

# Pengembangan *Backend* Sistem Kearsipan Dokumen menggunakan Framework Laravel di CV. Nakula Sadewa

Arief Hidayatullah<sup>1</sup>, Dianni Yusuf<sup>2</sup>

Jurusan Bisnis dan Informatika, Politeknik Negeri Banyuwangi, Kab. Banyuwangi, Indonesia<sup>1,2</sup>

Email: [ariefhidayatullah221@gmail.com](mailto:ariefhidayatullah221@gmail.com), [dianniyusuf@gmail.com](mailto:dianniyusuf@gmail.com)

## Abstract

*Archives are defined as the storage of important documents. These letters must be stored using a certain system so that it is easy to manage and access later. This study identifies and analyzes the problems that arise in document management within a technology company, particularly CV Nakula Sadewa. The main problem is changing the use of conventional methods in archive management, which often leads to confusion when accessing files stored in Google Drive. To address this issue, the study proposes a web-based application system. The process involves planning analysis, system design using entity relationship diagrams, development using Laravel and React.js frameworks, and project completion with delivery to CV Nakula Sadewa. System testing was carried out using unit test methods with the Postman tool, which showed that implementing the program code was successful with satisfactory responses. Therefore, this solution is expected to enhance efficiency in managing archives in the company.*

**Keywords:** Archiving, CV Nakula Sadewa, Management, Unit Test, Laravel, Postman

## Abstrak

Arsip didefinisikan sebagai simpanan surat – surat penting. Surat tersebut harus disimpan dengan menggunakan sistem tertentu sehingga mudah dikelola dan digunakan Kembali di lain waktu. Penelitian ini mengidentifikasi dan menganalisis masalah yang muncul dalam pengelolaan kearsipan di dalam sebuah perusahaan teknologi, khususnya CV Nakula Sadewa. Permasalahan utama terkait dengan penggunaan metode konvensional dalam pengelolaan arsip, yang sering kali menyebabkan kebingungan dalam mengakses berkas yang tersimpan di dalam Google Drive. Untuk mengatasi masalah ini, penelitian ini mengusulkan sebuah sistem aplikasi berbasis web. Tahapannya meliputi analisis perencanaan, perancangan sistem dengan menggunakan diagram entitas hubungan, pengembangan menggunakan framework Laravel dan React.js, serta penyelesaian proyek dengan penyampaian kepada pihak CV Nakula Sadewa. Pengujian sistem dilakukan dengan metode unit test menggunakan alat bantu Postman, yang menunjukkan implementasi kode program berhasil dengan respons yang memuaskan. Dengan demikian, solusi ini diharapkan dapat meningkatkan efisiensi dalam pengelolaan kearsipan di perusahaan tersebut.

**Kata Kunci:** Kearsipan, CV Nakula Sadewa, Pengelolaan, Unit Test, Laravel, Postman

## I. PENDAHULUAN

Konsep Arsip dan kearsipan menurut kamus besar Bahasa Indonesia Arsip didefinisikan sebagai simpanan surat – surat penting yang memiliki syarat sebagai berikut: 1. Surat tersebut memiliki sebuah kepentingan bagi Lembaga, Organisasi, Instansi, Perorangan maupun subjek lainnya. 2. Surat tersebut harus disimpan dengan menggunakan sistem tertentu sehingga mudah dikelola dan digunakan Kembali di lain waktu jika memerlukan. Berbeda dengan halnya Arsip, Kearsipan sendiri merupakan sebuah penempatan daripada arsip – arsip dalam tempat penyimpanan yang tertentu, sesuai dengan yang telah ditentukan terlebih dahulu, Sehingga arsip dapat ditemukan dengan mudah dan gampang [1]. Nakula Sadewa, CV. Merupakan Perusahaan yang bergerak dibidang teknologi informasi selama 5 tahun, dan disahkan sebagai badan usaha pada 19 Januari 2015 di Malang, Jawa Timur, Dikukuhkan dihadapan Notaris Benediktur Bosu, SH. Pengalaman yang didukung oleh tenaga muda profesional dari latar belakang, Profesi dan disiplin ilmu yang bervariasi ikut serta berpartisipasi secara elegan dengan profesional dalam pelaksanaan pembangunan

nasional khususnya di bidang Teknologi Informasi. Mengutip dari Website [2].

Adapun proyek yang pernah dikerjakan oleh CV Nakula Sadewa selain End-Que adalah Digital Signage adalah sebuah media yang menyampaikan sebuah informasi melalui display elektronik, Display yang ditampilkan menggunakan teknologi plasma, LCD atau LED, Sehingga digital signage sering digunakan untuk menyampaikan sebuah informasi secara menarik di tempat umum. Informasi yang diberikan dikemas dalam satu arah atau dua arah dengan tampilan yang sangat interaktif, penggunaan Digital Signage lebih efektif 6 ketimbang penyampaian informasi via suara maupun kertas seperti majalah. Sehingga dapat juga menghemat biaya. Digital Signage mempunyai tampilan yang sangat menarik, Sehingga Masyarakat lebih tertarik terhadap sebuah informasi yang diberikan dan harapannya dapat meningkatkan penjualan. Mesin informasi Digital Signage ini biasanya ditampilkan pada fasilitas umum yang sering dilewati orang lalu Lalang, Instansi pemerintah, maupun usaha Ritel yang lagi berkembang. Mengutip dari Website [3].

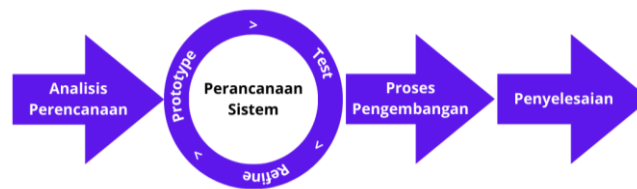
Masalah-masalah yang muncul dalam pengelolaan kearsipan dalam sebuah manajemen pengelolaan kearsipan biasanya terdapat sebuah masalah – masalah yang akan dihasilkan dari proses pengelolaan arsip tersebut, dimana masalah – masalah yang muncul baik secara langsung maupun tidak langsung. Meskipun CV Nakula Sadewa berbasis Perusahaan Teknologi, Segala bentuk kearsipan di sana masih menggunakan konvensional, Terkadang pula, digunakan layanan bernama *Google Drive*, dimana ketika membuka sebuah berkas yang telah dikirim, diperlukan penggunaan tautan yang telah diberikan, dengan hal itu membuat kebingungan Ketika membuka sebuah arsip mana yang akan buka.

Terdapat beberapa penelitian yang telah dilakukan sebelumnya, penelitian yang dilakukan oleh kadek dwi riska santi, dkk tentang sistem informasi pengelolaan kearsipan berbasis *web* kelurahan mulyasari, dengan menggunakan metode penelitian studi literatur, perencanaan, perancangan, pengkodean serta hasil [4]. Penelitian yang dilakukan oleh Andre, dkk 2021 dengan judul sistem informasi manajemen kearsipan berbasis *web*, dengan menggunakan metode pengumpulan data wawancara terhadap pegawai administrasi program studi PTIK, dengan menggunakan metode pengembangan *prototype* [5]. Penelitian terakhir yang dilakukan oleh Riana Suprpto dan Dedy Rahman Prehanto 2020. Melakukan penelitian dengan judul pengembangan sistem manajemen arsip dinamis dalam mendukung tata kelola kearsipan berbasis *web* menggunakan metode SDLC [6].

Dengan observasi lapang dan studi literatur yang telah dilakukan maka penelitian ini menggunakan metode pengembangan RAD (*Rapid Application Development*) dengan pengujian menggunakan *unit test* untuk menguji kode program yang telah dibuat. Harapan dari aplikasi yang dibuat pada penelitian kali ini agar sistem *backend* dapat merespon permintaan dari sisi *frontend* mulai dari permintaan pengelolaan *user*, pengelolaan *file* serta pengelolaan *folder*. Sisi *backend* harus bisa menerima permintaan itu semua.

## II. METODE PENELITIAN

Pendekatan berorientasi objek untuk pengembangan sistem, termasuk metodologi pengembangan dan perangkat lunak. Metode pengembangan perangkat lunak *Rapid Application Development* (RAD) adalah model proses pengembangan perangkat lunak tambahan yang dirancang khusus untuk waktu penyelesaian yang singkat [7]. Metodologi pengembangan perangkat lunak RAD merupakan metodologi yang menggunakan pendekatan berorientasi objek dalam pengembangan sistem, termasuk pengembangan perangkat dan pengembangan perangkat lunak. Implementasi metode ini dilakukan dalam empat tahap: analisis perencanaan, perancangan sistem, proses pengembangan, dan Penyelesaian [8].



Gambar 1. Metode RAD (*Rapid Application Development*)

### a. Analisis Perencanaan

Selama fase ini, pengguna dan penulis berkumpul untuk menyelidiki dan memecahkan masalah yang mereka hadapi saat ini dan memutuskan apa yang diperlukan untuk membuat sistem aplikasi. Sebab, tahap ini merupakan langkah awal keberhasilan pembuatan sistem. Selain itu, kesalahan komunikasi antara pengguna dan penulis dapat dihindari [9].

### b. Perancangan Sistem

Pada tahap ini perancangan sebuah sistem yang peneliti rancang hanya menggunakan *entitas relationship diagram* guna mengetahui relasi data pada bagian *backend* nantinya.

### c. Proses Pengembangan

Proses pengembangan dilakukan menggunakan *framework* laravel sebagai penerima *request* dari bagian *frontend* yang menggunakan *framework* react.js.

### d. Penyelesaian

Pada tahap ini peneliti atau pengembang menyerahkan proyek yang telah dikerjakan ke pada perusaha CV. Nakula Sadewa.

## III. HASIL DAN PEMBAHASAN

### 1. Hasil Penelitian

Saat sebuah tim mengembangkan situs web, data yang ditampilkan diambil, diproses, dan ditampilkan di *backend*, sedangkan di sisi *frontend* adalah pemrosesan dasar dari *backend* itu sendiri. CRUD, membuat data, membaca data, memperbarui data, dan menghapus data. Backend sendiri berfokus pada pembuatan perintah yang menangani semua proses di belakang sistem. Saat bekerja dengan sistem backend itu sendiri maka diperlukan analisis kebutuhan data yang ingin dikelola.

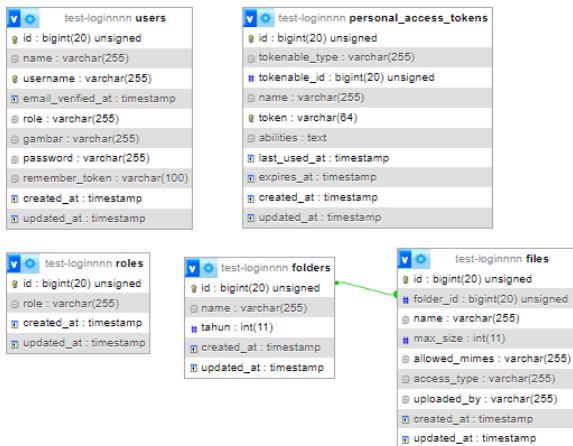
Batasan adalah aturan atau batasan yang mendefinisikan nilai atau data yang dibuat dan disimpan melalui operasi penyisipan, pembaruan, atau penghapusan dalam database. Berdasarkan standar ANSI SQL, batasan dibagi menjadi empat area: 1. Kunci utama (biasa disebut kunci master). Fungsinya untuk memastikan bahwa baris data dalam suatu tabel bersifat unik (berbeda dari baris ke baris). Kunci utama diterapkan pada kolom yang digunakan sebagai berbagai penanda, seperti: B.kolom ID. Kunci utama biasanya digunakan untuk mengenali kolom dan dapat digunakan untuk hubungan antar tabel lainnya. 2. Foreign key berguna untuk mendefinisikan kolom pada suatu tabel yang nilainya

mengarah ke tabel lain. Untuk tabel yang digunakan sebagai referensi, kunci asing harus berupa kunci utama. 3. Unique identik dengan primary key dan digunakan untuk memastikan bahwa data dalam suatu tabel bersifat unik (tidak sama). Perbedaannya adalah Unique dapat berisi nilai null (kosong). 4. Check, fungsi dari Check sendiri adalah untuk membatasi nilai-nilai yang dapat dimasukkan ke dalam kolom-kolom dalam sebuah tabel. Contoh sederhananya adalah kolom gender. Nilainya hanya dapat dibatasi untuk pria dan wanita, kecuali sisi server dapat menolak permintaan tersebut [10].

2. Pembahasan Hasil Penelitian

a. Entitas Relationship Diagram (ERD)

ERD adalah model jaringan yang menggunakan sekumpulan data yang disimpan oleh sistem acak. digunakan untuk merepresentasikan objek data dan relasi yang ada di dalam objek tersebut menggunakan Entitas dan Relasi yang pertama kali diperkenalkan pada tahun 1976 oleh P.P. Chen diperkenalkan [11]. Back-end merupakan suatu tempat dimana berfungsi untuk perintah – perintah yang akan dijalankan pada sebuah aplikasi, Di dalam sebuah pengembangan website yang dikerjakan dan tim, Data yang akan ditampilkan diperoleh dan diproses pada bagian back-end, dan kemudian akan ditampilkan oleh sisi front-end, yang kita ketahui proses dasar dari pada back-end sendiri yaitu CRUD, create data, read data, update data, dan juga delete data, back-end sendiri fokus untuk pembuatan sebuah perintah yang akan menangani segala proses di belakang sistem.



Gambar 2. Entitas Relationship Diagram Pada Sistem Kearsipan

Relasi antar table pada perancangan sistem dari proyek yang dibikin berbasis website dimana disini ada 5 tabel yang digunakan, yaitu ada table users, personal\_access\_tokens, roles, folders, dan juga files, yang dapat dilihat pada gambar 1.

b. Implementasi Code

Pada gambar 2 memperlihatkan kode untuk mendapatkan data pengguna. Fungsi kode ini adalah untuk menampilkan

semua data pengguna dan mencari data pengguna berdasarkan nama.

```

1 public function index(Request $request){
2     $keyword = $request->input('name');
3
4
5     $users = $keyword
6     ? User::where('name','LIKE','%'. $keyword. '%')->get()
7     : User::all();
8     return response()->json([
9         'message' => 'menampilkan User',
10        'data' => $users
11    ]);
12 }
    
```

Gambar 3. Code Menampilkan Data User dan Pencarian Data User

```

1 public function updateProfile(Request $request, $userId)
2 {
3     // Get current user
4     $user = User::findOrFail($userId);
5
6     // Validate the data submitted by user
7     $request->validate([
8         'name' => 'required|string|max:255',
9         'gambar' => 'nullable|image|max:1024|img_src_001|img_004',
10        'password' => 'nullable|string|min:8|confirmed',
11    ]);
12
13     // Update user profile if it is provided in the request
14     if ($request->has('name')) {
15         $user->name = $request->input('name');
16     }
17
18     if ($request->has('password')) {
19         $user->password = bcrypt($request->input('password'));
20     }
21
22     if ($request->has('gambar')) {
23         if ($user->gambar) {
24             // Delete old image
25             Storage::delete('public/'.$user->gambar);
26         }
27
28         // Upload new profile image and get its path
29         $gambarPath = $request->file('gambar')->store('public', 'profile_'.$user->id, ['request->file('gambar')->getClientOriginalExtension(), 'public']);
30         $user->gambar = $gambarPath;
31     }
32
33     // Save user to database
34     $user->save();
35
36     return response()->json($user);
37 }
    
```

Gambar 4. Merubah Data User

Pada gambar 3 menunjukkan pengambilan data pengguna berdasarkan id yang ditentukan dan memperbarui data pengguna dengan nilai baru yang diterima melalui \$request->input(). Data pengguna disimpan kembali ke database.

```

1 public function getProfileImage($userId)
2 {
3     $user = User::findOrFail($userId);
4
5     if ($user->gambar) {
6         $gambarPath = Storage::url($user->gambar); // Menggunakan Storage::url()
7         return response()->json(['gambar' => $gambarPath], 200);
8     } else {
9         return response()->json(['message' => 'Gambar tidak ditemukan'], 404);
10    }
11 }
    
```

Gambar 5. Mendapatkan Gambar Profil

Pada gambar 4 menjelaskan kode program untuk proses pengambilan foto profil pengguna menggunakan ID yang diberikan. Setelah mengambil gambar profil pengguna, kode mengembalikan respon dalam bentuk file gambar. Jawaban ini dapat digunakan untuk menampilkan gambar profil pengguna di bagian depan aplikasi. Ini berarti pengguna dapat dengan cepat dan mudah melihat foto profil mereka melalui antarmuka pengguna aplikasi.

Pada gambar 5 ini kode program untuk mendapatkan id pengguna yang kata sandinya sedang diseting ulang dan mengubah kata sandi pengguna ke nilai yang telah ditentukan sebelumnya. Setelah proses reset selesai, akan menerima pesan yang menunjukkan bahwa reset kata sandi

berhasil. Pastikan untuk menyesuaikan nilai kata sandi baru dengan persyaratan keamanan aplikasi.

```

1 public function resetPassword($id)
2 {
3     // Find the user by ID
4     $user = User::find($id);
5
6     if (!$user) {
7         return response()->json(['error' => 'User not found'], 404);
8     }
9
10    // Reset the password to a default value (e.g., 'admin123')
11    $user->password = Hash::make('masuk123');
12
13    // Save the updated user
14    $user->save();
15
16    return response()->json(['message' => 'Password reset successfully'], 200);
17 }

```

Gambar 6. Reset Password Default

Untuk memperbarui data pengguna pada gambar 6, pengguna harus memperbarui informasinya dengan mengisi formulir yang tersedia. Setelah memasukkan data baru ke dalam formulir, dapat menyimpan perubahan dengan menekan tombol Simpan atau fungsi serupa. Proses ini memperbarui informasi pengguna dalam *database* dan memperbarui tampilan profil pengguna dalam aplikasi. Hal ini memungkinkan pengguna untuk dengan mudah mengelola dan memperbarui informasi mereka sesuai kebutuhan.

```

1 public function update(Request $request, $id)
2 {
3
4     $this->validate($request,[
5         'role' => 'nullable',
6         'name' => 'nullable|max:255',
7         'username' => 'nullable|max:255',
8     ]);
9
10    $uprole = User::find($id);
11
12    $uprole->role = $request->role;
13    $uprole->name = $request->name;
14    $uprole->username = $request->username;
15
16    $uprole->save();
17
18    return response()->json([
19        'message' => 'Data Anda Berhasil Di Update',
20        'Success' => true,
21    ]);
22 }

```

Gambar 7. Update Data User

Untuk membuat pengguna baru di sistem, pengguna memasukkan informasi seperti nama, alamat email, dan kata sandi pada formulir pendaftaran yang tersedia. Setelah formulir diisi, pengguna dapat mengklik tombol "Daftar" atau fitur serupa untuk mengirimkan informasi pendaftaran. Sistem akan memproses informasi yang diberikan dan

membuat akun pengguna baru. Setelah akun dibuat, pengguna dapat masuk ke sistem menggunakan alamat email dan kata sandi yang terdaftar. Hal ini memungkinkan pengguna untuk mengakses fitur-fitur dalam sistem tergantung pada peran dan hak akses mereka.

```

1 public function store(Request $request)
2 {
3
4     $validateData = $request->validate([
5         'name' => 'required|max:255',
6         'username' => 'required',
7         'password' => 'required|min:8|max:255',
8         'role' => 'required'
9     ]);
10    $validateData['password'] = bcrypt($validateData['password']);
11    User::create($validateData);
12    return response()->json([
13        'Message' => 'Registration User Successfully',
14        'Success' => true,
15    ]);
16 }

```

Gambar 8. Membuat User Baru

Kode ini menjelaskan secara singkat kode pada gambar 8. Karena kode sebelumnya memiliki fungsi yang sama: mencari data pada kode pengontrol pengguna. Perbedaan kode ini terletak pada fungsi publik `searchName`. Jika dilihat pada baris `try {...}`, maka akan dicari tanggal berdasarkan tahun dan nama. Satu-satunya perbedaan adalah ini. Jika meminta permintaan pencarian nama belakang di depan. Pengguna harus memasukkan nilai data tahun di depan. Untuk nilai data tahun di bagian indeks fungsi publik maka harus melakukan filter data terlebih dahulu atau memasukkan tahun yang ditentukan pada halaman yang ada untuk mencari folder berdasarkan nama.

```

1 public function index(Request $request)
2 {
3     $keyword = $request->input('tahun');
4     $folders = $keyword ? Folder::where('tahun', 'LIKE', '%' . $keyword . '%')
5         ->get()
6         : Folder::all();
7
8     // Mengembalikan hasil sebagai respons JSON
9     return response()->json([
10        'message' => 'Menampilkan Folder',
11        'data' => $folders,
12    ], 200);
13 }
14
15 //search folder by name
16 public function searchName(Request $request,$tahun)
17 {
18     $search = $request->input('name');
19     try {
20         $foldername = Folder::where('tahun',$tahun)
21             ->where('name','LIKE','%$search%')
22             ->get();
23         return response()->json(['hasil' => $foldername]);
24     } catch (\Exception $e) {
25         return response()->json(['error' => $e->getMessage()], 500);
26     }
27 }

```

Gambar 9. Mencari Folder Berdasarkan Tahun dan Nama



```

1 public function createFolder(Request $request)
2 {
3     $this->validate($request, [
4         'folder_name' => 'required|string|max:255',
5     ]);
6
7     $folder = new Folder();
8     $folder->name = $request->input('folder_name');
9     $folder->tahun = date("Y");
10    $folder->save();
11
12    // Membuat folder baru di penyimpanan (public/storage)
13    $folderPath = 'uploads/' . $folder->name . '_' . $folder->tahun;
14    Storage::makeDirectory($folderPath);
15
16    return response()->json([
17        'message' => 'Created successfully folder',
18        'data' => $folder,
19    ]);
20    return redirect()->back()->with('success', 'Folder created successfully');
21 }

```

Gambar 10. Membuat Folder

Pada gambar 9 Untuk membuat folder di dalam sistem, pengguna dapat menggunakan antarmuka aplikasi yang disediakan. Pengguna biasanya melihat opsi "Buat Folder", tombol, atau fungsi serupa yang memungkinkan mereka membuat folder baru. Mengklik opsi ini akan meminta pengguna untuk memasukkan nama folder baru dan, tergantung pada sistem, pengaturan tambahan. Setelah folder dibuat, pengguna dapat melihatnya di daftar folder di sistem dan menggunakannya untuk mengatur file dan informasi lain sesuai kebutuhan. Membuat folder ini memungkinkan pengguna lebih mudah mengelola file dan informasi di sistem mereka. Fitur ini memungkinkan pengguna dengan cepat membuat struktur folder yang sesuai dengan kebutuhannya dan menyimpan file dengan lebih terorganisir. Hal ini memudahkan pengguna untuk mencari dan mengakses file yang dibutuhkan. Oleh karena itu, fitur pembuatan folder ini meningkatkan efisiensi dan produktivitas pengguna saat menggunakan sistem.

```

1 public function renameFolder(Request $request, $id)
2 {
3     $folder = Folder::findOrFail($id);
4     $newFolderName = $request->input('new_folder_name');
5
6     // Ambil tahun dari folder yang lama
7     $tahun = $folder->tahun;
8
9     // Konstruksi nama folder baru dengan tahun
10    $newFolderPath = 'uploads/' . $newFolderName . '_' . $tahun;
11
12    if ($folder->name != $newFolderName) {
13        // Hapus folder lama dari penyimpanan (public/storage)
14        $oldFolderPath = 'uploads/' . $folder->name . '_' . $tahun;
15        if (Storage::exists($oldFolderPath)) {
16            Storage::deleteDirectory($oldFolderPath);
17        }
18    }
19
20    // Update nama folder di database
21    $folder->name = $newFolderName;
22    $folder->save();
23
24    // Buat folder baru di penyimpanan (public/storage)
25    Storage::makeDirectory($newFolderPath);
26 }
27 return response()->json([
28     'message' => 'Folder renamed successfully'
29 ]);
30 return redirect()->route('fileManager')->with('success', 'Folder renamed successfully');
31 }

```

Gambar 11. Merubah Nama Folder

Pada gambar 10 Untuk mengganti nama folder di sistem, pengguna dapat melakukan langkah-langkah berikut

menggunakan antarmuka aplikasi yang disediakan: Pertama, pengguna harus membuka daftar folder atau navigasi. Di sana akan melihat daftar folder yang tersedia di sistem. Pengguna kemudian mencari folder yang ingin mereka ganti namanya dan mengklik kanan atau semacamnya untuk membuka opsi manajemen folder. Di sini pengguna akan menemukan opsi "Ganti nama", dll. Ini akan membuka kotak dialog atau kotak teks di mana pengguna dapat memasukkan nama baru untuk folder tersebut. Setelah pengguna memasukkan nama baru, mereka biasanya perlu mengklik tombol Simpan atau tindakan serupa untuk menyimpan perubahan mereka. Oleh karena itu, pengguna berhasil mengganti nama folder di sistem tanpa menggunakan kode program.

Pada gambar 11 Untuk mengunggah file, pengguna dapat menggunakan fungsionalitas yang disediakan di antarmuka aplikasi. Setelah pengguna memilih file yang ingin diunggah, mereka dapat mengklik tombol "Unggah" atau fungsi serupa untuk memulai proses pengunggahan. Sistem memproses file yang dipilih dan mengunggahnya ke server. Setelah proses selesai, pengguna akan menerima notifikasi atau pesan yang menandakan bahwa pengunggahan file berhasil. Fitur ini memungkinkan pengguna dengan mudah mengunggah file ke sistem untuk disimpan dan dibagikan sesuai kebutuhan.

```

1 public function uploadFile(Request $request, $folderId)
2 {
3     $folder = Folder::findOrFail($folderId);
4
5     // Memeriksa apakah ada file yang diunggah
6     if ($request->hasFile('file')) {
7         $this->validate($request, [
8             'file' => [
9                 'required',
10                'file',
11                'max:20480',
12                'mimes:pdf,jpg,png,jpeg,doc,pptx,xlsx,docx', // Hanya izinkan tipe MIME pdf, jpg, dan png
13            ],
14        ]);
15
16        $file = $request->file('file');
17        $fileName = $file->getClientOriginalName();
18
19        // Mendapatkan peran pengguna tujuan yang didapat dari input form
20        $targetUserRole = $request->input('access_type');
21        $userUpload = $request->input('uploaded_by');
22
23        // Mengatur access_type sesuai dengan peran pengguna
24        if ($targetUserRole == 'all') {
25            $targetUserRole = 'all';
26        }
27
28        // Simpan informasi file ke dalam tabel 'files'
29        $fileModel = new File();
30        $fileModel->folder_id = $folder->id;
31        $fileModel->name = $fileName;
32        $fileModel->max_size = $file->getSize();
33        $fileModel->allowed_mimes = $file->getMimeType();
34        $fileModel->access_type = $targetUserRole;
35        $fileModel->uploaded_by = $userUpload;
36        $fileModel->save();
37
38        // Simpan file di dalam folder yang sesuai di penyimpanan (public/storage)
39        $file->storeAs('uploads/' . $folder->name . '_' . $folder->tahun, $fileName, 'public');
40
41        return response()->json(['message' => 'File uploaded successfully']);
42        return redirect()->back();
43    } else {
44        // Tidak ada file yang diunggah
45        return redirect()->back()->with('error', 'No file uploaded');
46    }
47 }

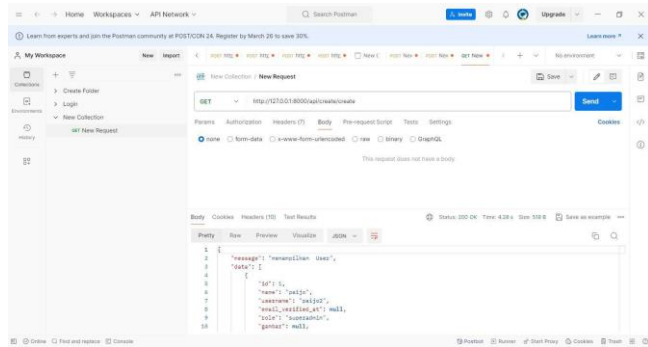
```

Gambar 12. Upload File

**c. Pengujian Kode Program Menggunakan Unit Test dengan Tools Postman**

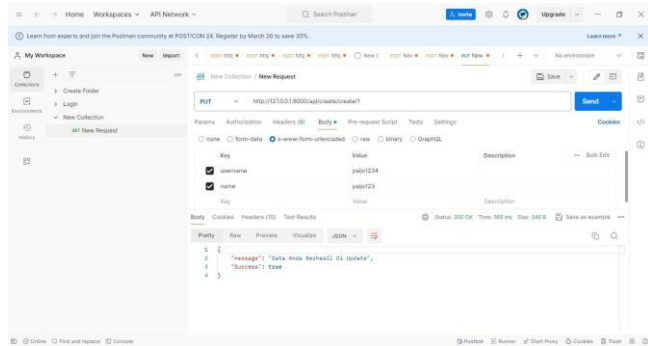
Pengujian yang dilakukan menggunakan unit test untuk menguji kode program yang telah dibuat dengan menggunakan bantuan tools postman untuk melihat respon dari pada kode program backend yang telah diterapkan. Unit

testing ini berfokus pada verifikasi unit terkecil daripada sistem yang dibuat. Pengujian kode program untuk mengetahui kode program yang telah dibuat telah sesuai dengan ekspektasi harapan yang diinginkan berikut akan dijelaskan mengenai pengujian kode program yang telah dibuat.

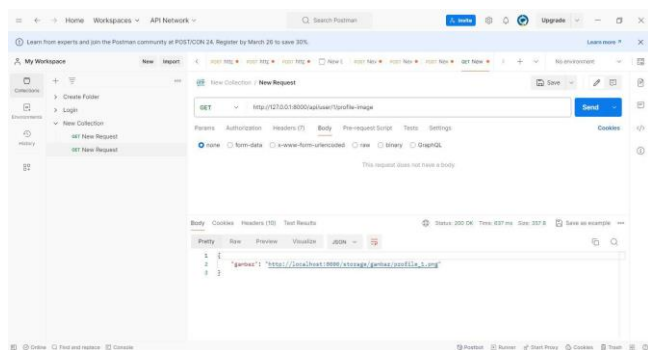


Gambar 13. Hasil Pengujian Mendapatkan Data User

Dapat dilihat pada gambar 12 menunjukkan keberhasilan sebuah kode program yang telah dibuat dengan pengujian untuk mendapatkan data user yang telah ada. Pada gambar 13 dapat dilihat dalam pengujian untuk merubah sebuah data user dimana pada pengujian ini yang dirubah data usernya adalah *value username* dan juga *name* telah berhasil dilakukan dengan response 200OK.

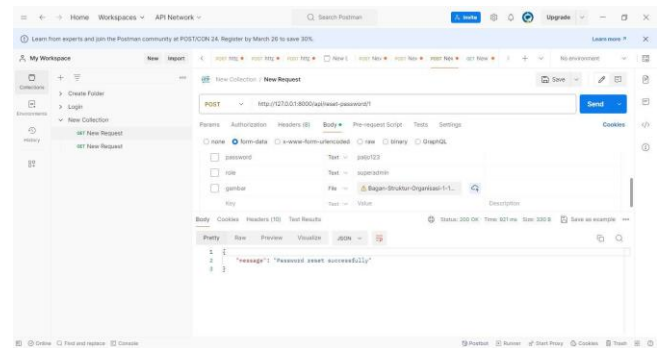


Gambar 14. Hasil Pengujian Update Data User



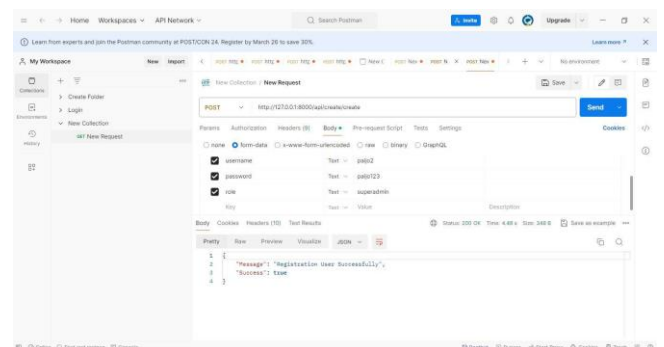
Gambar 15. Hasil Pengujian Mendapatkan Profile-image

Pada gambar 14 dengan pengujian kode program untuk mendapatkan *profile-image* dari *user* disini mencoba menguji untuk mendapatkan gambar *profile-image user* yang mempunyai id 1. Dan hasilnya berhasil menemukan *profile-image* dari *user* yang mempunyai id 1. Gambar 15 mencoba untuk mereset *password user* secara default disini mencoba pengujian mereset *password* dengan *user* yang mempunyai id 1. Dan dapat dilihat pada gambar 15 bahwa *reset password* telah berhasil.

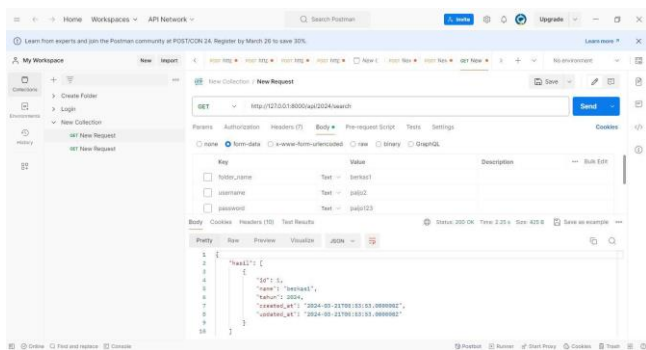


Gambar 16. Hasil Pengujian Reset Password

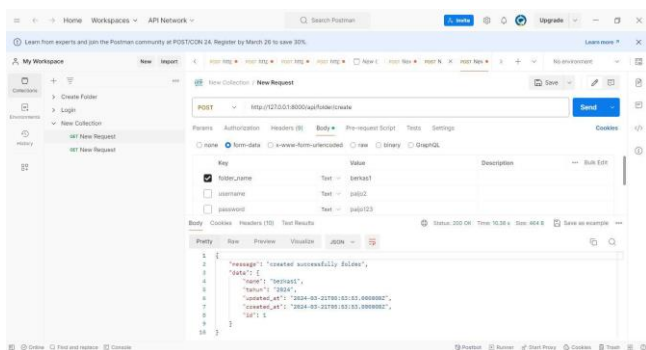
Gambar 16 adalah pengujian dari pembuatan *user* baru, disini peneliti mencoba untuk memasukan data *name*, *username*, *password*, dan *role* dan hasilnya berhasil membuat *user* baru. Pada gambar 17 peneliti mencoba untuk menguji pencarian folder dengan menggunakan *value* tahun hasil dari pencarian menghasilkan data yang sama dengan *value* tahun yang dicari. Gambar 18 adalah hasil dari pengujian untuk membuat sebuah folder baru, dapat dilihat gambar 18 menunjukkan bahwa pembuatan folder baru telah berhasil dan menampilkan data dari folder yang telah dibikin. Gambar 19 adalah hasil dari pengujian untuk merubah nama folder yang telah dibuat, dapat dilihat dari gambar 19 bahwa pengujian menyatakan keberhasilan dari perubahan nama folder yang diuji.



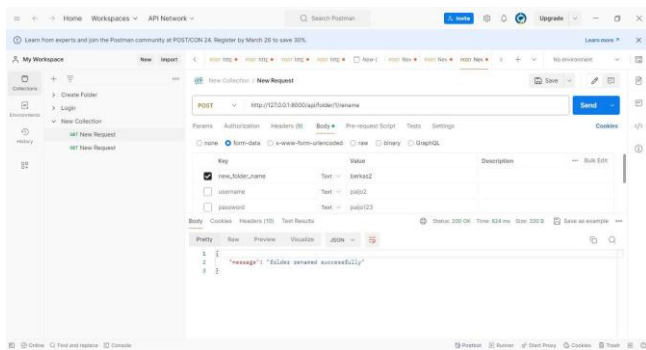
Gambar 17. Hasil Pengujian Pembuatan User Baru



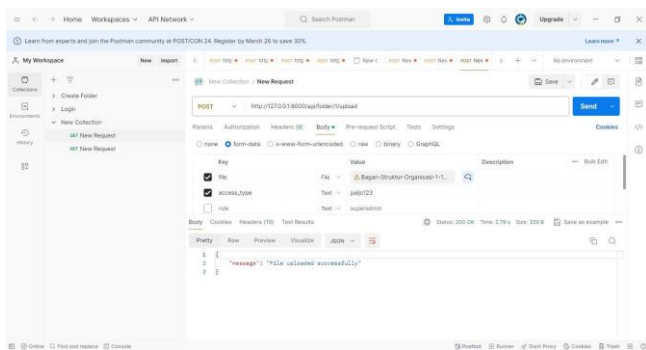
Gambar 18. Hasil Pengujian Mencari Folder



Gambar 19. Hasil Pengujian Pembuatan Folder



Gambar 20. Hasil Pengujian Merubah Nama Folder



Gambar 21. Hasil Pengujian Upload File

Pada gambar 21 peneliti mencoba untuk mengujian *upload file* pada *folder* yang akan diisi *filenya*, pada pengujian ini dapat dilihat dari gambar 21. Bahwa *upload file* yang dilakukan mencoba *upload file* kedalam *folder* yang mempunyai id 1, dan mendapatkan respon keberhasilan dalam *upload file*. Dapat dilihat dari pengujian yang telah dilakukan dengan menggunakan metode *unit test* dengan bantuan *tools postman* memberikan respond 200k yang menandakan suatu kode program yang telah dibuat berhasil diimplementasikan dan berhasil untuk dijalankan.

#### IV. KESIMPULAN

Pada implementasi *backend* yang dilakukan meliputi pembuatan data, pemuatan data, pemutakhiran data, dan penghapusan data. Saat mengerjakan sistem *backend* dimulai dengan tahap analisis kebutuhan data yang akan dikelola, seperti menyiapkan ERD (*Entitas Relationship Diagram*) untuk mengetahui bagaimana keterkaitan data dengan latar belakang. Pengujian dengan menggunakan *Postman* dengan metode *unit test*, telah menunjukkan keberhasilan sehingga sistem berfungsi dengan baik dan sesuai dengan yang diharapkan. Saran pengembangan ke depan bisa dilakukan pengembangan dari fitur laporan sehingga dapat memudahkan pengguna dalam mencetak dokumen dan informasi yang diinginkan.

#### REFERENSI

- [1] Iis Trisnawati, “Permasalahan Dalam Pengelolaan Kearsipan - Lembaga Penelitian dan Pengabdian Masyarakat,” LP2M. Diakses: 4 Januari 2024. [Daring]. Tersedia pada: <https://web.syekhnrjati.ac.id/lp2m/permasalahan-dalam-pengelolaan-kearsipan/>
- [2] CV. Nakula Sadewa, “About – Mesin antrian | Alat survey kepuasan pelanggan.” Diakses: 4 Januari 2024. [Daring]. Tersedia pada: <https://www.nakulasadewa.com/alat-antrian/>
- [3] CV. Nakula Sadewa, “Portofolio: Angkasa Pura Airports Adi Soemarmo Surakarta – Mesin antrian | Alat survey kepuasan pelanggan,” nakulasadewa.com. Diakses: 4 Januari 2024. [Daring]. Tersedia pada: <https://www.nakulasadewa.com/digital-signage-angkasa-pura/>
- [4] Kadek Dwi Riska Santi dan A. Ferico Octaviansyah Pasaribu, “Sistem Informasi Pengelolaan Kearsipan Berbasis Web Kelurahan Mulyasari,” *CHAIN: Journal of Computer Technology, Computer Engineering and Informatics*, vol. 1, no. 4, 2023.
- [5] A. Rumengan, A. Mewengkang, dan D. R. Kaparang, “SISTEM INFORMASI MANAJEMEN KEARSIPAN BERBASIS WEB,” *Eduetik : Jurnal Pendidikan Teknologi Informasi dan Komunikasi*, vol. 1, no. 6, hlm. 709–718, Des 2021, doi: 10.53682/edutik.v1i6.3296.
- [6] R. Suprpto dan D. Rahman Prehanto, “Pengembangan Sistem Informasi Manajemen Arsip Dinamis dalam Mendukung Tata Kelola Kearsipan berbasis Web menggunakan Metode SDLC,” *Journal of Emerging Information System and Business Intelligence (JEISBI)*, vol. 1, no. No 1, 2020, [Daring]. Tersedia pada: <https://siked.unesa.ac.id>.
- [7] T. Wulandari dan D. S. Nurmiati, “RANCANG BANGUN SISTEM PEMESANAN WEDDING ORGANIZER MENGGUNAKAN METODE RAD DI SHOFIA AHMAD WEDDING,” *Jurnal Rekayasa Informasi*, vol. 11, no. 1, 2022.
- [8] Awuy Valentino dan Kambuan Olivia, “481-488,” *INNOVATIVE: Journal Of Social Science Research*, vol. 4, 2024.
- [9] Nurman Hidayat dan Kusuma Hati, “Penerapan Metode Rapid Application Development (RAD) dalam Rancang Bangun Sistem Informasi Rapor Online (SIRALINE),” *Jurnal Sistem Informasi*, vol. 10, no. 1, hlm. 8–17, Feb 2021, doi: 10.51998/jsi.v10i1.352.

- 
- [10] Nurcholish Ahmad, "Membangun database arsip persuratan menggunakan pemrograman PHP dan Mysql ... - Ahmad Nurcholish - Google Buku," CV Jejak. Diakses: 4 Januari 2024. [Daring]. Tersedia pada:  
[https://books.google.co.id/books?hl=id&lr=&id=ofxsDwAAQBAJ&oi=fnd&pg=PP1&dq=perancangan+database+mysql&ots=BmJM65S4qL&sig=\\_tA4vfBtQfOQXq\\_3YxHU3T4qBB8&redir\\_esc=y#v=onepage&q=perancangan%20database%20mysql&f=false](https://books.google.co.id/books?hl=id&lr=&id=ofxsDwAAQBAJ&oi=fnd&pg=PP1&dq=perancangan+database+mysql&ots=BmJM65S4qL&sig=_tA4vfBtQfOQXq_3YxHU3T4qBB8&redir_esc=y#v=onepage&q=perancangan%20database%20mysql&f=false)
- [11] Rahman Khoir Arizal, Mardiyati Sri, dan Nugraha Yusuf, "Perancangan Sistem Informasi Penjualan barang Berupa Alat Music," 2022.