

# **Web Service E-Klaim INA-CBG**

Untuk Build 5.2.0.201712280730

## I. SETUP

Integrasi dengan SIMRS dipersyaratkan menggunakan data yang ter-enkripsi dengan symmetric encryption algorithm. Untuk itu **Encryption Key** harus di generate terlebih dahulu, melalui menu Setup - Integrasi - SIMRS:

Home	Setup	Migrasi	Backup / Restore	Personnel	Akun
------	-------	---------	------------------	-----------	------

### SETUP INTEGRASI SIMRS

Konfigurasi	
Kode RS	3174282
Encryption Key	-

Silakan klik tombol Generate Key disebelah kanan untuk Encryption Key baru.

Generate Key

Klik tombol **Generate Key** untuk membuat **Encryption Key**.

Anda akan men-generate Encryption Key baru.  
Maka aplikasi SIMRS harus disesuaikan dengan Encryption Key yang baru.

Generate Encryption Key?

Ya (Generate) Batal

Selanjutnya silakan klik tombol **Ya (Generate)**. Catatan: adanya konfirmasi untuk generate tujuannya adalah untuk menjaga supaya **Encryption Key** tidak sembarangan diubah tanpa sengaja.

itu Setelah muncul

Captcha : nUV8K

Masukkan Tulisan Pada Gambar Captcha : nUV8K

Masukkan Password Anda : .....

Ya (Generate) Tidak (Batal Generate)

rekonfirmasi dengan memasukkan kode yang tertera pada gambar dan memasukkan password Anda, kemudian klik tombol **Ya (Generate)**. Hasilnya:

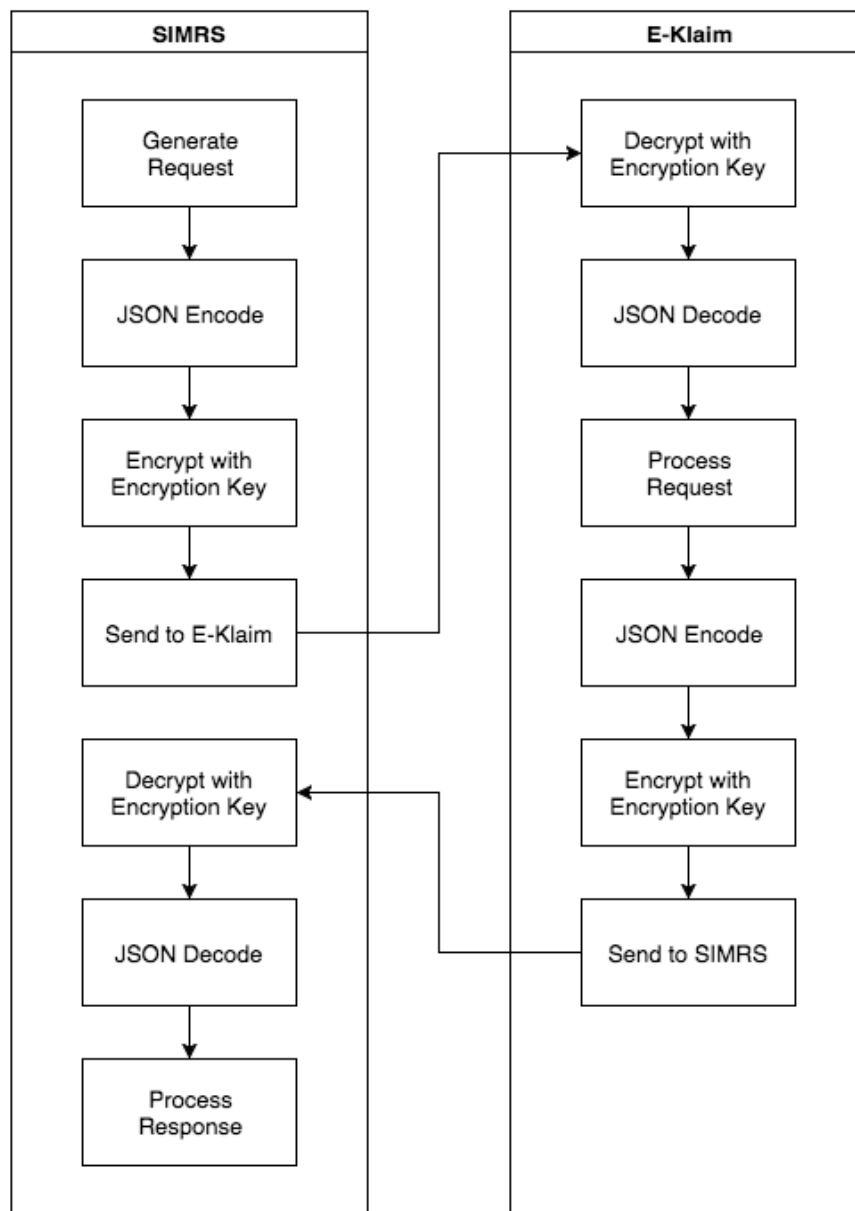
Konfigurasi	
Kode RS	3174282
Encryption Key	d26cbb6f64dadec194e6681c4a076ecdbbf5628f10f4416a6d9afe15309f1fae

Silakan copy Encryption Key tersebut diatas untuk digunakan dalam SIMRS dan dimohon untuk sangat dijaga kerahasiaannya.

Generate Key

**Encryption Key** akan digenerate oleh Aplikasi E-Klaim dan tersimpan didalam database untuk digunakan dalam proses enkripsi/dekripsi pada setiap pemanggilan dan response dari **Web Service**. Dimohon untuk sangat menjaga **Encryption Key** tersebut dengan hati-hati dan rahasia.

Berikut ini skema alur pertukaran data dalam Integrasi SIMRS dengan Aplikasi E-Klaim melalui **Web Service**, dimulai dari SIMRS men-generate-request:



Dengan alur tersebut diatas, diharapkan data tidak dipertukarkan dalam kondisi terbuka.

Untuk operasional selanjutnya, disarankan untuk men-generate ulang **Encryption Key** secara periodik sebulan sekali demi keamanan dan menyesuaikannya kembali dalam SIMRS.

## II. WEB SERVICE

Web Service Aplikasi E-Klaim ini dapat diakses pada endpoint:

`http://alamat_server_aplikasi/E-Klaim/ws.php`

Silakan disesuaikan `alamat_server_aplikasi` dengan ip address server E-Klaim.

Untuk keperluan pengembangan integrasi, endpoint tersebut dapat ditambahkan parameter debug sebagai berikut:

`http://alamat_server_aplikasi/E-Klaim/ws.php?mode=debug`

Untuk penggunaan mode debug ini, silakan edit `server.ini` dan ubah parameter `enable_debug` pada segmen `[web_service]` sama dengan 1 sebagai berikut:

```
30 [web_service]
31 enable_debug = 1
```

Dengan mode debug, maka pemanggilan dan response tidak perlu di-enkripsi. Namun penggunaan mode debug tersebut tidak diperbolehkan untuk operasional karena berpotensi menjadi lubang keamanan.

## III. ENKRIPSI / DEKRIPSI

Untuk setiap response web service yang bukan mode debug, maka response akan selalu ter-enkripsi. Contoh format yang ter-enkripsi sbb:

-----BEGIN ENCRYPTED DATA-----

/KsK5I2TcjfU6gu2pBwjANNvPRUrrpmqVgLkIZdUyUts1hz9xSk9ECgjjgMu5UBqSOeymPAA+DGF+M32WFSIr0dj/ctsKXTJEYupxVBQ5Fxe8pwEbheIEPMXlr2Z/ZsCqZvHQpPknNySiwnKrX/9sZSMj9pCWY9Al1Gz9mSenkAsaGab9FkjZwOP7K4ERA/dxIrcNMFJUj36X/yvspM+VQOit4GNvqOduoSv7Ckn5g3U+fdA80C5RpvKHTogd2AWwtc+1lWCL1bCc1Qj3BeCop1h8o/okYJdboZE63stYek1IyVeV

-----END ENCRYPTED DATA-----

Untuk melakukan dekripsi, silakan baris pertama "-----BEGIN ENCRYPTED DATA-----" dan baris terakhir "-----END ENCRYPTED DATA-----" dihilangkan terlebih dahulu.

Berikut ini source code PHP yang digunakan untuk melakukan enkripsi dan dekripsi. Sebelum itu Anda akan membutuhkan PHP dengan OpenSSL extension.

```
// Encryption Function
function inacbg_encrypt($data, $key) {

    /// make binary representasion of $key
    $key = hex2bin($key);

    /// check key length, must be 256 bit or 32 bytes
    if (mb_strlen($key, "8bit") !== 32) {
        throw new Exception("Needs a 256-bit key!");
    }

    /// create initialization vector
    $iv_size = openssl_cipher_iv_length("aes-256-cbc");
    $iv = openssl_random_pseudo_bytes($iv_size); // dengan catatan dibawah

    /// encrypt
    $encrypted = openssl_encrypt($data,
                                "aes-256-cbc",
```

```

        $key,
        OPENSSL_RAW_DATA,
        $iv );

    /// create signature, against padding oracle attacks
    $signature = mb_substr(hash_hmac("sha256",
                                     $encrypted,
                                     $key,
                                     true),0,10,"8bit");

    /// combine all, encode, and format
    $encoded = chunk_split(base64_encode($signature.$iv.$encrypted));

    return $encoded;
}

// Decryption Function
function inacbg_decrypt($str, $strkey){

    /// make binary representation of $key
    $key = hex2bin($strkey);

    /// check key length, must be 256 bit or 32 bytes
    if (mb_strlen($key, "8bit") !== 32) {
        throw new Exception("Needs a 256-bit key!");
    }

    /// calculate iv size
    $iv_size = openssl_cipher_iv_length("aes-256-cbc");

    /// breakdown parts
    $decoded = base64_decode($str);
    $signature = mb_substr($decoded,0,10,"8bit");
    $iv = mb_substr($decoded,10,$iv_size,"8bit");
    $encrypted = mb_substr($decoded,$iv_size+10,NULL,"8bit");

    /// check signature, against padding oracle attack
    $calc_signature = mb_substr(hash_hmac("sha256",
                                           $encrypted,
                                           $key,
                                           true),0,10,"8bit");
    if(!inacbg_compare($signature,$calc_signature)) {
        return "SIGNATURE_NOT_MATCH"; /// signature doesn't match
    }

    $decrypted = openssl_decrypt($encrypted,
                                "aes-256-cbc",
                                $key,
                                OPENSSL_RAW_DATA,
                                $iv);

    return $decrypted;
}

/// Compare Function
function inacbg_compare($a, $b) {

```

```

    /// compare individually to prevent timing attacks

    /// compare length
    if (strlen($a) !== strlen($b)) return false;

    /// compare individual
    $result = 0;
    for($i = 0; $i < strlen($a); $i++) {
        $result |= ord($a[$i]) ^ ord($b[$i]);
    }

    return $result == 0;
}

```

Contoh pemanggilan wev service dengan php curl:

```

// contoh encryption key, bukan aktual
$key = "5cb7e8e7d0f6d15a9c986f4accc5022893938092039";

// json query
$json_request = <<<EOT
{
    "metadata": {
        "method": "claim_print"
    },
    "data": {
        "nomor_sep": "16120507422"
    }
}
EOT;

// membuat json juga dapat menggunakan json_encode:
$ws_query["metadata"]["method"] = "claim_print";
$ws_query["data"]["nomor_sep"] = "16120507422";
$json_request = json_encode($ws_query);

// data yang akan dikirimkan dengan method POST adalah encrypted:
$payload = inacbg_encrypt($json_request,$key);

// tentukan Content-Type pada http header
$header = array("Content-Type: application/x-www-form-urlencoded");

// url server aplikasi E-Klaim,
// silakan disesuaikan instalasi masing-masing
$url = "http://192.168.56.101/E-Klaim/ws.php";

// setup curl
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HTTPHEADER,$header);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $payload);

// request dengan curl

```

```

$response = curl_exec($ch);

// terlebih dahulu hilangkan "----BEGIN ENCRYPTED DATA----\r\n"
// dan hilangkan "----END ENCRYPTED DATA----\r\n" dari response
$first = strpos($response, "\n")+1;
$last = strrpos($response, "\n")-1;
$response = substr($response,
                  $first,
                  strlen($response) - $first - $last);

// decrypt dengan fungsi inacbg_decrypt
$response = inacbg_decrypt($response,$key);

// hasil decrypt adalah format json, ditranslate kedalam array
$msg = json_decode($response,true);

// variable data adalah base64 dari file pdf
$pdf = base64_decode($msg["data"]);

// hasilnya adalah berupa binary string $pdf, untuk disimpan:
file_put_contents("klaim.pdf",$pdf);

// atau untuk ditampilkan dengan perintah:
header("Content-type:application/pdf");
header("Content-Disposition:attachment;filename='klaim.pdf'");
echo $pdf;

```

**Catatan:**

Untuk fungsi **openssl\_random\_pseudo\_bytes** tersebut diatas, disarankan untuk diganti dengan fungsi **random\_bytes()** yang bisa diperoleh dari package **random\_compat** ([https://github.com/paragonie/random\\_compat](https://github.com/paragonie/random_compat)). Hal tersebut dikarenakan pada fungsi **openssl\_random\_pseudo\_bytes** ditemukan permasalahan atau bug sehingga menghasilkan random yang tidak kuat secara kriptografi (<https://bugs.php.net/bug.php?id=70014>) terutama bagi SIMRS yang masih menggunakan PHP versi 5.6.10 kebawah.

#### IV. KATALOG METHOD WEB SERVICE

Khusus untuk semua field dalam metadata adalah mandatory.

Disarankan untuk mencoba web service menggunakan ARC (*Advanced Rest Client*, pada *Google Chrome*, buatan *chromerestclient.com*) untuk melacak jika terjadi kendala atau error.

Kecuali dinyatakan lain didalam penjelasan method dibawah, maka response untuk setiap method adalah sebagai berikut:

```
{
  "metadata": {
    "code": "200",
    "message": "OK"
  }
}
```

Atau contoh jika terjadi kesalahan:

```
{
  "metadata": {
    "code": 400,
    "message": "Nomor SEP terduplikasi",
    "error_no": "E2003"
  },
  "duplicate": [
    {
      "nama_pasien": "TEST PASIEN",
      "nomor_rm": "3849988",
      "tgl_masuk": "2016-12-19 21:10:07"
    },
    {
      "nama_pasien": "TEST TEST",
      "nomor_rm": "3887726",
      "tgl_masuk": "2016-12-23 04:48:53"
    }
  ]
}
```

Daftar kode error dapat dilihat dibagian bawah pada halaman 24.



Berikut ini daftar method:

**1. Membuat klaim baru (dan registrasi pasien jika belum ada):**

```
{
  "metadata": {
    "method": "new_claim"
  },
  "data": {
    "nomor_kartu": "0000668870001",
    "nomor_sep": "0001R0016120507422",
    "nomor_rm": "123-45-67",
    "nama_pasien": "NAMA TEST PASIEN",
    "tgl_lahir": "1940-01-01 02:00:00",
    "gender": "2"
  }
}
```

**Response:**

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "patient_id": 453,
    "admission_id": 1,
    "hospital_admission_id": 678
  }
}
```

**Response jika ada duplikasi nomor SEP:**

```
{
  "metadata": {
    "code": 400,
    "message": "Duplikasi nomor SEP",
    "error_no": "E2007"
  },
  "duplicate": [
    {
      "nama_pasien": "TEST PASIEN",
      "nomor_rm": "3849988",
      "tgl_masuk": "2016-12-19 21:10:07"
    },
    {
      "nama_pasien": "TEST TEST",
      "nomor_rm": "3887726",
      "tgl_masuk": "2016-12-23 04:48:53"
    }
  ]
}
```

**Mandatory:** nomor\_kartu, nomor\_sep, nomor\_rm, nama\_pasien, tgl\_lahir, gender

**Keterangan parameter:**

**nomor\_kartu** : Nomor Kartu peserta JKN  
**nomor\_sep** : Nomor SEP  
**nomor\_rm** : Nomor rekam medis pasien

**nama\_pasien** : Nama lengkap pasien  
**tgl\_lahir** : Tanggal lahir pasien dengan format "YYYY-MM-DD hh:mm:ss"  
 YYYY = tahun 4 digit  
 MM = bulan 2 digit  
 DD = hari 2 digit  
 hh = jam 2 digit  
 mm = menit 2 digit  
 ss = detik 2 digit  
**gender** : Jenis kelamin, diisi 1 = Laki-laki, 2 = Perempuan

## 2. Update data pasien:

```
{
  "metadata": {
    "method": "update_patient",
    "nomor_rm": "123-45-67"
  },
  "data": {
    "nomor_kartu": "0000668800001",
    "nomor_rm": "123-45-76",
    "nama_pasien": "NAMA TEST PASIEN",
    "tgl_lahir": "1940-01-01 02:00:00",
    "gender": "2"
  }
}
```

## 3. Hapus data pasien:

```
{
  "metadata": {
    "method": "delete_patient"
  },
  "data": {
    "nomor_rm": "123-45-67",
    "coder_nik": "123123123123"
  }
}
```

Mandatory: nomor\_rm, coder\_nik

Keterangan parameter:

**coder\_nik** : adalah NIK yang tersimpan pada data Personel Registration pada aplikasi E-Klaim.

Personnel Data	Addresses	Access Profile
#id	2/339	
Employee Name	<input type="text" value="INACBG"/> <input type="text" value=""/>	<input type="text" value=""/>
	<small>Title/Prefix</small>	<small>Suffix</small>
Employee ID	<input type="text" value="00001"/>	
Alias	<input type="text" value=""/> <small>Nama singkatan, wajib diisi max 5 karakter</small>	
NIK	<input type="text" value="123123123123"/> <small>Nomor Induk Kependudukan, wajib diisi</small>	

#### 4. Untuk mengisi/update data klaim:

```
{
  "metadata": {
    "method": "set_claim_data",
    "nomor_sep": "0901R001TEST0001"
  },
  "data": {
    "nomor_sep": "0901R001TEST0001",
    "nomor_kartu": "233333",
    "tgl_masuk": "2017-11-20 12:55:00",
    "tgl_pulang": "2017-12-01 09:55:00",
    "jenis_rawat": "1",
    "kelas_rawat": "1",
    "adl_sub_acute": "15",
    "adl_chronic": "12",
    "icu_indikator": "1",
    "icu_los": "2",
    "ventilator_hour": "5",
    "upgrade_class_ind": "1",
    "upgrade_class_class": "vip",
    "upgrade_class_los": "5",
    "add_payment_pct": "35",
    "birth_weight": "0",
    "discharge_status": "1",
    "diagnosa": "S71.0#A00.1",
    "procedure": "81.52#88.38",
    "tarif_rs": {
      "prosedur_non_bedah": "300000",
      "prosedur_bedah": "20000000",
      "konsultasi": "300000",
      "tenaga_ahli": "200000",
      "keperawatan": "80000",
      "penunjang": "1000000",
      "radiologi": "500000",
      "laboratorium": "600000",
      "pelayanan_darah": "150000",
      "rehabilitasi": "100000",
      "kamar": "6000000",
      "rawat_intensif": "2500000",
      "obat": "2000000",
      "alkes": "500000",
      "bmhp": "400000",
      "sewa_alat": "210000"
    },
    "tarif_poli_eks": "100000",
    "nama_dokter": "RUDY, DR",
    "kode_tarif": "AP",
    "payor_id": "3",
    "payor_cd": "JKN",
    "cob_cd": "0001",
    "coder_nik": "123123123123"
  }
}
```

Mandatory: coder\_nik

Keterangan parameter:

**tgl\_masuk** : Tanggal masuk pasien untuk episode perawatan yang diklaim  
**tgl\_pulang** : Tanggal pulang  
**jenis\_rawat** : 1 = rawat inap, 2 = rawat jalan  
**kelas\_rawat** : 3 = Kelas 3, 2 = Kelas 2, 1 = Kelas 1  
**adl\_sub\_acute** : ADL = Activities of Daily Living Score untuk pasien sub acute, nilainya 12 s/d 60  
**adl\_chronic** : Activities of Daily Living Score untuk pasien chronic nilainya 12 s/d 60  
**icu\_indicator** : Jika pasien masuk ICU selama dalam episode perawatan maka diisi "1" (satu).  
Jika tidak ada perawatan ICU maka diisi "0" (nol).  
**icu\_los** : Jumlah hari rawat di ICU  
**ventilator\_hour** : Jumlah jam pemakaian ventilator jika di ICU  
**upgrade\_class\_ind**, **upgrade\_class\_class**, **upgrade\_class\_los**, dan **add\_payment\_pct** dijelaskan sebagai berikut: Untuk naik kelas, gunakan parameter **upgrade\_class\_ind** = "1" (satu) jika ada naik kelas, dan "0" (nol) jika tidak ada naik kelas. Untuk kenaikan kelas yang dituju gunakan parameter **upgrade\_class\_class**:  
**kelas\_1** = naik ke kelas 1  
**kelas\_2** = naik ke kelas 2  
**vip** = naik ke kelas vip  
**vvip** = naik ke kelas vvip

Untuk lama hari rawat yang naik kelas gunakan parameter **upgrade\_class\_los**, diisi dalam format integer lama hari rawat yang naik kelas. Parameter **add\_payment\_pct** adalah koefisien tambahan biaya khusus jika pasien naik ke kelas VIP. Untuk penggunaan parameter **upgrade\_class\_ind**, **upgrade\_class\_class**, **upgrade\_class\_los** dan **add\_payment\_pct** harus disertakan 4 parameter tersebut secara bersamaan.

Parameter **payor\_id** dan **payor\_cd** dapat diperoleh pada aplikasi E-Klaim, dari group Pengaturan dan Pemeliharaan, menu Setup, Jaminan. Parameter **payor\_id** diisi dengan Payplan ID, sedangkan parameter **payor\_cd** diisi dengan Code, seperti tersebut dibawah ini:

Payplan ID	3
Payment Plan Name	JKN
Code	JKN

Khusus untuk **coder\_nik** sifatnya mandatory. Dan untuk NIK yang disertakan haruslah sudah terdaftar sebagai NIK pada user (Personnel Registration) di Aplikasi E-Klaim.

Jika NIK tersebut tidak terdaftar maka proses update akan gagal.

Parameter selain yang tercantum pada metadata dan parameter mandatory (**coder\_nik**) adalah sifatnya opsional, yaitu jika disertakan maka akan

mengubah (update, replace) namun jika tidak disertakan maka artinya tidak ada perubahan. Hal ini untuk memberikan kemungkinan bagi SIMRS untuk mengirim data secara bertahap menyesuaikan alur data yang sesuai alur kerja di rumah sakit.

Untuk penandaan kelas pasien rawat jalan (Kelas Regular dan Kelas Eksekutif), maka nilai **kelas\_rawat** adalah:

3 = regular  
1 = eksekutif

**discharge\_status** : Cara pulang didefinisikan sebagai berikut:

1 = Atas persetujuan dokter  
2 = Dirujuk  
3 = Atas permintaan sendiri  
4 = Meninggal  
5 = Lain-lain

**diagnosa** : Kode diagnosa akan dicek terhadap versi ICD-10 yang berlaku. Jika ada kode yang tidak terdaftar atau berlaku, maka kode tersebut tidak akan tersimpan.

**procedure** : Kode procedure akan dicek terhadap versi ICD-9-CM yang berlaku. Jika ada kode yang tidak terdaftar atau berlaku, maka kode tersebut tidak akan tersimpan.

Untuk kode diagnosa dan procedure, disediakan web service tersendiri untuk pencarian pada method nomor 16 dan 17 dibawah.

Khusus untuk parameter diagnosa dan prosedur disediakan fasilitas untuk menghapus, yaitu dengan tanda # (hash), dikarenakan mengirimkan parameter dengan tanpa isi seperti ini "" berarti tidak ada perubahan.

**tarif\_rs** : Untuk parameter tarif\_rs disediakan parameter breakdown seperti tersebut pada json diatas. Nilai tarif\_rs sendiri akan dihitung berdasarkan jumlah dari breakdown tersebut yaitu: **prosedur\_non\_bedah, prosedur\_bedah, konsultasi, tenaga\_ahli, keperawatan, penunjang, radiologi, laboratorium, pelayanan\_darah, rehabilitasi, kamar, rawat\_intensif, obat, alkes, , bmhp, dan sewa\_alat**. Masing-masing diisi dengan nilai integer. Untuk definisi operasional parameter tersebut silakan merujuk pada petunjuk teknis Aplikasi E-Klaim.

Contoh update data prosedur:

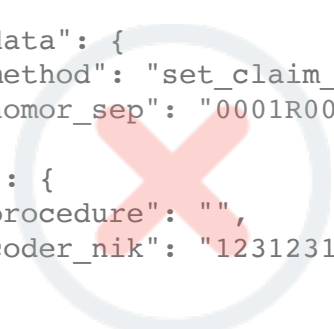
```
{
  "metadata": {
    "method": "set_claim_data",
    "nomor_sep": "0001R001612066662",
  },
  "data": {
    "procedure": "36.06#88.09",
    "coder_nik": "123123123123"
  }
}
```

Contoh hapus semua data prosedur:

```
{
  "metadata": {
    "method": "set_claim_data",
    "nomor_sep": "0001R0016120666662",
  },
  "data": {
    "procedure": "#",
    "coder_nik": "123123123123"
  }
}
```

Contoh cara hapus semua data prosedur **yang salah**, karena yang seperti berikut ini berarti tidak ada perubahan:

```
{
  "metadata": {
    "method": "set_claim_data",
    "nomor_sep": "0001R0016120666662",
  },
  "data": {
    "procedure": "",
    "coder_nik": "123123123123"
  }
}
```



**kode\_tarif** : Kode tarif adalah kelas tarif INA-CBG berdasarkan kelas rumah sakit dan kepemilikannya. Kode dan penjelasan sebagai berikut:

AP	=	TARIF RS KELAS A PEMERINTAH
AS	=	TARIF RS KELAS A SWASTA
BP	=	TARIF RS KELAS B PEMERINTAH
BS	=	TARIF RS KELAS B SWASTA
CP	=	TARIF RS KELAS C PEMERINTAH
CS	=	TARIF RS KELAS C SWASTA
DP	=	TARIF RS KELAS D PEMERINTAH
DS	=	TARIF RS KELAS D SWASTA
RSCM	=	TARIF RSUPN CIPTO MANGUNKUSUMO
RSJP	=	TARIF RSJPD HARAPAN KITA
RSD	=	TARIF RS KANKER DHARMAIS
RSAB	=	TARIF RSAB HARAPAN KITA

**cob\_cd** : Adalah jika klaim ini adalah klaim dengan Coordination of Benefit. Untuk **cob\_cd**, dapat dilihat pada pengaturan, menu COB. Untuk tidak memilih (menghapus) **cob\_cd** dari klaim silakan parameter tersebut diisi dengan kode "#".

## 5. Grouping Stage 1:

```
{
  "metadata": {
    "method": "grouper",
    "stage": "1"
  },
  "data": {
    "nomor_sep": "0001R0016120666662"
  }
}
```

Keterangan parameter:

**stage** : diisi "1" (satu)

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "cbg": {
      "code": "M-1-04-II",
      "description": "PROSEDUR PADA SENDI TUNGKAI BAWAH (SEDANG)",
      "tariff": "40388100"
    },
    "sub_acute": {
      "code": "SF-4-10-I",
      "description": "ADL Score: 15 (61 hari)",
      "tariff": 5027400
    },
    "chronic": {
      "code": "CF-4-10-I",
      "description": "ADL Score: 12 (41 hari)",
      "tariff": 1802200
    },
    "kelas": "kelas_2",
    "add_payment_amt": 18792000,
    "inacbg_version": "5.2.0.201712280730"
  },
  "special_cmg_option": [
    {
      "code": "RR04",
      "description": "Hip Implant / knee implant",
      "type": "Special Prosthesis"
    },
    {
      "code": "YY01",
      "description": "Hip Replacement / knee replacement",
      "type": "Special Procedure"
    }
  ],
  "tarif_alt": [
    {
      "kelas": "kelas_1",
      "tarif_inacbg": "47119400"
    },
    {
      "kelas": "kelas_2",
      "tarif_inacbg": "40388100"
    },
    {
      "kelas": "kelas_3",
      "tarif_inacbg": "33656700"
    }
  ]
}
```

```
}
```

## 6. Grouping Stage 2:

Untuk Grouping Stage 2 ini, jika dari hasil Grouping Stage 1 terdapat pilihan `special_cmg_option`, maka silakan masukkan didalam field `special_cmg`. Jika pilihan bisa dari satu karena dari type yang berbeda maka silakan ditambahkan tanda # diantara kode:

```
{
  "metadata": {
    "method": "grouper",
    "stage": "2"
  },
  "data": {
    "nomor_sep": "0001R0016120666662",
    "special_cmg": "RR04#YY01"
  }
}
```

Keterangan parameter:

**stage** : diisi "2" (dua)

**special\_cmg** : diisi dengan code yang diperoleh dari grouping stage 1 pada segment "**special\_cmg\_option**". Untuk mengisi lebih dari satu pilihan spesial\_cmg, code-nya dijoin dengan tanda #.

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "cbg": {
      "code": "M-1-04-II",
      "description": "PROSEDUR PADA SENDI TUNGKAI BAWAH (SEDANG)",
      "tariff": "40388100"
    },
    "special_cmg": [
      {
        "code": "YY-01-II",
        "description": "HIP REPLACEMENT / KNEE REPLACEMENT",
        "tariff": 13099000,
        "type": "Special Procedure"
      },
      {
        "code": "RR-04-III",
        "description": "HIP IMPLANT / KNEE IMPLANT",
        "tariff": 26197900,
        "type": "Special Prosthesis"
      }
    ],
    "kelas": "kelas_2",
    "add_payment_amt": 18792000,
    "inacbg_version": "5.2.0.201712280730"
  },
  "special_cmg_option": [
```



```

    {
      "code": "RR04",
      "description": "Hip Implant / knee implant",
      "type": "Special Prosthesis"
    },
    {
      "code": "YY01",
      "description": "Hip Replacement / knee replacement",
      "type": "Special Procedure"
    }
  ],
  "tarif_alt": [
    {
      "kelas": "kelas_1",
      "tarif_inacbg": "47119400",
      "tarif_sp": 13099000,
      "tarif_sr": 26197900
    },
    {
      "kelas": "kelas_2",
      "tarif_inacbg": "40388100",
      "tarif_sp": 13099000,
      "tarif_sr": 26197900
    },
    {
      "kelas": "kelas_3",
      "tarif_inacbg": "33656700",
      "tarif_sp": 13099000,
      "tarif_sr": 26197900
    }
  ]
}

```

Jika dari hasil grouper stage 1 tidak muncul parameter **special\_cmg\_option**, maka tidak perlu melakukan grouper stage 2.

#### 7. Untuk finalisasi klaim:

```

{
  "metadata": {
    "method": "claim_final"
  },
  "data": {
    "nomor_sep": "0001R0016120666662",
    "coder_nik": "123123123123"
  }
}

```

Mandatory: coder\_nik

#### 8. Untuk mengedit ulang klaim:

```

{
  "metadata": {
    "method": "reedit_claim"
  },
  "data": {
    "nomor_sep": "0001R0016120666662"
  }
}

```

```

    }
}

```

#### 9. Untuk mengirim klaim ke data center (kolektif per hari)

```

{
  "metadata": {
    "method": "send_claim"
  },
  "data": {
    "start_dt": "2016-01-07",
    "stop_dt": "2016-01-07",
    "jenis_rawat": "1",
    "date_type": "2"
  }
}

```

Keterangan parameter:

**start\_dt** : tanggal awal, format YYYY-MM-DD  
**stop\_dt** : tanggal akhir, format YYYY-MM-DD  
**jenis\_rawat** : 1 = ranap, 2 = rajal, 3 = ranap & rajal, default = 3  
**date\_type** : 1 = tanggal pulang, 2 = tanggal grouping, default = 1

Mandatory: start\_dt, stop\_dt

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "data": [
      {
        "SEP": "0001R0016120666662",
        "tgl_pulang": "2016-01-07 15:00:00",
        "kemkes_dc_Status": "sent",
        "bpjs_dc_Status": "unsent"
      }
    ]
  }
}

```

#### 10. Untuk mengirim klaim individual ke data center

```

{
  "metadata": {
    "method": "send_claim_individual"
  },
  "data": {
    "nomor_sep": "0001R0016120666662"
  }
}

```

Response:

```

{
  "metadata": {
    "code": 200,

```

```

        "message": "Ok"
    },
    "response": {
        "data": [
            {
                "no_sep": "0001R0016120666662",
                "tgl_pulang": "2016-01-07 15:00:00",
                "kemkes_dc_status": "sent",
                "bpjs_dc_status": "unsent",
                "cob_dc_status": "sent"
            }
        ]
    }
}

```

Jika terjadi error kegagalan pengiriman karena masalah koneksi:

```

{
    "metadata": {
        "code": 400,
        "message": "Error: Koneksi Gagal",
        "error_no": "E2029",
        "curl_error_no": 28,
        "curl_error_message": "Timeout was reached",
        "curl_error_constant": "CURLE_OPERATION_TIMEDOUT"
    }
}

```

Untuk referensi CURL error lainnya bisa dibaca di:

<https://curl.haxx.se/libcurl/c/libcurl-errors.html>

#### 11. Untuk menarik data klaim dari E-Klaim

```

{
    "metadata": {
        "method": "pull_claim"
    },
    "data": {
        "start_dt": "2016-01-07",
        "stop_dt": "2016-01-07",
        "jenis_rawat": "1"
    }
}

```

Response:

```

{
    "metadata": {
        "code": 200,
        "message": "Ok"
    },
    "response": {
        "data":
        "KODE_RS\tKELAS_RS\tKELAS_RAWAT\tKODE_TARIF\tPTD\tADMISSION_DATE\tDISCHARGE_DATE\tBIRTH_DATE\tBIRTH_WEIGHT\tSEX\tDISCHARGE_STATUS\tDIAGLIST\tPROCLIST\tADL1\tADL2\tIN_SP\tIN_SR\tIN_SI\tIN_SD\tINACBG\tSUBACUTE\tCHRONIC\tSP\tSR\tSI\tSD\tDESKRIPSI_INACBG\tTARIF_INACBG\tTARIF_SUBACUTE\tTARIF_CHRONIC\tDESKRIPSI_SP\tTARIF_SP\tDESKRIPSI_SR\tTARIF_SR\tDESKRIPSI_SI\tTARIF_SI\tDESKRIPSI_SD\tTARIF_SD\tTOTAL_TARIF\tTARIF_RS\tLOS\tICU_INDIKATOR\tICU_LOS\tVENT_HOUR\tNAMA_PASIEN\tMRN\tUMUR_TAHUN\tUMUR_HARI\tDPJP\tSEP\tNOK

```

```
ARTU\tPAYOR_ID\tCODER_ID\tVERSI_INACBG\tVERSI_GROUPER\tC1\tC2\tC3\tC4\n31
74282\tA\t3\tAP\t1\t01\07\2015\07\01\2016\01\01\1940\t0\t2\t2\tF2
0.6;A41.3;A37;A37.1;A39.4;A39.5;A35\t-\t15\t12\tNone\tNone\tNone\tNone\tF
-4-10-III\tSF-4-10-I\tCF-4-10-I\tNone\tNone\tNone\tNone\tSCHIZOFRENIA (BE
RAT)\t9973500\t5027400\t3384500\t-\t0\t-\t0\t-\t0\t-\t0\t18385400\t250000
0\t191\t1\t2\t5\tNAMA TEST PASIEN\t123-45-67\t75\t27575\tDR. ERNA\t0301R0
0112140006067\t0000668873981\t3;JKN\t123456789\t5.0.0\t4\t1\t0\t23\t0a1f0
1ecc6f508dcc64491c9e8327839\n"
    }
}
```

## 12. Untuk mengambil data detail per klaim

```
{
  "metadata": {
    "method": "get_claim_data"
  },
  "data": {
    "nomor_sep": "0001R0016120666662"
  }
}
```

### Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "data": {
      "kode_rs": "0000000",
      "kelas_rs": "A",
      "kelas_rawat": 1,
      "kode_tarif": "AP",
      "jenis_rawat": 1,
      "tgl_masuk": "26/10/2016",
      "tgl_pulang": "18/12/2016",
      "tgl_lahir": "15/03/1950",
      "berat_lahir": "0",
      "gender": 2,
      "discharge_status": 1,
      "diagnosa": "S71.0#A00.1",
      "procedure": "81.52#88.38",
      "adl_sub_acute": 15,
      "adl_chronic": 0,
      "tarif_rs": {
        "prosedur_non_bedah": "300000",
        "prosedur_bedah": "20000000",
        "konsultasi": "300000",
        "tenaga_ahli": "200000",
        "keperawatan": "80000",
        "penunjang": "1000000",
        "radiologi": "500000",
        "laboratorium": "600000",
        "pelayanan_darah": "150000",
        "rehabilitasi": "100000",
        "kamar": "6000000",

```

```

        "rawat_intensif": "2500000",
        "obat": "2000000",
        "alkes": "500000",
        "bmhp": "400000",
        "sewa_alat": "210000"
    },
    "los": "54",
    "icu_indikator": 1,
    "icu_los": "2",
    "ventilator_hour": "5",
    "upgrade_class_ind": "1",
    "upgrade_class_class": "vip",
    "upgrade_class_los": "5",
    "add_payment_pct": "0.0",
    "add_payment_amt": "18792000",
    "nama_pasien": "NAMA TEST PASIEN",
    "nomor_rm": "775343",
    "umur_tahun": 66,
    "umur_hari": "24332",
    "nama_dokter": "RUDY, DR",
    "nomor_sep": "16120507422",
    "nomor_kartu": "233333",
    "payor_id": "3",
    "payor_nm": "JKN",
    "coder_nm": "INACBG",
    "coder_nik": "00001",
    "patient_id": "328",
    "admission_id": "2",
    "hospital_admission_id": "2436",
    "grouping_count": "5",
    "grouper": {
        "response": {
            "cbg": {
                "code": "M-1-04-II",
                "description": "PROSEDUR PADA SENDI TUNG ...",
                "tariff": "47119400"
            },
            "special_cmgs": [
                {
                    "code": "YY-01-II",
                    "description": "HIP REPLACEMENT / KNEE ...",
                    "tariff": 13099000,
                    "type": "Special Procedure"
                },
                {
                    "code": "RR-04-III",
                    "description": "HIP IMPLANT / KNEE IMPLANT",
                    "tariff": 26197900,
                    "type": "Special Prosthesis"
                }
            ]
        },
        "inacbg_version": "5.2.0.201712280730"
    },
    "tarif_alt": [
        {
            "kelas": "kelas_1",

```

```

        "tarif_inacbg": "47119400",
        "tarif_sp": 13099000,
        "tarif_sr": 26197900
    },
    {
        "kelas": "kelas_2",
        "tarif_inacbg": "40388100",
        "tarif_sp": 13099000,
        "tarif_sr": 26197900
    },
    {
        "kelas": "kelas_3",
        "tarif_inacbg": "33656700",
        "tarif_sp": 13099000,
        "tarif_sr": 26197900
    }
]
},
"kemenkes_dc_status_cd": "unsent",
"kemenkes_dc_sent_dttm": "-",
"bpjs_dc_status_cd": "unsent",
"bpjs_dc_sent_dttm": "-",
"klaim_status_cd": "normal",
"bpjs_klaim_status_cd": "40",
"bpjs_klaim_status_nm": "40_Proses_Cabang"
}
}
}

```

### 13. Untuk mengambil status per klaim

Method ini membutuhkan `consumer_id` dan `secret` dari BPJS. Rumah sakit dipersilakan meminta kepada BPJS bagi yang belum memiliki. Kemudian dilakukan setup sebagai berikut, silakan sesuaikan isinya dengan masing-masing:

#### SETUP INTEGRASI BPJS

Kode Rumah Sakit :	0001R001 ( Kode BPJS )
Enable Server SEP	<input checked="" type="checkbox"/> Enable
Host :	172.16.5.100
Port :	18082
Consumer ID :	1001
Consumer Secret :	rs1234
? Service Name	SepLokalRest
? Versi Web Service	<input type="radio"/> Versi 1.4 <input checked="" type="radio"/> Versi 2.1
? Format Keluaran Web Service	<input type="radio"/> XML <input checked="" type="radio"/> JSON

Berikut pemanggilan method:

```

{
  "metadata": {
    "method": "get_claim_status"
  },
  "data": {
    "nomor_sep": "0001R0016120666662"
  }
}

```

```
}  
}
```

**Response:**

```
{  
  "metadata": {  
    "code": 200,  
    "message": "Ok"  
  },  
  "response": {  
    "kdStatusSep": "40",  
    "nmStatusSep": "40_Proses_Cabang"  
  }  
}
```

**14. Untuk menghapus klaim:**

```
{  
  "metadata": {  
    "method": "delete_claim"  
  },  
  "data": {  
    "nomor_sep": "0001R0016120666662",  
    "coder_nik": "37234567890121"  
  }  
}
```

**15. Cetak klaim:**

```
{  
  "metadata": {  
    "method": "claim_print"  
  },  
  "data": {  
    "nomor_sep": "0001R0016120666662"  
  }  
}
```

**Response:**

```
{  
  "metadata": {  
    "code": 200,  
    "message": "Ok"  
  },  
  "data": "7c7uNsPO4uXsTpr9zCtiTrYdzMjmHxZIEjDobAoujnJvdO7UWTB  
eRr9wb8mtnd9+gnzForViUj6QtD9xVBTJFxxz4N/DvR7IwT7RqdQ  
DsgFl5NnnWqZb/fNUKXQDQ+Q+e+yR48eo8bPF ... dst"  
}
```

Hasil dari method **claim\_print** adalah file pdf yang ter-encode dengan base 64 yang terdapat pada variable "data". Silakan decode terlebih dahulu untuk mendapatkan file pdf dalam bentuk binary untuk kemudian ditampilkan atau disimpan.

**16. Pencarian diagnosa:**

```
{  
  "metadata": {  
    "method": "search_diagnosis"  
  },  
  "data": {
```

```

    "keyword": "A00"
  }
}

```

Keterangan parameter:

**keyword** : diisi dengan kode, sebagian dari kode, atau sebagian dari nama diagnosa

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "count": 3,
    "data": [
      [
        "Cholera, unspecified",
        "A00.9"
      ],
      [
        "Cholera due to vibrio cholerae 01, biovar eltor",
        "A00.1"
      ],
      [
        "Cholera due to vibrio cholerae 01, biovar cholerae",
        "A00.0"
      ]
    ]
  }
}

```

**17. Pencarian prosedur:**

```

{
  "metadata": {
    "method": "search_procedures"
  },
  "data": {
    "keyword": "74.9"
  }
}

```

Keterangan parameter:

**keyword** : diisi dengan kode, sebagian dari kode, atau sebagian dari nama prosedur

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "count": 2,
    "data": [

```



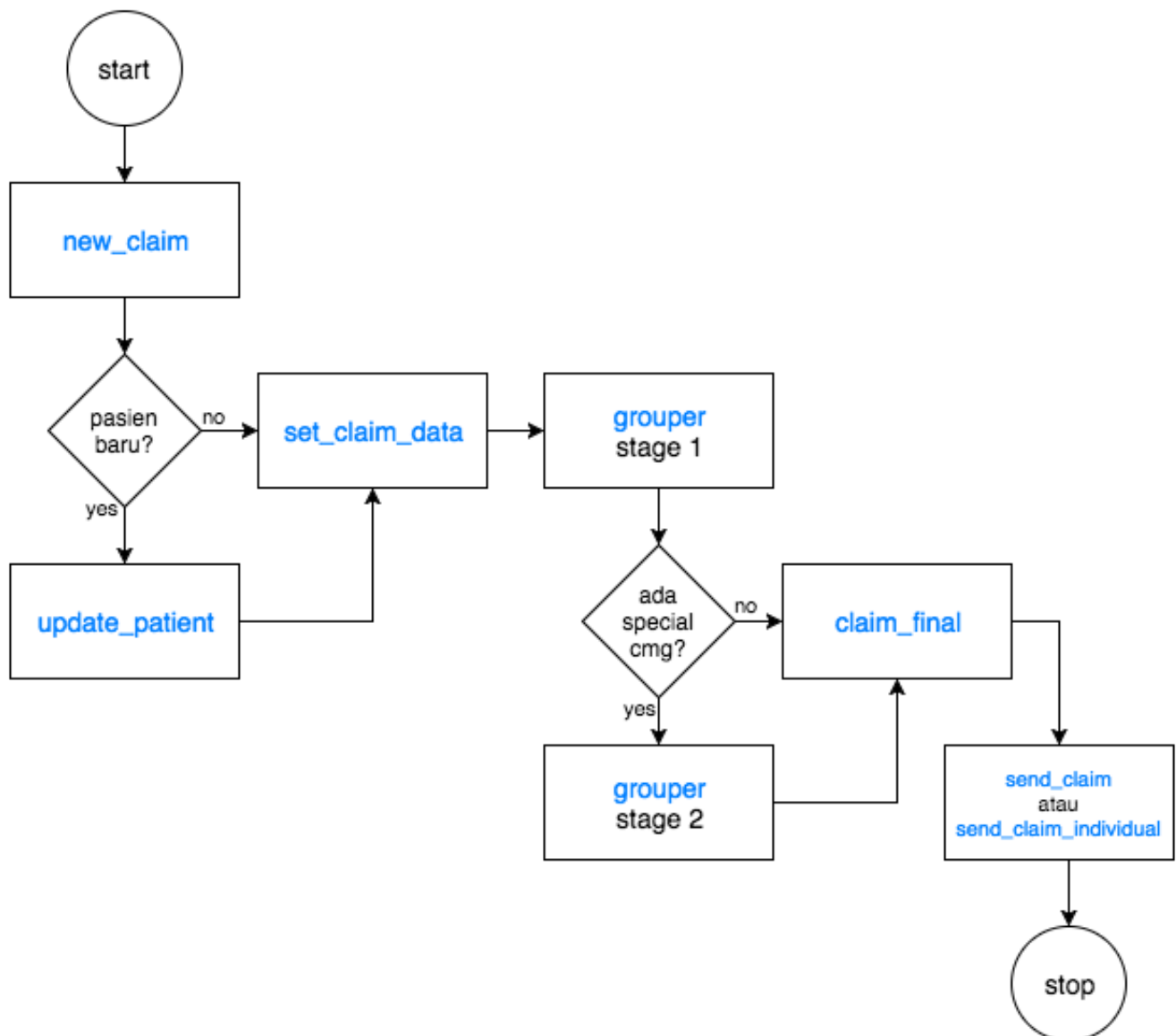
```
        "Other cesarean section of unspecified type",  
        "74.99"  
    ],  
    [  
        "Hysterotomy to terminate pregnancy",  
        "74.91"  
    ]  
]  
}  
}
```

## DAFTAR KODE ERROR

Kode	Deksripsi
E2001	Method tidak ada
E2002	Klaim belum final
E2003	Nomor SEP terduplikasi
E2004	Nomor SEP tidak ditemukan
E2005	NIK Coder masih kosong
E2006	NIK Coder tidak ditemukan
E2007	Duplikasi nomor SEP
E2008	Nomor RM tidak ditemukan
E2009	Klaim sudah final
E2010	Nomor SEP baru sudah terpakai
E2011	Klaim tidak bisa diubah/edit
E2012	Tanggal Pulang mendahului Tanggal Masuk
E2013	Lama rawat intensif melebihi total lama rawat
E2014	Kode tarif invalid
E2015	Kode RS belum disetup
E2016	CBG Code invalid, tidak bisa final
E2017	Klaim belum digrouping
E2018	Klaim masih belum final
E2019	Tanggal invalid
E2020	Response web service SEP kosong
E2021	Error : Gagal men-decode JSON - Maximum stack depth exceeded
E2022	Error : Gagal men-decode JSON - Underflow or the modes mismatch
E2023	Error : Gagal men-decode JSON - Unexpected control character found
E2024	Error : Gagal men-decode JSON - Syntax error, malformed JSON
E2025	Error : Gagal men-decode JSON - Malformed UTF-8 characters
E2026	Error : Gagal men-decode JSON - Unknown error
E2027	Rumah sakit belum terdaftar
E2028	Jenis rawat invalid
E2029	Error: Koneksi gagal
E2030	Parameter tidak lengkap
E2031	Error: Key Mismatch
E2099	Error tidak diketahui

## ALUR DASAR INTEGRASI (BASIC INTEGRATION FLOW)

Berikut ini adalah alur dasar yang dapat dipakai sebagai acuan minimal untuk mengintegrasikan SIMRS dengan E-Klaim. Method-method yang digunakan adalah contoh minimal, method yang lain silakan ditambahkan atau digunakan sesuai kebutuhan. Tulisan yang berwarna biru adalah nama method.



### Changelog:

#### 20171130

- Update hasil get\_claim\_data untuk menampilkan format tarif\_rs.

#### 20171128

- Penambahan parameter tarif breakdown pada set\_claim\_data.

- Breakdown parameter tarif\_rs pada set\_claim\_data.

- Pada method send\_claim, parameter jenis\_rawat ada penambahan value yaitu "3" (tiga) untuk rawat inap dan rawat jalan

- Pada method `send_claim` sekarang bisa memilih tanggal pulang atau tanggal grouping yaitu dengan penambahan parameter `date_type`, yaitu untuk menentukan bahwa parameter `start_dt` dan `stop_dt` adalah tanggal pulang atau tanggal grouping

#### **20170712**

- Fix "Error tidak diketahui" menjadi "Error key mismatch" untuk response `KEY_MISMATCH`

#### **20170605**

- Fix gender pada method `get_claim_data`

#### **20170605**

- Penambahan method `search_diagnosis`
- Penambahan method `search_procedures`
- Koreksi typo pada method `delete_patient`
- Fix bug `new_claim` ketika pasien sudah dihapus
- Fix `delete_patient` untuk no rm yang sama

#### **20170518**

- Penambahan katalog fungsi enkripsi / dekripsi dalam beberapa bahasa pemrograman di bagian akhir manual web service
- Refactoring, fungsi `php mc_*` menjadi `inacbg_*`
- Koreksi manual web service untuk naik kelas `vvip`
- Penambahan konfigurasi `enable_debug` di `server.ini` pada segment `[web_service]` untuk security

#### **20170511**

- Penambahan error code `E2030` Parameter tidak lengkap, sebagai response web service yang tidak menyertakan salah satu parameter yang dibutuhkan (mandatory)

#### **20170405**

- Penambahan parameter `cob_cd` pada method `set_claim_data`

#### **20170320**

- Penambahan error code `E2029` dan `E2099`
- Penambahan info jika terjadi kegagalan koneksi ketika `send_claim_individual`

#### **20170316**

- Penambahan parameter `add_payment_pct` pada method `set_claim_data`
- Penambahan result parameter `add_payment_amt` pada method `grouper` dan `get_claim_data`

#### **20161219**

- Penambahan kode error (`error_no`) pada setiap reponse dengan kesalahan
- Penambahan check duplikasi nomor sep untuk setiap method yang menggunakan nomor sep
- Penyeragaman format json variable hasil `grouper` dan `get_claim_data`
- Penambahan informasi `patient_id`, `admission_id` dan `hospital_admission_id` untuk response `new_claim` dan `get_claim_data`

#### **20161216**

- Penambahan method `claim_print`.
- Penambahan informasi tarif kelas 1,2 dan 3 untuk setiap response `grouper` dan `get_claim_data`. Dengan perubahan ini dimohon untuk setiap `simrs` yang telah melakukan integrasi sebelum ini untuk menyesuaikan kembali dengan format yang baru.
- Fix kode cara pulang (5 = Lain-lain) pada cetak klaim individual dan txt.
- Fix method `grouper` untuk klaim yang telah dihapus.
- Fix untuk `set_claim_data` pada saat `grouper` telah terfinal.
- Perubahan tanda delimiter untuk diagnosa dan prosedur pada method `get_claim_data` yang sebelumnya semicolon (;) menjadi hash (#).

#### **20161212**

- Penambahan parameter untuk ubah nomor\_kartu pada method `set_claim_data`

- Penambahan parameter untuk naik kelas: `upgrade_class_ind`, `upgrade_class_class` dan `upgrade_class_los` pada method `set_claim_data`

#### **20161123**

- Penambahan method `send_claim_individual`
- Perubahan json response untuk `send_claim` untuk key "List" menjadi "data"
- Penyeragaman format encrypted/non-encrypted untuk masing-masing mode

#### **20161116**

- Penambahan method `get_claim_status`

#### **20161111**

- Penambahan envelope key untuk encryption dengan DC Kemkes
- Pemisahan key untuk `pull_claim` oleh client BPJS

#### **20161020**

- Penambahan flag untuk poli eksekutif

#### **20160514**

- Fix mandatory `coder_nik` di `new_claim` masih bisa tembus, dan set NIK internal user supaya kosong

#### **20160511**

- Encryption & Decryption dan mode debug untuk development
- Update manual

#### **20160502**

- Waktu grouping adalah waktu yg dicatat ketika pemanggilan method `set_claim_data`, grouper dan `claim_final`. Untuk NIK Coder hanya dicatat pada pemanggilan method `set_claim_data`.
- NIK Coder sekarang mandatory dalam method `set_claim_data`, dan NIK tersebut harus terregister dalam data user.
- Fix penambahan kode ICD10 dan ICD9CM yang masih belum ada.
- Status Klaim "Siap" dihilangkan, diganti "Final" supaya lebih simple.
- Gender pada method `new_claim` dan `update_patient` berubah dari L/P menjadi 1 = Laki / 2 = Perempuan.
- Penambahan method `delete_claim`.
- Penambahan method `delete_patient`.
- Penambahan method `update_patient`.
- Penambahan method `get_claim_data`.
- Untuk `set_claim_data` ada penambahan metadata `nomor_sep` sebagai identifier, sedangkan yang `nomor_sep` didalam data adalah sebagai nilai perubahan jika akan dilakukan perubahan.
- Fix rounding tarif sub acute dan chronic.
- Penambahan kode cbg X-0-99-X FAILED: EMPTY RESPONSE, supaya lebih informatif untuk kasus UNU Grouper crash. Terkait juga dengan hasil grouping minus.
- Fix bug nama dengan single quote untuk simpan melalui ws

#### **20160421**

- Fix grouping untuk special CMG lebih dari 1.
- Fix error unduh data.
- Fix error untuk `nomor_sep` beda dalam 1 pasien.

KATALOG FUNGSI ENKRIPSI / DEKRIPSI  
DALAM BEBERAPA BAHASA PEMROGRAMAN

## PHP

```
// Encryption Function
function inacbg_encrypt($data, $key) {
    /// make binary representasion of $key
    $key = hex2bin($key);

    /// check key length, must be 256 bit or 32 bytes
    if (mb_strlen($key, "8bit") !== 32) {
        throw new Exception("Needs a 256-bit key!");
    }

    /// create initialization vector
    $iv_size = openssl_cipher_iv_length("aes-256-cbc");
    $iv = openssl_random_pseudo_bytes($iv_size); // dengan catatan dibawah

    /// encrypt
    $encrypted = openssl_encrypt($data, "aes-256-cbc", $key, OPENSSL_RAW_DATA, $iv);

    /// create signature, against padding oracle attacks
    $signature = mb_substr(hash_hmac("sha256", $encrypted, $key, true), 0, 10, "8bit");

    /// combine all, encode, and format
    $encoded = chunk_split(base64_encode($signature.$iv.$encrypted));

    return $encoded;
}

// Decryption Function
function inacbg_decrypt($str, $strkey){
    /// make binary representation of $key
    $key = hex2bin($strkey);

    /// check key length, must be 256 bit or 32 bytes
    if (mb_strlen($key, "8bit") !== 32) {
        throw new Exception("Needs a 256-bit key!");
    }

    /// calculate iv size
    $iv_size = openssl_cipher_iv_length("aes-256-cbc");

    /// breakdown parts
    $decoded = base64_decode($str);
    $signature = mb_substr($decoded, 0, 10, "8bit");
    $iv = mb_substr($decoded, 10, $iv_size, "8bit");
    $encrypted = mb_substr($decoded, $iv_size+10, NULL, "8bit");

    /// check signature, against padding oracle attack
    $calc_signature = mb_substr(hash_hmac("sha256", $encrypted, $key, true), 0, 10, "8bit");
    if(!inacbg_compare($signature, $calc_signature)) {
        return "SIGNATURE_NOT_MATCH"; /// signature doesn't match
    }

    $decrypted = openssl_decrypt($encrypted, "aes-256-cbc", $key, OPENSSL_RAW_DATA, $iv);

    return $decrypted;
}

/// Compare Function
function inacbg_compare($a, $b) {
    /// compare Individually to prevent timing attacks

    /// compare length
    if (strlen($a) !== strlen($b)) return false;

    /// compare individual
    $result = 0;
    for($i = 0; $i < strlen($a); $i++) {
        $result |= ord($a[$i]) ^ ord($b[$i]);
    }

    return $result == 0;
}
```

```

C#
// ENCRYPT

public string inacbg_encrypt(string text, string key) {
    var keys = Encoding.Default.GetBytes(hex2bin(key));
    AesCryptoServiceProvider aes = new AesCryptoServiceProvider();
    aes.BlockSize = 128;
    aes.KeySize = 256;
    aes.GenerateIV();
    var iv = aes.IV;
    aes.Key = keys;
    aes.Mode = CipherMode.CBC;
    aes.Padding = PaddingMode.PKCS7;
    byte[] src = Encoding.Default.GetBytes(text);

    using (ICryptoTransform encrypt = aes.CreateEncryptor()) {
        byte[] data = encrypt.TransformFinalBlock(src, 0, src.Length);

        HMACSHA256 hashObject = new HMACSHA256(keys);
        var hash_sign = hashObject.ComputeHash(data);
        byte[] signature = new byte[10];
        Array.Copy(hash_sign, 0, signature, 0, 10);

        byte[] ret = new byte[signature.Length + iv.Length + data.Length];
        Array.Copy(signature, 0, ret, 0, signature.Length);
        Array.Copy(iv, 0, ret, signature.Length, iv.Length);
        Array.Copy(data, 0, ret, signature.Length + iv.Length, data.Length);

        return Convert.ToBase64String(ret);
    }
}

// DECRYPT

public string inacbg_decrypt(string stencrypt, string key) {
    string encoded_str = stencrypt;
    byte[] chipper = Convert.FromBase64String(encoded_str);

    var length = chipper.Length;
    byte[] new_byte_iv = new byte[16];
    byte[] new_byte_msg = new byte[length - 26];
    Array.Copy(chipper, 10, new_byte_iv, 0, 16);
    Array.Copy(chipper, 26, new_byte_msg, 0, length - 26);

    byte[] byte_key = Encoding.Default.GetBytes(hex2bin(key));

    RijndaelManaged aes = new RijndaelManaged();
    aes.KeySize = 256;
    aes.BlockSize = 128;
    aes.Padding = PaddingMode.PKCS7;
    aes.Mode = CipherMode.CBC;
    aes.Key = byte_key;
    aes.IV = new_byte_iv;

    ICryptoTransform AESDecrypt = aes.CreateDecryptor(aes.Key, aes.IV);
    return Encoding.Default.GetString(AESDecrypt.TransformFinalBlock(new_byte_msg,
                                                                    0,
                                                                    new_byte_msg.Length));
}

private static string hex2bin(string input) {
    input = input.Replace("-", "");
    byte[] raw = new byte[input.Length / 2];
    for (int i = 0; i < raw.Length; i++) {
        raw[i] = Convert.ToByte(input.Substring(i * 2, 2), 16);
    }
    return Encoding.Default.GetString(raw);
}

```



## VB.NET

```
Imports System.Text
Imports System.Security.Cryptography

Module inacbg_encryption

    ' ENCRYPT
    Public Function inacbg_encrypt(text As String, key As String) As String
        Dim keys = Encoding.[Default].GetBytes(hex2bin(key))
        Dim aes As New AesCryptoServiceProvider()
        aes.BlockSize = 128
        aes.KeySize = 256
        aes.GenerateIV()
        Dim iv = aes.IV
        aes.Key = keys
        aes.Mode = CipherMode.CBC
        aes.Padding = PaddingMode.PKCS7
        Dim src As Byte() = Encoding.[Default].GetBytes(text)

        Using enc As ICryptoTransform = aes.CreateEncryptor()
            Dim data As Byte() = enc.TransformFinalBlock(src, 0, src.Length)

            Dim hashObject As New HMACSHA256(keys)
            Dim hash_sign = hashObject.ComputeHash(data)
            Dim signature As Byte() = New Byte(9) {}
            Array.Copy(hash_sign, 0, signature, 0, 10)

            Dim ret As Byte() = New Byte(signature.Length + iv.Length + (data.Length - 1)) {}
            Array.Copy(signature, 0, ret, 0, signature.Length)
            Array.Copy(iv, 0, ret, signature.Length, iv.Length)
            Array.Copy(data, 0, ret, signature.Length + iv.Length, data.Length)

            Return Convert.ToBase64String(ret)
        End Using
    End Function

    ' DECRYPT
    Public Function inacbg_decrypt(strencrypt As String, key As String) As String
        Dim encoded_str As String = strencrypt
        Dim chiper As Byte() = Convert.FromBase64String(encoded_str)

        Dim length = chiper.Length
        Dim new_byte_iv As Byte() = New Byte(15) {}
        Dim new_byte_msg As Byte() = New Byte(length - 27) {}
        Array.Copy(chiper, 10, new_byte_iv, 0, 16)
        Array.Copy(chiper, 26, new_byte_msg, 0, length - 26)

        Dim byte_key As Byte() = Encoding.[Default].GetBytes(hex2bin(key))

        Dim aes As New RijndaelManaged()
        aes.KeySize = 256
        aes.BlockSize = 128
        aes.Padding = PaddingMode.PKCS7
        aes.Mode = CipherMode.CBC
        aes.Key = byte_key
        aes.IV = new_byte_iv

        Dim AESDecrypt As ICryptoTransform = aes.CreateDecryptor(aes.Key, aes.IV)
        Return Encoding.[Default].GetString(AESDecrypt.TransformFinalBlock(new_byte_msg, 0, new_byte_msg.Length))
    End Function

    Private Shared Function hex2bin(input As String) As String
        input = input.Replace("-", "")
        Dim raw As Byte() = New Byte(input.Length / 2 - 1) {}
        For i As Integer = 0 To raw.Length - 1
            raw(i) = Convert.ToByte(input.Substring(i * 2, 2), 16)
        Next
        Return Encoding.[Default].GetString(raw)
    End Function
End Module
```

## JavaScript

```
const crypto = require('crypto');

const key = '';
const uri = '';

const inacbg_decrypt = (data)=>{
  //Replacing Text
  if(typeof data==='string'){
    data = data.replace(/-----BEGIN ENCRYPTED DATA-----|-----END ENCRYPTED DATA-----/g, '');
  }else{
    return `Should be String input`;
  }
  //make Key to binary type, stored in Buffer
  let keys = Buffer.from(key, 'hex');
  //make data to binary type, stored in Buffer
  let data_decoded = Buffer.from(data, 'base64');
  //make iv to binary type, stored in Buffer
  let iv = Buffer.from(data_decoded.slice(10, 26));
  //create Decipher with IV to decode data
  let dec = crypto.createDecipheriv('aes-256-cbc', keys, iv);
  //cutting data that has binary type -- 26 is 10 for char and 16 for IV for aes-256-cbc
  let encoded = Buffer.from(data_decoded.slice(26));
  //take Signature
  let signature = data_decoded.slice(0, 10);
  //check if signature is right
  if(!inacbg_compare(signature, encoded)) {
    return "SIGNATURE_NOT_MATCH"; /// signature doesn't match
  }
  //decrypt data
  let decrypted = Buffer.concat([dec.update(encoded), dec.final()]);
  return decrypted.toString('utf8');
}

const inacbg_encrypt = (data)=>{
  //stringify when data os object
  if(typeof data === 'object'){
    data = JSON.stringify(data);
  }
  //make Key to binary type, stored in Buffer
  let keys = Buffer.from(key, 'hex');
  //make data to binary type, stored in Buffer
  let data_encoded = Buffer.from(data);
  //make iv 16 byte of random
  let iv = crypto.randomBytes(16);
  //create cypher for encrypt
  let enc = crypto.createCipheriv('aes-256-cbc', keys, iv);
  // encrypt data
  let encrypt = Buffer.concat([enc.update(data_encoded), enc.final()]);
  //create signature
  let signature = crypto.createHmac('sha256', keys)
    .update(encrypt)
    .digest()
    .slice(0, 10);
  //concat buffer then return in string encode with base64
  return Buffer.concat([signature, iv, encrypt]).toString('base64');
}

const inacbg_compare = (signature, encrypt) => {
  let keys = Buffer.from(key, 'hex');
  let calc_signature = crypto.createHmac('sha256', keys)
    .update(encrypt)
    .digest()
    .slice(0, 10);

  if(signature.compare(calc_signature)===0){
    return true;
  }
  return false;
}
```