



# PANDUAN API

## APLIKASI AJO – ABG

Version 1.3

DIVISI SISTEM & TEKNOLOGI INFORMASI

PT PLN (PERSERO) KANTOR PUSAT

© 2019

[www.pln.co.id](http://www.pln.co.id)

# DAFTAR ISI

DAFTAR ISI.....	1
BAB I PENDAHULUAN.....	3
1.1. Teknologi API.....	3
1.2. Format Request dan Respon .....	3
1.3. Client Server .....	3
1.4 Authentication.....	4
BAB II WORKFLOW .....	5
2.1 Mendapatkan Access Token .....	5
2.2 Input Data Jaminan .....	5
2.3 Mendapatkan User dari Access Token .....	6
BAB III API ACCESS TOKEN .....	7
3.1 API URL .....	7
3.2 HTTP Request.....	7
3.3 Request Parameter .....	7
3.4 Contoh Skrip .....	7
3.5 API Response : Access Token .....	10
BAB IV INPUT DATA JAMINAN .....	11
4.1 API URL .....	11
4.2 HTTP Request.....	11
4.3 Request Parameter .....	12
4.4 Format File PDF .....	13
4.5 Contoh Skrip .....	14
4.6 API Response.....	19
BAB V INPUT DATA VALIDITY .....	23
5.1 API URL .....	23
5.2 HTTP Request.....	23
5.3 Request Parameter .....	24
5.4 Format File PDF .....	24

5.5 Contoh Skrip .....	25
5.6 API Response .....	31

# **BAB I PENDAHULUAN**

Application Programming Interface (API) ini digunakan oleh client server untuk mengirimkan data Jaminan dari client server AJO ke server ABG.

## **1.1. Teknologi API**

Berikut adalah teknologi yang digunakan dalam pengoperasian API antara server ABG dengan server client:

1. HTTP Request POST dan GET
2. Pengamanan API menggunakan protokol OAuth2
3. Authorization Grant : Resource Owner Password credentials (password)

## **1.2. Format Request dan Respon**

Format dari request yang dikirimkan adalah HTTP POST dan GET. Sedangkan response yang dihasilkan adalah menggunakan format JSON

## **1.3. Client Server**

Applikasi client yang sudah didaftarkan untuk API ini adalah 12 Bank yang terdaftar.

NO	PENERBIT JAMINAN	PERJANJIAN KERJASAMA PENERBIT JAMINAN	
1	PT Bank Rakyat Indonesia (Persero), Tbk.	1490.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
2	PT Bank Negara Indonesia (Persero), Tbk.	1491.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
3	PT Bank Mandiri Indonesia (Persero), Tbk.	1492.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
4	PT Bank Bukopin Tbk.	1493.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
5	PT Bank Central Asia Tbk.	1494.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
6	PT Bank Danamon Indonesia (Persero), Tbk.	1495.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
7	PT Bank CIMB Niaga Tbk.	1496.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
8	PT Bank Rakyat Indonesia Syariah, Tbk.	1497.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
9	PT Bank OCBC NISP Tbk.	1498.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
10	PT Bank Permata Tbk.	1499.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
11	PT Bank UOB Indonesia Tbk.	1500.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018
12	PT Bank MUFG Bank, Ltd., Cabang Jakarta	1501.PJ/KEU.00.03/KDIVTRE/2018	14 Desember 2018

## 1.4 Authentication

Agar dapat menggunakan API ABG, dibutuhkan informasi tentang client yang terdiri dari:

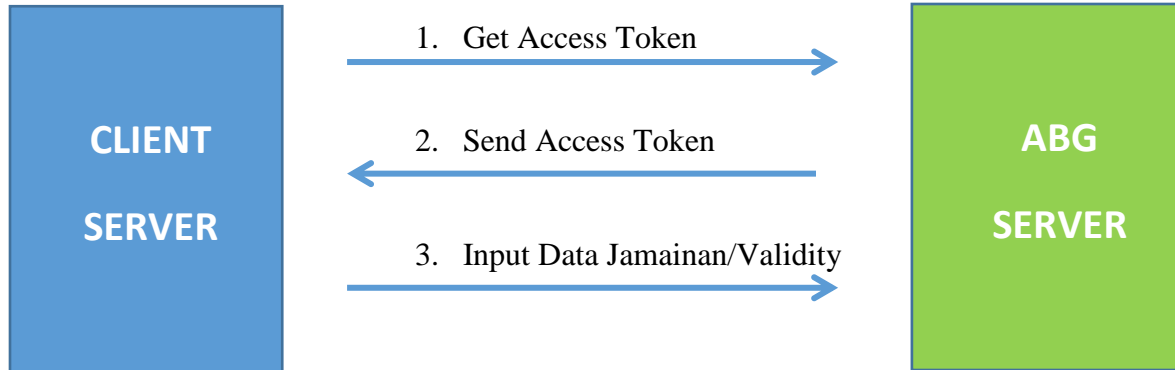
1. **username** : berisi tentang username API
2. **password** : password username
3. **client\_id** : id client yang sudah didaftarkan ke server API ABG
4. **client\_secret** : kode unik yang didapatkan dari server API ABG untuk kebutuhan interface

Adapun username dan password untuk seluruh komunikasi API yang dapat digunakan oleh PT Bank Negara Indonesia, Tbk adalah sebagai berikut:

- |                         |                   |
|-------------------------|-------------------|
| 1. <b>username</b>      | : <username>      |
| 2. <b>password</b>      | : <password>      |
| 3. <b>client_id</b>     | : <client_id>     |
| 4. <b>client_secret</b> | : <client_secret> |

## **BAB II WORKFLOW**

Berikut ini adalah proses mekanisme integrasi aplikasi client dengan aplikasi ABG.



### **2.1 Mendapatkan Access Token**

Client Server melakukan request ke ABG Server untuk mendapatkan Access Token dengan mengirimkan parameter yang sudah ditentukan. Access Token digunakan untuk mendapatkan identitas atau untuk melakukan input data ke server ABG.

Setelah client server mengirimkan request Access Token, server ABG akan melakukan proses otentikasi terhadap request pengguna API apakah valid atau tidak. Jika request valid maka server ABG akan mengirimkan Access Token.

### **2.2 Input Data Jaminan**

Access token yang sudah didapatkan dari Server ABG digunakan sebagai pass code untuk melakukan input data melalui API. Jangka waktu satu Access Token adalah 1 jam. Jika sudah melewati 1 Jam, Access Token akan expired (sudah tidak dapat digunakan kembali) dan harus melakukan request Access Token kembali untuk memperbarui Access Token.

## 2.3 Mendapatkan User dari Access Token

Selain untuk menginputkan data, Access Token dapat digunakan untuk mendapatkan informasi user API.

## **BAB III API ACCESS TOKEN**

API ini digunakan untuk mendapatkan access token dari server aplikasi ABG

### **3.1 API URL**

API URL yang digunakan untuk mendapatkan access token adalah:

- a. Development : <https://devapi.abg.pln.co.id/oauth/token>
- b. Production : <https://api.abg.pln.co.id/oauth/token>

### **3.2 HTTP Request**

HTTP Request yang digunakan adalah POST.

### **3.3 Request Parameter**

Parameter yang dikirimkan saat melakukan request access token adalah sebagai berikut:

- a. **grant\_type** : **password**
- b. **client\_id** : id client server yang sudah didaftarkan
- c. **client\_secret** : kode unik yang didapatkan dari server ABG
- d. **username** : username aplikasi client
- e. **password** : password aplikasi client

### **3.4 Contoh Skrip**

Berikut ini adalah contoh skrip PHP yang dijalankan di client server untuk mengirimkan request access token:



#### a. PHP

```
$url = 'https:// devapi.abg.pln.co.id /oauth/access_token';

$client_id = '1'; // masukkan client_id
$client_secret = 'bexF9zPWCdnDFzcosN7x'; // masukkan
client_secret $client_username = 'username'; // username client
$client_password = 'password'; // password client

// form data
$form_data = [
    'grant_type' => 'password',
    'client_id' => $client_id,
    'client_secret' => $client_secret,
    'username' => $client_username,
    'password' => $client_password,
];

//get access token
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_HEADER, false);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $form_data);
$tmp = curl_exec ($ch);
curl_close ($ch);
echo $tmp;
```

## b. Laravel

```
use GuzzleHttp\Client;

public function index()
{
    $client_id = '2';
    $client_secret = '3HQufrAUcd9YAk2eI9pbsfrNYDsKxhBx8043m2TQ';
    $client_username = 'admin@a.com';
    $client_password = '1234';

    $url = 'https://devapi.abg.pln.co.id/oauth/access_token';

    $http = new Client();

    $response = $http->post($url, [
        'form_params' => [
            'grant_type' => 'password',
            'client_id' => $client_id,
            'client_secret' => $client_secret,
            'username' => $client_username,
            'password' => $client_password,
            'scope' => ''
        ],
    ]);

    // retrieve access token
    $data = (array) json_decode((string) $response->getBody(), true);

    $token = $data['access_token'];
}
```

### c. Postman

The screenshot shows a Postman interface for a POST request to `http://localhost:8000/oauth/token`. The 'Body' tab is selected, and the 'form-data' radio button is chosen. The body contains the following key-value pairs:

Key	Value
<input checked="" type="checkbox"/> grant_type	password
<input checked="" type="checkbox"/> client_id	2
<input checked="" type="checkbox"/> client_secret	3HQufnAUcd9YAk2el9pbsfrNYDsKxhBx8043m2TQ
<input checked="" type="checkbox"/> username	bni
<input checked="" type="checkbox"/> password	bni@123

## 3.5 API Response : Access Token

Setelah memproses request permohonan access token yang datang dari client server, maka server ABG akan mengirimkan response dalam bentuk JSON sebagai berikut:

Code	Response	Keterangan
<b>200 OK</b>	<pre>{   "token_type": "Bearer",   "expires_in": 3600,   "access_token": "XXXXXXXXXXXXXXXXXXXXXXXXX ",   "refresh_token": "XXXXXXXd55f365d1b9b61babd" }</pre>	Sukses
<b>400 Bad Request</b>	<pre>{   "error": "unsupported_grant_type",   "message": "The authorization grant type is not supported by the authorization server.",   "hint": "Check the `grant_type` parameter" }</pre>	Grant Type tidak sesuai
<b>401 Unauthorized</b>	<pre>{   "error": "invalid_credentials",   "message": "The user credentials were incorrect." }</pre>	Password Salah
<b>401 Unauthorized</b>	<pre>{   "error": "invalid_client",   "message": "Client authentication failed" }</pre>	Client ID Salah

## **BAB IV INPUT DATA JAMINAN**

API ini digunakan untuk memasukkan data Jaminan

### **4.1 API URL**

API URL yang digunakan untuk input jaminan adalah:

- a. Development :
  - 1. Multipart/form-data : <https://devapi.abg.pln.co.id/jaminan/store>
  - 2. Json : <https://devapi.abg.pln.co.id/jaminan/json/store>
- b. Production : <https://api.abg.pln.co.id/jaminan/store>
  - 1. Multipart/form-data : <https://api.abg.pln.co.id/jaminan/store>
  - 2. Json : <https://api.abg.pln.co.id/jaminan/json/store>

### **4.2 HTTP Request**

HTTP Request yang digunakan adalah POST.

Metode pengiriman data, terdapat 2 yaitu :

- a. Multipart/form-data  
Pada Bagian Header ditambahkan
  - 1. Accept : application/json
  - 2. Authorization : Bearer <<access\_token>>
- b. JSON, dengan encrypt file PDF menggunakan base64  
Pada Bagian Header ditambahkan
  - 1. Accept : application/json
  - 2. Authorization : Bearer <<access\_token>>
  - 3. Content-type : application/json

### 4.3 Request Parameter

Parameter yang dikirimkan saat melakukan input data jaminan adalah sebagai berikut:

NO	NAMA KOLOM	VALIDASI	LEN	DESC
1	tgl_kirim	NOT NULL		YYYY-MM-DD
2	bank_penerbit	NOT NULL	3	Bank Code (defined by Bank Indonesia)
3	alamat_bank_penerbit	NOT NULL	255	Cukup Jelas
4	nomor_jaminan	NOT NULL	30	No Jaminan atau No Amandemen Jaminan
5	nomor_ref		30	Merunut ke No jaminan predecessor
6	jenis_jaminan	NOT NULL	2	01 : Jaminan Penawaran, 02 : Jaminan Sanggah, 03 : Jaminan Pelaksanaan, 04 : Jaminan Uang Muka, 05 : ... Note: Sesuai dengan Master Data.
7	jenis_produk	NOT NULL	1	1 : Bank Garansi, 2 : Demand Guarantee
8	beneficiary	NOT NULL	25	PT PLN (PERSERO)
9	unit_pengguna	NOT NULL	255	PLN Unit Induk : UID Jabar, dlsb
10	applicant	NOT NULL	255	PT HUTAMA KARYA
11	no_kontrak	NOT NULL	30	Contoh : 09876/PJ-01/...dst.
12	uraian_pekerjaan		255	Type free text : contoh "PAKET 3 FIXED PLTMG DOBO 2x10MW"
13	currency	NOT NULL	3	Contoh : USD, IDR, ... dst
14	nilai_jaminan	NOT NULL	17	Cukup Jelas, Dua Angka dibelakang koma. Decimal menggunakan titik
15	tgl_terbit	NOT NULL		YYYY-MM-DD
16	tgl_berlaku	NOT NULL		YYYY-MM-DD
17	tgl_berakhir	NOT NULL		YYYY-MM-DD
18	tgl_claim			YYYY-MM-DD

19	nama_file_jaminan	NOT NULL, Harus PDF, Max 5MB	255	Nama File "PDF" jaminan / BG harus sesuai
20	nama_user	NOT NULL	30	
21	email	NOT NULL	50	

#### 4.4 Format File PDF

Untuk Format File PDF Struktur File Pdf soft copy Jaminan, File softcopy Jaminan hasil scan di upload dengan extension "PDF" tanpa compress dan security,

- Nama file : **AAA-BBBB9999999999-YYYYMMDD.pdf**
- Nama file revisi : **AAA- BBBB9999999999-YYYYMMDD-REVS.pdf**
- Besar file : Maksimum **5 Mega Bytes** (5MB)
- AAA : Bank Code (defined by Bank Indonesia)
- BBBB :
  - "JM01" : Jaminan Penawaran
  - "JM02" : Jaminan Sanggah
  - "JM03" : Jaminan Pelaksanaan
  - "JM04" : Jaminan Uang Muka
  - "JM05" : Jaminan Retensi
  - "JM06" : Jaminan Bridging
- YYYYMMDD : Tanggal kirim file sysdate , 20190109 (9 Januari 2019)
- 999999999 : Nomor referensi/Sequence
- REV : Jika ada file susulan hasil revisi
- X = ke 1,2,3.. dst

## 4.5 Contoh Skrip

Berikut ini adalah contoh skrip PHP yang dijalankan di client server untuk mengirimkan data jaminan:

### 1. Multipart/form-data

#### a. PHP

```
$url = 'https://devapi.abg.pln.co.id/jaminan/store;

// form data
$form_data = [
    'tgl_kirim' = $tgl_kirim,
    'bank_penerbit' = $bank_penerbit,
    'alamat_bank_penerbit' = $alamat_bank_penerbit,
    'nomor_jaminan' = $nomor_jaminan,
    'nomor_ref' = $nomor_ref,
    'jenis_jaminan' = $jenis_jaminan,
    'jenis_produk' = $jenis_produk,
    'beneficiary' = $beneficiary,
    'unit_pengguna' = $unit_pengguna,
    'applicant' = $applicant,
    'no_kontrak' = $no_kontrak,
    'uraian_pekerjaan' = $uraian_pekerjaan,
    'currency' = $currency,
    'nilai_jaminan' = $nilai_jaminan,
    'tgl_terbit' = $tgl_terbit,
    'tgl_berlaku' = $tgl_berlaku,
    'tgl_berakhir' = $tgl_berakhir,
    'tgl_claim' = $tgl_claim,
    'nama_file_jaminan' = $nama_file_jaminan,
    'nama_user' = $nama_user,
    'email' = $email,
];

//get access token
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Accept: application/json',
    'Authorization: Bearer '. $access_token
));

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $form_data);
$tmp = curl_exec($ch);
curl_close($ch);
echo $tmp;
```

## b. Laravel

```
use GuzzleHttp\Client;

public function index()
{
    $url = 'https://devapi.abg.pln.co.id/jaminan/store';

    $http = new Client();

    $response = $http->post($url, [
        'header' => [
            'Accept' = 'application/json',
            'Authorization' = 'Bearer ' . $access_token ,
        ],
        'form_params' => [
            'tgl_kirim' = $tgl_kirim,
            'bank_penerbit' = $bank_penerbit,
            'alamat_bank_penerbit' = $alamat_bank_penerbit,
            'nomor_ref' = $nomor_ref,
            'nomor_jaminan' = $nomor_jaminan,
            'jenis_jaminan' = $jenis_jaminan,
            'jenis_produk' = $jenis_produk,
            'beneficiary' = $beneficiary,
            'unit_pengguna' = $unit_pengguna,
            'applicant' = $applicant,
            'no_kontrak' = $no_kontrak,
            'uraian_pekerjaan' = $uraian_pekerjaan,
            'currency' = $currency,
            'nilai_jaminan' = $nilai_jaminan,
            'tgl_terbit' = $tgl_terbit,
            'tgl_berlaku' = $tgl_berlaku,
            'tgl_berakhir' = $tgl_berakhir,
            'tgl_claim' = $tgl_claim,
            'nama_file_jaminan' = $nama_file_jaminan,
            'nama_user' = $nama_user,
            'email' = $email,
        ],
    ]);

    // retrieve access token
    $data = (array) json_decode((string) $response->getBody(), true);
```



### c. Postman

POST ▼ http://localhost:8000/jaminan/store

Authorization Headers (2) Body ● Pre-request Script Tests

Key	Value
<input checked="" type="checkbox"/> Accept	application/json
<input checked="" type="checkbox"/> Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImp0aSI6ImJjYWYyYTY2M3YjI3O...

POST ▼ http://localhost:8000/jaminan/store

Authorization Headers (2) Body ● Pre-request Script Tests

☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value
<input checked="" type="checkbox"/> tgl_kirim	2019-01-10
<input checked="" type="checkbox"/> bank_penerbit	072
<input checked="" type="checkbox"/> alamat_bank_penerbit	Jl. Tes
<input checked="" type="checkbox"/> nomor_jaminan	121d1118ssswdddddww
<input checked="" type="checkbox"/> nomor_ref	2
<input checked="" type="checkbox"/> jenis_jaminan	05
<input checked="" type="checkbox"/> jenis_produk	1
<input checked="" type="checkbox"/> beneficiary	PT. PLN
<input checked="" type="checkbox"/> unit_pengguna	UID Jawa Tengah
<input checked="" type="checkbox"/> applicant	PT. Sejahtera Sekali
<input checked="" type="checkbox"/> no_kontrak	9999992
<input checked="" type="checkbox"/> uraian_pekerjaan	Pekerjaan Lapangan
<input checked="" type="checkbox"/> currency	IDR
<input checked="" type="checkbox"/> nilai_jaminan	500000000
<input checked="" type="checkbox"/> tgl_terbit	2019-01-02
<input checked="" type="checkbox"/> tgl_berlaku	2019-01-02
<input checked="" type="checkbox"/> tgl_berakhir	2019-01-02
<input checked="" type="checkbox"/> tgl_claim	
<input checked="" type="checkbox"/> nama_file_jaminan	<input type="button" value="Choose Files"/> No file chosen
<input checked="" type="checkbox"/> nama_user	Budi
<input checked="" type="checkbox"/> email	dudsdsai@email.com

## 2. JSON

### a. PHP

```
$url = 'https://devapi.abg.pln.co.id/jaminan/store;

// form data
$form_data = [

    "tgl_kirim": "2019-01-02",
    "bank_penerbit": "072",
    "alamat_bank_penerbit": "Jl. Tes",
    "nomor_jaminan": "1222zsAa" ,
    "nomor_ref": "02" ,
    "jenis_produk": "1" ,
    "beneficiary": "PT. PLN" ,
    "unit_pengguna": "UID Jawa Tengah" ,
    "applicant": "PT. Sejahtera Sekaliss",
    "no_kontrak": "9999992",
    "uraian_pekerjaan": "Pekerjaan Lapangan" ,
    "currency": "IDR" ,
    "nilai_jaminan": "5000000000" ,
    "tgl_terbit": "2019-01-02",
    "tgl_berlaku" : "2019-01-02",
    "tgl_berakhir" : "2019-01-02",
    "tgl_claim" : "",
    "nama_user" : "Budi",
    "email" : "dudsdsai@email.com",
    "jenis_jaminan" : "2",
    "nama_file_jaminan" :
    "data:image/png;base64,JVBERi0xLjQKJbC4upUKMSAwIG9iajw8L1R5cGUvQ2F0YW
    xvZy9QYWdlcyAyIDAgUj4+Cm

    ] ];

//get access token
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Accept: application/json',
    'Authorization: Bearer '. $access_token
));

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $form_data);
$tmp = curl_exec ($ch);
curl_close ($ch);
echo $tmp;
```

## b. Laravel

```
use GuzzleHttp\Client;

public function index()
{
    $url = 'https://devapi.abg.pln.co.id/jaminan/store';

    $http = new Client();

    $response = $http->post($url, [
        'header' => [
            'Accept' = 'application/json',
            'Authorization' = 'Bearer '. $access_token ,
        ],
        'form_params' => [
            {
                "tgl_kirim": "2019-01-02",
                "bank_penerbit": "072",
                "alamat_bank_penerbit": "Jl. Tes",
                "nomor_jaminan": "1222zsAa" ,
                "nomor_ref": "02" ,
                "jenis_produk": "1" ,
                "beneficiary": "PT. PLN" ,
                "unit_pengguna": "UID Jawa Tengah" ,
                "applicant": "PT. Sejahtera Sekaliss" ,
                "no_kontrak": "99999992",
                "uraian_pekerjaan": "Pekerjaan Lapangan" ,
                "currency": "IDR" ,
                "nilai_jaminan": "5000000000" ,
                "tgl_terbit": "2019-01-02",
                "tgl_berlaku" : "2019-01-02",
                "tgl_berakhir" : "2019-01-02",
                "tgl_claim" : "",
                "nama_user" : "Budi",
                "email" : "dudsdsai@email.com",
                "jenis_jaminan" : "2",
                "nama_file_jaminan" :
                "data:image/png;base64,JVBERi0xLjQKJbC4upUKMSAwIG9iajw8L1R5cG
                UvQ2F0YWxvZy9QYWdlcyAyIDAgUj4+Cm
            }
        ],
    ]);

    // retrieve access token
    $data = (array) json_decode((string) $response->getBody(), true);
```



Code	Response	Keterangan
201 Created	<pre>{   "code": "201",   "status": "created",   "message": "Data berhasil di Simpan",   "data": {     "tgl_kirim": "2019-01-10",     "bank_penerbit": "072",     "alamat_bank_penerbit": "Jl. Tes",     "nomor_jaminan": "1222",     "nomor_ref": "02",     "jenis_jaminan": "2",     "jenis_produk": "1",     "beneficiary": "PT. PLN",     "unit_pengguna": "UID Jawa Tengah",     "applicant": "PT. Sejahtera Sekali",     "no_kontrak": "9999992",     "uraian_pekerjaan": "Pekerjaan Lapangan",     "currency": "IDR",     "nilai_jaminan": "500000000",     "tgl_terbit": "2019-01-02",     "tgl_berlaku": "2019-01-02",     "tgl_berakhir": "2019-01-02",     "tgl_claim": null,     "nama_file_jaminan": "2019-03-20163326-nd2019_8_17_2_rshnt.pdf",     "nama_user": "Budi",     "email": "dudsdsai@email.com",     "status": "0",     "updated_at": "2019-03-20 16:33:26",     "created_at": "2019-03-20 16:33:26",     "id": 58   } }</pre>	Sukses Terbuat

200 OK	<pre>{   "code": "200",   "status": "success",   "message": "Data berhasil di Update",   "data": {     "id": 58,     "tgl_kirim": "2019-01-10",     "bank_penerbit": "072",     "alamat_bank_penerbit": "Jl. Tes",     "nomor_jaminan": "1222",     "nomor_ref": "02",     "jenis_jaminan": "2",     "jenis_produk": "1",     "beneficiary": "PT. PLN",     "unit_pengguna": "UID Jawa Tengah",     "applicant": "PT. Sejahtera Sekali",     "no_kontrak": "9999992",     "uraian_pekerjaan": "Pekerjaan Lapangan",     "currency": "IDR",     "nilai_jaminan": "500000000",     "tgl_terbit": "2019-01-02",     "tgl_berlaku": "2019-01-02",     "tgl_berakhir": "2019-01-02",     "tgl_claim": null,     "nama_file_jaminan": "2019-03-20163613-nd2019_8_17_2_rshnt.pdf",     "nama_user": "Budi",     "email": "dudsdsai@email.com",     "no_validity": null,     "penanda_tangan": null,     "jabatan_penanda_tangan": null,     "tgl_konfirmasi": null,     "nama_file_validity": null,     "status": 0,     "created_at": "2019-03-20 16:33:26",     "updated_at": "2019-03-20 16:36:13"   } }</pre>	Sukses Update
401 Unauthorized	<pre>{   "message": "Unauthenticated." }</pre>	Kode Akses Token Tidak Sesuai

<b>422 Unprocessable Entity</b>	<pre>{   "message": "The given data was invalid.",   "errors": {     "XXXX": [       "XXXX harus Diisi!"     ],     "YYYYY": [       "YYYYY harus Diisi!"     ]   } }</pre>	Validasi Data Tidak Boleh Kosong
<b>422 Unprocessable Entity</b>	<pre>{   "message": "The given data was invalid.",   "errors": {     "XXXX": [       "XXXX harus Maximal 1 Karakter!"     ]   } }</pre>	Validasi Data Tidak Boleh Melebihi Maksimal Karakter
<b>422 Unprocessable Entity</b>	<pre>{   "message": "The given data was invalid.",   "errors": {     "XXXX": [       "XXXX harus tipe PDF!"     ]   } }</pre>	Validasi Tipe Data Tidak Sesuai
<b>500 Internal Server Error</b>	<pre>{   "code": "500",   "status": "error",   "error": "Internal Server Error",   "message": "Tidak Dapat Diproses", }</pre>	Error

## **BAB V INPUT DATA VALIDITY**

API ini digunakan untuk memasukkan data Jaminan

### **5.1 API URL**

API URL yang digunakan untuk input jaminan adalah:

- a. Development :
  - 1. Multipart/form-data : <https://devapi.abg.pln.co.id/validity/store>
  - 2. Json : <https://devapi.abg.pln.co.id/jaminan/validity/store>
- b. Production : <https://api.abg.pln.co.id/jaminan/store>
  - 1. Multipart/form-data : <https://api.abg.pln.co.id/validity/store>
  - 2. Json : <https://api.abg.pln.co.id/validity/json/store>

### **5.2 HTTP Request**

HTTP Request yang digunakan adalah POST.

Metode pengiriman data, terdapat 2 yaitu :

- a. Multipart/form-data  
Pada Bagian Header ditambahkan
  - 1. Accept : application/json
  - 2. Authorization : Bearer <<access\_token>>
- b. JSON, dengan encrypt file PDF menggunakan base64  
Pada Bagian Header ditambahkan
  - 1. Accept : application/json
  - 2. Authorization : Bearer <<access\_token>>
  - 3. Content-type : application/json



### 5.3 Request Parameter

Parameter yang dikirimkan saat melakukan input data jaminan adalah sebagai berikut:

NO	NAMA KOLOM	VALIDASI	LEN	DESC
1	no_validity	NOT NULL	30	Nomor validity bank checking
2	nomor_jaminan	NOT NULL	30	No Jaminan original yang dikirim sebelumnya
3	penanda_tangan	NOT NULL	255	Pejabat penandatanganan validity bank checking
4	Jabatan_penanda_tangan	NOT NULL	255	Jabatan pejabat penandatanganan validity bank checking
5	tgl_konfirmasi	NOT NULL		YYYY-MM-DD
6	nama_file_validity	NOT NULL		Nama File "PDF" validity bank checking" harus sesuai dengan nama file

### 5.4 Format File PDF

Untuk Format File PDF Struktur File Pdf soft copy validity bank checking/surat keabsahan dokumen, File softcopy validity bank checking/surat keabsahan dokumen hasil scan di upload dengan extension "PDF" tanpa compress dan security.

- Nama file : **AAA-BBBB9999999999-YYYYMMDD.pdf**
- Nama file revisi : **AAA- BBBB9999999999-YYYYMMDD-REVB.pdf**
- Besar file : Maksimum **5 Mega Bytes (5MB)**
- AAA : Bank Code (defined by Bank Indonesia)
- BBBB :
  - "VC01" : validity check Jaminan Penawaran
  - "VC02" : validity check Jaminan Sanggah
  - "VC03" : validity check Jaminan Pelaksanaan
  - "VC04" : validity check Jaminan Uang Muka
  - "VC05" : validity check Jaminan Retensi
  - "VC06" : validity check Jaminan Bridging dst.

- YYYYMMDD : 20190109 (9 Januari 2019)
- 999999999 : Nomor referensi/Sequence
- REV : Jika ada file susulan hasil revisi
- X = ke 1,2,3.. dst

## 5.5 Contoh Skrip

Berikut ini adalah contoh skrip PHP yang dijalankan di client server untuk mengirimkan data jaminan:

### 1. Multipart/form-data

#### a. PHP

```
$url = 'https://devapi.abg.pln.co.id/validity/store;

// form data
$form_data = [
    'no_validity' = $no_validity,
    'penanda_tangan' = $penanda_tangan,
    'jabatan_penanda_tangan' = $jabatan_penanda_tangan,
    'tgl_konfirmasi' = $tgl_konfirmasi,
    'nama_file_validity' = $nama_file_validity,
];

//get access token
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Accept: application/json',
    'Authorization: Bearer '. $access_token
));

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $form_data);
$tmp = curl_exec ($ch);
curl_close ($ch);
echo $tmp;
```

## b. Laravel

```
use GuzzleHttp\Client;

public function index()
{
    $url = 'https://devapi.abg.pln.co.id/jaminan/store';

    $http = new Client();

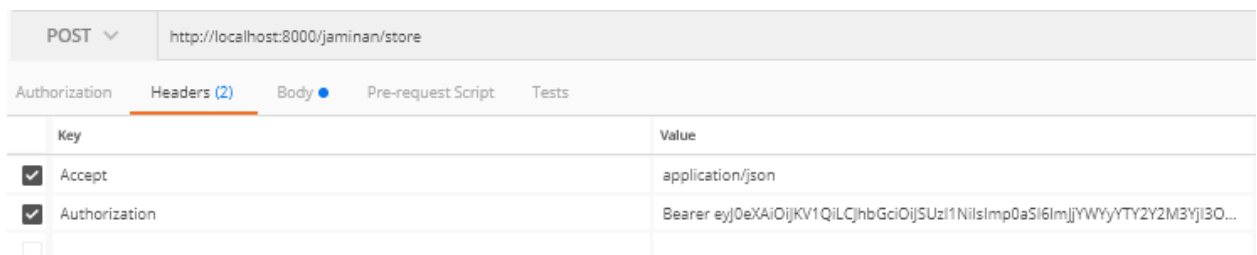
    $response = $http->post($url, [
        'header' => [
            'Accept' = 'application/json',
            'Authorization' = 'Bearer '. $access_token ,
        ],
        'form_params' => [
            'no_validity' = $no_validity,
            'penanda_tangan' = $penanda_tangan,
            'jabatan_penanda_tangan' = $jabatan_penanda_tangan,
            'tgl_konfirmasi' = $tgl_konfirmasi,
            'nama_file_validity' = $nama_file_validity,
        ],
    ]);

    // retrieve access token
    $data = (array) json_decode((string) $response->getBody(), true);

    $token = $data['access_token'];

}
```

## c. Postman



POST

localhost:8000/validity/store

Authorization

Headers (2)

Body

Pre-request Script

Tests

form-data

x-www-form-urlencoded

raw

binary

	Key	Value
<input checked="" type="checkbox"/>	no_validity	1111
<input checked="" type="checkbox"/>	penanda_tangan	Supparlan
<input checked="" type="checkbox"/>	jabatan_penanda_tangan	Manager
<input checked="" type="checkbox"/>	tgl_konfirmasi	2019-01-12
<input checked="" type="checkbox"/>	nama_file_validity	<div>Choose Files</div> No file chosen
<input checked="" type="checkbox"/>	nomor_jaminan	215

## 2. JSON

a. PHP

```
$url = 'https://devapi.abg.pln.co.id/validity/store;

// form data
$form_data = [
    {
        "tgl_konfirmasi": "2019-01-02",
        "nomor_jaminan": "1222AzAa" ,
        "penanda_tangan": "02" ,
        "no_validity": "1" ,
        "jabatan_penanda_tangan": "PT. PLN" ,
        "nama_file_validity" :
        "data:application/pdf;base64,JVBERi0xLjQKJbC4upUKMSAwIG9iajw8
L1R5cGUvQ2F0YWxvZy9QYWdlcyAyIDAgUj4+CmVuZG9iago1IDAgb2JqPDwvV
HlwZS9QYWdlL011ZGlhQm94WzAgMCA2MTIgaODQyXS9Sb3RhdGUgOCawIFIvQ2
9udGVudHMgNiAwIFIvUmVzb3VyY2VzIDcgMCBSL1BhcmVudCA0IDAgUj4+CmV
uZG9iago2IDAgb2JqPDwvRmlsdGVyL0ZsYXRlRGVjb2R1L0xlbmd0aCAxMCAw
IF"
    }
];

//get access token
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Accept: application/json',
    'Authorization: Bearer '. $access_token
));

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $form_data);
$tmp = curl_exec ($ch);
curl_close ($ch);
echo $tmp;
```

## b. Laravel

```
use GuzzleHttp\Client;

public function index()
{
    $url = 'https://devapi.abg.pln.co.id/jaminan/store';

    $http = new Client();

    $response = $http->post($url, [
        'header' => [
            'Accept' = 'application/json',
            'Authorization' = 'Bearer ' . $access_token ,
        ],

        'form_params' => [
            {
                "tgl_konfirmasi": "2019-01-02",
                "nomor_jaminan": "1222AzAa" ,
                "penanda_tangan": "02" ,
                "no_validity": "1" ,
                "jabatan_penanda_tangan": "PT. PLN" ,
                "nama_file_validity" :
                "data:application/pdf;base64,JVBERi0xLjQKJbC4upUKMSAwIG9iajw8L1
                R5cGUvQ2F0YWxvZy9QYWdlcyAyIDAgUj4+CmVuZG9iago1IDAgb2JqPDwvVHlwZ
                S9QYWdlL01lZGhlQm94WzAgMCA2MTIgODQyXS9Sb3RhdGUgOCwIFiVQ29udGVu
                dHMgNiAwIFiVUmVzb3VyY2VzIDcgMCBSL1BhcmVudCA0IDAgUj4+CmVuZG9iago
                2IDAgb2JqPDwvRmlsdGVyL0ZsYXRlRGVjb2R1L0xlbmd0aCAxMCAwIF"
            }
        ],
    ]);

    // retrieve access token
    $data = (array) json_decode((string) $response->getBody(), true);
```

## c. Postman





Code	Response	Keterangan
<b>200 OK</b>	<pre>{   "code": "200",   "status": "success",   "message": "Data berhasil di Update",   "data": {     "id": 58,     "tgl_kirim": "2019-01-10",     "bank_penerbit": "072",     "alamat_bank_penerbit": "Jl. Tes",     "nomor_jaminan": "1222",     "nomor_ref": "02",     "jenis_jaminan": "2",     "jenis_produk": "1",     "beneficiary": "PT. PLN",     "unit_pengguna": "UID Jawa Tengah",     "applicant": "PT. Sejahtera Sekali",     "no_kontrak": "9999992",     "uraian_pekerjaan": "Pekerjaan Lapangan",     "currency": "IDR",     "nilai_jaminan": "500000000",     "tgl_terbit": "2019-01-02",     "tgl_berlaku": "2019-01-02",     "tgl_berakhir": "2019-01-02",     "tgl_claim": null,     "nama_file_jaminan": "2019-03-20163613-nd2019_8_17_2_rshnt.pdf",     "nama_user": "Budi",     "email": "dudsdsai@email.com",     "no_validity": null,     "penanda_tangan": null,     "jabatan_penanda_tangan": null,     "tgl_konfirmasi": null,     "nama_file_validity": null,     "status": 0,     "created_at": "2019-03-20 16:33:26",     "updated_at": "2019-03-20 16:36:13"   } }</pre>	Sukses Update
<b>401 Unauthorized</b>	<pre>{   "message": "Unauthenticated." }</pre>	Kode Akses Token Tidak Sesuai

<b>404 Not Found</b>	<pre>{   "code": "404",   "status": "error",   "error": "Not Found",   "message": {     "nomor_jaminan": "Nomor Jaminan tidak Ditemukan"   } }</pre>	Nomor Jaminan Tidak Ditemukan
<b>422 Unprocessable Entity</b>	<pre>{   "message": "The given data was invalid.",   "errors": {     "XXXX": [       "XXXX harus Diisi!"     ],     "YYYYY": [       "YYYYY harus Diisi!"     ]   } }</pre>	Validasi Data Tidak Boleh Kosong
<b>422 Unprocessable Entity</b>	<pre>{   "message": "The given data was invalid.",   "errors": {     "XXXX": [       "XXXX harus Maximal 1 Karakter!"     ]   } }</pre>	Validasi Data Tidak Boleh Melebihi Maksimal Karakter
<b>422 Unprocessable Entity</b>	<pre>{   "message": "The given data was invalid.",   "errors": {     "XXXX": [       "XXXX harus tipe PDF!"     ]   } }</pre>	Validasi Tipe Data Tidak Sesuai
<b>500 Internal Server Error</b>	<pre>{   "code": "500",   "status": "error",   "error": "Internal Server Error",   "message": "Tidak Dapat Diproses", }</pre>	Error