPath,
$filename)){
        //             $valid = false;
        //         }
        //         else{
        //             $data->filename = $filename;
        //             $data->path = $path;
        //             if($request->watermark==1)
        //                 {
Helpers::watermark($destinationPath,$filename,$request->watermark_position);
        //                 }
        //             }
        //         }else{
        //             echo 'Failed Upload File';
        //         }
        //     }

        if($valid){
            $data->save();
            echo 'ok';
        }
        else{
            echo 'Internal Server Error';
        }
    }

    public function store(Request $request)
    {
        if($request->file('upfile') != "")
        {
            //validate resolution
            $path = date('Y').'/'.date('m').'/'.date('d').'/';
            $destinationPath = env('APP_UPLOAD').'/'. $path;

```



```

        $filename =
substr(md5(time()),0,5).'_' . strtolower(str_replace(' ','_', __slug
($request->title))).'.'.strtolower($request->file('upfile')-
>getClientOriginalExtension());
        if($request->file('upfile')->move($destinationPath,
$filename)){
            $type = $request->type;
if($type=='image'){ $type=1;}elseif($type=='video'){ $type=2;}elseif($type=='other
s'){ $type=3;}

            $save = new FileManager();
            $save->type = $type;
            $save->title = $request->title;
            $save->author = $request->author;
            $save->caption = $request->caption;
            $save->share = 1;
            $save->filename = $filename;
            $save->path = $path;
            $save->created_id = Auth::id();
            $save->status = 1;
            $save->save();

            if($request->watermark==1)
            {
Helpers::watermark($destinationPath,$filename,$request->watermark_position);
            }

            echo "ok";
        }
        else{
            echo 'Failed Upload File';
        }
    }
    else{
        echo 'Tentukan File Anda : Files : ' . $request->upfile;
    }
}
}
}

```

| Nama file | HomeController.php |
|---|--------------------|
| <pre> <?php namespace App\Http\Controllers; use Illuminate\Http\Request; class HomeController extends Controller { /** * Create a new controller instance. * * @return void */ public function __construct() { \$this->middleware('auth'); } /** * Show the application dashboard. * * @return \Illuminate\Contracts\Support\Renderable */ public function index() { return view('home'); } } </pre> | |

| | |
|---|-----------|
| Nama file | Arsip.php |
| <pre> <?php namespace App\Models; use Illuminate\Database\Eloquent\Factories\HasFactory; use Illuminate\Database\Eloquent\Model; class Arsip extends Model { use HasFactory; /** * The table associated with the model. * * @var string */ protected \$table = 'arsip'; /** * The primary key associated with the table. * * @var string */ protected \$primaryKey = 'id'; /** * The attributes that are mass assignable. * * @var array */ protected \$fillable = ['nomor', 'created_at', 'updated_at', 'created_by', 'status']; public function ScopeFilter(\$query, array \$filter) { if (isset(\$filter['search']) ? \$filter['search'] : false) { return \$query->where('arsip.nomor', 'LIKE', '%' . \$filter['search'] . '%'); } } } </pre> | |

| | |
|--|------------|
| Nama file | Bagian.php |
| <pre> <?php namespace App\Models; use Illuminate\Database\Eloquent\Factories\HasFactory; use Illuminate\Database\Eloquent\Model; class Bagian extends Model { use HasFactory; /** * The table associated with the model. * * @var string */ protected \$table = 'bagian'; } </pre> | |

```

/**
 * The primary key associated with the table.
 *
 * @var string
 */
protected $primaryKey = 'id';

/**
 * The attributes that are mass assignable.
 *
 * @var array
 */
protected $fillable = [
    'name',
    'created_at',
    'updated_at',
    'created_by',
    'status'
];

public function ScopeFilter($query, array $filter)
{
    if (isset($filter['search']) ? $filter['search'] : false) {
        return $query->where('bagian.name', 'LIKE', '%' . $filter['search']
        . '%');
    }
}
}

```

| Nama file | Klasifikasi.php |
|--|-----------------|
| <pre> <?php namespace App\Models; use Illuminate\Database\Eloquent\Factories\HasFactory; use Illuminate\Database\Eloquent\Model; class Klasifikasi extends Model { use HasFactory; /** * The table associated with the model. * * @var string */ protected \$table = 'klasifikasi'; /** * The primary key associated with the table. * * @var string */ protected \$primaryKey = 'id'; /** * The attributes that are mass assignable. * * @var array */ protected \$fillable = ['kode', 'name', 'retensi', 'created_at', 'updated_at', 'created_by', 'status']; public function ScopeFilter(\$query, array \$filter) </pre> | |

```

    {
        if (isset($filter['search']) ? $filter['search'] : false) {
            return $query->where('klasifikasi.kode', 'LIKE', '%' .
            $filter['search'] . '%')->orWhere('klasifikasi.name', 'LIKE', '%' .
            $filter['search'] . '%');
        }
    }
}

```

| | |
|-----------|------------|
| Nama file | Lokasi.php |
|-----------|------------|

```

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Lokasi extends Model
{
    use HasFactory;
    /**
     * The table associated with the model.
     *
     * @var string
     */
    protected $table = 'lokasi';

    /**
     * The primary key associated with the table.
     *
     * @var string
     */
    protected $primaryKey = 'id';

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'gedung',
        'lantai',
        'ruangan',
        'no_lemari',
        'created_at',
        'updated_at',
        'created_by',
        'status'
    ];

    public function ScopeFilter($query, array $filter)
    {
        if (isset($filter['search']) ? $filter['search'] : false) {
            return $query->where('lokasi.ruangan', 'LIKE', '%' .
            $filter['search'] . '%');
        }
    }
}

```

| | |
|-----------|---------------|
| Nama file | Subbagian.php |
|-----------|---------------|

```

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

```

```

class Subbagian extends Model
{
    use HasFactory;
    /**
     * The table associated with the model.
     *
     * @var string
     */
    protected $table = 'subbagian';

    /**
     * The primary key associated with the table.
     *
     * @var string
     */
    protected $primaryKey = 'id';

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name',
        'created_at',
        'updated_at',
        'created_by',
        'status'
    ];

    public function ScopeFilter($query, array $filter)
    {
        if (isset($filter['search']) ? $filter['search'] : false) {
            return $query->where('bagian.name', 'LIKE', '%' . $filter['search']
                . '%');
        }
    }
}

```

| Nama file | User.php |
|--|----------|
| <pre> <?php namespace App\Models; use Illuminate\Contracts\Auth\MustVerifyEmail; use Illuminate\Database\Eloquent\Factories\HasFactory; use Illuminate\Foundation\Auth\User as Authenticatable; use Illuminate\Notifications\Notifiable; use Spatie\Permission\Traits\HasRoles; class User extends Authenticatable { use HasFactory, Notifiable, HasRoles; /** * The table associated with the model. * * @var string */ protected \$table = 'users'; /** * The attributes that are mass assignable. * * @var array */ protected \$fillable = ['name', 'email', </pre> | |

```
        'image',
        'password',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    public function ScopeFilter($query, array $filter)
    {
        if (isset($filter['search']) ? $filter['search'] : false) {
            return $query->where('users.name', 'LIKE', '%' . $filter['search'] .
            '%');
        }
    }
}
```