BANK INDONESIA
BANK SENTRAL REPUBLIK INDONESIA

SNAP
terstandar, terintegrasi

Standar Nasional *Open* API Pembayaran (Standar Teknis dan Keamanan)

Versi 1.0.2
September 2024

PERNYATAAN

Standar Nasional *Open* API Pembayaran (Standar Teknis dan Keamanan) atau disebut "Standar Nasional *Open* API Pembayaran - Standar Teknis dan Keamanan" disusun oleh ASPI dengan mempertimbangkan dan memperhatikan masukan dari pelaku industri sistem pembayaran melalui wakilnya yang berpartisipasi dalam *sub-working group* nasional *Open* API Pembayaran yang dibentuk oleh ASPI bersama dengan Bank Indonesia.

Setiap pihak yang akan menggunakan Standar Nasional *Open* API Pembayaran - Standar Teknis dan Keamanan ini sepenuhnya bertanggung jawab untuk memastikan apakah kegiatannya atau pengembangannya memerlukan persetujuan dari pihak ketiga atau memerlukan konsultasi dengan konsultan yang berkompeten sebelum menerapkan Standar Nasional *Open* API Pembayaran - Standar Teknis dan Keamanan termasuk namun tidak terbatas pada penggunaan teknologi yang mungkin digunakan secara bersamaan.

Standar Nasional *Open* API Pembayaran - Standar Teknis dan Keamanan ini dapat diubah dan/atau disesuaikan sewaktu-waktu, bila diperlukan dan setiap perubahannya dituangkan dalam dokumen perubahan Standar Nasional *Open* API Pembayaran - Standar Teknis dan Keamanan terkini.

LEMBAR VERSI DOKUMEN

Versi	Tanggal	Penyusun	Keterangan
v.0.8.5	13-08-2021	ASPI	Disusun dengan dukungan perwakilan industri sistem pembayaran-Sub *Working Group* Nasional *Open* API Pembayaran, yang terdiri dari: - PT Bank Central Asia Tbk - PT Bank Mandiri Tbk - PT Bank Nationalnobu Tbk - PT Bank Negara Indonesia Tbk - PT Bank Rakyat Indonesi Tbk - PT Bukalapak.com Tbk - PT Dompet Anak Bangsa - PT Espay Debit Indonesia Koe - PT Fintek Karya Nusantara - PT Midtrans - PT Mitra Transaksi Indonesia - PT Multi Adiprakarsa Manunggal - PT Nusa Satu Inti Artha - PT Shopee International Indonesia - PT Tokopedia - PT Visionet Internasional
v.1.0	16-08-2021	ASPI	Berdasarkan dokumen versi 0.8.5 dari ASPI yang telah di-*review* oleh Bank Indonesia.
V.1.0.1	24-11-2021	Bank Indonesia dan ASPI	Berdasarkan dokumen versi 1.0.
V.1.0.2	September 2024	ASPI	Hapus kata "Private Key" pada: a. 1.7.1 Metode Penerimaan API Public Key b. 1.7.2 Metode Pengiriman API Piblic Key Pada API Access Token B2B, Ubah field sesuai Camelcase dari "granttype" menjadi "grantType" Pada 2.1.6. Komponen Struktur Format Header – Transaction (B2B dan B2B2C), Ubah Description pada field X-EXTERNAL- ID dari "Numeric String" menjadi "Alphanumeric"

DAFTAR ISI

1. BAGIAN I : PENGELOLAAN AKSES API

Dalam penyelenggaraan *Open* API Pembayaran berbasis SNAP diperlukan pedoman untuk pemberian akses dan *credential* dari Penyedia Layanan kepada Pengguna Layanan. Hal ini mempertimbangkan bahwa *Open* API Pembayaran berbasis SNAP yang disediakan oleh Penyedia Layanan, tidak dapat diakses secara langsung oleh publik. Penyedia Layanan mensyaratkan proses administrasi sebelum calon Pengguna Layanan dapat mengakses *Open* API Pembayaran berbasis SNAP yang diselenggarakan oleh Penyedia Layanan.

Pada tahap proses administrasi, Penyedia Layanan baik Penyelenggara Jasa Pembayaran (PJP) Bank maupun PJP Lembaga Selain Bank, memerlukan sejumlah data guna mengetahui identitas dari calon Pengguna Layanan sebagai dasar agar dapat memberikan *credential* dan akses.

1.1. BENTUK PENGAJUAN API

Dalam pengajuan kerjasama *Open* API Pembayaran berbasis SNAP, calon Pengguna Layanan disyaratkan untuk mengisi formulir yang berisi informasi calon Pengguna Layanan dan layanan API yang akan digunakan. Berdasarkan informasi tersebut, Penyedia Layanan akan melakukan pemrosesan. Setelah pengajuan kerjasama disetujui, Penyedia Layanan akan mendaftarkan Pengguna Layanan sebagai pihak yang bekerjasama.

Formulir pengajuan kerjasama *Open* API Pembayaran berbasis SNAP dapat berbentuk formulir tertulis (*physical registration*) maupun formulir daring (*online registration*).

1.1.1. Formulir Tertulis (*Physical Registration*)

Dalam hal Penyedia Layanan hanya menyediakan formulir pengajuan kerjasama *Open* API Pembayaran berbasis SNAP berbentuk formulir tertulis, maka calon Pengguna Layanan mengisi dan melengkapi formulir sebagaimana dimaksud. Formulir yang telah dilengkapi oleh calon Pengguna Layanan, akan diproses oleh Penyedia Layanan dan diinput secara manual pada sistem Penyedia Layanan.

1.1.2. Formulir Daring (*Online Registration*)

Dalam hal Penyedia Layanan hanya menyediakan formulir pengajuan kerjasama *Open* API Pembayaran berbasis SNAP berbentuk formulir daring, maka calon Pengguna Layanan mengisi dan melengkapi formulir pengajuan kerjasama secara daring baik melalui *website* atau aplikasi yang disediakan oleh Penyedia Layanan.

1.2. PERSETUJUAN TERHADAP KETENTUAN KERJASAMA LAYANAN PEMBAYARAN BERBASIS API

Dalam proses pengajuan kerjasama *Open* API Pembayaran berbasis SNAP, terdapat persyaratan kerjasama yang disepakati oleh calon Pengguna Layanan dan Penyedia Layanan. Hal ini untuk mencegah apabila dalam pelaksanaan atau penggunaan *Open* API Pembayaran berbasis SNAP terjadi kesalahpahaman atau kelalaian di antara kedua belah pihak. Calon Pengguna Layanan memberikan persetujuan terhadap persyaratan kerjasama tersebut dengan menandatangani pernyataan di atas materai agar pernyataan tersebut berkekuatan hukum. Pernyataan yang telah ditandatangani oleh calon Pengguna Layanan disampaikan kepada Penyedia Layanan sebagai bukti bahwa calon Pengguna Layanan telah setuju dengan persyaratan yang ditetapkan oleh Penyedia Layanan.

1.3. JENIS PENANGGUNG JAWAB API

Dalam setiap kerjasama *Open* API Pembayaran berbasis SNAP ditunjuk penanggung jawab sebagai pihak berwenang yang akan mewakili pihak Pengguna Layanan yang akan bekerjasama. Penunjukkan penanggung jawab bertujuan untuk mempermudah komunikasi antar pihak Penyedia Layanan dengan pihak Pengguna Layanan seperti untuk mempermudah proses diskusi atau penyampaian masalah.

Berikut merupakan jenis-jenis penanggung jawab yang diminta oleh Penyedia Layanan kepada calon Pengguna Layanan pada saat proses pengajuan kerja sama:

1. *Project Manager*
2. Penanggung jawab API Public Key

3. Penanggung jawab API Private Key
4. Penanggung jawab API *password key* dan API Private Key
5. Penanggung jawab transaksi API atau pertukaran data
6. Penanggung jawab integrasi API
7. Penanggung jawab administratif dan kerahasiaan dokumen
8. Penanggung jawab indikasi *fraud*

Penyedia Layanan menetapkan jumlah minimum dan maksimum untuk masing-masing penanggung jawab API yaitu minimum 1 (satu) orang dan maksimum 2 (dua) orang.

1.4. DATA ADMINISTRASI PENDAFTARAN BAGI CALON PENGGUNA LAYANAN

Data-data administrasi yang disyaratkan untuk pendaftaran kerjasama *Open* API Pembayaran berbasis SNAP terdiri dari informasi Pengguna Layanan, informasi legalitas, dan informasi pejabat berwenang. Data administrasi sebagaimana dimaksud adalah sebagai berikut, namun tidak terbatas pada:

1. Informasi calon Pengguna Layanan
    a. Nama perusahaan
    b. Nama dagang / merek (jika ada)
    c. Klasifikasi bisnis (deskripsi dan industri)*
        1) Jenis usaha / kategori usaha
        2) Kriteria usaha*
        3) MCC (*Merchant Category Code*)
    d. Alamat perusahaan
    e. Alamat korespondensi termasuk email
    f. Alamat usaha / operasional
    g. *Website**
    h. Logo*
    i. *National Merchant* ID
    j. Nomor Pokok Wajib Pajak (NPWP) usaha
    k. Nomor Surat Izin Usaha Perdagangan (SIUP)
    l. Informasi rekening (Nama dan No. Rekening, Nama Bank)
    m. Informasi tambahan*

1) Kesepakatan Komersial (sesuai *product inventory*)*
2) Informasi fitur (sesuai *product inventory*)*
3) Informasi konfigurasi merchant (nama dan *email merchant*)*

* Opsional

2. Informasi legalitas
   a. Salinan dokumen dasar pendirian badan termasuk dengan perubahannya (Anggaran Dasar, Akta Pendirian)
   b. Salinan dokumen pengesahan dan perijinan yang masih berlaku dari instansi terkait
   c. Dokumen perjanjian kerja sama
   d. Asli surat penunjukan dari perusahaan sebagai Pengguna Layanan
   e. Salinan NPWP / bukti potong pajak
   f. Salinan SIUP
   g. Salinan Nomor Induk Berusaha (NIB)**
   h. Salinan surat keterangan domisili usaha
   i. Salinan Tanda Daftar Perusahaan (TDP)
   j. Salinan surat pengukuhan Pengusaha Kena Pajak
   k. Salinan rekening koran atau buku tabungan
   l. Asli surat kuasa pendebetan rekening
   m. Nama pemilik usaha (salinan identitas diri terakhir)
   n. Nama penanggung jawab usaha dan pengurus (salinan identitas diri terakhir)
   o. Foto lokasi perusahaan*
   p. Dokumentasi hasil pengujian API di *Developer Site* SNAP

   * Opsional

   ** NIB menggantikan copy dokumen SIUP dan TDP

3. Informasi penanggung jawab usaha PIC atau pejabat berwenang dalam kontrak:
   a. Nama pemilik usaha / pejabat berwenang
   b. Nomor kontak PIC atau pejabat berwenang

c. Salinan identitas diri terakhir yang masih berlaku dari para pengurus atau pejabat berwenang yang bertindak untuk dan atas nama pemohon

1.5. DATA-DATA TEKNIS PENGAJUAN API

Selain data-data terkait informasi calon Pengguna Layanan, Penyedia Layanan juga meminta informasi data teknis untuk digunakan dalam melakukan konfigurasi koneksi antara kedua belah pihak. Data-data ini diperlukan agar komunikasi antar kedua pihak dapat berlangsung.

Data-data yang diperlukan oleh pihak Penyedia Layanan pada saat pengajuan kerjasama Layanan Pembayaran Berbasis API adalah sebagai berikut:

1. Fitur API yang ingin dipakai
2. Informasi apakah Pengguna Layanan akan menggunakan jalur koneksi yang sudah tersedia saat ini antara Pengguna Layanan dan Penyedia Layanan
3. Jenis koneksi yang akan digunakan (internet/*Multi Protocol Label Switching*)
4. Informasi *whitelist* IP
5. Informasi pemakaian *switcher* sebagai perantara koneksi
6. *Callback* URL (alamat domain aplikasi Pengguna Layanan)
7. *Public key for digital signature (for validate X-Signature in Access Token)*
8. *Subscription Package* (opsional)
9. Limit transaksi dan dokumen *underlying*
10. Jam operasional penggunaan API

1.6. PERUBAHAN DATA-DATA ADMINISTRASI DAN TEKNIS API

Mempertimbangkan bahwa hingga saat ini belum ada Penyedia Layanan yang telah memiliki API untuk layanan perubahan data Pengguna Layanan baik data administrasi maupun teknis, maka perubahan data dilakukan dengan cara mengisi dan mengirimkan formulir tertulis sebagaimana pada saat pengajuan kerjasama atau dengan mengubah sebagian data pada *website*

atau aplikasi yang disediakan oleh Penyedia Layanan. Dalam hal, data yang dapat diubah melalui aplikasi yang disediakan Penyedia Layanan terbatas, maka apabila dibutuhkan perubahan data lainnya Pengguna Layanan harus menghubungi pihak Penyedia Layanan untuk perubahan data yang tidak tersedia pada *website* atau aplikasi sebagaimana dimaksud. Pihak Penyedia Layanan membantu Pengguna Layanan untuk melakukan perubahan data berdasarkan informasi dari Pengguna Layanan.

1.7. PENGIRIMAN *KEY*

Pada kerjasama *Open* API Pembayaran berbasis SNAP, diperlukan adanya penggunaan *key* yaitu API Public Key antara Pengguna Layanan dengan Penyedia Layanan. Penggunaan *key* bertujuan untuk menjaga keamanan data transaksi yang dikirimkan dan untuk memastikan bahwa pengirim data adalah benar berasal dari pihak yang dikenal oleh Penyedia Layanan.
Proses pengiriman *key* kepada Pengguna Layanan perlu menjaga faktor kerahasiaan agar tidak disalahgunakan oleh pihak yang tidak berwenang.
Key untuk kerjasama Layanan Pembayaran Berbasis API dikirimkan melalui media *email* atau aplikasi yang disediakan oleh Penyedia Layanan.

1.7.1. Metode Penerimaan API Public Key

Penerimaan API Public Key dilakukan dengan pilihan metode sebagai berikut:

a. API Public Key diterima oleh PIC; atau
b. API Public Key diterima dan diproses oleh sistem aplikasi.

Pilihan mekanisme penerimaan tersebut dimaksudkan untuk meningkatkan keamanan terhadap API Public Key agar tidak dengan mudah disalahgunakan atau dibocorkan oleh pihak tidak berwenang.

1.7.2. Metode pengiriman API Public Key

Selain menentukan metode penerimaan API Public Key, peningkatan pengamanan pada pengiriman API Public Key dilakukan melalui cara penggunaan enkripsi atau *password* untuk melindungi keamanan data. API Public Key dienkripsi atau dilindungi menggunakan *password* oleh Penyedia Layanan sebagai pihak pengirim. Selanjutnya, Pengguna Layanan sebagai

pihak penerima perlu melakukan dekripsi atau menggunakan *password* yang diketahuinya untuk melihat atau membuka data API Public Key.

Pengiriman API Public Key dilakukan dengan pilihan metode sebagai berikut:

1. API Public Key dikirimkan melalui *email* dalam bentuk *zip file* yang diberi *password*;
2. API Public Key dikirimkan melalui *email* dalam bentuk *file* terenkripsi menggunakan *PGP Encryption*;
3. API Public Key dikirimkan melalui *email* dalam bentuk *file* terenkripsi menggunakan *OpenSSL* dan *public key partner*; atau
4. API Public Key dikirimkan secara *online* (melalui aplikasi).

Mekanisme pengiriman API Public Key sebagaimana dimaksud dilakukan dengan pilihan metode sebagai berikut:

1. Pengiriman dipisahkan dalam tiga *email* berbeda, yaitu:
   a. email 1 : Zip file berisi API Public Key
   b. email 3 : password zip untuk membuka *file* lampiran zip pada email 1
2. Pengiriman dipisahkan dalam dua *email* berbeda, yaitu:
   a. email 1 : Zip file berisi API Public Key
   b. email 2 : password zip untuk membuka *file* lampiran zip pada email 1
3. Pengiriman secara *online* (melalui aplikasi)

1.8. PENGELOLAAN *KEY*

Pengelolaan *key* baik API Public Key oleh Penyedia Layanan maupun Pengguna Layanan dilaksanakan berdasarkan asas kerahasiaan. Pengelolaan *key* sebagaimana dimaksud sekurang-kurangnya meliputi namun tidak terbatas pada:

1. Memiliki prosedur hak akses pengelolaan *key*.
2. Memiliki *database* yang aman sebagai tempat penyimpanan *key* dan hanya dapat diakses oleh pihak yang berwenang.
3. Memiliki prosedur pembuatan, pembaharuan, penghapusan, dan enkripsi/dekripsi *key*.

4. Menggunakan algoritma yang direkomendasikan lembaga standar internasional ataupun nasional dalam pembuatan *key.*
5. Terdapat kejelasan versi *master key* yang digunakan.
6. Memungkinkan dilakukan monitoring dan audit atas penggunaan *key*.

2. BAGIAN II: STANDAR TEKNIS DAN STANDAR KEAMANAN

Standar keamanan merupakan bagian dari Standar Nasional *Open* API Pembayaran yang bertujuan untuk memastikan kerahasiaan data, integritas data dan sistem, serta ketersediaan layanan, mengatur mengenai standar untuk otentikasi, otorisasi, enkripsi untuk menjamin integritas dan kerahasiaan data, terdapatnya *business continuity plan*, maupun penerapan *fraud detection system* untuk memitigasi potensi *fraud*. Selain mengacu pada standar keamanan tersebut, Penyedia Layanan dan Pengguna Layanan harus menerapkan kontrol dan perlindungan menyeluruh terhadap data dan informasi dari potensi risiko siber untuk melindungi sistem, data Konsumen maupun data terkait Penyedia Layanan dan/atau Pengguna Layanan.

2.1. KOMPONEN STANDAR TEKNIS DAN STANDAR KEAMANAN

Standar teknis dan keamanan dari Standar Nasional *Open* API Pembayaran menstandarkan hal-hal sebagai berikut:

1. Tipe Arsitektur
2. Format Data
3. *Character Encoding*
4. Komponen *HTTP Method*
5. Komponen Struktur Format *Header - Access Token* (*Business to Business* (B2B) dan *Business to Business to Consumer* (B2B2C))
6. Komponen Struktur Format *Header – Transaction* (B2B dan B2B2C)
7. Komponen *Server Authentication Method*
8. Komponen *Client Authentication Method*
9. Komponen Standar Enkripsi
10. Komponen *Secured Channel Communication*
11. Komponen Standardisasi *URI Path*
12. Komponen Standardisasi *Business Continuity Plan* (*Reliability*, *Availability*, dan *Scalability*)
13. Komponen Standardisasi Keamanan Lainnya

2.1.1. Tipe Arsitektur API

Tipe arsitektur yang digunakan **adalah *Representational State Transfer* (REST) API**.

2.1.2. Format Data

Format data yang digunakan pada *request body* dan *response body* adalah ***JavaScript Object Notation* (JSON)**.

2.1.3. Character Encoding

Standar *character encoding* yang digunakan adalah **UTF-8**.

2.1.4. Komponen *HTTP Method*

HTTP Method berfungsi sebagai identifikasi terhadap aksi yang ingin dilakukan pada suatu sumber daya (*resource*) dengan komponen HTTP-Verb yang pada umumnya digunakan. *HTTP-Verb* yang digunakan adalah:

1. ***POST** Request*
2. ***GET** Request*
3. ***DELETE** Request*
4. ***PUT** Request*

Sebagai pertimbangan keamanan, untuk *service* ***get Access Token*** menggunakan ***POST** Request*. Untuk *services* lainnya menggunakan HTTP-Verb yang disesuaikan untuk tipe operasi dan *resource* yang diakses. Penggunaan HTTP *method* untuk masing-masing *service* disebutkan pada tabel informasi umum pada dokumen spesifikasi teknis SNAP.

2.1.5. Komponen Struktur Format *Header – Access Token* (B2B dan B2B2C)

Setiap Pengguna Layanan yang ingin melakukan akses terhadap layanan API yang terdaftar untuk model *use case*:

1. B2B (integrasi antara PJP Penyedia Layanan dan Pengguna Layanan); atau
2. B2B2C (integrasi antara PJP Penyedia Layanan, Pengguna Layanan, dan Konsumen)

harus melakukan *access token request* terlebih dahulu dengan standar sebagai berikut:

a. Komponen Struktur *Format Header – Access Token Request* (B2B)

Service Code	73
Name	API Access Token B2B
Version	1.0
HTTP Method	POST
Path	../{version}/access-token/b2b

Struktur *Format Header* API untuk *Access Token Request* (B2B):

Area	***Field***	***Attribute***	***Type***	***Description***
Header	***Content-Type***	***Mandatory***	***String***	*String represents indicate the media type of the resource (e.g. application/json, application/pdf)*
	X-TIMESTAMP	***Mandatory***	***String***	*Client's current local time in yyyy-MM-ddTHH:mm:ss.SSSTZD format*
	X-CLIENT- KEY	***Mandatory***	***String***	*Client's client_id (PJP Name) (given at completion registration process)*
	X-SIGNATURE	***Mandatory***	***String***	*Non-Repudiation & Integrity checking* ***X-Signature*** *: dengan algoritma asymmetric signature SHA256withRSA (**Private_Key**, **stringToSign**).* ***stringToSign*** *= client_ID + "|" + X-TIMESTAMP*
Body	***grantType***	***Mandatory***	***String***	*"**client_credentials**" : The client can request an access token using only its*

Area	Field	Attribute	Type	Description
				client credentials (or other supported means of authentication) when the client is requesting access to the protected resources under its control (OAuth 2.0: RFC 6749 & 6750)
	additionalInfo	***Optional***	***Object***	*Additional Information*

b. Komponen Struktur *Format Header – Access Token Response* (B2B)

Sebagai *response* dari *access token request*, diatur standar dengan format sebagai berikut:

Area	Field	Attribute	Type	Description
Header	***X-TIMESTAMP***	***Mandatory***	***String***	*Client's current local time in yyyy-MM-ddTHH:mm:ss.SSSTZD format*
	X-CLIENT- KEY	***Mandatory***	***String***	*Client's client_id (PJP Name) (given at completion registration process)*
Body	***responseCode***	***Conditional***	***String***	*Refer to standar data dan spesifikasi teknis part 6 (Response Code).* *If access token failed to generate, this value must be filled.*
	responseMessage	***Conditional***	***String***	*Refer to standar data dan spesifikasi teknis part 6 (Response Message).* *If access token failed to generate, this value must be filled.*
	accessToken	***Mandatory***	***String (2048)***	*A string representing an authorization issued to the client that used to access protected resources*
	tokenType	***Mandatory***	***String***	*The access token type provides the client with the information required to successfully utilize the access token to make a protected resource request (along with type-specific attributes)* *Token Type Value:*

Area	Field	Attribute	Type	Description
				• ***"Bearer"****: includes the access token string in the request* • ***"Mac"****: issuing a Message Authentication Code (MAC) key together with the access token that is used to sign certain components of the HTTP requests* *Reference: OAuth2.0 RFC 6749 & 6750*
	expiresIn	***Mandatory***	***String***	*Session expiry in seconds: 900 (15 menit)*
	additionalInfo	***Optional***	***Object***	*Additional Information*

c. Komponen Struktur *Format Header – Access Token Request* (B2B2C)

Service Code	74
Name	API Access Token B2B2C
Version	1.0
HTTP Method	POST
Path	../{version}/access-token/b2b2c

Struktur *Format Header* API untuk *Access Token Request* (B2B2C):

Area	Field	Attribute	Type	Description
Header	***Content-Type***	***Mandatory***	***String***	*String represents indicate the media type of the resource (e.g. application/json, application/pdf)*
	X-TIMESTAMP	***Mandatory***	***String***	*Client's current local time in yyyy-MM-ddTHH:mm:ss.SSSTZD format*
	X-CLIENT-KEY	***Mandatory***	***String***	*Client's client_id (PJP Name)(given at completion registration process)*
	X-SIGNATURE	***Mandatory***	***String***	*Non-Repudiation & Integrity checking* ***X-Signature*** *: dengan algoritma asymmetric signature SHA256withRSA (****Private_Key****,* ***stringToSign****).*

Area	Field	Attribute	Type	Description
				stringToSign *= client_ID + "|" + X-TIMESTAMP*
Body	***grantType***	***Mandatory***	***String***	*Apply token request key type, can be AUTHORIZATION_CODE or REFRESH_TOKEN.*
	authCode	***Conditional***	***String (256)***	*The authorization code received after the User provides the consent. Mandatory if grantType = AUTHORIZATION_CODE*
	refreshToken	***Conditional***	***String (512)***	*Refresh token to get a new accessToken where the User doesn't need to provide the consent again. Mandatory if grantType = REFRESH_TOKEN. Refresh Token should be less than access token validity and will be manage by the PJP's application to generate a new access_token*
	additionalInfo	***Optional***	***Object***	*Additional Information*

d. Komponen Struktur *Format Header – Access Token Response* (B2B2C)

Sebagai *response* dari *access token request* diatur standar dengan format sebagai berikut:

Area	Field	Attribute	Type	Description
Header	***X-TIMESTAMP***	***Mandatory***	***String***	*Client's current local time in yyyy-MM-ddTHH:mm:ss.SSSTZD format*
	X-CLIENT-KEY	***Mandatory***	***String***	*Client's client_id (PJP Name)(given at completion registration process)*
Body	***responseCode***	***Conditional***	***String***	*Refer to standar data dan spesifikasi teknis part 6 (Response Code).* *If access token failed to generate, this*

Area	Field	Attribute	Type	Description
				value must be filled.
	responseMessage	***Conditional***	***String***	*Refer to standar data dan spesifikasi teknis part 6 (Response Code)* *If access token failed to generate, this value must be filled.*
	accessToken	***Mandatory***	***String (2048)***	*A string representing an authorization issued to the client that used to access protected resources.*
	tokenType	***Mandatory***	***String***	*The access token type provides the client with the information required to successfully utilize the access token to make a protected resource request (along with type-specific attributes)* *Token Type Value:* • ***"Bearer"****: includes the access token string in the request* • ***"Mac"****: issuing a Message Authentication Code (MAC) key together with the access token that is used to sign certain components of the HTTP requests* *Reference: OAuth2.0 RFC 6749 & 6750*
	accessTokenExpiryTime	***Mandatory***	***String***	*Time when the accessToken will be expired.* *Access token valid time will be 15 days* format ISO8601
	refreshToken	***Mandatory***	***String***	*A random string that can be used by specific client to get a refreshed accessToken to prolong the access to the User's resources.*
	refreshTokenExpiryTime	***Mandatory***	***String***	*Time when the refreshToken will be expired. Refresh Token should be less than access token validity and will be manage by the PJP's application to generate a new access_token*

Area	Field	Attribute	Type	Description
				format ISO8601
	additionalInfo	***Optional***	***Object***	*Additional Information*

2.1.6. Komponen Struktur Format *Header – Transaction* (B2B dan B2B2C)

Standar struktur *format header* untuk API level transaksi adalah sebagai berikut:

a. Komponen Struktur *Format Header – Transaction Request* (B2B)

Struktur *format header* API untuk *transaction request* (B2B):

Area	Field	Attribute	Type	Description
Header	***Content-Type***	***Mandatory***	***String***	*String represents indicate the media type of the resource (e.g. application/json, application/pdf)*
	Authorization	***Conditional***	***String***	*Represents access_token of a request; string starts with keyword "Bearer " followed by access_token (e.g. Bearer eyJraWQiOi...Jzc29zIiwiY)*
	X-TIMESTAMP	***Mandatory***	***String***	*Client's current local time in yyyy-MM-ddTHH:mm:ss.SSSTZD format*
	X-SIGNATURE	***Mandatory***	***String***	*Represents signature of a request.* *Identify Signature Type used* *Value:* *1 - Symmetric Signature with Get Token* *2 - Asymmetric Signature without Get Token* *Default Value: 1* *1.* ***Symetric-Signature*** *:*

Area	Field	Attribute	Type	Description
				HMAC_SHA512 (clientSecret, stringToSign) *dengan formula* ***stringToSign*** *= HTTPMethod +":"+ EndpointUrl +":"+ AccessToken +":"+ Lowercase(HexEncode(SHA-256(minify(RequestBody))))+ ":" + TimeStamp* 2. ***Asymetric-Signature*** *: **SHA256withRSA (clientSecret, stringToSign)** dengan formula **stringToSign** = HTTPMethod +":"+ EndpointUrl +":"+ Lowercase(HexEncode(SHA-256(minify(RequestBody)))) + ":" + TimeStamp* Catatan*:* 1. *Endpoint URL* lengkap termasuk seluruh parameter pada *URL* terkait (*Relative path*, contoh: *Path* pada informasi umum setiap API service) 2. Untuk parameter *minify(Request Body)*, dalam hal tidak terdapat *Request Body* maka digunakan string kosong.
	ORIGIN	***Optional***	***String***	*Origin Domain www.yourdomain.com*
	X-PARTNER-ID	***Mandatory***	***String (36)***	*Unique ID for a partner*
	X-EXTERNAL-ID	***Mandatory***	***String (36)***	*Alphanumeric. Reference number that should be unique in the same day*
	CHANNEL-ID	***Mandatory***	***String (5)***	*PJP's channel id* *Device identification on which the API services is currently being accessed by the end user (customer)*

Contoh *Header – Transaction Request* (B2B):

```
Content-type: application/json
Authorization: Bearer
gp9HjjEj813Y9JGoqwOeOPWbnt4CUpvIJbU1mMU4a11MNDZ7Sg5u9a"
X-TIMESTAMP: 2020-12-17T10:55:00+07:00
X-SIGNATURE:
85be817c55b2c135157c7e89f52499bf0c25ad6eeebe04a986e8c8625
61b19a5
ORIGIN: www.hostname.com
X-PARTNER-ID: 82150823919040624621823174737537
X-EXTERNAL-ID: 41807553358950093184162180797837
CHANNEL-ID: 95221
```

b. Komponen Struktur *Format Header – Transaction Request* (B2B2C)

Struktur *format header* API untuk *transaction request* (B2B2C):

Area	***Field***	***Attribute***	***Type***	***Description***
Header	***Content-Type***	***Mandatory***	***String***	*String represents indicate the media type of the resource (e.g. application/json, application/pdf)*
	Authorization	***Mandatory***	***String***	*Represents access_token of a request; string starts with keyword "Bearer " followed by access_token (e.g. Bearer eyJraWQiOi...Jzc29zIiwiY)*
	Authorization-Customer	***Mandatory***	***String***	*Represents access_token of a request belong customer; string starts with keyword "Bearer " followed by access_token (e.g. Bearer eyJrsWaiOi...Jzc523awiY)*
	X-TIMESTAMP	***Mandatory***	***String***	*Client's current local time in yyyy-MM-ddTHH:mm:ss.SSSTZD format*
	X-SIGNATURE	***Mandatory***	***String***	*Represents signature of a request* ***X-Signature****: algoritma symmetric signature* ***HMAC_SHA512 (clientSecret, stringToSign)*** *dengan formula* ***stringToSign*** *= HTTPMethod +":"+ EndpointUrl +":"+ AccessToken +":"+ Lowercase(HexEncode(SHA-256(minify(RequestBody))))+ ":" +*

Area	Field	Attribute	Type	Description
				TimeStamp Catatan*:* 1. *Endpoint URL* lengkap termasuk seluruh parameter pada *URL* terkait (*Relative path*, contoh: *Path* pada informasi umum setiap API service) 2. Untuk parameter *minify(Request Body),* dalam hal tidak terdapat *Request Body* maka digunakan string kosong.
	ORIGIN	***Optional***	***String***	*Origin Domain www.yourdomain.com*
	X-PARTNER-ID	***Mandatory***	***String (36)***	*Unique ID for a partner*
	X-EXTERNAL-ID	***Mandatory***	***String (36)***	*Numeric String. Reference number that should be unique in the same day*
	X-IP-ADDRESS	***Optional***	***String (15)***	*IP address of the end user (customer) using IPv4 format* *Example: 172.31.255.255*
	X-DEVICE-ID	***Mandatory***	***String (400)***	*Device identification on which the API services is currently being accessed by the end user (customer)* ***Sample****:* ***Web Application****:* *Mozilla / 5.0(Windows NT 10.0; Win64; x64)AppleWebKit / 537.36(KHTML, like Gecko)Chrome / 75.0.3770.100 Safari / 537.36 OPR / 62.0.3331.99* ***Mobile Application****:* *Android: android-20013adf6cdd8123f* *iOS: 72635bdfd223yvjm7246nsdj34hd4559393kjh42*
	CHANNEL-ID	***Mandatory***	***String (5)***	*PJP's channel id*

Area	*Field*	*Attribute*	*Type*	*Description*
				Device identification on which the API services is currently being accessed by the end user (customer)
	X-LATITUDE	***Optional***	***String (10)***	*Location on which the API services is currently being accessed by the end user (customer)* ***Refer to ISO 6709 Standard representation of geographic point location by coordinates*** *±DD.DDDD format (without minutes and seconds)* *±DD = three-digit integer degrees part of latitude* *.DDDD = variable-length fraction part in degrees* ***Sample****:* *New York City:* *Latitude: +40.75*
	X-LONGITUDE	***Optional***	***String (10)***	*Location on which the API services is currently being accessed by the end user (customer)* ***Refer to ISO 6709 Standard representation of geographic point location by coordinates*** *±DDD.DDDD format (without minutes and seconds)* *±DDD = four-digit integer degrees part of latitude* *.DDDD = variable-length fraction part in degrees* ***Sample****:* *New York City:* *Longitude: -074.00*

Contoh *Header – Transaction Request* (B2B2C)

```
Content-type: application/json
Authorization: Bearer
gp9HjjEj813Y9JGoqwOeOPWbnt4CUpvIJbU1mMU4a11MNDZ7Sg5u9a"
Authorization-Customer: Bearer
fa8sjjEj813Y9JGoqwOeOPWbnt4CUpvIJbU1mMU4a11MNDZ7Sg5u9a"
X-TIMESTAMP: 2020-12-23T09:10:11+07:00
X-SIGNATURE:
85be817c55b2c135157c7e89f52499bf0c25ad6eeebe04a986e8c8625
61b19a5
ORIGIN: www.hostname.com
X-PARTNER-ID: 82150823919040624621823174737537
X-EXTERNAL-ID: 41807553358950093184162180797837
X-IP-ADDRESS: 172.24.281.24
X-DEVICE-ID: 09864ADCASA
CHANNEL-ID: 95221
X-LATITUDE: -6.108841
X-LONGITUDE: 106.7782137
```

c. Komponen Struktur *Format Header – Transaction Response* (B2B dan B2B2C)

Struktur *format header* API untuk *transaction response* (B2B dan B2B2C):

Area	***Field***	***Attribute***	***Type***	***Description***
Header	***Content-Type***	***Mandatory***	***String***	*String represents indicate the media type of the resource*
	X-TIMESTAMP	***Mandatory***	***String***	*Client's current local time in yyyy-MM-ddTHH:mm:ss.SSSTZD format*

Contoh *Header – Transaction Response* (B2B dan B2B2C)

```
Content-type: application/json
X-TIMESTAMP: 2020-12-21T10:30:34+07:00
```

2.1.7. Komponen *Server Authorization* dan *Authentication Method*

Otorisasi adalah metode bagi Penyedia Layanan untuk memberikan akses *request* API dari Pengguna Layanan. Standar yang digunakan adalah:

- *OAuth* 2.0 sesuai RFC6749
- *Bearer token* sesuai RFC6750

Dalam memberikan akses kepada Pengguna Layanan, Penyedia Layanan melakukan otentikasi untuk memvalidasi Pengguna Layanan oleh Penyedia Layanan. Sarana yang digunakan adalah *credential* yang dipertukarkan pada saat proses pembentukan kerja sama, yaitu *client secret* dan pasangan *public/private key*, yang digunakan bersama dengan algoritma kriptografi tertentu.

2.1.8. Komponen *Client Authentication Method*

Client Authentication Method adalah metode otentikasi untuk memvalidasi konsumen. Standar *Two-Factor Authentication* yang digunakan adalah:

1. *Short Message Service* (SMS) TOTP (*Time based One Time Password*)
2. SMS TOTP dengan 6 digit- numerik dengan durasi 5 menit;
3. *Personal Identification Number* (PIN)
4. PIN dengan 6 digit-numerik
5. *Biometric* (*Fingerprint* & *Face Recognition*)
6. Lainnya

2.1.9. Komponen Standar Enkripsi

Model enkripsi terhadap *message* yang digunakan yaitu enkripsi asimetris dan simetris, menggunakan kombinasi *Private Key* dan *Public Key*, dengan standar sebagai berikut:

1. *Standard Asymmetric Encryption Signature*:
    - **SHA256withRSA** dengan *Private Key* (**Kpriv**) dan *Public Key* (**Kpub**) (256 bits)
2. *Standard Symmetric Encryption Signature*
    - **HMAC_SHA512** (512 bits)
3. *Standard Symmetric Encryption*
    - AES-256 dengan **client secret** sebagai *encryption key*.

2.1.10. Komponen *Secured Channel Communication*

Secured channel communication adalah kanal komunikasi yang aman untuk menjaga kerahasiaan *message* yang dikirimkan. Standar yang akan digunakan adalah:

1. *Transport Layer Security* (TLS) 1.3
2. Memiliki kemampuan untuk negosiasi ke TLS 1.2 namun dengan modul enkripsi minimum yang telah ditentukan sebagai berikut:
   a. TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
   b. TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
   c. TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
   d. TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

   Penggunaan TLS 1.2 dengan modul enkripsi minimum sebagaimana dimaksud pada angka 2 huruf a sampai dengan huruf d, hanya dapat diterapkan oleh Penyedia Layanan dan Pengguna Layanan sampai dengan tanggal 30 Juni 2026.

2.1.11. Komponen Standardisasi *Uniform Resources Indentifier* (URI*) Path*

Standardisasi URI *resource path* dengan format sebagai berikut:

/[domain_....._api]//[version]/[service-group]/[product-type]

1. [domain_...._api]
   The constant string of specific respective of PJP/Non-PJP api domain name
2. [version]
   The version of the APIs expressed as /**v**[major-version]**.**[minor-version]/
3. [service-group]
   The service-group identifies the group of endpoints
4. [product-type]
   Details of the resource if such service has another product definition underneath

2.1.12. Prinsip-Prinsip *Business Continuity Plan* (BCP)

Prinsip-prinsip standar BCP adalah sebagai berikut:

1. ***Reliability*** – untuk memastikan ketersediaan data dan layanan serta untuk menjamin kesinambungan proses bisnis.
2. ***Availability*** – memastikan sistem dan data tersedia untuk pengguna yang berwenang ketika mereka membutuhkannya. Melalui *Active-Active deployment* atau *Active - Stand-By default*.
3. ***Scalability*** – memastikan layanan dari produk jasa keuangan memiliki *response time* yang terukur.

Standar BCP adalah sebagai berikut:

No.	Infrastruktur Pendukung *Open* API Pembayaran berbasis SNAP	Persyaratan
1	Tipe *Data Recovery Center* untuk API *Management*	i. *HOT DRC* (RTO: <1 Hour, RPO: <1 Hour) ii. Replikasi data harus mendukung SLA RTO & RPO < 1 jam.
2	Kategori *data center* yang digunakan untuk API *Management*	RTO: <1 Hour RPO: <1 Hour
3	Terdapat regular *backup database* & *transaction log*	• *Backup database* (harian, mingguan, bulanan) • *Backup transaction log* • Retensi data & *log* : 10 tahun

2.1.13. Standar Keamanan Lainnya

a. Ketersediaan Kebijakan Tertulis Terkait Sistem Informasi

Penyedia Layanan dan Pengguna Layanan memiliki kebijakan atau prosedur tertulis terkait sistem informasi yang paling sedikit meliputi:

1) Manajemen *user*	4) Pengembangan *secure application*
2) Manajemen siber	5) *Change management*
3) Pengamanan dan perlindungan data (termasuk penyimpanan data)	6) Tata kelola sistem informasi

b. Pemenuhan Sertifikasi dan/atau Standar Keamanan dan Keandalan Sistem Informasi

1) Penyedia Layanan dan Pengguna Layanan *Open* API Pembayaran berbasis SNAP mengadopsi praktik-praktik umum terbaik dalam implementasi keamanan dan keandalan sistem informasi.
2) Penyedia dan Pengguna Layanan direkomendasikan memiliki sertifikasi dan/atau standar keamanan dan keandalan sistem informasi yang berlaku umum sesuai dengan jenis layanan yang diselenggarakan.

c. *Fraud Detection System* (FDS)

FDS adalah *tools* yang dipergunakan untuk mencegah, mendeteksi, memitigasi, menganalisis aktivitas *fraudulent* pada saat aktivitas tersebut teridentifikasi masuk ke dalam sistem serta mampu memberikan informasi/*alert* kepada petugas yang berwenang.

Open API Pembayaran berbasis SNAP dilengkapi dengan penerapan FDS. FDS didukung oleh kebijakan/prosedur dan sumber daya manusia yang diperlukan dalam implementasi/operasional FDS.

Fitur yang direkomendasikan diimplementasikan dalam FDS namun tidak terbatas pada:

1) Memiliki fleksiblitas untuk mengkonfigurasi *rules*/parameter sebelum dan sesudah implementasi FDS
2) Memiliki kemampuan untuk menerima dan mengolah data *fraud* yang bersumber dari luar

3) Memiliki kemampuan untuk menganalisis, memitigasi dan/atau memprioritaskan tindak lanjut berdasarkan potensi serangan/*fraud*
4) Kemampuan mendeteksi/mencegah anomali transaksi
5) Memiliki kemampuan untuk mendeteksi/mencegah potensial *fraud* sejak fase pendaftaran akun nasabah.

Rules/parameter yang direkomendasikan diimplementasikan dalam FDS namun tidak terbatas pada:

1) Waktu transaksi	5) Nominal	9) *Excessive login*
2) Frekuensi transaksi	6) *Negative balance*	10) *Device* ID
3) *Velocity**)	7) Akun *dormant*	11) *Fraudster* ID/*black list* akun
4) *Incorrect* PIN/OTP/*Password*/*other authentication method*	8) Negara asal dan/atau negara tujuan transaksi	12) Lokasi transaksi*)

*) dalam hal transaksi mencakup data lokasi

d. Pelaksanaan Audit Secara Berkala

Penyedia Layanan dan Pengguna Layanan melakukan audit secara berkala terhadap implementasi SNAP. Audit dilakukan oleh auditor independen.

e. Aspek Keamanan lainnya

1) Adanya penerapan *whitelisted IP* pada perangkat/aset yang digunakan untuk *Open* API Pembayaran berbasis SNAP dan perangkat pendukung lainnya.
2) Memiliki *firewall*

   Open API Pembayaran berbasis SNAP dilengkapi dengan *Web Application Firewall* baik menggunakan *Cloud Based*, *Network Based* ataupun *Host-Based Firewall* yang dapat melindungi dari *cyber attack* seperti *cross-site-scripting* (XSS), *cross-site forgery*, SQL *injection*, DDoS, *malware* dan lain lain.

Pengelolaan yang direkomendasikan diimplementasikan dalam *Firewall* namun tidak terbatas pada:

1) Adanya dokumen *firewall* (tujuan, layanan pengguna *firewall*, *rules*)	4) Manajemen/monitoring *network traffic*
2) *Access Control List* (ACLs)	5) Pengujian *firewall* secara berkala
3) *Rules* antara lain *packet filtering*, *antispoofing filter*, *user permit rules*, *permit management*, *alert* untuk *suspicious traffic* dan *traffic log*	6) Pengkinian *firewall* secara reguler

-Halaman Akhir-