PERANCANGAN SISTEM INFORMASI BERBASIS RESPONSIVE WEB SERVICE API DENGAN ALGORITMA RSA DAN RFC 7519 PADA PT. CAHAYA GEMILANG ABADI.

Mohamad Seh Fahrudin [1], Ferdiansyah [2]
[1,2]Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
e-mail: [1]msfahrudin.msf@gmail.com, [2]ferdiansyah@budiluhur.ac.id

Abstrak

Dalam Penyediaan Teknologi Informasi pada sebuah perusahaan saat ini merupakan sebuah kebutuhan paling penting. Salah satu teknologi yang sering dilakukan perusahaan adalah teknologi web service. Sebuah website dapat menjadi solusi yang dapat memudahkan perusahan dalam mengembangkan bisnisnya. Untuk mendukung sistem operational yang baik maka dibuatlah sebuah aplikasi dengan menggunakan teknologi RESTful web service yang diterapkan pada aplikasi berbasis web. Dan untuk sisi keamanan pada RESTful web service menggunakan gabungan Algoritma RSA Dan RFC 7519. Harapannya dari hasil penelitian ini dapat membuat sistem produksi bisa dilakukan dengan tepat waktu, lebih akurat dan lebih terintegrasi lagi.

Kata Kunci: *RESTful web service, Algoritma RSA, RFC 7519*

1. PENDAHULUAN

Perkembangan teknologi menggunakan *web service* dalam sistem informasi merupakan salah satu sarana yang baik. Teknologi ini merupakan mekanisme komunikasi dua aplikasi, terlepas dari arsitektur yang memiliki antar muka. Implementasinya yang dapat digunakan secara terpisah dari platform perangkat keras atau perangkat lunak yang digunakannya menjadikannya sebagai teknologi yang mandiri. Kemandirian ini mendorong aplikasi untuk *loosely coupled*, berorientasi secara komponen dan implementasi lintas-teknologi.

PT. Cahaya Gemilang Abadi merupakan perusahaan aksesoris otomotif berskala nasional yang berkantor di Jakarta Pusat. Perusahaan merupakan vendor resmi yang menyuplai aksesoris kendaraan beroda empat seperti produk mobil Toyota, Daihatsu yang masuk ke negara Indonesia. Perusahaan ini sudah didirikan sejak tahun 1989, berawal dari bengkel kecil di daerah kemayoran yang hanya menerima pengerjaan bongkar pasang untuk pemesanan aksesoris mobil dan berkembang hingga membuat inovasi produk aksesorisnya sendiri.

Untuk perusahaan aksesoris dapat bersaing dalam menjalankan bisnisnya, sebuah *website* bisa menjadi solusi yang dapat memudahkan perusahaan dalam mengembangkan bisnisnya dan kemudahan saat melayani konsumen. Dalam rangka menjalankan bisnisnya, perusahaan memerlukan sistem yang dapat digunakan di mana saja dan di manapun, selama ini perusahaan belum memiliki sistem apapun yang bahkan dapat memberikan laporan dalam setiap transaksinya. Berdasarkan kendala yang dimiliki, perusahaan mengalami kurang optimalnya dalam pelayanan dan manajemen perusahaan, atas seluruh masalah tersebut penulis[1].

2. TINJAUAN PUSTAKA

2.1 Sistem Pembelian, Penjualan dan Produksi

Sistem Informasi produksi merupakan sistem yang digunakan untuk mendukung fungsi produksi yang mencakup seluruh kegiatan terkait dengan perencanaan dan pengendalian proses produksi barang atau jasa. Jadi sistem informasi produksi merupakan sistem informasi manajemen yang menyediakan informasi yang mengenai kegiatan terkait dengan perencanaan dan pengendalian proses untuk memproduksi barang atau jasa[2].

2.2 HTML (Bootstrap)

Hypertext Markup Language (HTML) adalah bahasa untuk mengirimkan konten di *Web*. HTML tidak dimilikimoleh siapa pun, tetapi merupakan hasil dari orang yang bekerja di banyak negara dan banyak organisasi untuk mendefinisikannya fitur bahasa. Dokumen HTML adalah dokumen teks yang dapat Anda hasilkan menggunakan teks apa pun editor. Dokumen HTML berisi elemen yang dikelilingi oleh tag — teks yang dimulai dengan simbol <dan berakhir dengan simbol>. Contoh tag adalah . Tag khusus ini akan menampilkan gambar disimpan di file *home.gif*. Tag ini adalah markup. Melalui penggunaan tag itulah hyperlink, gambar,dan media lain termasuk dalam halaman *web*[3]**.**

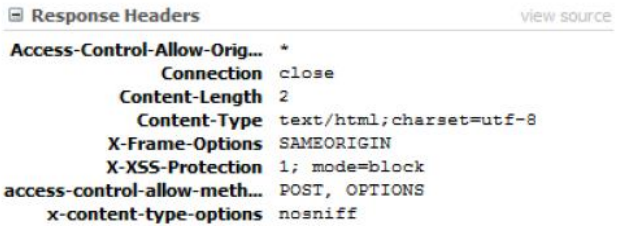
2.3 Algoritma RSA

Mengungkapkan RSA yang didirikan pada tahun 1977 adalah *cryptosystem* kunci publik. RSA adalah algoritma kriptografi asimetris yang dinamai setelah pendirinya Rivest, Shamir & Adelman. Ini menghasilkan dua kunci: kunci publik untuk enkripsi dan kunci pribadi untuk mendekripsi pesan. Algoritma RSA terdiri dari tiga langkah, langkah satu adalah pembuatan kunci yang akan digunakan sebagai kunci untuk mengenkripsi dan mendekripsi data, langkah kedua adalah enkripsi, di mana proses aktual konversi *plaintext* ke teks *cipher* sedang dilakukan dan langkah ketiga adalah dekripsi, di mana teks terenkripsi dikonversi menjadi teks biasa di sisi lain[4].

2.4 RESTFul Web Service

Gagasan utama dari *REST* sebagai komponen dari aplikasi yang perlu digunakan atau dialamatkan. *REST WS* cara yang lebih ringan dan sederhana, dan berfokus pada sumberdaya. Terdapat dua bagian pesan yang digunakan untuk membangun komunikasi dengan *server* yaitu pesan *Header* dan pesan *Body*, *HTTP Header*, yang umum meliputi *header request* diilustrasikan pada Gambar 1, *header response* pada Gambar 2, dan terdapat bidang entitas-*header*. Setiap *request* sumberdaya dari masing-masng *client* dapat dikendalikan dengan memanfaatkan *HTTP Header*. Setiap kolom *header* terdiri dari nama diikuti dengan titik dua (":") atau *white space* dan konten *field*. Nama *field* bersifat *case-sensitive*. *Header* berisikan semua informasi yang diperlukan untuk mengumpulkan metode *request* dan respon[5].

Gambar 1. *Header Request*



Gambar 2. *Header Response*

2.5 JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dandibuat *(generate)* oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman *JavaScript*, Standar *ECMA-262* Edisi ke-3 - Desember 1999.*JSON* merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk *C, C++, C#, Java, JavaScript, Perl, Python* dll. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran-data[6]. *JSON* terbuat dari dua struktur:

a. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object)*, rekaman (*record)*, struktur *(struct)*, kamus *(dictionary)*, tabel *hash (hash table)*, daftar berkunci *(keyed list)*, atau *associative array.*
b. Daftar nilai terurutkan *(an ordered list of values).* Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik *(array)*, daftar *(list)*, atau urutan *(sequence).*

2.6 RFC 7519/*JSON Web Token*

RFC 7519 atau yang disebutkan sebagai *JSON Web Token* adalah sebuah *token* berbentuk *string* panjang yang sangat *random* yang gunanya sendiri untuk melakukan sistem Autentikasi dan Pertukaran Informasi. Umumnya untuk melakukan *login* tidak seperti pada aplikasi *website* biasa dimana kita menggunakan *session* untuk mengingat siapa yang sedang *Login*. Tapi didalam API sendiri kita menggunakan konsep *JWT* atau dibacanya sebagai "jot"[7].

Gambar 3. *Header JWT*

Header hanya terdiri dari Algoritma HS256 yang kita gunakan dan *Type*nya *jwt* sebagai *defaultnya*[8].

PAYLOAD: DATA


```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

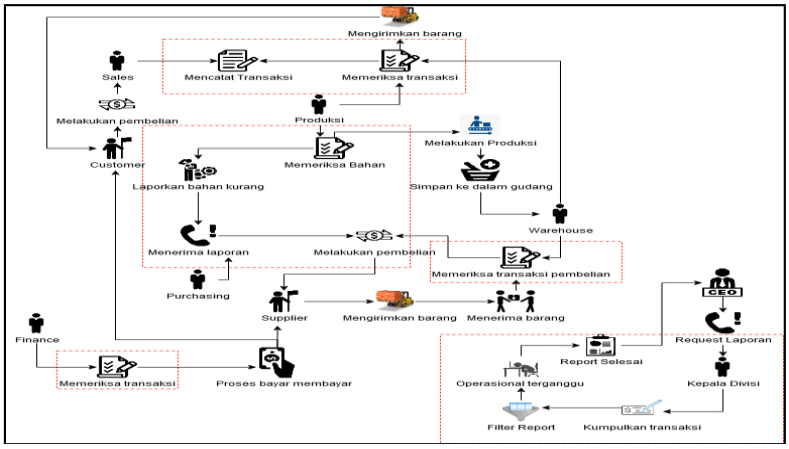

Gambar 4. *Payload JWT*

*Payload*nya sebagai informasi atau data yang ingin kita kirimkan untuk *users* misalnya *id usernya* atau tanggal *expired* nya dan lain-lain[8].

3. METODE PENELITIAN

3.1 Analisa Masalah

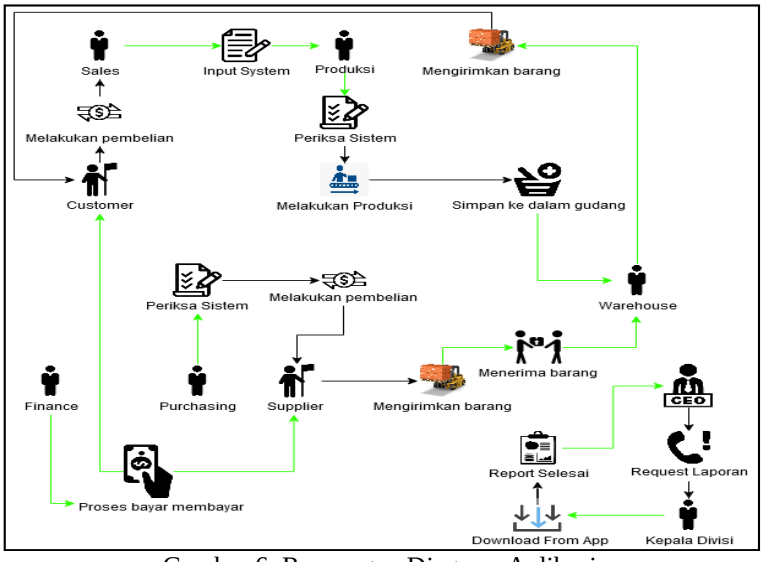
Berdasarkan hasil *observasi* dan wawancara oleh peeliti pada PT. Cahaya Gemilang Abadi. Perusahaan mengalami kendala dalam segi transaksi manual, kesulitan saat mengetahui sisa pelunasan. Pelaporan yang akan memakan *resource* karyawan. Berdasarkan Analisa masalah yang ada, penulis akan jelaskan dalam bentuk *Rich Picture Diagram:*



Gambar 5. Masalah Sekarang

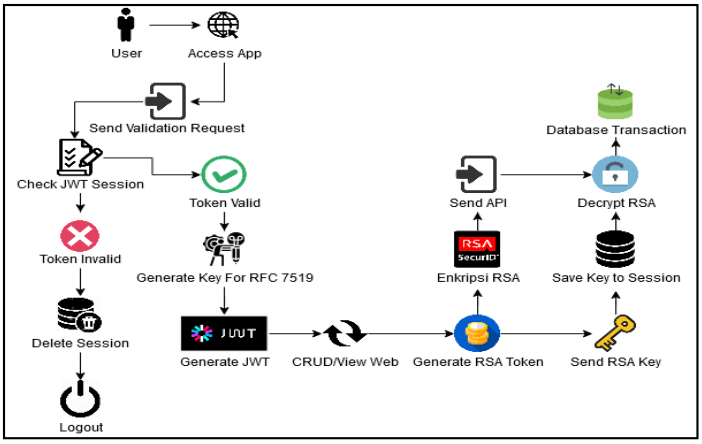
3.2 Rancangan Aplikasi

Berdasarkan masalah yang didapat, penulis akan merancang sistem yang akan mengatasi beberapa masalah yang dimiliki sebelumnya. Solusi ini juga akan memberikan efisiensi yang lebih baik dan mengoptimalkan tingkat kerja yang dibutuhkan oleh perusahaan. Berikut adalah *rich picture diagram* solusi atas masalahnya:



Gambar 6. Rancangan Diagram Aplikasi

Berikut adalah cara yang digunakan oleh peneliti untuk meningkatkan keamanan dari rancangan aplikasi yang telah dibuat dengan menggunakan gabungan dari *RFC 7519/JWT* Dan *Algoritma RSA*. Seperti dijelaskan pada gambar dibawah ini:



Gambar 7. Diagram Kerja RFC7519/*JWT* Dan Algoritma RSA

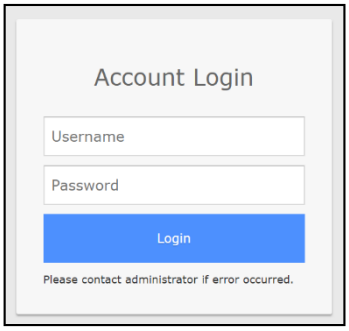
3.3 Pengujian Sistem

Pada tahap selanjutnya pengujian sistem dilakukan bertujuan untuk mengetahui seberapa efektif dan aman autentikasi yang telah dibuat. Pada ujicoba simulasi untuk mengetahui apakah sistem dapat bekerja dengan baik dan tidak, dan hasilnya apakah sudah sesuai dengan keinginan atau masih perlu dilakukan perbaikan.

4. HASIL DAN PEMBAHASAN

4.1 Tampilan Login

Berikut ini adalah tampilan login, yang smuncul setiap kali pengguna ingin akses/menggunakan aplikasi

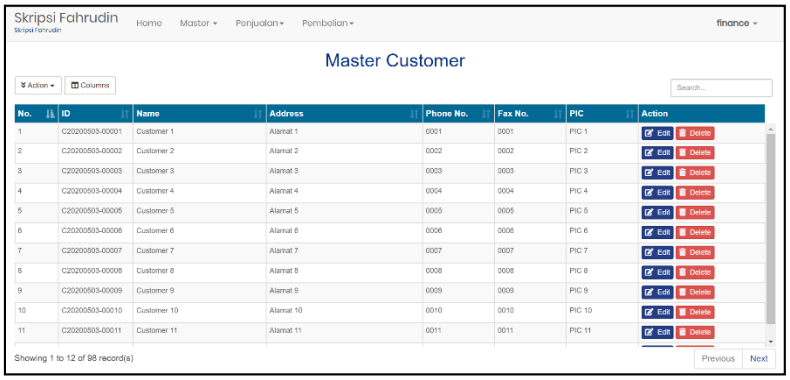


Gambar 8. Tampilan Login

Seorang pengguna harus memasukkan *username dan password* yang sesuai dengan nama bidangnya masing – masing. Jika pengguna salah dalam melakukan pengisian maka aplikasi akan memberi pengeringatan jika data yang dimasukkan tidak sesuai dengan bagiannya masing-masing.

4.2 Tampilan Master Customer

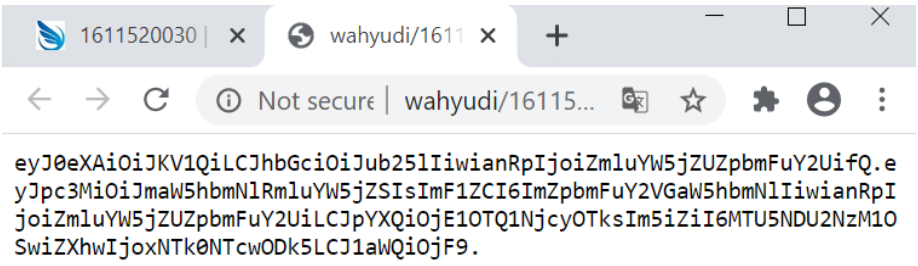
Berikut adalah tampilan *master customer*, tampilan ini hanya dapat diakses oleh *finance, sales, warehouse. Finance* menggunakan data ini untuk mengetahui jumlah *aging account recevable. Sales* menggunakan data ini untuk melakukan penjualan.



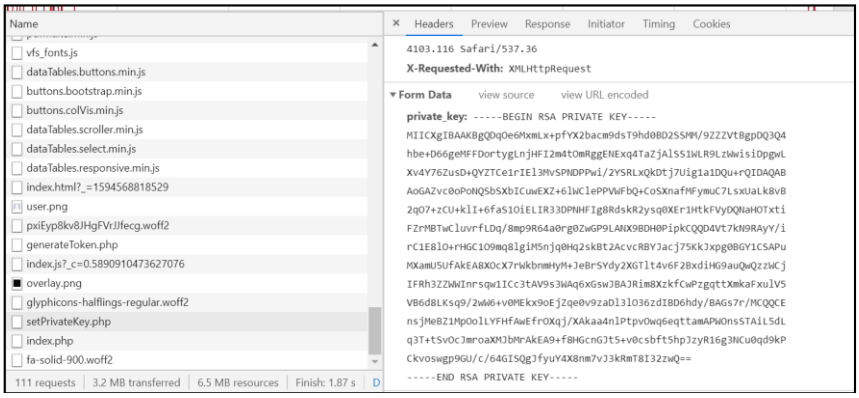
Gambar 9. Tampilan *Web Master Customer*

4.3 Tampilan Keamanan JWT & RSA

Berikut adalah tampilan *token* JWT yang didapatkan oleh pengguna saat pengguna berhasil masuk. Setiap validasi *token* ini berhasil, maka pengguna akan mendapatkan *token* baru untuk memperpanjang masa pengguna aplikasi.



Gambar 10. Tampilan Token JWT



Gambar 11. Tampilan Set Private Key RSA

Gambar diatas merupakan pengiriman *private key* RSA setelah berhasil melakukan pembuatan *token* RSA. Data iini nantinya akan digunakan oleh *server-side* PHP untuk melakukan dekripsi atas data *API* yang dikirimkan dalam bentuk enkripsi. Jika data yang dikirimkan mengalami kesalahan, maka dekripsi akan gagal. Dekripsi yang gagal akan langsung menghentikan seluruh proses transaksi yang akan dilakukan oleh PHP. Hal ini bertujuan untuk memberikan proteksi seperti *inject SQL*.

Table 1. Tabel Uji Sistem

No	Tombol	Ekspetasi	Bisa	Berhasil/Gagal
1	Login	Melakukan Login	☑	Berhasil
2	Master Customer	Melakukan penambahan data	☑	Berhasil
		Melakukan Pengubahan data	☑	Berhasil
		Melakukan penghapusan data	☑	Berhasil
		Melakukan tampilan data	☑	Berhasil
3	Session PHP	Penyimpanan data ke session	☑	Berhasil
4	Token JWT	Membuat token jwt	☑	Berhasil
		Memvalidasi token jwt	☑	Berhasil
5	Algoritma RSA	Membuat private key	☑	Berhasil
		Membuat public key	☑	Berhasil
		Enkripsi RSA pada client side	☑	Berhasil
		Dekripsi RSA pada server side	☑	Berhasil
		Dekripsi tanpa private key	☐	Gagal

Pada table 1. Merupakan hasil ujicoba sistem yang telah dirancang. Sistem ini tentu mempunyai hasil baik dan tidak secara tersendiri jika ditinjau dari kebutuhan pengguna yang beragam dengan kondisi dan situasi berbeda-beda.

5. KESIMPULAN

Berdasarkan evaluasi hasil pengujian aplikasi sistem penjualan produksi dan pembelian berbasis *web service* dan keamanan metode autentikasi RFC-7159 dan algoritma RSA. Maka peneliti mendapatkan kesimpulan bahwa Sistem penjualan produksi dan pembelian yang dibuat akan dapat membantu seluruh proses operational perusahaan dan membuat pekerjaan lebih optimal dan efisien.

Keamanan yang dibuatkan seperti RFC-7519, akan memberikan keamanan kepada pengguna saat melakukan akses aplikasi. Keamanan yang dibuatkan pada RSA, akan meningkatkan keamanan data yang dikirimkan ke *server-side.*Validasi *token* masih belum dioptimalkan karena hanya dilakukan pada setiap halaman yang diakses.

6. SARAN

Berdasarkan kesimpulan yang didapatkan oleh peneliti. Adapun saran yang dapat diberikan agar sistem dapat berjalan lebih baik bahwa Menambahkan fitur cetak laporan setiap transaksi yang saat ini masih belum dimiliki oleh perusahaan. Di mana pengguna tidak perlu melakukan cetak dengan cara *manual.*

Diberikan fitur pembuatan akses aplikasi pada admin. Diberikan pelatihan aplikasi yang dibuatkan oleh peneliti. Membuatkan laporan yang lebih baik, seperti dalam bentuk rangkuman table dan grafik.

DAFTAR PUSTAKA

[1] Achyani, Yuni Eka, Wahyudi, Mochamad, Yusuf, Lestari. 2015. *Sistem Informasi Penjualan Aksesoris Vespa Berbasis Web Pada CV. A.S. Hikmat Motor Bekasi*. Jurnal Sistem Informasi STMIK Antar Bangsa, Vol. IV, No. 2, pp. 185-193.

[2] Imaniawan, Fabriyan Fandi Dwi, Elsa, Umi Maelani. 2017. *Sistem Informasi Penjualan Sepatu Berbasis Web Pada Vegas Hyper Purwokerto*. IJSE Indonesian Journal On Software Engineering, Vol. 3, No. 2, pp. 82-91.

[3] Meyer, Jeanine. 2018. *The Essential Guide to HTML5: Using Games to Learn HTML5 and JavaScript*. New York: Apress.

[4] Patil, Priyadarshini et. all. 2016. *A Comprehesive Evaliation of Cryptographic Algorithm: DES, 3DES, AES, RSA and Blowfish*. Procedia Computer Science, Vol. 78, pp. 617-624.

[5] Tanaem, Penidasa Fiodinggo, Manongga, Danny & Iriani, Ade. 2016. *RESTFul Web Service Untuk Sistem Pencatatan Transaksi Studi Kasus PT XYZ*. Jurnal Teknik Informatika Dan Sistem Informasi, Vol. 2 No. 1, pp. 1-10.

[6] Izquierdo, Javier Luis Canovas & Cabot, Jordi. 2016. *JSON Discoverer: Visualizing the schema lurking behing JSON document*. Knowledge-Based Systems, Vol. 103, pp. 52-55.

[7] Jones, M., Brandley, J. & Sakimura, N. 2015. *JSON Web Token (JWT)*. Internet Engineering Task Force (IETF), pp. 1-30.

[8] Peyrott, Sebastian. 2016. *The JWT Handbook*. Bellevue: Auth0 Inc.

[9] Rachman, Muhammad Aziz Fatchur. 2018. *Perancangan Aplikasi Memo Menggunakan Algoritma Kriptografi Caesar Cipher dan RSA Berbasis Android*. Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri (SENIATI), Vol. 4, No. 2, pp. 121-127.

[10] Mestre, Pedro et. all. 2017. *Securing RESTful Web Services using Multiple JSON Web Tokens*. Proceedings of the World Congress on Engineering 2017, Vol. 1.