

IMPLEMENTASI DASHBOARD PROYEK DENGAN SISTEM RBAC PADA PT.XYZ

Vrica Gede Penggalih, Esi Putri Silmina

Teknologi Informasi, Universitas Aisyiyah Yogyakarta
Jl. Siliwangi No. 63 Nogotirto, Gamping, Sleman, Yogyakarta. Indonesia
vricagedepenggalih@gmail.com

ABSTRAK

Dashboard proyek adalah kumpulan data utama, informasi kualitatif, dan visual yang merangkum portofolio proyek. Dashboard ini mengambil informasi yang tersebar terkait semua bagian proyek yang bergerak—anggaran, status, transaksi, dll.—dan meringkasnya menjadi bagian yang paling penting. Dashboard Proyek berguna untuk manajemen proyek-proyek serta evaluasi dan observasi sebuah proyek oleh pihak berkepentingan seperti Project Manager. Untuk itu, eksistensi Dashboard Proyek adalah absolut melihat kegunaan yang di berikan, namun pada perusahaan PT.XYZ—sebuah perusahaan teknologi penyedia payment service di Yogyakarta—masih menggunakan spreadsheet (excel) untuk manajemen proyek mereka. Untuk itu, penelitian ini bertujuan untuk meng-implementasi web Dashboard Proyek dengan sistem Role-Based Access Control (RBAC) dengan metode pengembangan *extreme programming*. Pentingnya Role-Based Access Control (RBAC) dalam manajemen proyek telah banyak diakui karena kemampuannya untuk meningkatkan keamanan data, mengurangi risiko akses tidak sah, dan memastikan bahwa hanya pihak yang berwenang yang dapat melakukan tindakan tertentu pada data atau proyek tertentu. Dashboard akan di implementasikan menggunakan Laravel dan Spatie, pustaka dari Laravel untuk implementasi RBAC pada Dashboard Proyek. Berdasarkan pengujian kontrol akses pada sistem RBAC dalam Dashboard dapat menjamin keamanan hak akses *resource*. Dengan adanya Dashboard Proyek ini, kini perusahaan dapat manajemen proyek-proyek mereka lebih baik dan aman.

Kata kunci : Teknologi, Website, Laravel, Spatie, RBAC

1. PENDAHULUAN

Di era digitalisasi saat ini, pengelolaan proyek yang efisien dan terorganisir menjadi salah satu faktor kunci dalam menentukan keberhasilan perusahaan, terutama di industri yang melibatkan banyak proyek simultan. PT. XYZ, sebuah perusahaan yang bergerak di bidang teknologi, menghadapi tantangan dalam manajemen proyek, seperti kurangnya transparansi dalam pelacakan status proyek, kesulitan dalam pembagian peran dan tanggung jawab, serta akses data yang terbatas bagi para pemangku kepentingan. Permasalahan ini dapat memunculkan berbagai permasalahan seperti menghambat pengambilan keputusan yang cepat dan tepat. Sebuah Dashboard Proyek dengan sistem RBAC dapat menyelesaikan permasalahan tersebut.

Dashboard Proyek (akan dipanggil DP seterusnya dalam artikel ini) sangat dibutuhkan di era digital. Pengelolaan proyek yang efektif dan aman menjadi prioritas bagi perusahaan teknologi. DP adalah solusi yang banyak digunakan untuk memantau kemajuan, mengelola data, dan memastikan kolaborasi lintas tim berjalan lancar [4].

Kompleksitas dalam mengatur akses pengguna berdasarkan peran (Role-Based Access Control/RBAC) sayangnya sering kali menjadi tantangan besar, terutama di organisasi dengan struktur hirarki yang kompleks [5].

Salah satu pendekatan terbaik untuk mengatasi tantangan ini adalah dengan mengadopsi sistem RBAC yang mengelola akses pengguna berdasarkan peran dan izin. Pentingnya Role-Based Access Control (RBAC) dalam manajemen proyek telah banyak diakui

karena kemampuannya untuk meningkatkan keamanan data, mengurangi risiko akses tidak sah, dan memastikan bahwa hanya pihak yang berwenang yang dapat melakukan tindakan tertentu pada data atau proyek tertentu [6].

Penelitian ini bertujuan untuk mengembangkan DP dengan fitur RBAC menggunakan teknologi Laravel dan Spatie. Laravel dipilih karena fleksibilitasnya sebagai framework PHP modern, sementara Spatie Laravel Permission dipilih untuk mempermudah implementasi RBAC [8]. PostgreSQL digunakan sebagai basis data untuk mendukung pengelolaan data yang kompleks.

Fitur dari DP yang akan diimplementasikan dalam penelitian adalah:

a. *Import Excel*

Dikarenakan manajemen proyek masih dilakukan dengan *spreadsheet excel* pada perusahaan target penelitian, DP harus bisa melakukan *import excel*. Fitur ini akan diimplementasikan dengan *library* dari *laravel-excel*. DP akan menyimpan data-data dari *excel* tersebut sesuai dengan desain database terlebih dahulu. Di proses ini DP akan melakukan sanitasi, yaitu meng-*ignore* data-data yang tidak di inginkan, contoh adalah data kosong. Setelah itu, DP akan memvisualisasikan data tersebut, sehingga mudah dianalisis dan diobservasi oleh pengguna.

b. Sistem RBAC

Bagian integral dalam DP ini adalah sistem RBAC. Bagaimana pengguna DP mempunyai role tertentu dan role tersebut ber-asosiasi dengan

permissions tertentu. Sehingga hak akses pengguna akan terbatas pada role yang dia miliki. Limitasi hak akses ini adalah bagian penting dalam sistem RBAC. Sistem RBAC akan diimplementasikan dengan *library Spatie*.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Rahmawati dan Setiawan (2020) mengimplementasikan RBAC menggunakan Laravel dan Spatie Laravel Permission dalam aplikasi pengelolaan dokumen, yang mempermudah penerapannya [10].

Widodo dan Astuti (2021) mengintegrasikan RBAC dalam aplikasi web untuk mengelola data akademik di perguruan tinggi, yang terbukti efektif dalam membatasi akses, meningkatkan keamanan data, dan mengurangi kesalahan akses [11]. Penelitian ini memberikan kontribusi baru dengan fokus pada pengelolaan proyek berbasis web untuk PT. XYZ, memanfaatkan Spatie Laravel Permission dan PostgreSQL, yang belum banyak dibahas dalam penelitian sebelumnya.

2.2. XP (Extreme Programming)

XP (Extreme Programming) adalah pendekatan pengembangan perangkat lunak yang termasuk dalam metode agile, yang bertujuan untuk menghasilkan perangkat lunak berkualitas tinggi sekaligus mempermudah proses kerja tim pengembang. Metode ini umumnya diterapkan pada tim kecil dengan jumlah anggota hingga 20 orang, dengan pendekatan yang sangat kolaboratif. Artinya, seluruh anggota tim berbagi tanggung jawab dalam menyelesaikan produk, bukan hanya pemilik atau atasan perusahaan [1].

2.3. Manajemen Proyek

Manajemen proyek adalah proses perencanaan, pengorganisasian, pelaksanaan, dan pengendalian sumber daya untuk mencapai tujuan tertentu dalam waktu dan biaya yang telah ditentukan. Dalam konteks PT. XYZ, manajemen proyek mencakup pengelolaan data proyek, pelacakan progres, serta koordinasi antara tim yang terlibat.

2.4. Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) adalah model kontrol akses yang membatasi akses ke sumber daya dalam sistem berdasarkan peran yang dimiliki oleh pengguna. Dalam RBAC, setiap pengguna diberikan peran tertentu, dan peran tersebut menentukan hak akses yang dimiliki untuk mengakses objek-objek sistem. RBAC memungkinkan pengelolaan hak akses secara lebih efisien karena izin diberikan berdasarkan peran yang terkait dengan kebutuhan fungsional pengguna, bukan ditentukan untuk setiap pengguna secara individu [12].

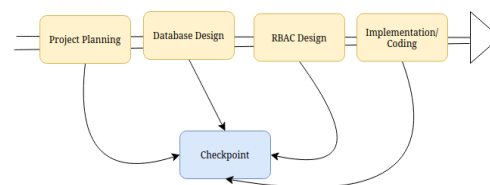
3. METODE PENELITIAN

3.1. Metode Pengembangan

Metode pengembangan yang digunakan ialah metode *extreme programming*. XP (Extreme Programming) adalah kerangka kerja pengembangan perangkat lunak yang gesit (agile) yang membantu menghasilkan perangkat lunak berkualitas tinggi sekaligus mempermudah pekerjaan tim pengembang. Biasanya, metode ini diterapkan pada tim kecil yang terdiri dari hingga 20 anggota, dengan pendekatan yang sangat berorientasi tim. Artinya, tanggung jawab untuk menyelesaikan produk dibagi bersama oleh seluruh anggota tim, bukan hanya pemilik atau atasan perusahaan [1].

3.2. Tahapan Implementasi

Tahapan implementasi DP digambarkan dalam diagram pada gambar 1:



Gambar 1. Diagram tahapan implementasi

a. Project Planning

Project planning adalah langkah awal dari pengembangan *software*. Tujuannya untuk mendefinisikan suatu aturan atau pedoman yang digunakan saat proses pengembangan. Menetapkan anggaran yang realistis dan memastikan bahwa sumber daya yang dibutuhkan, seperti tenaga kerja, material, dan peralatan, tersedia dan digunakan secara efisien [2]. Penjadwalan yang baik serta komunikasi yang efektif sangat penting untuk kelancaran proyek [7].

b. Database Design

Pembuatan (ERD) berguna untuk menggambarkan entitas, atribut, dan hubungan antar entitas dalam sistem [9]. Faktor-faktor seperti ukuran data, tingkat transaksi, dan kebutuhan akses akan mempengaruhi pilihan DBMS yang akan digunakan [3].

c. RBAC Design

RBAC design merupakan tahapan mendefinisikan role-role yang ada dan akses-akses atau *permissions* dari role-role tersebut.

d. Implementation/Coding

Implementasi adalah fase dalam pengembangan perangkat lunak di mana desain yang telah dibuat diubah menjadi kode yang dapat dieksekusi. Proses ini mencakup pengkodean, pengujian awal, dan integrasi sistem untuk memastikan perangkat lunak memenuhi spesifikasi yang ditentukan.

e. Checkpoint

Dalam setiap tahapan, dilakukan sebuah checkpoint dengan Project Manager. Checkpoint dalam proses pengembangan perangkat lunak

adalah titik-titik pemeriksaan atau evaluasi yang digunakan untuk menilai kemajuan, kualitas, dan kesesuaian guna memastikan pengembangan sesuai kriteria dan spesifikasi yang sudah ditetapkan. Pada tahap ini juga dilakukan pengidentifikasian masalah jika ada, dan mengambil tindakan korektif jika diperlukan.

4. HASIL DAN PEMBAHASAN

4.1. Teknologi Digunakan

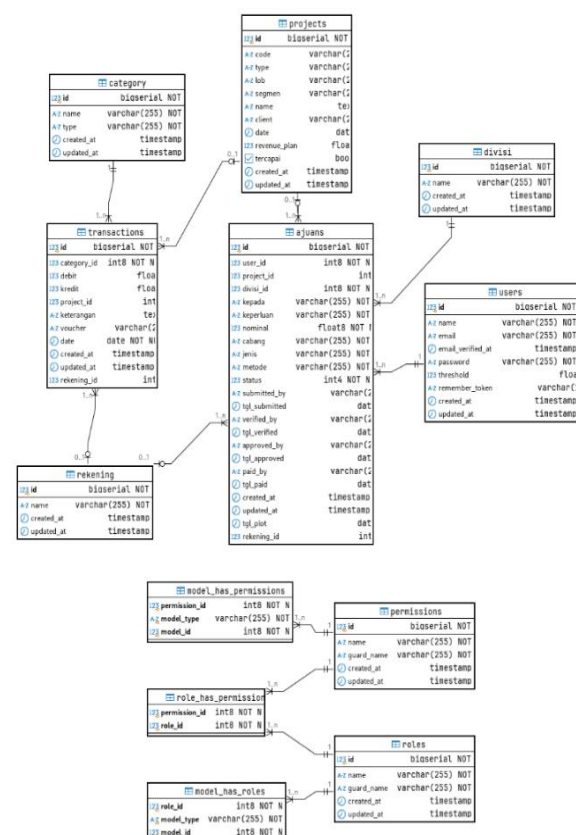
Dasbor diimplementasikan menggunakan beberapa teknologi. Lihat tabel 1 untuk melihat teknologi yang digunakan dalam penelitian ini.

Tabel 1. Tech Stack

Tech	Usage
Laravel	Framework dari PHP. Digunakan untuk mengembangkan website itu sendiri
Breeze	Library dari laravel untuk memudahkan melakukan autentikasi
Spatie	Library dari laravel untuk urusan otorisasi. Sistem RBAC akan dibangun di atas pustaka ini.
Laravel-excel	Library untuk implementasi fitur <i>import excel</i>
Posgresql	DBMS untuk DP

4.2. Database Design

Gambar 2 adalah gambar ERD dari DP:



Gambar 2. ERD dari Dasbor

Beberapa penjelasan penting dari ERD pada gambar 2:

a. Projects

Entitas 'projects' merupakan master tabel dari tabel 'ajuans' dan 'transactions'. Beberapa project mempunyai pemasukan dan pengeluaran (transaksi). Setiap project juga memiliki kode proyek (setiap kode unik).

b. Transactions

Entitas 'transactions' merupakan tabel dimana disimpan histori setiap transaksi (pemasukan dan pengeluaran) dari sebuah project. Setiap pemasukan (debit) dan pengeluaran (kredit) mempunyai kategori atau tipe masing-masing. Contoh pengeluaran kredit kategori adalah perjalanan dinas dan masih banyak lagi kategori. Dibuatlah tabel independen, yaitu tabel 'category'. Tabel transaksi berhubungan dengan tabel 'category' dengan menggunakan *foreign id* dari 'category'.

c. Ajuans

Entitas 'ajuans' merupakan tabel yang menyimpan data-data berhubungan dengan *accounting*.

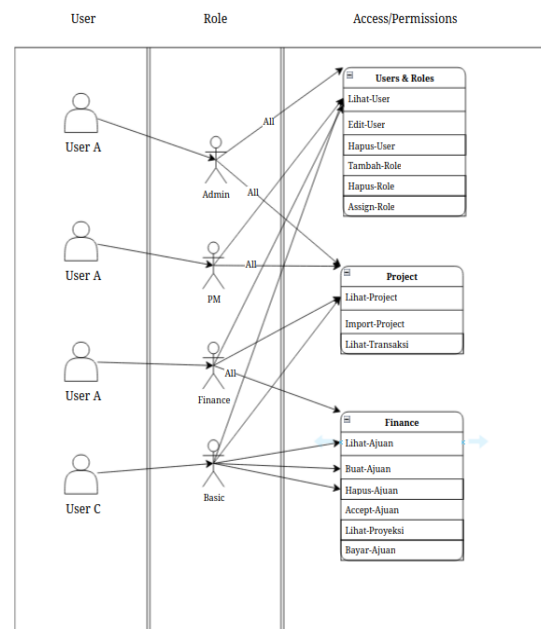
d. Roles & Permissions

Tabel-tabel yang berada dibawah merupakan tabel yang bersangkutan mengenai RBAC. Tabel-tabel tersebut merupakan dibuat otomatis oleh pustaka Spatie saat melakukan migrasi database.

4.3. RBAC Design

Gambar 3 merupakan diagram desain RBAC yang akan dibangun:

RBAC (Role Based Access Control)



Gambar 3. Desain RBAC

Dilihat dalam gambar 3, bahwa setiap *user* memiliki sebuah *role*, dan setiap *role* ini memiliki beberapa *permissions*. Dalam dasbor, semua *permissions* bisa dibagi menjadi tiga kategori, yakni 'Users & Roles', 'Project', 'Finance'. Setiap pengguna

dalam DP, memiliki rolenya masing-masing. Berikut penjelasan mengenai masing-masing role:

a. Admin

Role admin bisa diartikan sebagai 'super user', yaitu pengguna yang mempunyai hak akses di banyak tabel.

b. PM

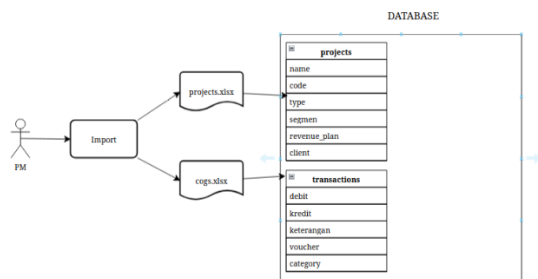
PM atau Project Manager merupakan role yang berurusan mengenai hal-hal berkaitan dengan project. Contoh, ia bisa melakukan 'import project', yakni *import* file project excel ke DP yang nantinya akan disimpan di database yang selanjutnya ditampilkan dalam *website* DP.

c. Finance

Role yang berurusan dengan hal-hal keuangan/akuntansi/ajuan. Role finance bisa menyetujui sebuah ajuan yang diajukan oleh pengguna lain dan selanjutnya dibayar, atau menolak ajuan tersebut.

4.4. Implementasi Import Excel

Terdapat dua file excel, yaitu 'Project.xlsx' dan 'COGS.xlsx'. File 'Project.xlsx' terdapat data-data penting terkait sebuah proyek, contoh yakni nama proyek atau kode proyek. Sedangkan file 'COGS.xlsx' menyimpan data-data pemasukan dan pengeluaran dana dari sebuah proyek dan tipe nya, contoh pengeluaran adalah 'perjalanan dinas'. Lihat ERD pada gambar 2 untuk melihat lebih jelas mengenai data-data pada file excel. Gambar 4 adalah gambar diagram untuk proses import:



Gambar 4. Diagram proses *import*

Sementara untuk menyimpan data dalam excel sesuai dengan atribut yang ada di tabel, gambar 5 adalah kode program untuk import berkas excel ke database:

```
class ProjectsImport implements ToModel, WithStartRow, WithCalculatedFormulas, WithHeadingRow
{
    /**
     * @param array $row
     *
     * @return \Illuminate\Database\Eloquent\Model|null
     */
    public function model(array $row)
    {
        $project = Project::create([
            'code' => $row['code'],
            'type' => $row['type'],
            'segmen' => $row['segmen'],
            'name' => $row['name'],
            'client' => $row['client'],
            'revenue_plan' => $row['revenue_plan'],
        ]);
    }
}
```

Gambar 5. Kode program untuk import excel ke model Project

Perlu diperhatikan bahwa laravel-excel akan memproses file excel baris per-baris. Baris (atau row) ini bisa diakses dengan variabel '\$row'. Dengan class 'WithHeadingRow' (pojok kanan atas) yang disediakan oleh laravel-excel, key dalam '\$row' otomatis akan ber-kolerasi dengan nama heading kolom dalam file excel yang diimport.

4.5. Pembuatan Role

Berdasarkan desain RBAC, DP terdapat 4 *role*. Berkat pustaka Spatie, pembuatan *role* ini sangat mudah dilakukan. Contoh pembuatan *role* PM pada gambar 6:

```
// daftar permission terkait PM
$permissions_untuk_pm = [
    'lihat-project',
    'import-project',
    'lihat-transaksi'
];

// looping untuk membuat semua permission di atas ke model Permissions
foreach ($permissions as $permission) {
    Permission::create(['name' => $permission]);
}

$project_manager = Role::create(['name' => 'PM']); // membuat role pm
$project_manager->givePermissionTo($permissions_untuk_pm::all()); // memberikan role pm
semua permission terkait pm
```

Gambar 6. Kode program pembuatan role PM

Role PM mendapat beberapa *permissions* atau hak yang berhubungan dengan manajemen proyek.

4.6. Pemberian Role

Assigning Role merupakan proses dimana memberikan sebuah *role* kepada seseorang. Contoh implementasi dalam bentuk kode program pada gambar 7:

```
$contoh_user_pm = User::create([
    'name' => 'foo',
    'email' => 'bar@gmail.com',
    'password' => 'biz1234',
]);
$contoh_user_pm->assignRole('PM')
```

Gambar 7. Kode program memberikan *role* kepada *user*

'\$contoh_user_pm' merupakan user yang dikembalikan saat membuat sebuah user dari model *User*. 'assignRole()' merupakan *method* yang disediakan oleh *Spatie* untuk memberikan *role* kepada obyek user yang dipanggil. *Method* tersebut menerima sebuah argumen yakni nama *role*.

4.7. Limitasi Rute

Sebuah *role* mempunyai *privilege* atau hak istimewa yang *role* lain tidak punya. Dalam konteks RBAC dalam *Website*, hak istimewa ini merupakan hak akses ke rute-rute tertentu yang *role* lain tidak diperbolehkan.

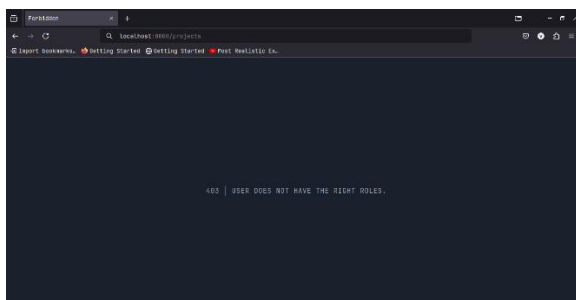
```
Route::get('/projects', [Project::class, 'index'])->middleware([
    'auth',
    'verified',
    'role:PM', // <==== hanya user dengan role PM dapat akses rute ini
]);
// ATAU
Route::get('/projects', [Project::class, 'index'])->middleware([
    'auth',
    'verified',
    'permissions:lihat-project', // <==== hanya user yang mempunyai permission 'lihat-project' dapat akses rute ini
]);
```

Gambar 8. Kode program untuk limitasi rute

Dalam opsi pertama menggunakan *middleware* 'role:PM' yang disediakan oleh *Spatie* pada rute '/projects'. *Middleware* ini akan memastikan bahwa hanya *role* PM yang hanya dapat mengakses rute ini. Dalam opsi kedua menggunakan *middleware* yang berbeda, yaitu 'permissions:lihat-project'. *Middleware* ini akan memastikan bahwa hanya user yang memiliki *permissions* lihat-project yang dapat mengakses rute ini. Contoh, jika seorang user dengan *role* PM mengakses rute ini, maka *middleware* akan mengizinkannya, dikarenakan *role* PM yang dimiliki user ini, memiliki *permissions* 'lihat-project'.

4.8. Testing

Mengetes jika seorang *user* yang tidak memiliki *role* PM ataupun *permissions* 'lihat-project' mencoba mengakses rute yang memerlukan hak tersebut.

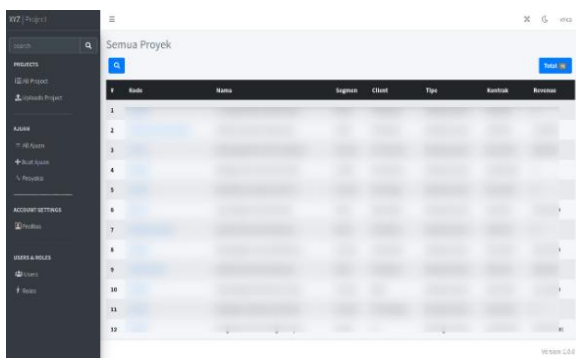


Gambar 9. Testing hak akses RBAC

Middleware tidak akan mengizinkan pengguna ini untuk mengakses rute ini. Bahkan *Spatie* sudah menyediakan *error message* yang sangat sesuai dan relevan.

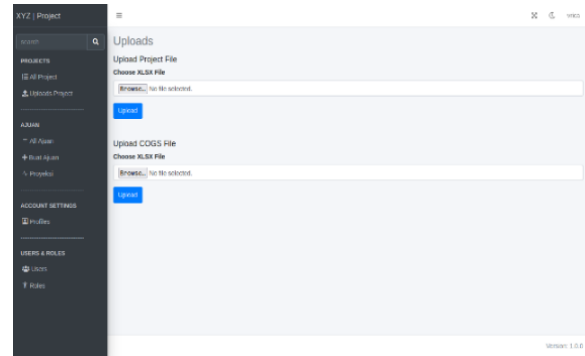
4.9. Hasil Akhir

Kumpulan dari hasil implementasi Dashboard Proyek yang disuguhkan dalam bentuk gambar, yakni beberapa screenshot dari Website Dashboard Proyek, namun tidak semua:



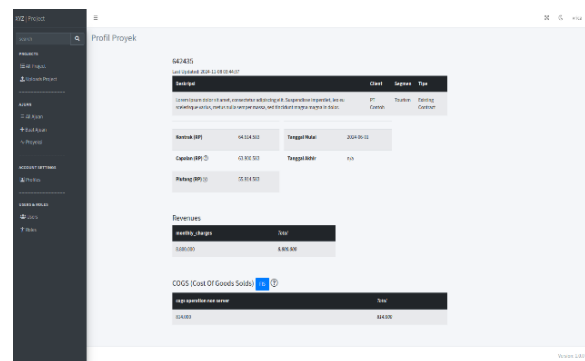
Gambar 10. Tampilan daftar proyek

Gambar 10 adalah halaman daftar semua proyek. Tabel di *masking* dikarenakan data rahasia perusahaan.



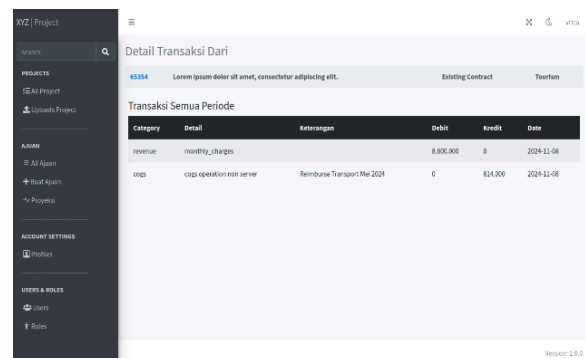
Gambar 11. Tampilan import excel

Gambar 11 merupakan halaman untuk *import* berkas *excel*.



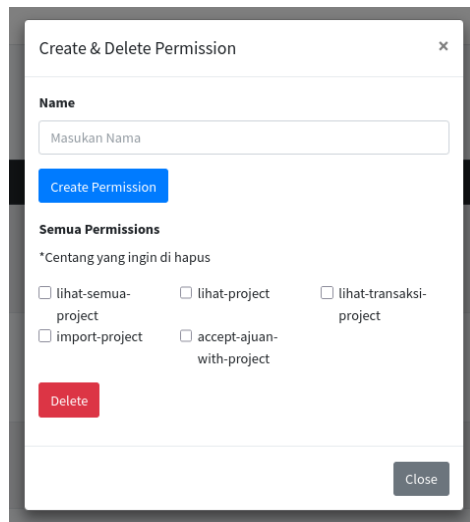
Gambar 12. Tampilan detail proyek

Gambar 12 merupakan halaman detail dari sebuah proyek. Terdapat informasi mengenai proyek meliputi pemasukan, pengeluaran, deskripsi dll.



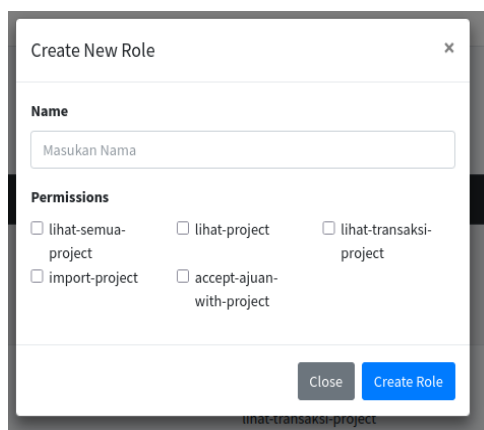
Gambar 13. Tampilan transaksi proyek

Gambar 13 merupakan halaman detail transaksi dari sebuah proyek.



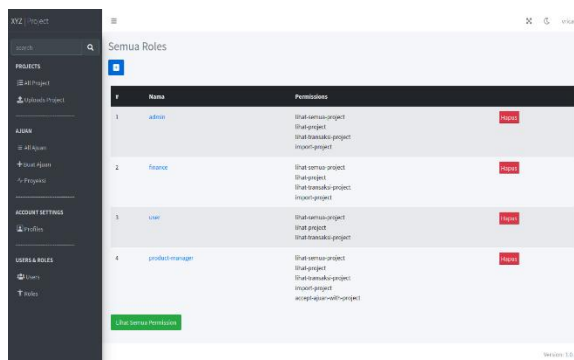
Gambar 14. Tampilan modal buat & hapus *permissions*

Gambar 14 merupakan modal untuk membuat atau menghapus sebuah *permissions*.



Gambar 15. Tampilan modal buat role baru

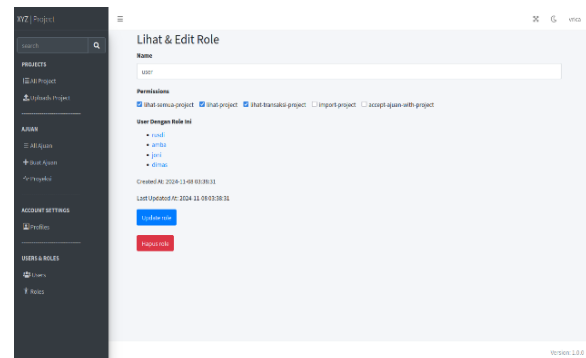
Gambar 15 merupakan tampilan modal untuk membuat sebuah role baru. Terdapat kotak *input* untuk memasukkan nama role dan pilihan *permissions* untuk role baru.



Gambar 16. Tampilan daftar semua role

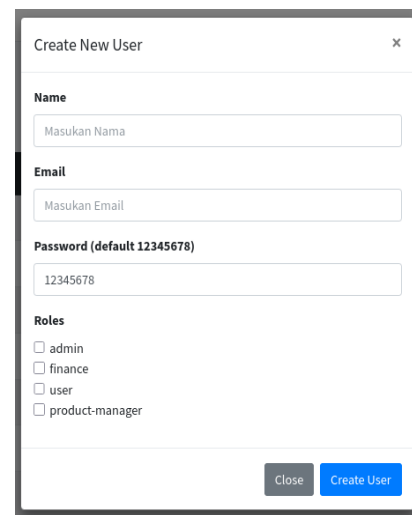
Gambar 16 merupakan tampilan halaman yang menampilkan tabel berisi semua *role* beserta

permissions yang *role* miliki. Terdapat juga aksi hapus untuk menghapus *role*.



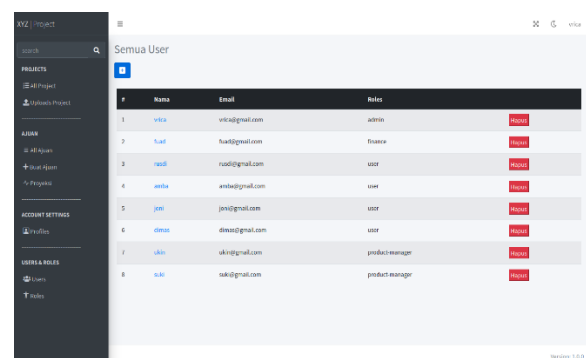
Gambar 17. Tampilan lihat & edit role

Gambar 17 merupakan tampilan halaman lihat dan edit dari sebuah *role*. Terdapat informasi mengenai semua pengguna yang memiliki *role* ini.



Gambar 18. Tampilan modal buat user baru

Gambar 18 merupakan tampilan modal untuk membuat pengguna baru. Seorang Admin dapat memasukkan nama, email, password, dan *role* dari pengguna tsb.



Gambar 19. Tampilan daftar semua user

Gambar 19 merupakan halaman yang menampilkan semua pengguna dashboard.

4.10. Pengujian

Tabel 2. Pengujian *Blackbox*

No	Pengujian	Hasil Diharapkan	Hasil
1	Login (valid)	Pengguna berhasil login dan diarahkan ke halaman home	Berhasil
2	Login (tidak valid)	Pengguna gagal login dengan pesan “email atau password salah”	Berhasil
3	Role bukan PM mencoba import excel	Pengguna tidak diperbolehkan dan menampilkan pesan “User does not have the right role”	Berhasil
4	Role bukan PM mencoba lihat transaksi dari sebuah proyek	Pengguna tidak diperbolehkan dan menampilkan pesan “User does not have the right role”	Berhasil
5	Role bukan Admin mencoba edit dan hapus user	Pengguna tidak diperbolehkan dan menampilkan pesan “User does not have the right role”	Berhasil
6	Role bukan Admin mencoba tambah, hapus dan memberi role kepada user.	Pengguna tidak diperbolehkan dan menampilkan pesan “User does not have the right role”	Berhasil
7	Role bukan finance mencoba accept dan bayar sebuahajuan	Pengguna tidak diperbolehkan dan menampilkan pesan “User does not have the right role”	Berhasil
8	Import excel (valid)	Berhasil menyimpan data-data ke database, lalu diarahkan ke halaman daftar proyek berisi proyek-proyek dari excel yang diimpor	Berhasil

Berdasarkan sejumlah pengujian pada tabel 2, bisa disimpulkan bahwa dasbor bisa menangani pengujian-pengujian dengan baik. Saat pengguna mencoba login dengan email atau password yang tidak valid, dasbor akan memberi pesan error yang relevan. Saat pengguna dengan role yang tidak sesuai mencoba melakukan aksi yang diluar kemampuan role tersebut, dasbor akan menolak aksi tersebut. Contoh lain adalah saat pengguna impor file excel, data-data dari file tersebut akan mengisi tabel-tabel yang ada di database sesuai dengan ERD, lalu data-data tersebut akan divisualisasikan di dasbor.

5. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa manajemen proyek dengan dashboard menggunakan sistem RBAC lebih mudah dilakukan jika dibandingkan dengan *spreadsheet* biasa pada kasus PT.XYZ. Kemampuan dashboard untuk mengimpor *spreadsheet* memudahkan integrasi manajemen lama dengan yang baru. Dengan adanya sistem RBAC, pengelolaan proyek menjadi lebih aman, khususnya dengan adanya pembagian peran. Pengujian sistem RBAC yang telah dilakukan menunjukkan bahwa hak kontrol akses sangat terjamin. Utilisasi peran sangat membantu mengamankan sebuah *resource* yang dibatasi. Seorang pengguna tanpa hak akses yang memadai tidak bisa mengakses rute sembarangan. Penelitian ini memberi wawasan praktis bagi bisnis lain mengenai pemanfaatan teknologi untuk pengelolaan proyek mereka. Untuk pengembangan lebih lanjut disarankan untuk memungkinkan impor excel secara paralel untuk meningkatkan performa.

DAFTAR PUSTAKA

- [1] A. Shrivastava, I. Jaggi, N. Katoch, D. Gupta, and S. Gupta, “A Systematic Review on Extreme Programming,” *Journal of Physics: Conference Series*, vol. 1969, no. 1, p. 012046, Jul. 2021, doi: 10.1088/1742-6596/1969/1/012046.
- [2] L. J. Turner, “Project planning and control: Framework for effective project delivery,” *Project Management Journal*, vol. 41, no. 4, pp. 66-78, Apr. 2020.
- [3] L. Harris, “Designing efficient database schemas: Best practices,” *Journal of Database Systems*, vol. 44, no. 3, pp. 112-124, Mar. 2020.
- [4] Birdview, “Project Dashboards: types, examples, and benefits,” Birdview PSA, 2023. [Online]. Available: <https://www.birdviewpsa.com/blog/project-dashboards-types-examples-benefits>
- [5] FineReport, “Digital Dashboard: Make You Prosper In Digital Era,” FineReport, 2023. [Online]. Available: <https://www.finereport.com/en/dashboard>
- [6] Traction Technology, “Harnessing the Power of Reports and Real-time Data for Digital Transformation in 2023,” Traction Technology, 2023. [Online]. Available: <https://www.tractiontechnology.com/articles/importance-of-data-driven-decision-making-in-digital-transformation>
- [7] D. Kumar and S. Gupta, “Effective strategies for project planning and execution,” *International Journal of Project Management*, vol. 38, no. 1, pp. 12-20, Jan. 2021.
- [8] Subecz, Z. (2021). Web-development with Laravel framework. *Gradus*, 8(1), 211-218
- [9] C. Chen, “Entity-relationship modeling in database design,” *Journal of Database Management*, vol. 31, no. 2, pp. 22-35, Feb. 2020.
- [10] Rahmawati, E., & Setiawan, R. (2020). Implementasi Role-Based Access Control menggunakan Laravel dan Spatie Laravel

- Permission. *Jurnal Sistem Informasi Indonesia*, 6(2), 78-89.
- [11] Widodo, T., & Astuti, D. (2021). Sistem pengelolaan data akademik berbasis web dengan Role-Based Access Control. *Jurnal Teknologi dan Sistem Informasi*, 15(1), 32-41
- [12] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38-47, Feb. 2020