Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 3 Iss. 2 October 2023, pp: 339-348
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

The Implementation of Role Based Access Control in a Cloud-Based Supply Chain Management System

Penerapan Role Based Access Control dalam Sistem Supply Chain Management Berbasis Cloud

Yuricha[1*], Irwan Kurnia Phan[2]

[1]Program Studi Sistem dan Teknologi Informasi, Institut Teknologi dan Bisnis Sabda Setia, Indonesia
[2]Fakultas Teknologi Informasi, Universitas Widya Dharma Pontianak, Indonesia

E-Mail: [1]yuricha@itbss.ac.id, [2]irwanphan@widyadharma.ac.id

Received Sep 10th 2023; Revised Oct 15th 2023; Accepted Nov 5th 2023
Corresponding Author: Yuricha

Abstract

Cloud-based Supply Chain Management (SCM) systems have become one of the breakthroughs that support business scalability and flexibility. The complexity of SCM in supporting business processes is not only found in the various modules and features, but also the various levels of users who access the system according to their respective goals. Various research on the development of SCM systems has been carried out with various implementations that have been carried out in various sectors, and concluded that internet access to the system that can be done from anywhere, anytime and by anyone allows for security vulnerabilities in the form of unauthorized resource access or access rights. The application of Role-Based Access Control (RBAC) becomes a solution that can be applied and prevents resource access from unauthorized parties. The application of RBAC in one of the frameworks that is widely used today, namely the Laravel Framework, can be done by utilizing the Laravel Spatie library. With the implementation of this, the research objectives can be fulfilled, namely to answer the data security challenges and maintain the integrity of the authorization of each stakeholder role in complex cloud-based SCM systems with various levels of stakeholder access levels.

Keyword: Authorization, Business Process, Laravel Spatie, RBAC, SCM

Abstrak

Sistem *Supply Chain Management* (SCM) yang berbasis *cloud* telah menjadi salah satu terobosan yang mendukung skalabilitas dan fleksibilitas bisnis. Kompleksitas SCM dalam mendukung proses bisnis, tidak hanya terdapat pada beragamnya modul dan fitur, tetapi juga berbagai tingkat pengguna yang mengakses sistem sesuai dengan tujuannya masing-masing. Berbagai penelitian mengenai pengembangan sistem SCM telah banyak dilakukan dengan berbagai implementasi yang telah dilakukan di berbagai sektor, dan dapat dikonklusikan bahwa akses sistem melalui internet yang dapat dilakukan dari mana saja, kapan saja dan oleh siapa saja memungkinkan adanya celah keamanan data berupa akses sumber daya oleh pengguna yang tidak memiliki otorisasi maupun hak akses. Penerapan *Role-Based Access Control* (RBAC) menjadi solusi yang dapat diterapkan dan mencegah akses *resource* dari pihak-pihak yang tidak memiliki otorisasi. Penerapan RBAC dalam salah satu *framework* yang banyak digunakan hingga saat ini, yaitu *Framework* Laravel, sangat dapat dilakukan dengan memanfaatkan *library* Laravel Spatie. Dengan adanya penerapan ini, tujuan dari penelitian dapat terpenuhi, yaitu menjawab tantangan keamanan data dan menjaga integritas otorisasi setiap peran *stakeholder* pada sistem SCM berbasis *cloud* yang kompleks dengan berbagai tingkat *level stakeholder* akses yang ada.

Kata Kunci: Laravel Spatie, Otorisasi, Proses Bisnis, RBAC, SCM

1. PENDAHULUAN

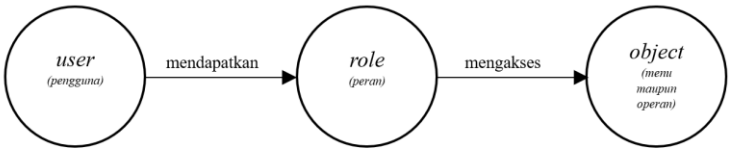
Sistem *Supply Chain Management* (SCM) berbasis *cloud* mendukung keberlangsungan rantai pasok telah menjadi tren yang digunakan oleh beberapa perusahaan kecil maupun menengah ke atas [1]. Beberapa sistem SCM juga telah bertransformasi sebagai *Software-as-a-Service* (SaaS) dengan mengusung fleksibilitas dan skalabilitas yang dikedepankan. Selama terhubung dengan internet, aplikasi tersebut dapat diakses dari mana saja dan kapan saja. Hal tersebut juga mendorong terciptanya sistem kolaborasi antar-*stakeholder* serta memungkinkan adanya pembagian tugas sesuai peran dalam penggunaan satu aplikasi yang sama untuk

mencapai tujuan bisnis. Sifat kolaborasi yang semakin kompleks di berbagai level manajemen yang ada dalam suatu perusahaan membutuhkan sistem yang mampu berperan dinamis dan memberikan keamanan data sistem yang ada [2].

Sistem SCM berfokus pada pencatatan dan pengelolaan data barang dimulai dari pembelian bahan dari pemasok, produksi hingga ke proses penjualan dan pengiriman barang ke pelanggan [3]. Setiap *stakeholder* yang berkaitan dengan sistem perlu untuk memasukkan *record* data untuk mendukung tercapainya proses bisnis yang ideal. Sistem pun akan mengelompokkan sub-proses yang ada ke dalam beberapa modul yang saling berkaitan untuk nantinya dapat menghasilkan informasi yang dibutuhkan oleh pemangku kepentingan dalam mendukung proses pengambilan keputusan strategis perusahaan [4].

Hal yang tidak dapat dipungkiri adalah terdapat beberapa modul yang memuat data-data sensitif dan hanya bisa diakses oleh stakeholder yang memiliki otorisasi dan mereka yang berada di level *top-middle management* [5]. Data sensitif yang ada biasanya berkaitan dengan keuangan seperti harga beli, data hutang, diskon pembelian, harga jual kepada pelanggan tertentu dan lainnya. Terdapat beberapa solusi yang dapat dilakukan untuk menjaga data-data sensitif yang ada, salah satu solusi yang dapat dilakukan adalah adalah membuat beberapa *route* versi halaman yang berbeda untuk tiap *stakeholder* yang akan mengakses sistem [6]. Namun hal tersebut bukan praktik ideal dan pemborosan waktu serta tenaga dalam pengembangan dan pengerjaannya [7].

Kontrol terhadap akses (*access control*) menjadi salah satu solusi ideal yang dapat diterapkan dalam sistem yang memiliki isu keamanan data dalam hal otorisasi setiap peran yang ada dalam sistem. Dalam penelitian yang dipublikasi pada tahun 2021, RBAC (*Role-Based Access Control*) menjadi salah satu mekanisme *access control* yang paling banyak digunakan dengan fleksibilitas dan keamanan yang dapat diandalkan [8]. Pemanfaatan RBAC menjadi salah satu solusi yang efektif dan efisien untuk dilakukan terutama untuk suatu sistem yang memiliki peran stakeholder yang kompleks dan membutuhkan sistem kolaborasi yang tinggi [9]. RBAC memungkinkan tiap modul maupun informasi yang ada dapat diatur dan dibatasi hak aksesnya sesuai dengan peran *stakeholder* yang ditentukan seperti pada Gambar 1.



Gambar 1. Kerangka RBAC [9]

Sistem SCM yang mengedepankan fleksibilitas dan skalabilitas tinggi sangat disarankan untuk dibangun menggunakan *framework* sehingga dapat dikembangkan dan mendukung *maintenance* sistem yang berkelanjutan [10]. Hal ini juga sejalan dengan tren pengembangan aplikasi berbasis *cloud* saat ini yang banyak menggunakan *framework* salah satunya adalah *framework* Laravel yang bersifat *open source* dan mampu mendukung sistem yang mendukung skalabilitas [11]. Penerapan *framework* Laravel juga mendukung penerapan RBAC dengan menggunakan *library* Laravel Spatie. Hal ini tidak hanya mengefisiensikan waktu pengerjaan tetapi efisiensi waktu implementasi ke *cloud* [12].

Penelitian sebelumnya terkait dengan pengembangan sistem aplikasi SCM telah banyak dilakukan selama beberapa tahun terakhir. Namun, masih sedikit yang membahas tentang isu keaman data yang berkaitan dengan otorisasi akses data. Penelitian berfokus pada rancang bangun aplikasi yang dapat diterapkan dalam perusahaan [13],[14],[15],[16],[17],[18] dan belum dibahas tentang isu keamanan yang perlu diterapkan dalam sistem tersebut. Dalam praktik nyata, isu tentang keamaman data telah menjadi isu global [2] yang sangat perlu diperhatikan dalam mendukung sustanbilitas perusahaan dalam mencapai tujuan bisnis [19].

Terdapat beberapa penelitian yang sudah mencoba mengamankan data berdasarkan peran dengan menggunakan RBAC. Salah satunya adalah penerapan RBAC pada sistem informasi manajemen pendidikan, sistem memastikan pengguna yang tidak memiliki hak akses tidak dapat mengakses satu atau lebih halaman tertentu [20]. Pada penelitian ini tidak dijelaskan secara rinci penerapan RBAC yang dapat dikembangkan lebih lanjut pada sistem lainnya selain manajemen pendidikan.

Penerapan RBAC pada sistem pelayanan kependudukan berbasis web di tingkat kabupaten sehingga keamanan sistem dengan adanya pembagian kewenangan pengaksesan sistem dapat ditingkatkan [21]. Pada penelitian ini, penerapan RBAC dilakukan pada *middleware* dengan memberikan hasil *response* sesuai *role* yang *login* ke sistem. Namun, penerapan RBAC seperti ini cocok digunakan pada sistem yang menggunakan API (*Application Programming Interface*) dan belum disajikan bagaimana penggunaannya pada aplikasi yang memerlukan *interface* bagi *end user*-nya. Model seperti ini juga memerlukan pendefinisian satu per satu dan tidak memungkinkan untuk mendukung kedinamisan kecuali ada perubahan pengkodean pada sistem yang ada.

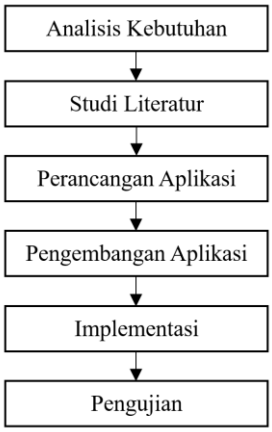
Konsep RBAC juga diterapkan dalam salah satu penelitian perancangan database sistem informasi sekolah [22]. Penerapan RBAC dilakukan pada level *database* menggunakan PostgresSQL. Pada level

database, penerapan berhasil memastikan otorisasi beberapa peran user yang ada dalam sistem namun penerapan tersebut belum dapat digunakan secara langsung oleh *end user* karena belum dikombinasikan dengan bahasa pemrograman pendukung interface yang dapat digunakan oleh *end user*.

Dari studi literatur yang dilakukan, belum ada penelitian yang membahas secara rinci penerapan langsung RBAC dalam sistem SCM yang menggunakan *framework* Laravel masih menjadi tren hingga saat ini [11] serta penerapan RBAC menggunakan *library* Laravel Spatie. Padahal dengan maraknya pengembangan sistem SCM yang diterapkan dan diteliti di beberapa perusahaan yang ada, nilai kebergunaan sistem bukanlah satu-satunya hal yang perlu dicapai, tetapi juga nilai keamanan data terutama pada sistem berbasis *cloud*. Melalui penelitian ini, peneliti ingin menyelesaikan tantangan keamanan data dan menjaga integritas otorisasi setiap peran *stakeholder* sehingga nantinya dapat menjadi suatu standar baru dalam pengembangan sistem selanjutnya seperti SCM maupun sistem lainnya.

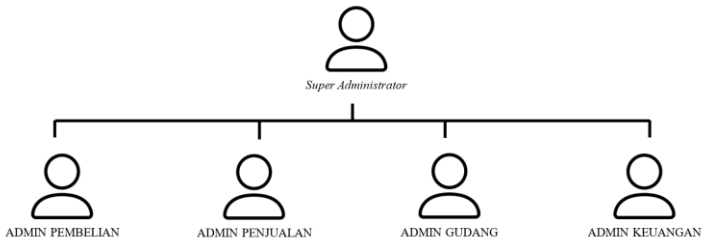
2. METODOLOGI PENELITIAN

Tahapan penelitian ini mengikuti metode penelitian SDLC (*System Development Life Cycle*) *Waterfall* yang dimulai dengan analisis kebutuhan, studi literatur, perancangan aplikasi, pengembangan aplikasi, implementasi, pengujian dan kesimpulan seperti pada Gambar 2.



Gambar 2. Metode Penelitian

Pada tahapan Analisis kebutuhan seputar kompleksitas sistem SCM yang digunakan oleh berbagai peran *stakeholder* dan membutuhkan keamanan terhadap data dalam mendukung keberlangsungan dan tujuan bisnis [1]. Analisis kebutuhan didasarkan pada kebutuhan fungsionalitas 5 (lima) bagian peran utama yang ada dalam sistem SCM, yaitu bagian pembelian, bagian penjualan, bagian keuangan dan bagian gudang serta bagian yang mengatur keseluruhan bagian lainnya (*super administrator*) seperti pada Gambar 3.



Gambar 3. Kebutuhan Peran Sistem SCM

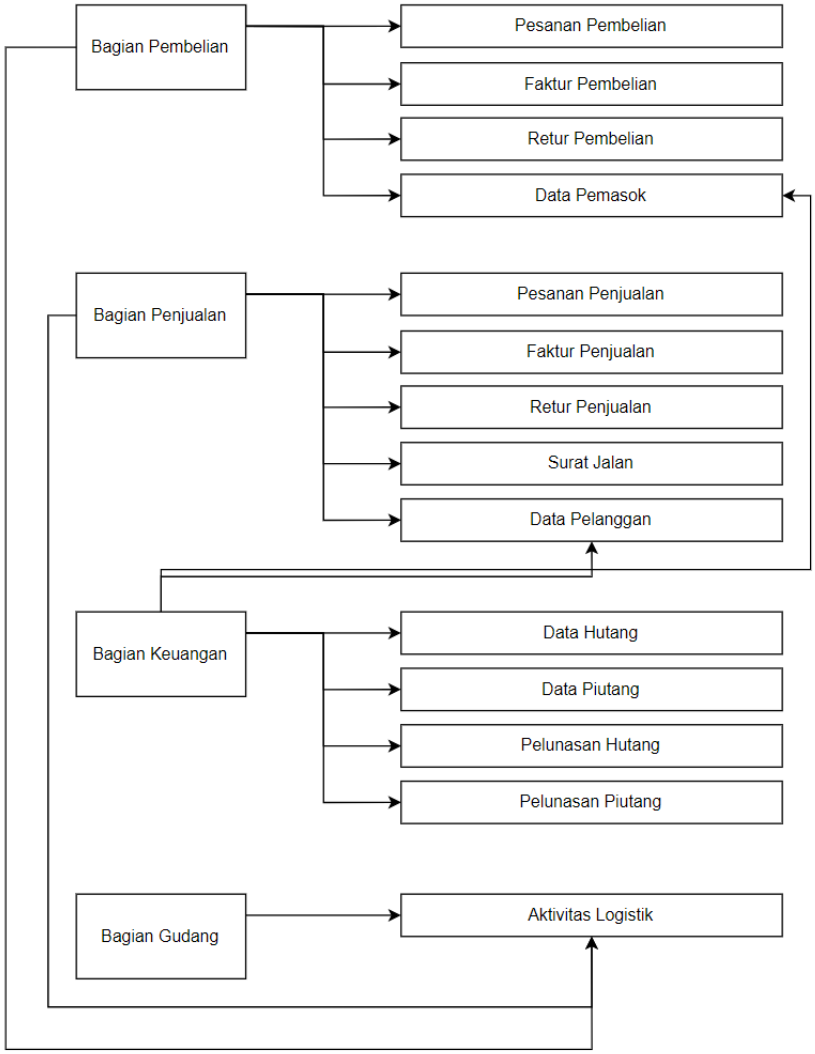
Studi literatur yang dilakukan berfokus pada RBAC dan penerapannya pada sistem-sistem SCM berbasis *cloud*. Pada tahapan ini juga, peneliti mengambil beberapa sampel penelitian terdahulu yang berfokus pada penelitian tanpa memperhatikan keamanan data atau penerapan sejenis RBAC. Hal ini dilakukan guna mendapatkan gap penelitian serta mendalami seberapa pengaruh penerapan RBAC pada sistem SCM.

Tahapan berikutnya adalah tahap perancangan aplikasi dengan memetakan bagian mana saja yang akan diimplementasikan model RBAC pada bagian *framework* Laravel yang ada termasuk bagaimana pembagian peran beserta modul yang ada untuk diterapkan ke dalam model RBAC. Selanjutnya adalah tahapan pengembangan aplikasi dengan menerapkan apa saja yang telah dibuat pada tahapan perancangan. Pada tahap ini pula, proses coding dilakukan guna menerapkan RBAC menggunakan *library* Laravel Spatie.

Setelah itu, tahapan selanjutnya adalah implementasi sistem agar dapat diakses melalui internet. Tahapan berikutnya setelah implementasi adalah tahapan pengujian. Pengujian aplikasi dilakukan dengan beberapa skenario yang telah ditentukan untuk diujikan ke sistem SCM yang telah di-*deploy*. Skenario-skenario dibuat sesuai dengan kebutuhan sistem terutama yang berhubungan dengan kerahasiaan data dan akses masing-masing peran ke sistem.

3. HASIL DAN PEMBAHASAN

Sistem SCM yang sebelumnya telah dibangun menggunakan *framework* Laravel selanjutnya akan diterapkan RBAC. Bagian pembelian berhubungan dengan modul seperti pesanan pembelian, faktur pembelian, dan retur pembelian. Bagian penjualan dapat mengakses pesanan penjualan, faktur penjualan, retur penjualan, dan surat jalan. Pada bagian keuangan berhubungan dengan data pemasok, data pelanggan, piutang, hutang dan pelunasan, sedangkan pada bagian gudang dapat mengakses aktivitas logistik yang berkaitan langsung dengan penerimaan dan pengiriman barang seperti pada Gambar 4.



Gambar 4. Gambaran Akses Modul Per Peran

Aspek kehandalan dari sistem ini adalah setiap bagian peran dapat mengakses modul yang diperlukan untuk mendukung operasional yang ada (seperti pada Gambar 4) dan *user* yang memiliki bagian tersebut tidak bisa mengakses bagian lainnya kecuali diberi akses atau user tersebut adalah *super administrator*. Pada bagian inilah RBAC akan diterapkan untuk memastikan kehandalan sistem dalam hal akses per peran menggunakan Laravel Spatie.

Selain pengaksesan modul, terdapat batasan-batasan tertentu yang dituangkan dalam *permission list* dalam satu modul. Dalam satu bagian peran terdiri atas beberapa user/pengguna yang memiliki *full access* di masing-masing modul tetapi juga ada yang hanya memiliki *access* tertentu saja. Seperti pada bagian pembelian, ada *user* yang dapat membuat pesanan pembelian tetapi tidak dapat melakukan perubahan terhadap data pesanan pembelian yang telah dibuat, sedangkan ada *user* yang dapat melakukan keduanya atau hanya dapat

mencetak hasil pesanan pembelian tanpa memasukkan/mengubah data pesanan pembelian. Rincian pembagian user terlihat pada Tabel 1.

Tabel 1. Pembagian *User* Per Peran

Nama Peran	Nama *User*	Modul	*S*	*C*	*U*	*D*	*P*
ADMIN_PEMBELIAN	ADMIN PEMBELIAN 001	Pesanan Pembelian	Y	Y	Y	Y	Y
		Faktur Pembelian	Y	Y	N	N	Y
		Retur Pembelian	Y	Y	N	N	Y
		Data Pemasok	Y	N	N	N	N
		Aktivitas Logistik	Y	Y	N	N	N
ADMIN_PEMBELIAN	SUPERVISOR PEMBELIAN 001	Pesanan Pembelian	Y	N	Y	Y	Y
		Faktur Pembelian	Y	N	Y	Y	Y
		Retur Pembelian	Y	N	Y	Y	Y
		Data Pemasok	Y	N	Y	Y	N
		Aktivitas Logistik	Y	N	Y	Y	N
ADMIN_PENJUALAN	ADMIN PENJUALAN 001	Pesanan Penjualan	Y	N	N	N	Y
		Faktur Penjualan	Y	Y	Y	N	Y
		Retur Penjualan	Y	Y	Y	N	Y
		Surat Jalan	Y	Y	Y	N	Y
		Data Pelanggan	Y	Y	N	N	Y
ADMIN_KEUANGAN	ADMIN KEUANGAN 001	Data Hutang	Y	N	Y	N	Y
		Data Piutang	Y	N	Y	N	Y
		Pelunasan Hutang	Y	Y	Y	N	Y
		Pelunasan Piutang	Y	Y	Y	N	Y
		Data Pemasok	Y	Y	Y	Y	Y
		Data Pelanggan	Y	Y	Y	Y	Y
ADMIN_GUDANG	ADMIN GUDANG 001	Aktivitas Logistik	Y	Y	Y	N	Y

Pada Tabel 1 terlihat kolom S untuk *Show* (menampilkan menu/modul dan data), kolom C untuk *Create* (membuat data baru), kolom U untuk *Update* (mengubah data yang sudah ada), kolom D untuk *Delete* (menghapus data yang telah disimpan sebelumnya) dan kolom P untuk *Print* (mencetak data). Data Y mewakili *Yes*/Ya sedangkan data N mewakili *No*/Tidak. Penerapan RBAC yang diharapkan adalah setiap user dapat memperoleh hak akses sesuai pemetaan yang dilakukan pada Tabel 1. Untuk data yang N maka user tersebut tidak memiliki akses untuk dapat melakukan kondisi yang diharapkan dalam sistem.

3.1. Perancangan dan Pengembangan Aplikasi

Sistem SCM menggunakan *framework* Laravel 9 diterapkan RBAC dengan penambahan *vendor* library Laravel Spatie. *Framework* Laravel menggunakan konsep MVC (*Model-View-Control*) [11] dan hal yang harus dipastikan adalah dalam perancangan memiliki model *user* yang telah diterapkan otentikasi login sehingga dapat ditambahkan *Trait* untuk mengawali penerapan *library* Laravel Spatie seperti pada Gambar 6.

```
use Illuminate\Foundation\Auth\User as Authenticatable;
use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable
{
    use HasRoles;

    // ...
}
```

Gambar 6. Penambahan *Trait* pada *User* Model [23]

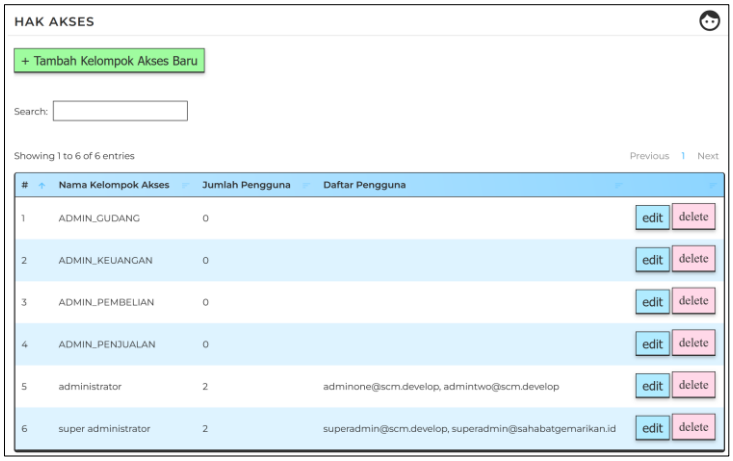
Sistem yang dirancang mengedepankan fleksibilitas dan skalabilitas maka penentuan *role* dan *permission list* yang ada dibuat dinamis dengan menggunakan Role::create(['name' => '']); untuk menambahkan peran yang dibutuhkan dan Permission::create(['name' => '']); untuk menambahkan *permission* [23]. Dalam sistem SCM ini, *role* dan *permission list* yang digunakan untuk memenuhi kebutuhan yang didefinisikan pada penelitian ini seperti pada Tabel 2.

Tabel 2. Penambahan *Role* dan *Permission List* Sistem

Nama *Role*	Pengaturan *Role*	Pengaturan *Permission*
ADMIN_PEMBELIAN	Role::create(['name' => 'ADMIN_PEMBELIAN']);	Permission::create(['name' => 'PEMBELIAN.SHOW']); (*Show*) Permission::create(['name' => 'PEMBELIAN.CREATE']); (*Create*) dan seterusnya
ADMIN_PENJUALAN	Role::create(['name' => 'ADMIN_PENJUALAN']);	Permission::create(['name' => 'PENJUALAN.DELETE']); (*Delete*) Permission::create(['name' => 'PENJUALAN.PRINT']); (*Print*) dan seterusnya
ADMIN_KEUANGAN	Role::create(['name' => 'ADMIN_KEUANGAN']);	Permission::create(['name' => "PELUNASAN_HUTANG.PRINT']); (*Print*) dan seterusnya
ADMIN_GUDANG	Role::create(['name' => 'ADMIN_GUDANG']);	Permission::create(['name' => 'AKTIVITAS_LOGISTIK.SHOW']); (*Show*) Permission::create(['name' => 'KEUANGAN.PRINT']); (*Print*) dan seterusnya

Setelah *role* dan *permission* sudah dibuat, hal selanjutnya yang dikerjakan adalah memasukkan *user-user* yang akan memiliki *role* dan akses terhadap *permission* yang dibuat. $role->givePermissionTo($permission); untuk memberikan *role* pada *user* yang telah dibuat. $permission->assignRole($role); untuk memberikan *permission* sesuai *role*. Setiap halaman telah didefinisikan dalam *route* akan dicek apakah *user* yang login memiliki permission dengan menggunakan *syntax* $user->hasPermissionTo('');.

Pada pengembangan sistem aplikasi dibuatkan *interface* yang memudahkan *user* untuk menggunakan sistem ini. Tampilan *interface* disederhana dengan nama menu Hak Akses yang akan digunakan oleh *Super Administrator* nantinya untuk dapat mengatur peran-peran dalam sistem yang dalam *interface* ini akan didefinisikan sebagai Kelompok Akses. Beberapa menu dalam mendukung 5 (lima) bagian peran utama dalam sistem. Salah satu bagian terpenting dalam penerapan RBAC adalah dengan membuat sebuah halaman Hak Akses yang memuat nama kelompok akses yang mewakili bagian peran utama seperti pada Gambar 7.



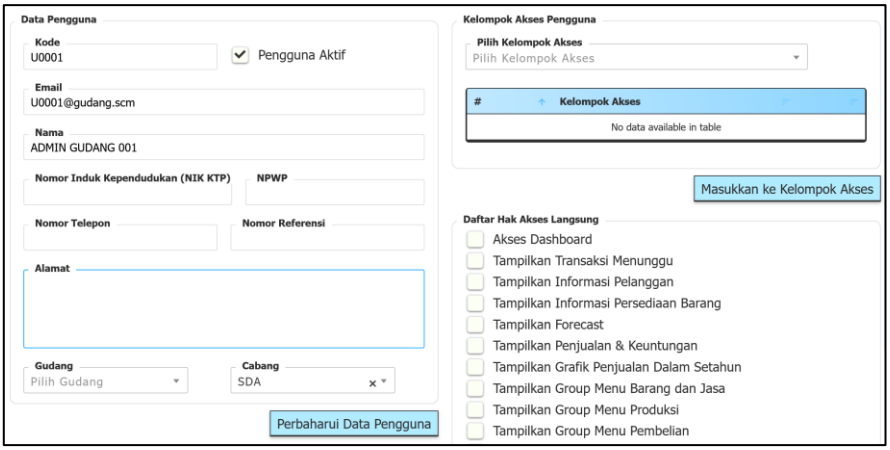
Gambar 7. Halaman *Interface* Pengaturan Hak Akses

Pada Gambar 7, *Interface* menampilkan tabel yang memuat kelompok akses, jumlah pengguna yang telah diberikan akses serta daftar pengguna yang berguna untuk menampilkan list *user* yang telah diberikan akses. Setiap kelompok akses yang ada akan ditambahkan *user-user* yang nantinya akan berhubungan dengan peran yang ada untuk mengakses modul yang sudah dianalisis sebelumnya.

Super Administrator dapat menambahkan permission pada *user-user* yang ada dengan melakukan pengubahan data dengan menekan tombol edit sesuai dengan kelompok akses yang ada pada Gambar 7. Kemudian *interface* berikutnya adalah menampilkan halaman pengaturan *permission* setiap kelompok akses seperti pada Gambar 8.

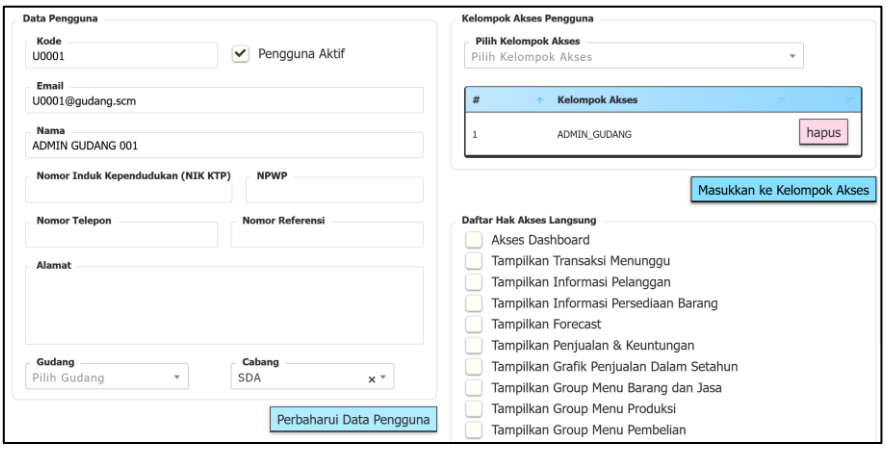
Gambar 8. Halaman Pengaturan Permission untuk Kelompok Akses ADMIN_GUDANG

Gambar 8 menunjukkan beberapa permission yang telah didefinisikan sebelumnya dalam hal ini adalah Aktivitas Logistik. *Super Administrator* dapat melakukan centang pada *checkbox permission* yang ada dan nantinya *backend* dari aplikasi akan menjalankan perintah *syntax* $user->givePermissionTo('melihat halaman aktivitas logistik'); dan begitupun pada permission lainnya. Setelah penambahan *role* beserta *permission* yang ada, *Super Administrator* dapat menambahkan *user-user* yang nantinya akan diberikan otentikasi untuk mengakses sistem. Sistem menyediakan halaman khusus yang dapat diakses oleh pihak berwenang dalam hal ini *Super Administrator* untuk dapat menambahkan user yang dibutuhkan seperti pada Gambar 9.



Gambar 9. Tampilan Halaman Penambahan *User*

Pada bagian kanan Gambar 9 terdapat bagian untuk menambahkan role pada user tersebut, misalnya user yang dibuat pada Gambar 9 ditambahkan *role* sebagai ADMIN_GUDANG maka akan muncul tampilan seperti Gambar 10.



Gambar 10. Tampilan Halaman *User* Setelah Penambahan Kelompok Akses ADMIN_GUDANG

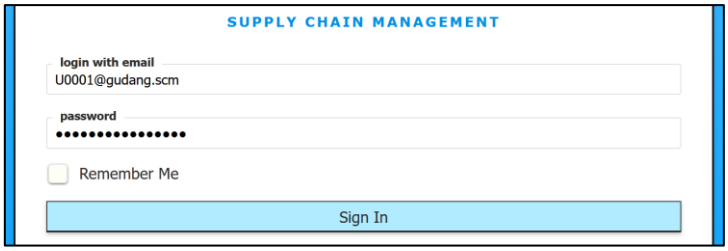
Super Administrator dapat menambahkan *permission* pada *user-user* yang ada dengan melakukan pengubahan data dengan menekan tombol edit sesuai dengan kelompok akses yang ada pada Gambar 9. Kemudian *interface* berikutnya adalah menampilkan halaman pengaturan *permission* setiap kelompok akses seperti pada Gambar 10.

3.2. Implementasi

Implementasi sistem di-*deploy* ke suatu *virtual private server* sehingga memungkinkan *user* untuk mengakses sistem dari mana saja dan kapan saja. VPS yang digunakan diinstall menggunakan sistem operasi Linux Ubuntu. Dalam hal mendukung kedinamisan *role* dan *permission* yang ada, setiap penambahan *role* maupun *permission* yang baru, admin *server* perlu menambahkan perintah CLI [23] php artisan cache:forget spatie.permission.cache untuk menghapus *cache permission* dan php artisan cache:clear untuk menghapus *cache* keseluruhan sistem agar *role* dan *permission* baru dapat terbaca di halaman interface Hak Akses.

3.3. Pengujian

Setelah penerapan RBAC pada sistem, *user* ADMIN GUDANG 001 yang telah dibuat dengan email sistem U0001@gudang.scm dan diuji mulai dari *login* ke sistemnya melalui *interface* sistem yang ada seperti pada Gambar 11.



Gambar 11. Halaman *Login* Sistem untuk ADMIN GUDANG 001

Hasil yang diharapkan ketika otentikasi berhasil, maka ADMIN GUDANG 001 dapat mengakses halaman yang hanya sesuai dengan penambahan *role* dan *permission* yang telah ditetapkan sebelumnya, yaitu Aktivitas Logistik yang berada pada menu utama Penjualan dengan tampilan seperti pada Gambar 12.

Gambar 12. Tampilan List Menu ADMIN GUDANG 001 Setelah *Login*

Pengujian selanjutnya adalah dengan menggunakan beberapa skenario seperti pada Tabel 6.

Tabel 6. Skenario Pengujian Sistem Setelah Penerapan RBAC

Aktor	Skenario	Ekspektasi	Hasil Pengujian
ADMIN PEMBELIAN 001	Aktor mengakses halaman penjualan https://(url)/sale/create (halaman input data baru faktur penjualan)	Muncul 403 Forbidden	Sesuai Ekspektasi
SUPERVISOR PEMBELIAN 001	Aktor melakukan update pada data faktur pembelian karena ada kesalahan input ADMIN PEMBELIAN 001	Dapat dilakukan	Sesuai Ekspektasi
ADMIN PEMBELIAN 001	Aktor melakukan update pada data faktur pembelian karena ada kesalahan input	Muncul 403 Forbidden	Sesuai Ekspektasi
ADMIN PEMBELIAN 001	Aktor mengakses halaman https://(url)/purchaseOrder/1/print (halaman print pesanan pembelian)	Dapat dilakukan	Sesuai Ekspektasi
ADMIN PENJUALAN 001	Aktor mengakses halaman https://(url)/sale/create (halaman input data baru faktur penjualan)	Dapat dilakukan	Sesuai Ekspektasi
ADMIN PENJUALAN 001	Setelah login, Aktor dapat melihat menu yang ada hanya menu penjualan	Hanya muncul menu penjualan tanpa menu pembelian	Sesuai Ekspektasi
ADMIN PENJUALAN 001	Aktor ingin melakukan penghapusan data yang telah diinput pada retur penjualan	Muncul 403 Forbidden	Sesuai Ekspektasi

3.4. Pembahasan

Hasil penelitian ini membuktikan bahwa penerapan RBAC sangat mungkin dilakukan pada sistem yang telah dibangun menggunakan *framework* Laravel dengan memanfaatkan *library* Laravel Spatie. Dengan

penerapan yang ada, setiap peran dalam sistem dapat mengakses menu (dalam hal ini modul) sesuai dengan kebutuhan dan tidak dapat mengakses halaman lain yang tidak berhubungan dengan perannya. Penelitian ini juga telah menghasilkan interface yang dapat digunakan oleh *Super Administrator* untuk mengatur user dan role-nya sedemikian rupa sehingga memudahkan kedinamisan pengaturan hak akses serta meningkatkan fleksibilitas dan skalabilitas sistem nantinya berhubungan dengan peningkatan jumlah *user* maupun *role* nantinya.

Dari penelitian ini, tujuan bisnis yang ingin dicapai tidak akan terganggu dengan adanya penerapan library yang ada dan tidak mengganggu pengembangan sistem yang berjalan terutama pada sistem yang menggunakan *framework* sehingga sistem tetap dapat mendukung sustanbilitas perusahaan dalam mencapai tujuan bisnis [19]. Selain itu, penerapan RBAC menggunakan library Laravel Spatie memungkinkan sistem untuk dikembangkan lebih lanjut karena telah berisi penerapan mendetil dan memungkinkan pengaturan akses yang lebih dinamis baik untuk sistem yang sederhana maupun sistem kompleks sekalipun seperti SCM. Penelitian ini juga memotret lebih jauh penerapan RBAC lainnya selain melalui API seperti pada penelitian yang dilakukan pada sistem kependudukan [21] ataupun pada level database [22]. Penerapan ini juga sekaligus memberikan gambaran pengaturan akses dinamis melalui *interface* yang dapat dilakukan oleh *end user*.

4. KESIMPULAN

Penerapan RBAC pada sistem SCM berbasis *cloud* sangat dapat dilakukan terutama dalam hal menangani sistem yang memiliki banyak level *stakeholder* akses dengan berbagai peran yang ada. Penerapan RBAC dalam sistem SCM yang dibangun menggunakan *framework* Laravel dengan menggunakan *library* Laravel Spatie dapat mendukung fleksibilitas dan skalabilitas sistem. Ketercapaian tujuan dari penelitian ini sekaligus menjadi keunggulan dalam menjawab tantangan keamanan data dan menjaga integritas otorisasi setiap peran stakeholder.

Hal yang dapat dikembangkan selanjutnya adalah mengukur efektivitas dan efisiensi penggunaan *library* maupun komparasinya terhadap beberapa model RBAC yang ada sehingga didapatkan model RBAC modern yang sesuai untuk sistem SCM yang kompleks. Penelitian ini masih ada dibatasan penerapan model RBAC yang dilakukan di level *framework* dan belum menyentuh sampai ke bagian lainnya yang berintegrasi dengan *framework* yang ada. Keamanan data pada sistem berbasis *cloud* juga perlu diperhatikan aspek lainnya selain dalam hal *development system* tetapi juga jaringan maupun *database system*. Penerapan model RBAC modern juga perlu dipertimbangkan dalam rangka meningkatkan fleksibilitas dan skalabilitas sistem.

REFERENCES

[1] E. M. Frazzon, C. M. T. Rodriguez, M. M. Pereira, M. C. Pires, and I. Uhlmann, "TOWARDS SUPPLY CHAIN MANAGEMENT 4.0," *Brazilian J. Oper. Prod. Manag.*, vol. 16, no. 2, pp. 180–191, Jun. 2019, doi: 10.14488/BJOPM.2019.v16.n2.a2.

[2] S. Harnal and R. K. Chauhan, "Efficient and Flexible Role-Based Access Control (EF-RBAC) Mechanism for Cloud," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 7, no. 26, pp. 1–10, 2020, doi: 10.4108/eai.13-7-2018.161438.

[3] H. Setyodewi, M. A. A. Rizqi, T. W. Adha, N. Ibrahim, and M. Y. Fathoni, "Inovasi Digital Model Rantai Pasok Pada Futuristik E-Commerce," *J. Inform. J. Pengemb. IT*, vol. 6, no. 3, pp. 194–202, 2021.

[4] K. Katircioglu, T. Brown, and M. Asghar, "An SQL-based cost-effective inventory optimization solution," *IBM J. Res. Dev.*, vol. 51, no. 3–4, pp. 433–445, 2007, doi: 10.1147/rd.513.0433.

[5] F. Men, R. M. S. Yaqub, R. Yan, M. Irfan, and A. Haider, "The impact of top management support, perceived justice, supplier management, and sustainable supply chain management on moderating the role of supply chain agility," *Front. Environ. Sci.*, vol. 10, Jan. 2023, doi: 10.3389/fenvs.2022.1006029.

[6] M. Nugraha, R. Agus, H. Fathi, and M. R. Baginda, "Development a Web-based Student Internship Application Using Laravel Framework & Waterfall Model," *J. Inf. Technol. ITS Util.*, vol. 6, no. 1, pp. 31–38, 2023.

[7] F. I. (Mrs) Oyeyinka, O. J. Omotosho, and I. K. (Ph. D. Oyeyinka, "A Modified Things Role Based Access Control Model for Securing Utilities in Cloud Computing," *I nternational J. I nnovative Resear ch I nformation Secur.*, vol. 5, pp. 21–25, 2015, [Online]. Available: www.ijiris.com

[8] S. Dubey and P. K. Rai, "A Review of Cloud Service Security with Various Access Control Methods," *Int. J. Comput. Sci. Mob. Comput.*, vol. 10, no. 3, pp. 39–45, Mar. 2021, doi: 10.47760/ijcsmc.2021.v10i03.005.

[9] D. Salunke *et al.*, "A Survey Paper on Role Based Access Control," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 2, 2013, [Online]. Available: https://www.researchgate.net/publication/368755664

[10] M. D. Karumanchi, S. Immanuvelrajkumar, and P. Devaneyan, "Cloud Based Supply Chain Management System Using Blockchain," in *4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques, ICEECCOT 2019*, Dec. 2019, pp. 390–395. doi: 10.1109/ICEECCOT46775.2019.9114692.

[11] P. R. Chavan and S. Pawar, "Comparison Study Between Performance of Laravel and Other PHP Frameworks," *Int. J. Res. Eng. Sci. Manag.*, vol. 4, no. 10, pp. 27–29, 2021.
[12] S. Talegaon and R. Krishnan, "Administrative Models for Role Based Access Control in Android," *J. Internet Serv. Inf. Secur.*, vol. 10, no. 3, pp. 31–46, Aug. 2020, doi: 10.22667/JISIS.2020.08.31.031.
[13] A. Asyahdina, E. Krisnanik, and R. Wirawan, "Rancang Bangun Supply Chain Management Budidaya Jamur Berbasis Web (Studi Kasus: Budidaya Jamur Jatayutm)," in *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, 2021, pp. 91–98.
[14] H. Hermawaty, R. Siswanto, P. Budi Santoso, and G. D. Ramady, "Rancang Bangun E-Supply Chain Management (SCM) pada Perusahaan Kacang Tanah Berbasis Website," *J. Isu Teknol. Sekol. Tinggi Teknol. Mandala*, vol. 16, no. 2, pp. 1–7, 2021.
[15] A. D. Bachtiar, A. Susilo, and Y. Amrozi, "Model Rancang Bangun Sistem Informasi SCM pada Industri Tekstil," *J. Method.*, vol. 7, no. 2, pp. 31–35, 2021.
[16] I. P. Somadanayasa, D. Putra Githa, A. A. Ngurah, and H. Susila, "Rancang Bangun Supply Chain Management Pada Pia Cemerlang Berbasis Website," *JITTER-Jurnal Ilm. Teknol. dan Komput.*, vol. 3, no. 1, 2022.
[17] S. Monalisa and D. Apsyarin, "Rancang Bangung Sistem Informasi Supply Chain Management Distribusi Barang dan Jasa Berbasis Web," *J. Ilm. Rekayasa dan Manaj. Sist. Inf.*, vol. 7, no. 2, pp. 138–144, 2021.
[18] A. Rohmah Zaidah *et al.*, "RANCANG BANGUN WEBSITE SISTEM INFORMASI SUPPLY CHAIN MANAJEMEN PADA BISNIS PERGURUAN TINGGI," *J. Method.*, vol. 6, no. 1, pp. 28–33, 2020.
[19] A. Kousalya and N. kyun Baik, "Enhance Cloud Security and Effectiveness Using Improved RSA-Based RBAC with XACML Technique," *Int. J. Intell. Networks*, vol. 4, pp. 62–67, Jan. 2023, doi: 10.1016/j.ijin.2023.03.003.
[20] M. A. C. Gyver, M. Marhaeni, and H. Arrang, "Design and Development of Educational Management Information Systems in Web-Based Qur'an Home with Implementation Role-Based Access Control," *J. Rekayasa Inf.*, vol. 12, no. 2, 2023.
[21] Rubiyanto and W. Selo, "Implementasi Role-Based Access Control (RBAC) pada Pemanfaatan Data Kependudukan di Tingkat Kabupaten," *Semin. Nas. Sains dan Teknol.*, pp. 1–10, 2017.
[22] A. Y. Arif, E. Utami, and S. Raharjo, "Implementasi Database Security Menggunakan Konsep Role-Based Access Control (RBAC) dalam Rancangan Database Sistem Informasi Manajemen Sekolah dengan PostgreSQL," *J. Inf. Interaktif*, vol. 4, no. 1, pp. 51–55, 2019, [Online]. Available: http://e-journal.janabadra.ac.id/
[23] "Spatie," *https://spatie.be/docs/laravel-permission/v6/introduction*. https://spatie.be/docs/laravel-permission/v6/introduction (accessed Dec. 30, 2023).