**Dokumentasi Teknis B'MINE (Mining Identity Number Electronic)**

**Struktur MVC (Model-View-Controller)**

**Models**

1. **KaryawanModel** (app/Models/KaryawanModel.php)
   * Mengelola data karyawan internal
   * Tabel: karyawan
   * Primary key: id\_kar
   * Atribut utama: nik, nama, departement, jabatan, email, password, level
   * Tidak menggunakan timestamps
2. **UserModel** (app/Models/UserModel.php)
   * Mengelola akun pengguna eksternal
   * Tabel: login\_external
   * Primary key: id\_login\_ext
   * Atribut utama: name, email, password, level
   * Tidak menggunakan timestamps
3. **DataReqModel** (app/Models/DataReqModel.php)
   * Menyimpan permintaan perizinan dalam proses
   * Tabel: data\_req
   * Atribut utama: nik, kode, nama, jab, dept, status, validasi\_in, access
   * Relasi: unitUsers (one-to-many dengan UnitUser)
   * Casting: access dan reject\_history sebagai array
   * Tidak menggunakan timestamps
4. **Data\_m\_s\_Model** (app/Models/Data\_m\_s\_Model.php)
   * Menyimpan perizinan yang telah disetujui
   * Tabel: data\_m\_s
   * Primary key: id
   * Atribut utama: serupa dengan DataReqModel
   * Relasi: unitUsers (one-to-many dengan UnitUser)
   * Menggunakan timestamps
5. **DataRejectModel** (app/Models/DataRejectModel.php)
   * Menyimpan permintaan yang ditolak
   * Tabel: data\_reject
   * Primary key: id
   * Atribut utama: kode, nik, nama, jab, dept, reject\_history
   * Casting: reject\_history sebagai array
   * Tidak menggunakan timestamps
6. **UnitModel** (app/Models/UnitModel.php)
   * Mengelola data unit peralatan
   * Tabel: units
   * Primary key: id\_units
   * Atribut utama: nama\_unit
   * Relasi: userUnits (one-to-many dengan UserUnit)
   * Tidak menggunakan timestamps
7. **UnitUser** (app/Models/UnitUser.php)
   * Relasi antara unit dan permintaan perizinan
   * Tabel: user\_unit
   * Primary key: id\_user\_unit
   * Atribut utama: unit, type\_unit, id\_uur
   * Relasi: unitData (belongs-to dengan UnitModel), dataReq (belongs-to dengan DataReqModel)
   * Tidak menggunakan timestamps
8. **DashboardModel** (app/Models/DashboardModel.php)
   * Model untuk statistik dashboard
   * Tabel: data\_m\_s
   * Metode statis: countAccess1(), countAccess2()

**Controllers**

1. **AuthController** (app/Http/Controllers/AuthController.php)
   * Menangani autentikasi pengguna
   * Metode: login(), auth(), logout(), resetPassword()
   * Fitur: login dengan NIK/email, verifikasi password, manajemen sesi
2. **DashboardController** (app/Http/Controllers/DashboardController.php)
   * Mengelola tampilan dan data dashboard
   * Metode: index(), about(), reset\_password(), getPermitRequestsData(), getCustomRangeData()
   * Fitur: statistik dashboard, grafik, peringatan kedaluwarsa
3. **RequestController** (app/Http/Controllers/RequestController.php)
   * Menangani pengajuan permintaan perizinan
   * Metode: index(), not\_found(), get\_data\_nik(), insert\_request(), data()
   * Fitur: pencarian NIK, pengisian form, upload dokumen, penyimpanan permintaan
4. **PersonalTaskController** (app/Http/Controllers/PersonalTaskController.php)
   * Mengelola tugas berbasis peran
   * Metode: personalTask(), sheTask(), pjoTask(), becTask(), kttTask(), viewData(), dll.
   * Fitur: tampilan tugas, persetujuan/penolakan, generasi kartu ID, QR code
5. **AkunController** (app/Http/Controllers/AkunController.php)
   * Mengelola akun pengguna
   * Metode: akun\_external(), akun\_internal(), store\_external(), store\_internal(), dll.
   * Fitur: manajemen akun, perubahan level, penghapusan, perubahan password
6. **CompletedController** (app/Http/Controllers/CompletedController.php)
   * Menangani permintaan yang telah selesai
   * Metode: data\_completed(), accept()
   * Fitur: tampilan permintaan selesai, penerimaan final
7. **HistoryController** (app/Http/Controllers/HistoryController.php)
   * Menangani riwayat permintaan
   * Metode: index()
   * Fitur: tampilan riwayat
8. **OutstandingController** (app/Http/Controllers/OutstandingController.php)
   * Menangani permintaan yang sedang berlangsung
   * Metode: index()
   * Fitur: tampilan permintaan yang sedang berlangsung
9. **ChunkedUploadController** (app/Http/Controllers/ChunkedUploadController.php)
   * Menangani upload file berukuran besar secara chunk
   * Metode: uploadChunk(), deleteDirectory()

**Views**

1. **Layouts**
   * main.blade.php: Template utama dengan header, footer, sidebar
   * navbar.blade.php: Navigasi atas
   * sidebar.blade.php: Navigasi samping dengan menu berbasis peran
   * footer.blade.php: Footer aplikasi
2. **Auth**
   * login.blade.php: Form login
3. **Dashboard**
   * dashboard.blade.php: Dashboard untuk pengguna internal
   * dashboard\_external.blade.php: Dashboard untuk pengguna eksternal
   * about.blade.php: Halaman tentang aplikasi
4. **Request**
   * request.blade.php: Form pencarian NIK
   * get\_data.blade.php: Form pengajuan permintaan
   * not\_found.blade.php: Tampilan ketika NIK tidak ditemukan
5. **Personal Task**
   * data\_req\_she.blade.php: Tugas untuk SHE
   * data\_req\_pjo.blade.php: Tugas untuk PJO
   * data\_req\_bec.blade.php: Tugas untuk BEC
   * data\_req\_ktt.blade.php: Tugas untuk KTT
   * data\_req\_rejected.blade.php: Permintaan yang ditolak
6. **Akun**
   * internal.blade.php: Manajemen akun internal
   * external.blade.php: Manajemen akun eksternal
   * add\_internal.blade.php: Form tambah akun internal
   * add\_external.blade.php: Form tambah akun eksternal
7. **History**
   * history.blade.php: Riwayat permintaan
8. **Outstanding**
   * outstanding.blade.php: Permintaan yang sedang berlangsung
9. **Completed**
   * data\_completed.blade.php: Permintaan yang telah selesai
10. **ID Card**
    * print\_front.blade.php: Template kartu ID depan
    * print\_back.blade.php: Template kartu ID belakang
11. **Verifikasi**
    * view\_qr.blade.php: Halaman verifikasi QR code
12. **Setting**
    * update\_password.blade.php: Form perubahan password
    * reset\_password.blade.php: Form reset password

**Alur Proses untuk Pengembang**

**1. Konfigurasi Awal**

1. **Clone Repository dan Setup Environment**
2. git clone [repository-url]
3. cd [project-folder]
4. composer install
5. cp .env.example .env
6. php artisan key:generate
7. **Konfigurasi Database**
   * Edit file .env dan sesuaikan koneksi database
8. DB\_CONNECTION=mysql
9. DB\_HOST=127.0.0.1
10. DB\_PORT=3306
11. DB\_DATABASE=bmine
12. DB\_USERNAME=root
13. DB\_PASSWORD=
14. **Migrasi Database**
15. php artisan migrate
16. **Link Storage untuk File Upload**
17. php artisan storage:link

**2. Mekanisme Autentikasi**

1. **Proses Login**
   * Controller: AuthController@auth
   * Metode autentikasi mendukung login dengan NIK atau email
   * Password dienkripsi menggunakan MD5
   * Setelah login berhasil, sesi dibuat dengan informasi user
   * Middleware EnsureUserIsLoggedIn digunakan untuk melindungi rute
2. **Struktur Sesi User**
3. Session::put('logged\_in\_user', [
4. 'identifier' => $request->identifier, // NIK atau email
5. 'nama' => $user['nama'],
6. 'level' => $level, // Role pengguna
7. 'departement' => $departement,
8. 'area' => $area, // Untuk user KTT
9. ]);
10. **Logout**
    * Controller: AuthController@logout
    * Menghapus sesi user dan mengarahkan ke halaman login

**3. Alur Pengajuan Permintaan**

1. **Pencarian NIK**
   * Controller: RequestController@get\_data\_nik
   * View: request.blade.php
   * Inputkan NIK karyawan
   * Validasi NIK dengan memeriksa keberadaan di database
   * Cek apakah ada permintaan yang sedang berlangsung untuk NIK tersebut
2. **Pengisian Form Permintaan**
   * Controller: RequestController@get\_data\_nik
   * View: get\_data.blade.php
   * Isi data permintaan:
     + Informasi pribadi (otomatis dari database)
     + Upload foto
     + Pemilihan akses area
     + Upload dokumen pendukung
     + Pemilihan jenis izin (1: Mine Permit, 2: Simper & Mine Permit)
     + Jika Simper, tambahkan unit dengan klasifikasi PRTIO
3. **Penyimpanan Permintaan**
   * Controller: RequestController@insert\_request
   * Validasi form input
   * Generate kode unik
   * Upload dan simpan file ke direktori terkait
   * Simpan data ke tabel data\_req dengan status awal 1 (SHE)
   * Jika jenis izin adalah 2 (Simper & Mine Permit), simpan data unit ke tabel user\_unit

**4. Alur Persetujuan**

1. **Persetujuan SHE**
   * Controller: PersonalTaskController@sheTask, PersonalTaskController@approveDataShe
   * View: data\_req\_she.blade.php
   * Periksa dokumen dan detail permintaan
   * Jika disetujui, status diubah menjadi 2 (PJO)
   * Jika ditolak, data dipindahkan ke tabel data\_reject dengan alasan penolakan
2. **Persetujuan PJO**
   * Controller: PersonalTaskController@pjoTask, PersonalTaskController@approveDataPjo
   * View: data\_req\_pjo.blade.php
   * Periksa detail permintaan
   * Jika disetujui, status diubah menjadi 3 (BEC)
   * Jika ditolak, data dipindahkan ke tabel data\_reject dengan alasan penolakan
3. **Persetujuan BEC**
   * Controller: PersonalTaskController@becTask, PersonalTaskController@approveDataBec
   * View: data\_req\_bec.blade.php
   * Periksa detail permintaan
   * Jika disetujui, status diubah menjadi 4 (KTT)
   * Jika ditolak, data dipindahkan ke tabel data\_reject dengan alasan penolakan
4. **Persetujuan KTT**
   * Controller: PersonalTaskController@kttTask, PersonalTaskController@approveDataKtt
   * View: data\_req\_ktt.blade.php
   * KTT dari area tertentu (BT, FSP, TA, TJ) menyetujui permintaan
   * Persetujuan disimpan dalam kolom JSON ktt
   * Jika semua area disetujui, data dipindahkan ke tabel data\_m\_s (final)
   * Jika ditolak, data dipindahkan ke tabel data\_reject dengan alasan penolakan
5. **Selesai**
   * Controller: CompletedController@data\_completed, CompletedController@accept
   * View: data\_completed.blade.php
   * Status diubah menjadi 1 (aktif) di tabel data\_m\_s
   * Kartu ID dapat dihasilkan

**5. Generasi ID Card dan QR Code**

1. **Generasi Kartu ID**
   * Controller: PersonalTaskController@generateIdCardFront, PersonalTaskController@generateIdCardBack
   * View: idcard/print\_front.blade.php, idcard/print\_back.blade.php
   * Mengambil data dari tabel data\_m\_s
   * Generate QR code yang mengarah ke URL verifikasi
   * Buat tampilan kartu ID menggunakan HTML dan CSS
2. **Verifikasi QR Code**
   * Controller: PersonalTaskController@scanQR
   * View: verifikasi/view\_qr.blade.php
   * Menampilkan detail lengkap perizinan
   * Status akses area dan unit
   * Informasi kedaluwarsa

**6. Dashboard dan Pelaporan**

1. **Dashboard Internal**
   * Controller: DashboardController@index
   * View: dashboard/dashboard.blade.php
   * Menampilkan statistik:
     + Jumlah Mine Permit dan Simper
     + Permintaan yang akan kedaluwarsa
     + Status permintaan (SHE, PJO, BEC, KTT)
     + Grafik distribusi permintaan
2. **Dashboard Eksternal**
   * Controller: DashboardController@index
   * View: dashboard/dashboard\_external.blade.php
   * Tampilan terbatas sesuai peran pengguna eksternal
3. **API untuk Grafik**
   * Controller: DashboardController@getPermitRequestsData, DashboardController@getCustomRangeData
   * Menyediakan data JSON untuk grafik dan statistik
   * Filter berdasarkan tahun atau rentang tanggal kustom

**7. Manajemen Akun**

1. **Akun Internal**
   * Controller: AkunController@akun\_internal, AkunController@store\_internal
   * View: akun/internal.blade.php, akun/add\_internal.blade.php
   * Mengelola akun karyawan internal
   * Level: admin, section\_admin, she, pjo, user
2. **Akun Eksternal**
   * Controller: AkunController@akun\_external, AkunController@store\_external
   * View: akun/external.blade.php, akun/add\_external.blade.php
   * Mengelola akun pengguna eksternal
   * Level: bec, ktt
3. **Perubahan Level dan Password**
   * Controller: AkunController@update\_level, AkunController@changePassword, AkunController@updatePassword
   * View: setting/update\_password.blade.php
   * Mengubah level akun dan password pengguna

**8. Penggunaan Middleware**

1. **EnsureUserIsLoggedIn**
   * app/Http/Middleware/EnsureUserIsLoggedIn.php
   * Memastikan pengguna telah login sebelum mengakses halaman terlindungi
   * Memeriksa keberadaan sesi logged\_in\_user
   * Mengarahkan ke halaman login jika tidak ada sesi
2. **Konfigurasi Kernel**
   * app/Http/Kernel.php
   * Registrasi middleware yang digunakan
   * Group middleware web untuk halaman web umum
   * Group middleware api untuk endpoint API

**9. Rute Utama**

1. **Autentikasi**
   * GET /login: Tampilkan form login
   * POST /auth: Proses login
   * GET|POST /logout: Proses logout
2. **Dashboard**
   * GET /: Tampilkan dashboard utama
   * GET /about: Tampilkan halaman tentang aplikasi
3. **Request**
   * GET /request: Tampilkan form pencarian NIK
   * POST /request/search\_nik: Proses pencarian NIK
   * POST /request/insert: Simpan permintaan
4. **Personal Task**
   * GET /personal\_task/she/view: Tampilkan tugas SHE
   * GET /personal\_task/pjo/view: Tampilkan tugas PJO
   * GET /personal\_task/bec/view: Tampilkan tugas BEC
   * GET /personal\_task/ktt/view: Tampilkan tugas KTT
   * GET /personal\_task/rejected/view: Tampilkan permintaan ditolak
5. **Approval**
   * GET /approve/data\_she/{kode}: Proses persetujuan SHE
   * GET /approve/data\_pjo/{kode}: Proses persetujuan PJO
   * GET /approve/data\_bec/{kode}: Proses persetujuan BEC
   * GET /approve/data\_ktt/{kode}: Proses persetujuan KTT
   * POST /reject-request/{stage?}/{kode?}: Proses penolakan
6. **Akun**
   * GET /akun/internal: Tampilkan akun internal
   * GET /akun/external: Tampilkan akun eksternal
   * POST /akun/internal/store: Simpan akun internal
   * POST /akun/external/store: Simpan akun eksternal
   * POST /akun/update-password: Ubah password
7. **ID Card**
   * GET /idcard/generate-idcardFront/{nik}: Generate kartu ID depan
   * GET /idcard/generate-idcardBack/{nik}: Generate kartu ID belakang
   * GET /verifikasi/{nik}: Tampilkan halaman verifikasi QR code

**10. Tips untuk Pengembang Selanjutnya**

1. **Struktur Folder**
   * Models: app/Models/
   * Controllers: app/Http/Controllers/
   * Middleware: app/Http/Middleware/
   * Views: resources/views/
   * Routes: routes/web.php
   * Public assets: public/adminlte/
2. **Konvensi Penamaan**
   * Model: Singular, PascalCase (e.g., UserModel)
   * Controller: Singular + Controller, PascalCase (e.g., AuthController)
   * View: lowercase, snake\_case (e.g., login.blade.php)
   * Route: lowercase, kebab-case untuk URL (e.g., /user-profile)
3. **Penanganan File Upload**
   * Gunakan ChunkedUploadController untuk file besar
   * Storage path: storage/app/public/
   * File categories subdirectories: fotos, medical\_certificates, drivers\_licenses, attachments
4. **Debugging**
   * Gunakan Log::info(), Log::error() untuk logging
   * Cek log di storage/logs/laravel.log
   * Setting debug mode di .env file (APP\_DEBUG=true)
5. **Keamanan**
   * Password dienkripsi menggunakan MD5
   * Validasi CSRF pada semua form
   * Validasi input pengguna di controller
   * Akses terbatas berdasarkan peran pengguna
6. **Pustaka dan Dependensi Utama**
   * Laravel Framework
   * AdminLTE untuk template UI
   * Chart.js untuk visualisasi
   * SimpleSoftwareIO/QrCode untuk generasi QR code
7. **Struktur Database Utama**
   * karyawan: Akun internal
   * login\_external: Akun eksternal
   * data\_req: Permintaan dalam proses
   * data\_m\_s: Permintaan yang disetujui
   * data\_reject: Permintaan yang ditolak
   * user\_unit: Relasi antara permintaan dan unit
8. **Areas for Improvement**
   * Migrasi dari MD5 ke algoritma hash yang lebih aman (bcrypt)
   * Implementasi notifikasi real-time
   * Optimasi query database untuk performa yang lebih baik
   * Unit testing untuk fitur-fitur kritis