

WABLASH – API Documentation V3

Backend REST API (Node.js + Express + MongoDB)

Version: 3.0 • Generated: 2025-09-15 21:48:50 WIB

Ringkasan

Dokumen ini merupakan rujukan resmi API WABLASH V3. Berisi daftar endpoint, skema request/response, autentikasi, error codes, dan contoh penggunaan via cURL & Postman. Disusun berdasarkan backend yang saat ini dipakai pada pengembangan WABLASH (auth, manajemen user, WhatsApp blast, dan utilitas lain).

1. Dasar Penggunaan

Base URL (Default Dev): `http://localhost:3001/api`

Gunakan header berikut untuk endpoint yang membutuhkan autentikasi:

```
Authorization: Bearer <JWT_TOKEN>
Content-Type: application/json
```

Role: *superadmin, admin, user*

Akses endpoint diatur oleh role. Keterangan akses tercantum di masing-masing endpoint.

2. Auth

2.1 POST /auth/login

Login dan mendapatkan JWT token.

POST /auth/login

Body (JSON):

```
{
  "username": "admin",
  "password": "secret"
}
```

Response 200:

```
{
  "ok": true,
  "token": "<jwt>",
  "user": {
    "id": "68c793f90fab0eca5267b1a4",
    "username": "admin",
    "role": "admin"
  }
}
```

Response 400/401:

```
{ "ok": false, "message": "Invalid credentials" }
```

2.2 POST /auth/register

Registrasi user baru. Hanya untuk **admin/superadmin** jika kebijakan produksi memerlukan approval.

POST /auth/register

Body (JSON):

```
{
  "username": "user1",
  "password": "secret",
  "role": "user"           // one of: user|admin|superadmin
}
```

Response 200:

```
{
  "ok": true,
  "user": {
    "id": "68c793f90fab0eca5267b1a4",
    "username": "user1",
    "role": "user"
  }
}
```

Response 400:

```
{ "ok": false, "message": "username/password required" }
```

2.3 GET /auth/me

Mendapatkan profil user dari token yang sedang aktif.

GET /auth/me

Headers: Authorization: Bearer <jwt>

Response 200:

```
{
  "ok": true,
  "user": {
    "id": "68c793f90fab0eca5267b1a4",
    "username": "admin",
    "role": "admin"
  }
}
```

3. User Management

Model user memuat field utama: `_id`, `username`, `role`, `active` (boolean), `totalPesan` (number), `createdAt`, `updatedAt`.

3.1 GET /users

List semua user (role-based filtering via query). **admin/superadmin**.

```
GET /users?role=user&limit=50&page=1
Headers: Authorization: Bearer <jwt>

Response 200:
{
  "ok": true,
  "users": [
    {
      "_id": "68c793f90fab0eca5267b1a4",
      "username": "user1",
      "role": "user",
      "active": true,
      "totalPesan": 12,
      "createdAt": "2025-09-14T10:20:00.000Z"
    }
  ]
}
```

3.2 POST /users

Tambah user baru. **admin/superadmin**.

```
POST /users
Body (JSON):
{
  "username": "marketingA",
  "password": "secret",
  "role": "user"
}

Response 200:
{ "ok": true, "user": { "_id": "...", "username": "marketingA", "role": "user" } }
```

3.3 GET /users/:id

Detail user. **admin/superadmin**.

```
GET /users/68c793f90fab0eca5267b1a4

Response 200:
{ "ok": true, "user": { "_id": "...", "username": "user1", "role": "user", "active": false, "totalPesan": 12 } }
```

3.4 PUT /users/:id

Update user. **admin/superadmin**.

```
PUT /users/68c793f90fab0eca5267b1a4
Body (JSON):
{ "username": "user1_updated", "role": "admin" }

Response 200:
{ "ok": true, "user": { "_id": "...", "username": "user1_updated", "role": "admin" } }
```

3.5 DELETE /users/:id

Hapus user. **admin/superadmin**.

```
DELETE /users/68c793f90fab0eca5267b1a4

Response 200:
{ "ok": true }
```

3.6 PATCH /users/:id/active

Set status online/active user (opsional sesuai implementasi FE). **admin/superadmin.**

PATCH /users/68c793f90fab0eca5267b1a4/active

Body (JSON):

```
{ "active": true }
```

Response 200:

```
{ "ok": true, "user": { "_id": "...", "active": true } }
```

4. WhatsApp Blast

Fitur pengiriman pesan massal (blast) ke nomor WhatsApp. Sistem menyimpan status (queued, sent, delivered, read) dan waktu terjadinya. Pastikan service WhatsApp gateway (resmi atau integrasi internal) telah dikonfigurasi.

4.1 POST /blasts

Kirim blast ke satu atau banyak penerima. **admin/superadmin**.

```
POST /blasts
Body (JSON):
{
  "content": { "text": "Promo Merdeka 🇮🇩" },
  "recipients": [ "6285117591609", "6281234567890" ],
  "channel": "whatsapp"
}

Response 200:
{
  "ok": true,
  "blast": {
    "_id": "68c885c3ec1eef5b611fd81e",
    "channel": "whatsapp",
    "content": { "text": "Promo Merdeka 🇮🇩" },
    "recipients": [
      { "phone": "6285117591609", "status": "queued" },
      { "phone": "6281234567890", "status": "queued" }
    ],
    "createdAt": "2025-09-15T21:31:47.400Z"
  }
}
```

4.2 GET /blasts

List blast (pagination & filter). **admin/superadmin**.

```
GET /blasts?limit=20&page=1&channel=whatsapp&status=read

Response 200:
{
  "ok": true,
  "items": [
    {
      "_id": "68c885c3ec1eef5b611fd81e",
      "channel": "whatsapp",
      "recipientsCount": 2,
      "delivered": 2,
      "read": 1,
      "createdAt": "2025-09-15T21:31:47.400Z"
    }
  ],
  "page": 1,
  "limit": 20,
  "total": 1
}
```

4.3 GET /blasts/:id

Detail blast, termasuk status tiap penerima.

```
GET /blasts/68c885c3ec1eef5b611fd81e

Response 200:
{
  "ok": true,
  "blast": {
    "_id": "68c885c3ec1eef5b611fd81e",
    "channel": "whatsapp",
    "content": { "text": "Belum tidur Bu, masih nyoba WA Blash hehe 🇮🇩" },
    "recipients": [
```

```
{
  "phone": "6285117591609",
  "status": "read",
  "timestamps": {
    "queuedAt": "2025-09-15T21:31:47.400Z",
    "sentAt": "2025-09-15T21:31:47.814Z",
    "readAt": "2025-09-15T21:xx:xx.xxxZ"
  }
}
]
```

4.4 GET /blasts/:id/status

Ringkasan status deliverability.

```
GET /blasts/68c885c3ecleef5b611fd81e/status
```

Response 200:

```
{
  "ok": true,
  "summary": {
    "queued": 0, "sent": 2, "delivered": 2, "read": 1, "failed": 0
  }
}
```

5. Aktivitas & Analitik

Aktivitas user diturunkan dari data user (field *active*, *totalPesan*) dan data blast.

5.1 GET /stats/summary

Ringkasan angka agregat untuk dashboard admin.

```
GET /stats/summary
```

Response 200:

```
{
  "ok": true,
  "data": {
    "totalUsers": 120,
    "activeUsers": 35,
    "totalBlastsThisMonth": 15420,
    "topMarketers": [
      { "username": "Marketing A", "sent": 5250 },
      { "username": "Marketing B", "sent": 8100 }
    ]
  }
}
```

6. Pola Response & Error

Seluruh response sukses punya format { ok: true, ... }. Error menggunakan { ok: false, message, code? }

HTTP Code	Meaning	Contoh Response
200	OK	{ "ok": true, ... }
201	Created	{ "ok": true, ... }
400	Bad Request	{ "ok": false, "message": "Invalid payload" }
401	Unauthorized	{ "ok": false, "message": "Invalid credentials" }
403	Forbidden	{ "ok": false, "message": "Insufficient role" }
404	Not Found	{ "ok": false, "message": "Not found" }
429	Too Many Requests	{ "ok": false, "message": "Rate limit exceeded" }
500	Server Error	{ "ok": false, "message": "Internal error" }

7. Contoh cURL & Postman

Contoh cURL:

```
# Login
curl -X POST http://localhost:3001/api/auth/login -H "Content-Type: application/json" -d '{"username": "Marketing A", "password": "1234567890"}'

# List Users (Bearer)
curl -X GET http://localhost:3001/api/users -H "Authorization: Bearer <jwt>"

# Kirim Blast
curl -X POST http://localhost:3001/api/blasts -H "Authorization: Bearer <jwt>" -H "Content-Type: application/json" -d '{"message": "Blast message", "target": "Marketing A"}'
```


8. Konfigurasi ENV & Keamanan

```
# .env contoh
PORT=3001
MONGO_URI=mongodb://localhost:27017/wablash
JWT_SECRET=<super_secret>
JWT_EXPIRES_IN=7d
WHATSAPP_GATEWAY_URL=<url_gateway_resmi_atau_internal>
WHATSAPP_GATEWAY_TOKEN=<token_gateway>
```

Best Practices: • Simpan JWT_SECRET dan token gateway di secret manager. • Terapkan rate limiting pada endpoint /auth dan /blasts. • Validasi payload: sanitasi nomor penerima (format 62...), panjang pesan, dsb. • Audit log untuk aksi admin (create/update/delete user, kirim blast).

9. Changelog V3

• Konsolidasi endpoint auth, users, blasts, stats dalam satu dokumen. • Penambahan contoh cURL dan pola error yang seragam. • Penjelasan field active & totalPesan untuk kebutuhan ActivityPage. • Penegasan akses berbasis role (superadmin/admin/user).

© 2025 WABLASH • Dokumentasi ini untuk penggunaan internal tim.