

## LAMPIRAN

### Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

#### SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Adam Ezra Haqqani  
NIM : 21090126  
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

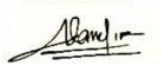
Nama : M. Nishom, S.Kom., M.Kom.  
Status : Dosen  
NIDN : 0619048701  
Jabatan Fungsional : Lektor  
Pangkat/Golongan : Penata /III C

Pada hari ini Jumat tanggal 2 Mei 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat melakukan bimbingan 1 kali dalam seminggu atau setidaknya 3 kali bimbingan dalam satu bulan (dengan progress), apabila saya tidak memenuhi syarat tersebut maka saya tidak berhak meminta surat rekomendasi mengikuti sidang skripsi. Dan saya juga berjanji akan memenuhi persyaratan tersebut dan menyelesaikan skripsi baik produk maupun laporan tepat waktu.

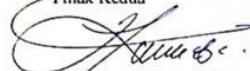
Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 2 Mei 2025

Pihak Pertama

  
Adam Ezra Haqqani

Pihak Kedua

  
M. Nishom, S.Kom., M.Kom.  
NIPY: 09.017.337

Mengetahui  
Ketua Program Studi Sarjana Terapan Teknik  
Informatika

  
Dyah Anggrani, S.T., M.Kom.  
NIPY: 09.015.225

### SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Adam Ezra Haqqani  
NIM : 21090126  
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Taufiq Abidin, S.Pd., M.Kom.  
Status : Dosen  
NIDN : 0603088305  
Jabatan Fungsional : Lektor  
Pangkat/Golongan : Penata /III C

Pada hari ini Rabu tanggal 30 April 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan hari Rabu pukul 15.00 di ruang dosen.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 30 April Maret 2025

Pihak Pertama

  
Adam Ezra Haqqani

Pihak Kedua

  
Taufiq Abidin, S.Pd., M.Kom.  
NIPY: 06.014.184

Mengetahui  
Ketua Program Studi Sarjana Terapan Teknik  
Informatika

  
Dyah Apriliani, S.T., M.Kom.  
NIPY: 09.015.225

## Lampiran 2. Surat Pernyataan Pengajuan HKI

### SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. N a m a : Adam Ezra Haqqani  
Kewarganegaraan : Indonesia  
Alamat : Jl. Pintu Air Gang Masjid Jami Al-Falah, RT. 04RW. 02,  
Kecamatan Tarub, Kabupaten Tegal, 52184
2. N a m a : M. Nishom, S.Kom., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Jl. Jepara, Perum Griya Putri Land Blok A6, RT. 03  
RW. 04, Kecamatan Margadana, Kota Tegal, 52143
3. N a m a : Taufiq Abidin, S.Pd., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Jl. Kaligangsa Asri Timur VIII No.33 RT. 3 RW. 7  
Perum Kaligangsa Asri Kelurahan Kaligangsa  
Kecamatan Margadana, Kota Tegal 52147

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:  
Berupa : Pemograman Komputer  
Berjudul : Platform Pengelolaan Kegiatan dan Keuangan Karang Taruna Berbasis Website  
Menggunakan Arsitektur Multi-Tenancy
  - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
  - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
  - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
  - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
  - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
  - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.

3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.
4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
  - a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
  - b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
  - c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam ber perkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 23 Juni 2025



(Adam Ezra Haqqani)

(M. Nishom, S.Kom., M.Kom.)

(Taufiq Abidin, S.Pd., M.Kom.)



### Lampiran 3. Surat Pengalihan HKI

#### SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini:

1. Nama : Adam Ezra Haqqani  
Kewarganegaraan : Indonesia  
Alamat : Jl. Pintu Air Gang Masjid Jami Al-Falah, RT. 08 RW. 02,  
Kecamatan Tarub, Kabupaten Tegal, 52184
2. Nama : M. Nishom, S.Kom., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Jl. Jepara, Perum Griya Putri Land Blok A6, RT. 03 RW. 04,  
Kecamatan Margadana, Kota Tegal, 52143
3. Nama : Taufiq Abidin, S.Pd., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Jl. Kaligangsa Asri Timur VIII No.33 RT. 3 RW. 7 Perum  
Kaligangsa Asri Kelurahan Kaligangsa Kecamatan Margadana, Kota Tegal  
52147

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada:

Nama : Pusat Penelitian dan Pengabdian Masyarakat  
(P3M) Politeknik Harapan Bersama  
Alamat : Jl. Mataram No. 9 Pasuruan Lor Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "**Platform Pengelolaan Kegiatan dan Keuangan Karang Taruna Berbasis Website Menggunakan Arsitektur Multi-Tenancy**". untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Tegal, 23 Juni 2025

Pemegang Hak Cipta

Kepala P3M

(Muhammad Fikri Midayattullah, S.T., M.Kom.)

Pencipta

( Adam Ezra Haqqani )

(M. Nishom, S.Kom., M.Kom.)

(Taufiq Abidin, S.Pd., M.Kom.)



PLATFORM PENGELOLAAN  
KEGIATAN DAN KEUANGAN  
KARANG TARUNA BERBASIS  
WEBSITE MENGGUNAKAN  
ARSITEKTUR MULTI-  
TENANCY

MANUAL BOOK

Disusun Oleh :

**Adam Ezra Haqqani**

**M. Nishom, S. Kom., M. Kom.**

**Taufiq Abidin, S. Pd., M. Kom.**

## **PENDAHULUAN**

### **1. Tentang Platform**

Platform Pengelolaan Kegiatan dan Keuangan Karang Taruna adalah sistem berbasis website yang dirancang khusus untuk membantu organisasi Karang Taruna dalam mengelola administrasi kegiatan dan keuangan secara digital. Platform ini menggunakan arsitektur multi-tenant yang memungkinkan multiple organisasi Karang Taruna dapat menggunakan sistem yang sama dengan tetap menjaga keamanan dan isolasi data antar organisasi. Sistem ini menyediakan fitur terintegrasi untuk pengelolaan kegiatan, proposal kegiatan, transaksi keuangan, dan pengaturan organisasi dalam satu platform terpadu. Dengan menggunakan platform ini, Karang Taruna dapat meninggalkan sistem pencatatan manual yang rentan terhadap kehilangan data dan kesalahan pencatatan, serta meningkatkan efisiensi dalam proses administrasi organisasi.

### **2. Siapa yang Dapat Menggunakan**

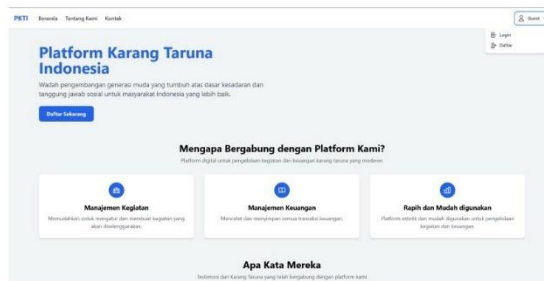
Platform ini diperuntukkan bagi pengurus dan anggota Karang Taruna di seluruh Indonesia yang membutuhkan sistem pengelolaan administrasi yang lebih terstruktur dan efisien. Pengguna utama meliputi ketua Karang Taruna, bendahara, sekretaris, dan pengurus lainnya yang bertanggung jawab dalam pengelolaan kegiatan dan keuangan organisasi. Platform ini juga dapat digunakan oleh anggota Karang Taruna yang memiliki akses untuk melihat informasi kegiatan dan laporan keuangan sesuai dengan tingkat otorisasi yang diberikan. Sistem ini dirancang dengan antarmuka yang user-friendly sehingga dapat digunakan oleh pengguna dengan berbagai tingkat kemampuan teknologi.

### **3. Persyaratan Sistem**

Untuk menggunakan platform ini, pengguna memerlukan perangkat dengan spesifikasi minimum berupa komputer, laptop, tablet, atau smartphone yang memiliki akses internet stabil dengan kecepatan minimal 1 Mbps untuk pengalaman penggunaan yang optimal. Platform mendukung berbagai web browser modern seperti Google Chrome versi 80 ke atas, Mozilla Firefox versi 75 ke atas, Safari versi 13 ke atas, atau Microsoft Edge versi 80 ke atas. Resolusi layar yang direkomendasikan adalah minimal 1024x768 pixel untuk tampilan desktop dan responsif untuk perangkat mobile.

#### 4. Cara Mengakses Platform

Platform dapat diakses melalui web browser dengan mengetikkan alamat URL yang telah disediakan pada address bar browser. Pengguna akan diarahkan ke halaman utama yang menampilkan informasi umum tentang platform dan menu navigasi untuk registrasi atau login. Untuk pengguna baru, tersedia tombol "Daftar" untuk melakukan registrasi akun, sedangkan pengguna yang sudah memiliki akun dapat langsung menggunakan tombol "Login" untuk login ke sistem.



Gambar 1.1 Halaman Utama

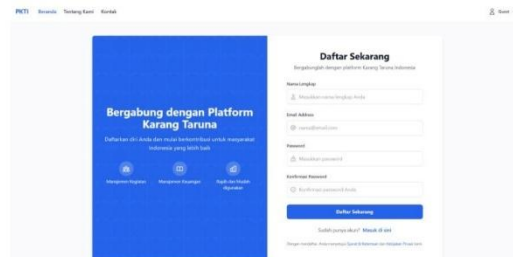


## PENGUNAAN

### 1. Registrasi dan Login

#### 1.1 Registrasi User Baru

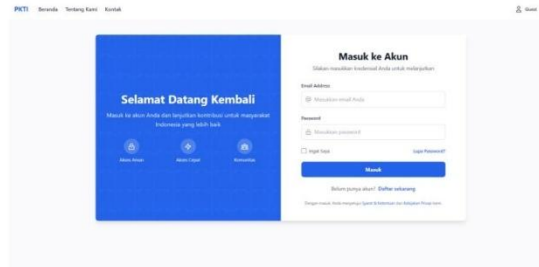
Proses registrasi user dimulai dengan mengakses halaman utama platform dan mengklik tombol "Daftar" yang tersedia di bagian atas halaman. Pengguna akan diarahkan ke form registrasi yang meminta informasi dasar seperti nama lengkap, alamat email yang valid, dan password yang aman. Password harus memenuhi kriteria keamanan dengan minimal 8 karakter yang terdiri dari kombinasi huruf besar, huruf kecil, dan angka. Setelah mengisi semua field yang diperlukan, pengguna perlu menyetujui syarat dan ketentuan penggunaan platform dengan mencentang checkbox yang tersedia. Proses registrasi akan diselesaikan dengan mengklik tombol "Daftar" dan sistem akan mengirimkan email verifikasi ke alamat email yang telah didaftarkan.



Gambar 2.1 Halaman Register User

#### 1.2 Login ke Platform

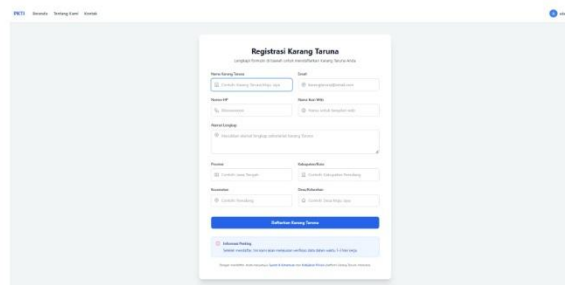
Proses login dapat dilakukan melalui halaman utama dengan mengklik tombol "Login" atau dapat diakses langsung melalui URL login. Pada halaman login, pengguna diminta memasukkan alamat email dan password yang telah didaftarkan sebelumnya. Jika pengguna lupa password, tersedia link "Lupa Password" yang akan mengarahkan ke halaman reset password melalui verifikasi email. Setelah berhasil login, pengguna akan diarahkan ke dashboard utama sesuai dengan role dan akses yang dimiliki.



Gambar 2.2 Halaman Login User

### 1.3 Registrasi untuk Karang Taruna

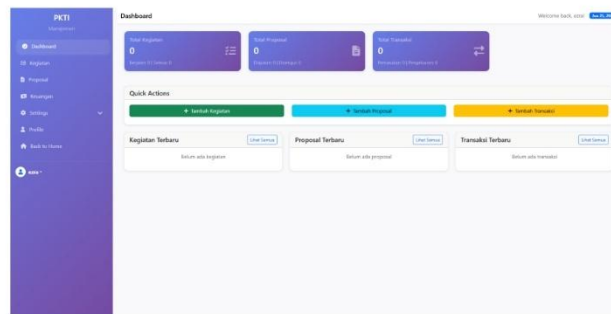
Setelah berhasil login sebagai user, pengguna perlu mendaftarkan organisasi Karang Taruna melalui menu "Daftar Sekarang" yang tersedia di dashboard atau navigasi utama. Form registrasi organisasi meminta informasi lengkap tentang Karang Taruna meliputi nama organisasi, alamat lengkap, nomor telepon, email organisasi, nama ketua, dan informasi legalitas seperti SK pendirian. Pengguna juga perlu mengunggah dokumen pendukung seperti surat keterangan dari kelurahan atau dokumen legalitas organisasi dalam format PDF dengan ukuran maksimal 2MB. Setiap Karang Taruna akan mendapatkan subdomain unik yang akan digunakan sebagai identifier dalam arsitektur multi-tenant platform. Proses verifikasi akan dilakukan oleh admin platform sebelum organisasi dapat mengakses fitur lengkap sistem.



Gambar 2.3 Halaman Registrasi Karang Taruna

## 2. Dashboard

Dashboard utama menyajikan ringkasan informasi penting tentang aktivitas Karang Taruna dalam bentuk cards dan grafik yang mudah dipahami. Bagian atas dashboard menampilkan statistik utama seperti total kegiatan, kegiatan aktif, kegiatan selesai, total anggaran, dan realisasi keuangan dalam periode tertentu. Panel tengah menampilkan grafik aktivitas kegiatan bulanan dan tren keuangan yang membantu pengurus dalam monitoring performa organisasi. Bagian bawah dashboard menyajikan daftar kegiatan terbaru, transaksi keuangan terakhir, dan notifikasi penting yang membutuhkan perhatian. Semua informasi di dashboard dapat di-refresh secara otomatis dan dapat di-filter berdasarkan periode waktu tertentu.

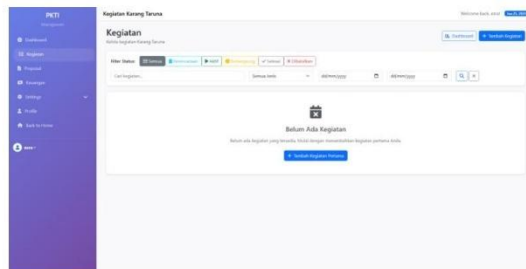


Gambar 2.4 Halaman Dashboard

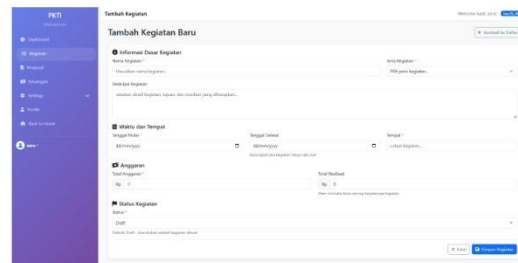
## 3. Kegiatan

Pengelolaan kegiatan dapat dilakukan melalui menu "Kegiatan" yang menyediakan fungsi menambah kegiatan, mengubah kegiatan yang sudah ada, menghapus kegiatan secara lengkap untuk memudahkan administrasi kegiatan organisasi. Untuk membuat kegiatan baru, user mengklik tombol "Tambah Kegiatan" dan mengisi form yang mencakup nama kegiatan, deskripsi, jenis kegiatan, tanggal mulai dan selesai, tempat, status, serta anggaran kegiatan dengan validasi otomatis untuk memastikan konsistensi data. Mengubah kegiatan dilakukan melalui tombol "Edit" pada daftar kegiatan yang akan membuka form edit dengan data yang sudah terisi sebelumnya, termasuk opsi untuk mengupdate status kegiatan dari draft hingga selesai melalui dropdown status yang tersedia. Penghapusan kegiatan dapat dilakukan melalui tombol "Hapus" dengan konfirmasi keamanan, namun kegiatan yang sudah berstatus

aktif atau memiliki transaksi terkait tidak dapat dihapus untuk menjaga integritas data. Halaman daftar kegiatan dilengkapi dengan fitur filter berdasarkan status, jenis, dan periode tanggal, serta search function untuk memudahkan pencarian kegiatan spesifik dalam database organisasi.



Gambar 2.5 Halaman Kegiatan

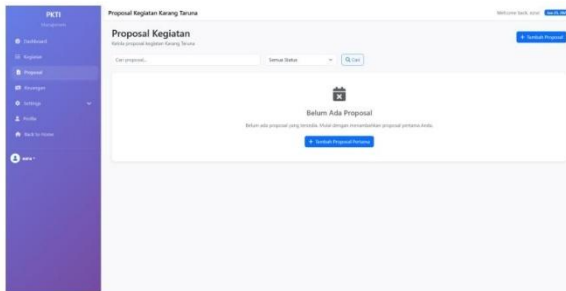


Gambar 2.6 Halaman Menambah Kegiatan

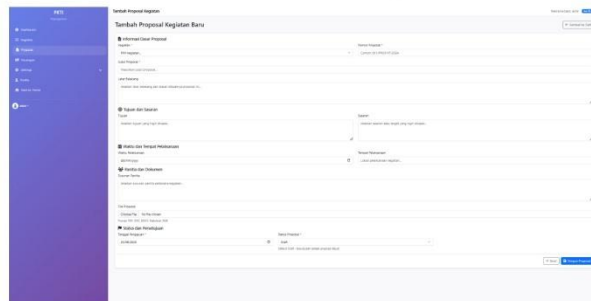
#### 4. Proposal

Manajemen proposal kegiatan terintegrasi dengan sistem kegiatan melalui menu "Proposal" yang memungkinkan pengguna melakukan operasi menambahkan proposal, mengubah proposal, dan menghapus proposal secara komprehensif. Pembuatan proposal dimulai dengan mengklik "Buat Proposal" dan memilih kegiatan terkait dari dropdown, kemudian mengisi informasi lengkap seperti nomor proposal (auto-generate), judul, latar belakang, tujuan, sasaran, waktu pelaksanaan, susunan panitia, dan upload file proposal dalam format PDF/DOC/DOCX maksimal 2MB. Editing proposal dapat dilakukan dengan mengakses tombol "Edit" pada daftar proposal, dimana user dapat mengubah semua informasi termasuk mengganti file proposal yang akan otomatis menggantikan file sebelumnya dalam storage system. Penghapusan

proposal dilakukan melalui tombol "Hapus" dengan konfirmasi yang akan menghapus data proposal beserta file terkait dari storage untuk menjaga kebersihan sistem. Interface daftar proposal menampilkan status dengan color coding (draft, diajukan, disetujui, ditolak, revisi), fitur filter berdasarkan status, search function, dan tombol download file proposal untuk setiap entry yang memiliki attachment.



Gambar 2.7 Halaman Proposal



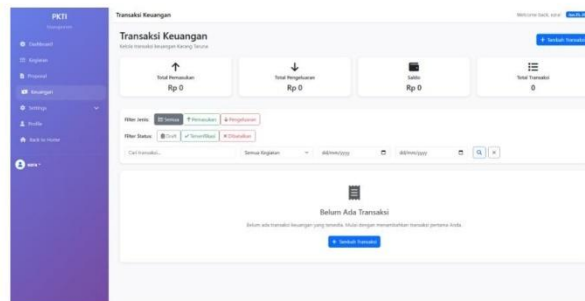
Gambar 2.8 Halaman Menambah Proposal

## 5. Keuangan

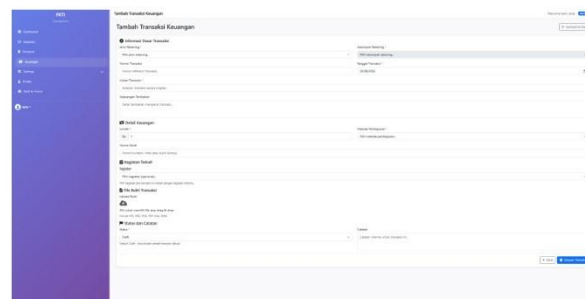
Pengelolaan transaksi keuangan melalui menu "Transaksi" memberikan kontrol penuh terhadap arus kas organisasi dengan fitur CRUD yang terintegrasi dengan sistem rekening. Penambahan transaksi dilakukan dengan mengisi form yang mencakup pilihan kegiatan terkait (opsional), jenis rekening, kelompok rekening yang terfilter otomatis, tanggal transaksi, uraian, jumlah, jenis transaksi (pemasukan/pengeluaran), metode pembayaran, nomor bukti, dan upload bukti transaksi berupa gambar atau PDF. Edit transaksi dapat dilakukan melalui tombol "Edit" dengan validasi khusus dimana transaksi yang sudah diverifikasi memiliki batasan editing untuk menjaga audit trail, termasuk opsi untuk mengubah file bukti transaksi dengan penggantian otomatis file lama. Penghapusan transaksi hanya dapat dilakukan pada transaksi dengan status draft



atau cancelled, dengan konfirmasi keamanan yang akan menghapus data transaksi beserta file bukti dari storage system. Halaman transaksi dilengkapi dengan summary keuangan real-time, filter berdasarkan jenis transaksi, status, periode, kegiatan, dan fitur verifikasi transaksi yang mengubah status dari draft menjadi verified dengan logging user dan timestamp untuk accountability.



Gambar 2.9 Halaman Keuangan

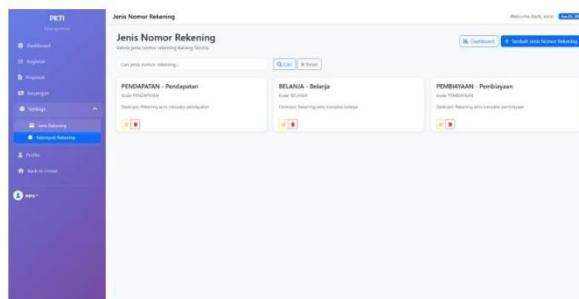


Gambar 2.10 Halaman Menambah Transaksi Keuangan

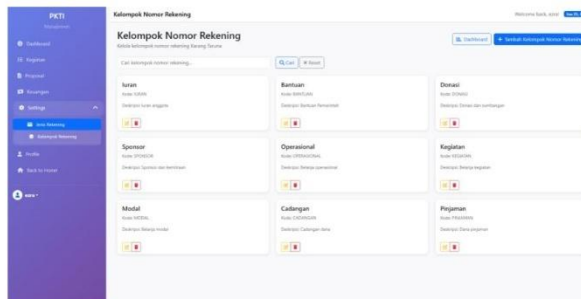
## 6. Setting

Menu Setting menyediakan konfigurasi sistem yang mencakup manajemen jenis rekening dan kelompok rekening sebagai fondasi chart of accounts organisasi. Pada submenu "Jenis Rekening", user dapat membuat jenis rekening baru dengan mengisi kode, nama, deskripsi, dan jenis transaksi (masuk/keluar), mengedit informasi jenis rekening yang sudah ada melalui form edit yang telah terisi data sebelumnya, dan menghapus jenis rekening yang tidak lagi dibutuhkan dengan validasi sistem yang mencegah penghapusan jenis rekening yang masih memiliki kelompok rekening atau transaksi terkait. Manajemen "Kelompok Rekening" dilakukan dengan membuat kelompok baru yang harus dipilih jenis rekeningnya terlebih dahulu, mengisi kode unik

per jenis rekening, nama, deskripsi, dan status "lainnya" untuk kelompok yang fleksibel, mengedit kelompok rekening melalui form yang menampilkan relasi dengan jenis rekening, dan menghapus kelompok rekening dengan validasi untuk memastikan tidak ada transaksi yang masih menggunakan kelompok tersebut. Sistem menampilkan relasi hierarkis antara jenis dan kelompok rekening dalam interface yang mudah dipahami, dengan validasi otomatis untuk mencegah duplikasi kode dan menjaga konsistensi referensi data dalam sistem transaksi.



Gambar 2.11 Halaman Jenis Norekening



Gambar 2.12 Halaman Kelompok Norekening



# PLATFORM PENGELOLAAN KEGIATAN DAN KEUANGAN KARANG TARUNA BERBASIS WEBSITE MENGGUNAKAN ARSITEKTUR MULTI- TENANCY

## DOKUMEN TEKNIS

Disusun Oleh :

**Adam Ezra Haqqani**

**M. Nishom, S. Kom., M. Kom.**

**Taufiq Abidin, S. Pd., M. Kom.**

## OVERVIEW SISTEM

Platform Pengelolaan kegiatan dan keuangan Karang Taruna merupakan aplikasi web berbasis arsitektur multi-tenant yang dikembangkan menggunakan framework Laravel 10 dengan database MySQL. Sistem ini memungkinkan multiple organisasi Karang Taruna menggunakan satu platform dengan isolasi data yang aman antar tenant. Platform menyediakan fitur CRUD terintegrasi untuk pengelolaan kegiatan, proposal, transaksi, jenis rekening, dan kelompok rekening, serta dilengkapi dengan panel admin untuk monitoring dan maintenance sistem secara menyeluruh.

## SOURCE CODE

1. Database Schema dan Migration
  - a. Central Database Migration
    - 1) Tenant

```
Schema::create('tenants', function (Blueprint $table) {
    $table->string('id')->primary();

    // your custom columns may go here

    $table->timestamps();
    $table->json('data')->nullable();
});
```

Migration untuk tabel tenants menggunakan struktur standar dari package Stancu/Tenancy yang berfungsi sebagai registry untuk setiap organisasi Karang Taruna yang terdaftar dalam platform. Kolom “id” bertipe string digunakan sebagai primary key yang akan menjadi identifier unik untuk setiap tenant, sedangkan kolom “data” bertipe JSON menyimpan informasi tambahan seperti nama organisasi, alamat, dan konfigurasi khusus tenant. Timestamps digunakan untuk tracking kapan tenant dibuat dan terakhir diupdate, yang penting untuk audit dan maintenance sistem multi-tenant.

- 2) Users

```
Schema::create('users', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('email')->unique();
    $table->timestamp('email_verified_at')->nullable();
    $table->string('password');
    $table->rememberToken();
    $table->timestamps();
});
```

Migration untuk tabel users dalam central database menyimpan informasi pengguna yang dapat mengakses multiple tenant. Struktur tabel mengikuti standar Laravel authentication dengan kolom name, email, password, dan remember\_token untuk session management. Email dibuat unique untuk memastikan tidak ada duplikasi pengguna di level central, sedangkan 'email\_verified\_at' digunakan untuk verifikasi email pengguna sebelum dapat mengakses sistem.

## b. Tenant Database Migration

### 1) Kegiatan

```
Schema::create('tenant_kegiatan', function (Blueprint $table) {
    $table->id();
    $table->string('nama_kegiatan', 200);
    $table->text('deskripsi')->nullable();
    $table->enum('jenis_kegiatan', ['event', 'pembelajaran',
    'baksos', 'olahraga', 'lainnya']);
    $table->date('tanggal_mulai');
    $table->date('tanggal_selesai')->nullable();
    $table->string('tempat', 200)->nullable();
    $table->enum('status', ['draft', 'diajukan', 'disetujui',
    'ditolak', 'berjalan', 'selesai', 'dibatalkan'])->default('draft');
    $table->decimal('total_anggaran', 15, 2)->default(0);
    $table->decimal('total_realisasi', 15, 2)->default(0);
    $table->timestamps();

    // Indexes
    $table->index('status');
    $table->index(['tanggal_mulai', 'tanggal_selesai']);
    $table->index('jenis_kegiatan');
});
```

Tabel tenant\_kegiatan menyimpan informasi lengkap tentang kegiatan yang diselenggarakan oleh Karang Taruna. Kolom 'nama\_kegiatan' dan 'deskripsi' menyimpan informasi dasar, sedangkan 'jenis\_kegiatan' menggunakan enum untuk kategorisasi. Kolom 'tanggal\_mulai' dan 'tanggal\_selesai' mendefinisikan periode pelaksanaan, dengan 'status' untuk tracking progress kegiatan. Total anggaran dan realisasi menggunakan decimal untuk akurasi keuangan, dan indexes ditambahkan untuk meningkatkan performa query

```

Route::prefix('dashboard/kegiatan')->name('dashboard.kegiatan')->group(function () {
    // Dashboard kegiatan
    Route::get('/dashboard', [TenantKegiatanController::class, 'dashboard'])->name('dashboard');

    // API endpoint untuk options
    Route::get('/api/options', [TenantKegiatanController::class, 'options'])->name('options');

    // Specific Filter Routes
    Route::get('/filter/aktif', [TenantKegiatanController::class, 'aktif'])->name('aktif');
    Route::get('/filter/berlangsung', [TenantKegiatanController::class, 'berlangsung'])->name('berlangsung');

    // Filter Routes berdasarkan Status
    Route::get('/status/status', [TenantKegiatanController::class, 'filterByStatus'])->name('filter-status');

    // Filter Routes berdasarkan Jenis
    Route::get('/jenis/jenis', [TenantKegiatanController::class, 'filterByJenis'])->name('filter-jenis');

    // CRUD Routes
    Route::get('/', [TenantKegiatanController::class, 'index'])->name('index');
    Route::get('/create', [TenantKegiatanController::class, 'create'])->name('create');
    Route::post('/', [TenantKegiatanController::class, 'store'])->name('store');
    Route::get('/{id}', [TenantKegiatanController::class, 'show'])->name('show');
    Route::get('/{id}/edit', [TenantKegiatanController::class, 'edit'])->name('edit');
    Route::put('/{id}', [TenantKegiatanController::class, 'update'])->name('update');
    Route::delete('/{id}', [TenantKegiatanController::class, 'destroy'])->name('destroy');

    // Action routes khusus
    Route::patch('/{id}/status', [TenantKegiatanController::class, 'updateStatus'])->name('update-status');
});

```

Route group untuk kegiatan menggunakan prefix dan name untuk konsistensi penamaan. Struktur dimulai dengan dashboard dan API endpoints, diikuti filter routes, kemudian CRUD routes standar, dan diakhiri dengan action routes khusus seperti update status

#### b. Route Proposal

```

Route::prefix('dashboard/proposal')->name('dashboard.proposal')->group(function () {
    // Index Routes
    Route::get('/', [TenantProposalKegiatanController::class, 'index'])->name('index');
    Route::get('/create', [TenantProposalKegiatanController::class, 'create'])->name('create');
    Route::post('/', [TenantProposalKegiatanController::class, 'store'])->name('store');
    Route::get('/{id}', [TenantProposalKegiatanController::class, 'show'])->name('show');
    Route::get('/{id}/edit', [TenantProposalKegiatanController::class, 'edit'])->name('edit');
    Route::put('/{id}', [TenantProposalKegiatanController::class, 'update'])->name('update');
    Route::delete('/{id}', [TenantProposalKegiatanController::class, 'destroy'])->name('destroy');

    // Download Route
    Route::get('/{id}/download', [TenantProposalKegiatanController::class, 'downloadFile'])->name('download');

    // Filter Routes berdasarkan Status
    Route::get('/status/status', [TenantProposalKegiatanController::class, 'filterByStatus'])->name('filter-status');

    // Specific Status Routes
    Route::get('/filter/pending', [TenantProposalKegiatanController::class, 'pending'])->name('pending');
    Route::get('/filter/approved', [TenantProposalKegiatanController::class, 'approved'])->name('approved');
    Route::get('/filter/rejected', [TenantProposalKegiatanController::class, 'rejected'])->name('rejected');
});

```

Route group untuk proposal memiliki struktur CRUD standar dengan tambahan route khusus untuk download file proposal dan filter berdasarkan status approval yang spesifik untuk workflow proposal

#### c. Route Transaksi



```

public function destroy($id)
{
    $karangtaruna = $this->getKarangTaruna();
    if (!$karangtaruna) {
        return redirect()->route('home')->with('error', 'No karang taruna found for this user.');
    }

    $this->switchToTenantDatabase($karangtaruna);

    try {
        $transaksi = TenantTransaksiKeuangan::find($id);
        if (!$transaksi) {
            return redirect()->route('dashboard.transaksi.index')->with('error', 'Transaksi tidak ditemukan.');
```

Menghapus transaksi berdasarkan ID yang diberikan

Method tambahan:

- 1) `filterByJenis()`: Mengambil transaksi berdasarkan jenis (pemasukan/pengeluaran).
- 2) `updateStatus()`: Memperbarui status transaksi.
- 3) `verify()`: Memverifikasi transaksi.
- 4) `getKelompokRekening()`: Mengambil kelompok rekening berdasarkan jenis rekening.
- 5) `laporan()`: Menghasilkan laporan keuangan berdasarkan periode yang ditentukan

Controller ‘TenantTransaksiKeuanganController’ ini mengelola semua operasi terkait transaksi keuangan, termasuk validasi, pengelolaan file, dan logika bisnis untuk memastikan integritas data. Dengan penekanan pada method ‘switchToTenantDatabase()’, controller ini menunjukkan bagaimana aplikasi beroperasi dalam konteks multi-tenant

#### 4. Routes

##### a. Route Kegiatan

```

public function show($id)
{
    $transcription = $this->specTransang($transa);
    if ($transcription) {
        return redirect($source['name'])->with('error', 'No korang taraka found for this user. ');
    }

    $this->switchto($mysqliDatabase($korangTaraka));

    try {
        $transack = (array)$mysqli->query("SELECT * FROM transack WHERE korangTaraka = '$korangTaraka' AND transackid = '$id'");
    } catch (Exception $e) {
        return redirect($source['name'])->with('error', 'Transack tidak ditemukan. ');
    }

    return view('dashboard.transack.show', compact('transack', 'korangTaraka'));
}

catch (Exception $e) {
    return redirect()->back()->with('error', 'Error retrieving transack: ' . $e->getMessage());
}
}
}

```

Menampilkan detail transaksi berdasarkan ID yang diberikan

[illegible]

Menampilkan form untuk mengedit transaksi yang sudah ada

[illegible]

Memperbarui transaksi yang sudah ada setelah memvalidasi input

Menampilkan daftar transaksi keuangan dengan opsi pencarian dan filter

```
public function create()
{
    $barangtaruna = $this->getKategoriTaruna();
    if (!$barangtaruna) {
        return redirect()->route('home')->with('error', 'No barang taruna found for this user.');
```

...

```
        // Debug: Cek data jenis rekening
        // Foreach ($jenisRekeningList as $jr) {
        //     Log::info('Jenis Rekening: ($jr->model) - ($jr->nama), jenis_transaksi: ($jr->jenis_transaksi)');
        // }

        $jenisRekeningGrouped = TenantTransaksiRekening::select('jenisRekening')
            ->select('id', 'nama', 'kode', 'jenis_rekening_id')
            ->select('jenis_rekening_id')
            ->groupBy('jenis_rekening_id');

        return view('dashboard.transaksi.create', compact(
            'barangtaruna',
            'jenisTransaksiOptions',
            'jenisRekeningOptions',
            'statusOptions',
            'registerList',
            'jenisRekeningList',
            'jenisRekeningGrouped'
        ));
    } catch (Exception $e) {
        return redirect()->back()->with('error', 'Unable to load data: ' . $e->getMessage());
    }
}
```

Menampilkan form untuk menambahkan transaksi baru

```
public function store(Request $request)
{
    $barangtaruna = $this->getKategoriTaruna();
    if (!$barangtaruna) {
        return redirect()->route('home')->with('error', 'No barang taruna found for this user.');
```

...

```
        // Debug: log request data
        // Log::info('Store Request Data: ' . $request->all());

        // Validasi Input
        $validatedData = $request->validate([
            'register_id' => 'nullable|exists:tenant_register_id',
            'jenis_rekening_id' => 'required|exists:tenant_jenis_rekening_id',
            'jenis_transaksi' => 'required|exists:tenant_jenis_transaksi',
            'jumlah' => 'required|numeric|between:0,100', // Minimal 0.01
            'jenis_transaksi' => ['required', 'Rule::in(array_keys(TenantTransaksiRekening::REKORD_PENGALIHAN))'],
            'metode_pembayaran' => ['required', 'Rule::in(array_keys(TenantTransaksiRekening::REKORD_PEMBAYARAN))'],
            'file_bukti' => 'nullable|string|max:255',
            'file_saksi' => 'nullable|file|max:10240',
            'status' => ['nullable', 'Rule::in(array_keys(TenantTransaksiRekening::STATUS))'],
            'catatan' => 'nullable|string'
        ]);

        // Debug: log validated data
        // Log::info('Validated Data: ' . $validatedData);

        if ($validatedData['jenis_transaksi'] === 'pembelian') {
            $validatedData['jenis_transaksi'] = 'masuk';
        } elseif ($validatedData['jenis_transaksi'] === 'pengalihan') {
            $validatedData['jenis_transaksi'] = 'keluar';
        }
    }
}
```

Menyimpan transaksi baru setelah memvalidasi input

Controller `TenantProposalKegiatanController` ini mengelola semua operasi CRUD untuk proposal kegiatan dengan implementasi arsitektur multi-tenant yang konsisten, pengelolaan file upload/download yang aman, dan validasi khusus untuk memastikan integritas data dalam konteks tenant database

### c. Transaksi

```
private function switchToTenantDatabase($karangtaruna)
{
    $tenantDb = 'tenant_' . $karangtaruna->domain;
    config(['database.default' => 'tenant']);
    config(['database.connections.tenant.database' => $tenantDb]);
    DB::purge($tenantDb);
    DB::reconnect($tenantDb);
}

private function getKarangTaruna()
{
    $karangtarunas = Karangtaruna::where('user_id', auth()->id)
    ()->get();
    return $karangtarunas->first() ?? null;
}
```

Method `'switchToTenantDatabase()'` adalah inti dari arsitektur multi-tenant yang memungkinkan aplikasi beralih ke database tenant yang sesuai berdasarkan domain Karang Taruna. Method ini mengubah konfigurasi database default menjadi database tenant spesifik, kemudian membersihkan koneksi lama dengan `'DB::purge()'` dan membuat koneksi baru dengan `'DB::reconnect()'`. Method `'getKarangTaruna()'` mengambil data Karang Taruna yang terkait dengan user yang sedang login untuk menentukan tenant database mana yang akan digunakan

```
public function index(Request $request)
{
    $karangtaruna = $this->getKarangTaruna();
    if ($karangtaruna) {
        return redirect()->route('home')->with('error',
        'No karang taruna found for this user.');
```

```
if ($request->has('halaman_rekening_id') && $request->
    halaman_rekening_id != '') {
    $query = $this->halamanRekening($request->
    halaman_rekening_id);
    if ($request->has('tanggal_mulai') && $request->
    tanggal_mulai != '') {
        $query = $query->where('tanggal_transaksi', '>=', $request->
    tanggal_mulai);
    }
    if ($request->has('tanggal_selesai') && $request->
    tanggal_selesai != '') {
        $query = $query->where('tanggal_transaksi', '<=', $request->
    tanggal_selesai);
    }
    $transaksi = $query->orderBy('tanggal_transaksi',
    'desc')->paginate(10);
    // set filter options
    $jenisTransaksiOptions = TenantTransaksiOptions::
    getJenisTransaksiOptions();
    $statusTransaksiOptions = TenantTransaksiOptions::
    getStatusTransaksiOptions();
    $tanggalMulai = $request->has('tanggal_mulai') ?
    $request->tanggal_mulai : null;
    $tanggalSelesai = $request->has('tanggal_selesai') ?
    $request->tanggal_selesai : null;
    $query = $query->where('tanggal_transaksi', '>=', $tanggalMulai)
    && $query->where('tanggal_transaksi', '<=', $tanggalSelesai);
    $transaksi = $query->paginate(10);
    return view('dashboard.transaksi.index', compact(
    'transaksi',
    'tanggalMulai',
    'tanggalSelesai',
    'jenisTransaksiOptions',
    'statusTransaksiOptions',
    'tanggalMulai',
    'tanggalSelesai',
    'halamanRekeningId',
    'halamanRekeningId'));
} catch (Exception $e) {
    return redirect()->back()->with('error',
    'Unable to connect to tenant database.')->noPage();
}
```

```

public function updateProposal(Request $Rq)
{
    $barangtamu = $this->getbarangtamu();

    if (!$barangtamu) {
        return redirect()->route('home')
            -with('error', 'No barang tamu found for this user. ');
    }

    $this->redirectIfUnauthorized($barangtamu);

    $proposal = TenantProposal::findOrFail($Rq->id);

    if (!$proposal) {
        return redirect()->route('dashboard.proposal.index')
            -with('error', 'Proposal tidak ditemukan. ');
    }

    // Check validation untuk memastikan request di atas di database tenant
    $proposal->update($Rq->all());

    // Validasi (keabsahan, Boleh, Boleh)
    if ($proposal->update($Rq->all()) && $proposal->exists()) {
        // Proposal yang sudah lama tidak ada
    }

    // Update file baru
    $file = $Rq->file;
    if ($file) {
        $file->store($proposal->tenant->storagePath('proposals'), 'public');
        $proposal->update(['file_path' => $file->getStoragePath()]);
    }

    $proposal->update($Rq->all());

    return redirect()->route('dashboard.proposal.index')
        -with('success', 'Proposal berhasil diupdate. ');
    } catch (Exception $e) {
        return redirect()->back()
            -with('error', 'Error updating proposal. ' . $e->getMessage())
            -with('back');
    }
}

```

Method ini memperbarui proposal dengan pengelolaan file yang aman, dimana file lama akan dihapus dari storage sebelum file baru diupload untuk mencegah penumpukan file yang tidak terpakai

```

public function destroy($id)
{
    $barangtamu = $this->getbarangtamu();

    if (!$barangtamu) {
        return redirect()->route('home')
            -with('error', 'No barang tamu found for this user. ');
    }

    $this->redirectIfUnauthorized($barangtamu);

    try {
        $proposal = TenantProposal::findOrFail($id);

        if (!$proposal) {
            return redirect()->route('dashboard.proposal.index')
                -with('error', 'Proposal tidak ditemukan. ');
        }

        // Check file (file ada atau tidak) request
        if ($proposal->file_path && Storage::disk('public')->exists($proposal->file_path)) {
            Storage::disk('public')->delete($proposal->file_path);
        }

        $proposal->delete();

        return redirect()->route('dashboard.proposal.index')
            -with('success', 'Proposal berhasil dihapus. ');
    } catch (Exception $e) {
        return redirect()->back()
            -with('error', 'Error deleting proposal. ' . $e->getMessage());
    }
}

```

Method ini menghapus proposal dengan memastikan file terkait juga dihapus dari storage untuk menjaga kebersihan sistem file dan mencegah file orphan

```

'kegiatan_id' => [
    'required',
    'integer',
    function ($attribute, $value, $fail) {
        if (!TenantKegiatan::where('id', $value)->exists()) {
            $fail('Kegiatan yang dipilih tidak valid. ');
        }
    },
],

```

Validasi khusus ini memastikan bahwa kegiatan\_id yang dipilih benar-benar ada di database tenant yang sedang aktif, karena validasi Laravel standar exists tidak dapat digunakan langsung pada database tenant yang dinamis

## b. Proposal

```
private function switchToTenantDatabase($karangtaruna)
{
    $tenantDb = 'tenant_' . $karangtaruna->domain;
    config(['database.default' => 'tenant']);
    config(['database.connections.tenant.database' => $tenantDb]);
    DB::purge('tenant');
    DB::reconnect('tenant');
}

private function getKarangTaruna()
{
    $karangtarunas = KarangTaruna::where('user_id', auth()->id)
    ()->get();
    return $karangtarunas->first() ?? null;
}
```

Method 'switchToTenantDatabase()' adalah inti dari arsitektur multi-tenant yang memungkinkan aplikasi beralih ke database tenant yang sesuai berdasarkan domain Karang Taruna. Method ini mengubah konfigurasi database default menjadi database tenant spesifik, kemudian membersihkan koneksi lama dengan 'DB::purge()' dan membuat koneksi baru dengan 'DB::reconnect()'. Method 'getKarangTaruna()' mengambil data Karang Taruna yang terkait dengan user yang sedang login untuk menentukan tenant database mana yang akan digunakan

```
public function store(Request $request)
{
    $karangtaruna = $this->getKarangTaruna();

    if (!$karangtaruna) {
        return redirect()->route('home')
            ->with('error', 'No Karang Taruna found for this user.');
```

```
validate($data = Request::all());

// Mendapatkan nama pengguna dari request
if ($request->hasHeader('X-tenant-id')) {
    // Mendapatkan nama pengguna dari request
    $tenantId = $request->header('X-tenant-id');
    $tenantId = $request->header('X-tenant-id');
    $tenantId = $request->header('X-tenant-id');
```

Method ini menyimpan proposal baru dengan validasi khusus untuk memastikan kegiatan\_id yang dipilih benar-benar ada di database tenant. File proposal diupload ke storage public dengan folder 'proposals', dan path file disimpan di database untuk referensi download nantinya



Menampilkan form edit untuk kegiatan yang sudah ada

```
public function update(Request $request, $id)
{
    $tenantname = $this->getTenantName();

    if (!$tenantname) {
        return redirect()->route('home')
            ->with('error', 'No tenant name found for this user.');
```

...

```
        $validationData = $request->validate([
            'nama_kegiatan' => 'required|string|max:255',
            'deskripsi' => 'required|string',
            'tenant_kegiatan' => ['required', Rule::in($array_keys($tenantkegiatan
                :IDSIS_KEGIATAN))],
            'tanggal_mulai' => 'required|date',
            'tanggal_akhir' => 'nullable|date|after_or_equal:tanggal_mulai',
            'target' => 'required|string|max:255',
            'status' => ['required', Rule::in($array_keys($tenantkegiatan:STATUS))],
            'total_anggaran' => 'required|numeric|min:0',
            'total_realisasi' => 'nullable|numeric|min:0|lte:total_anggaran',
        ]);

        $kegiatan->update($validationData);

        return redirect()->route('dashboard.kegiatan.index')
            ->with('success', 'Kegiatan berhasil diupdate!');
```

...

```
    } catch (\Exception $e) {
        return redirect()->back()
            ->with('error', 'Error updating kegiatan: ' . $e->getMessage())
            ->withInput();
    }
}
```

Method ini memperbarui data kegiatan dengan validasi yang sama seperti pada method store, memastikan integritas data tetap terjaga

```
public function destroy($id)
{
    $tenantname = $this->getTenantName();

    if (!$tenantname) {
        return redirect()->route('home')
            ->with('error', 'No tenant name found for this user.');
```

...

```
        $kegiatan = TenantKegiatan::find($id);

        if (!$kegiatan) {
            return redirect()->route('dashboard.kegiatan.index')
                ->with('error', 'Kegiatan tidak ditemukan.');
```

...

```
        // Check if kegiatan can be deleted (not in active status)
        if ($kegiatan->isAktif()) {
            return redirect()->back()
                ->with('error', 'Gagal delete action kegiatan. Please change status first.');
```

...

```
        $kegiatan->delete();

        return redirect()->route('dashboard.kegiatan.index')
            ->with('success', 'Kegiatan berhasil dihapus!');
```

...

```
    } catch (\Exception $e) {
        return redirect()->back()
            ->with('error', 'Error deleting kegiatan: ' . $e->getMessage());
    }
}
```

Method ini menghapus kegiatan dengan business logic yang memastikan kegiatan yang sedang aktif tidak dapat dihapus, menggunakan method isAktif() dari model

Controller TenantKegiatanController ini mengelola semua operasi CRUD untuk kegiatan dengan implementasi arsitektur multi-tenant yang konsisten, validasi data yang komprehensif, dan business logic yang memastikan integritas data kegiatan dalam sistem

```

public function create()
{
    $karangtaruna = $this->getKarangTaruna();

    if (!$karangtaruna) {
        return redirect()->route('home')
            ->with('error', 'No karang taruna found for this user.');
```

Menampilkan form dan membuat kegiatan baru

```

public function store(Request $request)
{
    $karangtaruna = $this->getKarangTaruna();

    if (!$karangtaruna) {
        return redirect()->route('home')
            ->with('error', 'No karang taruna found for this user.');
```

Method ini menyimpan kegiatan baru dengan validasi yang komprehensif, termasuk validasi tanggal selesai yang harus setelah atau sama dengan tanggal mulai, dan total realisasi yang tidak boleh melebihi total anggaran

```

public function edit($id)
{
    $karangtaruna = $this->getKarangTaruna();

    if (!$karangtaruna) {
        return redirect()->route('home')
            ->with('error', 'No karang taruna found for this user.');
```

```
private function switchToTenantDatabase($karangtaruna)
{
    $tenantDb = 'tenant_' . $karangtaruna->domain;
    config(['database.default' => 'tenant']);
    config(['database.connections.tenant.database' => $tenantDb]);
    DB::purge('tenant');
    DB::reconnect('tenant');
}

private function getKarangTaruna()
{
    $karangtarunas = KarangTaruna::where('user_id', auth()->id()->get());
    $karangtaruna = $karangtarunas->first();

    if (!$karangtaruna) {
        return null;
    }

    return $karangtaruna;
}
```

Method 'switchToTenantDatabase()' adalah inti dari arsitektur multi-tenant yang memungkinkan aplikasi beralih ke database tenant yang sesuai berdasarkan domain Karang Taruna. Method ini mengubah konfigurasi database default menjadi database tenant spesifik, kemudian membersihkan koneksi lama dengan 'DB::purge()' dan membuat koneksi baru dengan 'DB::reconnect()'. Method 'getKarangTaruna()' mengambil data Karang Taruna yang terkait dengan user yang sedang login untuk menentukan tenant database mana yang akan digunakan

```
public function index(Request $request)
{
    $karangtaruna = $this->getKarangTaruna();

    if (!$karangtaruna) {
        return redirect()->route('home')
            ->with('error', 'No karang taruna found for this user.');
```

```
// Apply search filter if provided
if ($request->has('search') && $request->search != '') {
    $searchTerm = $request->input('search');
    $query = KarangTaruna::where('user_id', auth()->id()->get())
        ->where('domain', 'like', '%' . $searchTerm . '%')
        ->where('description', 'like', '%' . $searchTerm . '%')
        ->where('status', 'like', '%' . $searchTerm . '%');
```

Method ini menampilkan daftar kegiatan dengan fitur pencarian dan filter berdasarkan status dan jenis kegiatan. Query menggunakan scope yang telah didefinisikan di model untuk memudahkan filtering data

```

public function save(array $options = [])
{
    // Auto-set jenis_transaksi berdasarkan jenis_rekening
    if ($this->jenisRekening && !$this->jenis_transaksi) {
        $this->setJenisTransaksiFromRekening();
    }

    // Auto generate nomor_transaksi jika belum ada
    if (!$this->nomor_transaksi) {
        $this->nomor_transaksi = $this->generateNomorTransaksi();
    }

    return parent::save($options);
}

```

```

protected static function boot()
{
    parent::boot();

    // Event ketika creating (sebelum save pertama kali)
    static::creating(function ($transaksi) {
        Mlog::info('Creating Transaksi Event', $transaksi->to);

        // Jangan auto set jenis_transaksi di sini karena sudah di-set di save()
        // Auto generate nomor_transaksi jika belum ada
        if (!$transaksi->nomor_transaksi) {
            $transaksi->nomor_transaksi = $transaksi->generateNomorTransaksi();
        }
    });

    // Event ketika updating
    static::updating(function ($transaksi) {
        Mlog::info('Updating Transaksi Event', $transaksi->to);

        // Validasi konsistensi jenis_transaksi hanya jika jenis_rekening
        if ($transaksi->isDirty('jenis_rekening_id')) {
            $jenisRekening = TenantJenisRekening::find($transaksi->jenis_rekening_id);
            if ($jenisRekening && $transaksi->jenis_transaksi && $jenisRekening->jenis_transaksi) {
                // Jenis_transaksi tidak konsisten dengan jenis_rekening yang dipilih
            }
        }
    });
}

```

- 1) generateNomorTransaksi: Menghasilkan nomor transaksi secara otomatis berdasarkan jenis transaksi dan waktu pembuatan.
- 2) verify: Memverifikasi transaksi dan memperbarui statusnya.
- 3) validateJenisTransaksi: Memastikan konsistensi jenis transaksi dengan jenis rekening.
- 4) setJenisTransaksiFromRekening: Mengatur jenis transaksi berdasarkan jenis rekening.
- 5) save: Override metode save untuk auto-set jenis transaksi dan nomor transaksi.

- 6) boot: Menangani event saat model dibuat atau diperbarui

Model ‘TenantTransaksiKeuangan’ ini memiliki relasi yang jelas dengan kegiatan dan rekening, serta menyediakan berbagai scopes, static methods, accessor methods, dan business logic untuk memudahkan pengelolaan data transaksi keuangan.

### 3. Controller

#### a. Kegiatan

```

public function getJenisTransaksiTextAttribute()
{
    //ambil dari relationship jenis-rekening
    if ($this->jenisRekening) {
        return $this->jenisRekening->jenis_transaksi ===
        'masuk' ? 'Pemasukan' : 'Pengeluaran';
    }

    // Fallback ke nilai langsung
    return $this->jenis_transaksi === 'masuk' ? 'Pemasukan' :
    'Pengeluaran';
}

public function getMetodePembayaranTextAttribute()
{
    return self::METODE_PEMBAYARAN[$this->metode_pembayaran];
}

public function getStatusTextAttribute()
{
    return self::STATUS[$this->status] ?? $this->status;
}

public function getCodeRekeningLengkapAttribute()
{
    if ($this->jenisRekening && $this->kelompokRekening) {
        return $this->jenisRekening->kode . '/' . $this->
        kelompokRekening->kode;
    }
    return '';
}

public function getNamaRekeningLengkapAttribute()
{
    if ($this->jenisRekening && $this->kelompokRekening) {
        return $this->jenisRekening->nama . ' - ' . $this->
        kelompokRekening->nama;
    }
    return '';
}

```

Accessor methods ini digunakan untuk mendapatkan representasi yang lebih mudah dibaca dari atribut tertentu, seperti jenis transaksi, metode pembayaran, dan nama rekening lengkap

```

public function generateKodeTransaksi()
{
    $prefix = $this->jenis_transaksi == 'masuk' ? 'TM' : 'P';
    $tahun = now()->year;
    $bulan = now()->format('m');
    $counter = self::where('jenis_transaksi', $this->
    jenis_transaksi)
    ->whereYear('created_at', $tahun)
    ->whereMonth('created_at', $bulan)
    ->count() + 1;

    return sprintf('%02s/%02s/%04d', $prefix, $bulan, $tahun,
    $counter);
}

public function verify($verifiedBy)
{
    $this->status = 'verified';
    $this->verified_by = $verifiedBy;
    $this->verified_at = now();
    $this->save();

    if ($this->kegiatan_id) {
        $this->kegiatan->updateTotalRealisasi();
    }
}

public function validateJenisTransaksi()
{
    // Jika belum ada jenisRekening, ambil dari database
    if ($this->jenisRekening && $this->jenis_rekening_id) {
        $jenisRekening = $this->jenisRekening->find($this->
        jenis_rekening_id);
        if (!$jenisRekening) {
            return false;
        }
        $expectedJenis = $jenisRekening->jenis_transaksi;
    } else if ($this->jenisRekening) {
        $expectedJenis = $this->jenisRekening->jenis_transaksi;
    } else {
        return false;
    }

    return $this->jenis_transaksi === $expectedJenis;
}

// Method untuk memverifikasi jenis transaksi berdasarkan jenis rekening
public function setJenisTransaksiFromRekening()
{
    if ($this->jenisRekening) {
        $this->jenis_transaksi = $this->jenisRekening->
        jenis_transaksi;
    }
}

```

```

public function scopeByJenis($query, $jenis)
{
    return $query->where('jenis_transaksi', $jenis);
}

public function scopeMasuk($query)
{
    return $query->where('jenis_transaksi', 'masuk');
}

public function scopeKeluar($query)
{
    return $query->where('jenis_transaksi', 'keluar');
}

public function scopeVerified($query)
{
    return $query->where('status', 'verified');
}

public function scopeByPeriode($query, $startDate, $endDate)
{
    return $query->whereBetween('tanggal_transaksi', [
        $startDate, $endDate]);
}

public function scopeByKegiatan($query, $kegiatanId)
{
    return $query->where('kegiatan_id', $kegiatanId);
}

public function scopeByJenisRekening($query, $jenisRekeningId)
{
    return $query->where('jenis_rekening_id', $jenisRekeningId);
}

public function scopeByKelompokRekening($query, $kelompokRekeningId)
{
    return $query->where('kelompok_rekening_id', $kelompokRekeningId);
}

```

Scopes ini memudahkan query untuk mendapatkan transaksi berdasarkan jenis, status, periode, dan relasi dengan kegiatan atau rekening

```

public static function getJenisRekeningOptions($jenisTransaksi = null)
{
    $query = Tenant::selectRaw('with("kelompok_rekening");');
    if ($jenisTransaksi == 'masuk') {
        $query->where('kode', 'PENDAPATAN');
    } elseif ($jenisTransaksi == 'keluar') {
        $query->whereIn('kode', ['BELANJA', 'PENDIAYAHAN']);
    }
    return $query->get();
}

public static function getKelompokRekeningOptions($jenisRekeningId = null)
{
    $query = Tenant::selectRaw('with("jenis_rekening");');
    if ($jenisRekeningId) {
        $query->where('jenis_rekening_id', $jenisRekeningId);
    }
    return $query->get();
}

public static function getJenisTransaksiOptions()
{
    return [
        'masuk' => 'Pendapatan',
        'keluar' => 'Pengeluaran'
    ];
}

```

Static methods ini digunakan untuk mendapatkan opsi jenis rekening dan kelompok rekening berdasarkan jenis transaksi, serta untuk mendapatkan opsi jenis transaksi

### c. Transaksi

```
use HasFactory;

protected $table = 'tenant_transaksi_keuangan';

protected $fillable = [
    'kegiatan_id',
    'jenis_rekening_id',
    'kelompok_rekening_id',
    'nomor_transaksi',
    'tanggal_transaksi',
    'uraian',
    'keterangan_lainnya',
    'jumlah',
    'jenis_transaksi',
    'metode_pembayaran',
    'nomor_bukti',
    'file_bukti',
    'status',
    'created_by',
    'verified_by',
    'verified_at',
    'catatan'
];
```

Model ini menggunakan trait 'HasFactory' untuk mendukung factory-based testing. Tabel yang digunakan adalah 'tenant\_transaksi\_keuangan', dan kolom yang dapat diisi secara massal didefinisikan dalam properti '\$fillable'

```
public function kegiatan()
{
    return $this->belongsTo(TenantKegiatan::class, 'kegiatan_id');
}

public function jenisRekening()
{
    return $this->belongsTo(TenantJenisRekening::class,
        'jenis_rekening_id');
}

public function kelompokRekening()
{
    return $this->belongsTo(TenantKelompokRekening::class,
        'kelompok_rekening_id');
}
```

- 1) kegiatan: Relasi belongsTo dengan model 'TenantKegiatan', menunjukkan bahwa setiap transaksi terkait dengan satu kegiatan.
- 2) jenisRekening: Relasi belongsTo dengan model 'TenantJenisRekening', menunjukkan jenis rekening dari transaksi.
- 3) kelompokRekening: Relasi belongsTo dengan model 'TenantKelompokRekening', menunjukkan kelompok rekening dari transaksi



```

public function scopeByStatus($query, $status)
{
    return $query->where('status_proposal', $status);
}

public function scopeDisetujui($query)
{
    return $query->where('status_proposal', 'disetujui');
}

```

Scopes ini memudahkan query untuk mendapatkan proposal berdasarkan status. Misalnya, 'scopeDisetujui' mengembalikan proposal yang statusnya 'disetujui'

```

public function getStatusProposalTextAttribute()
{
    return self::STATUS_PROPOSAL[$this->status_proposal] ?? $this->status_proposal;
}

public function getTotalAnggaranAttribute()
{
    return $this-> rincianAnggaranProposal()->sum('total_harga');
}

public function generateNomorProposal()
{
    $tahun = now()->year;
    $bulan = now()->format('m');
    $counter = self::whereYear('created_at', $tahun)
        ->whereMonth('created_at', $bulan)
        ->count() + 1;

    return sprintf('PROP/%s/%s/%03d', $bulan, $tahun, $counter);
}

```

- 1) getTotalAnggaranAttribute: Menghitung total anggaran proposal dengan menjumlahkan semua rincian anggaran yang terkait.
- 2) generateNomorProposal: Menghasilkan nomor proposal secara otomatis berdasarkan tahun dan bulan saat proposal dibuat, dengan format yang telah ditentukan

Model 'TenantProposalKegiatan' ini memiliki relasi yang jelas dengan kegiatan dan rincian anggaran, serta menyediakan berbagai scopes dan methods untuk memudahkan pengelolaan data proposal

- 1) `getSisaAnggaranAttribute`: Menghitung sisa anggaran berdasarkan total anggaran dan total realisasi.
- 2) `updateTotalRealisasi`: Mengupdate total realisasi dengan menjumlahkan semua transaksi yang sudah diverifikasi dan bertipe 'keluar'.

Model `TenantKegiatan` ini memiliki relasi yang jelas dengan proposal dan transaksi, serta menyediakan berbagai scopes dan methods untuk memudahkan pengelolaan data kegiatan.

#### b. Proposal

```
use HasFactory;

protected $table = 'tenant_proposal_kegiatan';

protected $fillable = [
    'kegiatan_id',
    'nomor_proposal',
    'judul_proposal',
    'latar_belakang',
    'tujuan',
    'sasaran',
    'waktu_pelaksanaan',
    'tempat_pelaksanaan',
    'susunan_panitia',
    'file_proposal',
    'tanggal_pengajuan',
    'status_proposal',
    'catatan_revisi',
    'disetujui_oleh',
    'tanggal_persetujuan'
];
```

Model ini menggunakan trait `HasFactory` untuk mendukung factory-based testing. Tabel yang digunakan adalah `'tenant_proposal_kegiatan'`, dan kolom yang dapat diisi secara massal didefinisikan dalam properti `'$fillable'`

```
public function kegiatan()
{
    return $this->belongsTo(TenantKegiatan::class, 'kegiatan_id');
}

public function rincianAnggaranProposal()
{
    return $this->hasMany(TenantRincianAnggaranProposal::class, 'proposal_id');
}
```

- 1) `kegiatan`: Relasi `belongsTo` dengan model `'TenantKegiatan'`, menunjukkan bahwa setiap proposal terkait dengan satu kegiatan.
- 2) `rincianAnggaranProposal`: Relasi `one-to-many` dengan model `'TenantRincianAnggaranProposal'`, menunjukkan bahwa setiap proposal dapat memiliki banyak rincian anggaran

- 2) transaksiKeuangan: Relasi one-to-many dengan model TenantTransaksiKeuangan, menunjukkan bahwa setiap kegiatan dapat memiliki banyak transaksi keuangan

```
public function scopeByStatus($query, $status)
{
    return $query->where('status', $status);
}

public function scopeByJenis($query, $jenis)
{
    return $query->where('jenis_kegiatan', $jenis);
}

public function scopeAktif($query)
{
    return $query->whereIn('status', ['disetujui', 'berjalan']);
}

public function scopeBerlangsung($query)
{
    return $query->where('tanggal_mulai', '<=', now())
        ->where(function($q) {
            $q->whereNull('tanggal_selesai')
                ->orWhere('tanggal_selesai', '>=', now());
        });
}
```

Scopes ini memudahkan query untuk mendapatkan kegiatan berdasarkan status, jenis, atau status aktif. Misalnya, 'scopeAktif' mengembalikan kegiatan yang statusnya 'disetujui' atau 'berjalan'

```
public function getJenisKegiatanTextAttribute()
{
    return self::JENIS_KEGIATAN[$this->jenis_kegiatan] ?? $this->jenis_kegiatan;
}

public function getStatusTextAttribute()
{
    return self::STATUS[$this->status] ?? $this->status;
}

public function getTotalAnggaranAttribute()
{
    return $this->total_anggaran + $this->total_realisasi;
}

public function getPersentaseRealisasiAttribute()
{
    if ($this->total_anggaran > 0) {
        return round(($this->total_realisasi / $this->total_anggaran) * 100, 2);
    }
    return 0;
}

public function getDurasiKegiatanAttribute()
{
    if ($this->tanggal_selesai) {
        return $this->tanggal_mulai->diffInDays($this->tanggal_selesai) + 1;
    }
    return 1;
}

public function isAktif()
{
    return in_array($this->status, ['disetujui', 'berjalan']);
}

public function updateTotalRealisasi()
{
    $this->total_realisasi = $this->transaksiKeuangan()
        ->where('status', 'verifikasi')
        ->where('jenis_transaksi', 'keluar')
        ->sum('jumlah');
    $this->save();
}
```

Tabel 'tenant\_transaksi\_keuangan' menyimpan informasi transaksi keuangan dengan multiple foreign key ke tabel kegiatan, jenis rekening, dan kelompok rekening. Kolom 'nomor\_transaksi' yang unique memastikan tidak ada duplikasi, sedangkan 'jumlah' menggunakan decimal untuk akurasi. Status transaksi memungkinkan workflow approval, dan kategori transaksi memberikan klasifikasi tambahan untuk reporting.

## 2. Model dan Relationship

### a. Kegiatan

```
use HasFactory;

protected $table = 'tenant_kegiatan';

protected $fillable = [
    'nama_kegiatan',
    'deskripsi',
    'jenis_kegiatan',
    'tanggal_mulai',
    'tanggal_selesai',
    'tempat',
    'status',
    'total_anggaran',
    'total_realisasi'
];
```

Model ini menggunakan trait 'HasFactory' untuk mendukung factory-based testing. Tabel yang digunakan adalah 'tenant\_kegiatan', dan kolom yang dapat diisi secara massal didefinisikan dalam properti '\$fillable'.

```
public function proposalKegiatan()
{
    return $this->hasOne(TenantProposalKegiatan::class,
        'kegiatan_id');
}

public function transaksiKeuangan()
{
    return $this->hasMany(TenantTransaksiKeuangan::class,
        'kegiatan_id');
}

public function logStatusKegiatan()
{
    return $this->hasMany(TenantLogStatusKegiatan::class,
        'kegiatan_id');
}
```

- 1) proposalKegiatan: Relasi one-to-one dengan model TenantProposalKegiatan, menunjukkan bahwa setiap kegiatan dapat memiliki satu proposal.

## 2) Proposal

```
Schema::create('tenant_proposal_kegiatan', function (Blueprint
$table) {
    $table->id();
    $table->foreignId('kegiatan_id')->constrained(
        'tenant_kegiatan')->onDelete('cascade');
    $table->string('nomor_proposal', 50)->nullable();
    $table->string('judul_proposal', 250);
    $table->text('latar_belakang')->nullable();
    $table->text('tujuan')->nullable();
    $table->text('sasaran')->nullable();
    $table->date('waktu_pelaksanaan')->nullable();
    $table->string('tempat_pelaksanaan', 200)->nullable();
    $table->text('susunan_panitia')->nullable();
    $table->string('file_proposal', 255)->nullable();
    $table->date('tanggal_pengajuan')->nullable();
    $table->enum('status_proposal', ['draft', 'diajukan',
        'disetujui', 'ditolak', 'revisi'])->default('draft');
    $table->text('catatan_revisi')->nullable();
    $table->string('disetujui_oleh', 100)->nullable();
    $table->date('tanggal_persetujuan')->nullable();
    $table->timestamps();

    $table->index('status_proposal');
});
```

Tabel 'tenant\_proposal\_kegiatan' memiliki foreign key ke tabel 'tenant\_kegiatan', menunjukkan relasi antara kegiatan dan proposal. Kolom 'nomor\_proposal', 'judul\_proposal', dan berbagai kolom text menyimpan konten lengkap proposal. Status proposal menggunakan enum untuk tracking tahapan persetujuan, dan kolom 'disetujui\_oleh' serta 'tanggal\_persetujuan' mencatat informasi approval

## 3) Transaksi

```
Schema::create('tenant_transaksi_keuangan', function (Blueprint
$table) {
    $table->id();
    $table->foreignId('kegiatan_id')->nullable()->
        constrained('tenant_kegiatan')->onDelete('cascade');
    $table->foreignId('jenis_rekening_id')->constrained(
        'tenant_jenis_rekening')->onDelete('cascade');
    $table->foreignId('kelompok_rekening_id')->constrained(
        'tenant_kelompok_rekening')->onDelete('cascade');
    $table->string('nomor_transaksi')->unique();
    $table->date('tanggal_transaksi');
    $table->string('uraian');
    $table->text('keterangan_lainnya')->nullable();
    $table->decimal('jumlah', 15, 2);
    $table->enum('jenis_transaksi', ['masuk', 'keluar']);
    $table->enum('metode_pembayaran', ['tunai', 'transfer',
        'cek', 'lainnya']);
    $table->string('nomor_bukti')->nullable();
    $table->string('file_bukti')->nullable();
    $table->enum('status', ['draft', 'verified',
        'cancelled'])->default('draft');
    $table->unsignedBigInteger('created_by')->nullable();
    $table->unsignedBigInteger('verified_by')->nullable();
    $table->timestamp('verified_at')->nullable();
    $table->text('catatan')->nullable();
    $table->timestamps();

    // Menetapkan kategori_transaksi di akhir tabel
    $table->enum('kategori_transaksi', [
        'kegiatan',
        // Transaksi terkait kegiatan/proposal
        'operasional', // Iuran, ALK, listrik, dll
        'sumbangan', // Donasi masuk/keluar
        'investasi', // Pembelian aset
        'lainnya', // Transaksi lain-lain
    ])->default('operasional');

    // Indexes
    $table->index('status');
    $table->index('kategori_transaksi');
});
```

```

Route::prefix('dashboard/transaksi')->name('dashboard.transaksi')->group(function () {
    // Dashboard Transaksi
    Route::get('/dashboard', [TenantTransaksiKeuanganController::class, 'dashboard'])->name(
        'dashboard');

    // API endpoint untuk options (untuk dropdown, select, dll)
    Route::get('/api/options', [TenantTransaksiKeuanganController::class, 'options'])->name(
        'options');

    // Specifc Filter Routes
    Route::get('/filter/pemasukan', [TenantTransaksiKeuanganController::class, 'pemasukan'])->
        name('filter.pemasukan');
    Route::get('/filter/pengeluaran', [TenantTransaksiKeuanganController::class, 'pengeluaran']
    )->name('filter.pengeluaran');

    // Filter Routes berdasarkan Status
    Route::get('/status/status', [TenantTransaksiKeuanganController::class, 'filterByStatus']
    )->name('filter.status');

    // CRUD Routes
    Route::get('/', [TenantTransaksiKeuanganController::class, 'index'])->name('index');
    Route::get('/create', [TenantTransaksiKeuanganController::class, 'create'])->name('create');
    Route::post('/', [TenantTransaksiKeuanganController::class, 'store'])->name('store');
    Route::get('/{id}', [TenantTransaksiKeuanganController::class, 'show'])->name('show');
    Route::get('/{id}/edit', [TenantTransaksiKeuanganController::class, 'edit'])->name('edit');
    Route::put('/{id}', [TenantTransaksiKeuanganController::class, 'update'])->name('update');
    Route::delete('/{id}', [TenantTransaksiKeuanganController::class, 'destroy'])->name('destroy');
});

// Update status transaksi
Route::patch('/{id}/status', [TenantTransaksiKeuanganController::class, 'updateStatus'])->
    name('update.status');

// Verifikasi transaksi
Route::post('/{id}/verify', [TenantTransaksiKeuanganController::class, 'verify'])->name(
    'verify');
});

```

Route group untuk transaksi keuangan memiliki kompleksitas tertinggi dengan dashboard khusus, API endpoints untuk dropdown options, filter berdasarkan jenis transaksi (pemasukan/pengeluaran), CRUD routes, dan action routes khusus untuk update status dan verifikasi transaksi

#### d. Route Registrasi Karang Taruna

```

// Routes register karang taruna
Route::get('/karangtaruna/register', [KarangtarunaController::class, 'register'])->name(
    'karangtaruna.register');
Route::post('/karangtaruna/register', [KarangtarunaController::class, 'registerStore'])->name(
    'karangtaruna.register.store');

// Publik Karang Taruna Routes
Route::get('/karangtaruna', [KarangtarunaController::class, 'index'])->name('karangtaruna.index');
// Show detail Karang Taruna tertentu
Route::get('/karangtaruna/{karangtaruna}', [KarangtarunaController::class, 'show'])->name(
    'karangtaruna.show');

// Edit form karang taruna
Route::get('/karangtaruna/{karangtaruna}/edit', [KarangtarunaController::class, 'edit'])->name(
    'karangtaruna.edit');

// Update Karang Taruna
Route::put('/karangtaruna/{karangtaruna}', [KarangtarunaController::class, 'update'])->name(
    'karangtaruna.update');

// Route khusus untuk profil
Route::get('/profile/{karangtaruna}', [KarangtarunaController::class, 'profile'])->name(
    'karangtaruna.profile');

```

Route publik untuk registrasi dan manajemen Karang Taruna yang tidak memerlukan authentication, memungkinkan organisasi untuk mendaftar dan mengelola profil mereka sebelum mengakses fitur tenant

Konfigurasi routes mengimplementasikan struktur yang sistematis untuk arsitektur multi-tenant dengan pemisahan yang jelas antara route publik dan protected routes, organisasi berdasarkan modul fungsional, dan konsistensi penamaan yang memudahkan development dan maintenance aplikasi



## Lampiran 5. Sertifikat HKI

  
**REPUBLIK INDONESIA**  
**KEMENTERIAN HUKUM**

### SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC002025077707, 28 Juni 2025

**Pencipta**  
Nama : Adam Ezra Haqqani, M. Nishom, S.Kom., M.Kom. dkk  
Alamat : Jl. Pintu Air Gang Masjid Jami Al-Falah, RT. 08 RW. 02, Desa Kanangjati, Tarub, Kab. Tegal, Jawa Tengah, 52184  
Kewarganegaraan : Indonesia

**Pemegang Hak Cipta**  
Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama  
Alamat : Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142  
Kewarganegaraan : Indonesia  
Jenis Ciptaan : Program Komputer  
Judul Ciptaan : Platform Pengelolaan Kegiatan dan Keuangan Karang Taruna Berbasis Website Menggunakan Arsitektur Multi-Tenancy

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 28 Juni 2025, di Kota Tegal

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor Pencatatan : 000917968

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.  
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM  
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL  
u.b  
Direktur Hak Cipta dan Desain Industri

  
Agung Dumasasengko, SH., MH.  
NIP. 196912261994031001





**Disclaim:**  
1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan saat penyusunan, Menteri bertanggung untuk mencatat saat pencatatan permohonan.  
2. Surat Pencatatan ini telah diupload secara elektronik menggunakan segel elektronik yang diterbitkan oleh Badan Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara.  
3. Surat Pencatatan ini dapat dibuktikan kebenarannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.



Lampiran 6. Lembar Bimbingan



**D IV TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

**LEMBAR BIMBINGAN TUGAS AKHIR**

Nama : Adam Ezra Haqqani  
 NIM : 21090126  
 No. Ponsel : 0895384217740  
 Judul TA : Platform Pengelolaan Kegiatan dan Keuangan Karang Taruna Berbasis Website. menggunakan Arsitektur Multi tenancy.  
 Dosen Pembimbing I : M. Nishom, S.Kom., M.Kom.

| No | Tanggal     | Pemeriksaan   | Perbaikan Yang Perlu Dilakukan                                                                             | Paraf Pembimbing |
|----|-------------|---------------|------------------------------------------------------------------------------------------------------------|------------------|
| 1. | 3/6 - 2025  | - Perancangan | ✓ DB - Buat DB Master<br>✓ Simulasi.<br>Lakukan simulasi terhadap DB untuk (minimal) 2 akun Karang Taruna. |                  |
| 2. | 12/6 - 2025 | - Sistem      | - Selesaikan pembangunan sistem.<br>(prioritaskan fungsi utama).                                           |                  |
| 3. | 19/6 - 2025 | - sistem      | - Lanjutkan pengembangan sistem                                                                            |                  |
|    |             | - Laporan     | - Mulailah susun Laporan BAB I-III                                                                         |                  |

|    |         |                                                                                                              |                                                                                                                                                      |                                                                                                                                                                                                                                                                           |
|----|---------|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7. | 26/6-25 | <ul style="list-style-type: none"> <li>- Laporan</li> <li>- HKI</li> <li>- Taat</li> <li>- Serbon</li> </ul> | <ul style="list-style-type: none"> <li>- Perbaiki sesuai catatan pada laporan</li> <li>- ok Lanjutkan</li> <li>- ok.</li> <li>- Minimal 2</li> </ul> | <br><br><br><br> |
| 5. | 7/7-25  | Kelengkapan berkas Pendaftaran sidang                                                                        | <ul style="list-style-type: none"> <li>Ok.</li> <li>Lanjutkan</li> </ul>                                                                             |                                                                                                                                                                                                                                                                           |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|--|--|--|--|--|

Tegal, 7 Juli 2025  
Dosen Pembimbing I



M. Nishom, S.Kom., M.Kom.  
NIPY. 09.017.337

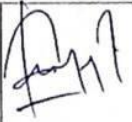


**D IV TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

**LEMBAR BIMBINGAN TUGAS AKHIR**

Nama : Adam Ezra Haqqani  
 NIM : 21090126  
 No. Ponsel : 0895384217740  
 Judul TA : Platform Pengelolaan Kegiatan dan Keuangan  
 Karang Taruna Berbasis Website.  
 Dosen Pembimbing II : Taufiq Abidin, S.Pd., M.Kom.

| No | Tanggal | Pemeriksaan | Perbaikan Yang Perlu Dilakukan | Paraf Pembimbing                                                                      |
|----|---------|-------------|--------------------------------|---------------------------------------------------------------------------------------|
| 1. | 21/5-25 | Konsep      | - Ok<br>- lanjut produk 80%    |  |
| 2. | 25/5-25 | Produk      | Validasi data & perhalusan     |  |
| 3. | 4/6-25  | Produk      | Perbaiki sesuai arahan         |  |
| 4. | 11/6-25 | Produk      | Oke, lanjut laporan            |  |

|    |         |         |                                        |                                                                                      |
|----|---------|---------|----------------------------------------|--------------------------------------------------------------------------------------|
| 5. | 13/6-25 | laporan | Perbaiki: sesuai Format                |   |
| 6. | 18/6-25 | laporan | Perbaiki: sesuai dengan yang di bundel |   |
| 7. | 25/6-25 | laporan | lampiran & lengkapi                    |   |
| 8. | 27/6-25 | laporan | Oke, hap<br>up on                      |  |

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|--|--|--|--|--|

Tegal, *juu 2025*  
Dosen Pembimbing II



Taufiq Abidin, S.Pd., M.Kom.  
NIPY. 06.014.184