

SISTEM PEMBARUAN STOK BARANG PADA TOKO PAKAIAN MENGGUNAKAN TEKNOLOGI RFID

Elysia Levina, Resmana Lim, Petrus Santoso
Program Studi Teknik Elektro, Universitas Kristen Petra
Jl. Siwalankerto No. 121-131, Surabaya, 60236, Indonesia

E-Mail: elysiaalevinaa13@gmail.com, resmana@petra.ac.id, petrus@petra.ac.id

Abstrak – Pendataan barang pada toko ritel penting untuk mengetahui status ketersediaan barang, dengan adanya pendataan stok barang, maka pencarian barang akan lebih mudah. Pencatatan keluar juga perlu dilakukan dengan cepat agar tidak mengganggu kegiatan jual beli, sehingga sistem diharapkan dapat mengelola stok barang yang dapat diakses dengan mudah dan dapat dilakukan dengan cepat. Website berfungsi sebagai *interface* untuk menampilkan data yang ada pada *database*. USB RFID reader digunakan sebagai alat untuk membaca RFID. Pembuatan website menggunakan node js sebagai *framework backend* dan vue js sebagai *framework frontend*. Akses website dibedakan menjadi dua jenis berdasarkan fiturnya, yaitu *storage*, untuk mencatat barang masuk dan *store*, untuk mencatat barang keluar. Pengujian dilakukan dengan menguji keberhasilan komunikasi Node js dengan MySQL menggunakan aplikasi Postman, dan menguji keberhasilan komunikasi *website* dengan *database*. Dari hasil pengujian tersebut, RFID dapat dibaca oleh USB RFID reader, lalu data barang dapat dibaca, ditulis, diedit dan dihapus melalui web.

Kata Kunci— RFID reader, sistem pembaruan stok barang

I. PENDAHULUAN

Pada bisnis ritel, salah satu poin penting dalam kesuksesan membangun bisnis tersebut adalah cara mengelola stok barang[1]. Jika pemilik bisnis ritel tidak bisa mengatur pemasukan dan pengeluaran barang, maka status ketersediaan barang tidak bisa dikontrol. Pengetahuan persediaan stok sangat penting bagi pembeli maupun penjual untuk melakukan proses jual beli. Jika konsumen tertarik pada barang tertentu dan penjual tidak tahu ketersediaan barang, maka proses jual beli tidak akan terjadi. Dalam pencatatan stok barang masuk, hal yang biasanya didata adalah kode barang, jenis barang, tanggal masuk, jumlah barang, dan harga jual. Pendataan barang keluar juga penting untuk mengetahui status ketersediaan barang. Dengan adanya data stok barang yang terbaru secara real time, maka mencari barang untuk dijual lebih mudah.

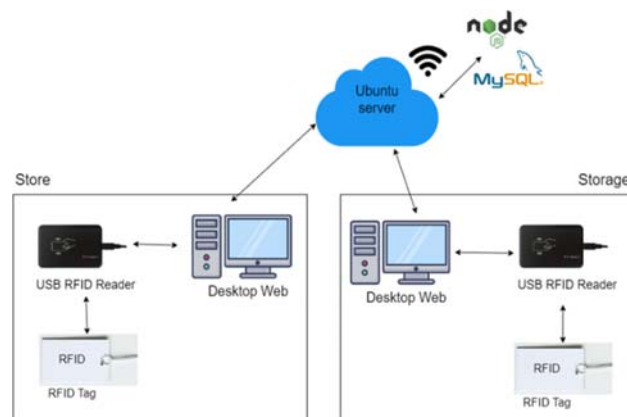
Pencatatan secara manual di kertas dapat dilakukan satu persatu, tetapi untuk mencari data yang ada, dan melakukan pencatatan barang masuk dan keluar dapat menimbulkan human error. Human error yang bisa dihadapi misalnya seperti salah input jenis barang, ukuran, harga dan lain lain, sehingga mengurangi efektivitas dan efisiensi. Atau bisa terjadi kerusakan pada kertas, misalnya robek atau tulisan tidak bisa dibaca.

Penggunaan RFID untuk pencatatan stok barang sudah banyak ditemui di bisnis ritel. RFID mempermudah pencatatan keluar masuknya barang. Dengan RFID, pencatatan dapat dilakukan dengan lebih akurat. Hal ini dapat menjadi solusi dari adanya kesalahan pencatatan dan dapat meningkatkan efektivitas dan

efisiensi dalam pengelolaan stok barang. RFID yang dipasang pada barang dapat memuat berbagai macam data yang disimpan ke dalam database. Data tersebut kemudian dapat ditambahkan, diubah, atau dihapus kapan saja sesuai kebutuhan. Dengan adanya RFID, pencatatan barang masuk dapat dilakukan secara online sehingga data yang didapatkan dapat berasal dari dua tempat atau lebih yang berbeda. Dengan adanya sistem ini, pencatatan barang masuk dan barang keluar dapat dilakukan dengan lebih mudah dan praktis.

II. PERANCANGAN SISTEM

Berikut adalah gambar blok diagram sistem secara keseluruhan



Gambar 1. Diagram Blok Sistem

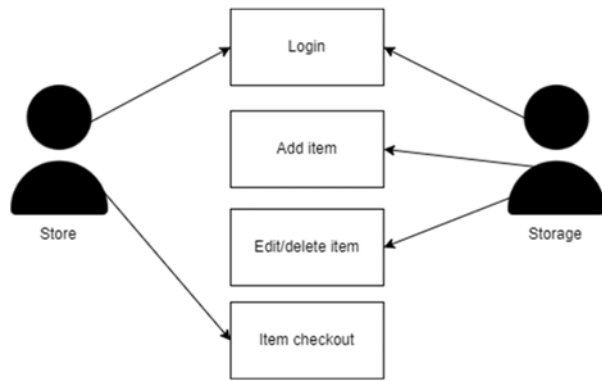
Pada gambar 1, terdapat dua bagian, yaitu gudang atau storage dan toko atau store. Pada usaha kelas kecil menengah, barang masuk dan keluar dilakukan di satu tempat, sehingga hanya perlu menggunakan satu database. Bagian storage dapat melakukan pencatatan barang masuk sedangkan pada store dapat melakukan pencatatan barang keluar.

Data pada tag RFID dapat dibaca menggunakan USB RFID reader yang disambungkan ke PC. Selanjutnya data disimpan ke dalam database penyimpanan menggunakan MySQL. Akses interface berupa web menggunakan node js sebagai framework backend dan vue js sebagai framework frontend.

Terdapat dua jenis akses untuk user role yang diberikan pada web.

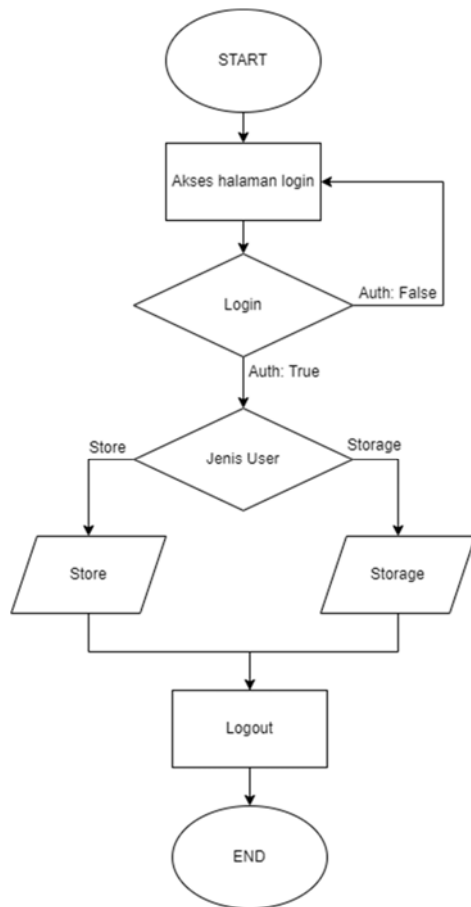
- 1) Store, bagian barang keluar
- 2) Storage, bagian barang masuk

Berikut adalah diagram akses pengguna pada website yang dibedakan berdasarkan fiturnya:



Gambar 2. Diagram Akses Pengguna pada Website

RFID tag dibaca oleh RFID reader dan di *input* bersama dengan informasi barang lainnya. Selanjutnya data barang tersebut disimpan kedalam *database*. Data yang telah tersimpan selanjutnya diolah dan ditampilkan ke dalam web. Berikut adalah flowchart untuk akses website:



Gambar 3. Flowchart akses penggunaan website

Proses pembuatan sistem ini dimulai dengan mengakses website, lalu pengguna diarahkan untuk melakukan proses *login* di halaman *login*. Setelah *login*, jika pengguna merupakan pengguna dengan akses *store*, maka pengguna akan diarahkan pada halaman *store*, dan begitupun juga jika pengguna merupakan pengguna dengan akses *storage*. Pengembangan *server* dilakukan menggunakan Node Js, MySQL sebagai *database*, dan Vue.Js sebagai *framework*

interface berbasis Javascript, dan digunakan Express Js sebagai *routing* dan HTTP *request*. Pemrograman dilakukan menggunakan aplikasi *text editor* Visual Studio Code. Pembuatan *database* dilakukan menggunakan *sequelize*. *Sequelize* berfungsi sebagai ORM(*Object Relational Mapping*) yang mengatur *database* dan relasi sekitar *database*[2]. Terdapat 2 tabel yang digunakan pada program, yaitu tabel *user* dan tabel *barang*. Pada tabel *user*, disimpan data *username* dan *password* pengguna yang digunakan untuk melakukan *login* dan autentikasi sesuai role pengguna. Tabel *barang* berisi informasi barang, yaitu nama barang, jenis barang, warna barang, ukuran barang, jumlah barang, tanggal barang masuk dan tanggal barang keluar. Berikut adalah tabel *database* yang digunakan:

Tabel 1. Tabel Database User

No.	Field	Type
1.	ID	Varchar
2.	Username user	Varchar
3.	Password user	Varchar
4.	Jenis user	Varchar

Tabel 2. Tabel Database Barang

No.	Field	Type
1.	ID	Int
2.	Nama_barang	Varchar
3.	Jenis_barang	Varchar
4.	Warna_barang	Varchar
5.	Ukuran_barang	Varchar
6.	Jumlah_barang	Varchar
7.	Tmasuk_barang	Varchar
8.	Tkeluar_barang	Varchar
9.	createdAt	Datetime
10.	updatedAt	Datetime

Aplikasi *website storage* memiliki layanan sebagai berikut:

1. *Login/logout*
2. Melihat seluruh barang
3. Mencari barang tertentu
4. Menambahkan barang baru
5. Menyunting barang
6. Menghapus barang

Sedangkan layanan pada *store* adalah sebagai berikut:

1. *Login/logout*
2. Melihat seluruh barang
3. Mencari barang berdasarkan RFID tag
4. Menambahkan keterangan barang keluar beserta tanggal keluar

Website mengirimkan *request* layanan kepada *API gateway*, yang selanjutnya mengirimkan atau menerima data yang disimpan di *database* sesuai dengan metode yang disediakan. Pengembangan dan pengecekan REST API dilakukan menggunakan aplikasi *postman*. Berikut adalah daftar *gateway API*:

Sistem Pembaruan Stok Barang
[Elysia et al]

Tabel 3. Daftar API Gateway

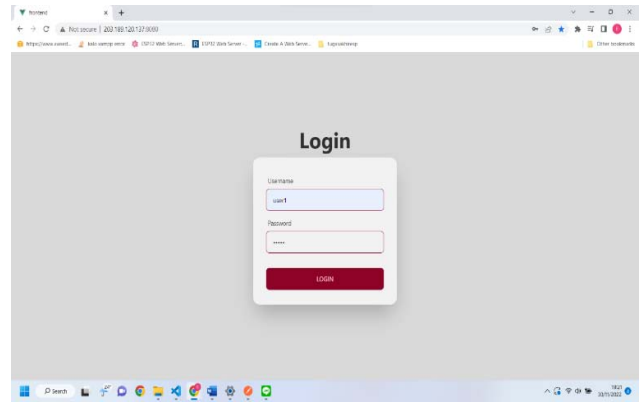
No.	API	Alamat	Keterangan
1.	Get	/penyimpanan/ tampilkan_barang _semua	Untuk menampilkan seluruh data pada <i>database</i>
2.	Get	/penyimpanan/ lihat_barang_by_id:id	Untuk melihat barang berdasarkan ID
3.	Post	/penyimpanan/ masukan_barang	Untuk memasukan barang ke dalam <i>database</i>
4.	Put	/penyimpanan/ update_barang:id	Untuk mengedit data yang ada berdasarkan ID
5.	Delete	/penyimpanan/ hapus_barang:id	Untuk menghapus data berdasarkan ID
6.	Post	/pengguna/ loginservice	Mengirimkan data login ke <i>database</i>
7.	Post	Pengguna/ masukan_user	Menambahkan user baru kedalam <i>database</i>
8.	Post	/sortir/by/tagbarang	Mengambil data berdasarkan tag barang tertentu
9.	Post	/sortir/by/namabarang	Mengambil data berdasarkan nama barang tertentu
10.	Post	/sortir/by/jenisbarang	Mengambil data berdasarkan jenis barang tertentu
11.	Post	/sortir/by/warnabarang	Mengambil data berdasarkan warna barang tertentu
12.	Post	/sortir/by/tanggalbarang	Mengambil data berdasarkan tanggal masuk barang tertentu

Tampilan website dibuat menggunakan Vue.js yang menggunakan bahasa pemrograman *javascript*. Gambar 4, 5 dan 6 adalah beberapa tampilan website.

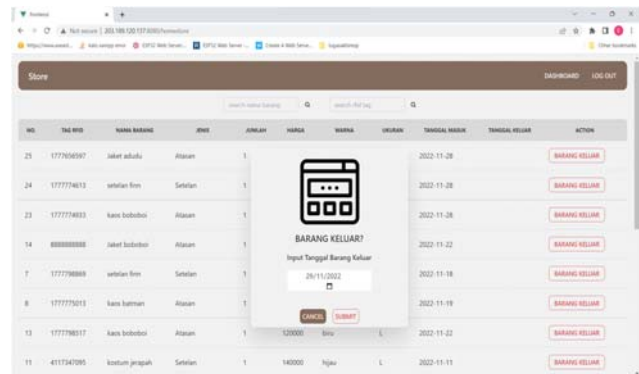
III. PENGUJIAN SISTEM DAN ANALISA

Pada bagian ini dilakukan pengujian terhadap sistem menggunakan 2 metode, yaitu:

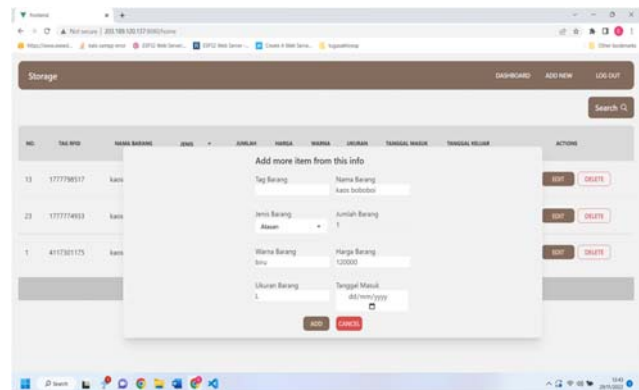
1. Pengujian keberhasilan komunikasi Node js dengan MySQL menggunakan aplikasi Postman
2. Pengujian keberhasilan komunikasi website dengan database.



Gambar 4. Tampilan Halaman Login



Gambar 5. Tampilan Halaman Store



Gambar 6. Tampilan Halaman Storage

A. Pengujian keberhasilan komunikasi Node js dengan MySQL menggunakan aplikasi Postman

Pengujian ini bertujuan untuk mengetahui jika node js dapat menerima dan mengirimkan data ke *database*. Pengujian dilakukan menggunakan aplikasi Postman. Postman mengirim permintaan API ke server dan menerima respons. Berikut adalah hasil pengujian API yang dilakukan menggunakan aplikasi Postman:

Tabel 4. Hasil Pengujian API

No.	Pengujian API	Hasil
1.	Mengambil semua data pada <i>database</i>	Berhasil
2.	Memasukan barang ke <i>database</i>	Berhasil
3.	Merubah nama dan jenis barang	Berhasil
4.	Menghapus barang berdasarkan id	Berhasil
5.	Mencari barang berdasarkan id/ RFID/ nama/ jenis/ warna/ tanggal	Berhasil
6.	Mencari barang berdasarkan id	Berhasil
7.	Mencari barang berdasarkan RFID	Berhasil
8.	Mencari barang berdasarkan nama barang	Berhasil
9.	Mencari barang berdasarkan jenis barang	Berhasil

B. Pengujian keberhasilan komunikasi website dengan *database*.

Pengujian ini bertujuan untuk mengetahui jika website dapat berkomunikasi dengan *database*.

Tabel 5. Hasil pengujian komunikasi website dengan *database*

No.	Pengujian Layanan Website	Hasil
1.	<i>Login, logout</i>	Berhasil melakukan login dan logout
2.	Menampilkan data <i>database</i> ke dalam tabel <i>store</i> dan <i>storage</i>	Berhasil menampilkan data ke dalam tabel <i>store</i> dan <i>storage</i>
3.	Mencari data barang berdasarkan tag RFID dan menampilkan hasil ke dalam tabel <i>store</i>	Berhasil melakukan pencarian berdasarkan RFID tag dan menampilkan ke dalam tabel <i>store</i>
4.	Pengujian barang keluar pada bagian <i>store</i>	Berhasil melakukan pendataan barang keluar
5.	Pengujian menambahkan data berdasarkan data yang sudah ada	Berhasil menambahkan data dari data yang sudah ada
6.	Pengujian mengedit data barang melalui tabel <i>storage</i>	Berhasil mengedit data
7.	Pengujian menghapus data dari <i>database</i> melalui tabel <i>storage</i>	Berhasil menghapus data
8.	Pengujian menambahkan barang baru pada <i>storage</i> .	Berhasil menambahkan barang baru
9.	Pengujian ketika pada tag yang sama terdapat dua barang yang berbeda	Kedua barang tampil dengan urutan barang yang belum keluar diatas
10.	Pengujian ketika terdapat barang dengan nama yang sama lebih dari satu	Barang-barang tersebut tampil dengan urutan barang yang belum keluar diatas

IV. KESIMPULAN

Kesimpulan yang dapat diambil sebagai berikut.

- 1) Akses *database* dapat dilakukan di *website* melalui jaringan internet dengan fitur-fitur seperti menampilkan, menambahkan, mengedit, menghapus data barang dan melakukan pencatatan barang keluar dapat berfungsi dengan baik. Pembaca RFID yang digunakan yaitu USB RFID *reader* sehingga dapat digunakan pada device lain seperti laptop, *smartphone*, *tablet*, Raspberry pi dan komputer. Raspberry pi tidak digunakan pada project ini, karena memerlukan lebih banyak alat untuk mengoperasikan pembacaan RFID tag, seperti modul RC522, dan Raspberry pi atau mikrokontroler, sehingga pada project ini, lebih baik jika USB RFID *reader* digunakan sebagai alat pembacaan RFID tag dan menggunakan server ubuntu sebagai server.
- 2) Terdapat kekurangan pada sistem pengeluaran barang, jika terdapat tag barang yang sudah keluar dan tag tersebut digunakan kembali untuk barang lain, maka data barang yang sudah keluar akan tetap muncul dan tetap dapat melakukan layanan seperti barang lainnya, sehingga kemungkinan terjadi kesalahan *input* data.

DAFTAR PUSTAKA

- [1] Folio. (2017). *Folio*. Retrieved from Foliopos.com: <https://www.foliopos.com/blog/detail/panduan-cara-mengelola-stok-toko-retail>
- [2] Alfian, G. (2018). *Membuat REST API dengan Express Js dan Sequelize*. Retrieved from medium: <https://medium.com/@gustialfianmp/membuat-rest-api-dengan-express-js-dan-sequelize-105c33db80c5>