 <b>TILAKA</b> NUSA TEKNOLOGI	PT TILAKA NUSA TEKNOLOGI	No. Dokumen	:	PRD-10-002
		Tanggal	:	26 Februari 2024
	Tilaka Sign Plus – Registrasi & Aktivasi Akun Regular	Revisi	:	3.0



## Halaman Persetujuan

DISUSUN OLEH	:	VICTORIA AMANDA	PRODUCT OWNER	
DITINJAU OLEH	:	ABDULLOH FAHMI	VP OF PRODUCT	
DISETUJUI OLEH	:	ABRAHAM FERDINAND INOERAWAN	CO-CTO	

## Riwayat Revisi Dokumen

Tanggal Revisi	Nomor Revisi	Penulis Revisi	Deskripsi/Uraian Revisi
12-10-2023	1.0	Product Owner	Pertama Kali Terbit
04-01-2024	2.0	Product Owner	<ul style="list-style-type: none"><li>• Penggantian penamaan callback notify verifikator/validator menjadi callback Certificate Status</li><li>• Penggantian penamaan callback KYC result menjadi callback Registration, dan update pada JSON body</li></ul>
26-02-2024	3.0	Product Owner	<ul style="list-style-type: none"><li>• Update pada flow penggunaan endpoint Tilaka</li><li>• Penambahan endpoint /checkcertstatus sebagai fallback ketika tidak menerima callback certificate status</li></ul>

## **Preface**

### Intended Audience

Dokumentasi ini ditujukan untuk customer Tilaka yang ingin mengintegrasikan sistem layanannya dengan layanan Tilaka untuk Digital Signature Personal. Pada dokumentasi ini dijelaskan detail penggunaan setiap endpoint dari API Digital Signature Personal yang disediakan oleh Tilaka yang meliputi autentikasi, parameter API dan juga webview yang memudahkan customer Tilaka untuk melakukan integrasi.

Pengguna dari dokumen ini adalah para pengembang dari customer Tilaka yang bertanggung jawab untuk melakukan integrasi sistem.

## Daftar Isi

Halaman Persetujuan.....	ii
Riwayat Revisi Dokumen.....	iii
Preface.....	iv
Daftar Isi.....	v
<b>1. Overview.....</b>	<b>1</b>
1.2. Conventions.....	1
1.3. Prerequisite .....	1
1.3.1. Endpoint Callback Certificate Status .....	1
1.3.2. Endpoint Penautan .....	1
1.3.3. Endpoint Callback Registration .....	2
1.3.4. Identifier and Credentials.....	2
1.4. Authentication and Authorization.....	2
1.4.1. Authentication.....	2
1.4.2. Authorization.....	3
1.5. Header Token Validation untuk Callback.....	3
1.6. URL Format .....	4
<b>2. Generate UUID .....</b>	<b>5</b>
2.1. Resource Information.....	5
2.2. Request .....	5
2.3. Parameters .....	5
2.4. Authorization .....	5
2.5. Response .....	5
2.6. Samples .....	6
2.6.1. Sample Request .....	6
2.6.2. Sample Response .....	6
<b>3. Pengecekan Akun .....</b>	<b>8</b>
3.1. Resource Information.....	8
3.2. Request .....	8
3.3. Request Body.....	8
3.4. Parameters .....	8
3.5. Authorization .....	9
3.6. Response .....	9
3.7. Samples .....	9
3.7.1. Sample Request .....	9

3.7.2.	Sample Response .....	10
4.	Register User .....	11
4.1.	Resource Information .....	11
4.2.	Request .....	11
4.3.	Request Body .....	11
4.4.	Parameters .....	12
4.5.	Authorization .....	13
4.6.	Response .....	13
4.7.	Samples .....	14
4.7.1.	Sample Request .....	14
4.7.2.	Sample Response .....	15
5.	Registration Webview .....	17
5.1.	Resource Information .....	17
5.2.	Request .....	17
5.3.	Parameters .....	17
5.4.	Authorization .....	17
5.5.	Response .....	17
6.	Callback Certificate Status .....	25
6.1.	Resource Information .....	25
6.2.	Request .....	25
6.3.	Request Body .....	25
6.4.	Parameters .....	25
6.5.	Response .....	26
6.6.	Samples .....	26
6.6.1.	Sample Request .....	26
7.	Penautan/Aktivasi Akun Webview .....	28
7.1.	Resource Information .....	28
7.2.	Request .....	28
7.3.	Parameters .....	28
7.4.	Authorization .....	29
7.5.	Response .....	29
8.	Callback Registration .....	34
8.1.	Resource Information .....	34
8.2.	Request .....	34
8.3.	Request Body .....	34
8.4.	Parameters .....	34

8.5.	Response .....	37
8.6.	Samples .....	37
8.6.1.	Sample Request .....	37
9.	User Registration Result.....	40
9.1.	Resource Information.....	40
9.2.	Request .....	40
9.3.	Request Body.....	40
9.4.	Parameters .....	40
9.5.	Authorization .....	40
9.6.	Response .....	41
9.7.	Samples .....	43
9.7.1.	Sample Request .....	43
9.7.2.	Sample Response .....	43
10.	Change MFA Webview .....	46
10.1.	Resource Information .....	46
10.2.	Request.....	46
10.3.	Parameters.....	46
10.4.	Authorization .....	47
10.5.	Response.....	47
11.	Change Set Signature Webview .....	50
11.1.	Resource Information .....	50
11.2.	Request.....	50
11.3.	Parameters.....	50
11.4.	Authorization .....	51
11.5.	Response.....	51
12.	Request Revoke Certificate .....	53
12.1.	Resource Information .....	53
12.2.	Request.....	53
12.3.	Request Body .....	53
12.4.	Parameters.....	53
12.5.	Authorization .....	54
12.6.	Response.....	54
12.7.	Samples .....	55
12.7.1.	Sample Request .....	55
12.7.2.	Sample Response .....	55
13.	Check Certificate Status.....	59

13.1.	Resource Information .....	59
13.2.	Request.....	59
13.3.	Request Body .....	59
13.4.	Parameters.....	59
13.5.	Authorization .....	60
13.6.	Response.....	60
13.7.	Samples .....	60
13.7.1.	Sample Request .....	60
13.7.2.	Sample Response .....	61
14.	Status Codes .....	65
15.	Appendix - Endpoint & Webview URL .....	66



## 1. Overview

Tilaka menyediakan API Personal untuk integrasi dengan penyedia layanan (pihak ketiga) yang membutuhkan fungsi *Digital Signature* (Tandatangan Elektronik). Tandatangan elektronik yang dimaksud dalam dokumen ini menggunakan sertifikat elektronik yang diterbitkan oleh PSrE Tilaka.

Sertifikat elektronik yang diterbitkan oleh PSrE Tilaka untuk API Personal pada dokumen ini adalah sertifikat elektronik untuk Personal, sesuai yang dijelaskan pada CPS Tilaka. Dengan demikian, user dari pihak ketiga yang melakukan tandatangan elektronik berperan sebagai user personal.

API Personal dapat dikelompokkan menjadi dua bagian utama, yaitu manajemen akun dan proses tandatangan elektronik. Manajemen akun bertujuan untuk menautkan akun user pada sistem pihak ketiga dengan akun user PSrE Tilaka sehingga sistem pihak ketiga dapat melakukan request kepada user untuk menandatangani dokumen elektronik.

Pada bagian manajemen akun, terdapat kondisi di mana:

a. User belum memiliki akun Tilaka

Untuk kondisi ini, maka user perlu melakukan registrasi terlebih dahulu untuk dapat memperoleh sertifikat elektronik yang akan digunakan pada proses penandatanganan dokumen elektronik.

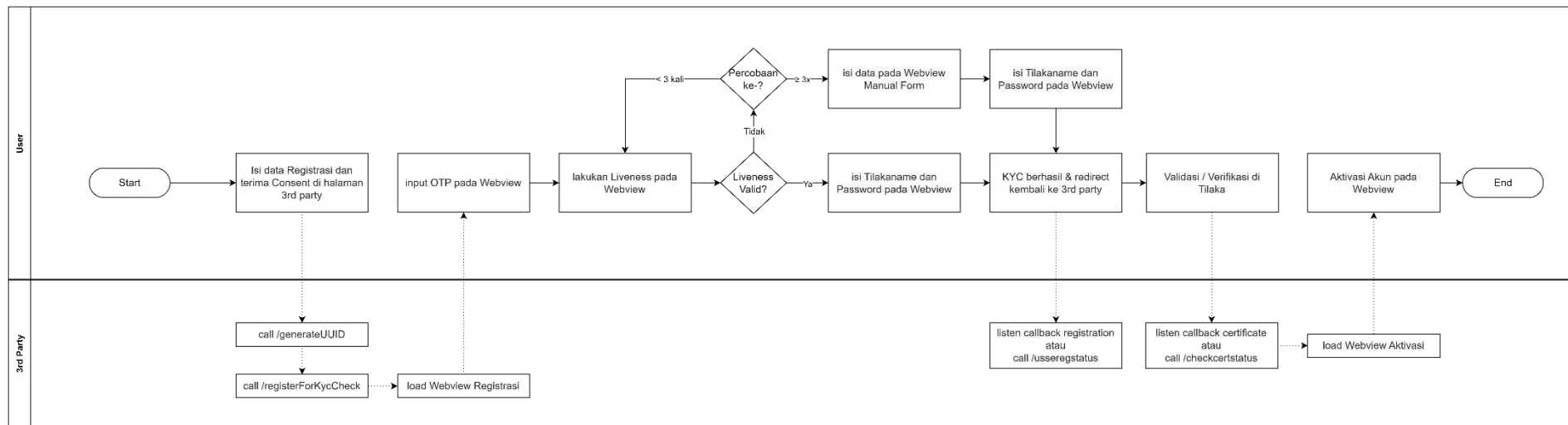
Pada API Personal, tersedia endpoint untuk melayani proses registrasi user, yang akan dijelaskan pada bagian selanjutnya pada dokumen ini.

b. User sudah memiliki akun Tilaka

Untuk kondisi ini, maka user dapat menautkan akunnya dengan cara login menggunakan Tilaka Name dan password miliknya.

Penautan akun dapat dilakukan menggunakan endpoint penautan akun yang akan memberikan nilai balikan yaitu Tilaka Name untuk ditautkan dengan data user pada sistem pihak ketiga.

Gambar di bawah ini menunjukkan flow proses secara garis besar untuk dapat memahami penggunaan API Tilaka pada tahap Registrasi dan Aktivasi Akun. Setelah sukses Aktivasi Akun, status sertifikat user akan berubah menjadi Aktif, dan akun user sudah dapat digunakan untuk melakukan signing.



## 1.2. Conventions

API Personal menggunakan konvensi berikut dalam dokumen ini:

- Respon dijelaskan pada bagian “nilai balikan” pada setiap method.
- Respon dapat dalam bentuk JSON format maupun webview.
- Setiap parameter dari request adalah wajib, kecuali ditandai dengan **optional**.
- Tipe yang diterima untuk sebuah parameter request ditunjukkan pada kolom **values**.
- Simbol | diartikan sebagai OR.

## 1.3. Prerequisite

Untuk dapat melakukan integrasi ke API Personal, ada beberapa hal yang harus dipersiapkan di awal.

### 1.3.1. Endpoint Callback Certificate Status

Endpoint callback certificate status ini disiapkan oleh pihak ketiga untuk memperoleh informasi hasil verifikasi dari verifikator, dan hasil validasi dari validator atas registrasi sertifikat user.

Endpoint ini menggunakan method POST, dengan url yang di define oleh pihak ketiga. Informasi detail mengenai endpoint ini dapat dilihat pada bagian 6 di dalam dokumen ini.

### 1.3.2. Endpoint Penautan

Endpoint penautan perlu disiapkan oleh pihak ketiga untuk dapat memperoleh informasi terkait Tilaka Name dari user yang akan melakukan tandatangan elektronik. Endpoint tersebut harus memiliki struktur sebagai berikut:

- Method : GET
- Parameter request:
  - request-id : berisi string UUID yang digunakan untuk request;
  - tilaka-name : berisi Tilaka Name pengguna yang akan ditautkan dengan user yang ada pada sistem pihak ketiga.

### 1.3.3. Endpoint Callback Registration

Endpoint callback Registration ini disiapkan oleh pihak ketiga untuk menerima hasil eKYC dari proses Registrasi. Endpoint ini menggunakan method POST, dengan url yang di define oleh pihak ketiga.

Informasi detail mengenai endpoint ini dapat dilihat pada bagian 8 di dalam dokumen ini.

### 1.3.4. Identifier and Credentials

Untuk dapat memanggil endpoint dari API Personal, dibutuhkan *identifier* dan *credential* berupa sertifikat untuk mTLS dan juga channel id dan secret yang disediakan oleh Tilaka untuk digunakan pada proses autentikasi dan otorisasi. *Identifier* dan *credential* ini dapat diperoleh melalui perjanjian kerjasama oleh pihak ketiga dengan Tilaka.

## 1.4. Authentication and Authorization

API Personal yang disediakan oleh PSrE Tilaka dilindungi dengan autentikasi dan otorisasi untuk dapat diakses. Autentikasi meliputi untuk sistem pihak ketiga yang akses ke API Personal dan user. Sedangkan, pada API Personal otorisasi menggunakan standard OAuth2, dimana ada yang melibatkan user untuk dapat akses API Personal dan ada yang hanya untuk sistem ke sistem menggunakan *client credential*.

### 1.4.1. Authentication

Ada dua entitas yang memerlukan autentikasi, yaitu sistem dan user. Setiap endpoint pada dokumen ini akan ada keterangan autentikasi yang digunakan. Tidak semua endpoint akan melibatkan autentikasi user, namun sudah pasti akan melibatkan autentikasi sistem.

Untuk sistem ada 2 mekanisme autentikasi, yaitu:

- 1) mTLS. Di layer proxy akan ada mutual autentikasi menggunakan mTLS. Sertifikat TLS akan disediakan oleh Admin Tilaka.
- 2) Client id dan secret. Setiap pihak ketiga yang akan integrasi dengan API Personal akan diberikan client id (pada dokumen ini lebih dikenal sebagai channel id) dan secret oleh Admin Tilaka yang akan digunakan oleh sistem pihak ketiga untuk mendapatkan access token.

Untuk user, autentikasi menggunakan *multi factor authentication* (MFA) yang terdiri dari:

- 1) Tilaka Name dan Kata Sandi; dikombinasikan dengan
- 2) Email OTP atau *face recognition* (FR)

### 1.4.2. Authorization

Setiap endpoint yang disiapkan API Personal akan memeriksa otorisasi untuk memastikan request berasal dari channel pihak ketiga yang bekerjasama dengan Tilaka. Untuk mendapatkan kode otorisasi dapat mengakses:

POST https://<host>:<port>/auth/token

Dengan parameter sebagai berikut:

Name	Type	Description
client_id	string	Berisi client id yang diberikan oleh Tilaka.
grant_type	string	Diisi dengan : client_credentials
client_secret	string	Berisi secret yang diberikan oleh Tilaka.

Respon yang didapat adalah:

```
{
  "access_token": "<string access_token>",
  "expires_in": 300,
  "refresh_expires_in": 0,
  "token_type": "Bearer",
  "not-before-policy": 1649663499,
  "scope": "register email profile"
}
```

String access token tersebut yang digunakan sebagai bearer untuk akses ke endpoint.

### 1.5. Header Token Validation untuk Callback

Token callback berlaku untuk seluruh callback, dan di define oleh Tilaka dengan spec token berikut ini:

- 1) Token dapat dilihat pada header 'x-validation-token'
- 2) Formulasi untuk validasi token di 'x-validation-token' adalah  
 hmac-sha-256(client-id + header x-request-timestamp + payload \*berupa json string) dengan key client-secret

Contoh penggunaan formulasi validasi token:

Client id:

```
33e8ca46-affe-4c39-804a-g4ft7w24pcq9
```

Plain text – callback certificate status:

```
{"user_identifier":"usertest1",  
"status":2}
```

header x-request-timestamp:

```
2022-11-07 13:30:58
```

Formulasi = client-id + header x-request-timestamp + payload

```
33e8ca46-affe-4c39-804a-g4ft7w24pcq92022-11-  
0713:30:58{"user_identifier":"usertest1","status":2}
```

Secret Key:

```
p4a3e36d-95fb-46aa-be26-7e82432jk423
```

Kemudian di lakukan cryptographic hash dengan metode SHA-256, sehingga didapatkan output hash berupa X-validation-token yang ada pada bagian header:

Hashed output in hex format:

```
ced2d53124e11faba1ef579f4e006f96769d95ff4e8bb6f96c78ce2d23a1291e
```

## 1.6. URL Format

API Personal menggunakan URL dengan format sebagai berikut:

```
<protocol>://<host>:<port>/<MethodName>
```

Example:

```
GET https://api.website.com/generateUUID
```

## 2. Generate UUID

Endpoint ini dipanggil pertama kali ketika hendak menyambungkan akun user ke akun user Tilaka. UUID selanjutnya akan digunakan untuk tracking proses hingga proses selesai.

### 2.1. Resource Information

Method	Purpose
Response formats	JSON
Requires authentication	Ya, menggunakan client id dan secret untuk mendapatkan bearer token

### 2.2. Request

Method	URL
POST	https://<host>:<port>/generateUUID

### 2.3. Parameters

Tidak ada parameter.

### 2.4. Authorization

Request ini membutuhkan otorisasi OAuth2 menggunakan client credential flow yaitu menggunakan client id dan secret (lihat bagian 1.4.2 pada dokumen ini).

### 2.5. Response

Response dalam bentuk JSON dengan struktur sebagai berikut:

```
{
  "success": boolean,
  "message": string,
  "data": [
    "UUID"
  ]
}
```

Name	Type	Description
success	boolean	Boolean value untuk menunjukkan bahwa UUID berhasil <i>digenerate</i> atau tidak. Bernilai true menunjukan berhasil dan false jika terjadi permasalahan.
message	string	Berisi informasi alasan terjadi error.
data	array	Akan berisi UUID jika request berhasil, jika gagal akan null.

## 2.6. Samples

### 2.6.1. Sample Request

Tidak ada parameter pada body.

### 2.6.2. Sample Response

Jika berhasil akan mendapatkan response dengan status 200 dan data JSON sebagai berikut:

```
{
  "success": true,
  "message": "Success",
  "data": [
    "094a6148-0da8-411c-87ba-5d4f2d43f3eb"
  ]
}
```



Jika channel id tidak ditemukan pada token namun authentication berhasil maka akan response 200 namun dengan informasi sebagai berikut:

```
{  
  "success": false,  
  "message": "Error, No channel-id Not Exist",  
  "data": null  
}
```

Jika gagal, akan menerima error dengan kemungkinan sebagai berikut:

Status	Response
401	<pre>{   "success": false,   "message": "Unauthenticated",   "data": null }</pre>

### 3. Pengecekan Akun

Endpoint ini dipanggil untuk melakukan pengecekan akun yang terdaftar berdasarkan Nomor Induk Kependudukan (NIK). Melalui response pada endpoint, dapat dilihat apakah user:

1. Belum memiliki nik terdaftar sehingga perlu melakukan registrasi; atau
2. NIK sudah terdaftar dan sudah memiliki tilaka id sehingga dapat melakukan penautan akun. Sebelum melakukan penautan, step pengecekan akun harus dijalankan. Apabila pengecekan akun di skip, maka penautan akun tidak dapat dilakukan.

#### 3.1. Resource Information

Method	Purpose
Response formats	JSON
Requires authentication	Ya, menggunakan client id dan secret

#### 3.2. Request

Method	URL
POST	<a href="https://&lt;host&gt;:&lt;port&gt;/checkAkunDSExist">https://&lt;host&gt;:&lt;port&gt;/checkAkunDSExist</a>

#### 3.3. Request Body

Bentuk request dalam JSON dengan struktur sebagai berikut:

```
{
  "request_id": string,
  "nik": string,
}
```

#### 3.4. Parameters

Parameter pada request body adalah sebagai berikut:

Name	Type	Description	Required
request_id	string		Yes
Nik	string	Nomor Induk Kependudukan	Opsional

### 3.5. Authorization

Request ini membutuhkan otorisasi OAuth2 menggunakan client credential flow yaitu menggunakan client id dan secret (lihat bagian 1.4.2 pada dokumen ini).

### 3.6. Response

Response dalam bentuk JSON dengan struktur sebagai berikut:

```
{
  "tilaka_id": string,
  "message": string,
  "status": boolean
}
```

Name	Type	Description
tilaka_id	string	Jika sudah memiliki akun tilaka, maka akan berisi Tilaka ID, jika belum maka akan dalam bentuk empty string.
message	string	Akan berisi pesan error jika belum memiliki Tilaka ID.
status	boolean	Menunjukkan kondisi response, true jika sudah ada Tilaka ID dan false jika belum.

### 3.7. Samples

#### 3.7.1. Sample Request

```
{
  "request_id": "094a6148-0da8-411c-87ba-5d4f2d43f3eb",
  "nik": "3276030304990002"
}
```

### 3.7.2. Sample Response

Jika berhasil, akan ada tiga kondisi yang mungkin muncul, yaitu:

- 1) User sudah ada akun Personal, sehingga bisa lanjut ke proses penautan akun:

```
{
  "tilaka_id": "3641537023829",
  "message": null,
  "status": true
}
```

- 2) Akun user dengan NIK yang diperiksa sedang dalam proses registrasi:

```
{
  "tilaka_id": "",
  "message": "Account Verification In Progress",
  "status": false
}
```

- 3) Belum ada user Tilaka dengan NIK yang diperiksa, sehingga akan lanjut ke proses pendaftaran:

```
{
  "tilaka_id": "",
  "message": "NIK Not Exist",
  "status": false
}
```

Sedangkan error yang mungkin terjadi adalah:

Status	Response
401	<pre>{   "success": false,   "message": "Unauthenticated",   "data": null }</pre>

## 4. Register User

Merupakan endpoint untuk initiate proses registrasi user. Inisiasi ini penting sebelum user dapat melakukan registrasi. Jika proses ini error, maka user tidak dapat melakukan registrasi.

### 4.1. Resource Information

The resource information is as follows:

Method	Purpose
Response formats	JSON
Requires authentication	Ya, menggunakan client id dan secret

### 4.2. Request

Method	URL
POST	https://<host>:<port>/registerForKycCheck

### 4.3. Request Body

Bentuk request dalam JSON dengan struktur sebagai berikut:

```
{
  "registration_id": string,
  "email": string,
  "name": string,
  "company_name": string,
  "date_expire": string,
  "nik": string,
  "photo_ktp": string, -- base 64
  "is_approved": boolean,
  "consent_text": string,
  "version": string,
```

```

    "hash_consent": string,

    "consent_timestamp": string
}

```

#### 4.4. Parameters

Parameter pada request body adalah sebagai berikut:

Name	Type	Description	Required
registration_id	string	UUID yang didapat dari Generate UUID	Yes
email	string	Alamat email pengguna yang akan didaftarkan	Yes
name	string	Nama lengkap pengguna yang akan didaftarkan sesuai KTP-el	Yes
company_name	string	Perusahaan di mana akun dikelompokkan. Untuk kebutuhan user personal, company name diisi dengan "Personal" (case sensitive).	Yes
nik	string	Nomor Induk Kependudukan	type B and D
photo_ktp	base 64		type B and D
date_expire	string	Tanggal Kadaluarsa (memiliki format YYYY-MM-DD hh:mm) dengan ketentuan jam 00:00-23:59	Yes
is_approved	boolean	True/false, sesuai dengan aksi yang dilakukan user terhadap SnK/TnC	Yes
consent_text	string	String content dari SnK/TnC	Yes
version	string	Versi dari SnK/TnC (contoh: TNT – v.1.0.1)  <b>note: maks karakter 20</b>	Yes
hash_consent	string	Berisi hash value dari consent, perhitungan hash dapat merefer pada rumus di bawah	Yes
consent_timestamp	string	YYYY-MM-DD hh:mm:ss  berisikan timestamp versi suatu SnK/TnC (tidak berubah, kecuali ada	Yes

		perubahan pada SnK/TnC)	
--	--	-------------------------	--

Catatan:

type B adalah pihak ketiga yang sudah memiliki data pengguna berupa NIK dan foto KTP;  
type D adalah pihak ketiga yang sudah memenuhi type B dan juga sudah melakukan liveness check terhadap user.

Formulasi untuk hash\_consent adalah:

```
hmac-sha-256 (client_secret, (channel_id + consent_text + version + consent_timestamp))
```

Contoh penggunaan formulasi hash\_consent:

Channel id:

```
33e8ca46-affe-4c39-804a-g4ft7w24pcq9
```

Consent\_text:

```
Terms of service are abc and d
```

Version:

```
TNT - v.1.0.1
```

Consent timestamp:

```
2023-01-01 18:30:00
```

Kemudian di lakukan cryptographic hash dengan metode SHA-256, dengan key nya adalah client\_secret sehingga didapatkan output berupa hash

## 4.5. Authorization

Request ini membutuhkan otorisasi OAuth2 menggunakan client credential flow yaitu menggunakan client id dan secret (lihat bagian 1.4.2 pada dokumen ini).

## 4.6. Response

Response dalam bentuk JSON dengan struktur sebagai berikut:

```
{
  "success": boolean,
  "message": string,
  "data": object
}
```

Name	Type	Description
success	boolean	Menunjukkan kondisi response, true jika berhasil diinisiasi untuk registrasi dan false jika tidak berhasil inisiasi untuk registrasi.
message	string	Memberikan informasi error jika gagal inisiasi untuk registrasi.
data	array	Berisi data jika diperlukan.

## 4.7. Samples

### 4.7.1. Sample Request

Ini contoh request untuk type B, yaitu user dapat langsung melakukan liveness dan selanjutnya set Tilaka Name dan Kata Sandi:

```
{
  "registration_id": "8aec06d8-78ba-4136-9ecb-497a98d63529",
  "email": " anita@mail.com ",
  "name": "Anita",
  "company_name": "Personal",
  "nik": "1012021209910001",
  "photo_ktp": "data:image/jpeg;base64,<base64 string of photo_ktp>",
  "date_expire": "2022-10-10 11:02",
  "is_approved": true,
  "consent_text": "Terms of service are abc and d",
  "version": "TNT - v.1.0.1",
  "hash_consent": "8edd10ac22ff37c353ab62915c2235057658a47dfcfe48994748ef16b2f63962",
  "consent_timestamp": "2023-01-01 18:30:00"
}
```

Note:

Ketika client telah mengirimkan data melalui /registerForKycCheck, maka data NIK user akan terikat pada suatu registration id & date\_expire nya. Use case:

User A dengan NIK 2711...001, registration id A, date expire 28-08-2023 13:00 melakukan liveness, namun saat sedang liveness koneksi nya terputus. Ketika koneksi nya terhubung kembali, user A dapat melanjutkan mengisi form dengan registration id A. Sehingga data registrasi ini harus disimpan juga di sisi client.



Pada case dimana data tidak disimpan oleh client, maka user A harus melakukan proses registrasi dari awal. Dan user A tidak dapat melakukan registrasi apabila date\_expire nya belum terlewati (karena NIK tersebut masih dalam proses di registration id yang sebelumnya).

#### 4.7.2. Sample Response

Jika berhasil meregistrasi pengguna, maka akan memberikan respon 200 dengan data sebagai berikut:

```
{
  "success": true,
  "message": "Data Diterima",
  "data": [
    "8aec06d8-78ba-4136-9ecb-497a98d63529",
    "anita@mail.com"
  ]
}
```

Jika mencoba melakukan registrasi pengguna & pengguna tersebut masih memiliki registration id lain yang belum expired, maka akan memberikan respon 200 dengan data sebagai berikut:

```
{
  "success": false,
  "message": "NIK sedang dalam proses pendaftaran/verifikasi",
  "data": null
}
```

Jika request berhasil namun registrasi tidak dapat dilanjutkan, akan memberikan respon 200 dengan informasi sebagai berikut:

```
{
  "success": false,
  "message": "Error Response from Call eKyc API",
  "data": [
    "budi0201"
  ]
}
```

Sedangkan error yang mungkin terjadi adalah:

Status	Response
401	<pre>{   "success": false,   "message": "Unauthenticated",   "data": null }</pre>
500	<pre>{   "code": 500,   "message": "There was an error processing your request. It has been logged (ID b069879f99de99a7)." }</pre>

## 5. Registration Webview

Untuk proses registrasi, API Personal menyediakan webview untuk memudahkan integrasi. Webview ini langsung dipanggil oleh client, dengan menyertakan uuid yang digunakan sebagai parameter pada saat memanggil endpoint Register User (/registerForKycCheck).

### 5.1. Resource Information

Method	Purpose
Response formats	Webview
Requires authentication	Tidak ada

### 5.2. Request

Method	URL
GET	https://<host>:<port>/personal-webview/guide?request_id=<uuid>

### 5.3. Parameters

Parameter pada request body adalah sebagai berikut:


Name	Type	Description	Required
request_id	string	Merupakan parameter uuid yang digunakan saat memanggil endpoint Register User. Uuid ini digunakan untuk tracking proses registrasi.	Yes


### 5.4. Authorization






Tidak ada otorisasi, endpoint ini langsung diakses oleh aplikasi di pengguna.

### 5.5. Response

Respon dalam bentuk webview dengan flow untuk registrasi. Berikut adalah webview yang menjadi respon dari endpoint Registration Webview.

No	Webview	Description
1.	<p><b>Liveness</b></p> <p>Mohon perhatikan hal-hal berikut saat pengambilan wajah untuk penilaian yang maksimal.</p>  <ul style="list-style-type: none"> <li>• Wajah menghadap kamera dengan latar belakang yang jelas.</li> <li>• Lepaskan atribut seperti kacamata, topi dan masker, serta rambut tidak menutupi wajah.</li> <li>• Pastikan pencahayaan baik, tidak terlalu terang atau terlalu gelap.</li> </ul> <p><b>Mulai</b></p> <p>POWERED BY <b>TILAKA</b> NUSA TEKNOLOGI</p>	<p>Merupakan halaman pertama yang akan muncul pada flow, halaman ini berisi panduan untuk melakukan liveness.</p> <p>Pengguna dapat klik button 'mulai' untuk memulai proses liveness.</p>

<p>2.</p>	<p><b>Liveness</b></p> <p>Pastikan wajah di dalam garis panduan dan ikuti petunjuk dengan benar</p> 	<p>Halaman ini digunakan pengguna untuk melakukan liveness. Pada halaman ini terdapat garis panduan dan arahan yang dapat diikuti pengguna.</p> <p>Dalam proses liveness, pengguna akan diminta melakukan gerakan sesuai dengan panduan. Gerakan ini akan di capture oleh Liveness Engine.</p>
<p>3.</p>	<p><b>Liveness</b></p> <p>Terima kasih telah mengikuti proses Liveness. Hasil dinilai berdasarkan keaslian serta kesesuaian foto dengan aksi yang diminta.</p> 	<p>Setelah pengguna selesai melakukan seluruh gerakan yang diminta, akan dilakukan pemeriksaan pada hasil liveness dengan kondisi:</p> <ol style="list-style-type: none"> <li>1. Liveness berhasil, lanjut ke halaman Aktivasi Akun</li> <li>2. Liveness gagal dan percobaan masih &lt; 3x, lanjut ke halaman nomor 5, dan</li> <li>3. Liveness pengguna gagal Liveness sebanyak 3x, akan tampil halaman nomor 6.</li> </ol>

4.	<div data-bbox="399 235 590 268"> <h3>Aktivasi Akun</h3> </div> <div data-bbox="544 324 754 506">  </div> <div data-bbox="402 553 882 607"> <p>Mohon mengisi data-data berikut sebagai proses aktivasi akun Tilaka:</p> </div> <div data-bbox="413 629 563 656"> <p>Tilaka Name </p> </div> <div data-bbox="413 683 657 712"> <input type="text" value="Masukkan Tilaka Name"/> </div> <div data-bbox="413 763 526 792"> <p>Kata Sandi</p> </div> <div data-bbox="413 817 643 846"> <input type="password" value="Masukkan Kata Sandi"/>  </div> <div data-bbox="413 898 632 927"> <p>Konfirmasi Kata Sandi</p> </div> <div data-bbox="413 952 754 981"> <input type="password" value="Masukkan Konfirmasi Kata Sandi"/>  </div> <div data-bbox="399 1041 906 1097"> <p><input type="checkbox"/> Saya setuju dengan <a href="#">CP/CPS</a>, <a href="#">Kebijakan Jaminan</a>, <a href="#">Kebijakan Privasi</a>, dan <a href="#">Perjanjian Pemilik Sertifikat</a></p> </div> <div data-bbox="558 1140 740 1169"> <p>AKTIVASI AKUN</p> </div> <div data-bbox="588 1229 711 1292"> <p>POWERED BY </p> </div>	<p>Halaman Aktivasi Akun ini akan tampil jika liveness berhasil, user akan diminta untuk set Tilaka Name dan Kata Sandi yang nantinya akan digunakan untuk autentikasi user.</p> <p>Setelah halaman ini akan ada kondisi di mana:</p> <ol style="list-style-type: none"> <li>1. Tilaka Name dan Kata Sandi yang dimasukkan sesuai dengan aturan sehingga lanjut ke halaman nomor 8.</li> <li>2. Tilaka Name dan Kata Sandi yang dimasukkan belum sesuai dengan aturan sehingga muncul halaman nomor 7.</li> </ol>
----	---	---

5.	<div data-bbox="537 226 746 266" data-label="Section-Header"> <h3>Liveness Gagal</h3> </div> <div data-bbox="501 369 783 645" data-label="Image"> </div> <div data-bbox="416 754 871 844" data-label="Text"> <p>Maaf, proses Liveness Anda gagal. Foto dan aksi yang diminta tidak sesuai. Mohon ulangi proses Liveness dan ikuti petunjuk dengan benar.</p> </div> <div data-bbox="595 967 691 999" data-label="Text"> <p>ULANGI</p> </div> <div data-bbox="582 1057 703 1120" data-label="Image"> </div>	<p>Halaman webview ini akan ditampilkan apabila pengguna gagal dalam percobaan liveness pertama dan kedua.</p> <p>Pengguna dapat melakukan Liveness kembali menggunakan button 'ulangi' untuk kembali ke halaman nomor 2.</p>
----	--	---

6.	<div data-bbox="542 230 751 268" data-label="Section-Header"> <h3>Liveness Gagal</h3> </div> <div data-bbox="504 374 791 656" data-label="Image"> </div> <div data-bbox="427 768 868 855" data-label="Text"> <p>Mohon mengisi Formulir yang dikirim ke email yeshica@gmail.com untuk melanjutkan proses aktivasi akun.</p> </div> <div data-bbox="477 981 818 1016" data-label="Text"> <p><a href="#">Kembali ke Halaman Utama</a></p> </div> <div data-bbox="587 1072 707 1135" data-label="Image"> </div>	<p>Halaman ini akan muncul jika user telah gagal liveness 3x. Untuk registrasi, user akan melakukan registrasi manual yang dikirim melalui email.</p> <p>Halaman webview berhenti di sini dan user dapat kembali ke halaman pihak ketiga dengan klik 'Kembali ke Halaman Utama'.</p>
----	---	--



7.

## Aktivasi Akun



Mohon mengisi data-data berikut sebagai proses aktivasi akun Tilaka:

Tilaka Name 

yes123

Tilaka Name terdiri dari 6-15 karakter, harus berupa kombinasi alfanumerik. Spesial karakter selain garis bawah “\_” tidak diperbolehkan.

Kata Sandi

••••••••

Kata sandi dan konfirmasi kata sandi tidak sama

Konfirmasi Kata Sandi

P@ss0000


Kata sandi dan konfirmasi kata sandi tidak sama

☒ Saya setuju dengan [CP/CPS](#), [Kebijakan Jaminan](#), [Kebijakan Privasi](#), dan [Perjanjian Pemilik Sertifikat](#)

AKTIVASI AKUN

POWERED BY  
 TILAKA  
NUSA TEKNOLOGI

Apabila pada halaman Aktivasi Akun terdapat field yang diisi tidak sesuai rules, maka masing-masing field akan menampilkan error validation.

8.	<p data-bbox="456 237 852 309"><b>Permohonan Aktivasi Akun Tilaka Berhasil Diajukan</b></p>  <p data-bbox="518 786 788 837">Mohon menunggu 1 x 24 jam untuk proses validasi akun.</p> <p data-bbox="486 1003 820 1032"><a href="#">Kembali ke Halaman Utama</a></p> <p data-bbox="596 1093 711 1153">POWERED BY <b>TILAKA</b> NUSA TEKNOLOGI</p>	<p data-bbox="995 188 1474 344">Merupakan halaman terakhir dari flow registrasi pengguna. Halaman ini berfungsi untuk memberikan informasi ke pengguna bahwa permohonan sertifikat elektronik sedang diproses.</p> <p data-bbox="995 367 1474 456">Halaman webview berhenti di sini dan user dapat kembali ke halaman pihak ketiga dengan klik 'Kembali ke Halaman Utama'.</p>
----	---	--

## 6. Callback Certificate Status

Endpoint yang disiapkan oleh pihak ketiga untuk memperoleh informasi hasil verifikasi /validasi registrasi sertifikat user.

### 6.1. Resource Information

Method	Purpose
Response formats	JSON
Requires authentication	3 <sup>rd</sup> party perlu Whitelist IP Tilaka Tidak perlu otorisasi dengan client id dan secret

### 6.2. Request

Method	URL
POST	3 <sup>rd</sup> party url

### 6.3. Request Body

Bentuk request dalam JSON dengan struktur sebagai berikut:

```
{
  "user_identifier": string,
  "success": boolean,
  "status": int
}
```

### 6.4. Parameters

Parameter pada request body adalah sebagai berikut:

Name	Type	Description	Required
User_identifier	string	Berupa tilaka-name/username yang diisi user saat aktivasi akun	Yes
success	boolean	Berisi true bila request sukses dan false bila gagal	Yes

Status	int	Status registrasi sertifikat	Yes
--------	-----	------------------------------	-----

Note:

- status 1 → registrasi sertifikat masih dalam proses verifikator/validator (belum final),
- status 2 → sertifikat registered (telah diterbitkan),
- status 3 → sertifikat aktif (user telah memvalidasi data pada sertifikat, atau telah melewati masa validasi 9 hari sehingga dianggap valid by system),
- status 4 → registrasi sertifikat ditolak (final) oleh verifikator/validator.

## 6.5. Response

Untuk response cukup memberikan reply HTTP status 200.

## 6.6. Samples

### 6.6.1. Sample Request

Berikut contoh request bila registrasi sertifikat masih dalam proses verifikator/validator:

```
{
  "user_identifier": "anita001",
  "success": true,
  "status": "1"
}
```

Berikut contoh request bila sertifikat sudah diterbitkan, namun user belum memvalidasi sertifikat:

```
{
  "user_identifier": "anita002",
  "success": true,
  "status": "2"
}
```

Berikut contoh request bila sertifikat sudah aktif, yaitu saat sertifikat telah terbit dan user telah memvalidasi sertifikat (atau user telah melewati masa validasi 9 hari sehingga sertifikat secara otomatis dianggap valid by system):

```
{
  "user_identifier": "anita003",
  "success": true,
  "status": "3"
}
```

Berikut contoh request bila registrasi sertifikat ditolak final oleh verifikator/validator.

```
{  
  "user_identifier": "anita004",  
  "success": true,  
  "status": "4"  
}
```

## 7. Penautan/Aktivasi Akun Webview

Webview ini digunakan saat pertama kali user pada sistem pihak ketiga akan ditautkan dengan akun user pada sistem Tilaka.

### 7.1. Resource Information

Method	Purpose
Response formats	Webview
Requires authentication	Tilaka Name dan Kata Sandi

### 7.2. Request

Method	URL
POST	https://<host>:<port>/personal-webview/link-account?request_id=<request_id yang digunakan saat CheckAkunDSExist>&setting=1&channel_id=<channel id milik client>&redirect_url=<redirect url yang di set oleh client>

### 7.3. Parameters

Parameter API saat membuka webview penautan meliputi:

Name	Type	Description	Required
request_id	string	Didapatkan dengan memanggil endpoint generate UUID.	Yes
setting	int	Parameter setting=1 digunakan untuk menampilkan halaman Informasi Sertifikat & Pengaturan Tanda Tangan & MFA setelah proses penautan berhasil	Yes
channel_id	string	Channel id client	Yes
Redirect_url	string	url yang di set oleh client, user dapat kembali ke halaman client yang diset disini apabila proses di webview tilaka sudah selesai	Optional

Parameter API pada saat redirect setelah penautan berhasil meliputi:




Name	Type	Description	Required
request-id	string	Didapatkan dengan memanggil endpoint generate UUID.	Yes
tilaka-name	string	Didapatkan ketika proses penautan akun berhasil	Yes

#### 7.4. Authorization

Tidak ada otorisasi, endpoint ini langsung diakses oleh aplikasi di pengguna.

#### 7.5. Response

Response dalam bentuk Webview dengan flow sebagai berikut:

No	Webview	Description
1.	<p><b>Aktivasi Akun</b></p>  <p>NIK Anda telah terdaftar di Tilaka. Mohon mengisi data-data berikut sebagai proses aktivasi akun Tilaka:</p> <p>Tilaka Name </p> <input type="text" value="Masukkan Tilaka Name"/> <p>Kata Sandi</p> <input type="password" value="Masukkan Kata Sandi"/> <p><a href="#">Lupa Kata Sandi</a>   <a href="#">Lupa Tilaka Name</a></p> <p><b>AKTIVASI AKUN</b></p> <p>POWERED BY </p>	<p>Untuk menautkan akun, user harus memasukkan Tilaka Name dan kata sandi pada halaman Penautan Akun.</p> <p>Pada halaman ini terdapat juga button 'Lupa Kata Sandi' dan 'Lupa Tilaka Name' yang dapat digunakan oleh user.</p> <p>Saat memanggil webview penautan akun, ada parameter yang dapat ditambahkan di url, yaitu:</p> <ul style="list-style-type: none"> <li>Setting=1, berfungsi untuk menampilkan halaman Informasi Sertifikat dan Pengaturan Tandatangani &amp; MFA ketika Tilaka Name &amp; password yang dimasukkan user benar</li> </ul>

2.

Informasi data pada sertifikat Anda:

Negara	ID
Nama	naufal danbag dev14
Organisasi	Personal
Email	naufaldanbagdev14@yopmail.com

Apabila dalam jangka waktu **sembilan hari kalender tidak ada keluhan**, maka pelanggan dianggap telah menerima bahwa semua informasi yang terdapat dalam sertifikat adalah benar.

SESUAI

[AJUKAN KOMPLAIN](#)

POWERED BY  
**TILAKA**  
NUSA TEKNOLOGI

Apabila saat pemanggilan webview penautan, dimasukkan parameter setting=1 maka untuk pengguna yang baru pertama kali login ke sistem Tilaka akan tampil halaman informasi sertifikat.

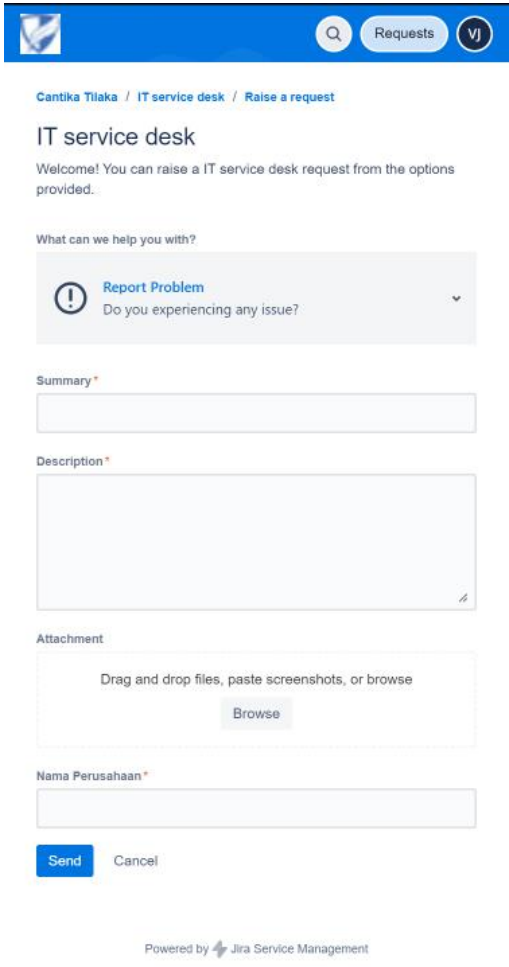
Proses approve sertifikat ini hanya akan dilakukan 1x.



Apabila dalam 9 hari kalender user tidak melakukan action (Ajukan Komplain maupun Menyetujui) maka informasi pada sertifikat secara otomatis dianggap sesuai.

- Bila informasi sertifikat sesuai, user klik button 'Sesuai' dan akan lanjut ke halaman 'Pengaturan Tanda Tangan & MFA' di halaman nomor 3
- Bila informasi sertifikat tidak sesuai, user klik button 'Ajukan Komplain' dan akan lanjut ke form Komplain di halaman nomor 4



3.	<div data-bbox="387 181 901 253"> <h3>Pengaturan Tanda Tangan &amp; MFA <span>✕</span></h3> </div> <div data-bbox="491 280 799 499"> </div> <div data-bbox="408 544 627 568"> <p>Pilih tipe tanda tangan</p> </div> <div data-bbox="408 584 877 647"> <p><i>i</i> Tanda tangan yang dipilih akan digunakan di dalam dokumen yang Anda tandatangani.</p> </div> <div data-bbox="408 672 670 745"> <p><input checked="" type="radio"/> Tanda tangan goresan</p> <p><input type="radio"/> Tanda tangan font</p> </div> <div data-bbox="411 763 874 992"> <div></div> <div></div> </div> <div data-bbox="408 1030 627 1055"> <p>Pilih metode otentikasi</p> </div> <div data-bbox="408 1070 877 1209"> <p><i>i</i> Untuk meningkatkan keamanan, diperlukan <i>Multi Factor Authentication</i> yang harus Anda gunakan saat melakukan aktivitas tandatangan digital ataupun aktivitas lainnya di tilaka.id. Silakan pilih metode MFA yang sesuai dengan kenyamanan Anda.</p> </div> <div data-bbox="408 1240 612 1314"> <p><input checked="" type="radio"/> Face Recognition</p> <p><input type="radio"/> OTP via Email</p> </div> <div data-bbox="411 1370 879 1429"> <p>LANJUT</p> </div> <div data-bbox="587 1471 699 1527"> <p>POWERED BY</p> </div>
----	---

4.		<p>Halaman Form complain tampil ketika data pada sertifikat elektronik yang diterbitkan oleh Tilaka tidak sesuai dengan data diri.</p> <p>Halaman ini akan mengirimkan informasi ke service desk Tilaka dan kembali ke halaman pihak ketiga.</p>
----	--	--

<p>5.</p>	<p><b>Aktivasi Akun Selesai</b></p>  <p>Akun Tilaka Anda telah berhasil diaktifkan.</p> <p><a href="#">Kembali ke Halaman Utama</a></p> <p>POWERED BY TILAKA NUSA TEKNOLOGI</p>	<p>Halaman ini akan muncul apabila penautan berhasil dan halaman ini akan memberikan data ke sistem pihak ketiga melalui redirect URL.</p> <p>Pada saat user klik 'Kembali ke Halaman Utama' untuk redirect ke halaman pihak ketiga maka parameter request id dan tilaka name akan dikembalikan (lihat bagian 8.3 untuk detail parameter).</p>
<p>6.</p>	<p><b>Aktivasi Akun Gagal</b></p>  <p><a href="#">Aktivasi Akun Tilaka</a></p> <p>POWERED BY TILAKA NUSA TEKNOLOGI</p>	<p>Halaman ini akan muncul saat penautan akun gagal, user dapat kembali ke halaman nomor 1 dengan klik button 'Tautkan Akun Tilaka'.</p>

## 8. Callback Registration

Merupakan endpoint yang digunakan pihak ketiga untuk memperoleh hasil KYC dari proses Registrasi user.

### 8.1. Resource Information

Method	Purpose
Response formats	JSON
Requires authentication	3 <sup>rd</sup> party perlu Whitelist IP Tilaka Tidak perlu otorisasi dengan client id dan secret

### 8.2. Request

Method	URL
POST	3 <sup>rd</sup> party url

### 8.3. Request Body

Bentuk request dalam JSON dengan struktur sebagai berikut:

```
{
  "registerId": string,
  -- dan untuk field-field lainnya dapat di refer pada response dari
    endpoint /userregstatus --
}
```

### 8.4. Parameters

Parameter pada request body adalah sebagai berikut:

Name	Type	Description	Required
registerId	string	registration id	Yes
Success	boolean	bernilai true bila request berhasil dan false bila gagal	Yes
message	string	berisi string informasi hasil proses	Yes

status	string	Status kyc	Yes
nik	boolean	bernilai true bila nik sesuai dan false bila tidak	Yes
nama	boolean	bernilai true bila nama lengkap sesuai dan false bila tidak	Yes
photo_selfie	string	format base64	Yes
fr_score	string	Nilai fr dengan range A-D	Yes
fr_score_percentage	string	Nilai fr dengan range 0-100	Yes
liveness_result	boolean	bernilai true bila valid dan false bila tidak valid	Yes
liveness_fail_message	string	berisi string informasi bila ada	Yes
summary_verification_result	boolean	bernilai true bila valid dan false bila tidak valid	Yes
tilaka_name	string	Tilaka name yang diisi oleh user pada proses registrasi	Yes
reason_code	string	Berisi angka, penjelasan dapat dilihat dibawah	Yes
date_of_birth	boolean	null	Yes
manual_registration_status	string	Berisi huruf yang melambangkan status manual registration, lihat keterangan dibawah untuk penjelasan	Yes

Nilai pada 'fr\_score' berupa huruf A-D, dan nilai pada 'fr\_score\_percentage' berupa angka dengan range 0-100. Kedua nilai ini menggambarkan hasil balikan dukcapil dengan keterangan:

fr_score	fr_score_percentage	keterangan
A	> 75.00	Lolos dukcapil
B	75.00	Lolos dukcapil
C	$75.00 < x \leq 50.00$	Tidak Lolos dukcapil
D	< 50.00	Tidak Lolos dukcapil

'status' berupa huruf dengan keterangan:

status	keterangan
S	Sukses
F	Fail dukcapil (ada data yang tidak sesuai, misal nik tidak ditemukan pada database dukcapil)
E	Error dukcapil (misal: gagal saat akan melakukan koneksi ke dukcapil)
B	User masih dalam tahap verifikasi (belum melakukan kyc)
D	User masih dalam tahap pengisian formulir (belum mengisi PIN form)

'reason\_code' berupa angka dengan keterangan:

reason_code	keterangan
0	Sukses KYC
1	Gagal dukcapil (ada data yang tidak sesuai, misal nik tidak ditemukan pada database dukcapil)
2	Liveness Gagal
3	RegisterId expired

'manual\_registration\_status' merupakan status atas proses Manual Registration user (dilakukan apabila user gagal KYC) berupa kode huruf dengan keterangan:

manual_registration_status	keterangan
P	User belum memulai proses Manual Registration
I	User dalam tahap pengisian final form Manual Registration
V	Data manual registration user masuk ke verifikator/validator Tilaka (sedang diperiksa)
S	Manual registration sukses (lolos dari verifikator/validator Tilaka)
F	Manual registration gagal (ditolak final dari

	verifikator/validator Tilaka)
E	Register id user expired

## 8.5. Response

Untuk response cukup memberikan reply HTTP status 200.

## 8.6. Samples

### 8.6.1. Sample Request

Berikut contoh request bila KYC (liveness&dukcapil) berhasil:

```
{
  "RegisterID": "8aec06d8-78ba-4136-9ecb-497a98d63529",
  "success": true,
  "message": "Berhasil mendapatkan data hasil kyc"
  "data": {
    "status": "S",
    "nik": true,
    "nama": true,
    "photo_selfie": "data:image/jpeg;base64,<base64 string of photo_selfie>",
    "fr_score": "A",
    "fr_score_percentage": "79.10",
    "liveness_result": true,
    "liveness_fail_message": "",
    "summary_verification_result": true,
    "tilaka_name": "mekJvfbL5ShP206",
    "reason_code": "0",
    "date_of_birth": null,
    "manual_registration_status": null
  }
}
```

Berikut contoh request bila KYC (liveness) gagal 3x dan user belum mulai proses manual registrasi:

```
{
  "RegisterID": "8aec06d8-78ba-4136-9ecb-497a98d63529",
  "success": true,
  "message": "Berhasil mendapatkan data hasil kyc"
  "data": {
    "status": "F",
    "nik": null,
    "nama": null,
    "photo_selfie": "data:image/jpeg;base64,<base64 string of photo_selfie>",
    "fr_score": null,
    "fr_score_percentage": null,
    "liveness_result": false,
    "liveness_fail_message": "",
    "summary_verification_result": false,
    "tilaka_name": null,
    "reason_code": "2"
    "date_of_birth": null,
    "manual_registration_status": "P"
  }
}
```

Berikut contoh request bila KYC (liveness) gagal dan proses manual registrasi sudah berhasil:

```
{
  "RegisterID": "8aec06d8-78ba-4136-9ecb-497a98d63529",
  "success": true,
  "message": "Berhasil mendapatkan data hasil kyc"
  "data": {
    "status": "F",
    "nik": null,
    "nama": null,
    "photo_selfie": "data:image/jpeg;base64,<base64 string of photo_selfie>",
    "fr_score": null,
```



```
    "fr_score_percentage": null,  
    "liveness_result": false,  
    "liveness_fail_message": "",  
    "summary_verification_result": false,  
    "tilaka_name": "mekJvfbL5ShP206",  
    "reason_code": "2"  
    "date_of_birth": null,  
    "manual_registration_status": "S"  
}
```

## 9. User Registration Result

Dengan menggunakan endpoint ini pihak ketiga dapat melihat hasil registrasi pengguna, termasuk ketika user gagal liveness/dukcapil lalu melanjutkan registrasi melalui manual registration.

### 9.1. Resource Information

Method	Purpose
Response formats	JSON
Requires authentication	Ya, menggunakan client id dan client secret

### 9.2. Request

Method	URL
POST	https://<host>:<port>/userregstatus

### 9.3. Request Body

Bentuk request dalam JSON dengan struktur sebagai berikut:

```
{
  "register_id": string
}
```

### 9.4. Parameters

Parameter pada request body adalah sebagai berikut:

Name	Type	Description	Required
Register_id	string	Merupakan parameter uuid yang digunakan saat memanggil endpoint Register User	Yes

### 9.5. Authorization

Request ini membutuhkan otorisasi OAuth2 menggunakan client credential flow yaitu menggunakan client id dan secret (lihat bagian 1.4.2 pada dokumen ini).

## 9.6. Response

Response dalam bentuk JSON dengan struktur sebagai berikut:

```
{
  "success": boolean,
  "message": string,
  "data": {
    "status": string,
    "nik": boolean,
    "nama": boolean,
    "photo_selfie": string,
    "fr_score": string,
    "fr_score_percentage": string,
    "liveness_result": boolean,
    "liveness_fail_message": string,
    "summary_verification_result": boolean,
    "tilaka_name": string,
    "reason_code ": string,
    "date_of_birth": string,
    "manual_registration_status": string
  }
}
```

Name	Type	Description
Success	Boolean	Menunjukkan kondisi response true jika verifikasi berhasil, dan false jika verifikasi tidak berhasil.
Message	String	Memberikan informasi berhasil maupun error sesuai dengan hasil verifikasi.
Data	Array	Berisi data jika diperlukan.

Nilai pada 'fr\_score' berupa huruf A-D, dan nilai pada 'fr\_score\_percentage' berupa angka dengan range 0-100. Kedua nilai ini menggambarkan hasil balikan dukcapil dengan keterangan:

fr_score	fr_score_percentage	keterangan
A	> 75.00	Lolos dukcapil
B	75.00	Lolos dukcapil
C	$75.00 < x \leq 50.00$	Tidak Lolos dukcapil
D	< 50.00	Tidak Lolos dukcapil

'status' merupakan status atas proses KYC user, berupa huruf dengan keterangan:

status	keterangan
S	Sukses KYC
F	Fail dukcapil (ada data yang tidak sesuai, misal nik tidak ditemukan pada database dukcapil)
E	Error dukcapil (misal: gagal saat akan melakukan koneksi ke dukcapil)
B	User masih dalam tahap verifikasi (belum melakukan kyc)
D	User masih dalam tahap pengisian formular (belum mengisi PIN form)

'reason\_code' berupa angka dengan keterangan:

reason_code	keterangan
0	Sukses KYC
1	Gagal dukcapil (ada data yang tidak sesuai, misal nik tidak ditemukan pada database dukcapil)
2	Liveness Gagal
3	RegisterId expired

'manual\_registration\_status' merupakan status atas proses Manual Registration user (dilakukan apabila user gagal KYC) berupa kode huruf dengan keterangan:

manual_registration_status	keterangan
P	User belum memulai proses Manual Registration
I	User dalam tahap pengisian final form Manual Registration
V	Data manual registration user masuk ke verifikator/validator Tilaka (sedang diperiksa)
S	Manual registration sukses (lolos dari verifikator/validator Tilaka)
F	Manual registration gagal (ditolak final dari verifikator/validator Tilaka)
E	Register id user expired

## 9.7. Samples

### 9.7.1. Sample Request

Berikut contoh request:

```
{
  "register_id": "8aec06d8-78ba-4136-9ecb-497a98d63529"
}
```

### 9.7.2. Sample Response

Jika kyc telah dilakukan dan berhasil verifikasi (liveness & dukcapil berhasil), maka akan memberikan respon 200 dengan data status S seperti pada contoh berikut:

```
{
  "success": true,
  "message": "Berhasil mendapatkan data hasil kyc",
  "data": {
    "status": "S",
    "nik": true,
    "nama": true,
  }
}
```

```

    "photo_selfie": data:image/jpeg;base64,<base64 string of
        photo_selfie>",
    "fr_score": "A",
    "fr_score_percentage": "79.10",
    "liveness_result": true,
    "liveness_fail_message": "",
    "summary_verification_result": true,
    "tilaka_name": "anita001",
    "reason_code": "0",
    "date_of_birth": null,
    "manual_registration_status": null
}

```

Jika gagal liveness hingga 3x, dan user belum melakukan registrasi manual maka akan memberikan respon 200 dengan data status F, manual\_registration\_status P dan tilaka\_name null seperti pada contoh berikut:

```

{
    "success": true,
    "message": "Berhasil mendapatkan data hasil kyc",
    "data": {
        "status": "F",
        "nik": null,
        "nama": null,
        "photo_selfie": data:image/jpeg;base64,<base64 string of
            photo_selfie>",
        "fr_score": null,
        "fr_score_percentage": null,
        "liveness_result": false,
        "liveness_fail_message": "",
        "summary_verification_result": false,
        "tilaka_name": null,
        "reason_code": "2",
        "date_of_birth": null,
        "manual_registration_status": "P"
    }
}

```

```

    }
}

```

Jika liveness berhasil namun gagal dukcapil (misal dikarenakan nik yang dipakai tidak ditemukan pada database dukcapil) & user sukses melakukan registrasi manual, maka akan memberikan respon 200 dengan data 'status' F, manual\_registration\_status S seperti pada contoh berikut:

```

{
  "success": true,
  "message": "Berhasil mendapatkan data hasil kyc",
  "data": {
    "status": "F",
    "nik": false,
    "nama": null,
    "photo_selfie": data:image/jpeg;base64,<base64 string of
      photo_selfie>",
    "fr_score": null,
    "fr_score_percentage": null,
    "liveness_result": true,
    "liveness_fail_message": "",
    "summary_verification_result": false,
    "tilaka_name": "anita001",
    "reason_code": "1",
    "date_of_birth": null,
    "manual_registration_status": "S"
  }
}

```

Jika liveness berhasil namun terjadi error dukcapil & user belum melakukan registrasi manual, maka akan memberikan respon 200 dengan data 'status' E, manual\_registration\_status P dan tilaka\_name null.

## 10. Change MFA Webview

Dengan menggunakan webview ini, user yang telah aktif dapat merubah setting MFA nya dari OTP menjadi FR maupun sebaliknya. Webview ini dapat langsung dipanggil oleh client.

### 10.1. Resource Information

Method	Purpose
Response formats	Webview
Requires authentication	Tidak ada

### 10.2. Request

Method	URL
GET	https://<host>:<port>/personal-webview/login?setting=2&tilaka_name=<tilakaname user>&channel_id=<channel_id>&redirect_url=<url yang di set client>

### 10.3. Parameters

Parameter pada request body adalah sebagai berikut:

Name	Type	Description	Required
setting	Int	setting=2 digunakan untuk memanggil webview change MFA	Yes
tilaka_name	String	Tilakaname milik user yang akan melakukan change MFA	Yes
channel_id	string	Channel id client	Yes
Redirect_url	string	url yang di set oleh client, user dapat kembali ke halaman client yang diset disini apabila proses di webview tilaka sudah selesai	Optional

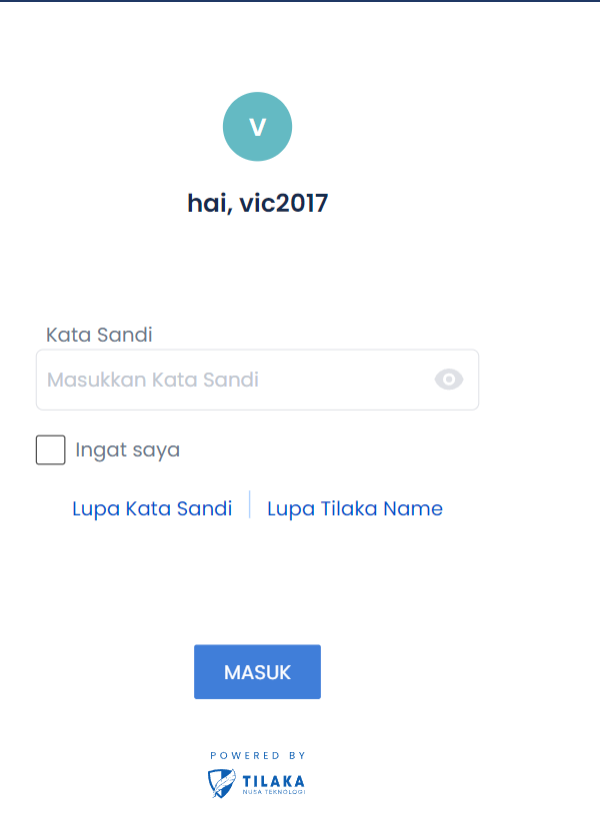


## 10.4. Authorization

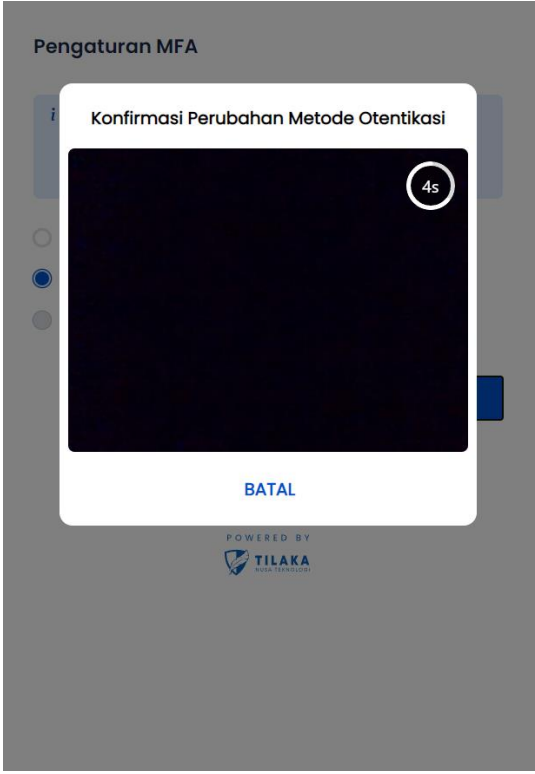
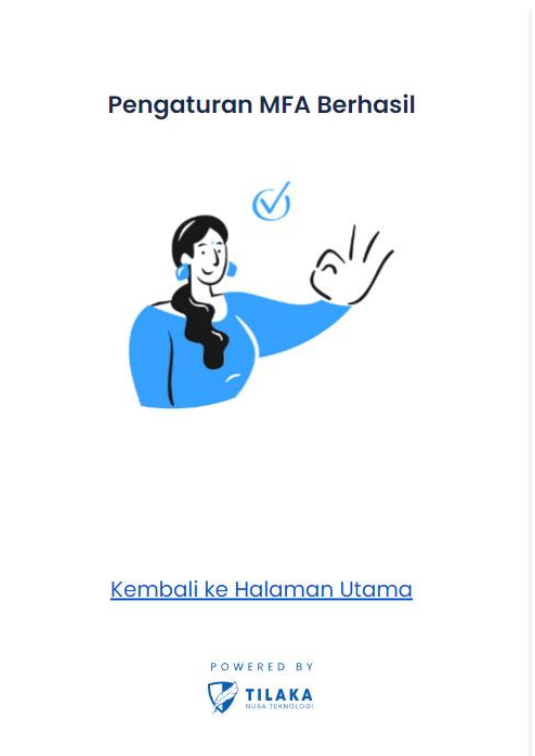
Tidak ada otorisasi, endpoint ini langsung diakses oleh aplikasi di pengguna.

## 10.5. Response

Berikut adalah webview yang menjadi respon dari change MFA Webview.

No	Webview	Description
1.		Merupakan halaman pertama yang akan muncul, pada halaman ini user perlu memasukkan kata sandi nya untuk login.

<p>2.</p>	<p><b>Pengaturan MFA</b></p> <p><i>i</i> Untuk meningkatkan keamanan, diperlukan <i>Multi Factor Authentication</i> yang harus Anda gunakan saat melakukan aktivitas tandatangan digital ataupun aktivitas lainnya di tilaka.id. Silakan pilih metode MFA yang sesuai dengan kenyamanan Anda.</p> <p> <input type="radio"/> Face Recognition  <input checked="" type="radio"/> OTP via Email  <input type="radio"/> OTP via Ponsel <span>Belum Tersedia</span> </p> <p><b>Ubah Metode Otentikasi</b></p> <p>Batal</p> <p>POWERED BY <b>TILAKA</b> NUSA TEKNOLOGI</p> <p><b>Pengaturan MFA</b></p> <p><i>i</i> Untuk meningkatkan keamanan, diperlukan <i>Multi Factor Authentication</i> yang harus Anda gunakan saat melakukan aktivitas tandatangan digital ataupun aktivitas lainnya di tilaka.id. Silakan pilih metode MFA yang sesuai dengan kenyamanan Anda.</p> <p><b>Konfirmasi Perubahan Metode Otentikasi</b></p> <p>Apa Anda yakin mengubah metode otentikasi menjadi Face Recognition</p> <p><b>Konfirmasi</b></p> <p>Batal</p> <p>POWERED BY <b>TILAKA</b> NUSA TEKNOLOGI</p>	<p>Apabila setting awal adalah OTP via email &amp; user ingin mengganti ke FR, maka akan tampil halaman konfirmasi</p>
-----------	--	--

<p>3.</p>		<p>Apabila setting awal adalah FR &amp; user ingin mengganti ke OTP via Email, maka akan tampil halaman FR dengan counter 5 detik.</p>
<p>4.</p>		<p>Apabila sukses melakukan penggantian, halaman ini akan tampil. User dapat kembali ke halaman client dengan klik 'Kembali ke Halaman Utama'</p>

## 11. Change Set Signature Webview

Dengan menggunakan webview ini, user yang telah aktif dapat merubah setting tampilan tanda tangan nya dari Goresan menjadi Font, atau mengganti jenis Font nya maupun sebaliknya. Webview ini dapat langsung dipanggil oleh client.

### 11.1. Resource Information

Method	Purpose
Response formats	Webview
Requires authentication	Tidak ada

### 11.2. Request

Method	URL
GET	https://<host>:<port>/personal-webview/login?setting=3&tilaka_name=<tilakaname user>&channel_id=<channel_id>&redirect_url=<url yang di set client>

### 11.3. Parameters

Parameter pada request body adalah sebagai berikut:

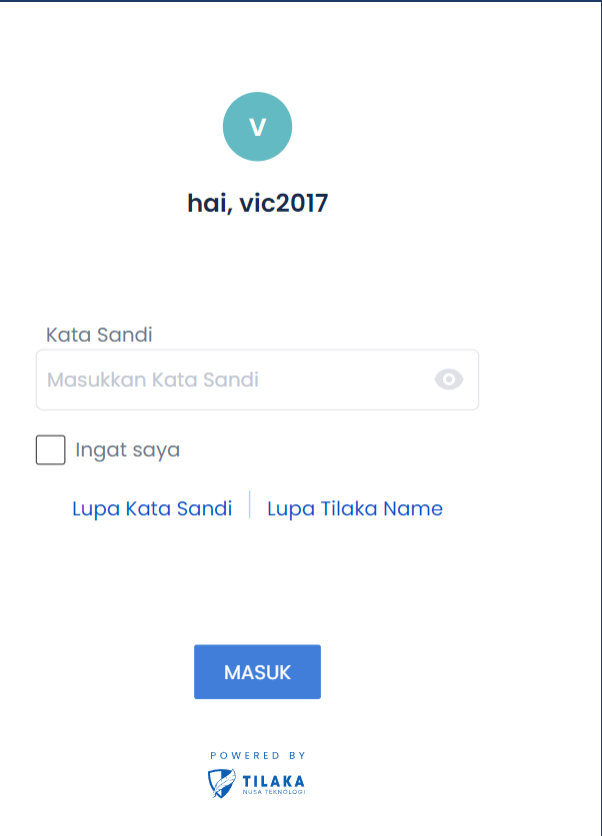
Name	Type	Description	Required
setting	Int	setting=3 digunakan untuk memanggil webview Set Signature	Yes
tilaka_name	String	Tilakaname milik user yang akan melakukan Set Signature	Yes
channel_id	string	Channel id client	Yes
Redirect_url	string	url yang di set oleh client, user dapat kembali ke halaman client yang diset disini apabila proses di webview tilaka sudah selesai	Optional



## 11.4. Authorization

Tidak ada otorisasi, endpoint ini langsung diakses oleh aplikasi di pengguna.

## 11.5. Response

Berikut adalah webview yang menjadi respon dari set Signature Webview.

No	Webview	Description
1.		Merupakan halaman pertama yang akan muncul, pada halaman ini user perlu memasukkan kata sandi nya untuk login.

2.	<p><b>Pengaturan Tanda Tangan</b></p>  <p>Pilih tipe tanda tangan</p> <p><i>i</i> Tanda tangan yang dipilih akan digunakan di dalam dokumen yang Anda tandatangani.</p> <p><input type="radio"/> Tanda tangan goresan</p> <p><input checked="" type="radio"/> Tanda tangan font</p> <div> <div>de 2017</div> <div>de 2017</div> <div>de 2017</div> <div>de 2017</div> <div>de 2017</div> <div>de 2017</div> </div> <p><b>Lanjut</b></p> <p>POWERED BY <b>TILAKA</b> NUSA TEKNOLOGI</p>	<p>Kemudian user dapat mengatur pilihan tampilan tanda tangan yang diinginkan pada halaman ini</p>
3.	<p><b>Pengaturan Tanda Tangan Berhasil</b></p>  <p><a href="#">Kembali ke Halaman Utama</a></p> <p>POWERED BY <b>TILAKA</b> NUSA TEKNOLOGI</p>	<p>Apabila sukses melakukan penggantian, halaman ini akan tampil. User dapat kembali ke halaman client dengan klik 'Kembali ke Halaman Utama'</p>

## 12. Request Revoke Certificate

Merupakan endpoint yang dipanggil saat ada user pihak ketiga yang ingin melakukan revoke certificate. Memberikan response data token dan url yang berikutnya digunakan untuk memanggil webview Liveness untuk Revoke.

User memiliki 3x kesempatan liveness, apabila gagal liveness 3x maka proses Revoke akan gagal. Ketika revoke gagal, user dapat mengajukan kembali request Revoke (dalam 1 hari, user hanya dapat melakukan pengajuan revoke maksimal sebanyak 3 kali).

### 12.1. Resource Information

Method	Purpose
Response formats	JSON
Requires authentication	Ya, menggunakan client id dan client secret

### 12.2. Request

Method	URL
POST	https://<host>:<port>/requestRevokeCertificate

### 12.3. Request Body

Bentuk request dalam JSON dengan struktur sebagai berikut:

```
{
  "user_identifier": string,
  "reason": string
}
```

### 12.4. Parameters

Parameter pada request body adalah sebagai berikut:

Name	Type	Description	Required
------	------	-------------	----------

user_identifier	string	Data tilaka name	Yes
reason	string	Alasan revoke	Yes

Berikut adalah list alasan Revoke yang dapat digunakan:

Revoke Reason
Resign
PHK
Habis Kontrak
Mutasi
Pemindahan Departemen
Pindah Divisi
Internal Fraud
Penutupan Hak Akses
Pelanggaran Hukum dari User
Perangkat Hilang, Perangkat Dicuri

## 12.5. Authorization

Request ini membutuhkan otorisasi OAuth2 menggunakan client credential flow yaitu menggunakan client id dan secret (lihat bagian 1.4.2 pada dokumen ini).

## 12.6. Response

Response dalam bentuk JSON dengan struktur sebagai berikut:

```
{
  "success": boolean,
  "message": string,
  "data": []
}
```



Name	Type	Description
Success	boolean	Bernilai true menunjukan request berhasil dan false jika terjadi permasalahan.
Message	string	Berisi keterangan informasi proses (ex: request revoke berhasil/gagal).
Data	array	Berisi token dan url jika request berhasil, dan null jika gagal.

## 12.7. Samples

### 12.7.1. Sample Request

Berikut contoh request:

```
{
  "user_identifier": "anita001",
  "reason": "Resign"
}
```

### 12.7.2. Sample Response

Jika user\_identifier yang dimasukkan pada parameter url adalah:

- user berstatus 0 (belum memiliki sertifikat yang sedang dalam proses)
  - user berstatus 2 (sertifikat telah terbit, namun user belum lakukan approval sertifikat)
- maka akan menerima response dengan status 200 dan data JSON sebagai berikut:

```
{
  "success": false,
  "message": " request revoke sertifikat gagal. user_identifier anita001 tidak memiliki sertifikat yang aktif",
  "data": null
}
```

Jika user\_identifier yang dimasukkan pada parameter url adalah:

- user berstatus 1 (sertifikat user masih dalam proses verifikasi)
  - user berstatus 4 (ditolak final oleh verifikator/validator)
  - user bukan berasal dari channel yang sesuai
- maka akan menerima response dengan status 200 dan data JSON sebagai berikut:

```
{
  "success": false,
```

```

    "message": " request revoke sertifikat gagal. user_identifier
                anita001 tidak valid",

    "data": null

}

```

Jika user\_identifier yang dimasukkan pada parameter url ingin melakukan percobaan revoke >3x di hari yang sama, maka akan menerima response dengan status 200 dan data JSON sebagai berikut:

```

{

    "success": false,

    "message": " request revoke sertifikat gagal. Silahkan coba lagi besok",

    "data": null

}

```

Jika berhasil, akan menerima response dengan status 200 dan data JSON sebagai berikut:

```

{

    "success": true,

    "message": "request revoke sertifikat berhasil",

    "data": [

        "rev7d7d8a1a-709f-4be0-9b32-4136169bed7e",

        "https://sb-api.tilaka.id/personal-webview/kyc/revoke?
            revoke_id=rev7d7d8a1a-709f-4be0-9b32-4136169bed7e"



    ]

}

```

Ketika url yang didapat dari response di load, maka akan menampilkan halaman webview liveness berikut:

No	Webview	Description
1.	<p><b>Liveness</b></p> <p>Pastikan wajah di dalam garis panduan dan ikuti petunjuk dengan benar.</p> 	<p>Merupakan halaman webview yang pertama muncul dalam proses Revoke Sertifikat.</p> <p>Dalam proses liveness, user akan diminta melakukan gerakan sesuai dengan panduan. Gerakan ini akan di capture oleh Liveness Engine.</p>
2.		<p>Apabila liveness untuk revoke berhasil, maka akan mendapatkan response:</p> <p>redirect_url, status=Sukses, revoke id dan user_identifier</p> <p>Disamping adalah contoh tampilan halaman bila redirect url diarahkan pada laman <a href="https://www.google.com/">https://www.google.com/</a></p> <p>Halaman redirect url yang ingin ditampilkan ke user dapat diatur oleh pihak ketiga</p>

3.	<p style="text-align: center;"><b>Liveness Gagal</b></p>  <p style="text-align: center;">Maaf, proses Liveness Anda gagal. Foto dan aksi yang diminta tidak sesuai. Mohon ulangi proses Liveness dan ikuti petunjuk dengan benar.</p> <p style="text-align: center;"><b>Ulangi</b></p> <p style="text-align: center;">POWERED BY <b>TILAKA</b> NUSA TEKNOLOGI</p>	<p>Apabila liveness gagal dengan percobaan &lt;3x maka user akan diarahkan ke halaman webview ini.</p> <p>Pengguna dapat melakukan Liveness kembali menggunakan button 'ulangi' untuk kembali ke halaman nomor 1.</p>
4.		<p>Apabila liveness untuk revoke gagal dengan percobaan hingga 3x, maka akan mendapatkan response:</p> <p>Redirect_url, status=Gagal, revoke id dan user_idenfier</p> <p>Disamping adalah contoh tampilan halaman bila redirect url diarahkan pada laman <a href="https://www.google.com/">https://www.google.com/</a></p> <p>Halaman redirect url yang ingin ditampilkan ke user dapat diatur oleh pihak ketiga</p>

## 13. Check Certificate Status

Merupakan endpoint yang dipanggil saat ada user pihak ketiga ketika ingin mengambil data status sertifikat user.

### 13.1. Resource Information

Method	Purpose
Response formats	JSON
Requires authentication	Ya, menggunakan client id dan client secret

### 13.2. Request

Method	URL
POST	https://<host>:<port>/checkcertstatus

### 13.3. Request Body

Bentuk request dalam JSON dengan struktur sebagai berikut:

```
{
  "user_identifier": string
}
```

### 13.4. Parameters

Parameter pada request body adalah sebagai berikut:

Name	Type	Description	Required
user_identifier	string	Data tilaka name	Yes

### 13.5. Authorization

Request ini membutuhkan otorisasi OAuth2 menggunakan client credential flow yaitu menggunakan client id dan secret (lihat bagian 1.4.2 pada dokumen ini).

### 13.6. Response

Response dalam bentuk JSON dengan struktur sebagai berikut:

```
{
  "success": boolean,
  "status": int,
  "message": string,
  "data": []
}
```

Name	Type	Description
Success	boolean	Bernilai true menunjukan request berhasil dan false jika terjadi permasalahan.
Status	int	Berisi angka yang menggambarkan status sertifikat user
Message	string	Berisi informasi status sertifikat
Data	array	Berisi data sertifikat yang butuh approval + jika ada sertifikat sebelumnya, yang revoked atau expired

Note:

- Status 0= tidak ada sertifikat atas user\_identifier tersebut
- status 1= registrasi sertifikat masih dalam proses verifikator/validator
- status 2= sertifikat registered (telah terbit) & perlu konfirmasi user
- status 3 = sertifikat aktif (telah dikonfirmasi user)
- status 4 = registrasi sertifikat ditolak (final) oleh verifikator/validator.

### 13.7. Samples

#### 13.7.1. Sample Request

Berikut contoh request:

```
{
  "user_identifier": "anita001"
}
```

### 13.7.2. Sample Response

Jika user\_identifier yang dimasukkan salah (tidak terdaftar) akan menerima response dengan status 200 dan data JSON sebagai berikut:

```
{
  "success": false,
  "status": 0,
  "message": {
    "info": "Gagal cek status sertifikat. User Identifier tidak ditemukan",
    "name": null,
    "email": null,
    "company": null,
    "country": null,
    "serialnumber": null
  },
  "data": null
}
```

Jika user\_identifier yang dimasukkan milik user yang sertifikat nya masih enroll, telah di revoke atau telah expired, maka akan menerima response dengan status 200 dan data JSON sebagai berikut:

```
{
  "success": true,
  "status": 0,
  "message": {
    "info": "Belum memiliki sertifikat",
    "name": "",
    "email": "",
    "company": "",
    "country": "",
    "serialnumber": ""
  }
}
```

```

},
"data": [
  {
    "status": "Revoke",
    "serialnumber": "48AE1C837CEE3A44D246C5E5C2A0ECF56C4B5F80",
    "subject_dn": "CN=anita00,OU=Personal,C=ID,dnQualifier=usercb0279e-96cf-46a1-884b-a4c15dbc6d98",
    "start_active_date": "2022-08-10 02:55:24",
    "expiry_date": "2023-08-10 02:55:23",
    "certificate": "<string of certificate>",
  }
]
}

```

Jika sertifikat sedang dalam proses verifikasi, akan menerima response dengan status 200 dan data JSON sebagai berikut:

```

{
  "success": true,
  "status": 1,
  "message": {
    "info": "Proses permohonan sertifikat dalam proses",
    "name": null,
    "email": null,
    "company": null,
    "country": null,
    "serialnumber": null
  },
  "data": null
}

```

Jika sertifikat telah diterbitkan, namun belum di konfirmasi oleh user maka akan menerima response dengan status 200 dan data JSON:

```

{
  "success": true,
  "status": 2,

```



```

"message": {
  "info": "Ada sertifikat yang butuh approval",
  "name": "anita",
  "email": "anita@mail.com",
  "company": "Personal",
  "country": "ID",
  "serialnumber": "48AE1C837CEE3A44D246C5E5C2A0ECF56C4B5F57",
}
"data": [
  {
    "status": "Registered",
    "serialnumber": "48AE1C837CEE3A44D246C5E5C2A0ECF56C4B5F57",
    "subject_dn": "CN=anita00,OU=Personal,C=ID,dnQualifier=usercb0279e-96cf-46a1-884b-a4c15dbc6d98",
    "start_active_date": "2022-08-10 02:55:24",
    "expiry_date": "2023-08-10 02:55:23",
    "certificate": "<string of certificate>",
  }
]
}

```

Jika user telah melakukan konfirmasi (approve certificate), maka status akan berubah menjadi '3' dengan message-info: **aktif** dan data-status: **aktif**.

```

{
  "success": true,
  "status": 3,
  "message": {
    "info": "Aktif",
    "name": "",
    "email": "",
    "company": "",
    "country": "",
    "serialnumber": "",
  }
}

```

```

    "data": [
      {
        "status": "Aktif",
        "serialnumber": "48AE1C837CEE3A44D246C5E5C2A0ECF56C4B5F57",
        "subject_dn": "CN=anita00,OU=Personal,C=ID,dnQualifier=usercb0279e-96cf-46a1-884b-a4c15dbc6d98",
        "start_active_date": "2022-08-10 02:55:24",
        "expiry_date": "2023-08-10 02:55:23",
        "certificate": "<string of certificate>",
      }
    ]
  }
}

```

Jika pengajuan sertifikat user ditolak final baik oleh verifikator maupun validator, maka status akan berubah menjadi '4' dengan message-info: alasan tolak yang diinput oleh verifikator atau validator.

```

{
  "success": true,
  "status": 4,
  "message": {
    "info": "KTP image is not valid",
    "name": null,
    "email": null,
    "company": null,
    "country": null,
    "serialnumber": null
  },
  "data": null
}

```

## 14. Status Codes

The API Personal menggunakan kode status HTTP di bawah ini. 2XX – Success; 4XX - Error in client; 5XX - Error in server.

Status Code	Description
200	OK
201	Created
202	Accepted (Request accepted, and queued for execution)
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict
412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

## 15. Appendix - Endpoint & Webview URL

Di bawah ini adalah list endpoint & Webview URL yang dapat digunakan oleh client untuk environment Sandbox dan Production.

Name	Sandbox Endpoint	Production Endpoint
Generate UUID	<a href="https://sb-api.tilaka.id/generateUUID">https://sb-api.tilaka.id/generateUUID</a>	<a href="https://api.tilaka.id/generateUUID">https://api.tilaka.id/generateUUID</a>
Pengecekan Akun	<a href="https://sb-api.tilaka.id/checkAkunDSExist">https://sb-api.tilaka.id/checkAkunDSExist</a>	<a href="https://api.tilaka.id/checkAkunDSExist">https://api.tilaka.id/checkAkunDSExist</a>
Registrasi Akun	<a href="https://sb-api.tilaka.id/registerForKycCheck">https://sb-api.tilaka.id/registerForKycCheck</a>	<a href="https://api.tilaka.id/registerForKycCheck">https://api.tilaka.id/registerForKycCheck</a>
Webview Registration	<a href="https://sb-api.tilaka.id/personal-webview/guide?request_id=&lt;uuid&gt;">https://sb-api.tilaka.id/personal-webview/guide?request_id=&lt;uuid&gt;</a>	<a href="https://api.tilaka.id/personal-webview/guide?request_id=&lt;uuid&gt;">https://api.tilaka.id/personal-webview/guide?request_id=&lt;uuid&gt;</a>
Webview Penautan	<a href="https://sb-api.tilaka.id/personal-webview/link-account?request_id=&lt;request_id yang digunakan saat CheckAkunDSExist&gt;&amp;setting=1">https://sb-api.tilaka.id/personal-webview/link-account?request_id=&lt;request_id yang digunakan saat CheckAkunDSExist&gt;&amp;setting=1</a>	<a href="https://api.tilaka.id/personal-webview/link-account?request_id=&lt;request_id yang digunakan saat CheckAkunDSExist&gt;&amp;setting=1">https://api.tilaka.id/personal-webview/link-account?request_id=&lt;request_id yang digunakan saat CheckAkunDSExist&gt;&amp;setting=1</a>