

# Dokumentasi Desain Sistem

---

## Sistem Manajemen Tugas Kelompok

Dokumen ini menjelaskan desain **alur sistem (flowchart)** dan **struktur database** yang digunakan.

Tujuannya agar tim developer memahami bagaimana interaksi user di aplikasi berhubungan dengan data yang disimpan.

---

### 1. AUTH (Login & Registrasi)

#### Alur Sistem

1. User baru → registrasi akun.
2. Sistem menyimpan data user ke database.
3. Setelah registrasi, user langsung diarahkan ke dashboard.
4. Jika sudah punya akun → login.
5. Sistem memvalidasi email & password:
  - Jika valid → masuk dashboard.
  - Jika tidak valid → ulangi login.

#### Database

- **users**
  - Menyimpan semua data user dalam sistem.
  - Kolom:
    - **id** (PK) → identitas unik user
    - **name** → nama user
    - **email** → unik, untuk login
    - **password** → disimpan dengan hash
    - **role** → apakah user berperan sebagai **ketua** atau **anggota**
    - **created\_at, updated\_at** → log waktu

👉 Semua aktivitas login & registrasi merujuk ke tabel ini.

---

### 2. GROUP (Manajemen Kelompok)

#### Alur Sistem

1. User dengan role **ketua** membuat kelompok baru.
2. Ketua dapat **mengundang anggota** untuk bergabung.
3. Anggota menerima undangan → status berubah dari *pending* ke *active*.
4. Anggota juga bisa bergabung dengan **kode undangan**.

#### Database

- **groups**

- Menyimpan data kelompok.
- Kolom:
  - `id` (PK)
  - `name, description`
  - `created_by` (FK → `users.id`, ketua)
  - `created_at, updated_at`

- **group\_members**

- Relasi antar `users` dengan `groups`.
- Kolom:
  - `id` (PK)
  - `group_id` (FK → `groups.id`)
  - `user_id` (FK → `users.id`)
  - `role` (ketua / anggota, di level kelompok)
  - `status` (pending / active)
  - `joined_at`

👉 Dengan desain ini:

- Satu user bisa gabung ke banyak kelompok.
- Satu kelompok bisa memiliki banyak anggota.

---

### 3. PROJECT (Manajemen Proyek)

#### Alur Sistem

1. Di dalam kelompok, user (ketua / anggota) dapat membuat proyek.
2. Proyek disimpan dan terhubung ke kelompok tertentu.
3. Satu kelompok dapat memiliki banyak proyek.

#### Database

- **projects**
  - Menyimpan data proyek.
  - Kolom:
    - `id` (PK)
    - `group_id` (FK → `groups.id`) → proyek milik kelompok tertentu
    - `name, description`
    - `created_by` (FK → `users.id`) → siapa pembuat proyek
    - `created_at, updated_at`

---

### 4. TASK (Manajemen Tugas)

#### Alur Sistem

1. Ketua membuat tugas baru di dalam proyek.

2. Tugas di-*assign* ke anggota tertentu.
3. Anggota membaca detail tugas lalu mengubah status (to-do → in-progress → done).
4. Setelah selesai, anggota bisa **upload hasil** + catatan pengerjaan.
5. Ketua melakukan review hasil:
  - Jika disetujui → progress proyek diperbarui.
  - Jika ditolak → tugas kembali ke status sebelumnya.

## Database

- **tasks**

- Menyimpan detail tugas.
- Kolom:
  - **id** (PK)
  - **project\_id** (FK → projects.id)
  - **title, description**
  - **status** (to-do / in-progress / done)
  - **assigned\_to** (FK → users.id)
  - **created\_at, updated\_at**

- **task\_submissions**

- Menyimpan hasil pengumpulan tugas.
- Kolom:
  - **id** (PK)
  - **task\_id** (FK → tasks.id)
  - **submitted\_by** (FK → users.id)
  - **file\_url** (opsional, lampiran hasil kerja)
  - **note** (catatan pengerjaan)
  - **review\_status** (pending / approved / rejected)
  - **submitted\_at**

---

## 5. COLLAB (Kolaborasi & Diskusi)

### Alur Sistem

1. Anggota bisa memberikan komentar di setiap tugas.
2. Komentar memicu notifikasi untuk anggota lain.
3. Semua komentar tersimpan sebagai riwayat diskusi.
4. Anggota juga bisa upload file tambahan → tersimpan di sistem.
5. File bisa dipreview atau diunduh oleh anggota lain.

## Database

- **comments**

- Menyimpan diskusi terkait tugas.
- Kolom:

- `id` (PK)
- `task_id` (FK → tasks.id)
- `user_id` (FK → users.id)
- `content`
- `created_at`

- **attachments**

- Menyimpan file tambahan.
- Kolom:
  - `id` (PK)
  - `task_id` (FK → tasks.id)
  - `user_id` (FK → users.id)
  - `file_url`
  - `created_at`

👉 Semua komentar & file selalu terkait ke tugas tertentu.

---

## 6. REPORT (Monitoring & Laporan)

### Alur Sistem

1. Ketua membuka menu laporan proyek.
2. Sistem menghitung progress berdasarkan status semua tugas.
3. Progress ditampilkan dalam bentuk persentase (%) + grafik kontribusi anggota.
4. Ketua bisa menyimpan laporan untuk dokumentasi.

### Database

- **reports**

- Menyimpan data laporan proyek.
- Kolom:
  - `id` (PK)
  - `project_id` (FK → projects.id)
  - `progress` (persentase %)
  - `details` (catatan tambahan)
  - `generated_at`

---

## Ringkasan Alur Keseluruhan

1. User daftar → login.
2. Ketua membuat kelompok → undang anggota.
3. Di dalam kelompok → buat proyek.
4. Di dalam proyek → buat tugas.
5. Tugas dikerjakan anggota → upload hasil → direview ketua.
6. Progress proyek otomatis diperbarui.
7. Laporan proyek bisa dibuka ketua.

8. Komentor & lampiran tambahan mendukung kolaborasi tim.

---