Web Service Security System Analysis With Rest Architecture Using The Aes Method With JWT

Raymond Cahyadi Saputra[1]; YB Dwi Setianto[2]
Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata
email: 16k10001@student.unika.ac.id[1]; setianto@unika.ac.id[2]

Abstract

Security on web services is a vital thing. Almost all applications created by using web services as a tool for communication between other application and for simplify the update process. This shows that the security problem of the service presented should not be vulnerable to its level of security. Therefore, the problem to be faced is about how to create and know that the web service is an efficient and not vulnerable. Security does not escape the term called cryptography. Then cryptography is an important thing, especially in the data transfer section in an application. Cryptography that used is the AES method, and with the RESTFUL API architecture in the web service section. Also by using authentication token for account so that not just any account can access the API. The final goal of this research is to analyze the performance of the service provided and also compare the results of the response time that comes out when using the encryption method between by not using it.

Keywords: AES, Encryption, Web Services, REST API, Cryptography, Security

Abstrak

Keamanan pada layanan web merupakan hal yang vital. Hampir semua aplikasi dibuat dengan menggunakan layanan web sebagai alat komunikasi antar aplikasi lain dan untuk mempermudah proses update. Ini menunjukkan bahwa masalah keamanan layanan yang disajikan seharusnya tidak rentan terhadap tingkat keamanannya. Oleh karena itu, permasalahan yang akan dihadapi adalah bagaimana membuat dan mengetahui bahwa web service yang efisien dan tidak rentan. Keamanan tidak luput dari istilah yang disebut kriptografi. Maka kriptografi merupakan suatu hal yang penting terutama pada bagian transfer data dalam suatu aplikasi. Kriptografi yang digunakan adalah metode AES, dan dengan arsitektur RESTFUL API pada bagian web service. Juga dengan menggunakan token otentikasi untuk akun sehingga tidak sembarang akun dapat mengakses API. Tujuan akhir dari penelitian ini adalah menganalisis kinerja layanan yang diberikan dan juga membandingkan hasil waktu respon yang keluar saat menggunakan metode enkripsi antara dengan tidak menggunakannya.

Kata kunci: ***AES, Encryption, Web Services, REST API, Cryptography, Security***

PENDAHULUAN

Keamanan komputer adalah hal yang sangat penting. Selain dari kecanggihan fitur-fitur yang terdapat pada komputer tersebut, keamanan sistem juga sangat berpengaruh pada penggunaan beberapa fitur dan aplikasi yang ada pada komputer tersebut. Ini membuat keamanan menjadi kebutuhan untuk setiap aplikasi di komputer. Banyak sisi yang harus diperhatikan dalam pengamanan suatu aplikasi, salah satunya pada jalur komunikasi atau disebut juga API. API (Application Programming Interface) adalah seperangkat alat canggih yang dapat digunakan pengembang untuk mengintegrasikan dua bagian aplikasi. Tujuan dari pembuatan API ini sendiri adalah untuk mempercepat pembuatan dan pengembangan suatu aplikasi karena

memungkinkan developer untuk mengakses fitur-fitur platform. Formulir API itu sendiri dapat berupa fungsi, metode, atau titik akhir URL. Layanan web / API dibagi menjadi dua jenis, Simple Object Access Protocol (SOAP) dan Representational State Transfer (REST). REST menyatakan bahwa setiap URL unik adalah sebuah objek; objek konten dapat diperoleh dengan menggunakan HTTP GET, dan dapat dimodifikasi dengan menggunakan metode POST, PUT, atau DELETE. Demi keamanan, karena REST menggunakan HTTP atau HTTPS, administrator REST (firewall) dapat melihat maksud dari setiap pesan dengan menganalisis perintah HTTP yang digunakan dalam permintaan. Otentikasi dan otorisasi dari REST sendiri mengasumsikan bahwa urusan ini telah didukung oleh server web. SOAP adalah protokol pertukaran pesan terstruktur dalam implementasi layanan web dan menggunakan Extensible Markup Language (XML). Permintaan SOAP menggunakan perintah POST untuk layanan tertentu. Perintah SOAP itu sendiri adalah perintah yang menghabiskan sumber daya, di mana semua konten permintaan tidak dapat dideteksi oleh sebagian besar firewall. Masih banyak aplikasi yang sistem keamanan layanannya masih kurang. Misalnya aplikasi yang mengirimkan data pribadi tetapi tidak memiliki otentikasi pada layanan tersebut, sehingga pengguna lain dapat mengakses layanan tersebut.

Penelitian ini membahas solusi dari masalah keamanan otentikasi. Dengan memanfaatkan fitur JWT untuk masalah otentikasi yang dikombinasikan dengan metode enkripsi AES pada pesan yang dikirim akan dapat meningkatkan tingkat keamanan layanan yang dibuat.

TELAAH LITERATUR

Deviana (2011), mengimplementasikan web service untuk mendukung integrasi antara platform sistem dan aplikasi. Dataset menggunakan data stok dari apotik pusat dan outlet. Metodenya menggunakan SOAP untuk arsitektur web service. Evaluasi data transaksi dapat dilihat karena dengan menggunakan XML skema dan tipe data dapat dilacak antara client dan server.

Perkasa dan Setiawan (2018), menyampaikan bahwa data dari Dinas Kependudukan dan Pencatatan Sipil belum dimanfaatkan dengan baik dan admin belum memiliki aplikasi untuk memantau penggunaan data publik. Dataset yang digunakan dalam artikel ini menggunakan data komunitas dari Departemen itu sendiri. Metode yang digunakan dalam hal ini adalah menggunakan arsitektur REST untuk layanan dan akses token untuk keamanan dalam aplikasi. Hasil dari metode ini adalah server dapat menangani permintaan dengan cepat. Sayangnya, setiap token akses memiliki batas permintaan data 60 permintaan.

Sibagariang (2016), membahas komunikasi antara server dan client cenderung rentan dalam hal keamanan, akselerasi, efektifitas, dan efisiensi suatu sistem. Dataset menggunakan data dari buku-buku di perpustakaan Universitas Katolik Santo Thomas. Arsitektur dari layanan web menggunakan REST API. Dan evaluasi dari artikel ini menggunakan layanan web, setiap klien dapat mengakses data yang sama dari platform yang berbeda. Batasan dalam artikel ini adalah bahwa sistem keamanan untuk layanan web tidak dijelaskan.

Rulloh, Mahmudah, dan Kabetta (2017), memberikan kemudahan bagi para developer untuk memberi nama notasi objek dan juga harus dikenali oleh

komputer. Dataset tersebut menggunakan data dari internet yang mengacu pada sejarah bendera tersebut. Metode yang digunakan dalam hal ini adalah menggunakan arsitektur REST untuk layanan tersebut. Evaluasi dengan menggunakan arsitektur REST dan dengan dukungan notasi JSON memudahkan dalam proses penerapan teknologi pada aplikasi yang dibangun. Sayangnya, sistem keamanan untuk web service tidak dijelaskan.

Aziz, Wiharto, dan Wicaksono (2013), menemukan bahwa sistem Moodle dengan sistem eksternal lain yang ingin diintegrasikan dengan sistem Moodle. Oleh karena itu, web service ini pada dasarnya tidak sesuai jika digunakan untuk perangkat mobile. Arsitektur dari layanan web menggunakan REST API. Dengan layanan web REST, layanan web Moodle yang awalnya dibuat untuk kebutuhan System to System juga dapat digunakan untuk System to user. Oleh karena itu perlu dibuat model aplikasi agar smart client dapat mengakses fungsi web service yang sesuai dengan kebutuhan penggunaan mobile.

Naskah (2010), mengemukakan bagaimana cara mengintegrasikan program aplikasi yang berbeda dan antar platform, namun dalam penelitian ini terbatas pada pengambilan fungsi php pada web server satu untuk digunakan pada web server lain yaitu kelola situs web. Dataset tersebut menggunakan beberapa partisi dimana setiap partisi diberi masukan dan membandingkan keluaran dengan kebutuhan yang diberikan. Arsitektur dari web service menggunakan SOAP API. Evaluasi yang dilakukan adalah bahwa web service Web service yang dibangun mampu memberikan hubungan antara pengusaha dan pencari kerja secara efisien, hal ini disebabkan adanya fasilitas link ke situs web pencari kerja tersebut. Sayangnya, sistem keamanan untuk web service tidak dijelaskan.

Yusrizal, Dawood, dan Roslidar (2017), mengemukakan bahwa dokter praktek swasta ingin mencari rekam medis pasien, sehingga dokter praktek swasta harus mencari rekam medis pasien di gudang. lemari dan membutuhkan waktu lebih lama. Dataset yang digunakan diambil dari data rekam medis yang ada serta informasi pasien yang diambil dari beberapa dokter swasta di Banda Aceh. Arsitektur dari layanan web menggunakan REST API. Evaluasi agar web service dapat diakses oleh aplikasi rekam medis praktik swasta dokter yang telah dikembangkan, sehingga sudah dapat digunakan oleh dokter dan perawat. Namun tidak ada sistem keamanan di layanan tersebut dan ada beberapa kesalahan dalam model kelas dan pengontrol kelas.

Wagh dan Thool (2012), menemukan bahwa menyediakan layanan web di perangkat seluler karena jumlah penggunanya adalah seluler dan selalu mengalami kemajuan. Maka penulis ingin membuat perbandingan antara SOAP dan REST pada web services. Dataset tersebut menggunakan beberapa framework yang dapat digunakan di mobile. Evaluasi tersebut adalah framework yang digunakan untuk pengujian dapat bekerja dengan baik dengan arsitektur SOAP dan REST. Namun penulis tidak memasukkan security system dalam artikelnya dan ada beberapa framework yang belum pernah dicoba.

METODE PENELITIAN

Penelitian ini membutuhkan beberapa dataset dan juga alat-alat yang digunakan dalam proses analisa kinerja.

Kumpulan Data Set

Data yang digunakan diambil dari icon + data monitoring penjualan SBU Regional Medan. Data terdiri dari data produk klien dari setiap penjualan, harga pemasangan, dan alamat penghentian produk ini.

Perangkat Lunak

- Visual Studio Code sebagai perangkat lunak editor untuk mengetik kode dan menjalankan program.
- Golang sebagai bahasa pemrograman yang digunakan untuk aplikasi.
- Elasticsearch untuk basis data.
- Postman untuk aplikasi menjalankan layanan dan memeriksa token otentikasi.
- Mux dalam golang untuk multiplekser permintaan HTTP.
- JMeter untuk mendeteksi respon waktu dari klien ke server.
- Go-elasticsearch untuk koneksi antara aplikasi dan database

Skenario Pengujian

Pengujian menggunakan library di Golang. Berikut detail skenario pengujian.

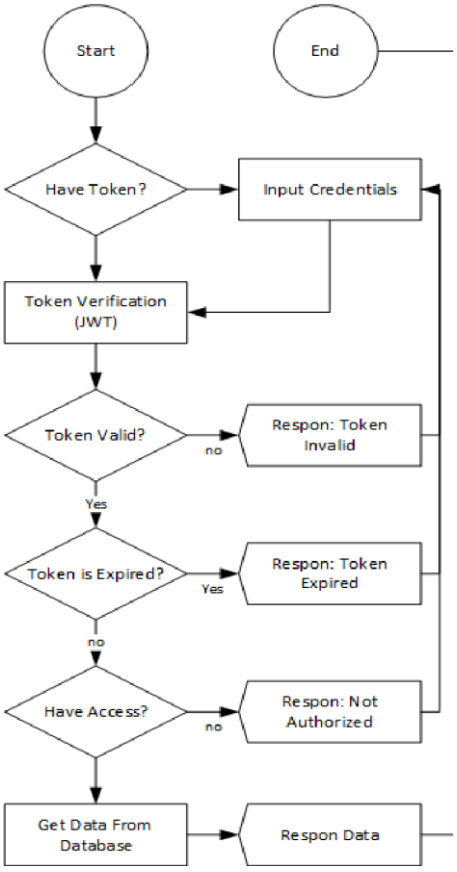
- Metode AES untuk data enkripsi dan dekripsi.
- JWT untuk Golang untuk otentikasi token.
- MD5 untuk mengenkripsi sandi dari akun pengguna dan menyimpannya ke database.

HASIL PENELITIAN DAN PEMBAHASAN

Database yang digunakan dalam proyek ini menggunakan Elasticsearch. Fokus awal pada layanan yang dibuat adalah pencarian; Oleh karena itu, Elasticsearch adalah pilihan karena merupakan mesin pencari dan analisis teks lengkap sumber terbuka yang sangat skalabel. Sehingga akan lebih cepat dalam proses pencarian dan juga berbasis ketenangan.

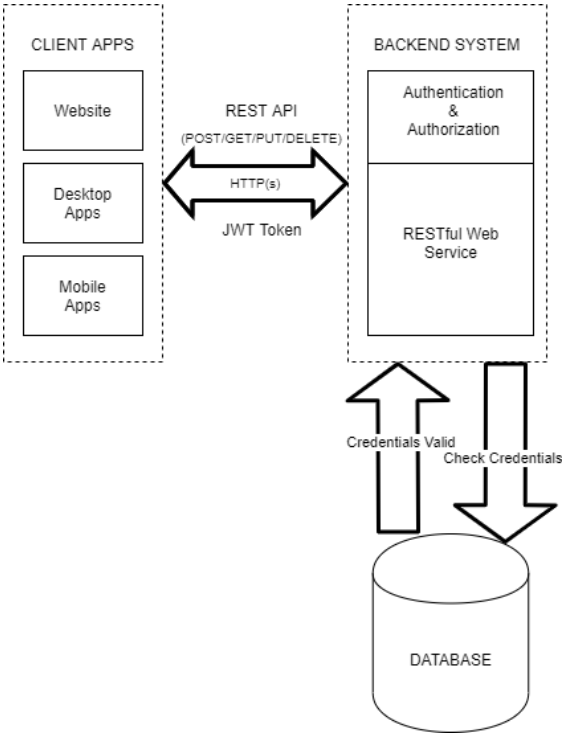
Untuk otentikasi menggunakan JSON Web Token. JWT digunakan untuk mengotentikasi REST API, sehingga tidak semua pengguna dapat mengakses sisanya dengan bebas. Dengan menggunakan JWT ini, pengguna tidak perlu masuk ke server otorisasi karena pengguna sudah terdaftar dan memiliki kunci privat.

Metode pengiriman pesan juga ditambahkan dengan metode AES. Jadi jika layanan dikirim, penerima juga harus memiliki kunci yang sama di server untuk mendekripsi pesan tersebut. Proyek ini menggunakan metode AES karena fleksibilitasnya dalam mengenkripsi teks biasa dan juga pesan yang dikirim dalam bentuk JSON. Kunci yang digunakan dalam metode AES ini adalah kunci jangka panjang. Jadi antara server dan aplikasi ada kesepakatan kunci.



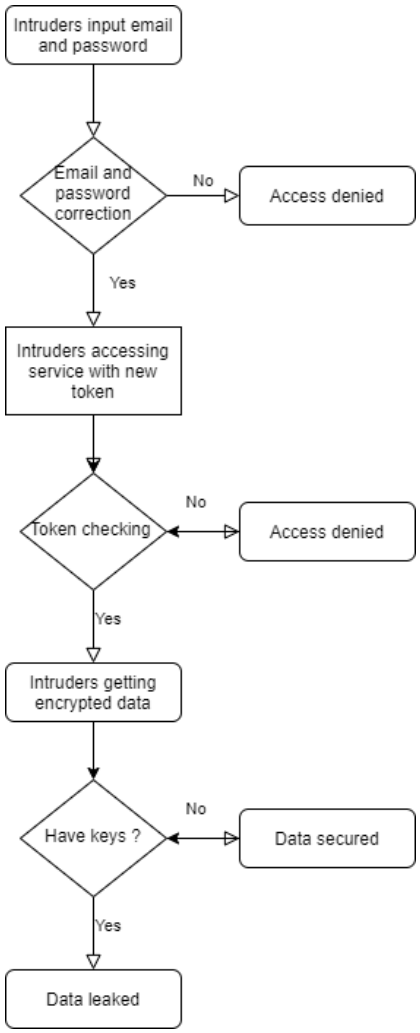
Gambar 1
JWT Flowchart

Ilustrasi di atas menjelaskan bagaimana JWT melakukan otentikasi akun pengguna saat mengakses layanan. Token yang dicentang tidak boleh menjadi token sembarangan. Meskipun token tidak kadaluwarsa dan pengguna terdaftar, aksesibilitas pengguna tersebut juga diperiksa, sehingga level setiap pengguna juga terdeteksi.



Gambar 2
Penerapan JWT pada arsitektur layanan web RESTFUL

Ilustrasi di atas menunjukkan proses perjalanan dalam layanan arsitektur istirahat. Permintaan yang masuk ke layanan diperiksa untuk otentikasi dan otorisasi dari pengguna itu. Setelah berhasil layanan selanjutnya akan memeriksa kredensial atau ID pengguna yang meminta untuk didaftarkan di database.

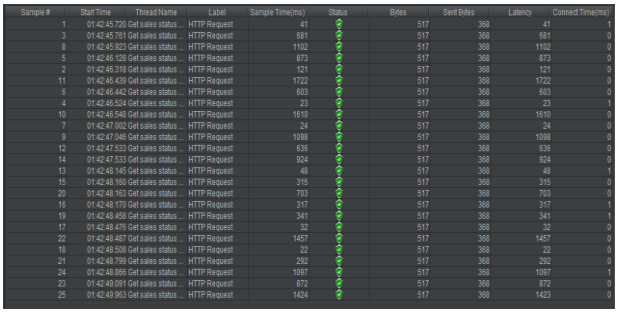


Gambar 3
Diagram Alir Pengujian Penyusup.

Berikut ilustrasi untuk pengujian penyusup. Ketika penyusup mendapatkan email dan kata sandi secara tidak sengaja, mereka akan mendapatkan token untuk mengakses layanan. Data dari login akan diperoleh dan digunakan untuk mengakses layanan lain. Dengan token yang didapat bersamaan dengan data yang dikirim dari layanan login, maka hal-hal yang dibutuhkan untuk mengakses layanan tersebut terpenuhi. Setelah penyusup mengakses layanan, mereka akan mendapatkan data yang dikembalikan dari server. Namun data tersebut telah dienkripsi dengan metode AES dan hanya dapat didekripsi dengan kunci yang sama yang telah ditetapkan.

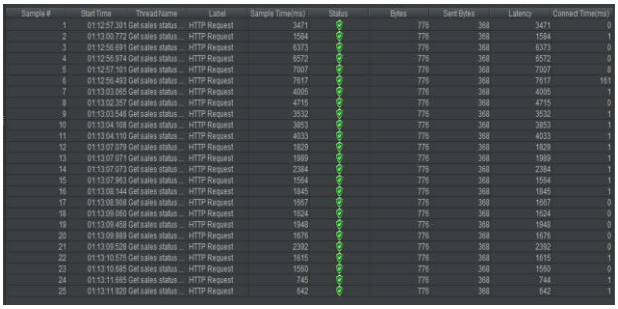
Gambar 4
Hasil Login Pengguna dengan Data Terenkripsi Secara Bersamaan 5 Kali.

Dari ilustrasi di atas dapat diketahui bahwa rata-rata waktu yang dibutuhkan saat login dengan data terenkripsi cenderung lebih lama yaitu 373,88ms. Waktu tercepat adalah 8ms dan waktu terlama adalah 1442ms.



Gambar 5
Pengguna (Threads) Mengakses GetSales Tanpa Data Terenkripsi Secara Bersamaan 5 Kali

Ilustrasi di atas menunjukkan waktu yang dibutuhkan oleh 5 pengguna (threads) saat mengakses layanan GetSales dalam 1 detik sebanyak 5 kali. Rata-rata waktu yang dibutuhkan adalah 655.12ms, waktu tercepat 22ms, dan waktu terlama 1722ms.



Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	01:12:57.391	Get sales status ...	HTTP Request	3471		776	368	3471	0
2	01:13:00.772	Get sales status ...	HTTP Request	1584		776	368	1584	1
3	01:12:56.691	Get sales status ...	HTTP Request	6373		776	368	6373	0
4	01:12:56.974	Get sales status ...	HTTP Request	6572		776	368	6572	0
5	01:12:57.101	Get sales status ...	HTTP Request	7007		776	368	7007	8
6	01:12:56.493	Get sales status ...	HTTP Request	7617		776	368	7617	161
7	01:13:03.065	Get sales status ...	HTTP Request	4005		776	368	4005	1
8	01:13:02.357	Get sales status ...	HTTP Request	4715		776	368	4715	0
9	01:13:03.548	Get sales status ...	HTTP Request	3532		776	368	3532	1
10	01:13:04.108	Get sales status ...	HTTP Request	3853		776	368	3853	1
11	01:13:04.110	Get sales status ...	HTTP Request	4033		776	368	4033	1
12	01:13:07.079	Get sales status ...	HTTP Request	1829		776	368	1829	1
13	01:13:07.071	Get sales status ...	HTTP Request	1989		776	368	1989	1
14	01:13:07.073	Get sales status ...	HTTP Request	2384		776	368	2384	1
15	01:13:07.963	Get sales status ...	HTTP Request	1564		776	368	1564	1
16	01:13:08.144	Get sales status ...	HTTP Request	1845		776	368	1845	1
17	01:13:08.909	Get sales status ...	HTTP Request	1667		776	368	1667	0
18	01:13:09.669	Get sales status ...	HTTP Request	1624		776	368	1624	0
19	01:13:09.458	Get sales status ...	HTTP Request	1948		776	368	1948	0
20	01:13:09.989	Get sales status ...	HTTP Request	1676		776	368	1676	0
21	01:13:09.528	Get sales status ...	HTTP Request	2392		776	368	2392	0
22	01:13:10.575	Get sales status ...	HTTP Request	1615		776	368	1615	1
23	01:13:10.685	Get sales status ...	HTTP Request	1560		776	368	1560	0
24	01:13:11.665	Get sales status ...	HTTP Request	745		776	368	744	1
25	01:13:11.920	Get sales status ...	HTTP Request	642		776	368	642	1

Gambar 6
Pengguna (Threads) Mengakses GetSales Dengan Data Terenkripsi Secara Bersamaan 5 Kali

SIMPULAN

Pada akhirnya, keamanan layanan web merupakan hal yang sangat penting. Dengan token yang tersedia, setiap layanan yang dikirimkan menjadi lebih aman dan tidak sembarang pengguna dapat mengakses layanan tersebut. Token yang terdapat di JWT dapat membuat pengguna memiliki otentikasi untuk mengakses layanan yang tersedia. Dikombinasikan dengan enkripsi tambahan menggunakan metode AES agar data yang dikirim melalui layanan dapat melindungi data dari penyusup. Meskipun waktu yang dibutuhkan untuk mengenkripsi pesan akan memakan waktu 373.88ms yang 238.32 milidetik lebih lambat daripada yang tidak dienkripsi.

Saran agar suatu aplikasi atau layanan yang akan dibuat selanjutnya adalah tidak hanya fokus pada fitur-fitur yang menarik, tetapi keamanan datanya juga harus diperhatikan dan menggunakan metode enkripsi dan otentikasi dari setiap akses layanan agar data tersebut aman dari penyusup.

DAFTAR PUSTAKA

Aziz, Abdul. Wiharto. Wicaksono, Bayu. (2013). Pemanfaatan Web Service Moodle Berbasis REST-JSON untuk Membangun Moodle Online Learning Extension berbasis Android. JURNAL ITSMART, 2(2), 1-6.

Deviana, Hartati. (2011). Penerapan XML Web service Pada Sistem Distribusi Barang. Jurnal Generic, 6(2), 61-70.

Naskah, Ihsan. (2010). MEMANFAATKAN WEB SERVICES UNTUK LAYANAN INFORMASI PEKERJAAN ONLINE.

Perkasa, Muhammad Iqbal. Setiawan, Eko Budi. (2018). Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token. ULTIMA COMPUTING, X(1), 19-26.

Rulloh, Amin. Mahmudah, Dewi Erla. Kabetta, Herman. (2017). Implementasi REST API pada Aplkiasi Panduan Kepaskibraan Android. Teknikom, 1(2), 85-89.

Sibagariang, Swono. (2016). PENERAPAN WEB SERVICE PADA PERPUSTAKAAN BERBASIS ANDROID. Jurnal Mahajana Inforamasi, 1(1), 28-32.

Wagh, Kishor. Thool, Dr Ravindra. (2012). A Comparative Study of Soap Vs REST Web Services Provisioning Techniques for Mobile Host. Journal of Informatics Engineering and Applications, 2(5), 12-16.

Yusrizal. Dawood, Rahmad. Roslidar. (2017). Rancang Bangun Layanan Web (Web Service) Untuk Aplikasi Rekam Medis Praktik Pribadi Dokter. KITEKTRO: Jurnal Online Teknik Elektro, 2(1), 1-8.