JURNAL RESPONSIF, Vol. 4 No.2 Agustus 2022, pp. 142~151
E-ISSN**:** 2685-6964

SISTEM INFORMASI *DECRYPT* RESPON *BRIDGING* BPJS KESEHATAN DENGAN ALGORITMA AES 256

Ramdhan Saepul Rohman[1], Dasya Arief Firmansah[2], Erni Ermawati[3]

[1]Universitas Bina Sarana Infromatika
e-mail: ramdhan.rpe@bsi.ac.id

[2]Universitas Bina Sarana Infromatika
e-mail:dasya.daf@bsi.ac.id

[3]Universitas Bina Sarana Infromatika
e-mail: erni.ert@bsi.ac.id

Abstrak

Kebijakan bridging sistem antar BPJS Kesehatan dengan rumah sakit telah berlangsung sejak lama sesuai dengan perjanjian yang telah disepakati oleh kedua belah pihak. Bridging BPJS mengalami pengembangan dari waktu ke waktu, yang pada mulanya bridging versi 1 hingga versi 2 saat ini. Pada bridging versi 1 proses pertukaran data berlangsung lebih mudah karna data yang dikirimkan tidak dienkripsi terlebih dahulu. Namun dikarenakan alasan peningkatan keamanan data, maka pada bridging versi 2 setiap proses pertukaran data untuk semua service yang diminta oleh rumah sakit telah dienkripsi ke dalam bahasa mesin yang tidak mudah dibaca secara langsung. Hal ini tentunya menjadi masalah bagi rumah sakit karena harus mendekripsi data yang dikirimkan oleh bpjs kesehatan kedalam bentuk normal yang bisa dibaca seperti halnya bentuk sebelumnya. Metode penelitian deskriptif kualitatif diterapkan untuk dapat menginterpretasikan serta merinci setiap data serta tahapan yang dilakukan dalam proses ini, mulai dari data mentah yang masih terenkripsi hingga data yang sudah terdekripsi dengan baik. Dengan adanya proses dekrip maka setiap data yang dikirimkan oleh BPJS kesehatan akan sangat mudah untuk dikelola dan proses pelayanan pun berlangsung lebih optimal. BPJS Kesehatan mengenkripsi data yang dikirimkan menggunakan SHA 256. Sebaliknya, pihak rumah sakit harus dapat mendekrip respon yang diterima menggunakan SHA 256. Proses ini juga dilengkapi dengan generate signature dan timestamp yang harus selalu diperbarui setiap detiknya. Hasil dari dekripsi data yang berhasil dilakukan oleh rumah sakit akan diterima dalam bentuk JSON yang selanjutnya data tersebut dapat dengan mudah diterapkan pada sistem rumah sakit.

Kata Kunci: Bridging, BPJS Kesehatan, Rumah sakit, AES 256, Dekripsi

Abstract

The bridging system policy between BPJS Health and hospitals has been going on for a long time in accordance with the agreement that has been agreed by both parties. BPJS bridging undergoes development from time to time, which was originally bridging version 1 to version 2 at this time. In bridging version 1 the data exchange process takes place more easily because the data sent is not encrypted first. However, due to reasons for increasing data security, in bridging version 2 every data exchange process for all services requested by the hospital has been encrypted into machine language which is not easy to read directly. This is certainly a problem for hospitals because they have to decrypt the data sent by the health bpjs into a normal form that can be read as well as the previous form. Qualitative descriptive research methods are applied to be able to interpret and detail each data and the stages carried out in this process, starting from raw data that is still encrypted to data that has been well decrypted. With the decryption process, any data sent by BPJS Kesehatan will be very easy to manage and the service process will take place more optimally. BPJS Kesehatan encrypts data sent using SHA 256. On the other hand, the hospital must be able to decrypt responses received using SHA 256. This process is also equipped with a generated signature and timestamp that

Naskah diterima 1 Juli 2022; direvisi 21 Juli 2022; diterbitkan 30 Agustus 2022

must be updated every second. The results of the data decryption that was successfully carried out by the hospital will be received in the form of JSON which then the data can be easily applied to the hospital system.

***Keywords** : Bridging, BPJS Health, Hospital, AES 256, Decrypt*

1. Pendahuluan

Dalam upaya meningkatkan mutu layanan yang lebih baik kepada peserta maupun terhadap fasilitas kesehatan (RS), BPJS Kesehatan mengembangkan bridging system, yaitu penggunaan fasilitas IT (web service) yang memungkinkan dua sistem yang berbeda pada saat yang sama mampu melakukan dua proses tanpa adanya intervensi satu sistem pada sistem lainnya secara langsung. Bridging system bertujuan meningkatkan efektivitas entry data processing serta efisiensi penggunaan sumber daya dengan tetap menjaga keamanan dan kerahasiaan data, namun bersifat transparan. Selain itu, bridging system diharapkan dapat meningkatkan kecepatan dalam proses pengelolaan klaim, piutang, maupun verifikasi (BPJS Kesehatan, 2014)

Dalam prosesnya, pihak BPJS menyelengarakan bridging secara bertahap, mulai dari bridging BPJS versi 1 hingga bridging BPJS versi 2 saat ini. Proses bridging ini tidak lain adalah sebagai sarana pertukaran data antara server bpjs sebagai server pusat dan server rumah sakit di setiap wilayah di Indonesia. Pertukaran data yang dimaksud meliputi data kepesertaan pasien BPJS, antrian online, JKN Mobile, ketersediaan kamar di setiap rumah sakit, virtual claim, ketersediaan obat dan lain sebagainya.

Pada mulanya bridging BPJS versi 1 dalam proses pertukaran data antar rumah sakit dilakukan dengan mudah tanpa melalui proses enkripsi data. Data yang berikan oleh pihak BPJS pada rumah sakit berupa data dengan format JSON. Namun guna meningkatkan keamanan serta menghindari hal-hal yang tidak diharapkan dari pihak yang tidak berkepentingan, BPJS kesehatan melakukan perbaruan pada bridging BPJS menjadi versi 2, salah satu perbaruan tersebut yaitu mengenkripsi setiap data yang diminta oleh rumah sakit setiap kali terjadi pertukaran data. Atas alasan itu pula, rumah sakit pun tentunya dituntut harus dapat beradaptasi dengan kebijakan terkait. Rumah sakit tentunya harus dapat mendekripsi atau menerjemahkan setiap respon yang diberikan oleh bpjs yang masih terbungkus oleh sandi kriptografi agar setiap pesan yang diterima mudah untuk dibaca.

Dalam mengenkirpsi setiap data yang dikirmkan, BPJS menggunakan algoritma AES 256 ditambah dengan LZstring sebagai kompresi data serta membutuhkan signature dan waktu tangkap data sekian detik, sehingga tingkat keamaan data saat dikirimkan sangat tinggi dan kerahasiaan data dapat terjaga dengan baik hal ini tentunya dapat meminimalisir adanya aktifitas yang tidak dikehendaki oleh pihak yang tidak berkepentingan.

Enkripsi adalah proses dimana informasi atau data yang hendak dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya dengan menggunakan algoritma tertentu. Dekripsi adalah kebalikan dari enkripsi yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi semula(Muharram et al., 2018).

Advanced Encryption Standard (AES) adalah lanjutan dari algoritma enkripsi standar DES (Data Encryption Standard)yang masa berlakunya dianggap telah usai karena factor kemananan. AES dibangun bertujuan untuk mengamankan pemerintahan diberbagibidang. Algoritma AES didesign menggunkan blok chiper minimal dari blok 128 bit input dan mendukung ukuran 3 kunci (3-key-sizes), yaitu kunci 128 bit,192 bit, dan 256 bit(Tahir et al., 2020).

Dalam penerapannya, bridging BPJS dikombinasikan dengan sistem yang tersedia pada masing-masing rumah sakit dengan menggunakan salah satu bahasa pemrogaraman yang diperbolehkan meliputi PHP, C#, ruby, pyton, VB .net, Java dan Cocoa (IOS & Mac).

Pada dasarnya bridging BPJS merupakan web service mengingatkan pertukaran data tidak hanya terjadi pada satu server saja namun meliputi lebih dari satu server.

Meskipun proses request dapat dilakukan dengan bahasa pemrograman yang tersebut diatas, namun untuk tahapan ujicoba setiap URL katalog BPJS dilakukan dengan menggunakan aplikasi Postman. Postman dapat mengecek setiap URL yang

menyediakan fasilitas service API dengan ketentuan parameter yang telah disepakati oleh server utama. Setiap kali terjadi request data, BPJS memerlukan signature dan timestamp yang selalu berubah disetiap detiknya.

Timestamp merupakan identitas waktu dari pembuatan dan pengiriman pesan otomatis dari sistem sehingga setiap data yang dikirmikan memiliki waktu respon yang berbeda(Wirara, et al., 2020).

Timestamp merupakan waktu yang akan di-generate oleh client saat ingin memanggil setiap service. Format waktu ini ditulis dengan format unix-based-time (berisi angka, tidak dalam format tanggal sebagaimana mestinya). Format waktu menggunakan Coordinated Universal Time (UTC), dalam penggunaannya untuk mendapatkan timestamp, rumus yang digunakan adalah (local time in UTC timezone in seconds) - (1970-01-01 in seconds). Sedangkan signature merupakan hasil dari pembuatan signature yang dibuat oleh client. Signature yang digunakan menggunakan pola HMAC-SHA256(Trustmark, 2021)

Postman merupakan sebuah REST client berbasis web yang tersedia dalam bentuk ekstensi pada Google Chrome. Postman mempunyai tampilan antarmuka (user interface) yang baik dan juga lengkap. Postman mempunyai fitur seperti design, build, test dan documentation API(Fajrin, 2017).

API atau (Application Programming Interface) adalah adalah sekumpulan fungsi, subroutine, protocol komunikasi, atau kakas/tools untuk membuat software perangkat lunakDalam penelitian ini, bahasa pemrograman yang digunakan berupa PHP mengingat request bridging akan diterapkan pada sistem berbasis website(Ramadhanu & Priandika, 2021).

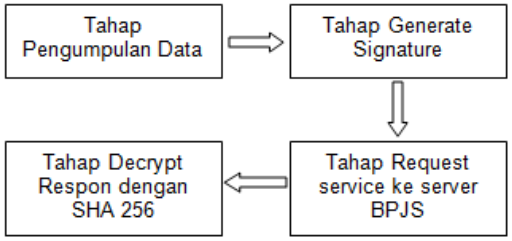
PHP (hypertext preprocessor) adalah suatu bahasa pemrograman yang digunakan untuk menterjemahkan basis kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat server-side yang ditambahkan ke HTML(Prahasti et al., 2022).

UML (Unifield Modeling Language) merupakan salah satu metode pemodelanvisual yang digunakan dalam perancangan dan pembuatan sebuah software yang berorientasikan pada objek,UML juga sebagai metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem(Darmansah & Raswini, 2022).

2. Metode Penelitian

Jenis metode penetlitian yang digunakan yaitu deskriptif kualitatif. Penelitian ini merupakan suatu suatu teknik yang menggambarkan dan menginterpretasikan arti data-data yang telah terkumpul dengan memberikan perhatian dan merekam sebanyak mungkin aspek situasi yang diteliti pada saat itu, sehingga memperoleh gambaran secara umum dan menyeluruh tentang keadaan sebenarnya(Akhmad, 2015)

Proses yang dilakukan terbagi kedalam empat tahapan, berikut skema yang dimaksud :



Gambar 1. Skema penelitian
Sumber: Peneliti (2022)

a. Tahap Pengumpulan Data

Tahapan ini merupakan tahapan persiapan awal sebelum dilakukan proses dekripsi respon BPJS kesehatan. Tahapan ini terdiri dari observasi, wawancara dan juga studi pustaka. Semua tahapan dilakukan guna memproleh informasi yang akurat serta dapat menunjang setiap tahapan yang dilakukan. Pada tahapan wawancara dan observasi dilakukan langsung pada objek atau pihak yang terlibat pada sistem, pada tahap ini diproleh akun bridging untuk dapat mengakses server BPJS. Akun tersebut merupakan akun utama yang terdiri dari consid, userkey dan secretkey). Sementara untuk studi pustaka berkaitan dengan teori yang menjadi sumber rujukan penelitian.

b. Tahap Generate Signature

Tahap ini merupakan tahap lanjutan setelah akun utama telah diproleh. Pada tahap ini dilakukan pembuatan signature dan timestamp berdasarkan akun tersebut

diatas. Signature dan timestamp merupakan kunci untuk masuk ke layanan API service BPJS kesehatan. BPJS kesehatan akan menverifikasi setiap signature yang dibuat dengan kurun waktu tertentu. Jika signature yang dibuat dinyatakan *valid* oleh BPJS, maka server BPJS akan memberikan respon berupa data json yang terenkripsi. Setiap respon yang dikirimkan memiliki jangka waktu sekian detik sesuai dengan timestamp yang diproleh. Berikut contoh hasil signature yang berhasil digenerate :

```
<?php
$consid     = "xxxx";
$secretKey= "xxxxxxxxxx";
$userKey   = "xxxxxxxxxxxxxxxxxxxxxxxxx";
date_default_timezone_set('UTC');
$tStamp = strval(time()-strtotime('1970-01-
01 00:00:00'));
$signature=hash_hmac('sha256',
$consid."&".$tStamp, $secretKey, true);
$encodedSignatur=
base64_encode($signature);
$urlencodedSignature=
urlencode($encodedSignature);
echo "X-cons-id     : " .$consid ."'"."<br>";
echo "user_key      : " .$userKey."<br>";
echo "X-timestamp   : " .$tStamp ."'"."<br>";
echo"X-signature   : "
.$encodedSignature."'"."<br>";
?>
```

Berikut tampilan hasil dari proses generate signature menggunakan PHP :

```
X-cons-id : 0000
X-timestamp : 1656293659
X-signature : piSiJqKw9TBfof8q2HMhZUZ9GyR55grYXXMXMRV8MVc=
user_key : dcd359517a3896fac2f8cee98f9a0859
```

Gambar 2. Hasil generate signature
Sumber: Peneliti (2022)

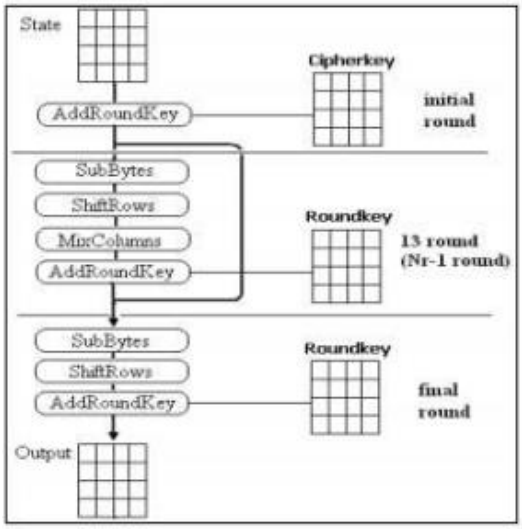
c. Tahap Request service ke server BPJS

Tahapan ini dapat dilakukan jika signature sudah diproleh, sebaliknya jika signature belum diproleh maka tahapan ini tidak bisa dilakukan. Proses request dilakukan menggunakan aplikasi POSTMAN dengan komponen URL service serta parameter consid, timestamp, signature dan userkey.

Consid merupakan kode consumer (pengakses web-service). Kode ini akan diberikan oleh BPJS Kesehatan. Userkey merupakan key untuk mengakses webservice. Setiap service consumer memiliki user_key masing-masing(Trustmark, 2021)

d. Tahap Decrypt Respon dengan SHA 256

Bagian ini merupakan tahap akhir dalam proses dekrip respon BPJS untuk mendapatkan data yang utuh sehingga mudah untuk dibaca dan dikelola. Saat data sudah terdekripsi dengan baik, data yang ditampilkan berupa data dengan format json yang selanjutnya dapat diparsing dengan mudah untuk diterapkan pada sistem rumah sakit.



Gambar 3. Ilustrasi Proses Enkripsi AES
Sumber:(Muharram et al., 2018)

Saat ini, AES (Advanced Encryption Standard) merupakan algoritma cipher yang cukup aman untuk melindungi data atau informasi yang bersifat rahasia. Pada tahun 2001, AES digunakan sebagai standar algoritma kriptografi terbaru yang dipublikasikan oleh NIST (National Institute of Standard and Technology) sebagai pengganti algoritma DES (Data Encryption Standard) yang sudah berakhir masa penggunaannya. Algoritma AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit.(Yuniati et al., 2011)

3. Hasil dan Pembahasan

3.1.Pengumpulan Data

Berikut tahapan pengumpulan data yang dilakukan :

a. Wawancara

Tahapan ini berupa tanya jawab secara langsung antara peneliti dengan narasumber untuk mendapatkan informasi lengkap terkait dekrip respon bridging BPJS serta hal-hal yang terkait dengan sistem yang dimaksud yang dapat mempengaruhi aktivitas yang terjadi.

b. Observasi

Pengamatan secara langsung terhadap sistem bridging pada RSI Assyifa Sukabumi. Dengan cara mengamati, mengumpulkan serta mempelajari secara langsung terkait alur dari bridging BPJS terutama saat terjadi proses request data pada server BPJS. Pada tahapan ini setiap data didokumentasikan secara rinci baik berupa sceen shoot tampilan katalog BPJS melalui laman https://dvlp.bpjs-kesehatan.go.id:8888/trust-markmaupun dicatat langsung pada media kertas, sehingga dapat menghasilkan informasi yang relavan dan akurat persis seperti yang terjadi di lapangan.

c. Studi Pustaka

Guna mendukung penelitian yang dilakukan, agar dapat menghasilkan informasi yang relevan dilakukan juga pencarian melalui berbagai sumber terpercaya baik melalui jurnal, website maupun E-book.

3.2 Proses Bisnis sistem

Setiap rumah sakit yang akan melakukan bridging dengan BPJS diharuskan memiliki beberapa akun utama. Akun ini diberikan oleh pihak BPJS setelah rumah sakit menandatangani kesekapatan terkait akan dilakukannya bridging web service. Akun yang dimaksud berupa username dan password untuk masuk ke halaman dashboard katalog BPSJ kesehatan. Selain itu pihak rumah sakit juga harus memiliki akun penting lainnya untuk dapat melakukan permintaan service data pada server bpjs. Akun tersebut meliputi consid, secret key dan userkey.

Pada katalog bpjs tercantum beberapa service yang diperlukan untuk kedua belah pihak mulai dari service antrian online (web service RS/BPJS), aplicare, vclaim, Pcare dan lain sebagainya.

Setelah semua akun diproleh, selanjutnya pihak rumah sakit melalui unit IT/SIMRSnya sebagai user melakukan ujicoba secara berkala request service server BPJS melalui aplikasi postman atau

1. Dekrip secara manual

sejenisnya, aplikasi ini dapat bertindak seperti layaknya penghubung antar satu server dengan server lainnya sehingga dapat saling berkomunikasi antar satu sama lain. Aplikasi ini sedikitnya memiliki dua aktivitas pertukaran data yaitu request dan respon antar server. Untuk dapat melakukan request, hal pertama yang harus disiapkan adalah user harus dapat membuat generate signature dan time stamp melalui tiga akun parameter utama yakni consid, secret key dan user key. Perlu diketahui bahwa signature dan timestamp selalu berubah setiap detiknya bahkan setiap kali user mengirim request baru, signature dan timestamp akan berubah dengan cepat.

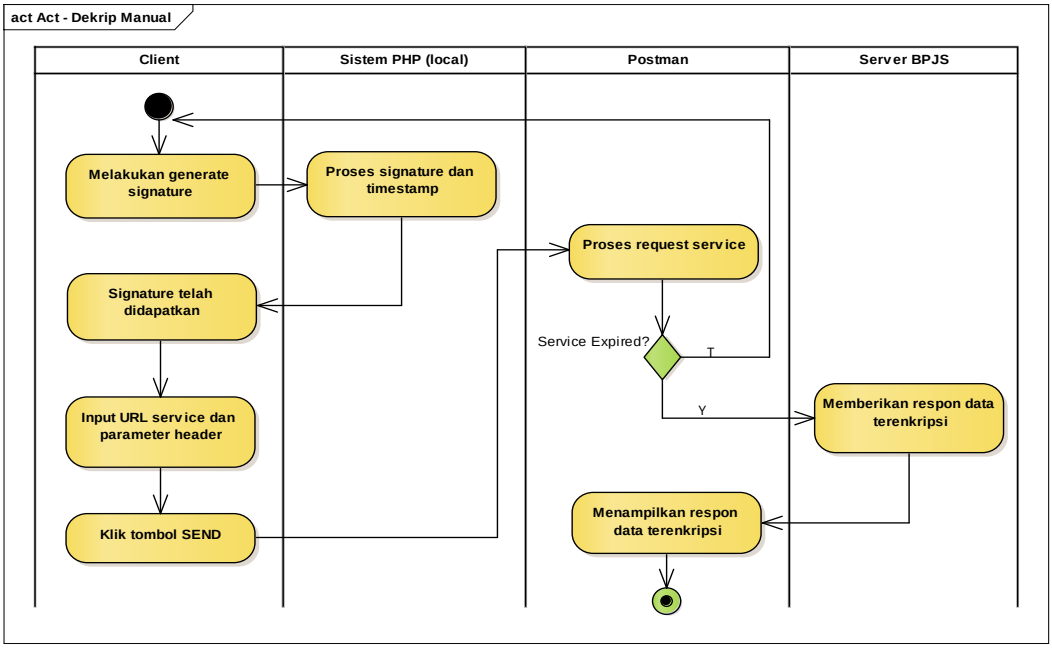
Jika signature dan timestamp telah berhasil dibuat selanjutnya user dapat membuat request untuk service tertentu pada service BPJS. User harus menentukan metode request yang digunakan lalu menyisipkan url service yang dituju selanjutnya menyisipkan empat parameter header yakni consid, timestamp terbaru,signature terbaru dan user key. Jika request berhasil selanjutnya server BPJS akan memberikan respon berupa data dengan format JSON yang masih terenkripsi dengan algoritma AES 256 dan terkompres LZstring.

Langkah yang perlu dilakukan yakni mendekrip reposn yang diberikan server BPJS menggunakan library LZstring dan decrypt AES 256. Data yang terenkripsi AES 256 memilikidigit lebih panjang dibanding algoritma chipper lainnya. Tahapan decrypt dan dekompresi yang dilakukan selayaknya kita menerjemahkan sebuah bahasa yang rumit dan seulit dimengerti kedalam bahasa yang lebih mudah untuk difahami. Setelah repon terdekrip, user akan lebih mudah dalam mengelola data untuk berbagai kepentingan pelayanan kesehatan serta diterapkan pada aplikasi rumah sakit.

Adanya sistem bridging ini tentunya lebih memudahkan kedua belah pihak dalam merekam setiap data yang terlibat dalam aktivitas pelayanan kesehatan secara realtime sehingga proses pelayanan dapat terpantau dan berjalan dengan cepat dan optimal

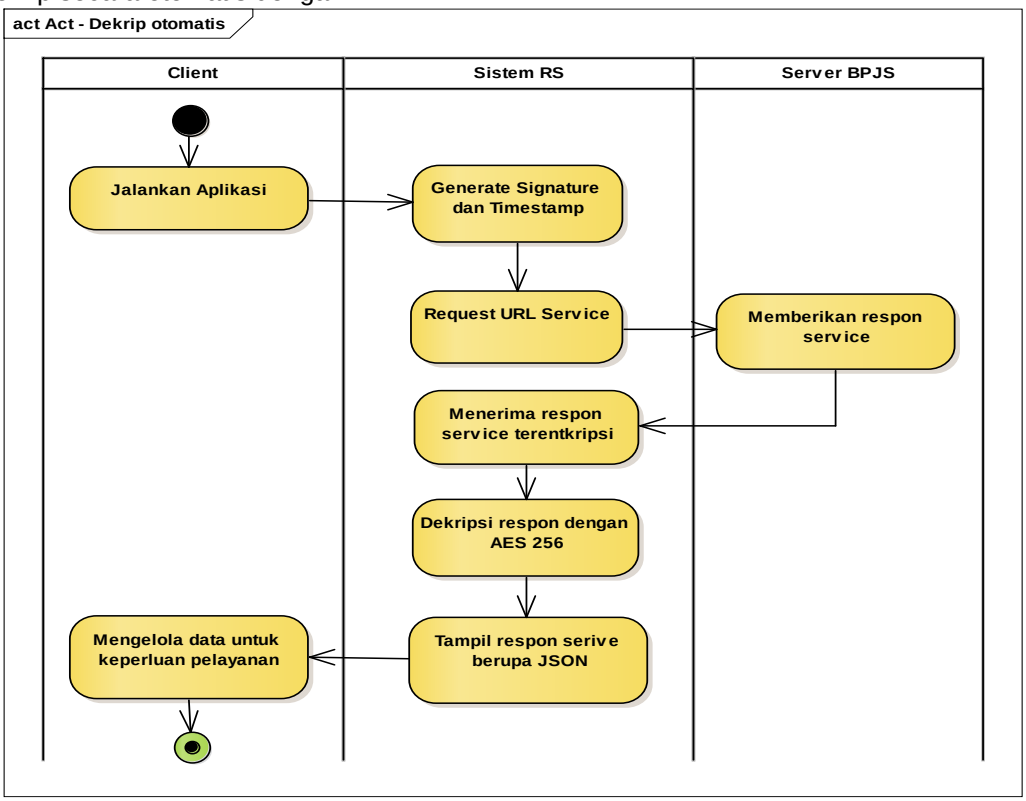
3.3Activity diagram request service respon BPJS Kesehatan

Berikut merupakan tahapan request respon service BPJS Kesehatan :



Gambar 4. *Activity diagram* request service manual dengan sistem PHP dan postman
Sumber: Peneliti (2022)

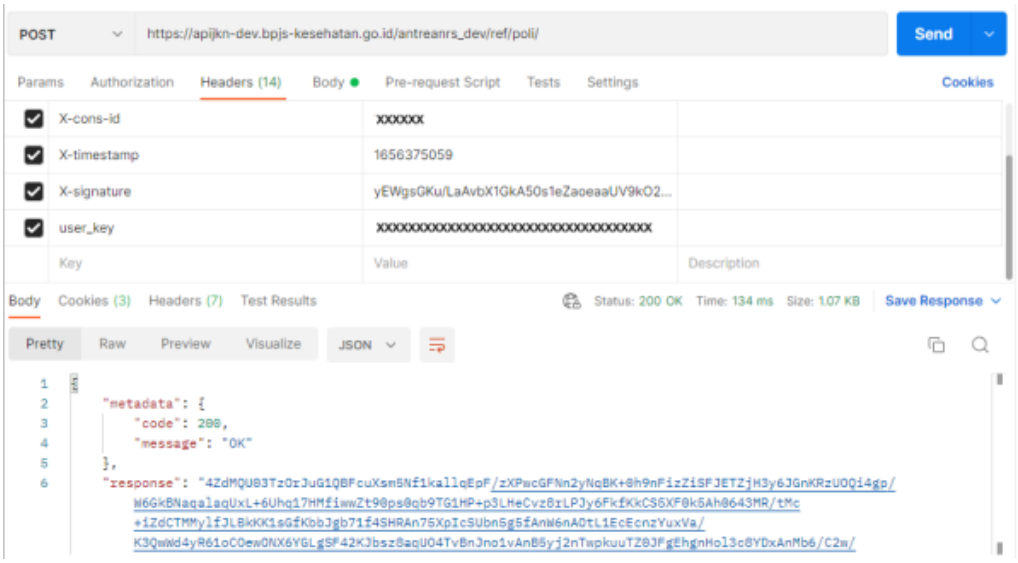
2. Dekrip secara otomatis dengan PHP



Gambar 5. *Request service* dengan sistem rumah sakit secara otomatis
Sumber: Peneliti (2022)

3.4 *Interface*

1.Proses request service menggunakan postman



Gambar 6. Request service via postman
Sumber: Peneliti (2022)

2.Service respon dari BPJS Kesehatan yang masih terenkripsi dengan AES 256


```
{
"metadata": {
"code": 200,
"message": "OK"
},
"response": "4ZdMQU03TzOrJuG1QBFcuXsm5Nf1kallqEpF/zXPwcGFNn2yNqBK+0h9nFizZiSFJETZjH3y6JGnKRzUOQi4gp/
W6GkBNaqalaqUxL+6Uhq17HMfiwwZt90ps0qb9TG1HP+p3LHeCvz8rLPJy6FkfKkCS5XF0k5Ah0643MR/tMc
+iZdCTMMylfJLBkKK1sGfKbbJgb71f4SHRAn75XpIcSUbn5g5fAnW6nAOtL1EcEcnzYuxVa/
K3QwWd4yR61oCOewONX6YGLgSF42KJbsz8aqUO4TvBnJno1vAnB5yj2nTwpkuuTZ0JFgEhgnHol3c8YDxAnMb6/C2w/
9vX8cpsyO8UFoNdwcheOYCiN2DfkJRvNmchl/bnE/h0hM3nQxWqA9sT4F+5/uKNvmZ4hhNmtEc9ypf0cR8FipGuPz9LiOvAWJNR/
3gK7kQaXbPvPywwxJUyktzEtIm3M5BzAyK5M3uLmjvVQwmYD90+f99ZykzrqeU36bn8M4SEfyBbjsAilVxBX60wCv4zvf
+5BbxaCD1JZNoRDX+oUkP+jpVc6ZM2qaEmRZViYGwpoCbGawV+VKORmMvwjSyxAg+ky9OC87C+zFjcPgrzxh/4pf58uhYQW2OjPurJ/
94Dy2gxHJrTPSlyVzrrEPNI80vzqPrzRml06xGzfqndRWTfxf2opdU/Vnl2xuugc6s5USGy0i85YSox+I7NGa0h75NIBAftoYUlBd/sKQF/
eivy5UsiUA="
}
```


Gambar 7. Respon service BPJS terenkripsi
Sumber: Peneliti (2022)

3. Proses generate signature dengan PHP
Pada tahap ini signature digenerate terlebih dahulu sebelum dilakukan proses dekripsi respon BPJS. Berikut ini kode program dari tahaoan yang dimaksud :

```
<?php
include 'LZString.php';
include 'LZReverseDictionary.php';
include 'LZData.php';
include 'LZUtil.php';
include 'LZContext.php';
$consid     = "xxxx";
$secretKey = "xxxxxxxxxx";
$userKey    = "xxxxxxxxxxxxxxxxxx";
date_default_timezone_set('UTC');
$tStamp      = strval(time()-strtotime('1970-01-01
00:00:00'));
$signature   = hash_hmac('sha256',
$consid."&".$tStamp, $secretKey, true);
$encodedSignature =
base64_encode($signature);
$urlencodedSignature =
urlencode($encodedSignature);
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://apijkn-dev.bpjs-
kesehatan.go.id/antreanrs_dev/ref/poli/',
CURLOPT_RETURNTRANSFER => true,
CURLOPT_ENCODING => '',
CURLOPT_MAXREDIRS => 10,
CURLOPT_TIMEOUT => 0,
CURLOPT_FOLLOWLOCATION => true,
CURLOPT_HTTP_VERSION =>
CURL_HTTP_VERSION_1_1,
CURLOPT_CUSTOMREQUEST => 'GET',
CURLOPT_HTTPHEADER =>
array('X-cons-id: '.$consid,
      'X-timestamp: '.$tStamp,
      'X-signature: '.$encodedSignature,
      'user_key: '.$userKey),
));
?>
```

4. Proses dekripsi respon
Setelah *signature* berhasil terbuat, langkah selanjutnya adalah mendekrip respon yang telah diberikan oleh BPJS yang masih terenrkipsi menggunakan AES 256.

```
<?php
```

```
$response = curl_exec($curl);
curl_close($curl);
$data      = json_decode($response, true);
function stringDecrypt($key, $string){
$encrypt_method = 'AES-256-CBC';
$key_hash= hex2bin(hash('sha256', $key));
$iv = substr(hex2bin(hash('sha256', $key)), 0,
16);
$output  =
openssl_decrypt(base64_decode($string),
$encrypt_method, $key_hash,
OPENSSL_RAW_DATA, $iv);
   return $output;
}
function decompress($string){
   return
LZString::decompressFromEncodedURICompon
ent($string);
}
$kunci         = $consid.$secretKey.$tStamp;
$nilairespon = $data["response"];
echo "Response Kunci : ".$kunci."<br>"."<br>";
echo "Response Encrypt : ".$nilairespon;
$hasilakhir    =
decompress(stringDecrypt($kunci, $nilairespon));
echo "Response Decrypt : "."<br>".$hasilakhir;
?>
```

5. Hasil respon yang sudah terdekripsi


```
[
  {
    "nmpoli": "AKUPUNTUR MEDIK",
    "nmsubspesialis": "AKUPUNTUR MEDIK",
    "kdsubspesialis": "AKP",
    "kdpoli": "AKP"
  },
  {
    "nmpoli": "ANAK",
    "nmsubspesialis": "ANAK ALERGI IMUNOLOGI",
    "kdsubspesialis": "027",
    "kdpoli": "ANA"
  },
  {
    "nmpoli": "ANAK",
    "nmsubspesialis": "ANAK ENDOKRINOLOGI",
    "kdsubspesialis": "028",
    "kdpoli": "ANA"
  },
  {
    "nmpoli": "ANAK",
    "nmsubspesialis": "ANAK GASTRO-HEPATOLOGI",
    "kdsubspesialis": "029",
    "kdpoli": "ANA"
  },
  {
    "nmpoli": "ANAK",
    "nmsubspesialis": "ANAK HEMATOLOGI ONKOLOGI",
    "kdsubspesialis": "030",
    "kdpoli": "ANA"
  },
```


Gambar 8. Hasil dekrip berupa data JSON
Sumber: Peneliti (2022)

3.5 Pengujian Sistem

Pengujian dilakukan menggunakan blackbox testing guna melakukan pengecekan pada setiap fungsi dari setiap parameter ketika berada dalam kondisi tertentu. Berikut pengjuan yang dimaksud :

Tabel 1. Pengujian generate signature dan timestamp

No.	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil	Simpulan
1	Consid kosong, secret key kosong, reload sistem	Consid = kosong Secretkey = kosong	Eror generate signature	Sesuai dengan yang diharapkan	**Sesuai**
2	Consid sesuai, secret key tidak sesuai, reload sistem	Consid = sesuai Secret key = tidak sesuai	Generate signature & timestamp berhasil, namun tidak dapat digunakan request service	Sesuai dengan yang diharapkan	**Sesuai**
3	Consid tidak sesuai, secret key sesuai, reload sistem	Consid = tidak sesuai Secret key = sesuai	Generate signature & timestamp berhasil, namun tidak dapat digunakan request service	Sesuai dengan yang diharapkan	**Sesuai**
4	Condif sesuai, secret key sesuai, reload sistem	Consid = sesuai Secret key = sesuai	Generatesignature dan timestamp berhasil dan bisa digunakan untuk request service	Sesuai dengan yang diharapkan	**Sesuai**

Tabel 2. Pengujian request serviceke server BPJS Kesehatan

No.	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil	Simpulan
1	Consid kosong, timestamp kosong, signature kosong, userkey kosong, klik tombol SEND	Consid = kosong Timestamp=kosong Signature= kosong Userkey=kosong	Request Error	Sesuai dengan yang diharapkan	**Sesuai**
2	Consid kosong atau tidak valid, timestamp valid, signature valid, userkey valid, klik tombol SEND	Consid = kosong/tidak valid Timestamp=valid Signature= valid Userkey=valid	Request Error	Sesuai dengan yang diharapkan	**Sesuai**
3	Consid valid, timestamp kosong/tidak valid, signature valid, userkey valid, klik tombol SEND	Consid = valid Timestamp=kosong/tidak valid Signature= valid Userkey=valid	Unauthorized, invalid signature	Sesuai dengan yang diharapkan	**Sesuai**
4	Consid valid, timestamp valid, signature kosong/tidak valid, userkey valid, klik tombol SEND	Consid = valid Timestamp=valid Signature= kosong/tidakvalid Userkey=valid	Unauthorized, invalid signature	Sesuai dengan yang diharapkan	**Sesuai**
5	Consid valid, timestamp valid, signature valid userkey kosong/tidak valid,, klik tombol SEND	Consid = valid Timestamp=valid Signature= valid Userkey=kosong/tidak valid	Request Error	Sesuai dengan yang diharapkan	**Sesuai**
6	Consid valid, timestamp expired, signature expired userkey valid,, klik tombol SEND	Consid = valid Timestamp=expired Signature= expired Userkey=valid	Expired Service	Sesuai dengan yang diharapkan	**Sesuai**
7	Consid valid, timestamp valid, signature valid userkey valid,, klik tombol SEND	Consid = valid Timestamp=valid Signature= valid Userkey=valid	Request berhasil, respon diterima berupa decrypt data json	Sesuai dengan yang diharapkan	**Sesuai**

4. Kesimpulan

Pertukaran data antar BPJS dengan rumah sakit yang berlangsung setiap hari tentunya menuntut pihak BPJS untuk selalu meningkatkan keamanan data guna menjaga kerahasiaan data yang dikirimkan setiap kali rumah sakit melakukan permintaan data. Data yang

dikirimkan oleh BPJS kesehatan telah dienkripsi menggunakan algoritma AES 256 sehingga pihak rumah sakit harus dapat mendekripsi data tersebut dengan algoritma serupa agar data mudah untuk dikelola dan diterapkan pada sistem rumah sakit.

Respon data BPJS yang masih terenkripsi ditampilkan dalam bentuk susunan huruf dan angka yang tidak beraturan dengan digit yang sangat panjang seperti layaknya bahasa mesin yang tidak mudah difahami secara kasat mata. Dengan dilakukannya dekripsi AES 256, data yang diterima rumah sakit telah diterjemahkan dalam bentuk aslinya dengan format JSON. Permintaan data rumah sakit pada BPJS kesehatan berlangsung secara realtime sehingga dengan adanya proses dekrip secara otomatis data yang diterima mudah untuk diimplementasikan serta proses pelayanan berlangsung lebih cepat.

Referensi

Akhmad, K. A. (2015). Pemanfaatan Media Sosial bagi Pengembangan Pemasaran UMKM (Studi Deskriptif Kualitatif pada Distro di Kota Surakarta). *DutaCom Journal*, *9*(1), 43–54. http://journal.stmikdb.ac.id/index.php/dutacom/article/view/17

BPJS Kesehatan. (2014). *22 Rumah Sakit Siap Implementasikan (Komprehensif) Bridging System BPJS Kesehatan*. BPJS Kesehatan. https://www.bpjs-kesehatan.go.id/bpjs/index.php/post/read/2014/229/22-Rumah-Sakit-Siap-Implementasikan-Komprehensif-Bridging-System-BPJS-Kesehatan/berita?&per_page=222

Darmansah, D., & Raswini, R. (2022). Perancangan Sistem Informasi Pengelolaan Data Pedagang Menggunakan Metode Prototype pada Pasar Wage. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, *6*(1), 340–350. http://ejurnal.tunasbangsa.ac.id/index.php/jsakti/article/view/449

Fajrin, R. (Jurnal P. C. R. (2017). Rachmat Fajrin. *Jurnal Komputer Terapan*, *3*(1), 33–40. http://jurnal.pcr.ac.id

Muharram, F., Azis, H., & Manga, A. R. (2018). Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard (AES). *Proc. of the Seminar Nasional Ilmu Komputer Dan Teknologi Informasi*, *3*(2), 112–115.

Prahasti, P., Sapri, S., & Utami, F. H. (2022). Aplikasi Pelayanan Antrian Pasien Menggunakan Metode FCFS Menggunakan PHP dan MySQL. *Jurnal Media Infotama*, *18*(1), 153–160.

Ramadhanu, P. B., & Priandika, A. T. (2021). Rancang Bangun Web Service Api Aplikasi Sentralisasi Produk Umkm Pada Uptd Plut Kumkm Provinsi Lampung. *Jurnal Teknologi Dan Sistem Informasi (JTSI)*, *2*(1), 59–64. http://jim.teknokrat.ac.id/index.php/JTSI

Trustmark, B. (2021). *Create Signature*. BPJS Kesehatan Trust Mark Versi 1.0.0. https://dvlp.bpjs-kesehatan.go.id:8888/trust-mark/main.html#/mitra/katalog/vclaim/createsignature

Wirara, A., Hardiawan, B., & Salman, M. (2020). Identifikasi Bukti Digital pada Akuisisi Perangkat Mobile dari Aplikasi Pesan Instan "WhatsApp." *Teknoin*, *26*(1), 66–74. https://doi.org/10.20885/teknoin.vol26.iss1.art7

Yuniati, V., Indriyanta, G., & Rachmat C., A. (2011). Enkripsi Dan Dekripsi Dengan Algoritma Aes 256 Untuk Semua Jenis File. *Jurnal Informatika*, *5*(1). https://doi.org/10.21460/inf.2009.51.69