

Web Service E-Klaim INA-CBG

Untuk Build 5.8.x

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:

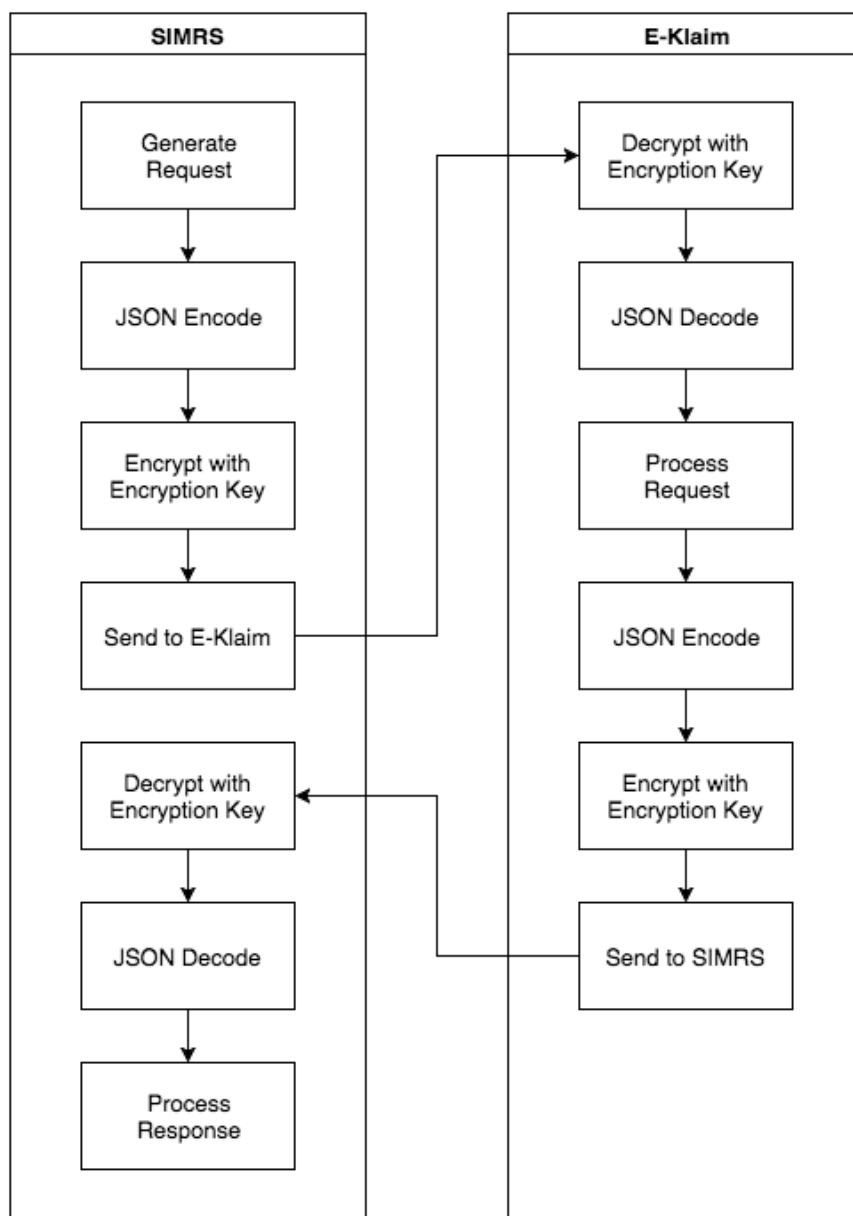
Konfigurasi	
Kode RS	3174282
Encryption Key	d26cbb6f64dade194e6681c4a076ecdbbf5628f10f4416a6d9afe15309f1fae

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 `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--

/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). 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.

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:

```
{
  "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="INACBG"/>
	<small>Title/Prefix</small>	<small>Full Name</small>
Employee ID	00001	
Alias	<input type="text"/> <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": "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": "2023-01-26 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",
    "diagnosa": "S71.0#A00.1",
    "procedure": "81.52#88.38#86.22",
    "diagnosa_inagrouper": "S71.0#A00.1",
    "procedure_inagrouper": "81.52#88.38#86.22+3#86.22",
    "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"
  },
  "pemulasaraan_jenazah": "1",
  "kantong_jenazah": "1",
  "peti_jenazah": "1",
  "plastik_erat": "1",
  "desinfektan_jenazah": "1",
  "mobil_jenazah": "0",
  "desinfektan_mobil_jenazah": "0",
  "covid19_status_cd": "1",
  "nomor_kartu_t": "nik",
  "episodes": "1;12#2;3#6;5",
  "covid19_cc_ind": "1",
  "covid19_rs_darurat_ind": "1",
  "covid19_co_insidence_ind": "1",
  "covid19_penunjang_pengurang": {
    "lab_asam_laktat" : "1",
    "lab_procalcitonin" : "1",
    "lab_crp" : "1",
    "lab_kultur" : "1",
    "lab_d_dimer" : "1",
    "lab_pt" : "1",
    "lab_aptt" : "1",
    "lab_waktu_pendarahan" : "1",
    "lab_anti_hiv" : "1",
    "lab_analisa_gas" : "1",
    "lab_albumin" : "1",
    "rad_thorax_ap_pa" : "0"
  },
  "terapi_konvalesen": "1000000",
  "akses_naat": "C",
  "isoman_ind": "0",
  "bayi_lahir_status_cd": 1,
  "dializer_single_use": "0",
  "kantong_darah": 1,
  "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"
        },
        {
            "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"
        }
    ]
},

"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** diisni dengan Code, seperti tersebut dibawah ini:

Payplan ID	3
Payment Plan Name	JKN
Code	JKN

Parameter menjadi versi 5.4.11.

payor_id ini mandatory sejak

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 method tersendiri untuk pencarian pada method nomor 16 dan 17 dibawah.

diagnosa_inagrouper : 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.

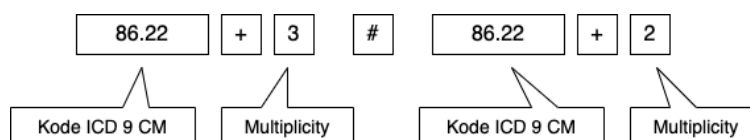
procedure_inagrouper : Kode procedure akan dicek terhadap versi ICD-9-IM (Indonesian Modification) yang

berlaku. Jika ada kode yang tidak terdaftar atau berlaku, maka kode tersebut tidak akan tersimpan.

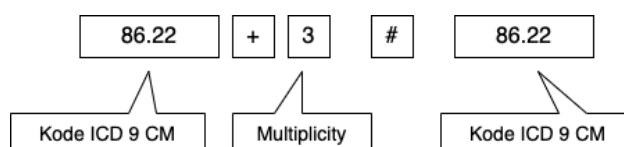
Untuk kode diagnosa dan procedure, disediakan method tersendiri untuk pencarian pada method nomor 21 dan 22 dibawah.

Prosedur pada INA Grouper 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 didalam 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.

sistole & diastole: Tekanan darah, dalam mmHg

dializer_single_use: Khusus untuk hemodialisa, diisi:

"1" = single use

"0" = multiple use

apgar score: Terdiri dari dua bagian yaitu menit pertama (menit_1) dan menit kelima (menit_5), yang masing-masing bagian memiliki element: apparence, pulse, grimace, activity dan respiration. Untuk setiap element diisi nilai antara 0,1, atau 2

persalinan:

usia_kehamilan: diisi angka dalam minggu

gravida: jumlah kehamilan, diisi angka

partus: jumlah kelahiran, diisi angka

abortus: jumlah keguguran, diisi angka

onset_kontraksi: diisi pilihan: **spontan, induksi, non_spontan_non_induksi**

delivery: berupa array object sebagai berikut:

delivery_sequence: urutan kelahiran (jika lahir lebih dari satu bayi dalam satu kali perawatan), diisi angka dimulai dari 1

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

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

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.

Untuk pasien COVID-19 yang meninggal dunia, disediakan parameter untuk mencatat pemakaian tambahan klaim untuk rangkaian pemulasaraan jenazah sebagai berikut: **pemulasaraan_jenazah, kantong_jenazah, peti_jenazah, plastik_erat, desinfektan_jenazah, mobil_jenazah, dan desinfektan_mobil_jenazah**.

Parameter tersebut diisi dengan nilai 1 jika ada pemakaian, 0 jika tidak ada pemakaian.

nomor_kartu_t : Untuk tambahan khusus pasien Jaminan COVID-19, parameter ini membedakan nilai yang tersebut didalam

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

covid19_status_cd : Untuk tambahan khusus pasien Jaminan COVID-19, parameter ini berisi status ODP/PDP/Terkonfirmasi. Yang valid diisi dengan nilai sebagai berikut:

- 1 = untuk ODP
- 2 = untuk PDP
- 3 = untuk pasien terkonfirmasi positif COVID-19
Terhitung mulai 15 Agustus 2020, maka parameter ini valid jika diisi dengan nilai:
- 4 = untuk suspek
- 5 = untuk probabel
- 3 = untuk pasien terkonfirmasi positif COVID-19
Terhitung mulai 1 Oktober 2021 parameter ini tidak diperlukan.

episodes : Untuk tambahan khusus pasien Jaminan COVID-19 yang rawat inap, parameter ini berisi lama rawat masing-masing episode ruangan perawatan yang dijalani oleh pasien selama rawat inap. Format pengisiannya dapat melihat contoh diatas sebagai berikut:

"episodes": "1;12#2;3#6;5"

Penjelasannya adalah setiap episode dibatasi (delimited by) tanda hash (#), kemudian masing-masing episode dinotasikan dengan jenis ruangan + titik koma + lama rawat.

Jenis ruangan didefinisikan sebagai berikut:

- 1 = ICU dengan ventilator
- 2 = ICU tanpa ventilator
- 3 = Isolasi tekanan negatif dengan ventilator
- 4 = Isolasi tekanan negatif tanpa ventilator
- 5 = Isolasi non tekanan negatif dengan ventilator
- 6 = Isolasi non tekanan negatif tanpa ventilator

Terhitung mulai tanggal masuk pasien 20 April 2021, maka definisi jenis ruangan sebagai berikut:

- 7 = ICU tekanan negatif dengan ventilator
- 8 = ICU tekanan negatif tanpa ventilator
- 9 = ICU tanpa tekanan negatif dengan ventilator
- 10 = ICU tanpa tekanan negatif tanpa ventilator
- 11 = Isolasi tekanan negatif
- 12 = Isolasi tanpa tekanan negatif

Sebagai contoh tersebut diatas, artinya adalah:

- episode pertama: ICU dengan ventilator selama 12 hari

- episode kedua : ICU tanpa ventilator selama 3 hari
- episode ketiga : Isolasi non tekanan negatif tanpa ventilator 5 hari

Perhatian: Bahwa jumlah total hari dalam episode ini harus sama dengan jumlah lama rawat berdasarkan tanggal masuk dan tanggal keluar. Jika tidak sama maka akan error.

Parameter ini tidak berlaku untuk rawat per 1 Oktober 2021.

covid19_cc_ind : Indikator kalau ada cc (comorbidity/complexity). Nilai diisi 1 kalau ada cc, 0 kalau tidak ada cc.

covid19_rs_darurat_ind : 1 = Indikator kalau pasien dirawat di lokasi rs darurat atau rs lapangan.

0 = Indikator kalau pasien tidak dirawat di lokasi rs darurat atau rs lapangan.

covid19_co_insidence_ind : Indikator kalau ada kasus co-insidence.

Per 1 Oktober 2021 hanya berlaku untuk JKN, payor_id = 3.

covid19_no_sep : Nomor klaim COVID-19 untuk kasus co-insidence pada klaim JKN.

covid19_penunjang_pengurang : Parameter ini berisi penanda jika ada pemeriksaan penunjang tersebut dibawah yang tidak diberikan selama masa perawatan.

Detail parameter sebagai berikut:

lab_asam_laktat	: Pemeriksaan Lab. Asam Laktat
lab_procalcitonin	: Pemeriksaan Lab. Procalcitonin
lab_crp	: Pemeriksaan Lab. CRP
lab_kultur	: Pemeriksaan Lab. Kultur MO (aerob) dengan resistansi
lab_d_dimer	: Pemeriksaan Lab. D Dimer
lab_pt	: Pemeriksaan Lab. PT
lab_aptt	: Pemeriksaan Lab. APTT
lab_waktu_pendarahan	: Pemeriksaan Lab. Waktu Pendarahan
lab_anti_hiv	: Pemeriksaan Lab. Anti HIV
lab_analisa_gas	: Pemeriksaan Lab. Analisa Gas
lab_albumin	: Pemeriksaan Lab. Albumin
rad_thorax_ap_pa	: Pemeriksaan Radiologi Thorax AP / PA

Parameter ini diisi dengan angka 0 = jika dilakukan atau angka 1 = jika tidak dilakukan.

Parameter ini tidak berlaku pada pasien masuk 1 Oktober 2021.

terapi_konvalesen : Parameter ini diisi dengan nilai tambahan untuk terapi plasma konvalesen. Sebelum 1 Oktober 2021 diisi dengan nilai rupiah, setelah 1 Oktober 2021 diisi dengan jumlah kantong.

akses_naat : Parameter ini diisi dengan nilai kode kategori akses NAAT yaitu A, B atau C. Tidak diperlukan per 1 Oktober 2021.

isoman_ind : Parameter ini diisi 0 jika bukan isolasi mandiri, atau diisi 1 jika isolasi mandiri.

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

Contoh update data prosedur:

```
{
  "metadata": {
    "method": "set_claim_data",
```

```

        "nomor_sep": "0001R0016120666662",
    },
    "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": "5.4.2.202004202041",
        "covid19_data": {
            "no_kartu_t": "nik",
            "covid19_status_cd": "1",
            "covid19_status_nm": "ODP",
            "episodes": [
                {
                    "episode_id": "1",
                    "episode_class_cd": "1",
                    "episode_class_nm": "ICU DENGAN VENTILATOR",
                    "los": "2",
                    "tariff": "33000000",
                    "order_no": "10"
                }
            ]
        },
        "pemulasaraan_jenazah": {
            "pemulasaraan": "0",
            "kantong": "0",
            "peti": "0",
            "plastik": "0",
            "desinfektan_jenazah": "0",

```

```

        "mobil": "0",
        "desinfektan_mobil": "0"
    },
    "cc_ind": "1",
    "top_up_rawat_gross": "33000000",
    "top_up_rawat_factor": "0.7",
    "top_up_rawat": "23100000",
    "top_up_jenazah": "0"
}
},
"response_inagrouper": {
    "mdc_number": "31",
    "mdc_description": "Multiple Significant Trauma",
    "drg_code": "3103110",
    "drg_description": "MST w/ Multiple Wound Debridement w/ No CC"
},
"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 special_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": "5.4.5.202007270459",
    "covid19_data": {
      "no_kartu_t": "nik",
      "covid19_status_cd": "1",
      "covid19_status_nm": "ODP",
      "episodes": [
        {
          "episode_id": "1",
          "episode_class_cd": "1",
          "episode_class_nm": "ICU DENGAN VENTILATOR",
          "los": "2",
          "tariff": "33000000",
          "order_no": "10"
        }
      ]
    },
    "pemulasaraan_jenazah": {
      "pemulasaraan": "0",

```

```

        "kantong": "0",
        "peti": "0",
        "plastik": "0",
        "desinfektan_jenazah": "0",
        "mobil": "0",
        "desinfektan_mobil": "0"
    },
    "cc_ind": "1",
    "top_up_rawat_gross": "33000000",
    "top_up_rawat_factor": "0.7",
    "top_up_rawat": "23100000",
    "top_up_jenazah": "0"
},
"response_inagrouper": {
    "mdc_number": "31",
    "mdc_description": "Multiple Significant Trauma",
    "drg_code": "3103110",
    "drg_description": "MST w/ Multiple Wound Debridement w/ No CC"
},
"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": [
      {
        "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_TIMEOUT"
  }
}

```

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 (method sudah ditutup)

```

{
  "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\tNOKARTU\tPAYOR_ID\tCODER_ID\tVERSI_INACBG\tVERSI_GROUPER\tC1\tC2\tC3\tC4\n3174282\tA\t3\tAP\t1\t01\07\2015\07\01\2016\01\01\1940\t0\t2\t2\tF20.6;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 (BERAT)\t9973500\t5027400\t3384500\t-\t0\t-\t0\t-\t0\t-\t0\t18385400\t250000\t191\t1\t2\t5\tNAMA TEST PASIEN\t123-45-67\t75\t27575\tDR. ERNA\t0301R00112140006067\t0000668873981\t3;JKN\t123456789\t5.0.0\t4\t1\t0\t23\t0a1f01ecc6f508dcc64491c9e8327839\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",
  "obat_kronis": "1000000",
  "obat_kemoterapi": "5000000",
  "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",
"tarif_poli_eks": "100000",
"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_cmg": [
  {
    "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": "5.4.2.202007270459",

"covid19_data": {
  "no_kartu_t": "nik",
  "covid19_status_cd": "1",
  "covid19_status_nm": "ODP",
  "episodes": [
    {
      "episode_id": "1",
      "episode_class_cd": "1",
      "episode_class_nm": "ICU DENGAN VENT..",
      "los": "2",
      "tariff": "33000000",
      "order_no": "10"
    }
  ],
  "pemulasaraan_jenazah": {
    "pemulasaraan": "0",
    "kantong": "0",
    "peti": "0",
    "plastik": "0",
    "desinfektan_jenazah": "0",
    "mobil": "0",
    "desinfektan_mobil": "0"
  },
  "cc_ind": "1",
  "top_up_rawat_gross": "33000000",
  "top_up_rawat_factor": "0.7",
  "top_up_rawat": "23100000",
  "top_up_jenazah": "0"
},
"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+gnzForViUj6QtD9xVBTJFxz4N/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"
    ]
]
}
}

```

18. Generate nomor pengajuan klaim:

Method ini digunakan sebelum new_claim.

```

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

```

Keterangan parameter:

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

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "claim_number": "0000005ICC200483B9"
  }
}

```

Nomor pengajuan klaim (claim_number) yang diperoleh dari method ini digunakan untuk parameter **nomor_sep** di method "new_claim" (no. 1).

19. Upload file:

Method ini digunakan untuk upload file pendukung klaim.

```

{
  "metadata": {
    "method": "file_upload",
    "nomor_sep": "0000005ICC20040001",
    "file_class": "resume_medis",
    "file_name": "resumse.pdf",
  },
  "data": "... base64_encoded binary file ..."
}

```

Keterangan parameter:

file_class : diisi dengan klasifikasi file yaitu: **resumse_medis**, **ruang_rawat**, **laboratorium**, **radiologi**, **penunjang_lain**, **resep_obat**, **tagihan**, **kartu_identitas**, **dokumen_kipi**, atau **lain_lain**.

Pertanggal masuk 1 Oktober 2021, **file_class** diisi antara lain: **resume_medis**, **kartu_identitas**, **bebas_biaya**, **surat_kematian**.

file_name : diisi dengan nama file.

data : diisi dengan file yang dikirim dalam format base64 string.

Response:

```
{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "file_id": "1",
    "file_name": "resume.pdf",
    "file_type": "application/pdf",
    "file_size": 130992,
    "file_class": "resumse_medis"
  }
}
```

Keterangan rsnponse:

file_id : id file, nilai urut mulai dari 1 untuk tiap nomor_sep.

file_type : mime types dari file yang di upload.

file_size : ukuran dalam bytes

upload_dc_bpjs : status upload ke data center BPJS. Jika 1 artinya sukses, jika 0 artinya gagal.

Jika terjadi kegagalan upload, response sebagai berikut:

```
{
  "metadata": {
    "code": 400,
    "message": "Gagal upload.",
    "error_no": "E2037",
    "response": {
      "file_id": "1",
      "file_type": "application/pdf",
      "file_name": "resumse.pdf",
      "file_size": "130992",
      "file_class": "resume_medis"
    }
  },
  "upload_dc_bpjs_response": {
    "metaData": {
      "code": "401",
      "message": "Berkas tidak dapat dikirim untuk no .. ini"
    },
    "response": null
  }
}
```

```

    }
}

```

Keterangan response:

upload_dc_bpjs_response : berisi response asli dari service upload BPJS. Untuk feedback pada user interface yang lebih bermakna dipersilakan masing-masing SIMRS untuk menterjemahkan dari atribut ini.

Catatan: Jika terjadi kegagalan upload dengan nomor error_no E2037, file tidak otomatis terhapus dari server E-Klaim. Dipersilakan untuk menghapusnya menggunakan file_id yang tersebut pada response.

20. Hapus file:

Method ini digunakan untuk menghapus file pendukung klaim yang telah diupload.

```

{
  "metadata": {
    "method": "file_delete"
  },
  "data": {
    "nomor_sep": "0000005ICC20040001",
    "file_id": "1"
  }
}

```

Keterangan parameter:

file_id : diisi dengan file_id yang sebelumnya telah di unggah.

Response:

```

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

```

Jika terjadi kegagalan maka response:

```

{
  "metadata": {
    "code": 400,
    "message": "Gagal hapus file, berkas sudah dihapus",
    "error_no": "E2039"
  }
}

```

atau

```

{
  "metadata": {
    "code": 400,
    "message": "Gagal hapus file, klaim sudah diproses",
    "error_no": "E2038",
    "delete_dc_bpjs_response": {
      "metaData": {
        "code": "401",
        "message": "Berkas tidak dapat dihapus"
      }
    }
  }
}

```

```

    },
    "response": null
  }
}
}

```

20. Daftar file:

Method ini digunakan untuk mendapatkan data file pendukung klaim yang telah diupload.

```

{
  "metadata": {
    "method": "file_get"
  },
  "data": {
    "nomor_sep": "0000005ICC20040018"
  }
}

```

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "count": 7,
    "data": [
      {
        "file_id": "7",
        "file_name": "resumse.pdf",
        "file_size": "1809",
        "file_type": "application/pdf",
        "file_class": "resume_medis",
        "upload_dc_bpjs": "1",
        "upload_dc_bpjs_response": {
          "metaData": {
            "code": "200",
            "message": "Sukses"
          },
          "response": {
            "keterangan": "Sukses"
          }
        }
      },
      {
        "file_id": "8",
        "file_name": "01_icu_ventilator.pdf",
        "file_size": "98506",
        "file_type": "application/pdf",
        "file_class": "ruang_rawat",
        "upload_dc_bpjs": "0",
        "upload_dc_bpjs_response": null
      },
      {
        "file_id": "12",
        "file_name": "lab5.pdf",

```

```

        "file_size": "303955",
        "file_type": "application/pdf",
        "file_class": "laboratorium",
        "upload_dc_bpjs": "0",
        "upload_dc_bpjs_response": null
    },
    {
        "file_id": "13",
        "file_name": "lab3.pdf",
        "file_size": "303955",
        "file_type": "application/pdf",
        "file_class": "laboratorium",
        "upload_dc_bpjs": "0",
        "upload_dc_bpjs_response": {
            "metaData": {
                "code": "400",
                "message": "Sambungan internet gagal"
            }
        }
    }
]
}
}

```

20. Check Status Klaim:

Method ini digunakan untuk mendapatkan status klaim COVID-19.

```

{
    "metadata": {
        "method": "retrieve_claim_status"
    },
    "data": {
        "nomor_sep": "0000005ICC20040018",
        "nomor_pengajuan": "0000005R47010601"
    }
}

```

Response:

```

{
    "metadata": {
        "code": 200,
        "message": "Ok"
    },
    "response": {
        "nomor_pengajuan": "0000005R47010601",
        "nomor_sep": "0000005ICC20040018",
        "bpjs_statusKlaim": "2",
        "bpjs_keterangan": "Revisi",
        "yankes_tahap_cd": "0",
        "yankes_tahap_msg": "bpjs",
        "yankes_status_cd": "0",
        "yankes_status_msg": "info tidak tersedia",
        "yankes_revisi_cd": "0",
        "yankes_revisi_msg": "tidak ada revisi",
        "yankes_bayar_cd": "0",
        "yankes_bayar_msg": "info tidak tersedia"
    }
}

```

```

    }
}

```

21. Pencarian diagnosa INA Grouper:

```

{
  "metadata": {
    "method": "search_diagnosis_inagrouper"
  },
  "data": {
    "keyword": "J44"
  }
}

```

Keterangan parameter:

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

Response:

```

{
  "metadata": {
    "code": 200,
    "message": "Ok"
  },
  "response": {
    "count": 8,
    "data": [
      {
        "description": "Other chronic obstructive pulmonary disease",
        "code": "J44",
        "validcode": "0",
        "accpdx": "Y",
        "code_asterisk": "J44",
        "asterisk": "0",
        "im": "0"
      },
      {
        "description": "Chronic obstructive pulmonary disease with acute lower respiratory infection",
        "code": "J44.0",
        "validcode": "1",
        "accpdx": "Y",
        "code_asterisk": "J44.0",
        "asterisk": "0",
        "im": "0"
      },
      {
        "description": "Chronic obstructive pulmonary disease with acute exacerbation, unspecified",
        "code": "J44.1",
        "validcode": "1",
        "accpdx": "Y",
        "code_asterisk": "J44.1",
        "asterisk": "0",
        "im": "0"
      },
      {

```

```

      "description": "Wheezing associated respiratory illness
(IM)",
      "code": "J44.2",
      "validcode": "1",
      "accpdx": "Y",
      "code_asterisk": "J44.2",
      "asterisk": "0",
      "im": "1"
    },
    {
      "description": "Obstruction syndrome post tuberculosis (IM)",
      "code": "J44.3",
      "validcode": "1",
      "accpdx": "Y",
      "code_asterisk": "J44.3",
      "asterisk": "0",
      "im": "1"
    },
    {
      "description": "Other specified chronic obstructive pulmonary
disease",
      "code": "J44.8",
      "validcode": "1",
      "accpdx": "Y",
      "code_asterisk": "J44.8",
      "asterisk": "0",
      "im": "0"
    },
    {
      "description": "Chronic obstructive pulmonary disease,
unspecified",
      "code": "J44.9",
      "validcode": "1",
      "accpdx": "Y",
      "code_asterisk": "J44.9",
      "asterisk": "0",
      "im": "0"
    },
    {
      "description": "Asthma COPD overlap syndrome (ACOS) (IM)",
      "code": "J44.90",
      "validcode": "1",
      "accpdx": "Y",
      "code_asterisk": "J44.90",
      "asterisk": "0",
      "im": "1"
    }
  ]
}

```

22. Pencarian prosedur INA Grouper:

```

{
  "metadata": {
    "method": "search_procedures_inagrouper"
  },

```

```

    "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": 3,
    "data": [
      {
        "description": "Cesarean section of unspecified type",
        "code": "74.9",
        "validcode": "0",
        "im": "0"
      },
      {
        "description": "Hysterotomy to terminate pregnancy",
        "code": "74.91",
        "validcode": "1",
        "im": "0"
      },
      {
        "description": "Other cesarean section of unspecified type",
        "code": "74.99",
        "validcode": "1",
        "im": "0"
      }
    ]
  }
}

```

23. Validasi Nomor Register SITB:

```

{
  "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"
    },
    "validation": {
        "data": [
            {
                "id": "00000001",
                "nama": "NAMA PASIEN TB",
                "nik": "357999999990005",
                "jenis_kelamin_id": "2"
            }
        ],
        "success": true
    }
}

```

atau jika error:

```

{
    "metadata": {
        "code": 200,
        "message": "Ok"
    },
    "response": {
        "status": "INVALID",
        "detail": "Nomor register tidak ditemukan"
    }
}

```

24. Membatalkan validasi Nomor Register SITB:

```

{
    "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"
    }
}

```

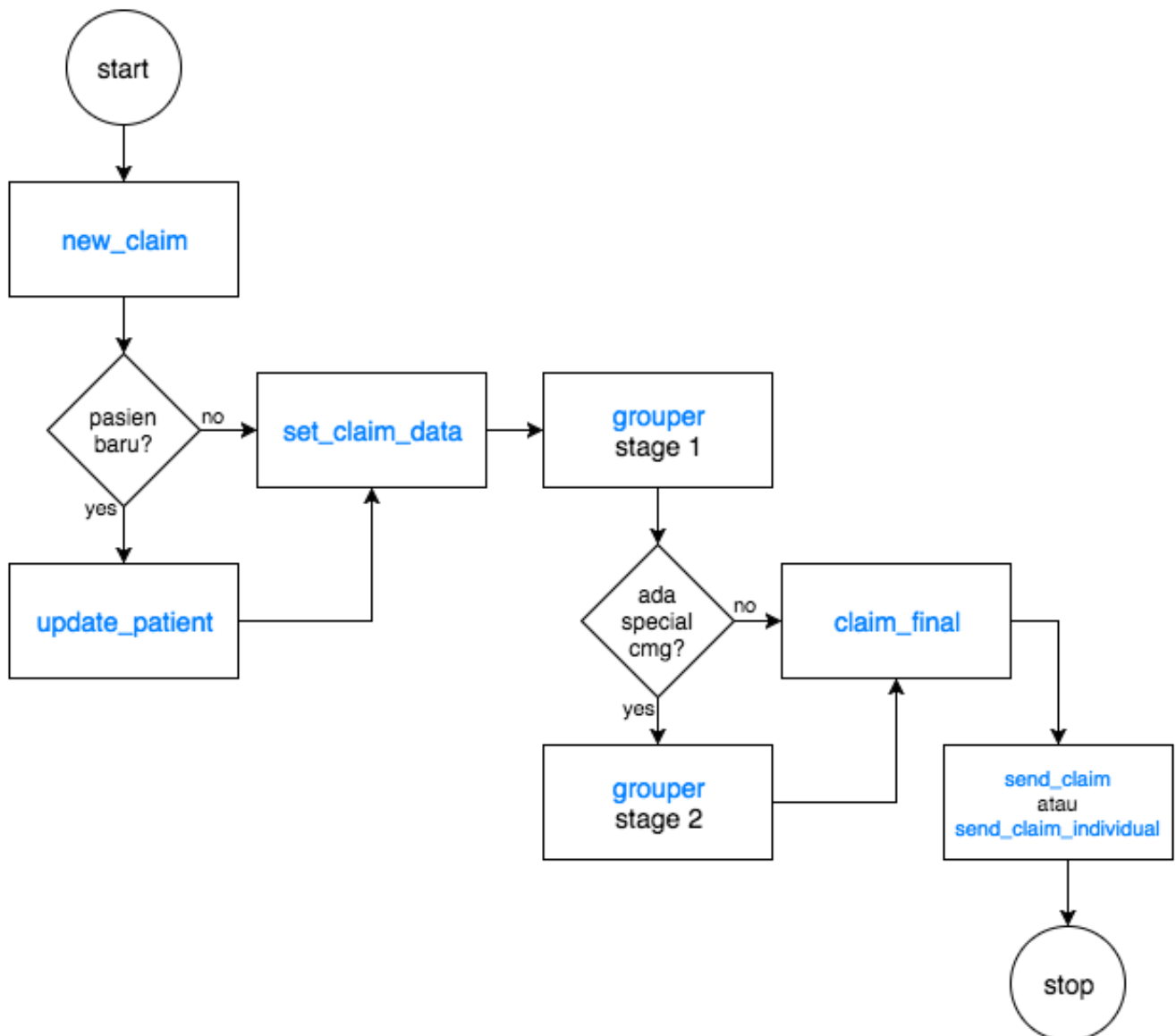
DAFTAR KODE ERROR

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	Error: 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	Error : Jenis rawat invalid
E2029	Error : Koneksi gagal
E2030	Error : Parameter tidak lengkap
E2031	Error : Key Mismatch
E2032	Error : Parameter kenaikan kelas tersebut tidak diperbolehkan
E2033	Error : Parameter payor_id tidak boleh kosong
E2034	Error : Nomor klaim tidak ditemukan
E2035	Error : Lama hari episode ruang rawat tidak sama dengan total lama rawat
E2036	Error : Tipe file tidak diterima
E2037	Error : Gagal upload
E2038	Error : Gagal hapus, klaim sudah diproses
E2039	Error : Gagal edit ulang, klaim sudah dikirim
E2040	Error : Gagal final. Belum ada berkas yang diunggah.
E2041	Error : Gagal final. Ada berkas yang masih gagal diunggah.
E2042	Error : Menyatakan covid19_cc_ind = 1 tanpa diagnosa sekunder.
E2043	Error : Nomor Klaim sudah terpakai.
E2044	Error : Gagal upload. Error ketika memindahkan berkas.

Kode	Deksripsi
E2045	Error : Gagal upload. Ukuran file melebihi batas maksimal.
E2046	Error : Nilai parameter covid19_status_cd tidak berlaku.
E2047	Error: Gagal mendapatkan status klaim.
E2048	Error: Tanggal masuk tidak berlaku untuk Jaminan KIPi.
E2049	Error: Usia 7 hari ke atas tidak berlaku untuk Jaminan Bayi Baru Lahir.
E2050	Error: Tanggal masuk tidak berlaku untuk Jaminan Perpanjangan Masa Rawat.
E2050	Error: Tanggal masuk tidak berlaku untuk Jaminan Co-Insidense.
E2051	Error: Parameter payor_id kosong atau invalid.
E2052	Error: Parameter nomor_kartu_t invalid.
E2053	Error: Nomor klaim ibu invalid.
E2054	Error: Parameter bayi_lahir_status_cd invalid.
E2055	Error: Kode jenis ruangan pada parameter episodes invalid.
E2056	Error: Parameter akses_naat invalid.
E2057	Error: Nilai terapi_konvalesen pada non ranap atau non terkonfirmasi COVID-19.
E2058	Error: Parameter file_class invalid.
E2059	Error: Parameter covid19_no_sep invalid.
E2060	Error: Diagnosa Primer untuk COVID-19 tidak sesuai ketentuan.
E2061	Error: Isolasi mandiri di RS pada rawat IGD.
E2062	Error: Lama rawat kelas upgrade lebih lama dari total lama rawat.
E2063	Error: Gagal final. Hasil INA Grouper tidak valid.
E2064	Error: upgrade_class_payor masih kosong atau tidak sesuai ketentuan.
E2065	Error: Kelas 3 tidak diperkenankan naik kelas.
E2066	Error: Gagal final. Pasien dengan TB belum ada validasi SITB.
E2099	Error tidak diketahui

ALUR DASAR INTEGRASI (BASIC INTEGRATION FLOW)

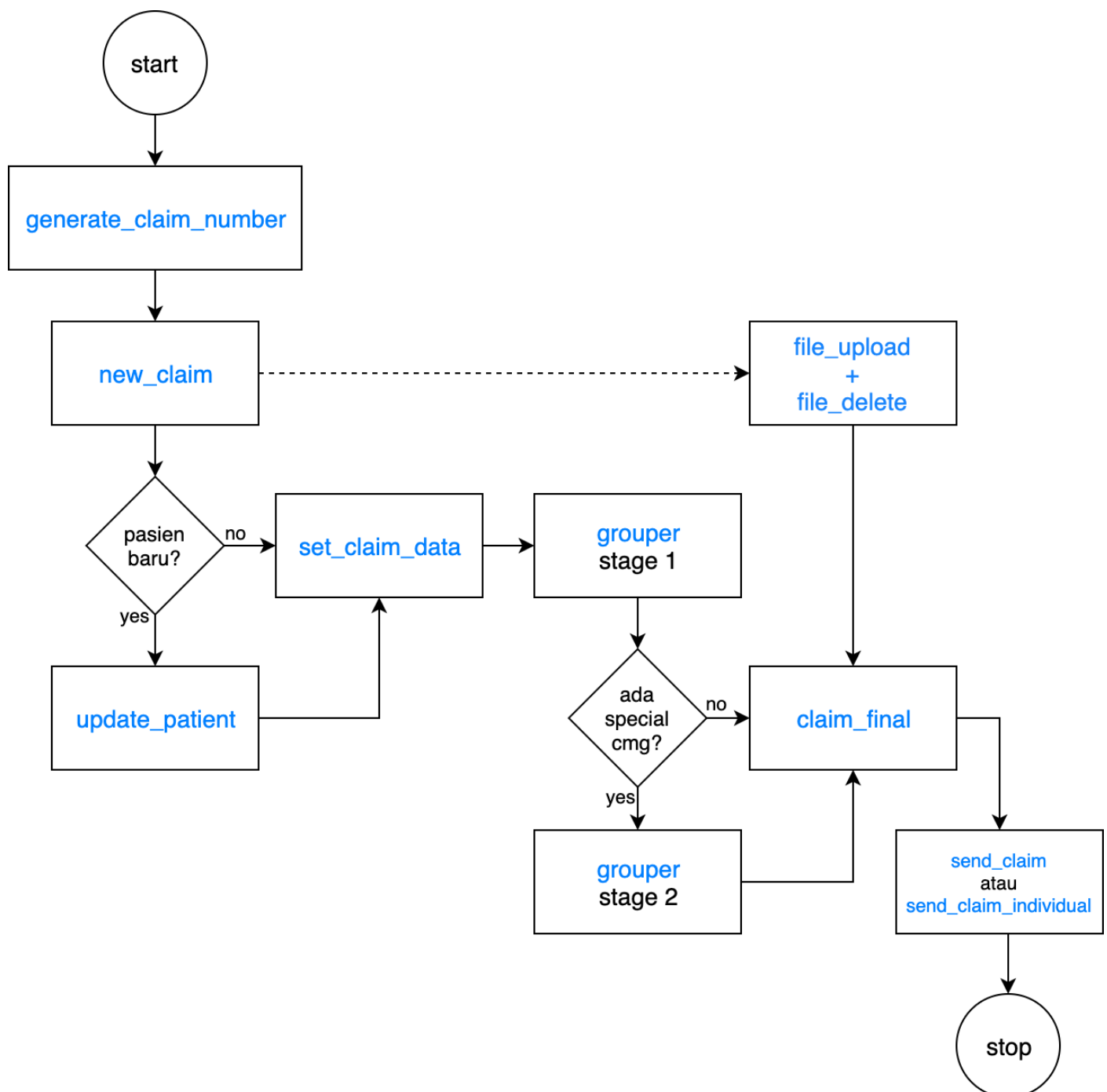
Berikut ini adalah alur dasar yang dapat dipakai sebagai acuan minimal untuk mengintegrasikan SIMRS dengan E-Klaim. Method-method yang digunakan adalah contoh minimal, method yang lain silakan ditambahkan atau digunakan sesuai kebutuhan. Tulisan yang berwarna biru adalah nama method.



=====

ALUR INTEGRASI UNTUK JAMINAN COVID-19

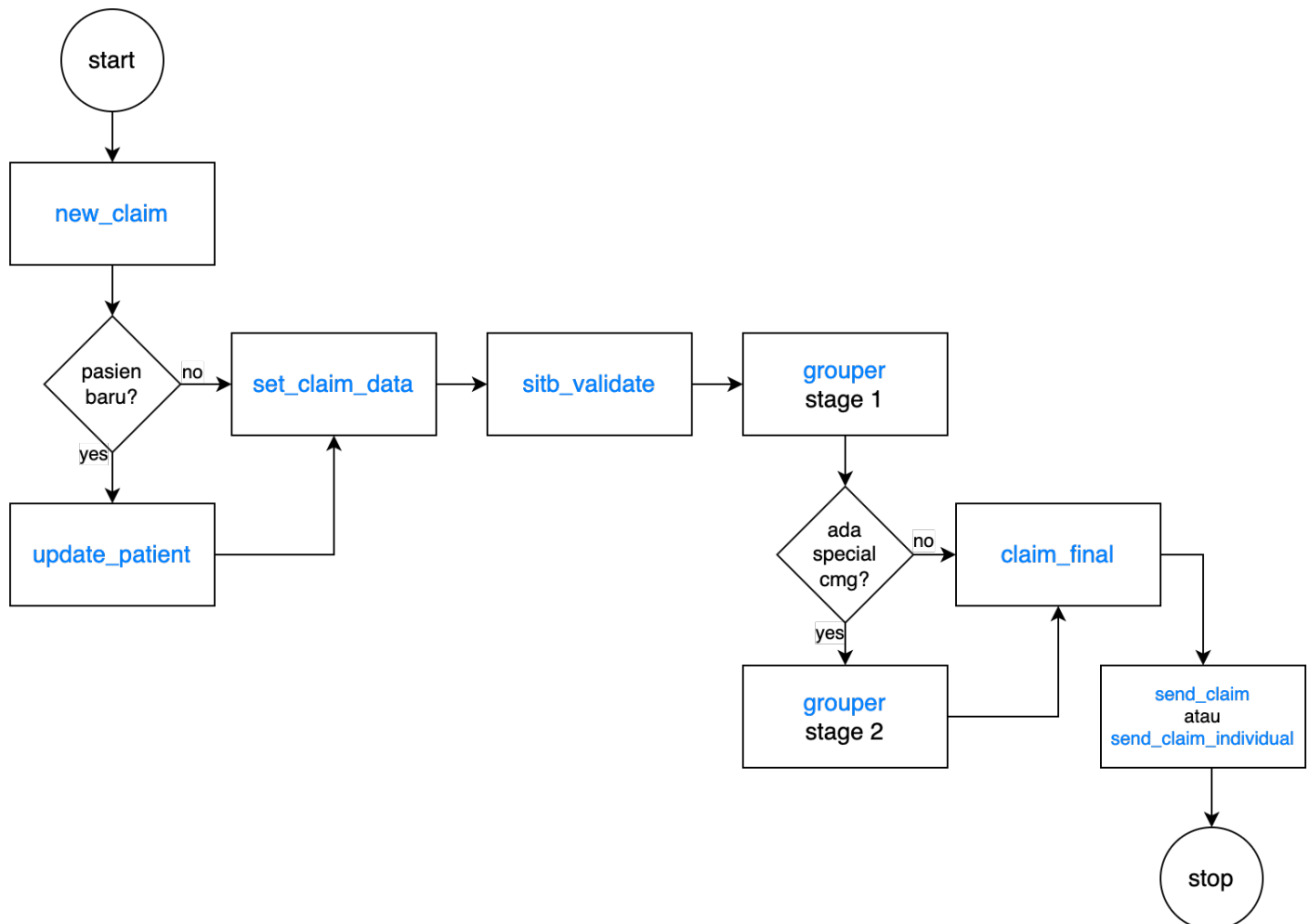
Berikut ini adalah alur sebagai acuan untuk mengintegrasikan SIMRS dengan E-Klaim dengan Jaminan COVID-19. Method-method yang khusus untuk Jaminan COVID-19 tidak berlaku untuk JKN. Tulisan yang berwarna biru adalah nama method.



Untuk method **file_upload** / **file_delete** boleh dipanggil kapan saja asalkan setelah method **new_claim** dan sebelum method **claim_final**.

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.



Changelog:

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_naat**

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 method **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 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 strencrypt, string key) {
    string encoded_str = strencrypt;
    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 url = '';

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;
}
```

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', '')
        enc = enc.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
```