

Web Service E-Klaim

Untuk Build 5.10.x

Table of Content

Table of Content.....	1
I. SETUP.....	3
II. WEB SERVICE.....	6
III. ENKRIPSI / DEKRIPSI.....	6
IV. KATALOG METHOD WEB SERVICE.....	9
1. Membuat klaim baru (dan registrasi pasien jika belum ada).....	10
2. Update data pasien.....	12
3. Hapus data pasien.....	12
4. Untuk mengisi/update data klaim.....	13
5. Set Diagnosa iDRG.....	22
6. Get Diagnosa iDRG.....	23
7. Set Prosedur iDRG.....	25
8. Get Prosedur iDRG.....	27
9. Grouping iDRG Stage 1.....	29
10. Finalisasi iDRG.....	30
11. Re-edit coding iDRG.....	30
12. Import Coding iDRG ke INACBG.....	31
13. Set Diagnosa INACBG.....	33
14. Set Prosedur INACBG.....	34
15. Grouping INACBG Stage 1.....	35
16. Grouping INACBG Stage 2.....	36
17. Finalisasi INACBG.....	38
18. Re-edit coding INACBG.....	38
19. Finalisasi Klaim.....	39
20. Re-edit Klaim.....	39
21. Kirim online klaim kolektif (per hari) ke data center.....	39
22. Kirim online klaim individual ke data center.....	40
23. Get data detail per klaim.....	41
24. Untuk mengambil status per klaim.....	43
25. Hapus klaim.....	44
26. Cetak klaim.....	45
27. Pencarian diagnosa iDRG.....	45
28. Pencarian prosedur iDRG.....	46
29. Pencarian diagnosa INACBG.....	49
30. Pencarian prosedur INACBG.....	49
31. Generate nomor pengajuan klaim.....	51
32. Validasi Nomor Register SITB.....	52
33. Membatalkan validasi Nomor Register SITB:.....	53

V. DAFTAR KODE ERROR.....	54
VI. DAFTAR KODE UNGROUPABLE AND UNRELATED.....	57
VII. ALUR DASAR INTEGRASI (BASIC INTEGRATION FLOW).....	59
IX. ALUR INTEGRASI DENGAN SITB.....	60
X. CHANGELOG.....	61
XI. KATALOG FUNGSI ENKRIPSI / DEKRIPSI DALAM BEBERAPA BAHASA PEMROGRAMAN.	
65	
PHP.....	65
C#.....	67
VB.NET.....	69
JavaScript.....	71
Python.....	73
Python.....	74
DAFTAR PUSTAKA.....	76

I. SETUP

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

Home	Setup	Batch Grouper	Migrasi	Utilitas	Personnel	Akun
------	-------	---------------	---------	----------	-----------	------

SETUP INTEGRASI SIMRS

Konfigurasi	
Kode RS	0000000
Encryption Key	-
Generated Datetime	-

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

Generate Key

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

Home	Setup	Batch Grouper	Migrasi	Utilitas	Personnel	Akun
------	-------	---------------	---------	----------	-----------	------

SETUP INTEGRASI SIMRS

Konfigurasi	
Kode RS	0000000
Encryption Key	-
Generated Datetime	-

Anda akan men-generate Encryption Key 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.

Anda akan men-generate Encryption Key baru.

Generate Encryption Key?

Captcha :



Masukkan Tulisan Pada Gambar Captcha :

Masukkan Password Anda :

Setelah itu muncul re-konfirmasi dengan memasukkan kode yang tertera pada gambar dan memasukkan password Anda, kemudian klik tombol **Ya (Generate)**. Hasilnya:

[Home](#)

[Setup](#)

[Batch Grouper](#)

[Migrasi](#)

[Utilitas](#)

[Personnel](#)

[Akun](#)

SETUP INTEGRASI SIMRS

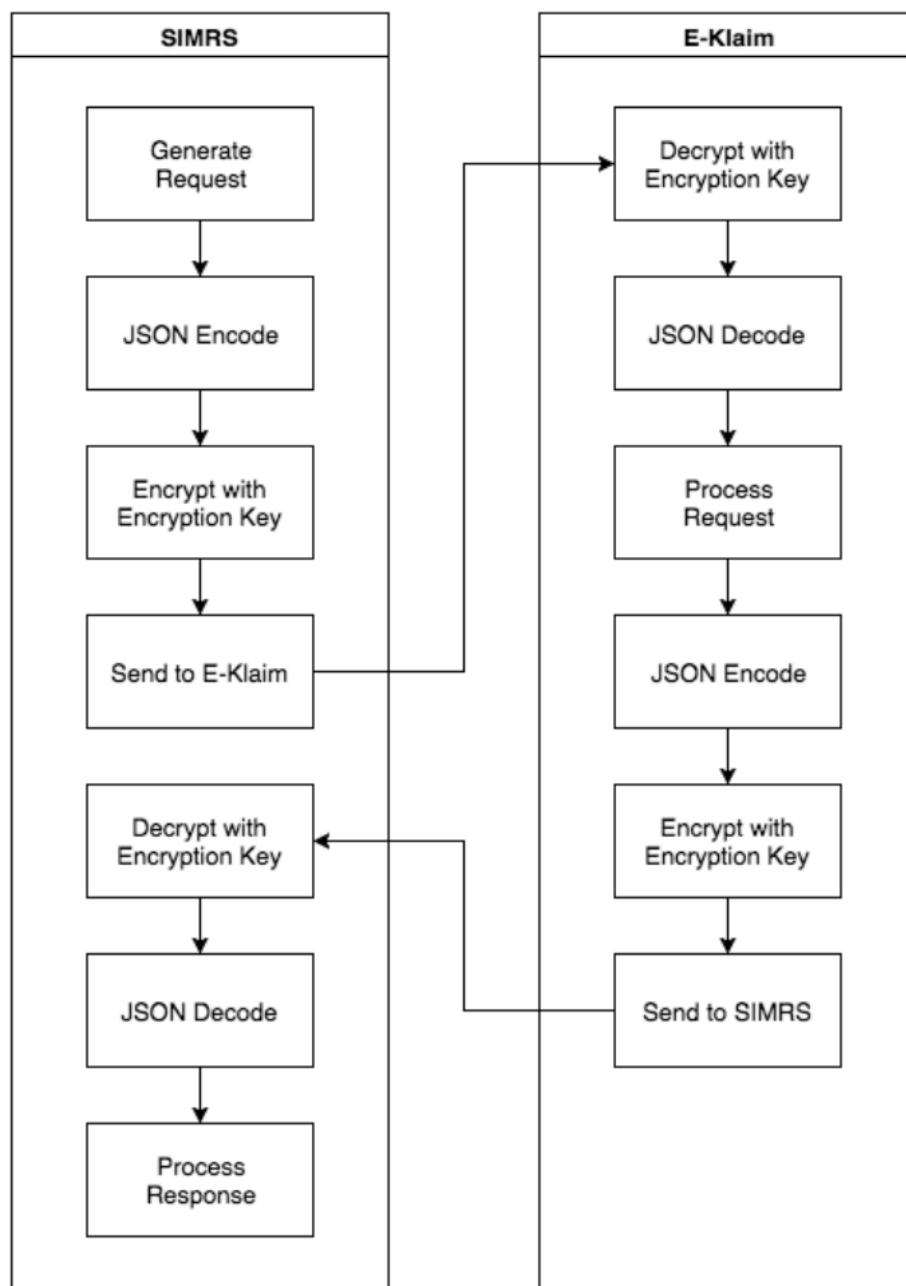
Konfigurasi

Kode RS	0000000
Encryption Key	f86a8662cfd3a3df7e0673c39fa4cc97c8a9107ae5bae1fe4ae99c0c903a1080
Generated Datetime	2025-10-24 17:02:40

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

Encryption Key akan di-generate 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 `C:\E-Klaim\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----
/KsK5I2TcjfU6gu2pBwjANNvPRUrrpmqVgLkIZdUyUts1hz9xSk9ECgjgMu5UBqSOeymPAA+DGF+
M32WFSIr0dj/ctsKXTJEYupxVBQ5Fxe8pwEbheIEPMXlr2Z/ZsCqZvHQpPknNySiwnKrX/9sZSMj
9pCWY9Al1Gz9mSenkAsaGab9FkjZwOP7K4ERA/dxIrcNMFJUj36X/yvspM+VQ0it4GNvq0duoSv7
Ckn5g3U+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.

```
<?php
// Encryption Function
function inacbg_encrypt($data, $key)
```

```

{
    /// make binary representation 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 web 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;
$last = strpos($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, di-translate ke dalam 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). 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 Postman 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 di bagian bawah pada bagian daftar kode error.

Berikut ini daftar method:

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

Request Body:

```

{
  "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. Khusus pasien jenis berikut, Nomor SEP diperoleh dari method "generate_claim_number" (no. 18). Jenisnya yaitu:

- JAMINAN COVID-19

- JAMINAN KIPI
- JAMINAN BAYI BARU LAHIR
- JAMINAN CO-INSIDENSE

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

Request Body:

```
{
  "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

Request Body:

```
{
  "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. Coder harus terdaftar pada aplikasi E-Klaim. Jika belum terdaftar maka perlu didaftarkan melalui aplikasi E-Klaim.

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

4. Untuk mengisi/update data klaim

Request Body:

```
{
  "metadata": {
    "method": "set_claim_data",
    "nomor_sep": "0001R0016120507422"
  },
  "data": {
    "nomor_sep": "0001R0016120507422",
    "nomor_kartu": "233333",
    "tgl_masuk": "2023-01-25 12:55:00",
    "tgl_pulang": "2023-01-31 09:55:00",
    "cara_masuk": "gp",
    "jenis_rawat": "1",
    "kelas_rawat": "1",
    "adl_sub_acute": "15",
    "adl_chronic": "12",
    "icu_indikator": "1",
    "icu_los": "2",
    "ventilator_hour": "5",
    "ventilator": {
      "use_ind": "1",
      "start_dttm": "2023-01-26 12:55:00",
      "stop_dttm": "2023-02-26 17:50:00"
    },
    "upgrade_class_ind": "1",
    "upgrade_class_class": "vip",
    "upgrade_class_los": "5",
    "upgrade_class_payor": "peserta",
    "add_payment_pct": "35",
    "birth_weight": "0",
    "sistole": 120,
    "diastole": 70,
    "discharge_status": "1",
    "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": "100000",
    "obat_kronis": "1000000",
    "obat_kemoterapi": "5000000",
    "alkes": "500000",
    "bmhp": "400000",
    "sewa_alat": "210000"
  },
  "nomor_kartu_t": "nik",
  "bayi_lahir_status_cd": 1,
  "dializer_single_use": 0,
  "kantong_darah": 1,
  "alteplase_ind": 0,
  "apgar": {
    "menit_1": {
      "appearance": 1,
      "pulse": 2,
      "grimace": 1,
      "activity": 1,
      "respiration": 1
    },
    "menit_5": {
      "appearance": 2,
      "pulse": 2,
      "grimace": 2,
      "activity": 2,
      "respiration": 2
    }
  },
  "persalinan": {
    "usia_kehamilan": "22",
    "gravida": "2",
    "partus": "4",
    "abortus": "2",
    "onset_kontraksi": "induksi",
    "delivery": [
      {
        "delivery_sequence": "1",
        "delivery_method": "vaginal",
        "delivery_dttm": "2023-01-21 17:01:33",
        "letak_janin": "kepala",
        "kondisi": "livebirth",
        "use_manual": "1",
        "use_forcep": "0",

```

```

        "use_vacuum": "1",
        "shk_spesimen_ambil": "ya",
        "shk_lokasi": "tumit",
        "shk_spesimen_dttm": "2023-11-21 18:11:33"
    },
    {
        "delivery_sequence": "2",
        "delivery_method": "vaginal",
        "delivery_dttm": "2023-01-21 17:03:49",
        "letak_janin": "lintang",
        "kondisi": "livebirth",
        "use_manual": "1",
        "use_forcep": "0",
        "use_vacuum": "0",
        "shk_spesimen_ambil": "tidak",
        "shk_alasan": "akses-sulit"
    }
]
},
{
    "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
cara_masuk	: gp = Rujukan FKTP, hosp-trans = Rujukan FKRTL, mp = Rujukan Spesialis, outp = Dari Rawat Jalan, inp = Dari Rawat Inap, emd = Dari Rawat Darurat, born = Lahir di RS, nursing = Rujukan Panti Jompo, psych = Rujukan dari RS Jiwa, rehab = Rujukan Fasilitas Rehab, other = Lain-lain
jenis_rawat	: 1 = rawat inap, 2 = rawat jalan, 3 = rawat igd
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
 Tambahan element ventilator:
use_ind: 1 = ada pemakaian, 0 = tidak ada pemakaian
start_dttm: waktu mulai, format yyyy-mm-dd hh:mm:ss
stop_dttm: waktu selesai, format yyyy-mm-dd hh:mm:ss
 Tambahan element ventilator ini tidak menghilangkan element ventilator_hour sebelumnya untuk backward compatibility

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 (diatas Kelas 1). Parameter **upgrade_class_payor** diisi dengan "peserta" atau "pemberi_kerja" atau "asuransi_tambahan". Untuk penggunaan parameter **upgrade_class_ind**, **upgrade_class_class**, **upgrade_class_los** dan **add_payment_pct**, **upgrade_class_payor** harus disertakan 5 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

Parameter **payor_id** ini menjadi mandatory sejak versi 5.4.11.

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

sistole & diastole: Tekanan darah, dalam mmHg

dializer_single_use: Khusus untuk hemodialisa (CBG: **N-3-15-0**), diisi:

- "1" = single use
- "0" = multiple use

Q Cari Diagnosa

Q 9904

Hemodialysis 39.95 Primary

Import Coding

Grouping

Info	1110101 @ 21 Okt 2025 18:03 •• Kelas C •• Tarif : TARIF RS KELAS C SWASTA				
Jenis Rawat	Rawat Jalan Regular				
Group	PROSEDUR DIALISIS	N-3-15-0	Rp	875,000	
Sub Acute	-	-	Rp	0	
Chronic	-	-	Rp	0	
Special Procedure	None	-	Rp	0	
Special Prosthesis	None	-	Rp	0	
Special Investigation	None	-	Rp	0	
Special Drug	None	-	Rp	0	
Penggunaan Dializer	<input checked="" type="radio"/> Multiple Use (reuse) <input type="radio"/> Single Use		85%	Rp	743,750
Transfusi Darah	Jumlah kantong darah: 0 kantong	-			
Total Klaim			Rp	743,750	
Status	normal				

[debug]

Final INACBG

delivery method: diisi pilihan: **vaginal**, atau **sc**

delivery_dttm: waktu kelahiran, format yyyy-mm-dd hh:mm:ss
letak_janin: letak janin, diisi pilihan: **kepala, sungsang,**
lintang kondisi: kondisi bayi waktu lahir, pilihan: **livebirth,**
stillbirth
use_manual: lahir dengan bantuan manual, diisi 0 = tidak, 1 = ya
use_forcep: penggunaan forcep, diisi 0 = tidak, 1 = ya
use_vacuum: penggunaan vacuum, diisi 0 = tidak, 1 = ya
shk_spesimen_ambil: diisi "ya", "tidak"
shk_lokasi: diisi "tumit", "vena"
shk_alasan: diisi "tidak-dapat", "akses-sulit"
shk_spesimen_dttm: waktu pengambilan, format yyyy-mm-dd hh:mm:ss

kantong_darah: Diisi dengan jumlah kantong darah yang diberikan kepada pasien. Parameter ini digunakan berdampingan dengan parameter **pelayanan_darah** pada **tarif_rs**. dimunculkan ketika :

1. Kode CBGnya adalah N-3-15-0,
2. Terdapat diagnosa Beta thalassaemia (D56.1) atau
3. Terdapat diagnosa Leukimia (C91* - C95*).

Ketentuan lebih lanjut merujuk kepada Permenkes RI No. 3 Tahun 2023, Pasal 45 ayat 1, "Pelayanan kantong darah sebagaimana dimaksud dalam Pasal 36 ayat (1) huruf h diberikan untuk thalassemia mayor, hemodialisa, dan kanker (leukemia) yang membutuhkan pelayanan darah pada rawat jalan."

Diagnosa (ICD-10):

Q Cari Diagnosa

Follow-up exam after other treatment for other conditions **Z09.8** Primary

Prosedur (ICD-9-CM):

Q 9904

Hemodialysis **39.95** Primary

Import Coding

Grouping

Hasil Grouping INACBG

Info	1110101 @ 21 Okt 2025 18:03 •• Kelas C •• Tarif : TARIF RS KELAS C SWASTA				
Jenis Rawat	Rawat Jalan Regular				
Group	PROSEDUR DIALISIS	N-3-15-0	Rp	875,000	
Sub Acute	-	-	Rp	0	
Chronic	-	-	Rp	0	
Special Procedure	None	-	Rp	0	
Special Prosthesis	None	-	Rp	0	
Special Investigation	None	-	Rp	0	
Special Drug	None	-	Rp	0	
Penggunaan Dializer	<input checked="" type="radio"/> Multiple Use (reuse) <input type="radio"/> Single Use		85%	Rp	743,750
Transfusi Darah	Jumlah kantong darah: 0 kantong	-			
Total Klaim				Rp	743,750
Status	normal				

[debug]

Final INACBG

alteplase_ind: Diisi 1 jika ada pemberian alteplase, 0 jika tidak.
pilihan alteplase hanya dimunculkan ketika kode CBG
adalah G-4-14-* KECEDERAAN PEMBULUH DARAH OTAK DENGAN
INFARK, semua level, baik RINGAN, SEDANG, dan BERAT.

Jenis Rawat	Rawat Inap Kelas 2 (4 Hari)			
Group	KECEDERAAN PEMBULUH DARAH OTAK DENGAN INFARK (RINGAN)	G-4-14-I	Rp	8,529,100
Sub Acute	-	-	Rp	0
Chronic	-	-	Rp	0
Special Procedure	None		Rp	0
Special Prosthesis	None		Rp	0
Special Investigation	None		Rp	0
Special Drug	None		Rp	0
Pemberian Alteplase	<input type="radio"/> Tidak diberikan <input checked="" type="radio"/> Diberikan -			
Total Klaim			Rp	8,529,100
Status	normal			

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, obat_kronis, obat_kemoterapi, alkes, , bmhp, dan sewa_alat.** Masing-masing diisi dengan nilai integer. Untuk definisi operasional parameter tersebut silakan merujuk pada petunjuk teknis Aplikasi E-Klaim.

nomor_kartu_t : Untuk tambahan khusus pasien Jaminan COVID-19, parameter ini membedakan nilai yang tersebut di dalam parameter **nomor_kartu**. Isinya dengan pilihan:

nik = untuk Nomor Induk Kependudukan

kitas = untuk KITAS/KITAP.

KITAS : Kartu Ijin Tinggal Terbatas

KITAP : Kartu Ijin Tinggal Tetap

paspor = untuk Nomor Passport, jika WNA.

kartu_jkn = untuk Nomor Kartu Peserta JKN (BPJS

kk = untuk nomor pada Kartu Keluarga

unhcr = untuk nomor pada dokumen dari UNHCR
kelurahan = untuk nomor pada dokumen dari kelurahan
dinsos = untuk nomor pada dokumen dari Dinas Sosial
dinkes = untuk nomor pada dokumen dari Dinas Kesehatan
sjp = untuk nomor Surat Jaminan Perawatan (SJP)
klaim_ibu = mandatori untuk jaminan bayi baru lahir.
lainnya = untuk nomor identitas lainnya yang dapat dipertanggungjawabkan oleh rumah sakit dan lembaga yang berwenang lainnya

bayi_lahir_status_cd : Diisi hanya jika **payor_id** = 73 (Jaminan Bayi Baru Lahir) dengan nilai acuan sebagai berikut:
1 = Tanpa Kelainan
2 = Dengan Kelainan

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. Set Diagnosa iDRG

Request Body:

```
{
  "metadata": {
    "method": "idrg_diagnosa_set",
    "nomor_sep": "000R000TEST"
  },
  "data": {
    "diagnosa": "S71.0#S87.9#E11.9"
  }
}
```

Response :

```
{
  "metadata": {
    "code": 200,
    "message": "Ok",
    "method": "idrg_diagnosa_set"
  },
  "data": {
    "string": "S71.0#S87.9#E11.9",
    "expanded": [
      {
        "code": "S71.0",
        "display": "Open wound of hip",
        "no": "1",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      },
      {
        "code": "S87.9",
        "display": "Crushing injury of unspecified parts of knee and leg (IM)",
        "no": "2",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      },
      {
        "code": "E11.9",
        "display": "Non-insulin-dependent diabetes mellitus without complications",
        "no": "3",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      }
    ]
  }
}
```

diagnosa : Kode diagnosa akan dicek terhadap versi ICD-10-IM (Indonesian Modification) yang berlaku. Jika ada kode yang tidak terdaftar atau berlaku, maka kode tersebut tidak akan tersimpan.

6. Get Diagnosa iDRG

Request Body :

```
{
```



```

    "metadata": {
      "method": "idrg_diagnosa_get"
    },
    "data": {
      "nomor_sep": "000R000TEST"
    }
  }
}

```

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok",
    "method": "idrg_diagnosa_get"
  },
  "data": {
    "string": "S71.0#S87.9#E11.9",
    "expanded": [
      {
        "code": "S71.0",
        "display": "Open wound of hip",
        "no": "1",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      },
      {
        "code": "S87.9",
        "display": "Crushing injury of unspecified parts of knee and leg (IM)",
        "no": "2",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      },
      {
        "code": "E11.9",
        "display": "Non-insulin-dependent diabetes mellitus without complications",
        "no": "3",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      }
    ]
  }
}

```

7. Set Prosedur iDRG

Request Body :

```
{
  "metadata": {
    "method": "idrg_procedure_set",
    "nomor_sep": "000R000TEST"
  },
  "data": {
    "procedure": "81.51#86.28+2#91.799"
  }
}
```

Response :

```
{
  "metadata": {
    "code": 200,
    "message": "Ok",
    "method": "idrg_procedure_set"
  },
  "data": {
    "string": "81.51#86.28+2#91.799",
    "expanded": [
      {
        "code": "81.51",
        "display": "Total hip replacement",
        "multiplicity": 1,
        "no": "1",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      },
      {
        "code": "86.28",
        "display": "Nonexcisional debridement of wound, infection or burn",
        "multiplicity": 2,
        "no": "2",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      },
      {
        "code": "91.799",
        "display": "Other Microscopic examination of specimen from operative wound (IM)",
        "multiplicity": 1,
        "no": "3",
        "validcode": "1",
        "metadata": {
          "code": "200",

```

```

    "message": "Ok"
  }
}
]
}
}

```

Contoh hapus semua data prosedur

Request Body:

```

{
  "metadata": {
    "method": "idrg_procedure_set",
    "nomor_sep": "000R000TEST"
  },
  "data": {
    "procedure": "#"
  }
}

```

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

Request Body:

```

{
  "metadata": {
    "method": "idrg_procedure_set",
    "nomor_sep": "000R000TEST"
  },
  "data": {
    "procedure": ""
  }
}

```

procedure : Kode procedure akan dicek terhadap versi ICD-9-IM (Indonesian Modification) yang berlaku. Jika ada kode yang tidak terdaftar atau berlaku, maka kode tersebut tidak akan tersimpan.

Untuk kode diagnosa dan procedure, disediakan method tersendiri untuk pencarian pada method nomor [27](#) dan [28](#) dibawah.

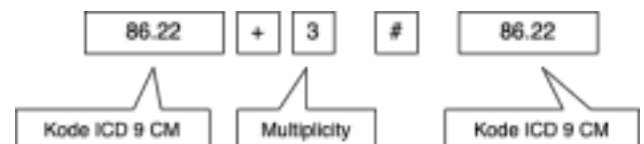
Prosedur pada iDRG dapat dicatat berulang dengan menambahkan kode ekstensi dan/atau ditulis beberapa kali (multiplicity dan setting). Setting untuk menandakan tindakan tersebut dilakukan pada operasi yang berbeda dalam satu episode. Setting ditulis dengan

mencantumkan kode tindakan beberapa kali, misal: "86.22#86.22". Multiplicity digunakan untuk mencatat tindakan tersebut dilakukan beberapa kali di dalam satu kali operasi. Multiplicity dituliskan dengan notasi "+" diikuti jumlahnya, misal: "86.22+2". Demikian pula dimungkinkan untuk mencatat jika tindakan tersebut dilakukan dalam beberapa kali operasi dalam satu episode dan dilakukan lebih dari satu kali dalam salah satu operasinya dengan contoh notasi sebagai berikut: "86.22+2#86.22".

Contoh: tindakan debridement yang dilakukan dalam dua operasi dalam satu episode, yang pada operasi pertama dilakukan tiga kali (tiga lokasi) dan yang pada operasi kedua dilakukan dua kali (dua lokasi).



Contoh: tindakan debridement yang dilakukan dalam dua operasi dalam satu episode, yang pada operasi pertama dilakukan tiga kali (tiga lokasi) dan yang pada operasi kedua dilakukan satu kali (satu lokasi).



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.

8. Get Prosedur iDRG

Request Body :

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

```
    "nomor_sep": "000R000TEST"
  }
}
```

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok",
    "method": "inacbg_procedure_get"
  },
  "data": {
    "string": "81.51#86.28#91.79",
    "expanded": [
      {
        "code": "81.51",
        "display": "Total hip replacement",
        "no": "1",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      },
      {
        "code": "86.28",
        "display": "Nonexcisional debridement of wound, infection, or burn",
        "no": "2",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      },
      {
        "code": "91.79",
        "display": "Microscopic examination of specimen from operative wound, Other microscopic examination",
        "no": "3",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      }
    ]
  }
}
```

9. Grouping iDRG Stage 1

Request Body:

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

Keterangan parameter:

stage : diisi "1" (satu)

grouper : diisi "idrg"

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response_idrg": {
    "mdc_number": "33",
    "mdc_description": "Injuries, Poisonings and Toxic Effects of Drugs",
    "drg_code": "3303110",
    "drg_description": "Other OR Procedures for Injuries w/ No CC",
    "script_version": "1.0.29",
    "logic_version": "0.2.1741.202510071029",
    "cost_weight": "1.34",
    "sub_acute_weight": "0.00",
    "chronic_weight": "0.00",
    "total_cost_weight": "1.34",
    "nbr": "8037060",
    "status_cd": "normal"
  }
}
```

Jika hasil grouping adalah error:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response_idrg": {
    "mdc_number": "36",
    "mdc_description": "Ungroupable or Unrelated",
    "drg_code": "3600019",
    "drg_description": "Unrelated OR Procedures",
    "script_version": "1.0.29",
  }
}
```

```

    "logic_version": "0.2.1747.202510161025",
    "cost_weight": "0.00",
    "sub_acute_weight": "0.00",
    "chronic_weight": "0.00",
    "total_cost_weight": "0",
    "nbr": "8037060",
    "status_cd": "normal"
  }
}

```

Semua hasil grouping yang error pada idrg diawali dengan kode 36. Ketika hasil groupingnya error maka tombol final tidak boleh muncul sehingga tidak bisa lanjut koding inacbg.

Referensi kode error idrg dapat dilihat pada [daftar kode ungroupable and unrelated](#)

10. Finalisasi iDRG

Request Body:

```

{
  "metadata": {
    "method": "idrg_grouper_final"
  },
  "data": {
    "nomor_sep": "000R000TEST"
  }
}

```

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  }
}

```

Finalisasi iDRG hanya dapat dilakukan ketika berhasil melakukan grouping iDRG.

11. Re-edit coding iDRG

Request Body:

```

{
  "metadata": {
    "method": "idrg_grouper_reedit"
  },
  "data": {
    "nomor_sep": "000R000TEST"
  }
}

```

Response:

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

12. Import Coding iDRG ke INACBG

Request Body:

```
{
  "metadata": {
    "method": "idrg_to_inacbg_import"
  },
  "data": {
    "nomor_sep": "000R000TEST"
  }
}
```

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok",
    "method": "idrg_to_inacbg_import"
  },
  "data": {
    "diagnosa": {
      "string": "S71.0#S87.9#E11.9",
      "expanded": [
        {
          "code": "S71.0",
          "display": "Open wound of hip",
          "no": "1",
          "validcode": "1",
          "metadata": {
            "code": "200",
            "message": "Ok"
          }
        }
      ],
    },
    {
      "code": "S87.9",
      "display": "Crushing injury of unspecified parts of knee and leg (IM)",
      "no": "2",
      "validcode": "0",
      "metadata": {
        "code": "400",
        "error_no": "E2101",
        "message": "IM tidak berlaku"
      }
    }
  }
}
```



```

    }
  },
  {
    "code": "E11.9",
    "display": "Non-insulin-dependent diabetes mellitus without complications",
    "no": "3",
    "validcode": "1",
    "metadata": {
      "code": "200",
      "message": "Ok"
    }
  }
]
},
"procedure": {
  "string": "81.51#86.28#91.799",
  "expanded": [
    {
      "code": "81.51",
      "display": "Total hip replacement",
      "no": "1",
      "validcode": "1",
      "metadata": {
        "code": "200",
        "message": "Ok"
      }
    }
  ],
  {
    "code": "86.28",
    "display": "Nonexcisional debridement of wound, infection, or burn",
    "no": "2",
    "validcode": "1",
    "metadata": {
      "code": "200",
      "message": "Ok"
    }
  },
  {
    "code": "91.799",
    "display": "Other Microscopic examination of specimen from operative wound (IM)",
    "no": "3",
    "validcode": "0",
    "metadata": {
      "code": "400",
      "error_no": "E2101",
      "message": "IM tidak berlaku"
    }
  }
]
}
}
}

```

13. Set Diagnosa INACBG

Request Body:

```
{
  "metadata": {
    "method": "inacbg_diagnosa_set",
    "nomor_sep": "000R000TEST"
  },
  "data": {
    "diagnosa": "S71.0#S87.0#E11.9"
  }
}
```

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok",
    "method": "inacbg_diagnosa_set"
  },
  "data": {
    "string": "S71.0#S87.0#E11.9",
    "expanded": [
      {
        "code": "S71.0",
        "display": "Open wound of hip",
        "no": "1",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      },
      {
        "code": "S87.0",
        "display": "Crushing injury of knee",
        "no": "2",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      },
      {
        "code": "E11.9",
        "display": "Non-insulin-dependent diabetes mellitus without complications",
        "no": "3",
        "validcode": "1",
        "metadata": {
          "code": "200",
          "message": "Ok"
        }
      }
    ]
  }
}
```

```
}  
}
```

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.

14. Set Prosedur INACBG

Request Body:

```
{  
  "metadata": {  
    "method": "inacbg_procedure_set",  
    "nomor_sep": "000R000TEST"  
  },  
  "data": {  
    "procedure": "81.51#86.28#91.79"  
  }  
}
```

Response:

```
{  
  "metadata": {  
    "code": 200,  
    "message": "Ok",  
    "method": "inacbg_procedure_set"  
  },  
  "data": {  
    "string": "81.51#86.28#91.79",  
    "expanded": [  
      {  
        "code": "81.51",  
        "display": "Total hip replacement",  
        "no": "1",  
        "validcode": "1",  
        "metadata": {  
          "code": "200",  
          "message": "Ok"  
        }  
      },  
      {  
        "code": "86.28",  
        "display": "Nonexcisional debridement of wound, infection, or burn",  
        "no": "2",  
        "validcode": "1",  
        "metadata": {  
          "code": "200",  
          "message": "Ok"  
        }  
      }  
    ]  
  }  
}
```

```

        "code": "91.79",
        "display": "Microscopic examination of specimen from operative wound, Other microscopic
examination",
        "no": "3",
        "validcode": "1",
        "metadata": {
            "code": "200",
            "message": "Ok"
        }
    }
]
}
}

```

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.

15. Grouping INACBG Stage 1

Request Body:

```

{
  "metadata": {
    "method": "grouper",
    "stage": 1,
    "grouper": "inacbg"
  },
  "data": {
    "nomor_sep": "000R000TEST"
  }
}

```

Keterangan parameter:

stage : diisi "1" (satu)
grouper : diisi "inacbg"

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response_inacbg": {
    "cbg": {
      "code": "M-1-04-I",
      "description": "PROSEDUR PADA SENDI TUNGKAI BAWAH (RINGAN)"
    },
    "base_tariff": "31713400",
    "tariff": "31713400",
    "kelas": "kelas_3",
  }
}

```

```

    "inacbg_version": "5.10.5.202509291030",
    "status_cd": "normal"
  },
  "special_cmg_option": [
    {
      "code": "RR04Hip",
      "description": "Hip Implant",
      "type": "Special Prosthesis"
    },
    {
      "code": "RR04Knee",
      "description": "Knee Implant",
      "type": "Special Prosthesis"
    },
    {
      "code": "YY01",
      "description": "Hip Replacement / Knee Replacement",
      "type": "Special Procedure"
    }
  ]
}

```

Jika hasil grouping adalah error

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response_inacbg": {
    "cbg": {
      "code": "X-0-13-X",
      "description": "ERROR: MALE WITH GROUPING CRITERIA NOT MET"
    },
    "kelas": "kelas_3",
    "inacbg_version": "5.10.5.202510071053",
    "status_cd": "normal"
  }
}

```

Semua hasil grouping yang error pada inacbg diawali dengan kode X. Ketika hasil groupingnya error maka tombol grouping untuk stage 2 tidak boleh muncul sehingga tidak dapat melanjutkan grouping inacbg stage 2

16. Grouping INACBG 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:

Request Body:

```

{
  "metadata": {
    "method": "grouper",

```

```

    "stage": "2",
    "grouper": "inacbg"
  },
  "data": {
    "nomor_sep": "0001R0016120666662",
    "special_cmig": "RR04#YY01"
  }
}

```

Keterangan parameter:

stage : diisi "2" (dua)
special_cmig : diisi dengan code yang diperoleh dari grouping stage 1 pada segment "**special_cmig_option**". Untuk mengisi lebih dari satu pilihan spesial_cmig, code-nya dijoin dengan tanda #.

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response_inacbg": {
    "cbg": {
      "code": "M-1-04-I",
      "description": "PROSEDUR PADA SENDI TUNGKAI BAWAH (RINGAN)"
    },
    "base_tariff": "31713400",
    "tariff": "62812400",
    "special_cmig": [
      {
        "code": "YY-01-II",
        "description": "HIP REPLACEMENT / KNEE REPLACEMENT",
        "tariff": 13099000,
        "type": "Special Procedure"
      },
      {
        "code": "RR-04-III-Hip",
        "description": "HIP IMPLANT",
        "tariff": 18000000,
        "type": "Special Prosthesis"
      }
    ],
    "kelas": "kelas_3",
    "inacbg_version": "5.10.5.202509291030",
    "status_cd": "normal"
  },
  "special_cmig_option": [
    {
      "code": "RR04Hip",
      "description": "Hip Implant",
      "type": "Special Prosthesis"
    },
    {

```

```

    "code": "RR04Knee",
    "description": "Knee Implant",
    "type": "Special Prosthesis"
  },
  {
    "code": "YY01",
    "description": "Hip Replacement / Knee Replacement",
    "type": "Special Procedure"
  }
]
}

```

Jika dari hasil grouper stage 1 tidak muncul parameter **special_cmng_option**, maka tidak perlu melakukan grouper stage 2.

17. Finalisasi INACBG

Request Body:

```

{
  "metadata": {
    "method": "inacbg_grouper_final"
  },
  "data": {
    "nomor_sep": "000R000TEST"
  }
}

```

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  }
}

```

Finalisasi INACBG hanya dapat dilakukan ketika berhasil melakukan grouping INACBG.

18. Re-edit coding INACBG

Request Body:

```

{
  "metadata": {
    "method": "inacbg_grouper_reedit"
  },
  "data": {
    "nomor_sep": "000R000TEST"
  }
}

```

Response:

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

19. Finalisasi Klaim

Request Body:

```
{
  "metadata": {
    "method": "claim_final"
  },
  "data": {
    "nomor_sep": "000R000TEST",
    "coder_nik": "123123123123"
  }
}
```

Mandatory: coder_nik

20. Re-edit Klaim

Request Body:

```
{
  "metadata": {
    "method": "reedit_claim"
  },
  "data": {
    "nomor_sep": "000R000TEST"
  }
}
```

21. Kirim online klaim kolektif (per hari) ke data center

Request Body:

```
{
  "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"
      }
    ]
  }
}
```

22. Kirim online klaim individual ke data center

Request Body:

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

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "data": [
      {
        "nomor_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>

23. Get data detail per klaim

Request Body:

```

{
  "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": 3,
      "kode_tarif": "AP",
      "jenis_rawat": 1,
      "tgl_masuk": "22/08/2025",
      "tgl_pulang": "25/08/2025",
      "cara_masuk": "gp",

```

```

"tgl_lahir": "31/01/1992",
"berat_lahir": "0",
"gender": 2,
"discharge_status": 1,
"diagnosa": "082.9",
"procedure": "81.51",
"diagnosa_inagrouper": "S71.0",
"procedure_inagrouper": "81.51",
"adl_sub_acute": 0,
"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": 100000,
  "obat_kronis": 1000000,
  "obat_kemoterapi": 5000000,
  "alkes": 500000,
  "bmhp": 400000,
  "sewa_alat": 210000
},
"sistole": "110",
"diastole": "60",
"los": "4",
"icu_indikator": 0,
"icu_los": "0",
"ventilator_hour": "0",
"upgrade_class_ind": "0",
"upgrade_class_class": "",
"upgrade_class_los": "0",
"add_payment_pct": "0.0",
"add_payment_amt": "0",
"upgrade_class_payor": null,
"nama_pasien": "TEST PASIEN",
"nomor_rm": "R002188",
"umur_tahun": 33,
"umur_hari": "12257",
"tarif_poli_eks": "0",
"dializer_single_use": "1",
"nama_dokter": "DR. BUDI SANTOSO, SP.A",
"nomor_sep": "000R000TEST",
"nomor_kartu": "0001234999",
"payor_id": "3",
"payor_nm": "JKN",
"coder_nm": "INACBG",
"coder_nik": "00001",

```

```

"patient_id": "1",
"admission_id": "1",
"hospital_admission_id": "1",
"grouping_count": "5",
" grouper": {
  "response_inacbg": {
    "cbg": {
      "code": "X-0-00-X",
      "description": "ERROR: NO CBG ASSIGN"
    },
    "kelas": "kelas_3",
    "inacbg_version": "5.10.5.202510071053",
    "status_cd": "normal"
  },
  "response_idrg": {
    "mdc_number": "33",
    "mdc_description": "Injuries, Poisonings and Toxic Effects of Drugs",
    "drg_code": "3303110",
    "drg_description": "Other OR Procedures for Injuries w/ No CC",
    "script_version": "1.0.29",
    "logic_version": "0.2.1747.202510161025",
    "cost_weight": "1.34",
    "sub_acute_weight": "0.00",
    "chronic_weight": "0.00",
    "total_cost_weight": "1.34",
    "nbr": "8037060",
    "status_cd": "final"
  }
},
"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": "-",
"bpjs_klaim_status_nm": "-"
}
}
}

```

24. 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

Konfigurasi VClaim

Enable Integration	<input checked="" type="radio"/> <u>Enable</u>
Host :	<input type="text" value="new-api.bpjs-kesehatan.go.id"/>
Port :	<input type="text" value="8080"/>
Consumer ID :	<input type="text" value="1001"/>
Consumer Secret :	<input type="text" value="secret"/>
Service Name	<input type="text" value="new-vclaim-rest"/>

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"
  }
}
```

25. Hapus klaim

Request Body:

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

```
}
```

26. Cetak klaim

Request Body:

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

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "data": "7c7uNsP04uXsTpr9zCtiTrYdzMjmHxZIEjDobAoujnJvd07UWTB
eRr9wb8mtnd9+gnzForViUj6QtD9xVBTJFxz4N/DvR7IwT7RqdQ DsgF15NnnWqZb/fNUKXQDQ+Q+e+yR48eo8bPF ... dst"
}
```

Hasil dari method **claim_print** adalah file pdf yang ter-encode dengan base64 yang terdapat pada variable "data". Silakan decode terlebih dahulu untuk mendapatkan file pdf dalam bentuk binary untuk kemudian ditampilkan atau disimpan.

27. Pencarian diagnosa iDRG

Request Body:

```
{
  "metadata": {
    "method": "search_diagnosis_inagrouper"
  }, "data": {
    "keyword": "D63"
  }
}
```

Keterangan parameter:

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

Response:

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

```

},
"response": {
  "count": 4,
  "data": [
    {
      "description": "Anaemia in chronic diseases classified elsewhere",
      "code": "D63",
      "validcode": "0",
      "accpdx": "Y",
      "code_asterisk": "D63*",
      "asterisk": "1",
      "im": "0"
    },
    {
      "description": "Anaemia in neoplastic disease",
      "code": "D63.0",
      "validcode": "1",
      "accpdx": "N",
      "code_asterisk": "D63.0*",
      "asterisk": "1",
      "im": "0"
    },
    {
      "description": "Anaemia due to liver and endocrine disorders (IM)",
      "code": "D63.1",
      "validcode": "1",
      "accpdx": "N",
      "code_asterisk": "D63.1*",
      "asterisk": "1",
      "im": "1"
    },
    {
      "description": "Anaemia in other chronic diseases classified elsewhere",
      "code": "D63.8",
      "validcode": "1",
      "accpdx": "N",
      "code_asterisk": "D63.8*",
      "asterisk": "1",
      "im": "0"
    }
  ]
}
}

```

28. Pencarian prosedur iDRG

Request Body:

```

{
  "metadata": {
    "method": "search_procedures_inagrouper"
  }, "data": {
    "keyword": "990"
  }
}

```

Keterangan parameter:

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

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "count": 16,
    "data": [
      {
        "description": "Perioperative autologous transfusion of whole blood or blood components",
        "code": "99.00",
        "validcode": "1",
        "im": "0"
      },
      {
        "description": "Transfusion of blood and blood components",
        "code": "99.0",
        "validcode": "0",
        "im": "0"
      },
      {
        "description": "Exchange transfusion",
        "code": "99.01",
        "validcode": "1",
        "im": "0"
      },
      {
        "description": "Transfusion of previously collected autologous blood",
        "code": "99.02",
        "validcode": "1",
        "im": "0"
      },
      {
        "description": "Other transfusion of whole blood",
        "code": "99.03",
        "validcode": "1",
        "im": "0"
      },
      {
        "description": "Transfusion of packed cells",
        "code": "99.04",
        "validcode": "1",
        "im": "0"
      },
      {
        "description": "Transfusion of platelets",
        "code": "99.05",

```



```

        "validcode": "1",
        "im": "0"
    },
    {
        "description": "Transfusion of coagulation factors",
        "code": "99.06",
        "validcode": "1",
        "im": "0"
    },
    {
        "description": "Transfusion of other serum",
        "code": "99.07",
        "validcode": "1",
        "im": "0"
    },
    {
        "description": "Transfusion of blood expander",
        "code": "99.08",
        "validcode": "1",
        "im": "0"
    },
    {
        "description": "Transfusion of other substance",
        "code": "99.09",
        "validcode": "1",
        "im": "0"
    },
    {
        "description": "Insertion of non-drug-eluting peripheral (non-coronary) vessel
stent(s)",
        "code": "39.90",
        "validcode": "1",
        "im": "0"
    },
    {
        "description": "Uterine compression suture (B-Lynch) (IM)",
        "code": "69.90",
        "validcode": "1",
        "im": "1"
    },
    {
        "description": "Unspecified operation on bone injury, unspecified site",
        "code": "79.90",
        "validcode": "1",
        "im": "0"
    },
    {
        "description": "Microscopic examination of specimen from lower gastrointestinal tract
and of stool, Microscopic ISH examination (IM)",
        "code": "90.990",
        "validcode": "1",
        "im": "1"
    },
    {
        "description": "Microscopic examination of specimen from unspecified site, Microscopic

```

```

ISH examination (IM)",
    "code": "91.990",
    "validcode": "1",
    "im": "1"
  }
]
}
}

```

29. Pencarian diagnosa INACBG

Request Body:

```

{
  "metadata": {
    "method": "search_diagnosis"
  },
  "data": {
    "keyword": "D63"
  }
}

```

Keterangan parameter:

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

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "count": 2,
    "data": [
      [
        "Anaemia in other chronic diseases classified elsewhere",
        "D63.8"
      ],
      [
        "Anaemia in neoplastic disease",
        "D63.0"
      ]
    ]
  }
}

```

30. Pencarian prosedur INACBG

Request Body:

```

{

```

```
{
  "metadata": {
    "method": "search_procedures"
  }, "data": {
    "keyword": "990"
  }
}
```

Keterangan parameter:

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

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "count": 10,
    "data": [
      [
        "Transfusion of coagulation factors",
        "99.06"
      ],
      [
        "Transfusion of other serum",
        "99.07"
      ],
      [
        "Transfusion of blood expander",
        "99.08"
      ],
      [
        "Transfusion of other substance",
        "99.09"
      ],
      [
        "Transfusion of platelets",
        "99.05"
      ],
      [
        "Transfusion of packed cells",
        "99.04"
      ],
      [
        "Exchange transfusion",
        "99.01"
      ],
      [
        "Transfusion of previously collected autologous blood",
        "99.02"
      ],
      [

```

```

        "Other transfusion of whole blood",
        "99.03"
    ],
    [
        "Perioperative autologous transfusion of whole blood or blood components",
        "99.00"
    ]
]
}
}

```

31. Generate nomor pengajuan klaim

Method ini digunakan sebelum new_claim.

Request body:

```

{
  "metadata": {
    "method": "generate_claim_number"
  }
}

```

Response :

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "claim_number": "0000000ICC25100002"
  }
}

```

Request body yang salah:

```

{
  "metadata": {
    "method": "generate_claim_number"
  },
  "data": {
    "payor_id": "71"
  }
}

```

Keterangan parameter:

payor_id : Parameter ini sudah tidak diperlukan lagi untuk generate. Penentuan payor_id sudah digantikan pada method **set_claim_data** menjadi mandatory.

Nomor pengajuan klaim (claim_number) yang diperoleh dari method ini

digunakan untuk parameter **nomor_sep** di method "new_claim" (no. 1).

32. Validasi Nomor Register SITB

Validasi SITB mesti dilakukan ketika terdapat diagnosa TB pada INACBG, dengan kode antara lain: **A15.***, **A16.***, **A17.***, **A18.*** atau **A19.***. Berikut cara pemanggilan method

Request Body:

```
{
  "metadata": {
    "method": "sitb_validate"
  },
  "data": {
    "nomor_sep": "0001R0016120666662",
    "nomor_register_sitb": "00000001"
  }
}
```

Keterangan parameter:

nomor_register_sitb : diisi dengan nomor register dari SITB

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "status": "VALID",
    "detail": "Nomor register valid (dan masuk dalam data klaim)"
  },
  "validation": {
    "data": [
      {
        "id": "00000001",
        "nama": "NAMA PASIEN TB",
        "nik": "3579999999990005",
        "jenis_kelamin_id": "2"
      }
    ],
    "success": true
  }
}
```

atau jika error:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "status": "INVALID",
  }
}
```

```
    "detail": "Nomor register tidak ditemukan"
  }
}
```

33. Membatalkan validasi Nomor Register SITB:

Request Body:

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

Keterangan parameter:

nomor_register_sitb : diisi dengan nomor register dari SITB

Response:

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

V. DAFTAR KODE ERROR

No	Kode	Deksripsi
1	E2001	Method tidak ada
2	E2002	Klaim belum final
3	E2003	Nomor SEP terduplikasi
4	E2004	Nomor SEP tidak ditemukan
5	E2005	NIK Coder masih kosong
6	E2006	NIK Coder tidak ditemukan
7	E2007	Duplikasi nomor SEP
8	E2008	Nomor RM tidak ditemukan
9	E2009	Klaim sudah final
10	E2010	Nomor SEP baru sudah terpakai
11	E2011	Klaim tidak bisa diubah/edit
12	E2012	Tanggal Pulang mendahului Tanggal Masuk
13	E2013	Lama rawat intensif melebihi total lama rawat
14	E2014	Kode tarif invalid
15	E2015	Kode RS belum disetup
16	E2016	CBG Code invalid, tidak bisa final
17	E2017	Klaim belum di-grouping
18	E2018	Klaim masih belum final
19	E2019	Error: Tanggal invalid.
20	E2020	Response web service SEP kosong
21	E2021	Error : Gagal men-decode JSON - Maximum stack depth exceeded
22	E2022	Error : Gagal men-decode JSON - Underflow or the modes mismatch
23	E2023	Error : Gagal men-decode JSON - Unexpected control character found
24	E2024	Error : Gagal men-decode JSON - Syntax error, malformed JSON
25	E2025	Error : Gagal men-decode JSON - Malformed UTF-8 characters
26	E2026	Error : Gagal men-decode JSON - Unknown error
27	E2027	Rumah sakit belum terdaftar
28	E2028	Error : Jenis rawat invalid
29	E2029	Error : Koneksi gagal

No	Kode	Deksripsi
30	E2030	Error : Parameter tidak lengkap
31	E2031	Error : Key Mismatch
32	E2032	Error : Parameter kenaikan kelas tersebut tidak diperbolehkan
33	E2033	Error : Parameter payor_id tidak boleh kosong
34	E2034	Error : Nomor klaim tidak ditemukan
35	E2035	Error : Lama hari episode ruang rawat tidak sama dengan total lama rawat
36	E2036	Error : Tipe file tidak diterima
37	E2037	Error : Gagal upload
38	E2038	Error : Gagal hapus, klaim sudah diproses
39	E2039	Error : Gagal edit ulang, klaim sudah dikirim
40	E2040	Error : Gagal final. Belum ada berkas yang diunggah.
41	E2041	Error : Gagal final. Ada berkas yang masih gagal diunggah.
42	E2042	Error : Menyatakan covid19_cc_ind = 1 tanpa diagnosa sekunder.
43	E2043	Error : Nomor Klaim sudah terpakai.
44	E2044	Error : Gagal upload. Error ketika memindahkan berkas.
45	E2045	Error : Gagal upload. Ukuran file melebihi batas maksimal.
46	E2046	Error : Nilai parameter covid19_status_cd tidak berlaku.
47	E2047	Error: Gagal mendapatkan status klaim.
48	E2048	Error: Tanggal masuk tidak berlaku untuk Jaminan KIPi.
49	E2049	Error: Usia 7 hari ke atas tidak berlaku untuk Jaminan Bayi Baru Lahir.
50	E2050	Error: Tanggal masuk tidak berlaku untuk Jaminan Perpanjangan Masa Rawat.
51	E2050	Error: Tanggal masuk tidak berlaku untuk Jaminan Co-Insidense.
52	E2051	Error: Parameter payor_id kosong atau invalid.
53	E2052	Error: Parameter nomor_kartu_t invalid.
54	E2053	Error: Nomor klaim ibu invalid.
55	E2054	Error: Parameter bayi_lahir_status_cd invalid.
56	E2055	Error: Kode jenis ruangan pada parameter episodes invalid.
57	E2056	Error: Parameter akses_naas invalid.
58	E2057	Error: Nilai terapi_konvalesen pada non ranap atau non terkonfirmasi COVID-19.
59	E2058	Error: Parameter file_class invalid.
60	E2059	Error: Parameter covid19_no_sep invalid.

No	Kode	Deksripsi
61	E2060	Error: Diagnosa Primer untuk COVID-19 tidak sesuai ketentuan.
62	E2061	Error: Isolasi mandiri di RS pada rawat IGD.
63	E2062	Error: Lama rawat kelas upgrade lebih lama dari total lama rawat.
64	E2063	Error: Gagal final. Hasil INA Grouper tidak valid.
65	E2064	Error: upgrade_class_payor masih kosong atau tidak sesuai ketentuan.
66	E2065	Error: Kelas 3 tidak diperkenankan naik kelas.
67	E2066	Error: Gagal final. Pasien dengan TB belum ada validasi SITB.
68	E2099	Error tidak diketahui

VI. DAFTAR KODE UNGROUPABLE AND UNRELATED

Ungroupable dan Unrelated muncul ketika sistem grouper tidak dapat menemukan kecocokan antara kode diagnosis dan tindakan dalam sistem.

Penyebab umum:

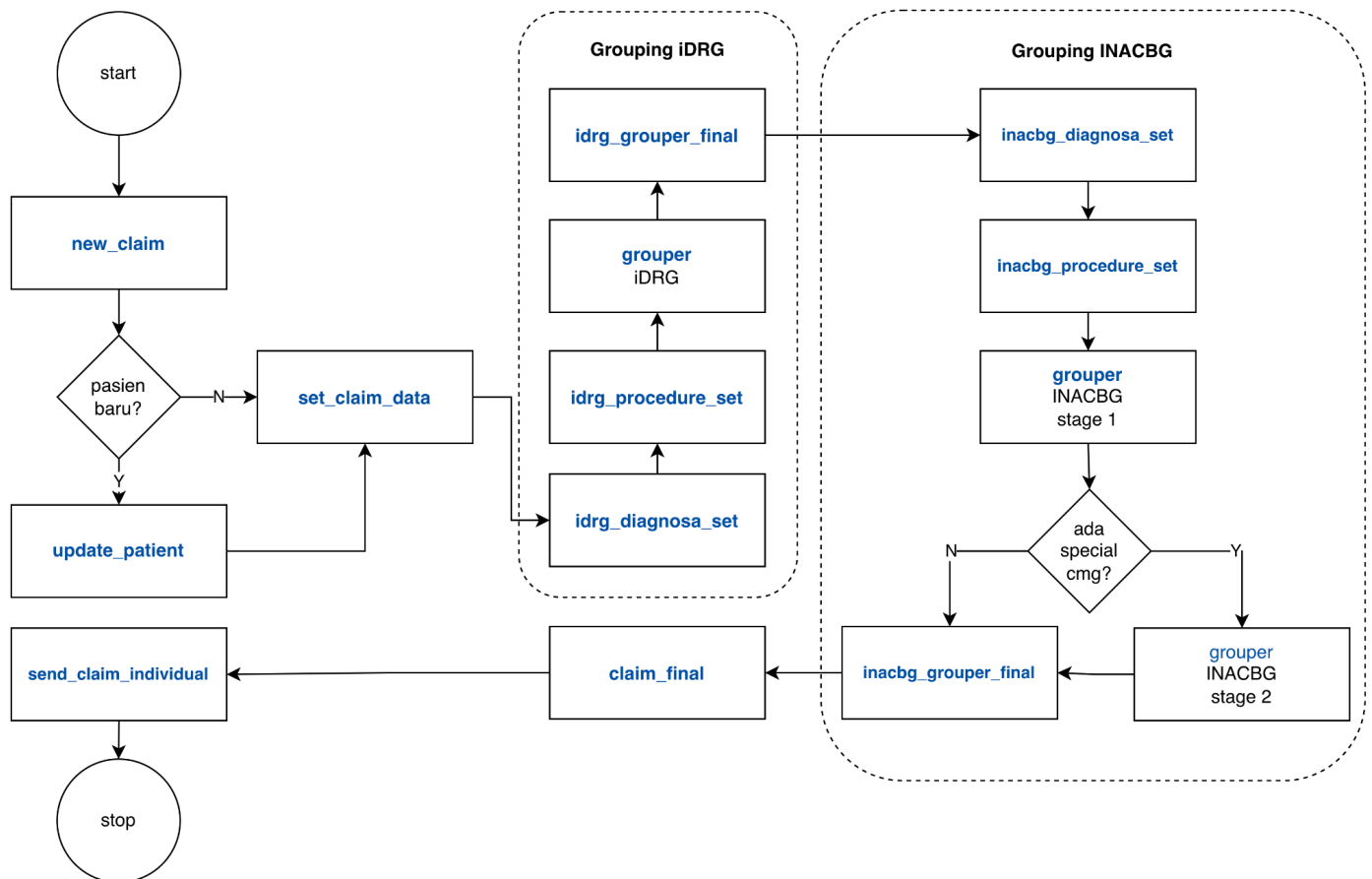
- Jenis kelamin tidak sesuai
- Usia tidak sesuai (mis. kasus perinatologi ≤28 hari)
- Tidak sesuai kaidah pengodean (mis. tidak input jumlah pembuluh darah untuk kasus pemasangan stent)
- Tidak sesuai pedoman iDRG (mis. tindakan rehabilitasi medis tanpa kode Z50.-)

No	Kode	Deskripsi
1	3611199	Primary Diagnosis code is not valid for Female or Male — Ketidaksesuaian jenis kelamin dengan diagnosis.
2	3611299	Diagnosis code is not valid for patient's age — Ketidaksesuaian usia pasien dengan diagnosis.
3	3600019	Unrelated OR Procedure — Ketidaksesuaian antara diagnosis dan tindakan.
4	3635929	Ungroupable: Missing Rehab. Consultation Proc — Tambahkan kode konsultasi rehabilitasi medis: • 89.01–89.09 (variasi konsultasi/rehabilitasi).
5	3612011 / 3612031	Missing code also for lens insertion proc — Tambahkan kode: • 13.70–13.72 (Insertion of pseudophakos).
6	3614129	Respiratory Drug Resistance TB w/o Drug Type — Tambahkan kode resistensi obat: • U82.20–U85.0.
7	3614229	Respiratory Drug Resistance TB w/o Episode Type — Tambahkan kode episode TB: • U84.31–U84.35.
8	3615029	Missing Code Also: Angiocardiology Proc — Tambahkan kode angiokardiografi: • 88.51–88.58.
9	3615129	Missing Code Also: Catheterization Proc — Tambahkan kode kateterisasi: • 37.21–37.23.

No	Kode	Deskripsi
10	3615229	Stent Insertion w/o Number of Stent or Vessels — Tambahkan kode jumlah stent: • 00.45–00.48.
11	3615329	Number of Stent w/o Insertion Proc — Tambahkan kode: • 00.55, 39.90 (Stent insertion).
12	3615429	Angioplasty w/o Number of Vessels — Tambahkan kode: • 00.40–00.43.
13	3615529	Missing code also (00.56 or 00.57) — Tambahkan kode: • 00.56–00.57 (Defibrillator implantation).
14	36001x9	Ungroupable: Radiotherapy Session w/o Radiotherapy Proc — Tambahkan kode radioterapi: • 92.20–92.27.
15	36002x9	Ungroupable: Radiotherapy w/o Z51.0 — Tambahkan diagnosis Z51.0 (Radiotherapy session).
16	36003x9	Ungroupable: Chemotherapy Proc w/o Z51.1 — Tambahkan diagnosis Z51.1 (Chemotherapy session).
17	3635129	Ungroupable: Rehabilitation without Procedures — Tambahkan kode tindakan rehabilitasi medis yang spesifik.

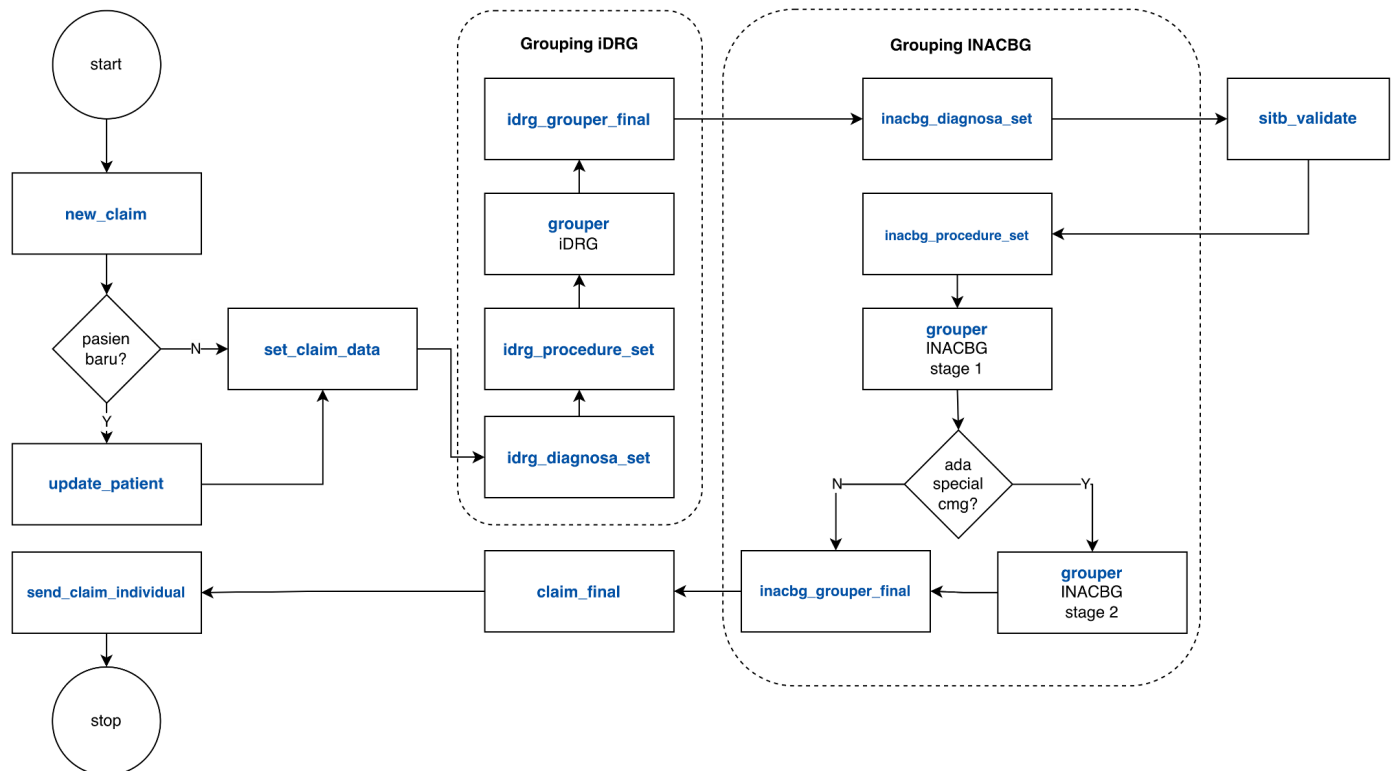
VII. ALUR DASAR INTEGRASI (BASIC INTEGRATION FLOW)

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.



IX. ALUR INTEGRASI DENGAN SITB

Berikut ini adalah alur sebagai acuan mengintegrasikan SIMRS dengan E-Klaim untuk klaim pembiayaan pasien JKN dengan TB (Tuberculosis). Tulisan yang berwarna biru adalah nama method.



X. CHANGELOG

20251023

- Perubahan Alur Integrasi
- Penambahan Daftar Kode Ungroupable And Unrelated
- Perubahan parameter method **set_claim_data**
- Perubahan parameter method **grouper**
- Penambahan method **idrg_procedure_set**
- Penambahan method **idrg_procedure_get**
- Penambahan method **idrg_diagnose_set**
- Penambahan method **idrg_diagnose_get**
- Penambahan method **idrg_grouper_final**
- Penambahan method **idrg_grouper_reedit**
- Penambahan method **idrg_to_inacbg_import**
- Penambahan method **inacbg_procedure_set**
- Penambahan method **inacbg_procedure_get**
- Penambahan method **inacbg_diagnose_set**
- Penambahan method **inacbg_diagnose_get**
- Penambahan method **inacbg_grouper_final**
- Penambahan method **inacbg_grouper_reedit**
- Drop method **pull_claim** (deprecated)

20240701

- Penambahan element **shk**
- Penambahan element **alteplase_ind**

20230106

- Koreksi method **sitb_validate** dan **sitb_invalidate**
- Bagan alur integrasi SITB

20221221

- Penambahan method **sitb_validate**
- Penambahan method **sitb_invalidate**

20220108

- Penambahan error code E2063

20220104

- Penambahan method **search_diagnosis_inagrouper**
- Penambahan method **search_procedures_inagrouper**
- Penambahan parameter **diagnosa_inagrouper** pada method **set_claim_data**
- Penambahan parameter **procedure_inagrouper** pada method **set_claim_data**

20210607

- Penambahan **payor_id** 72 s/d 75
- Parameter **payor_id** menjadi mandatory untuk method **set_claim_data**
- Penambahan jenis kartu identitas (**no_kartu_t**)
- Penambahan parameter **isoman_ind**
- Penambahan parameter **terapi_konvalesen**
- Penambahan parameter **akses_naot**

20200727

- Penambahan jenis kartu identitas (**no_kartu_t**)
- Penambahan parameter **covid19_rs_darurat_ind**

- Penambahan parameter covid19_co_insidense_ind
- Penambahan parameter covid19_penunjang_pengurang

20200326

- Penambahan Jaminan untuk pasien COVID-19
- Penambahan metode generate_claim_number untuk nomor pengajuan claim COVID-19 - Fix ketika error grouper, jenis rawat di hasil grouper selalu rawat jalan **20190116**
- Fix error set_claim_data untuk rawat jalan poli eksekutif
- Penambahan parameter tarif_poli_eks di method get_claim_data

20190114

- Penambahan error code E2032
- Perubahan aturan naik kelas dibatasi hanya 1 tingkat diatas.
- Penambahan variable obat_kronis dan obat_kemoterapi pada set_claim_data dan get_claim_data.

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 teregister 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.

XI. KATALOG FUNGSI ENKRIPSI / DEKRIPSI DALAM BEBERAPA BAHASA PEMROGRAMAN

PHP

```
// Encryption Function
function inacbg_encrypt($data, $key) {
    /// make binary representation 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
End Module
```

```
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 Deciper 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 cyper 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;  
}
```

Python

```
import base64
import hmac,hashlib
from Crypto import Random
from Crypto.Cipher import AES
BS = 16
pad = lambda s: s + (BS - len(s) % BS) * chr(BS - len(s) % BS)
unpad = lambda s : s[0:-ord(s[-1])]
def inacbg_encrypt( data, key ):
    key = hex2bin(key)
    data = pad(data)
    iv = Random.new().read( AES.block_size )
    cipher = AES.new( key, AES.MODE_CBC, iv )
    encrypted = cipher.encrypt(data)
    signature = inacbg_signature(encrypted, key)
    return base64.b64encode( signature + iv + encrypted )
def inacbg_decrypt( enc, key ):
    key = hex2bin(key)
    enc = base64.b64decode(enc)
    signature = enc[:10]
    iv = enc[10:26]
    cipher = AES.new(key, AES.MODE_CBC, iv )
    own_signature = inacbg_signature(enc[26:], key)
    if(list(signature)==list(own_signature)):
        return unpad(cipher.decrypt( enc[26:] ))
    else:
        return "SIGNATURE_NOT_MATCH"
def inacbg_signature(data, key):
    res = hmac.new(key, data, hashlib.sha256).digest()
    return res[:10]
def hex2bin( hexStr ):
    bytes = []
    hexStr = ''.join( hexStr.split(" ") )
    for i in range(0, len(hexStr), 2):
        bytes.append( chr( int (hexStr[i:i+2], 16 ) ) )
    return ''.join( bytes )
```

Python

```
import hmac
import OpenSSL
import hashlib
import binascii
from base64 import b64decode
from base64 import b64encode
from Crypto import Random
from Crypto.Cipher import AES
BLOCK_SIZE = 16 # Bytes
def mb_substr(s,start,length=None,encoding="utf8"):
    u_s = bytes(s)
    return (u_s[start:(start+length)] if length else u_s[start:])
def utf8_encode(t):
    return unicode(t).encode()
def hash_hmac(algo, data, key):
    digest = hmac.new(key, data, algo).digest()
    return digest
def chunk_split(data):
    LINELEN = 64
    chunk = lambda s: b'\n'.join(s[i:min(i+LINELEN, len(s))])
    for i in range(0, len(s), LINELEN))
    return chunk(data)
def preventOracleAttack(a, b):
    if len(a)!= len(b):
        return False
    result = 0
    for i in range(len(a)):
        if a[i] is not b[i]:
            result += 1
    return result == 0
class EKLAIM:
    """
    Penggunaan:
    c = EKLAIM('key').encrypt('data')
    m = EKLAIM('key').decrypt(encrypted_data)
    """
    def __init__(self, key):
        self.key = binascii.unhexlify(key)
    def encrypt(self, raw):
        padding_len = BLOCK_SIZE - (len(raw) % BLOCK_SIZE)
        if isinstance(raw, str):
            padded_plaintext = raw + (chr(padding_len) * padding_len)
        else:
            padded_plaintext = raw + (bytearray([padding_len] * padding_len))
        iv = Random.new().read(BLOCK_SIZE)
        cipher = AES.new(self.key, AES.MODE_CBC, iv)
        encrypted = cipher.encrypt(padded_plaintext)
        digest = hash_hmac(hashlib.sha256, encrypted, self.key)
        signature = mb_substr(digest, 0, 10)
        encoded = chunk_split(b64encode(signature + iv + encrypted))
        return encoded
    def decrypt(self, enc):
        enc = enc.replace('----BEGIN ENCRYPTED DATA----\r\n', '')
        .replace('----END ENCRYPTED DATA----\r\n', '')
        decoded = b64decode(enc)
```

```
signature = mb_substr(decoded, 0, 10)
iv = mb_substr(decoded, 10, BLOCK_SIZE)
cipher = AES.new(self.key, AES.MODE_CBC, iv)
encrypted = mb_substr(decoded, BLOCK_SIZE+10)
digest = hash_hmac(hashlib.sha256, encrypted, self.key)
calc_signature = mb_substr(digest, 0, 10)
if not preventOracleAttack(signature, calc_signature):
    return '{"error": "SIGNATURE_NOT_MATCH"}'
padded_plaintext = cipher.decrypt(encrypted)
if isinstance(padded_plaintext, str):
    padding_len = ord(padded_plaintext[-1])
else:
    padding_len = padded_plaintext[-1]
plaintext = padded_plaintext[:-padding_len]
return plaintext
```

DAFTAR PUSTAKA

[1]Kementerian Kesehatan Republik Indonesia. (2023). Peraturan Menteri Kesehatan Republik Indonesia Nomor 3 Tahun 2023 tentang Standar Tarif Pelayanan Kesehatan dalam Penyelenggaraan Program Jaminan Kesehatan. Jakarta: Kementerian Kesehatan RI.