

Dokumentasi API Born Angel

Dokumentasi lengkap seluruh endpoint REST API Born Angel – platform booking layanan kecantikan/makeup.

Base URL: `http://127.0.0.1:8000/api` (development) atau `https://nama-app.up.railway.app/api` (production)

Daftar Isi

- 1. [Informasi Umum](#)
- 2. [Autentikasi](#)
- 3. [Profil Pengguna](#)
- 4. [Manajemen Pengguna \(Admin\)](#)
- 5. [Manajemen Layanan](#)
- 6. [Manajemen Instruktur](#)
- 7. [Manajemen Jadwal](#)
- 8. [Manajemen Booking](#)
- 9. [Manajemen Review](#)
- 10. [Pembayaran \(Midtrans\)](#)
- 11. [Dashboard & Laporan Admin](#)
- 12. [Skema Database](#)

Informasi Umum

Format Request & Response

- Semua request dan response menggunakan format **JSON**
- Endpoint upload gambar menggunakan **multipart/form-data**
- Semua response mengikuti format Laravel standar

Header yang Dibutuhkan

```
Accept: application/json
Content-Type: application/json
Authorization: Bearer {token}    ← hanya untuk endpoint terotentikasi
```

Kode Status HTTP

Kode	Arti
200	Berhasil
201	Berhasil dibuat
400	Request tidak valid / logika bisnis gagal
403	Akses ditolak (tidak punya izin)
404	Data tidak ditemukan
422	Validasi gagal
500	Error server

Format Error Validasi (422)

```
{
  "message": "The email field is required. (and 1 more error)",
  "errors": {
    "email": ["The email field is required."],
    "password": ["The password field is required."]
  }
}
```

Hierarki Peran (RBAC)

super_admin > admin > instructor > user

Peran	Akses
super_admin	Akses penuh, kelola admin, lihat pendapatan. User ID 1 tidak bisa diubah/dihapus.
admin	Kelola layanan, jadwal, instruktur, pengguna. Tidak bisa menyentuh super admin.
instructor	Lihat jadwal & review milik sendiri saja
user	Booking, review, kelola profil sendiri

Paginasi

Endpoint yang mengembalikan daftar data menggunakan paginasi Laravel:

```
{
  "data": [...],
  "meta": {
    "current_page": 1,
    "from": 1,
    "last_page": 5,
    "per_page": 15,
    "to": 15,
    "total": 72
  }
}
```

Parameter query `per_page` tersedia di semua endpoint berpaginasi (default bervariasi per endpoint).

Autentikasi

1. Registrasi Pengguna Baru

POST /api/register

Autentikasi: Tidak perlu (publik)

Request Body:

Field	Tipe	Wajib	Keterangan
name	string	Ya	Maks 255 karakter
email	string	Ya	Harus valid & unik
password	string	Ya	Minimal 8 karakter
phone_number	string	Tidak	Harus unik jika diisi

Contoh Request:

```
{
  "name": "Sari Dewi",
  "email": "sari@example.com",
  "password": "password123",
  "phone_number": "081234567890"
}
```

Response Berhasil (201):

```
{
  "message": "Registration successful",
  "access_token": "1|abc123def456...",
  "token_type": "Bearer",
  "user": {
    "id": 5,
    "name": "Sari Dewi",
    "email": "sari@example.com",
    "phone_number": "081234567890",
    "role": "user",
    "created_at": "2026-02-17T10:00:00.000000Z",
    "updated_at": "2026-02-17T10:00:00.000000Z"
  }
}
```

Catatan:

- Pengguna baru selalu mendapat peran `user`
- Password otomatis di-hash
- Langsung mendapat token (tidak perlu login lagi)

2. Login

```
POST /api/login
```

Autentikasi: Tidak perlu (publik)

Request Body:

Field	Tipe	Wajib	Keterangan
email	string	Ya	Email terdaftar
password	string	Ya	Password akun

Contoh Request:

```
{
  "email": "sari@example.com",
  "password": "password123"
}
```

Response Berhasil (200):

```
{
  "message": "Login successful",
  "access_token": "2|xyz789...",
  "token_type": "Bearer",
  "user": {
    "id": 5,
    "name": "Sari Dewi",
    "email": "sari@example.com",
    "phone_number": "081234567890",
    "role": "user",
    "instructor": null
  }
}
```

Catatan:

- Jika user berperan `instructor`, field `instructor` berisi data profil instruktur
- Jika email/password salah, response 422 dengan pesan error

3. Logout

POST `/api/logout`

Autentikasi: Bearer Token (wajib)

Request Body: Kosong

Response Berhasil (200):

```
{
  "message": "Logout successfully!"
}
```

Catatan: Token yang digunakan akan dicabut dan tidak bisa dipakai lagi.

Profil Pengguna

4. Lihat Profil Sendiri

GET `/api/profile`

Autentikasi: Bearer Token (wajib) **Akses:** Semua peran

Response Berhasil (200):

```
{
  "id": 5,
  "name": "Sari Dewi",
  "email": "sari@example.com",
  "phone_number": "081234567890",
  "role": "user",
  "created_at": "2026-02-17T10:00:00.000000Z",
  "updated_at": "2026-02-17T10:00:00.000000Z"
}
```

5. Perbarui Profil Sendiri

PUT `/api/profile`

Autentikasi: Bearer Token (wajib) **Akses:** `user`, `instructor`, `super_admin` — `admin` diblokir (403)

Request Body:

Field	Tipe	Wajib	Keterangan
name	string	Tidak	Maks 255 karakter
email	string	Tidak	Harus valid & unik (selain milik sendiri)
phone_number	string	Tidak	Harus unik (selain milik sendiri)
password	string	Tidak	Minimal 8 karakter

Contoh Request:

```
{
  "name": "Sari Dewi Updated",
  "phone_number": "089876543210"
}
```

Response Berhasil (200):

```
{
  "message": "Profile updated successfully.",
  "user": {
    "id": 5,
    "name": "Sari Dewi Updated",
    "email": "sari@example.com",
    "phone_number": "089876543210",
    "role": "user",
    "created_at": "2026-02-17T10:00:00.000000Z",
    "updated_at": "2026-02-17T11:00:00.000000Z"
  }
}
```

Catatan:

- Hanya field yang dikirim yang akan diperbarui (partial update)
- Admin tidak diizinkan memperbarui profil lewat endpoint ini (403)

6. Hapus Akun Sendiri

```
DELETE /api/profile
```

Autentikasi: Bearer Token (wajib) Akses: Hanya peran user

Response Berhasil (200):

```
{
  "message": "Account deleted successfully."
}
```

Response Gagal (403):

```
{
  "message": "Admin and super_admin accounts cannot be deleted through this endpoint."
}
```

Catatan:

- Hanya user biasa yang bisa hapus akunnya sendiri
- Super admin dan admin mendapat 403
- Data menggunakan soft delete (tidak benar-benar terhapus dari database)
- Semua token dicabut sebelum penghapusan

Manajemen Pengguna (Admin)

Semua endpoint di bagian ini membutuhkan middleware `auth:sanctum` + `role:admin,super_admin`

7. Daftar Semua Pengguna

```
GET /api/users
```

Autentikasi: Bearer Token (wajib) **Akses:** `admin`, `super_admin`

Parameter Query:

Parameter	Tipe	Default	Keterangan
<code>per_page</code>	integer	15	Jumlah data per halaman
<code>role</code>	string	—	Filter berdasarkan peran: <code>super_admin</code> , <code>admin</code> , <code>instructor</code> , <code>user</code>

Contoh Request:

```
GET /api/users?role=instructor&per_page=10
```

Response Berhasil (200):

```
{
  "data": [
    {
      "id": 3,
      "name": "Instructor 1",
      "email": "instructor@example.com",
      "phone_number": "081111111111",
      "role": "instructor",
      "created_at": "2026-02-17T10:00:00.000000Z",
      "updated_at": "2026-02-17T10:00:00.000000Z"
    }
  ],
  "meta": {
    "current_page": 1,
    "from": 1,
    "last_page": 1,
    "per_page": 10,
    "to": 1,
    "total": 1
  }
}
```

8. Detail Pengguna

```
GET /api/users/{id}
```

Autentikasi: Bearer Token (wajib) **Akses:** `admin`, `super_admin`

Parameter URL:

Parameter	Tipe	Keterangan
<code>id</code>	integer	ID pengguna

Response Berhasil (200):

```
{
  "id": 3,
  "name": "Instructor 1",
  "email": "instructor@example.com",
  "phone_number": "08111111111",
  "role": "instructor",
  "created_at": "2026-02-17T10:00:00.000000Z",
  "updated_at": "2026-02-17T10:00:00.000000Z"
}
```

9. Buat Akun Pengguna

POST /api/users

Autentikasi: Bearer Token (wajib) **Akses:** admin, super_admin

Request Body:

Field	Tipe	Wajib	Keterangan
name	string	Ya	Maks 255 karakter
email	string	Ya	Harus valid & unik
password	string	Ya	Minimal 8 karakter
phone_number	string	Ya	Harus unik
role	string	Ya	Salah satu: admin, super_admin, instructor

Contoh Request:

```
{
  "name": "Admin Baru",
  "email": "admin.baru@example.com",
  "password": "password123",
  "phone_number": "08222222222",
  "role": "admin"
}
```

Response Berhasil (201):

```
{
  "id": 6,
  "name": "Admin Baru",
  "email": "admin.baru@example.com",
  "phone_number": "08222222222",
  "role": "admin",
  "created_at": "2026-02-17T12:00:00.000000Z",
  "updated_at": "2026-02-17T12:00:00.000000Z"
}
```

Aturan Otorisasi:

- Hanya **super admin** yang bisa membuat akun **super_admin** (admin biasa mendapat 403)
- Tidak bisa membuat akun **user** biasa (gunakan /api/register)

10. Perbarui Pengguna

PUT /api/users/{id}

Autentikasi: Bearer Token (wajib) Akses: admin , super_admin

Parameter URL:

Parameter	Tipe	Keterangan
id	integer	ID pengguna yang akan diperbarui

Request Body:

Field	Tipe	Wajib	Keterangan
name	string	Tidak	Maks 255 karakter
email	string	Tidak	Harus valid & unik (selain milik target)
phone_number	string	Tidak	Harus unik (selain milik target)
role	string	Tidak	Salah satu: admin , super_admin , instructor , user
password	string	Tidak	Minimal 8 karakter

Response Berhasil (200): Data pengguna yang sudah diperbarui.

Aturan Otorisasi (Hierarki Ketat):

Aturan	Kode Status
Tidak bisa mengubah Master Account (ID 1) kecuali oleh dirinya sendiri	403
Admin biasa tidak bisa mengubah admin/super admin lain	403
Admin biasa tidak bisa mempromosikan user ke admin/super admin	403
Hanya super admin yang bisa mengubah peran ke admin/super admin	403

11. Hapus Pengguna

```
DELETE /api/users/{id}
```

Autentikasi: Bearer Token (wajib) Akses: admin , super_admin

Parameter URL:

Parameter	Tipe	Keterangan
id	integer	ID pengguna yang akan dihapus

Response Berhasil (200):

```
{
  "message": "User deleted successfully"
}
```

Aturan Otorisasi (Hierarki Ketat):

Aturan	Kode Status
Tidak bisa menghapus diri sendiri	400
Tidak bisa menghapus Master Account (ID 1)	403
Admin biasa tidak bisa menghapus admin/super admin	

lain Aturan	403 Kode Status
Hanya super admin yang bisa menghapus admin lain	—

Manajemen Layanan

12. Daftar Semua Layanan

GET /api/services

Autentikasi: Tidak perlu (publik)

Parameter Query:

Parameter	Tipe	Default	Keterangan
per_page	integer	10	Jumlah data per halaman

Response Berhasil (200):

```
{
  "data": [
    {
      "id": 1,
      "name": "Makeup Natural",
      "description": "Kelas makeup natural untuk sehari-hari",
      "price": 350000.00,
      "duration_minutes": 120,
      "image": "https://res.cloudinary.com/xxx/image/upload/v123/services/abc.jpg",
      "created_at": "2026-02-17T10:00:00.000000Z",
      "updated_at": "2026-02-17T10:00:00.000000Z",
      "deleted_at": null,
      "instructors": [
        {
          "id": 1,
          "user_id": 3,
          "service_id": 1,
          "bio": "Makeup artist berpengalaman 5 tahun",
          "photo": "https://res.cloudinary.com/xxx/image/upload/v123/instructors/def.jpg"
        }
      ]
    }
  ],
  "meta": { ... }
}
```

Catatan: Menyertakan daftar instruktur untuk setiap layanan.

13. Detail Layanan

GET /api/services/{id}

Autentikasi: Tidak perlu (publik)

Parameter URL:

Parameter	Tipe	Keterangan
id	integer	ID layanan

Response Berhasil (200):

```
{
  "id": 1,
  "name": "Makeup Natural",
  "description": "Kelas makeup natural untuk sehari-hari",
  "price": 350000.00,
  "duration_minutes": 120,
  "image": "https://res.cloudinary.com/xxx/image/upload/v123/services/abc.jpg",
  "instructors": [ ... ],
  "schedules": [
    {
      "id": 1,
      "service_id": 1,
      "instructor_id": 1,
      "start_time": "2026-02-20T09:00:00.000000Z",
      "end_time": "2026-02-20T11:00:00.000000Z",
      "total_capacity": 10,
      "remaining_slots": 7
    }
  ]
}
```

Catatan: Menyertakan instruktur **dan** jadwal untuk detail lengkap.

14. Buat Layanan Baru

POST /api/services

Autentikasi: Bearer Token (wajib) **Akses:** admin, super_admin **Content-Type:** multipart/form-data (jika upload gambar)

Request Body:

Field	Tipe	Wajib	Keterangan
name	string	Ya	Maks 255 karakter
description	string	Ya	Deskripsi layanan
price	numeric	Ya	Harga layanan
duration_minutes	integer	Ya	Durasi dalam menit
image	file	Tidak	Format: jpeg, png, jpg, gif. Maks 5MB

Response Berhasil (201):

```
{
  "id": 2,
  "name": "Bridal Makeup",
  "description": "Kelas makeup pengantin profesional",
  "price": 500000.00,
  "duration_minutes": 180,
  "image": "https://res.cloudinary.com/xxx/image/upload/v123/services/ghi.jpg",
  "created_at": "2026-02-17T12:00:00.000000Z",
  "updated_at": "2026-02-17T12:00:00.000000Z"
}
```

Catatan:

- Gambar diupload ke Cloudinary folder services/
- URL yang disimpan adalah secure_url dari Cloudinary
- Jika upload gagal, response 500 dengan pesan error

15. Perbarui Layanan

```
PUT /api/services/{id}
```

Autentikasi: Bearer Token (wajib) **Akses:** admin , super_admin **Content-Type:** multipart/form-data (jika upload gambar baru)

Parameter URL:

Parameter	Tipe	Keterangan
id	integer	ID layanan

Request Body:

Field	Tipe	Wajib	Keterangan
name	string	Tidak	Maks 255 karakter
description	string	Tidak	Deskripsi layanan
price	numeric	Tidak	Harga layanan
duration_minutes	integer	Tidak	Durasi dalam menit
image	file	Tidak	Format: jpeg, png, jpg, gif. Maks 5MB

Response Berhasil (200): Data layanan yang sudah diperbarui.

Catatan: Gambar baru akan menggantikan gambar lama di Cloudinary.

16. Hapus Layanan

```
DELETE /api/services/{id}
```

Autentikasi: Bearer Token (wajib) **Akses:** admin , super_admin

Response Berhasil (200):

```
{
  "message": "Service deleted successfully"
}
```

Catatan: Menggunakan soft delete (data tidak benar-benar terhapus).

Manajemen Instruktur

17. Daftar Semua Instruktur

```
GET /api/instructors
```

Autentikasi: Tidak perlu (publik)

Parameter Query:

Parameter	Tipe	Default	Keterangan
per_page	integer	10	Jumlah data per halaman

Response Berhasil (200):

```
{
  "data": [
    {
      "id": 1,
      "user_id": 3,
      "service_id": 1,
      "bio": "Makeup artist berpengalaman 5 tahun",
      "photo": "https://res.cloudinary.com/xxx/image/upload/v123/instructors/def.jpg",
      "created_at": "2026-02-17T10:00:00.000000Z",
      "updated_at": "2026-02-17T10:00:00.000000Z",
      "deleted_at": null,
      "service": {
        "id": 1,
        "name": "Makeup Natural",
        "description": "Kelas makeup natural",
        "price": 350000.00,
        "duration_minutes": 120,
        "image": "...",
      },
      "user": {
        "id": 3,
        "name": "Instructor 1",
        "email": "instructor@example.com",
        "phone_number": "081111111111",
        "role": "instructor"
      }
    }
  ],
  "meta": { ... }
}
```

Catatan: Menyertakan relasi `service` dan `user` (untuk nama instruktur).

18. Detail Instruktur

GET `/api/instructors/{id}`

Autentikasi: Tidak perlu (publik)

Parameter URL:

Parameter	Tipe	Keterangan
<code>id</code>	integer	ID instruktur

Response Berhasil (200):

```
{
  "id": 1,
  "user_id": 3,
  "service_id": 1,
  "bio": "Makeup artist berpengalaman 5 tahun",
  "photo": "https://res.cloudinary.com/xxx/image/upload/v123/instructors/def.jpg",
  "service": { ... },
  "user": { ... },
  "schedules": [
    {
      "id": 1,
      "service_id": 1,
      "instructor_id": 1,
      "start_time": "2026-02-20T09:00:00.000000Z",
      "end_time": "2026-02-20T11:00:00.000000Z",
      "total_capacity": 10,
      "remaining_slots": 7
    }
  ]
}
```

Catatan: Menyertakan `service`, `user`, dan seluruh `schedules`.

19. Buat Instruktur Baru

POST /api/instructors

Autentikasi: Bearer Token (wajib) **Akses:** `admin`, `super_admin` **Content-Type:** `multipart/form-data` (jika upload foto)

Request Body:

Field	Tipe	Wajib	Keterangan
<code>name</code>	string	Ya	Maks 255 karakter
<code>email</code>	string	Ya	Harus valid & unik
<code>password</code>	string	Ya	Minimal 6 karakter
<code>phone_number</code>	string	Ya	Maks 20 karakter
<code>service_id</code>	integer	Ya	ID layanan yang diajarkan (harus ada di tabel <code>services</code>)
<code>bio</code>	string	Ya	Biografi instruktur
<code>photo</code>	file	Tidak	Format: jpeg, png, jpg, gif. Maks 5MB

Contoh Request:

```
{
  "name": "Dewi Pratiwi",
  "email": "dewi@example.com",
  "password": "password123",
  "phone_number": "083333333333",
  "service_id": 1,
  "bio": "Makeup artist profesional dengan pengalaman 5 tahun"
}
```

Response Berhasil (201):

```
{
  "id": 2,
  "user_id": 7,
  "service_id": 1,
  "bio": "Makeup artist profesional dengan pengalaman 5 tahun",
  "photo": null,
  "created_at": "2026-02-17T12:00:00.000000Z",
  "updated_at": "2026-02-17T12:00:00.000000Z",
  "user": {
    "id": 7,
    "name": "Dewi Pratiwi",
    "email": "dewi@example.com",
    "phone_number": "083333333333",
    "role": "instructor"
  },
  "service": {
    "id": 1,
    "name": "Makeup Natural",
    "description": "...",
    "price": 350000.00,
    "duration_minutes": 120
  }
}
```

Catatan:

- Endpoint ini membuat **2 record sekaligus**: akun User (peran `instructor`) + profil Instructor
- Foto diupload ke Cloudinary folder `instructors/`
- Jika upload foto gagal, instruktur tetap dibuat tanpa foto

20. Perbarui Instruktur

PUT /api/instructors/{id}

Autentikasi: Bearer Token (wajib) Akses: `admin`, `super_admin` Content-Type: `multipart/form-data` (jika upload foto baru)

Parameter URL:

Parameter	Tipe	Keterangan
<code>id</code>	integer	ID instruktur

Request Body:

Field	Tipe	Wajib	Keterangan
<code>name</code>	string	Tidak	Maks 255 karakter
<code>email</code>	string	Tidak	Harus valid & unik (selain milik instruktur ini)
<code>phone_number</code>	string	Tidak	Maks 20 karakter
<code>service_id</code>	integer	Tidak	ID layanan (harus ada di tabel services)
<code>bio</code>	string	Tidak	Biografi instruktur
<code>photo</code>	file	Tidak	Format: jpeg, png, jpg, gif. Maks 5MB

Response Berhasil (200): Data instruktur yang sudah diperbarui (termasuk relasi `user` dan `service`).

Catatan: Memperbarui record User **dan** Instructor sekaligus.

21. Hapus Instruktur

```
DELETE /api/instructors/{id}
```

Autentikasi: Bearer Token (wajib) **Akses:** admin, super_admin

Response Berhasil (200):

```
{
  "message": "Instructor deleted successfully"
}
```

Catatan: Soft delete pada record instruktur saja (akun user tidak dihapus).

Manajemen Jadwal

22. Daftar Jadwal (Sadar Konteks)

```
GET /api/schedules
```

Autentikasi: Opsional (publik atau terotentikasi)

Parameter Query:

Parameter	Tipe	Default	Keterangan
per_page	integer	15	Jumlah data per halaman
instructor_id	integer	—	Filter berdasarkan instruktur (untuk publik/user/admin)

Response Berhasil (200):

```
{
  "data": [
    {
      "id": 1,
      "service_id": 1,
      "instructor_id": 1,
      "start_time": "2026-02-20T09:00:00.000000Z",
      "end_time": "2026-02-20T11:00:00.000000Z",
      "total_capacity": 10,
      "remaining_slots": 7,
      "created_at": "2026-02-17T10:00:00.000000Z",
      "updated_at": "2026-02-17T10:00:00.000000Z",
      "deleted_at": null,
      "service": {
        "id": 1,
        "name": "Makeup Natural",
        "description": "...",
        "price": 350000.00,
        "duration_minutes": 120
      },
      "instructor": {
        "id": 1,
        "user_id": 3,
        "service_id": 1,
        "bio": "...",
        "photo": "...",
        "user": {
          "id": 3,
          "name": "Instructor 1",
          "email": "instructor@example.com",
          "phone_number": "081111111111",
          "role": "instructor"
        }
      }
    }
  ],
  "meta": { ... }
}
```

Perbedaan Data Berdasarkan Peran:

Peran	Data yang Ditampilkan
Publik / User	Hanya jadwal yang akan datang (<code>start_time > sekarang</code>). Bisa filter ? <code>instructor_id</code>
Instruktur	Hanya jadwal milik sendiri (lalu, sekarang, dan akan datang)
Admin / Super Admin	Semua jadwal tanpa filter waktu. Bisa filter ? <code>instructor_id</code>

23. Detail Jadwal

GET /api/schedules/{id}

Autentikasi: Tidak perlu (publik)

Parameter URL:

Parameter	Tipe	Keterangan
id	integer	ID jadwal

Response Berhasil (200): Data jadwal lengkap dengan relasi service dan instructor (termasuk instructor.user).

24. Buat Jadwal Baru

```
POST /api/schedules
```

Autentikasi: Bearer Token (wajib) Akses: admin, super_admin

Request Body:

Field	Tipe	Wajib	Keterangan
service_id	integer	Ya	ID layanan (harus ada di tabel services)
instructor_id	integer	Ya	ID instruktur (harus ada di tabel instructors)
start_time	datetime	Ya	Format ISO 8601, harus setelah waktu sekarang
end_time	datetime	Ya	Harus setelah start_time
total_capacity	integer	Ya	Minimal 1

Contoh Request:

```
{
  "service_id": 1,
  "instructor_id": 1,
  "start_time": "2026-02-20T09:00:00",
  "end_time": "2026-02-20T11:00:00",
  "total_capacity": 10
}
```

Response Berhasil (201): Data jadwal yang dibuat (termasuk relasi service dan instructor).

Validasi Tambahan:

Validasi	Kode Status	Pesan
Instruktur harus mengajar layanan tersebut (instructor.service_id == service_id)	422	"The selected instructor does not teach this service"
Jadwal tidak boleh overlap dengan jadwal instruktur yang sudah ada	422	"Instructor already has a schedule that overlaps..."

Catatan:

- remaining_slots otomatis diisi sama dengan total_capacity
- Field capacity juga diterima sebagai alias untuk total_capacity

25. Perbarui Jadwal

```
PUT /api/schedules/{id}
```

Autentikasi: Bearer Token (wajib) Akses: admin, super_admin

Parameter URL:

Parameter	Tipe	Keterangan
id	integer	ID jadwal

Request Body:

Field	Tipe	Wajib	Keterangan
start_time	datetime	Tidak	Harus setelah waktu sekarang
end_time	datetime	Tidak	Harus setelah start_time
total_capacity	integer	Tidak	Minimal 1
instructor_id	integer	Tidak	ID instruktur (harus ada)

Response Berhasil (200): Data jadwal yang sudah diperbarui.

Validasi Tambahan:

Validasi	Kode Status
Kapasitas baru tidak boleh kurang dari jumlah booking aktif	400
Pengecekan overlap (tidak termasuk jadwal ini sendiri)	422

Catatan: remaining_slots dihitung ulang: kapasitas_baru - jumlah_booking_aktif

26. Hapus Jadwal

```
DELETE /api/schedules/{id}
```

Autentikasi: Bearer Token (wajib) **Akses:** admin , super_admin

Response Berhasil (200):

```
{
  "message": "Schedule deleted successfully"
}
```

Response Gagal (400):

```
{
  "message": "Cannot delete schedule with active bookings. Please cancel all bookings first."
}
```

Catatan:

- Tidak bisa dihapus jika masih ada booking aktif (status bukan cancelled)
- Menggunakan soft delete

Manajemen Booking

27. Daftar Booking (Sadar Konteks)

```
GET /api/bookings
```

Autentikasi: Bearer Token (wajib)

Parameter Query:

Parameter	Tipe	Default	Keterangan
per_page	integer	15	Jumlah data per halaman
user_id	integer	—	Filter berdasarkan user (hanya admin)

Perbedaan Data Berdasarkan Peran:

Peran	Data yang Ditampilkan
User	Hanya booking milik sendiri
Admin / Super Admin	Semua booking . Bisa filter ? user_id

Response Berhasil (200):

```

{
  "data": [
    {
      "id": 1,
      "user_id": 5,
      "schedule_id": 1,
      "booking_code": "BA-ABCD1234",
      "status": "confirmed",
      "total_price": 350000.00,
      "booking_date": "2026-02-17",
      "created_at": "2026-02-17T10:00:00.000000Z",
      "updated_at": "2026-02-17T10:30:00.000000Z",
      "deleted_at": null,
      "user": {
        "id": 5,
        "name": "Sari Dewi",
        "email": "sari@example.com",
        "phone_number": "081234567890"
      },
      "schedule": {
        "id": 1,
        "service_id": 1,
        "instructor_id": 1,
        "start_time": "2026-02-20T09:00:00.000000Z",
        "end_time": "2026-02-20T11:00:00.000000Z",
        "total_capacity": 10,
        "remaining_slots": 7,
        "service": {
          "id": 1,
          "name": "Makeup Natural",
          "price": 350000.00
        },
        "instructor": {
          "id": 1,
          "user_id": 3,
          "bio": "...",
          "photo": "...",
          "user": {
            "id": 3,
            "name": "Instructor 1"
          }
        }
      },
      "payment": {
        "id": 1,
        "booking_id": 1,
        "transaction_id": "txn-123",
        "payment_type": "gopay",
        "gross_amount": 350000.00,
        "transaction_status": "settlement",
        "fraud_status": null,
        "snap_token": "snap-abc..."
      },
      "review": null
    }
  ],
  "meta": { ... }
}

```

Catatan: Menyertakan relasi lengkap: user, schedule (+ service + instructor + instructor.user), payment, dan review.

28. Buat Booking Baru

POST /api/bookings

Autentikasi: Bearer Token (wajib)

Request Body:

Field	Tipe	Wajib	Keterangan
schedule_id	integer	Ya	ID jadwal yang ingin di-booking (harus ada di tabel schedules)

Contoh Request:

```
{
  "schedule_id": 1
}
```

Response Berhasil (201):

```
{
  "id": 2,
  "user_id": 5,
  "schedule_id": 1,
  "booking_code": "BA-EFGH5678",
  "status": "pending",
  "total_price": 350000.00,
  "booking_date": "2026-02-17",
  "created_at": "2026-02-17T13:00:00.000000Z",
  "updated_at": "2026-02-17T13:00:00.000000Z",
  "schedule": {
    "id": 1,
    "service_id": 1,
    "instructor_id": 1,
    "start_time": "2026-02-20T09:00:00.000000Z",
    "end_time": "2026-02-20T11:00:00.000000Z",
    "total_capacity": 10,
    "remaining_slots": 6,
    "service": {
      "id": 1,
      "name": "Makeup Natural",
      "price": 350000.00
    }
  },
  "payment": {
    "id": 2,
    "booking_id": 2,
    "transaction_id": null,
    "payment_type": "tbd",
    "gross_amount": 350000.00,
    "transaction_status": "pending",
    "fraud_status": null,
    "snap_token": null
  }
}
```

Validasi & Logika Bisnis:

Validasi	Kode Status	Pesan
Slot habis (remaining_slots = 0)	400	"No available slots for this schedule"
User sudah booking jadwal yang sama (aktif)	400	"You already have an active booking for this schedule"

Mekanisme Keamanan:

- **Pessimistic locking** (`lockForUpdate()`) dalam `DB::transaction()` untuk mencegah race condition saat banyak user booking slot terakhir secara bersamaan
- Operasi atomik: `remaining_slots` dikurangi, booking dibuat, payment record dibuat — semua dalam satu transaksi

Catatan:

- Status awal booking: `pending`
- `total_price` diambil dari `schedule.service.price`
- `booking_code` format: BA-XXXXXXX (8 huruf kapital acak)
- Payment record otomatis dibuat dengan `payment_type`: `"tbd"` dan `transaction_status`: `"pending"`

29. Batalkan Booking

```
POST /api/bookings/{id}/cancel
```

Autentikasi: Bearer Token (wajib)

Parameter URL:

Parameter	Tipe	Keterangan
<code>id</code>	integer	ID booking

Request Body: Kosong

Response Berhasil (200):

```
{
  "message": "Booking cancelled successfully"
}
```

Validasi & Otorisasi:

Validasi	Kode Status	Pesan
Bukan pemilik booking dan bukan admin	403	"Unauthorized to cancel this booking"
Booking sudah dibatalkan sebelumnya	400	"Booking is already cancelled"

Operasi Atomik (dalam transaksi):

1. Status booking diubah ke `cancelled`
2. `remaining_slots` jadwal ditambah kembali (slot dikembalikan)
3. Status payment diubah ke `cancel` (jika ada record payment)

Manajemen Review

30. Daftar Testimonial (Publik)

```
GET /api/testimonials
```

Autentikasi: Tidak perlu (publik)

Parameter Query:

Parameter	Tipe	Default	Keterangan
<code>per_page</code>	integer	10	Jumlah data per halaman
<code>instructor_id</code>	integer	—	Filter review berdasarkan instruktur

Response Berhasil (200):

```
{
  "data": [
    {
      "id": 1,
      "booking_id": 1,
      "rating": 5,
      "comment": "Kelasnya sangat bagus dan informatif!",
      "created_at": "2026-02-17T15:00:00.000000Z",
      "updated_at": "2026-02-17T15:00:00.000000Z",
      "booking": {
        "id": 1,
        "user_id": 5,
        "schedule_id": 1,
        "booking_code": "BA-ABCD1234",
        "status": "confirmed",
        "total_price": 350000.00,
        "user": {
          "id": 5,
          "name": "Sari Dewi"
        },
        "schedule": {
          "id": 1,
          "start_time": "2026-02-20T09:00:00.000000Z",
          "end_time": "2026-02-20T11:00:00.000000Z",
          "service": {
            "id": 1,
            "name": "Makeup Natural"
          },
          "instructor": {
            "id": 1,
            "user": {
              "id": 3,
              "name": "Instructor 1"
            }
          }
        }
      }
    }
  ],
  "meta": { ... }
}
```

Catatan: Endpoint ini untuk halaman utama (homepage testimonials).

31. Daftar Review (Sadar Konteks)

GET /api/reviews

Autentikasi: Bearer Token (wajib)

Parameter Query:

Parameter	Tipe	Default	Keterangan
per_page	integer	15	Jumlah data per halaman
instructor_id	integer	—	Filter berdasarkan instruktur (untuk admin)

Perbedaan Data Berdasarkan Peran:

Peran	Data yang Ditampilkan
User	Hanya review dari booking milik sendiri

Peran Instruktur	Data yang Ditampilkan: Hanya review untuk kelas miliknya (404 jika belum punya profil instruktur)
Admin / Super Admin	Semua review. Bisa filter ?instructor_id

Response Berhasil (200): Sama seperti format testimonial, dengan relasi booking → user, schedule → service, instructor .

32. Buat Review

POST /api/reviews

Autentikasi: Bearer Token (wajib)

Request Body:

Field	Tipe	Wajib	Keterangan
booking_id	integer	Ya	ID booking yang ingin di-review (harus ada)
rating	integer	Ya	Nilai 1 sampai 5
comment	string	Tidak	Komentar review

Contoh Request:

```
{
  "booking_id": 1,
  "rating": 5,
  "comment": "Kelasnya sangat bagus dan informatif!"
}
```

Response Berhasil (201):

```
{
  "id": 1,
  "booking_id": 1,
  "rating": 5,
  "comment": "Kelasnya sangat bagus dan informatif!",
  "created_at": "2026-02-17T15:00:00.000000Z",
  "updated_at": "2026-02-17T15:00:00.000000Z"
}
```

Validasi & Otorisasi:

Validasi	Kode Status	Pesan
Booking bukan milik user yang login	403	"You can only review your own bookings"
Kelas belum selesai (end_time > sekarang)	400	"You can only review after the class has ended"
Booking sudah pernah di-review	400	"You have already reviewed this booking"

33. Perbarui Review

PUT /api/reviews/{id}

Autentikasi: Bearer Token (wajib) **Akses:** Pemilik review saja

Parameter URL:

Parameter	Tipe	Keterangan
id	integer	ID review

Request Body:

Field	Tipe	Wajib	Keterangan
rating	integer	Tidak	Nilai 1 sampai 5
comment	string	Tidak	Komentar review (bisa null)

Response Berhasil (200): Data review yang sudah diperbarui.

Otorisasi: Hanya pemilik booking terkait yang bisa memperbarui (403 jika bukan pemilik).

34. Hapus Review

```
DELETE /api/reviews/{id}
```

Autentikasi: Bearer Token (wajib) Akses: Pemilik review atau admin / super_admin

Parameter URL:

Parameter	Tipe	Keterangan
id	integer	ID review

Response Berhasil (200):

```
{
  "message": "Review deleted successfully"
}
```

Otorisasi: Pemilik booking terkait atau admin/super admin (403 jika tidak memenuhi).

Pembayaran (Midtrans)

35. Dapatkan Snap Token

```
GET /api/payments/snap-token/{booking_id}
```

Autentikasi: Bearer Token (wajib) Akses: Pemilik booking saja

Parameter URL:

Parameter	Tipe	Keterangan
booking_id	integer	ID booking

Response Berhasil (200):

```
{
  "snap_token": "66e4fa55-fdac-4ef9-91b5-733b97d1b862",
  "client_key": "Mid-client-xxxxxxxxxxx"
}
```

Validasi & Otorisasi:

Validasi	Kode Status	Pesan
Booking bukan milik user yang login	403	"Unauthorized"

Validasi Booking sudah dikonfirmasi/dibayar	Kode Status 400	Pesan "Booking is already confirmed"
Tidak ada record payment	500	"Payment record not found"
Gagal generate token dari Midtrans	500	Pesan error dari Midtrans

Catatan:

- Token digunakan untuk menampilkan halaman pembayaran Midtrans Snap di frontend
- `client_key` dibutuhkan oleh Midtrans JS library di frontend
- Format `order_id` yang dikirim ke Midtrans: `{booking_code}-{timestamp}`
- Snap token disimpan di record payment untuk referensi

Payload yang Dikirim ke Midtrans:

```
{
  "transaction_details": {
    "order_id": "BA-ABCD1234-1708012345",
    "gross_amount": 350000
  },
  "customer_details": {
    "first_name": "Sari Dewi",
    "email": "sari@example.com",
    "phone": "081234567890"
  },
  "item_details": [
    {
      "id": 1,
      "price": 350000,
      "quantity": 1,
      "name": "Makeup Class: Makeup Natural"
    }
  ],
  "enabled_payments": ["credit_card", "gopay", "shopeepay", "other_qris", "bca_va", "bni_va", "bri_va", "permata_va"]
}
```

36. Callback / Webhook Midtrans

```
POST /api/payments/callback
```

Autentikasi: Tidak perlu (publik) — diverifikasi lewat signature **Pemanggil:** Server Midtrans (bukan user/frontend)

Request Body (dari Midtrans):

Field	Tipe	Keterangan
<code>transaction_id</code>	string	ID transaksi Midtrans
<code>order_id</code>	string	Format: <code>BA-xxxxxxx-{timestamp}</code>
<code>status_code</code>	string	Kode status HTTP (200, 201, 202, dll.)
<code>gross_amount</code>	string	Jumlah pembayaran
<code>transaction_status</code>	string	Status: <code>pending</code> , <code>capture</code> , <code>settlement</code> , <code>deny</code> , <code>expire</code> , <code>cancel</code> , <code>challenge</code>
<code>payment_type</code>	string	Metode bayar: <code>credit_card</code> , <code>bank_transfer</code> , <code>gopay</code> , dll.
<code>fraud_status</code>	string/null	Status fraud: <code>challenge</code> , <code>accept</code> , <code>deny</code>
<code>signature_key</code>	string	Hash SHA512 untuk verifikasi

Response Berhasil (200):

```
{
  "message": "Callback processed"
}
```

Response Gagal:

Kode Status	Keterangan
403	Signature tidak valid
404	Booking tidak ditemukan

Verifikasi Signature:

```
SHA512(order_id + status_code + gross_amount + server_key) == signature_key
```

Pemetaan Status Pembayaran:

Status Midtrans	Status Payment	Status Booking
capture + fraud:accept	settlement	confirmed
settlement	settlement	confirmed
capture + fraud:challenge	challenge	— (tidak berubah)
pending	pending	— (tidak berubah)
deny	deny	— (tidak berubah)
expire	expire	cancelled
cancel	cancel	cancelled

Dashboard & Laporan Admin

Semua endpoint di bagian ini membutuhkan middleware `auth:sanctum` + `role:admin,super_admin`

37. Statistik Dashboard

```
GET /api/admin/dashboard/stats
```

Autentikasi: Bearer Token (wajib) **Akses:** `admin`, `super_admin`

Response Berhasil (200):

```
{
  "stats": {
    "total_users": 25,
    "total_instructors": 5,
    "upcoming_classes": 12,
    "total_bookings": 150,
    "pending_bookings": 10,
    "confirmed_bookings": 130,
    "total_revenue": 45500000.00
  },
  "recent_bookings": [
    {
      "id": 150,
      "booking_code": "BA-WXYZ9012",
      "status": "confirmed",
      "total_price": 350000.00,
      "booking_date": "2026-02-17",
      "user": {
        "id": 5,
        "name": "Sari Dewi",
        "email": "sari@example.com"
      },
      "schedule": {
        "id": 1,
        "service": {
          "id": 1,
          "name": "Makeup Natural"
        }
      },
      "payment": {
        "id": 150,
        "transaction_status": "settlement",
        "payment_type": "gopay"
      }
    }
  ],
  "popular_services": [
    {
      "name": "Makeup Natural",
      "total_bookings": 45
    },
    {
      "name": "Bridal Makeup",
      "total_bookings": 30
    },
    {
      "name": "Evening Look",
      "total_bookings": 25
    }
  ]
}
```

Perbedaan Berdasarkan Peran:

Data	Admin	Super Admin
total_revenue	null	Jumlah pendapatan dari booking confirmed
popular_services	[] (array kosong)	Top 3 layanan berdasarkan jumlah booking
recent_bookings	5 booking terakhir	5 booking terakhir

38. Laporan Pendapatan

```
GET /api/reports/revenue
```

Autentikasi: Bearer Token (wajib) Akses: admin, super_admin

Parameter Query:

Parameter	Tipe	Default	Keterangan
period	string	monthly	Periode: weekly, monthly, yearly

Response Berhasil (200):

```
{
  "chartData": [
    {
      "name": "Jan 2026",
      "revenue": 1500000.00
    },
    {
      "name": "Feb 2026",
      "revenue": 1250000.00
    }
  ],
  "totalRevenue": 4550000.00
}
```

Format Label Berdasarkan Periode:

Periode	Rentang Data	Format Label	Contoh
weekly	7 hari terakhir	D, dd M	Mon, 17 Feb
monthly	12 bulan terakhir	M Y	Feb 2026
yearly	5 tahun terakhir	YYYY	2026

Catatan: Hanya menghitung booking dengan status confirmed.

39. Laporan Performa Layanan

```
GET /api/reports/services-performance
```

Autentikasi: Bearer Token (wajib) Akses: admin, super_admin

Response Berhasil (200):

```
[
  {
    "name": "Makeup Natural",
    "value": 1575000.00,
    "bookings_count": 45
  },
  {
    "name": "Bridal Makeup",
    "value": 1500000.00,
    "bookings_count": 30
  }
]
```

Keterangan:

- name — Nama layanan

- `value` — Total pendapatan dari layanan (SUM `total_price` booking confirmed)
- `bookings_count` — Jumlah booking confirmed
- Diurutkan berdasarkan pendapatan (tertinggi dulu)

40. Laporan Statistik Operasional

GET `/api/reports/operational-stats`

Autentikasi: Bearer Token (wajib) **Akses:** `admin`, `super_admin`

Response Berhasil (200):

```
{
  "occupancyRate": 75.5,
  "cancellationRate": 8.3,
  "totalInstructors": 5,
  "totalUsers": 25,
  "activeClassesToday": 3
}
```

Keterangan:

Field	Perhitungan
<code>occupancyRate</code>	$(\text{total slot terisi} / \text{total kapasitas}) \times 100$ — dari kelas yang sudah selesai <code>end_time</code> < sekarang)
<code>cancellationRate</code>	$(\text{booking dibatalkan} / \text{total booking}) \times 100$
<code>totalInstructors</code>	Jumlah instruktur
<code>totalUsers</code>	Jumlah user dengan peran user
<code>activeClassesToday</code>	Jadwal yang <code>start_time</code> -nya hari ini

41. Laporan Performa Instruktur

GET `/api/reports/instructor-performance`

Autentikasi: Bearer Token (wajib) **Akses:** `admin`, `super_admin`

Response Berhasil (200):

```
[
  {
    "id": 1,
    "name": "Instructor 1",
    "totalClasses": 12,
    "totalBookings": 45,
    "totalRevenue": 15750000.00,
    "occupancyRate": 82.5
  },
  {
    "id": 2,
    "name": "Dewi Pratiwi",
    "totalClasses": 8,
    "totalBookings": 30,
    "totalRevenue": 15000000.00,
    "occupancyRate": 75.0
  }
]
```

Keterangan:

Field	Perhitungan
name	Nama instruktur (dari tabel users)
totalClasses	Jumlah jadwal yang dimiliki
totalBookings	Jumlah booking confirmed di jadwal-jadwalnya
totalRevenue	Total pendapatan dari booking confirmed
occupancyRate	Tingkat keterisian kelas instruktur (%)

Diurutkan berdasarkan totalRevenue (tertinggi dulu).

42. Laporan Jam Sibuk (Lalu Lintas Mingguan)

```
GET /api/reports/peak-hours
```

Autentikasi: Bearer Token (wajib) Akses: admin, super_admin

Response Berhasil (200):

```
[
  { "day": "Mon", "bookings": 25 },
  { "day": "Tue", "bookings": 18 },
  { "day": "Wed", "bookings": 30 },
  { "day": "Thu", "bookings": 22 },
  { "day": "Fri", "bookings": 35 },
  { "day": "Sat", "bookings": 40 },
  { "day": "Sun", "bookings": 15 }
]
```

Keterangan:

- Mengelompokkan booking confirmed berdasarkan hari dalam seminggu (start_time jadwal)
- Selalu mengembalikan 7 hari (Senin-Minggu), nilai 0 jika tidak ada booking
- Berguna untuk menentukan hari tersibuk

Endpoint Debug

43. Cek Status Autentikasi

```
GET /api/test-auth
```

Autentikasi: Bearer Token (wajib) Akses: Semua peran

Response Berhasil (200):

```
{
  "authenticated": true,
  "user": {
    "id": 5,
    "name": "Sari Dewi",
    "email": "sari@example.com",
    "phone_number": "081234567890",
    "role": "user"
  },
  "role": "user",
  "guard_check": true
}
```

Skema Database

Tabel Users

Kolom	Tipe	Keterangan
id	BIGINT (PK)	Auto increment
name	VARCHAR(255)	Nama pengguna
email	VARCHAR(255)	Unik
phone_number	VARCHAR(255)	Nullable, unik
email_verified_at	TIMESTAMP	Nullable
password	VARCHAR(255)	Di-hash otomatis
role	ENUM	super_admin, admin, instructor, user (default: user)
remember_token	VARCHAR(100)	Nullable
created_at	TIMESTAMP	—
updated_at	TIMESTAMP	—
deleted_at	TIMESTAMP	Soft delete

Tabel Services

Kolom	Tipe	Keterangan
id	BIGINT (PK)	Auto increment
name	VARCHAR(255)	Nama layanan
description	TEXT	Deskripsi layanan
price	DECIMAL(10,2)	Harga layanan
duration_minutes	INT	Durasi dalam menit
image	VARCHAR(255)	Nullable, URL Cloudinary
deleted_at	TIMESTAMP	Soft delete
created_at	TIMESTAMP	—
updated_at	TIMESTAMP	—

Tabel Instructors

Kolom	Tipe	Keterangan
id	BIGINT (PK)	Auto increment
user_id	BIGINT (FK → users)	Unik, cascade delete
	BIGINT (FK →	

service_id Kolom	services) Tipe	Cascade delete Keterangan
bio	TEXT	Biografi instruktur
photo	VARCHAR(255)	Nullable, URL Cloudinary
deleted_at	TIMESTAMP	Soft delete
created_at	TIMESTAMP	—
updated_at	TIMESTAMP	—

Tabel Schedules

Kolom	Tipe	Keterangan
id	BIGINT (PK)	Auto increment
service_id	BIGINT (FK → services)	Cascade delete
instructor_id	BIGINT (FK → instructors)	Cascade delete
start_time	DATETIME	Ter-index
end_time	DATETIME	—
total_capacity	INT	Kapasitas total
remaining_slots	INT	Slot tersisa
deleted_at	TIMESTAMP	Soft delete
created_at	TIMESTAMP	—
updated_at	TIMESTAMP	—

Tabel Bookings

Kolom	Tipe	Keterangan
id	BIGINT (PK)	Auto increment
user_id	BIGINT (FK → users)	Cascade delete
schedule_id	BIGINT (FK → schedules)	Cascade delete
booking_code	VARCHAR(255)	Unik, format: BA-XXXXXXX
status	ENUM	pending, confirmed, cancelled, finished
total_price	DECIMAL(10,2)	Harga saat booking
booking_date	DATE	Tanggal booking dibuat
deleted_at	TIMESTAMP	Soft delete
created_at	TIMESTAMP	—
updated_at	TIMESTAMP	—

Constraint: UNIQUE(user_id, schedule_id) — satu user tidak bisa booking jadwal yang sama dua kali.

Tabel Reviews

Kolom	Tipe	Keterangan
id	BIGINT (PK)	Auto increment
booking_id	BIGINT (FK → bookings)	Cascade delete
rating	TINYINT	Nilai 1–5
comment	TEXT	Nullable
created_at	TIMESTAMP	—
updated_at	TIMESTAMP	—

Tabel Payments

Kolom	Tipe	Keterangan
id	BIGINT (PK)	Auto increment
booking_id	BIGINT (FK → bookings)	Cascade delete
transaction_id	VARCHAR(255)	Nullable, unik. Dari webhook Midtrans
payment_type	VARCHAR(50)	Metode bayar (awal: tbd)
gross_amount	DECIMAL(10,2)	Jumlah pembayaran
transaction_status	VARCHAR(50)	pending , settlement , capture , deny , expire , cancel , challenge
fraud_status	VARCHAR(50)	Nullable: challenge , accept , deny
snap_token	VARCHAR(255)	Nullable. Token untuk Midtrans Snap
created_at	TIMESTAMP	—
updated_at	TIMESTAMP	—

Diagram Relasi

