



MANUAL BOOK API DEVELOPER GUIDE

NAKHODA API

Panduan Teknis bagi *Developer/Pengembang Sistem*
API Publikasi Dokumen Kebijakan Kabupaten Belitung Timur, Portal Nakhoda



**BADAN PERENCANAAN PEMBANGUNAN, RISET, DAN INOVASI DAERAH
(BAPPERIDA) KABUPATEN BELITUNG TIMUR**

Versi API v1 - Disusun 01 September 2026

Tentang Manual Book Ini

Dokumen ini adalah panduan teknis bagi *developer* atau pengembang sistem, baik dari internal Bapperida maupun dari instansi/OPD lain, yang ingin mengambil data dokumen kebijakan yang sudah dipublikasikan di Nakhoda secara terprogram (*machine to machine*), misalnya untuk ditampilkan kembali di portal instansi sendiri, diintegrasikan ke Portal Satu Data, atau diolah untuk kebutuhan analisis lain.

API Nakhoda bersifat baca saja (*read only*). Tidak ada satu pun *endpoint* yang dapat digunakan untuk membuat, mengubah, atau menghapus data. Seluruh *endpoint* hanya mengeluarkan dokumen yang berstatus **published** dan tidak disembunyikan dari pembaca, apa pun parameter yang dikirimkan oleh klien.

CATATAN Manual ini merupakan versi cetak dari laman Dokumentasi API yang juga tersedia langsung di situs Nakhoda pada alamat `/dokumentasi-api`, lengkap dengan fitur untuk mencoba setiap *endpoint* langsung dari peramban. Kedua versi ini saling melengkapi: gunakan manual ini sebagai referensi *offline* yang bisa dicetak atau dibagikan, dan laman web untuk mencoba langsung secara interaktif.

Gunakan daftar isi pada halaman berikut untuk berpindah cepat ke bagian yang ingin dibaca. Setiap judul pada daftar isi dapat diklik langsung (tekan tombol Ctrl sambil mengklik, atau klik biasa tergantung versi Microsoft Word) dan akan langsung membawa ke bagian yang dituju.

DAFTAR ISI

Tentang Manual Book Ini	i
1. Tentang API Nakhoda.....	1
1.1 Ringkasan	1
1.2 Alamat Dasar (<i>Base URL</i>)	1
1.3 Prinsip Utama	1
2. Cara Mendapatkan Akses	2
2.1 Kontak.....	2
3. Autentikasi	3
4. Konvensi Umum.....	4
4.1 Format Respons Daftar (<i>List</i>).....	4
4.2 Format Respons Detail Entitas	4
4.3 Paginasi	4
4.4 Format Tanggal	5
4.5 <i>Rate Limit</i>	5
5. Referensi <i>Endpoint</i>.....	6
5.1 <i>Endpoint</i> Dokumen	6
GET /api/v1/documents	6
GET /api/v1/documents/{slug}.....	6
<i>Skema Objek Dokumen</i>	6
5.2 <i>Endpoint</i> Instansi	7
GET /api/v1/institutions.....	7
GET /api/v1/institutions/{slug}.....	7
<i>Skema Objek Instansi</i>	8
5.3 <i>Endpoint</i> Bidang Urusan	8
GET /api/v1/bidang-urusan.....	8
GET /api/v1/bidang-urusan/{slug}.....	8
<i>Skema Objek Bidang Urusan</i>	8
5.4 <i>Endpoint</i> Penulis	8
GET /api/v1/penulis	8
GET /api/v1/penulis/{slug}.....	9
<i>Skema Objek Penulis</i>	9
5.5 <i>Endpoint</i> Pencarian	9
GET /api/v1/search	9
6. Kode <i>Error</i>	10
7. Contoh Implementasi.....	11
7.1 curl	11
7.2 JavaScript (fetch).....	11
7.3 PHP (cURL).....	11
7.4 Python (requests).....	11
8. Uji Coba Interaktif.....	12
9. Praktik Baik	13
10. Kontak dan Dukungan	14

1. Tentang API Nakhoda

1.1 Ringkasan

API Nakhoda adalah REST API berbasis JSON yang menyediakan akses baca terhadap katalog dokumen kebijakan (*Policy Brief*, *Policy Paper*, dan Makalah Kebijakan) yang telah dipublikasikan, beserta data pendukungnya: instansi, bidang urusan, dan penulis. Seluruh respons berbentuk JSON standar, dapat dikonsumsi dari bahasa pemrograman apa pun yang mampu melakukan permintaan HTTP.

1.2 Alamat Dasar (*Base URL*)

Seluruh *endpoint* API diakses lewat alamat dasar berikut, diikuti *path* masing-masing *endpoint*:

```
https://nakhoda.beltim.go.id/api/v1/
```

PERHATIAN Ganti alamat di atas dengan domain resmi Nakhoda yang berlaku pada saat manual ini digunakan. Selama masa pengembangan/lokal, alamat dasar mengikuti domain lingkungan pengujian yang diberikan admin.

1.3 Prinsip Utama

- Baca saja (*read only*): tidak ada *endpoint* untuk menulis, mengubah, atau menghapus data.
- Hanya dokumen *published*: dokumen berstatus draf, sedang ditelaah, atau perlu revisi tidak akan pernah muncul lewat API, terlepas dari parameter apa pun yang dikirim klien.
- Butuh autentikasi: setiap permintaan wajib menyertakan *API Key* yang sah lewat *header Authorization*.
- Dibatasi laju permintaan (*rate limit*): setiap Klien API dibatasi maksimal 60 permintaan per menit untuk menjaga stabilitas layanan bagi semua pengguna.

2. Cara Mendapatkan Akses

Akses API tidak dapat diperoleh lewat pendaftaran mandiri. *API Key* hanya dapat dibuat oleh Admin Nakhoda lewat panel administrasi, melalui prosedur berikut:

1. *Developer* atau instansi yang membutuhkan akses menghubungi Admin Nakhoda, dengan menyebutkan nama instansi/organisasi, nama sistem atau aplikasi yang akan mengakses API, serta tujuan penggunaan datanya.
2. Admin membuat entri Klien API baru lewat menu Klien API pada panel administrasi, lalu men-*generate API Key* untuk klien tersebut.
3. *API Key* diserahkan ke pihak yang meminta lewat kanal komunikasi yang aman, di luar situs Nakhoda.

PERHATIAN *API Key* harus disimpan dengan baik dan tidak boleh disebarluaskan atau dipublikasikan di kode sumber publik (misalnya di repositori GitHub yang terbuka). Siapa pun yang memegang *key* tersebut dapat memakainya untuk mengakses API atas nama klien yang bersangkutan.

Apabila *API Key* hilang, bocor, atau ingin diganti, hubungi kembali Admin untuk di-*generate* ulang. *Key* lama akan otomatis tidak berlaku begitu *key* baru dibuat, sehingga tidak ada jeda waktu ketika dua *key* sama-sama aktif.

2.1 Kontak

Permintaan *API Key*, pertanyaan teknis seputar integrasi, atau laporan kendala dapat disampaikan lewat:

- Email: bapperida@beltim.go.id atau beltimrida@gmail.com
- WhatsApp Admin: +62 877 3970 2422

3. Autentikasi

Nakhoda menggunakan skema *Bearer Token* untuk autentikasi API, sesuai standar *header Authorization* pada HTTP. Setiap permintaan ke *endpoint* mana pun wajib menyertakan *header* berikut:

```
Authorization: Bearer <API_KEY_ANDA>
```

Contoh permintaan lengkap memakai `curl`:

```
curl -H "Authorization: Bearer <API_KEY_ANDA>" \  
      "https://nakhoda.beltim.go.id/api/v1/documents"
```

Permintaan yang tidak menyertakan *header* ini, atau menyertakan *key* yang salah ataupun sudah dicabut, akan ditolak dengan status 401 Unauthorized. Permintaan dengan *key* yang sah namun sedang dinonaktifkan Admin akan ditolak dengan status 403 Forbidden.

4. Konvensi Umum

4.1 Format Respons Daftar (*List*)

Setiap *endpoint* yang mengembalikan daftar data selalu berbentuk objek dengan tiga kunci berikut:

Kunci	Tipe	Keterangan
data	array	Daftar hasil, satu objek per item.
links	object	Tautan halaman pertama, terakhir, sebelumnya, dan berikutnya.
meta	object	Info paginasi: <code>current_page</code> , <code>last_page</code> , <code>per_page</code> , <code>total</code> , dan lainnya.

```
{
  "data": [ { ... }, { ... } ],
  "links": {
    "first": "https://nakhoda.beltim.go.id/api/v1/documents?page=1",
    "last":  "https://nakhoda.beltim.go.id/api/v1/documents?page=5",
    "prev":  null,
    "next":  "https://nakhoda.beltim.go.id/api/v1/documents?page=2"
  },
  "meta": {
    "current_page": 1,
    "last_page": 5,
    "per_page": 20,
    "total": 92
  }
}
```

4.2 Format Respons Detail Entitas

Endpoint detail untuk Instansi, Bidang Urusan, dan Penulis (bukan *endpoint* Dokumen) mengembalikan bentuk gabungan: data entitasnya sendiri, ditambah daftar dokumen terkait yang berbentuk paginasi sama seperti pada bagian 4.1:

```
{
  "data": { ... detail entitas ... },
  "documents": {
    "data": [ { ... }, { ... } ],
    "links": { ... },
    "meta": { ... }
  }
}
```

4.3 Paginasi

Parameter yang berlaku pada setiap *endpoint* daftar (dapat digabung satu sama lain):

Parameter	Tipe	Keterangan
per_page	integer	Jumlah data per halaman, maksimal 50 (nilai baku bervariasi per <i>endpoint</i> , umumnya 20).
page	integer	Nomor halaman yang diminta, dimulai dari 1.

4.4 Format Tanggal

Seluruh *field* tanggal dan waktu (misalnya `published_at`) menggunakan format ISO 8601 lengkap dengan zona waktu, contoh: `2026-09-07T10:46:06+07:00`.

4.5 Rate Limit

Setiap Klien API dibatasi maksimal 60 permintaan per menit. Batas ini dihitung per *API Key*, bukan per alamat IP, sehingga tidak saling memengaruhi antar klien yang berbeda. Permintaan yang melebihi batas akan ditolak dengan status 429 Too Many Requests, dan permintaan berikutnya dapat dicoba lagi setelah beberapa saat.

TIPS Untuk kebutuhan pengambilan data dalam jumlah besar, disarankan menyimpan hasil di sisi klien (*cache*) dan hanya memanggil ulang API secara berkala, bukan pada setiap kali halaman aplikasi klien dibuka.

5. Referensi *Endpoint*

Bagian ini merinci seluruh *endpoint* yang tersedia, lengkap dengan parameter dan contoh pemanggilan.

5.1 *Endpoint* Dokumen

GET /api/v1/documents

Mengembalikan daftar dokumen *published*, dapat disaring dan dipaginasi.

Parameter	Tipe	Keterangan
type	string	policy_brief, policy_paper, atau makalah_kebijakan.
category_id	integer	Saring berdasarkan ID kategori.
topic_id	integer	Saring berdasarkan ID topik.
year	integer	Saring berdasarkan tahun.
q	string	Cari judul yang mengandung teks ini (cocok <i>substring</i> , bukan <i>full text</i>).
per_page	integer	Jumlah data per halaman, maksimal 50 (baku 20).
page	integer	Nomor halaman.

```
curl -H "Authorization: Bearer <API_KEY_ANDA>" \
  "https://nakhoda.beltim.go.id/api/v1/documents?type=policy_brief&per_page=10"
```

GET /api/v1/documents/{slug}

Mengembalikan detail satu dokumen berdasarkan *slug* (didapat dari *field slug* pada respons daftar). Mengembalikan status 404 apabila *slug* tidak ditemukan atau dokumennya belum/tidak lagi *published*.

Skema Objek Dokumen

Setiap objek dokumen pada *field data* memuat kunci berikut:

Field	Tipe	Keterangan
id	integer	ID unik dokumen.
slug	string	Identitas dokumen pada URL, dipakai untuk <i>endpoint</i> detail.
type	string	Kode jenis: policy_brief, policy_paper, atau makalah_kebijakan.
type_label	string	Nama jenis yang mudah dibaca, contoh "Policy Brief".

Field	Tipe	Keterangan
title	string	Judul dokumen.
abstract	string	Abstrak/ringkasan dokumen.
authors	array	Daftar penulis, tiap item memuat <code>name</code> dan <code>institution</code> .
year	integer	Tahun penerbitan.
language	string	Kode bahasa naskah.
pages	integer null	Jumlah halaman, bila tercatat.
doc_number	string null	Nomor dokumen, bila ada.
doi	string null	Digital Object Identifier, bila ada.
institution	string null	Nama instansi utama penulis.
category	string null	Nama kategori.
topic	string null	Nama topik.
tags	array	Daftar nama <i>tag</i> .
published_at	string null	Tanggal publikasi, format ISO 8601.
view_count	integer	Jumlah dilihat.
download_count	integer	Jumlah diunduh.
cover_url	string null	URL gambar sampul dokumen.
download_url	string null	URL berkas PDF publikasi resmi untuk diunduh.
public_url	string	URL halaman detail dokumen di situs publik Nakhoda.

CATATAN *Field* internal seperti pencatat pembuat naskah, tanggal pengajuan, catatan telaah, dan *path* penyimpanan berkas asli SENGAJA tidak pernah dikeluarkan lewat API ini, untuk menjaga privasi proses telaah internal.

5.2 Endpoint Instansi

GET /api/v1/institutions

Mengembalikan daftar instansi/OPD.

Parameter	Tipe	Keterangan
q	string	Cari berdasarkan nama atau nama singkat instansi.
is_pemda	boolean	<code>true</code> untuk instansi pemerintah daerah, <code>false</code> untuk lainnya.
per_page	integer	Jumlah data per halaman, maksimal 50 (baku 20).
page	integer	Nomor halaman.

GET /api/v1/institutions/{slug}

Mengembalikan detail satu instansi beserta daftar dokumen `published` dari instansi tersebut (dipaginasi lewat parameter `per_page` dan `page`).

Skema Objek Instansi

Field	Tipe	Keterangan
id	integer	ID unik instansi.
name	string	Nama lengkap instansi.
short_name	string null	Nama singkat/akronim instansi.
slug	string	Identitas instansi pada URL.
is_pemda	boolean	Penanda instansi pemerintah daerah atau bukan.
documents_count	integer	Jumlah dokumen <code>published</code> dari instansi ini.

5.3 Endpoint Bidang Urusan

GET /api/v1/bidang-urusan

Mengembalikan daftar bidang urusan pemerintahan daerah.

Parameter	Tipe	Keterangan
q	string	Cari berdasarkan nama atau kode bidang urusan.
per_page	integer	Jumlah data per halaman, maksimal 50 (baku 20).
page	integer	Nomor halaman.

GET /api/v1/bidang-urusan/{slug}

Mengembalikan detail satu bidang urusan beserta daftar dokumen `published` yang terkait dengannya.

Skema Objek Bidang Urusan

Field	Tipe	Keterangan
id	integer	ID unik bidang urusan.
kode	string	Kode bidang urusan.
nama	string	Nama bidang urusan.
slug	string	Identitas bidang urusan pada URL.
urusan	object null	Urusan induk, memuat kode dan nama.
documents_count	integer	Jumlah dokumen <code>published</code> yang terkait.

5.4 Endpoint Penulis

Penulis pada Nakhoda tidak memiliki tabel data tersendiri: daftar penulis diagregasi secara otomatis dari data penulis yang tercatat pada setiap dokumen **published**. Identitas penulis pada URL berupa *slug* hasil dari nama penulis.

GET /api/v1/penulis

Mengembalikan daftar penulis beserta jumlah dokumennya.

Parameter	Tipe	Keterangan
q	string	Cari berdasarkan nama penulis.
per_page	integer	Jumlah data per halaman, maksimal 50 (baku 20).
page	integer	Nomor halaman.

GET /api/v1/penulis/{slug}

Mengembalikan detail satu penulis beserta daftar dokumen **published** yang ditulisnya.

Skema Objek Penulis

Field	Tipe	Keterangan
slug	string	Identitas penulis pada URL.
name	string	Nama penulis.
institution_name	string null	Nama instansi penulis, bila tercatat.
documents_count	integer	Jumlah dokumen published yang ditulis.
badge	object null	Lencana penghargaan (<i>tier</i> , <i>label</i> , <i>number</i>) berdasarkan jumlah dokumen.

5.5 Endpoint Pencarian

GET /api/v1/search

Melakukan pencarian teks penuh (*full text*) di judul, abstrak, dan nama penulis sekaligus, memakai mesin pencarian yang sama dengan fitur pencarian pada situs publik Nakhoda, bukan sekadar mencocokkan potongan kata.

Parameter	Tipe	Keterangan
q	string	Wajib diisi. Kata kunci pencarian, minimal 2 karakter.
type	string	Saring jenis dokumen, sama seperti <i>endpoint</i> Dokumen.
bidang_urusan_id	integer	Saring berdasarkan ID bidang urusan.
institution_id	integer	Saring berdasarkan ID instansi.
year	integer	Saring berdasarkan tahun.

Parameter	Tipe	Keterangan
sort	string	relevance (baku), newest, popular, atau download.
per_page, page	integer	Paginasi, sama seperti <i>endpoint</i> lain.

```
curl -H "Authorization: Bearer <API_KEY_ANDA>" \  
  "https://nakhoda.beltim.go.id/api/v1/search?q=tata+kelola&type=policy_pape  
r"
```

6. Kode *Error*

Setiap kegagalan permintaan dikembalikan dengan struktur JSON yang konsisten, minimal memuat kunci `message`. Untuk kegagalan validasi parameter (kode 422), ditambahkan kunci `errors` yang merinci kesalahan tiap parameter.

Status	Arti	Penyebab Umum
401	Unauthorized	<i>Header Authorization</i> tidak ada, atau <i>API Key</i> tidak valid/sudah dicabut.
403	Forbidden	<i>API Key</i> sah, tetapi Klien API-nya sedang dinonaktifkan Admin.
404	Not Found	<i>Slug</i> salah, atau data yang diminta belum/tidak lagi published.
422	Unprocessable Entity	Parameter yang dikirim tidak valid, lihat detail di kunci <code>errors</code> .
429	Too Many Requests	Melebihi batas 60 permintaan per menit untuk Klien API tersebut.

```
{
  "message": "Parameter tidak valid.",
  "errors": {
    "q": ["Parameter q (kata kunci pencarian) wajib diisi."]
  }
}
```

7. Contoh Implementasi

7.1 curl

```
curl -H "Authorization: Bearer <API_KEY_ANDA>" \
      "https://nakhoda.beltim.go.id/api/v1/documents?per_page=5"
```

7.2 JavaScript (fetch)

```
fetch("https://nakhoda.beltim.go.id/api/v1/documents?per_page=5", {
  headers: {
    "Authorization": "Bearer <API_KEY_ANDA>",
    "Accept": "application/json"
  }
})
.then(response => response.json())
.then(data => console.log(data));
```

7.3 PHP (cURL)

```
<?php

$ch = curl_init("https://nakhoda.beltim.go.id/api/v1/documents?per_page=5");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, [
    "Authorization: Bearer <API_KEY_ANDA>",
    "Accept: application/json",
]);
$response = curl_exec($ch);
curl_close($ch);

$data = json_decode($response, true);
```

7.4 Python (requests)

```
import requests

response = requests.get(
    "https://nakhoda.beltim.go.id/api/v1/documents",
    params={"per_page": 5},
    headers={"Authorization": "Bearer <API_KEY_ANDA>"},
)
data = response.json()
```

8. Uji Coba Interaktif

Selain lewat manual cetak ini, seluruh *endpoint* di atas dapat dicoba langsung dari peramban tanpa menulis kode sama sekali, lewat fitur "Coba Langsung" pada laman Dokumentasi API di situs Nakhoda:

```
https://nakhoda.beltim.go.id/dokumentasi-api
```

Fitur ini menyediakan formulir untuk memasukkan *API Key*, memilih *endpoint* yang ingin dicoba, mengisi parameter tambahan, lalu langsung melihat hasil respons JSON apa adanya. Permintaan pada fitur ini dikirim langsung dari peramban ke Nakhoda, tanpa melalui server perantara mana pun, dan *API Key* yang dimasukkan tidak pernah disimpan atau dicatat oleh laman tersebut.

9. Praktik Baik

- Jangan menulis *API Key* langsung di kode sumber yang tersimpan di repositori publik. Simpan sebagai variabel lingkungan (*environment variable*) atau berkas konfigurasi yang tidak ikut diunggah ke repositori.
- Simpan hasil respons di sisi klien (*cache*) untuk data yang jarang berubah, dan hindari memanggil API pada setiap kali halaman dimuat.
- Selalu periksa kode status HTTP sebelum mengolah isi respons, bukan hanya mengandalkan isi *body* JSON-nya saja.
- Gunakan parameter *per_page* secukupnya sesuai kebutuhan tampilan, tidak perlu selalu mengambil jumlah maksimal (50) apabila hanya menampilkan beberapa data saja.
- Tangani kemungkinan status 429 dengan jeda dan mencoba lagi (*retry* dengan *delay*), bukan langsung mengulang permintaan secara beruntun.

10. Kontak dan Dukungan

Untuk pertanyaan teknis, laporan kendala, permintaan *API Key* baru, atau permintaan penggantian *API Key*, silakan hubungi Admin Nakhoda melalui:

- Email: bapperida@beltim.go.id atau beltimrida@gmail.com
- WhatsApp Admin: +62 877 3970 2422