

Perancangan Backend To-Do List Management for Social Media

1. Pendahuluan

To-Do List Management for Social Media adalah sistem manajemen tugas yang berfokus pada perencanaan, penjadwalan, dan publikasi postingan di berbagai platform sosial media. Sistem ini mendukung kerja tim dengan fitur kolaborasi, workflow approval, serta notifikasi.

2. Teknologi yang Digunakan

- Backend Framework: Next.js API Routes
- Database: PostgreSQL dengan Prisma ORM
- Autentikasi: JWT & OAuth (Google, Facebook)
- Job Scheduling: node-cron / Celery
- Notifikasi: Firebase Cloud Messaging / Twilio
- Dokumentasi API: Swagger UI

3. Rancangan API

3.1. User Registration & Login

Endpoint:

- **POST /register** – Mendaftarkan user dengan email/password atau OAuth.
- **POST /login** – Login user dan kembalikan token JWT.
- **GET /oauth/callback** – Handle callback dari OAuth (Google, Facebook).
- **POST /refresh-token** – Memperbarui token JWT yang hampir kadaluarsa.

Fungsi:

- Simpan token OAuth untuk akses API sosial media.
- Generate dan verifikasi JWT untuk autentikasi user.
- Refresh token untuk keamanan sesi.

3.2. Manajemen Tim & Akun Sosial Media

Endpoint:

- POST /teams – Membuat tim baru (hanya admin).
- GET /teams – Mendapatkan daftar tim yang user tergabung.
- POST /teams/:teamId/members – Menambahkan anggota ke tim.
- DELETE /teams/:teamId/members/****:userId – Menghapus anggota tim.
- POST /teams/:teamId/accounts – Menghubungkan akun sosial media ke tim.
- GET /teams/:teamId/accounts – Melihat daftar akun sosial media yang terhubung.

Fungsi:

- Role-based Access Control (RBAC) dengan Admin, Editor, dan Viewer.
 - Tim dapat mengelola postingan bersama.
 - Integrasi OAuth untuk mengakses API sosial media.
-

3.3. Membuat & Menjadwalkan Postingan

Endpoint:

- POST /posts – Membuat postingan baru.
- GET /posts – Mengambil daftar postingan (bisa difilter berdasarkan status, tim, dll.).
- PUT /posts/****:postId – Mengedit postingan sebelum dijadwalkan.
- DELETE /posts/****:postId – Membatalkan postingan sebelum dipublikasikan.
- POST /posts/schedule – Menjadwalkan postingan pada waktu tertentu.
- POST /posts/preview – Melihat tampilan postingan sebelum dipublikasikan.
- POST /posts/repost/****:postId – Repost otomatis di waktu tertentu.

Fungsi:

- Menyimpan postingan dan jadwalnya di database.
- Job scheduler memastikan postingan dipublikasikan sesuai waktu yang ditentukan.

- Preview posting untuk memastikan format dan isi sesuai.
 - Repost postingan otomatis di berbagai waktu tertentu.
-

3.4. Workflow Approval (Opsional)

Endpoint:

- POST /posts/:postId/approve – Approve postingan.
- POST /posts/:postId/reject – Reject postingan dengan feedback.

Fungsi:

- Postingan memiliki status (draft, pending, approved, rejected).
 - Admin atau Editor dapat melakukan approval/rejection.
 - Feedback disimpan untuk referensi tim.
-

3.5. Notifikasi & Monitoring

Endpoint:

- POST /notifications – Mengirim notifikasi ke user.
- GET /activity – Mendapatkan log aktivitas tim.
- POST /notifications/email – Kirim notifikasi email ke user.

Fungsi:

- Menyimpan log aktivitas (posting berhasil/gagal, approval, dll.).
- Notifikasi dikirim melalui email atau push notification.
- Webhook dapat digunakan untuk integrasi dengan Slack/Discord.

4. Struktur Database (PostgreSQL + Prisma)

4.1. Users Table

Field	Type	Description
id	UUID	Primary Key
email	String	Unique Email
password	String	Hashed Password
role	Enum	Admin, Editor, Viewer
createdAt	Timestamp	User registration date

4.2. Teams Table

Field	Type	Description
id	UUID	Primary Key
name	String	Team Name
ownerId	UUID	User ID of the team owner
createdAt	Timestamp	Date created

4.3. Posts Table

Field	Type	Description
id	UUID	Primary Key
content	Text	Post content
mediaUrl	String	URL of attached media
scheduledAt	Timestamp	Scheduled post time
status	Enum	Draft, Pending, Approved, Rejected, Posted
teamId	UUID	Team ID
createdAt	Timestamp	Date created

5. Alur Kerja Sistem

1. **User Registrasi/Login:** User daftar atau login melalui email/password atau OAuth.

2. **Manajemen Tim:** Admin membuat tim, mengundang anggota, dan menghubungkan akun sosial media.
3. **Pembuatan Postingan:** User membuat draft postingan, menambahkan media, dan menjadwalkannya.
4. **Approval Workflow:** Jika diperlukan, postingan akan masuk ke status pending dan harus disetujui Admin.
5. **Penjadwalan & Posting:** Postingan yang dijadwalkan akan dipublikasikan secara otomatis oleh sistem.
6. **Notifikasi:** User mendapatkan notifikasi jika postingan berhasil/gagal atau jika ada perubahan status.
7. **Monitoring & Log Aktivitas:** Admin dapat melihat aktivitas dan status postingan tim.

6. Kesimpulan

Rancangan ini menyediakan fitur lengkap untuk mengelola postingan sosial media dalam tim dengan sistem workflow approval, notifikasi, dan job scheduler. Backend dirancang dengan PostgreSQL + Prisma, menggunakan Next.js API Routes, dan terintegrasi dengan OAuth untuk autentikasi sosial media.

Next Step: Implementasi berdasarkan rancangan ini!