**PERANCANGAN TUGAS AKHIR**

**Fahri dwi hermawan XI-RPL**

### ****Analisis Masalah****

Saat ini, proses transaksi di Pesantren Cemerlang Annajach Bahrul Ulum masih dilakukan secara manual, di mana orang tua atau wali santri (walisantri) melakukan pembayaran melalui m-banking atau ATM, kemudian mengirimkan bukti transfer (bukti TF) melalui WhatsApp untuk konfirmasi. Setelah itu, bendahara mencatat pembayaran tersebut secara manual di kertas dan mengirimkan catatan tersebut ke grup wali santri. Proses ini sering kali menyebabkan kesalahan atau kelalaian dalam mencatat pembayaran, serta menambah beban administratif.

### ****Tujuan Sistem****

Membangun sebuah platform berbasis web yang memungkinkan wali santri dan pihak pesantren untuk melakukan transaksi secara lebih mudah, efisien, dan cepat. Dengan sistem ini, wali santri dapat melakukan pembayaran secara langsung melalui platform, dan sistem akan secara otomatis mengonfirmasi dan merekap transaksi.

**1. Fitur dan Peran Pengguna**

Aplikasi ini akan memiliki dua jenis pengguna utama: **walisantri** dan **Admin**.

**Role dan Permissions**

* **Admin**: Admin memiliki akses penuh untuk mengelola data santri, kategori pembayaran, tagihan, dan verifikasi pembayaran.
* **Santri**: Santri hanya dapat melihat tagihan mereka, memilih cara pembayaran, dan membayar tagihan.

Untuk manajemen role dan permissions, Anda bisa menggunakan paket **Spatie Laravel Permission** untuk mempermudah pengelolaan akses.

**2. Fitur Utama**

**a. Manajemen Santri (Admin)**

* **Menambah, mengedit, dan menghapus data santri**.
* **Melihat riwayat pembayaran santri** dan status tagihan mereka.

**b. Kategori Pembayaran (Admin)**

* **Menambah, mengedit, dan menghapus kategori pembayaran** (misalnya SPP, Makan, Buku, dll).
* Setiap kategori memiliki **nominal yang ditentukan** yang bisa berubah sesuai dengan periode (misalnya per bulan).

**c. Tagihan Santri (Admin)**

* **Menggenerate tagihan** untuk santri berdasarkan kategori dan periode.
* Jika ada tagihan yang belum dibayar, maka tagihan bulan baru akan digabungkan dengan tagihan bulan sebelumnya.

**d. Pembayaran (Santri)**

* **Santri bisa memilih untuk membayar seluruh tagihan atau hanya tagihan untuk bulan tertentu**.
* **Upload bukti pembayaran** dan memilih metode pembayaran (misalnya bank transfer atau gateway pembayaran online).

**e. Verifikasi Pembayaran (Admin)**

* Admin dapat memverifikasi pembayaran yang dilakukan oleh santri dan mengubah status tagihan menjadi **"paid"** atau **"pending"**.
* Admin bisa mengelola bukti pembayaran yang di-upload oleh santri.

**f. Laporan Keuangan (Admin)**

* Admin bisa melihat laporan keuangan seperti **jumlah pembayaran per bulan** atau **daftar pembayaran yang tertunda**.
* Admin juga bisa mengekspor laporan dalam format **Excel** atau **CSV**.

### 3. ****1. Tabel**** users ****(Santri dan Admin)****

Tabel ini menyimpan data pengguna yang terdiri dari **santri** dan **admin**. Kolom-kolom dalam tabel ini adalah sebagai berikut:

| **Kolom** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | BIGINT | Primary Key, Auto Increment |
| name | VARCHAR(255) | Nama lengkap pengguna |
| email | VARCHAR(255) | Alamat email, Unique |
| password | VARCHAR(255) | Enkripsi password |
| role | ENUM('santri', 'admin') | Role pengguna, menentukan akses ke fitur aplikasi |
| created\_at | TIMESTAMP | Waktu pembuatan akun |
| updated\_at | TIMESTAMP | Waktu update terakhir akun |

**Relasi**:

* Setiap pengguna (santri atau admin) memiliki **role** yang menentukan akses mereka ke aplikasi.

### ****2. Tabel**** categories ****(Kategori Pembayaran)****

Tabel ini menyimpan data kategori pembayaran seperti **SPP**, **Makan**, **Buku**, dll. Setiap kategori memiliki **nominal** yang tetap, yang mungkin akan diperbarui secara berkala.

| **Kolom** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | BIGINT | Primary Key, Auto Increment |
| name | VARCHAR(255) | Nama kategori pembayaran (SPP, Makan, dll.) |
| amount | DECIMAL(10, 2) | Nominal pembayaran untuk kategori ini |
| description | TEXT | Deskripsi tentang kategori pembayaran |
| created\_at | TIMESTAMP | Waktu pembuatan kategori pembayaran |
| updated\_at | TIMESTAMP | Waktu update terakhir kategori pembayaran |

**Relasi**:

* Setiap **kategori** pembayaran akan digunakan dalam pembuatan **tagihan** untuk santri.

### ****3. Tabel**** invoices ****(Tagihan Santri)****

Tabel ini menyimpan data tagihan untuk setiap santri yang terhubung dengan kategori pembayaran tertentu. Tabel ini juga akan mencatat **status** tagihan apakah sudah dibayar atau belum, serta tanggal jatuh temponya.

| **Kolom** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | BIGINT | Primary Key, Auto Increment |
| user\_id | BIGINT | Foreign Key ke tabel users (santri) |
| category\_id | BIGINT | Foreign Key ke tabel categories (kategori pembayaran) |
| amount | DECIMAL(10, 2) | Jumlah tagihan yang harus dibayar |
| status | ENUM('pending', 'paid', 'partially\_paid') | Status pembayaran tagihan |
| due\_date | DATE | Tanggal jatuh tempo pembayaran |
| paid\_amount | DECIMAL(10, 2) | Jumlah yang sudah dibayar oleh santri (jika sudah ada pembayaran) |
| created\_at | TIMESTAMP | Waktu pembuatan tagihan |
| updated\_at | TIMESTAMP | Waktu update terakhir tagihan |

**Relasi**:

* Setiap **tagihan** berhubungan dengan seorang **santri** (user\_id) dan satu **kategori pembayaran** (category\_id).
* **Status** tagihan bisa berupa **pending** (belum dibayar), **paid** (sudah dibayar), atau **partially\_paid** (sebagian sudah dibayar).

### ****4. Tabel**** payments ****(Pembayaran Santri)****

Tabel ini menyimpan data tentang pembayaran yang dilakukan oleh santri. Tabel ini mencatat **tagihan** yang dibayar, **jumlah yang dibayar**, dan **tanggal pembayaran**.

| **Kolom** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | BIGINT | Primary Key, Auto Increment |
| invoice\_id | BIGINT | Foreign Key ke tabel invoices (tagihan yang dibayar) |
| amount\_paid | DECIMAL(10, 2) | Jumlah yang dibayar oleh santri |
| payment\_date | TIMESTAMP | Waktu pembayaran dilakukan |
| payment\_method | ENUM('bank\_transfer', 'e-wallet', 'cash', 'other') | Metode pembayaran yang digunakan oleh santri |
| proof\_of\_payment | VARCHAR(255) | URL atau path file bukti pembayaran yang di-upload oleh santri |
| created\_at | TIMESTAMP | Waktu pembuatan pembayaran |
| updated\_at | TIMESTAMP | Waktu update terakhir pembayaran |

**Relasi**:

* Setiap **pembayaran** berhubungan dengan satu **tagihan** (invoice\_id).
* **Pembayaran** akan mencatat informasi **jumlah yang dibayar** serta **metode pembayaran** yang digunakan.

### ****Relasi Antar Tabel****

1. **Users → Invoices**:
   * Setiap **santri** dapat memiliki banyak **tagihan**.
   * Relasi ini menghubungkan tabel users (santri) dengan tabel invoices, dimana **user\_id** pada tabel invoices mengacu pada **id** pada tabel users.
2. **Categories → Invoices**:
   * Setiap **kategori pembayaran** dapat memiliki banyak **tagihan**.
   * Relasi ini menghubungkan tabel categories dengan tabel invoices, dimana **category\_id** pada tabel invoices mengacu pada **id** pada tabel categories.
3. **Invoices → Payments**:
   * Setiap **tagihan** dapat memiliki banyak **pembayaran**.
   * Relasi ini menghubungkan tabel invoices dengan tabel payments, dimana **invoice\_id** pada tabel payments mengacu pada **id** pada tabel invoices.

### ****Diagram Relasi****

Untuk memberikan gambaran lebih jelas mengenai relasi antar tabel, berikut adalah penjelasan ringkas diagram relasi yang bisa digambarkan:

1. **Tabel users** berhubungan dengan **Tabel invoices** melalui user\_id (satu ke banyak).
2. **Tabel categories** berhubungan dengan **Tabel invoices** melalui category\_id (satu ke banyak).
3. **Tabel invoices** berhubungan dengan **Tabel payments** melalui invoice\_id (satu ke banyak).

Dengan struktur dan relasi database ini, Anda dapat dengan mudah mengelola data pembayaran, tagihan, dan kategori pembayaran untuk santri dan admin di aplikasi yang Anda rancang.

**4. Alur Sistem**

**a. Mengenerate Tagihan Bulan Baru**

1. **Cek Tagihan Bulan Sebelumnya**: Setiap kali bulan baru dimulai, sistem akan memeriksa apakah ada tagihan yang belum dibayar dari bulan sebelumnya.
2. **Gabungkan Tagihan Lama dan Baru**: Jika ada tagihan bulan sebelumnya yang belum dibayar (misalnya SPP bulan Januari sebesar Rp 300.000), dan pada bulan Februari ada tagihan dengan kategori yang sama, maka total tagihan untuk bulan Februari akan menjadi **Rp 600.000** (penjumlahan tagihan bulan Januari dan Februari).
3. **Buat Tagihan Baru**: Setelah tagihan digabung, tagihan baru akan otomatis dibuat dengan status **pending**.

**b. Proses Pembayaran oleh Santri**

1. Santri login dan masuk ke halaman dashboard mereka.
2. Santri melihat **total tagihan** mereka (yang sudah digabung jika ada tagihan bulan sebelumnya yang belum dibayar).
3. Santri memilih apakah ingin **membayar seluruh tagihan** atau **tagihan per bulan**.
4. Setelah memilih metode pembayaran, santri meng-upload **bukti pembayaran**.
5. Pembayaran yang diterima akan diproses dan sistem akan mengubah status tagihan sesuai dengan jumlah yang dibayar (misalnya status menjadi **paid** atau **partially\_paid**).

**c. Verifikasi Pembayaran oleh Admin**

1. Admin melihat **riwayat pembayaran** yang dilakukan oleh santri.
2. Admin memeriksa bukti pembayaran yang di-upload oleh santri dan memastikan jumlah yang dibayar sesuai.
3. Setelah verifikasi, admin mengubah status tagihan menjadi **paid** (jika seluruhnya dibayar) atau **partially\_paid** (jika sebagian dibayar).

**5. Tampilan UI**

**a. Halaman Dashboard Santri**

Setelah login, santri akan melihat halaman dashboard yang menampilkan:

* **Total Tagihan**: Rincian total tagihan yang harus dibayar, termasuk tagihan bulan sebelumnya yang belum dibayar.
* **Pilihan Pembayaran**: Opsi untuk membayar **seluruh tagihan** atau **tagihan per bulan**.
* **Riwayat Pembayaran**: Daftar pembayaran yang telah dilakukan, dengan status tagihan (misalnya **paid**, **pending**, **partially\_paid**).

**b. Halaman Pembayaran**

Halaman pembayaran memungkinkan santri untuk memilih cara pembayaran dan mengunggah bukti pembayaran. Elemen-elemen pada halaman ini meliputi:

* **Total Tagihan** yang harus dibayar.
* **Pilihan Pembayaran**: [Bayar Seluruh Tagihan] atau [Bayar Per Bulan].
* **Form Upload Bukti Pembayaran**.
* **Nominal Pembayaran** yang harus dilakukan.

**c. Halaman Admin**

Admin memiliki dashboard yang menampilkan:

* **Ringkasan Pembayaran**: Daftar tagihan yang belum dibayar dan total pembayaran yang sudah diterima.
* **Manajemen Santri**: Daftar santri dan status tagihan mereka.
* **Verifikasi Pembayaran**: Halaman untuk melihat bukti pembayaran dan mengonfirmasi pembayaran yang telah dilakukan oleh santri.

**6. Teknologi dan Paket yang Digunakan**

* **Laravel**: Framework utama untuk backend aplikasi.
* **Laravel Blade**: Untuk tampilan frontend.
* **Laravel Jetstream / Breeze**: Untuk sistem autentikasi dan manajemen pengguna.
* **Spatie Laravel Permission**: Untuk pengelolaan role dan permissions.
* **Laravel Excel**: Untuk mengekspor laporan ke dalam format Excel.
* **Midtrans / Doku / Stripe**: Untuk integrasi dengan sistem pembayaran seperti gateway bank atau e-wallet.
* **MySQL**: Sebagai database utama untuk aplikasi.

**7. Pengujian dan Implementasi**

Setelah aplikasi dibangun, penting untuk melakukan pengujian menyeluruh pada fitur-fitur utama seperti:

* **Penjumlahan Tagihan**: Pastikan tagihan bulan sebelumnya yang belum dibayar digabung dengan tagihan bulan baru dengan benar.
* **Pilihan Pembayaran**: Pastikan santri dapat memilih pembayaran secara **seluruhnya** atau **per bulan**.
* **Verifikasi Pembayaran**: Pastikan admin dapat memverifikasi dan mengonfirmasi pembayaran santri dengan tepat.

**8. Fitur Tambahan (Opsional)**

* **Notifikasi Pembayaran**: Sistem notifikasi via email atau aplikasi untuk mengingatkan santri tentang pembayaran atau mengonfirmasi pembayaran.
* **Grafik Keuangan**: Menampilkan grafik atau laporan visual untuk admin, misalnya pendapatan per bulan atau pembayaran yang tertunda.
* **Multi-Bahasa**: Aplikasi dapat diubah untuk mendukung beberapa bahasa jika diperlukan.

| **Kolom** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | BIGINT | Primary Key, Auto Increment |
| nama | VARCHAR(255) | Nama lengkap pengguna |
| email | VARCHAR(255) | Alamat email, Unique |
| kata\_sandi | VARCHAR(255) | Enkripsi kata sandi |
| peran | ENUM('santri', 'admin') | Peran pengguna dalam sistem |
| tempat\_lahir | VARCHAR(255) | Tempat lahir santri |
| tanggal\_lahir | DATE | Tanggal lahir santri |
| alamat | TEXT | Alamat santri |
| sekolah | VARCHAR(255) | Nama sekolah santri |
| jenis\_kelamin | ENUM('L', 'P') | Jenis kelamin (L = Laki-laki, P = Perempuan) |
| nama\_wali | VARCHAR(255) | Nama Orang Tua/Wali |
| no\_hp\_wali | VARCHAR(20) | Nomor HP Orang Tua/Wali |
| dibuat\_pada | TIMESTAMP | Waktu pembuatan akun |