

# Fungsi-Fungsi Umum

[SignIn](#), [SignOut](#)

---

## fungsi SignIn()

Kegunaan : Mengawali sesi transaksi data dengan autentikasi user dan mendapatkan Session ID

PathInfo : `/SignIn`

Request Method : `POST`

Request Content Type : `application/json`

Request String : 

```
{
  "LoginInfo":
  {
    "LoginName":<string>,
    "Salt":<string>,
    "AuthHash":<string>
  }
}
```

### Keterangan:

**Salt**: Random string, length 10 - 12 chars

**AuthHash**: Hash autentikasi yang didapatkan dengan fungsi dan parameter:

**MD5**(Salt + **MD5**(LoginName + Password))

### Contoh:

```
LoginName = "user"; Password = "password"; Salt = "56Yh02gF882s";
```

```
AuthHash = MD5("56Yh02gF882s" + MD5("user" + "password"));
           = MD5("56Yh02gF882s" + "d440aed189a13ff970dac7e7e8f987b2");
           = "cb398c7d06e7c34c342e3b8d36d0a0c9";
```

Online MD5 Generator tersedia [di sini](#).

Response String : 

```
{
  "Result":<boolean>,
  "SID":<string>
}
```

### Keterangan:

- Jika **Result** bernilai **True** maka **SID** akan berisi Session ID yang dapat dipakai untuk autentikasi di fungsi-fungsi berikutnya. Jika bernilai **False** maka **SID** akan berisi Error Message
- Session ID akan expire jika dalam waktu 30 menit tidak ada aktivitas.

Catatan : Awali sesi dengan fungsi **SignIn** jika berikutnya akan memanggil fungsi-fungsi melalui RPC dengan mengikutsertakan SID sebagai parameter.

Jika fungsi RPC yang akan dipanggil hanya satu saja dan relatif jarang-jarang, **SignIn** tidak diperlukan karena fungsi-fungsi yang tersedia untuk RPC mendukung **LoginInfo** sebagai parameter pengganti **SID**.

---

## fungsi SignOut()

Kegunaan : Mengakhiri sesi transaksi data dan logged out dari server

PathInfo : `/SignIn`

Request Method : `POST`

Request Content Type : `application/json`

Request String : `{  
 "SID":<string>  
}`

Response String : `{  
 "Result":<boolean>,  
 "Remark":<string>  
}`

Catatan : Akhiri sesi dengan fungsi `SignIn` jika sebelumnya diawali dengan fungsi `SignIn` untuk secara eksplisit menghapus sesi user pada server.

Jika fungsi-fungsi RPC dipanggil menggunakan parameter `LoginInfo` (tanpa `SID` yang didapat dari `SignIn`) maka fungsi `SignIn` tidak perlu dijalankan.

## Fungsi-Fungsi Blokir Request Kendaraan Baru

[RanRuRequestBlokir](#), [RanRuGetBlokirEntryByDate](#), [RanRuGetBlokirEntryByNoKa](#), [RanRuRequestReblokir](#), [RanRuGetReblokirEntryByDate](#), [RanRuGetReblokirEntryByNoKa](#)

---

### fungsi `RanRuRequestBlokir()`

Kegunaan : Mendaftarkan kendaraan-kendaraan baru berdasarkan list nomor rangka untuk otomatis diblokir saat BPKBnya diproses.

PathInfo : `/RanRuRequestBlokir`

Request Method : `POST`

Request Content Type : `application/json`

Request String : `{  
 "SID":<string>,  
 "RequestBlokirItemList":  
 [  
 {  
 "NoRangka":<string>,  
 "NoPermohonan":<string>  
 },  
 .  
 .  
 .  
 ]  
}`

atau jika tanpa Session ID

```
{
  "LoginInfo":
  {
    "LoginName":<string>,
    "Salt":<string>,
    "AuthHash":<string>
  }
  "RequestBlokirItemList":
  [
    {
      "NoRangka":<string>,
      "NoPermohonan":<string>
    }
  ]
}
```

```

    },
    .
    .
    .
  ]
}

Response String : {
  "Result":<boolean>,
  "ResponseBlokirItemList":
  [
    {
      "NoRangka":<string>,
      "KodeError":<int>
    },
    .
    .
    .
  ]
}

```

**Keterangan:**

- Jika nilai **Result** = **true**, seluruh data diterima tanpa error. Array **ResponseBlokirItemList** akan tidak ada atau berupa empty array. Jika **false**, maka ada 1 atau lebih record yang ditolak, array **ResponseBlokirItemList** berisi info record-record tsb.
- Elemen array **ResponseBlokirItemList** memiliki field sbb:
  - **NoRangka**: nomor rangka yang ditolak
  - **KodeError**:
    - 1 = record sudah ada;
    - 2 = BPKB sudah diblokir;
    - 3 = BPKB dengan nomor rangka ybs sudah diterbitkan lebih dari waktu treshold
    - 4 = BPKB dengan nomor rangka ybs sudah diterbitkan dan statusnya tidak aktif
    - 5 = BPKB dengan nomor rangka ybs sudah diterbitkan dan sudah balik nama

## fungsi RanRuGetBlokirEntryByDate()

Kegunaan : Mengambil list data request blokir yang pernah dikirim berdasarkan rentang waktu

PathInfo : **/RanRuGetBlokirEntryByDate**

Request Method : **POST**

Request Content Type : **application/json**

```

Request String : {
  "SID":<string>,
  "FromDate":<string>, /* yyyyymmdd */
  "ToDate":<string>    /* yyyyymmdd */
}

```

atau jika tanpa Session ID

```

{
  "LoginInfo":
  {
    "LoginName":<string>,
    "AuthHash":<string>,
    "Salt":<string>
  },
  "FromDate":<string>, /* yyyyymmdd */
  "ToDate":<string>    /* yyyyymmdd */
}

```

```

Response String : {
  "Result":<boolean>,
  "RequestBlokirEntryList":
  [
    {
      "NoRangka":<string>,
      "NoBpkb":<string>,
      "NoBlokir":<string>,
      "TglBlokir":<string>,          /* yyyymmdd */
      "TglSubmit":<string>,         /* yyyymmddhhnnss */
      "TglBpkb":<string>,
      "NoPolisi":<string>,
      "SubID":<int>
      "NoRegBpkb":<string>,
      "NoMesin":<string>,
      "NoFaktur":<string>
      "TglFaktur":<string>,         /* yyyymmdd */
      "Merk":<string>,
      "Tipe":<string>,
      "Jenis":<string>,
      "Model":<string>,
      "BahanBakar":<string>,
      "ThnBuat":<int>,
      "ThnRakit":<int>,
      "Warna":<string>,
      "VolSilinder":<int>,
      "NamaPemilik":<string>,
      "AlamatPemilik":<string>,
      "NoIdentitas":<string>,
      "TglDaftar":<string>,         /* yyyymmddhhnnss */
      "TglEntri":<string>,          /* yyyymmddhhnnss */
      "TglVerifikasi":<string>,     /* yyyymmddhhnnss */
      "TglCetakKartuInduk":<string>, /* yyyymmddhhnnss */
      "TglCetakBpkb":<string>,      /* yyyymmddhhnnss */
      "TglPenyerahan":<string>,     /* yyyymmddhhnnss */
      "Status":<string>
    },
    .
    .
    .
  ]
}

```

#### Keterangan:

- Nilai dari **Result** selalu **true**.
- Jika hasil query tidak menghasilkan data, maka **RequestBlokirEntryList** akan tidak ada atau berupa empty array.
- Jika request blokir belum diproses, field-field yang berisi hanya **NoRangka** dan **TglSubmit**.
- **SubID** adalah nilai **Submission ID**, merupakan identitas record yang akan menjadi referensi saat proses Buka Blokir kendaraan baru.

## fungsi RanRuGetBlokirEntryByNoKa()

Kegunaan : Mengambil list data request blokir yang pernah dikirim berdasarkan list nomor rangka  
 PathInfo : **/RanRuGetBlokirEntryByNoKa**  
 Request Method : **POST**  
 Request Content Type : **application/json**  
 Request String : {  
     **"SID":<string>**,

```

    "NoRangkaList":
    [
        <string>,
        <string>,
        .
        .
        .
        <string>
    ]
}

```

atau jika tanpa Session ID

```

{
    "LoginInfo":
    {
        "LoginName":<string>,
        "Salt":<string>,
        "AuthHash":<string>
    },
    "NoRangkaList":
    [
        <string>,
        <string>,
        .
        .
        .
        <string>
    ]
}

```

Response String : Lihat [Response String pada fungsi RanRuGetBlokirEntryByDate\(.\)](#).

---

## fungsi RanruRequestReblokir()

**Kegunaan** : Mendaftarkan kendaraan-kendaraan berdasarkan list nomor rangka untuk otomatis diblokir saat BPKB penggantinya (BPKB baru ganti kepemilikan diterbitkan). Kendaraan dalam list nomor rangka harus dalam kondisi sedang terblokir dan oleh pemohon yang sama dengan yang saat ini melakukan request.

**PathInfo** : `/RanruRequestReblokir`

**Request Method** : `POST`

**Request Content Type** : `application/json`

**Request String** : {

```

    "SID":<string>,
    "RequestReblokirItemList":
    [
        {
            "NoRangka":<string>,
            "NoPermohonan":<string>
        },
        .
        .
        .
    ]
}

```

atau jika tanpa Session ID

```

{
    "LoginInfo":
    {
        "LoginName":<string>,
        "Salt":<string>,

```

```

    "AuthHash":<string>
  }
  "RequestReblokirItemList": // array
  [
    {
      "NoRangka":<string>,
      "NoPermohonan":<string>
    },
    .
    .
    .
  ]
}

Response String : {
  "Result":<boolean>,
  "ResponseReblokirItemList":
  [
    {
      "NoRangka":<string>,
      "KodeError":<int>
    },
    .
    .
    .
  ]
}

```

**Keterangan:**

- Jika nilai **Result** = **true**, seluruh data diterima tanpa error. Array **ResponseReblokirItemList** akan tidak ada atau berupa empty array. Jika **false**, maka ada 1 atau lebih record yang ditolak, array **ResponseReblokirItemList** berisi info record-record tsb.
- Elemen array **ResponseReblokirItemList** memiliki field sbb:
  - **NoRangka**: nomor rangka yang ditolak
  - **KodeError**:
    - 1 = BPKB tidak ditemukan;
    - 2 = Status BPKB ybs tidak aktif;
    - 3 = BPKB sudah terdaftar untuk blokir otomatis
    - 4 = BPKB Sedang dalam status blokir Pidana
    - 5 = BPKB Sedang dalam status tidak terblokir
    - 6 = Pemohon blokir balik nama tidak sesuai dengan pemohon blokir saat ini
    - else = unspecified error

---

## fungsi RanruGetReblokirEntryByDate()

Kegunaan : Mengambil list data request blokir balik nama yang pernah dikirim berdasarkan rentang waktu

PathInfo : **/RanruGetReblokirEntryByDate**

Request Method : **POST**

Request Content Type : **application/json**

Request String : {

```

  "SID":<string>,
  "FromDate":<string>, /* yyyymmdd */
  "ToDate":<string> /* yyyymmdd */
}

```

atau jika tanpa Session ID

```
{
  "LoginInfo":
  {
    "LoginName":<string>,
    "AuthHash":<string>,
    "Salt":<string>
  },
  "FromDate":<string>,    /* yyyyymmdd */
  "ToDate":<string>      /* yyyyymmdd */
}
```

```
Response String : {
  "Result":<boolean>,
  "RequestReblokirEntryList":
  [
    {
      "NoBpkb":<string>,
      "NoBlokir":<string>,
      "TglBlokir":<string>,    /* yyyyymmdd */
      "TglSubmit":<string>,    /* yyyyymmddhhnnss */
      "TglBpkb":<string>,      /* yyyyymmdd */
      "NoPolisi":<string>,
      "NoRegBpkb":<string>,
      "NoMesin":<string>,
      "NoRangka":<string>,
      "NoFaktur":<string>,
      "TglFaktur":<string>,    /* yyyyymmdd */
      "Merk":<string>,
      "Tipe":<string>,
      "Jenis":<string>,
      "Model":<string>,
      "BahanBakar":<string>,
      "ThnBuat":<int>,
      "ThnRakit":<int>,
      "Warna":<string>,
      "VolSilinder":<int>,
      "NamaPemilik":<string>,
      "AlamatPemilik":<string>,
      "NoIdentitas":<string>,
      "TglDaftar":<string>,    /* yyyyymmddhhnnss */
      "TglEntri":<string>,     /* yyyyymmddhhnnss */
      "TglVerifikasi":<string>, /* yyyyymmddhhnnss */
      "TglCetakKartuInduk":<string>, /* yyyyymmddhhnnss */
      "TglCetakBpkb":<string>,    /* yyyyymmddhhnnss */
      "TglPenyerahan":<string>,   /* yyyyymmddhhnnss */
      "Status":<string>
    },
    .
    .
    .
  ]
}
```

#### Keterangan:

- Nilai dari **Result** selalu **true**.
- Jika hasil query tidak menghasilkan data, maka **RequestReblokirEntryList** akan tidak ada atau berupa empty array.
- Jika request blokir belum diproses, field-field yang berisi hanya **NoRangka** dan **TglSubmit**.

## fungsi RanruGetReblokirEntryByNoKa()

Kegunaan : Mengambil list data request blokir balik nama yang pernah dikirim berdasarkan nomor rangka

PathInfo : `/RanruGetReblokirEntryByNoKa`

Request Method : `POST`

Request Content Type : `application/json`

Request String : {

```

    "SID":<string>,
    "NoRangkaList":
    [
        <string>,
        <string>,
        .
        .
        .
        <string>
    ]
}

```

atau jika tanpa Session ID

```

{
    "LoginInfo":
    {
        "LoginName":<string>,
        "AuthHash":<string>,
        "Salt":<string>
    },
    "NoRangkaList":
    [
        <string>,
        <string>,
        .
        .
        .
        <string>
    ]
}

```

Response String : Lihat [Response String pada fungsi RanRuGetReblokirEntryByDate\(.\)](#).

## Fungsi-Fungsi Blokir Request Kendaraan Lama

[RanMaGetDataRanmor](#), [RanMaVerifiedForBlokir](#), [RanMaExecuteBlokir](#)

---

### fungsi RanMaGetDataRanmor()

Kegunaan : Mengambil data ranmor untuk diverifikasi

PathInfo : `/RanMaGetDataRanmor`

Request Method : `POST`

Request Content Type : `application/json`

Request String : {

```

    "SID":<string>,
    "NoBPKB":<string>, /* deprecated, boleh tidak ada, gunakan field "Param" */
    "Param": {
        "Type":<int>',
        "Value":<string>
    }
}

```



```

    }
  }
atau jika tanpa Session ID
{
  "LoginInfo":
  {
    "LoginName":<string>,
    "Salt":<string>,
    "AuthHash":<string>
  },
  "NoBPKB":<string>,    /* deprecated, boleh tidak ada, gunakan field "Param" */
  "Param": {
    "Type":<int>',
    "Value":<string>
  }
}

```

**Keterangan:**

- Parameter **NoBPKB** dan **Param** *mutually exclusive*.  
Parameter **NoBPKB** deprecated, dianjurkan untuk tidak dipakai. Kalau dipakai, field **Param** harus tidak ada.  
Kalau obyek dalam field **Param** berisi, maka field **NoBPKB** akan diabaikan atau boleh tidak ada.
- Jika menggunakan field **Param**, berikut adalah keterangan field-fieldnya:
  - **Type = 1: Value** adalah **Nomor BPKB**
  - **Type = 2: Value** adalah **Nomor Rangka**
  - **Type = 3: Value** adalah **Nomor Mesin**

```

Response String : {
  "Result":<boolean>,
  "DataRanmor":
  {
    "NoBPKB":<string>,
    "TglBPKB":<string>,          /* yyyyymmdd */
    "NoRegister":<string>
    "TempatPenerbitan":<string>,
    "NoPolisi":<string>,
    "WarnaTNKB":<string>,
    "NoRangka":<string>,
    "NoMesin":<string>,
    "Merk":<string>,
    "Tipe":<string>,
    "Jenis":<string>,
    "Model":<string>,
    "JumlahRoda":<int>,
    "JumlahSumbu":<int>,
    "ThnBuat":<int>,
    "ThnRakit":<int>,
    "VolSilinder":<int>,
    "BahanBakar":<string>,
    "Warna":<string>,
    "NamaPemilik":<string>,
    "NoIdentitas":<string>,
    "Pekerjaan":<string>,
    "Alamat":<string>,
    "KodePos":<string>,
    "WilayahSamsat":<string>,
    "NoFaktur":<string>,
    "TglFaktur":<string>,      /* yyyyymmdd */
    "Peruntukan":<string>,
    "JenisDaftaran":<string>,
    "NoPabea":<string>,

```

```

    "TglPabean":<string>,                                /* yyyyymmdd */
    "Pelabuhan":<string>,
    "CaraImpor":<string>,
    "NoCKD":<string>,
    "KeteranganPabean":<string>,
    "Status":<string>,
    "KeteranganStatus":<string>,
    "Pemohon":<string>,
    "BpkbSebelumnya":
    {
        "NoBPKB":<string>,
        "TglBPKB":<string>,                                /* yyyyymmdd */
        "NoRegister":<string>
        "TempatPenerbitan":<string>,
        "NamaPemilik":<string>,
        "Alamat":<string>,
        "SebabTidakBerlaku":<string>
    }
    }
}

```

**Keterangan:**

- Jika nilai **Result** = **true**, maka obyek **DataRanmor** akan berisi, jika **false** maka **DataRanmor** akan tidak ada atau berisi obyek kosong.
- Obyek **BpkbSebelumnya** hanya akan ada/berisi jika data ranmor memiliki buku BPKB pendahulu.

---

## fungsi RanMaVerifiedForBlokir()

**Kegunaan** : Menyatakan ranmor telah diverifikasi dan disetujui untuk diblokir. Data yang dikirim ke server akan mendapatkan ID Approval.

**PathInfo** : **/RanMaVerifiedForBlokir**

**Request Method** : **POST**

**Request Content Type** : **application/json**

**Request String** : {  
     "SID":<string>,  
     "NoBPKB":<string>  
 }

atau jika tanpa Session ID

```

{
    "LoginInfo":
    {
        "LoginName":<string>,
        "Salt":<string>,
        "AuthHash":<string>
    },
    "NoBPKB":<string>
}

```

**Response String** : {  
     "Result":<boolean>,  
     "NoBPKB":<string>,  
     "KodeError":<int>,      /\* 0: no error;  
                               1: BPKB not found;  
                               2: already verified;  
                               3: already blokir-ed \*/  
     "IdApproval":<int>  
 }

**Keterangan:**

- Jika **Result** bernilai **true**, maka **IdApproval** akan berisi integer value yang nantinya akan dipakai untuk identifikasi saat melakukan blokir secara online. Jika **false**, **IdApproval** akan bernilai **0**.  
**IdApproval** ini juga akan dipakai sebagai referensi untuk melakukan Buka Blokir kendaraan lama.
- **KodeError** dapat berisi nilai sbb:
  - **0**: tidak ada error (**Result = true**)
  - **1**: BPKB tidak ditemukan
  - **2**: verifikasi sudah pernah disubmit sebelumnya
  - **3**: BPKB telah diblokir

## fungsi RanMaExecuteBlokir()

Kegunaan : Mengeksekusi proses blokir secara remote.

PathInfo : **/RanMaExecuteBlokir**

Request Method : **POST**

Request Content Type : **application/json**

Request String : {  
     "SID":<string>,  
     "DataPermohonan":  
     {  
         "NoBPKB":<string>,  
         "NoPermohonan":<string>,  
         "IdApproval":<int>  
     }  
 }

atau jika tanpa Session ID

```
{
  "LoginInfo":
  {
    "LoginName":<string>,
    "Salt":<string>,
    "AuthHash":<string>
  },
  "DataPermohonan":
  {
    "NoBPKB":<string>,
    "NoPermohonan":<string>,
    "IdApproval":<int>
  }
}
```

Response String : {  
     "Result":<boolean>,  
     "NoBPKB":<string>,  
     "KodeError":<int>,      /\* 0: no error  
                               1: BPKB not found  
                               2: approval entry does not exist  
                               3: BPKB not match  
                               4: Pemohon don't match  
                               5: already blokired \*/  
     "BlokirData":  
     {  
         "NoRegBlokir":<string>,  
         "TglBlokir":<string>,      /\* yyyyymmdd \*/  
         "TglAkhirBerlaku":<string>   /\* yyyyymmdd \*/  
     }  
 }

**Keterangan:**

- Jika **Result** bernilai **true**, maka **BlokirData** akan berisi informasi hasil pemblokiran. Jika **false**, **BlokirData** akan tidak ada atau berupa *empty object*..
- **KodeError** dapat berisi nilai sbb:
  - **0**: tidak ada error (**Result = true**)
  - **1**: BPKB tidak ditemukan
  - **2**: data tidak ditemukan dalam database verified request
  - **3**: BPKB yang disubmit saat verifikasi tidak sesuai dengan BPKB yang hendak diblokir
  - **4**: User yang melakukan verifikasi blokir tidak sama dengan user yang melakukan blokir
  - **5**: BPKB statusnya telah diblokir

## Fungsi-Fungsi Complementary (Pelengkap)

[ComplGetBerkasCheckPoint](#), [ComplGetDataAtpmSub](#), [ComplExecuteBukaBlokir](#), [ComplGetBpkbEntriesByTglBpkb](#), [ComplGetBpkbEntriesByTglBpkbBrief](#)

---

### fungsi ComplGetBerkasCheckPoint()

Kegunaan : Mengambil data berkas ranmor, berikut dengan informasi timeline tahapan proses penerbitan.

PathInfo : **/ComplGetBerkasCheckPoint**

Request Method : **POST**

Request Content Type : **application/json**

Request String : {

```

  "SID":<string>,
  "Criteria":
  {
    "Param":<string>,
    "ParamKind":<int> /* menerangkan jenis Param
                        1: Param adalah Nomor Rangka
                        2: Param adalah Nomor Mesin */
  }
}
```

atau jika tanpa Session ID

```

{
  "LoginInfo":
  {
    "LoginName":<string>,
    "Salt":<string>,
    "AuthHash":<string>
  },
  "Criteria":
  {
    "Param":<string>,
    "ParamKind":<int> /* menerangkan jenis Param
                        1: Param adalah Nomor Rangka
                        2: Param adalah Nomor Mesin */
  }
}
```

Response String : {

```

  "Result":<boolean>,
  "Data":
  {
    "NoBpkb":<string>,
```

```

    "TglBpkb":<string>, /* yyyyymmdd */
    "NoPolisi":<string>,
    "NoRangka":<string>,
    "NoMesin":<string>,
    "NoFaktur":<string>, /* yyyyymmdd */
    "TglFaktur":<string>,
    "Merk":<string>,
    "Tipe":<string>,
    "Jenis":<string>,
    "Model":<string>,
    "BahanBakar":<string>,
    "ThnBuat":<int>,
    "ThnRakit":<int>,
    "Warna":<string>,
    "VolSilinder":<int>,
    "Pemilik":<string>,
    "NoIdentitas":<string>,
    "Alamat":<string>,
    "Pemohon":<string>,
    "TglDaftar":<string>, /* yyyyymmdd */
    "TglEntri":<string>, /* yyyyymmdd */
    "TglVerifikasi":<string>, /* yyyyymmdd */
    "TglCetakKartuInduk":<string>, /* yyyyymmdd */
    "TglCetakBpkb":<string>, /* yyyyymmdd */
    "TglPenyerahan":<string>, /* yyyyymmdd */
    "Status":<string>
  }
}

```

#### Keterangan:

- Jika **Result** bernilai **true**, maka **Data** akan berisi informasi berkas ranmor. Jika **false**, artinya data tidak ditemukan dan object **Data** akan tidak ada atau berupa *empty object*.
- Jika berkas BPKB telah diserahkan, **TglPenyerahan** akan berisi, dan penerimanya adalah sama dengan field **Pemohon**.
- Nilai field **Status** dapat berupa:
  - AKTIF
  - TIDAK AKTIF (mati, besi tua, dll)
  - BLOKIR
  - MUTASI LUAR DAERAH

## fungsi ComplGetDataAtpmSub()

Kegunaan : Mengambil data ranmor yang dikirim oleh ATPM dari database ATPM Online.

PathInfo : /ComplGetDataAtpmSub

Request Method : POST

Request Content Type : application/json

Request String : {

```

  "SID":<string>,
  "Criteria":
  {
    "Param":<string>,
    "ParamKind":<int> /* menerangkan jenis Param
                        1: Param adalah Nomor Rangka
                        2: Param adalah Nomor Mesin */
  }
}

```

atau jika tanpa Session ID

```

{
  "LoginInfo":
  {
    "LoginName":<string>,
    "Salt":<string>,
    "AuthHash":<string>
  },
  "Criteria":
  {
    "Param":<string>,
    "ParamKind":<int> /* menerangkan jenis Param
                        1: Param adalah Nomor Rangka
                        2: Param adalah Nomor Mesin */
  }
}

```

Response String : {

```

  "Result":<boolean>,
  "Data":
  {
    "NoFaktur":<string>,
    "TglFaktur":<string>, /* yyymmdd */
    "NoRangka":<string>,
    "NoMesin":<string>,
    "Merk":<string>,
    "Tipe":<string>,
    "Model":<string>,
    "Jenis":<string>,
    "Silinder":<string>,
    "BahanBakar":<string>,
    "ThnBuat":<string>,
    "ThnRakit":<string>,
    "Pemilik":<string>,
    "Alamat":<string>,
    "NoIdentitas":<string>,
    "KodePos":<string>
  }
}

```

#### Keterangan:

- Jika **Result** bernilai **true**, maka **Data** akan berisi informasi berkas ranmor. Jika **false**, artinya data tidak ditemukan dan object **Data** akan tidak ada atau berupa *empty object*..

## fungsi ComplExecuteBukaBlokir()

Kegunaan : Membuka blokir kendaraan lama atau baru yang sebelumnya diblokir menggunakan fasilitas Blokir Online.

PathInfo : /ComplExecuteBukaBlokir

Request Method : POST

Request Content Type : application/json

Request String : {

```

  "SID":<string>,
  "Param": {
    "SubID":<int>,
    "IdApproval":<int>,
    "NoSuratPermohonan":<string>,
    "Alasan":<string>, /* max 128 chars */
    "NoSuratRef":<string>,
    "TglSuratPemohon":<string> /* yyymmdd */
  }
}

```

```

    }
  }
atau jika tanpa Session ID
{
  "LoginInfo": {
    "LoginName":<string>,
    "Salt":<string>,
    "AuthHash":<string>
  },
  "Param": {
    "SubID":<int>,
    "IdApproval":<int>,
    "NoRangka":<string>,
    "NoSuratPermohonan":<string>,
    "Alasan":<string>,                /* max 128 chars */
    "NoSuratRef":<string>
    "TglSuratPemohon":<string>      /* yyyyymmdd */
  }
}

```

**Keterangan:**

- Parameter **SubID** dan **IdApproval** adalah *mutually exclusive*.
  - Untuk kendaraan baru, **SubID** harus berisi value yang berasal dari fungsi [RanRuGetBlokirEntryByDate](#) atau [RanRuGetBlokirEntryByNoKa](#). Field **IdApproval** boleh tidak ada atau berisi *empty string*.
  - Untuk kendaraan lama, **IdApproval** harus berisi value yang berasal dari fungsi [RanMaVerifiedForBlokir](#). Field **SubID** boleh tidak ada atau berisi *empty string*.

```

Response String : {
  "ResultCode":<int>,
  "OutputData": {
    "NoBukaBlokir":<string>,
    "TglBukaBlokir":<string>        /* yyyyymmdd */
  }
}

```

**Keterangan:**

- Keterangan value **ResultCode** :
  - **0**: sukses, tidak ada error
  - **1**: error, data tidak ditemukan
  - **2**: error, ranmor tidak dalam status blokir
  - **3**: error, ranmor sedang dalam status blokir non perdata (pidana atau duplikat)
  - **4**: error, pemohon blokir tidak sama dengan pemohon buka blokir
  - **5**: error, BPKB sudah tidak aktif
  - **else**: unspecified error
- Jika value **ResultCode** tidak sama dengan **0**, maka object **OutputData** akan kosong atau tidak tersedia.

---

## fungsi ComplGetBpkbEntriesByTglBpkb()

Kegunaan : Mengambil data entri ranmor BPKB berdasarkan tanggal BPKB.  
 PathInfo : [/ComplGetBpkbEntriesByTglBpkb](#)  
 Request Method : **POST**  
 Request Content Type : **application/json**  
 Request String : {
 "SID":<string>,
 }

```

"Param": {
  "TglBpkb":<string>      /* yyyyymmdd */
  "MerkId":<int>,          /* bila berisi dan value-nya > 0,
                           maka value dari "MerkNama" dan "IsR2" boleh
                           kosong karena akan diabaikan */
  "MerkNama":<string>,    /* string nama merk, digunakan jika MerkID kosong */
  "IsR2":<boolean>        /* true: sepeda motor,
                           false: selain sepeda motor
                           digunakan jika MerkID kosong */
}
}
atau jika tanpa Session ID
{
  "LoginInfo": {
    "LoginName":<string>,
    "Salt":<string>,
    "AuthHash":<string>
  },
  "Param": {
    "TglBpkb":<string>      /* yyyyymmdd */
    "MerkId":<int>,          /* bila berisi dan value-nya > 0,
                             maka value dari "MerkNama" dan "IsR2" boleh
                             kosong karena akan diabaikan */
    "MerkNama":<string>,    /* string nama merk, digunakan jika MerkID kosong */
    "IsR2":<boolean>        /* true: sepeda motor,
                             false: selain sepeda motor
                             digunakan jika MerkID kosong */
  }
}

```

**Keterangan:**

- Menggunakan MerkID sudah secara implisit menspesifikasikan R2 atau R4 karena entri merk sudah diasosiasikan dengan jenis roda kendaraan.

Response String : {

```

  "Result":<boolean>,          /* true: sukses; false: gagal */
  "List": [
    {
      "NoBpkb":<string>,
      "TglBpkb":<string>,      /* yyyyymmdd */
      "NoRangka":<string>,
      "NoPolisi":<string>,
      "NoRegBpkb":<string>,
      "NoMesin":<string>,
      "NoFaktur":<string>,
      "TglFaktur":<string>,    /* yyyyymmdd */
      "Merk":<string>,
      "Tipe":<string>,
      "Jenis":<string>,
      "Model":<string>,
      "BahanBakar":<string>,
      "ThnBuat":<int>,
      "ThnRakit":<int>,
      "Warna":<string>,
      "VolSilinder":<int>,
      "NamaPemilik":<string>,
      "AlamatPemilik":<string>,
      "NoIdentitas":<string>,
      "TglDaftar":<string>,    /* yyyyymmddhhnnss */
      "TglEntri":<string>,     /* yyyyymmddhhnnss */
      "TglVerifikasi":<string>, /* yyyyymmddhhnnss */
      "TglCetakKartuInduk":<string>, /* yyyyymmddhhnnss */
    }
  ]
}

```



```

        "TglCetakBpkb":<string>,          /* yyyyymmddhhnnss */
        "TglPenyerahan":<string>,        /* yyyyymmddhhnnss */
        "Status":<string>
    }
    .
    .
    .
]
}

```

## fungsi ComplGetBpkbEntriesByTglBpkbBrief()

Kegunaan : Mengambil data entri ranmor BPKB berdasarkan rentang tanggal BPKB dengan field terbatas.

PathInfo : /ComplGetBpkbEntriesByTglBpkbBrief

Request Method : POST

Request Content Type : application/json

Request String : {

```

    "SID":<string>,
    "Param": {
        "TglBpkbFrom":<string> /* yyyyymmdd */
        "TglBpkbTo":<string>   /* yyyyymmdd */
        "MerkId:<int>,          /* bila berisi dan value-nya > 0,
                                maka value dari "MerkNama" dan "IsR2" boleh
                                kosong karena akan diabaikan */
        "MerkNama:<string>,     /* string nama merk, digunakan jika MerkID kosong */
        "IsR2":<boolean>       /* true: sepeda motor,
                                false: selain sepeda motor
                                digunakan jika MerkID kosong */
    }
}

```

atau jika tanpa Session ID

```

{
    "LoginInfo": {
        "LoginName":<string>,
        "Salt":<string>,
        "AuthHash":<string>
    },
    "Param": {
        "TglBpkbFrom":<string> /* yyyyymmdd */
        "TglBpkbTo":<string>   /* yyyyymmdd */
        "MerkId:<int>,          /* bila berisi dan value-nya > 0,
                                maka value dari "MerkNama" dan "IsR2" boleh
                                kosong karena akan diabaikan */
        "MerkNama:<string>,     /* string nama merk, digunakan jika MerkID kosong */
        "IsR2":<boolean>       /* true: sepeda motor,
                                false: selain sepeda motor
                                digunakan jika MerkID kosong */
    }
}

```

### Keterangan:

- **TglBpkbFrom** : tanggal awal BPKB sebagai awal range pencarian  
**TglBpkbTo**: tanggal akhir BPKB sebagai akhir range pencarian.  
**TglBpkbFrom** dan **TglBpkbTo** adalah inclusive.
- Menggunakan MerkID sudah secara implisit menspesifikasikan R2 atau R4 karena entri merk sudah diasosiasikan dengan jenis roda kendaraan.

Response String : {

```

    "Result":<boolean>,          /* true: sukses; false: gagal */

```

```

    "List": [
      {
        "NoBpkb":<string>,
        "TglBpkb":<string>,           /* yyyyymmdd */
        "NoMesin":<string>,
        "TglFaktur":<string>,         /* yyyyymmdd */
        "TglDaftar":<string>,         /* yyyyymmddhhnnss */
        "TglCetakBpkb":<string>,     /* yyyyymmddhhnnss */
        "TglPenyerahan":<string>     /* yyyyymmddhhnnss */
      }
      .
      .
      .
    ]
  }

```

## Catatan

### [Exception, Server Access](#)

---

## Exception

Setiap Response String yang merupakan kembalian dari fungsi RPC sebelum di-*unmarshalled* dan ditangani secara *normal flow* di caller, harus dipastikan bahwa bukan berupa *exception response message* dari server.

Exception response dari server memiliki struktur sbb:

```

{
  "Status":"Exception",
  "ExceptionClass":<string>,
  "IsSystemException":<boolean>,
  "Message":<string>
}

```

Jika terdapat field dengan nama **"Status"** dengan value **"Exception"** maka caller harus handle exception dan tidak meneruskan *normal flow*.

Keterangan exception object

- **Status**: string, selalu bernilai **"Exception"**
  - **ExceptionClass**: nama *exception class* di server
  - **IsSystemException**: boolean, jika **true** berarti exception adalah system exception yang diakibatkan misalnya masalah koneksi database, access violation, dsb. Jika **false** berarti application exception misalnya kesalahan format data, atau Session ID yang expired.
  - **Message**: detail keterangan exception
- 

## Server Access

Alamat web service adalah **http://service.poldametro.net/bro/browsj.dll**

Untuk tiap fungsi, URL yang dipergunakan ditentukan oleh PathInfo, dengan format:

**http://poldametro.bpkb.net/Blokir-Online/browsj.dll<path\_info>**

Misalnya untuk melakukan SignIn, dimana PathInfo adalah **/SignIn**, maka URL yang diakses adalah:

**http://poldametro.bpkb.net/Blokir-Online/browsj.dll/SignIn**

Semua request method adalah **HTTP POST** dengan content-type = **application/json** dan JSON string request berada pada body request (POST data).

*Doc Ver : 1.14*