

# Product Requirements Document (PRD)

## Sistem Konsultasi & EMR - Klinik Swasta

Kod Dokumen: KLINIK-KonsultasiEMR-PR2026-01 **Tajuk:** Rekod Rawatan Pesakit (Electronic Medical Records) **Versi:** 1.0 **Tarikh:** 13 Januari 2026 **Penulis:** Product Team **Status:** Draft

### 1. Ringkasan Eksekutif

#### 1.1 Gambaran Keseluruhan

Sistem Konsultasi & EMR (Electronic Medical Records) adalah modul untuk merekod nota klinikal, diagnosis, dan maklumat rawatan pesakit secara digital. Sistem ini membolehkan doktor dan jururawat mendokumentasikan konsultasi dengan cepat menggunakan templates dan voice-to-text, serta berintegrasi dengan modul Farmasi, Billing, dan Makmal.

#### 1.2 Metadata

- **Nama Feature:** Rekod Rawatan Pesakit (EMR)
- **Modul:** Konsultasi & EMR
- **Submodul:** Rekod Rawatan Pesakit
- **Peranan Sasaran:** Doktor, Jururawat, Pengarah Perubatan
- **Keutamaan:** Tinggi
- **Status:** Perancangan
- **Anggaran Usaha:** Besar

#### 1.3 Objektif

- Mengurangkan masa dokumentasi konsultasi dari manual ke digital
- Memudahkan doktor merekod nota klinikal dengan templates dan voice-to-text
- Memastikan rekod perubatan lengkap dan teratur dalam format SOAP
- Mengintegrasikan EMR dengan Farmasi (prescription), Billing (invoice), dan Makmal (lab orders)
- Mematuhi standard klinikal dan PDPA dengan audit trail lengkap
- Meningkatkan kualiti rawatan dengan akses mudah kepada medical history

#### 1.4 Skop

##### Dalam Skop:

- Nota klinikal dalam format SOAP (Subjective, Objective, Assessment, Plan)
- Diagnosis dengan ICD-10 code search
- Voice-to-text untuk nota klinikal
- Templates nota klinikal mengikut jabatan/specialty
- Vital signs capture (BP, temperature, pulse, weight, BMI)
- Attachment images dan PDF (X-ray, lab results)
- Medical history timeline dengan expand/collapse
- Integration dengan Farmasi (auto-populate prescription)
- Integration dengan Billing (auto-generate bill items)
- Integration dengan Makmal (order tests + view results)
- Schedule follow-up appointment dari EMR
- Full audit trail dan version history
- Role-based access (Doktor vs Jururawat)
- Search dan filter EMR

##### Luar Skop:

- E-prescription printing (dalam modul Farmasi)
- Billing payment processing (dalam modul Billing)

- Lab test processing (dalam modul Makmal)
  - Telemedicine/video consultation
  - Patient portal access ke EMR (future phase)
- 

## 2. Pernyataan Masalah

### 2.1 Masalah Semasa

1. Dokumentasi konsultasi ambil masa lama kerana manual writing atau slow typing
2. Nota klinikal tidak teratur, sukar untuk trace medical history pesakit
3. Tiada standard format untuk nota klinikal, setiap doktor guna cara sendiri
4. Prescription, billing, dan lab orders perlu tulis berulang kali (redundant)
5. Sukar untuk search rekod lama pesakit dengan cepat
6. Tiada audit trail untuk perubahan pada rekod perubatan
7. Medical errors kerana tiada akses mudah kepada history dan allergies

### 2.2 Impak Kepada Perniagaan

- Doktor habiskan banyak masa untuk dokumentasi instead of patient care
- Kualiti rawatan menurun kerana tiada continuity of care
- Risk of medical errors tinggi tanpa proper medical history
- Compliance issues dengan standard klinikal dan PDPA
- Kehilangan hasil kerana billing items tertinggal
- Kepuasan pesakit menurun kerana waiting time panjang

### 2.3 Hasil Yang Diingini

- Masa dokumentasi berkurangan 50% dengan templates dan voice-to-text
  - Semua nota klinikal dalam format standard (SOAP)
  - Medical history accessible dalam 2-3 clicks
  - Auto-integration dengan Farmasi, Billing, Makmal untuk eliminate redundancy
  - Full audit trail untuk compliance
  - Meningkatkan kualiti rawatan dan kepuasan pesakit
- 

## 3. User Stories

### 3.1 User Stories Utama

- **Sebagai** Doktor, **saya mahu** merekod nota klinikal dengan cepat menggunakan templates **supaya** saya boleh fokus kepada pesakit instead of typing
- **Sebagai** Doktor, **saya mahu** menggunakan voice-to-text untuk nota klinikal **supaya** saya boleh dokumentasi sambil examine pesakit tanpa perlu type
- **Sebagai** Doktor, **saya mahu** search dan pilih diagnosis ICD-10 code dengan mudah **supaya** coding adalah standard dan accurate
- **Sebagai** Doktor, **saya mahu** melihat medical history pesakit dalam chronological timeline **supaya** saya boleh buat keputusan rawatan yang tepat berdasarkan history
- **Sebagai** Doktor, **saya mahu** prescription auto-populate ke modul Farmasi selepas saya rekod diagnosis **supaya** saya tidak perlu tulis prescription berulang kali
- **Sebagai** Doktor, **saya mahu** order lab tests dari EMR dan view results dalam EMR **supaya** semua maklumat klinikal ada dalam satu tempat
- **Sebagai** Doktor, **saya mahu** schedule follow-up appointment terus dari EMR **supaya** pesakit tidak perlu queue di kaunter semula

- **Sebagai** Jururawat, **saya mahu** capture vital signs pesakit sebelum konsultasi **supaya** doktor dapat melihat vitals dalam EMR
- **Sebagai** Jururawat, **saya mahu** view nota klinikal dan medical history untuk triage **supaya** saya boleh prioritize urgent cases
- **Sebagai** Jururawat, **saya mahu** add nursing notes dalam EMR **supaya** doktor dapat lihat observation dan monitoring yang saya buat
- **Sebagai** Pengarah Perubatan, **saya mahu** audit semua perubahan pada EMR dengan version history **supaya** saya boleh ensure compliance dan quality control
- **Sebagai** Pengarah Perubatan, **saya mahu** search dan analyze EMR by diagnosis untuk reporting **supaya** saya boleh monitor disease patterns dan kualiti rawatan

### 3.2 Edge Cases & User Stories Sekunder

- **Sebagai** Doktor, **bila** saya salah rekod diagnosis, **saya sepatutnya** boleh edit dengan system log perubahan untuk audit trail
- **Sebagai** Doktor, **bila** pesakit ada multiple diagnoses, **saya sepatutnya** boleh add beberapa ICD-10 codes dengan primary dan secondary diagnosis
- **Sebagai** Doktor, **bila** pesakit emergency dan saya perlu rekod cepat, **saya sepatutnya** boleh save quick notes dahulu dan complete SOAP notes kemudian
- **Sebagai** Jururawat, **bila** saya add nursing notes, **saya sepatutnya** tidak boleh edit atau delete nota doktor untuk maintain data integrity
- **Sebagai** System, **bila** doktor attach X-ray atau lab results, **saya sepatutnya** scan file untuk malware dan validate file type untuk security

## 4. Keperluan Fungsian

### 4.1 Ciri-ciri Teras

- **FR-01:** Sistem mesti membenarkan doktor merekod nota klinikal dalam format SOAP (Subjective, Objective, Assessment, Plan)
- **FR-02:** Sistem mesti provide templates nota klinikal mengikut jabatan (Pediatric, Dermatologi, General Medicine, etc)
- **FR-03:** Sistem mesti support voice-to-text untuk nota klinikal menggunakan Web Speech API
- **FR-04:** Sistem mesti provide search ICD-10 diagnosis codes dengan autocomplete
- **FR-05:** Sistem mesti allow multiple diagnoses dengan designation (Primary/Secondary)
- **FR-06:** Sistem mesti capture vital signs: BP, Temperature, Pulse, Weight, Height dengan auto-calculate BMI
- **FR-07:** Sistem mesti allow attachment images (JPG, PNG) dan PDF files dengan max 10MB per file
- **FR-08:** Sistem mesti display medical history dalam chronological timeline dengan expand/collapse
- **FR-09:** Sistem mesti auto-populate prescription ke modul Farmasi berdasarkan diagnosis
- **FR-10:** Sistem mesti auto-generate bill items ke modul Billing (consultation fee + procedures)
- **FR-11:** Sistem mesti allow doktor order lab tests dan view results dalam EMR
- **FR-12:** Sistem mesti allow doktor schedule follow-up appointment dari EMR
- **FR-13:** Sistem mesti rekod full audit trail (siapa, bila, apa changes) dengan version history
- **FR-14:** Sistem mesti support search EMR by pesakit, diagnosis, tarikh, doktor
- **FR-15:** Sistem mesti auto-save draft setiap 30 saat untuk prevent data loss

### 4.2 Kebenaran & Kawalan Akses

- **Peranan Diperlukan:**

- Doktor: Full access (create, view, edit, delete own notes)
- Jururawat: Limited access (view all notes, create nursing notes only)
- Pengarah Perubatan: Full access semua notes untuk audit

- **Kebenaran Diperlukan:**

- `emr.view` - View EMR records
- `emr.create` - Create nota klinikal
- `emr.edit` - Edit nota klinikal
- `emr.delete` - Delete nota klinikal (soft delete sahaja)
- `emr.audit` - View audit trail dan version history
- `nursing_notes.create` - Create nursing notes

- **Authorization Logic:**

- Doktor hanya boleh edit nota sendiri (kecuali Pengarah Perubatan)
- Jururawat tidak boleh edit atau delete nota doktor
- Semua edits pada EMR direkod dalam audit trail
- Delete operation adalah soft delete (data tidak hilang, hanya hidden)

#### **4.3 Validasi Data**

- **Field Wajib:**

- Pesakit ID (link to modul Pendaftaran Pesakit)
- Tarikh dan Masa Konsultasi
- Doktor yang handle konsultasi
- Chief Complaint (Subjective)
- Diagnosis (minimal 1 ICD-10 code)
- Plan/Treatment

- **Peraturan Validasi:**

- ICD-10 code mesti valid (dari database ICD-10)
- Vital signs mesti dalam range yang reasonable (contoh: BP 50-300, Temp 30-45°C)
- Attachment file type mesti JPG, PNG, atau PDF sahaja
- Attachment file size maksimum 10MB
- Voice-to-text text length maksimum 5000 characters per field

- **Peraturan Perniagaan:**

- Tidak boleh create EMR untuk pesakit yang tiada dalam sistem (mesti registered dahulu)
- Tidak boleh backdate EMR lebih dari 7 hari (untuk prevent fraud)
- Nota klinikal mesti finalized dalam 24 jam, otherwise flag sebagai "Pending Completion"
- Doktor mesti review dan approve auto-populated prescription sebelum hantar ke Farmasi

#### **4.4 Audit Trail & PDPA Compliance**

- **Adakah feature ini perlu audit trail? Ya**

- **Field Audit:**

- `created_by` - Doktor/Jururawat yang create nota
- `updated_by` - Siapa yang update nota (setiap update)
- `version` - Version number untuk tracking changes
- `change_log` - JSON field untuk rekod apa yang berubah
- `deleted_at` - Soft delete timestamp
- `deleted_by` - Siapa yang delete

- **Data Consent:** Ya, pesakit mesti beri consent untuk:

- Penggunaan data perubatan untuk rawatan
  - Sharing data dengan third parties (insurance, lab)
  - Retention period untuk medical records (minimal 7 tahun)
- **Data Retention:**
    - Medical records disimpan minimal **7 tahun** (mengikut undang-undang Malaysia)
    - Soft delete sahaja, tidak boleh hard delete
    - Audit trail disimpan permanently untuk compliance

---

## 5. Keperluan Teknikal

### 5.1 Teknologi Stack

- **Framework:** Laravel 12
- **Frontend:** Blade Templates + Bootstrap 5 + CoreUI
- **Database:** MySQL 8.0
- **Authentication:** Laravel Breeze/Sanctum
- **File Storage:** Laravel Storage (S3 untuk production, local untuk development)
- **Queue:** Laravel Queue (Redis) untuk async operations (file scanning, notifications)
- **Cache:** Redis untuk cache ICD-10 codes dan templates
- **Voice-to-Text:** Web Speech API (browser native)
- **WYSIWYG Editor:** TinyMCE atau Summernote untuk rich text notes

### 5.2 Arsitektur Aplikasi

Mengikut pattern yang ditetapkan dalam `DEVELOPER_GUIDE.md` :

```
Route Attributes (dalam Controller)
  ↓
Controller (HTTP Layer)
  ↓
FormRequest (Validation Layer)
  ↓
Service Layer (Business Logic)
  ↓
Repository Layer (Data Access)
  ↓
Model (Eloquent ORM)
  ↓
Database
```

### 5.3 Struktur Modul

```
app/
├── Http/
|   ├── Controllers/
|   |   └── Admin/
|   |       ├── EmrController.php (dengan Route Attributes)
|   |       ├── VitalSignsController.php
|   |       └── NursingNotesController.php
|   └── Requests/
|       ├── StoreEmrRequest.php
|       ├── UpdateEmrRequest.php
|       ├── StoreVitalSignsRequest.php
|       └── StoreNursingNotesRequest.php
```

```

├── Services/
|   ├── EmrService.php
|   ├── VitalSignsService.php
|   ├── DiagnosisService.php
|   └── MedicalHistoryService.php
├── Repositories/
|   ├── EmrRepository.php
|   ├── VitalSignsRepository.php
|   ├── DiagnosisRepository.php
|   └── Icd10Repository.php
├── Models/
|   ├── Emr.php
|   ├── VitalSigns.php
|   ├── Diagnosis.php
|   ├── Icd10Code.php
|   ├── EmrAttachment.php
|   ├── NursingNotes.php
|   └── EmrAuditLog.php
├── Traits/
|   └── HasAuditTrail.php
└── Exceptions/
    └── EmrException.php

config/
└── emr.php (configuration file)

resources/
└── views/
    └── admin/
        └── emr/
            ├── index.blade.php
            ├── create.blade.php
            ├── edit.blade.php
            ├── show.blade.php
            ├── history.blade.php
            └── partials/
                ├── soap-form.blade.php
                ├── vital-signs.blade.php
                ├── diagnosis-search.blade.php
                └── medical-timeline.blade.php

```

#### 5.4 Command untuk Generate

```

# Models dengan migrations dan factories
php artisan make:model Emr -mf
php artisan make:model VitalSigns -mf
php artisan make:model Diagnosis -mf
php artisan make:model Icd10Code -m
php artisan make:model EmrAttachment -mf
php artisan make:model NursingNotes -mf
php artisan make:model EmrAuditLog -m

# Controllers
php artisan make:controller Admin/EmrController
php artisan make:controller Admin/VitalSignsController

```

```

php artisan make:controller Admin/NursingNotesController

# FormRequests
php artisan make:request StoreEmrRequest
php artisan make:request UpdateEmrRequest
php artisan make:request StoreVitalSignsRequest
php artisan make:request StoreNursingNotesRequest

# Services (manual create)
# Create files: app/Services/EmrService.php, VitalSignsService.php, DiagnosisService.php,
MedicalHistoryService.php

# Repositories (manual create)
# Create files: app/Repositories/EmrRepository.php, VitalSignsRepository.php, DiagnosisRepository.php,
Icd10Repository.php

# Exception
php artisan make:exception EmrException

# Trait (manual create)
# Create file: app/Traits/HasAuditTrail.php

```

## 5.5 Database Schema

### Jadual Baharu

**Jadual: emr (Electronic Medical Records)**

Column	Type	Description
<code>id</code>	<code>bigint UNSIGNED PK</code>	Primary key
<code>kod_emr</code>	<code>varchar(50) UNIQUE NOT NULL</code>	Auto-generated (EMR-YYYYMMDD-0001)
<code>pesakit_id</code>	<code>bigint UNSIGNED NOT NULL</code>	<code>FK → pesakit.id</code>
<code>temujanji_id</code>	<code>bigint UNSIGNED NULL</code>	<code>FK → temujanji.id (if from appointment)</code>
<code>doktor_id</code>	<code>bigint UNSIGNED NOT NULL</code>	<code>FK → users.id (role=doktor)</code>
<code>tarikh_konsultasi</code>	<code>datetime NOT NULL</code>	Tarikh dan masa konsultasi
<code>chief_complaint</code>	<code>text NULL</code>	Subjective: Aduan utama pesakit
<code>history_present_illness</code>	<code>text NULL</code>	Subjective: History of present illness
<code>past_medical_history</code>	<code>text NULL</code>	Subjective: Past medical/surgical history
<code>medications</code>	<code>text NULL</code>	Subjective: Current medications
<code>allergies</code>	<code>text NULL</code>	Subjective: Allergies
<code>family_history</code>	<code>text NULL</code>	Subjective: Family history
<code>social_history</code>	<code>text NULL</code>	Subjective: Social history (smoking, alcohol, etc)
<code>physical_examination</code>	<code>text NULL</code>	Objective: Physical exam findings
<code>investigations</code>	<code>text NULL</code>	Objective: Investigation results

assessment	text NULL	Assessment: Clinical impression
plan	text NULL	Plan: Treatment plan
notes	text NULL	Additional notes
status	enum('draft','finalized','amended') NOT NULL DEFAULT 'draft'	Status EMR
finalized_at	datetime NULL	Bila EMR di-finalize
template_used	varchar(100) NULL	Template yang digunakan (jika ada)
version	int NOT NULL DEFAULT 1	Version number untuk audit
created_by	bigint UNSIGNED NULL	FK → users.id
updated_by	bigint UNSIGNED NULL	FK → users.id
created_at	timestamp	Waktu rekod dicipta
updated_at	timestamp	Waktu rekod dikemaskini
deleted_at	timestamp NULL	Soft delete
deleted_by	bigint UNSIGNED NULL	FK → users.id

#### Indexes:

- idx\_pesakit\_id on pesakit\_id
- idx\_doktor\_id on doktor\_id
- idx\_tarikh\_konsultasi on tarikh\_konsultasi
- idx\_status on status
- idx\_kod\_emr on kod\_emr

#### Foreign Keys:

- fk\_emr\_pesakit FOREIGN KEY( pesakit\_id ) REFERENCES pesakit ( id ) ON DELETE RESTRICT
- fk\_emr\_doktor FOREIGN KEY( doktor\_id ) REFERENCES users ( id ) ON DELETE RESTRICT

#### Jadual: vital\_signs

Column	Type	Description
id	bigint UNSIGNED PK	Primary key
emr_id	bigint UNSIGNED NOT NULL	FK → emr.id
blood_pressure_systolic	int NULL	Systolic BP (mmHg)
blood_pressure_diastolic	int NULL	Diastolic BP (mmHg)
temperature	decimal(4,1) NULL	Temperature (°C)
pulse_rate	int NULL	Pulse rate (bpm)
respiratory_rate	int NULL	Respiratory rate (breaths/min)
oxygen_saturation	int NULL	SpO2 (%)
weight	decimal(5,2) NULL	Weight (kg)

height	decimal(5,2) NULL	Height (cm)
bmi	decimal(4,2) NULL	BMI (auto-calculated)
bmi_category	varchar(50) NULL	Underweight/Normal/Overweight/Obese
measured_by	bigint UNSIGNED NULL	FK → users.id (jururawat)
measured_at	datetime NOT NULL	Waktu measurement
created_at	timestamp	Waktu rekod dicipta
updated_at	timestamp	Waktu rekod dikemaskini

**Indexes:**

- idx\_emr\_id on emr\_id

**Foreign Keys:**

- fk\_vitals\_emr FOREIGN KEY( emr\_id ) REFERENCES emr ( id ) ON DELETE CASCADE

**Jadual: diagnoses**

Column	Type	Description
id	bigint UNSIGNED PK	Primary key
emr_id	bigint UNSIGNED NOT NULL	FK → emr.id
icd10_code_id	bigint UNSIGNED NOT NULL	FK → icd10_codes.id
diagnosis_type	enum('primary','secondary') NOT NULL DEFAULT 'primary'	Primary atau secondary diagnosis
notes	text NULL	Notes untuk diagnosis ini
sequence	int NOT NULL DEFAULT 1	Urutan diagnosis
created_at	timestamp	Waktu rekod dicipta
updated_at	timestamp	Waktu rekod dikemaskini

**Indexes:**

- idx\_emr\_id on emr\_id
- idx\_icd10\_code\_id on icd10\_code\_id

**Foreign Keys:**

- fk\_diagnosis\_emr FOREIGN KEY( emr\_id ) REFERENCES emr ( id ) ON DELETE CASCADE
- fk\_diagnosis\_icd10 FOREIGN KEY( icd10\_code\_id ) REFERENCES icd10\_codes ( id ) ON DELETE RESTRICT

**Jadual: icd10\_codes**

Column	Type	Description
id	bigint UNSIGNED PK	Primary key
code	varchar(10) UNIQUE NOT NULL	ICD-10 code (contoh: A00.0)
description_en	text NOT NULL	Description in English

description_ms	text NULL	Description in Malay
category	varchar(50) NULL	Category (Infectious diseases, Neoplasms, etc)
is_active	boolean DEFAULT TRUE	Active or deprecated
created_at	timestamp	Waktu rekod dicipta
updated_at	timestamp	Waktu rekod dikemaskini

**Indexes:**

- idx\_code on code
- idx\_category on category
- idx\_is\_active on is\_active
- FULLTEXT index on description\_en , description\_ms for search

**Jadual: emr\_attachments**

Column	Type	Description
id	bigint UNSIGNED PK	Primary key
emr_id	bigint UNSIGNED NOT NULL	FK → emr.id
file_name	varchar(255) NOT NULL	Original filename
file_path	varchar(500) NOT NULL	Storage path
file_type	varchar(50) NOT NULL	MIME type (image/jpeg, application/pdf)
file_size	bigint NOT NULL	File size in bytes
description	varchar(255) NULL	Description (X-ray, Lab result, etc)
uploaded_by	bigint UNSIGNED NULL	FK → users.id
uploaded_at	datetime NOT NULL	Upload timestamp
created_at	timestamp	Waktu rekod dicipta
updated_at	timestamp	Waktu rekod dikemaskini

**Indexes:**

- idx\_emr\_id on emr\_id

**Foreign Keys:**

- fk\_attachment\_emr FOREIGN KEY ( emr\_id ) REFERENCES emr ( id ) ON DELETE CASCADE

**Jadual: nursing\_notes**

Column	Type	Description
id	bigint UNSIGNED PK	Primary key
emr_id	bigint UNSIGNED NOT NULL	FK → emr.id
jururawat_id	bigint UNSIGNED NOT NULL	FK → users.id (role=jururawat)
note_type	enum('assessment','intervention','evaluation','general') NOT NULL	Jenis nursing note

notes	text NOT NULL	Nursing notes content
recorded_at	datetime NOT NULL	Waktu observation/intervention
created_at	timestamp	Waktu rekod dicipta
updated_at	timestamp	Waktu rekod dikemaskini

**Indexes:**

- `idx_emr_id` on `emr_id`
- `idx_jururawat_id` on `jururawat_id`

**Foreign Keys:**

- `fk_nursing_emr` FOREIGN KEY( `emr_id` ) REFERENCES `emr` ( `id` ) ON DELETE CASCADE

**Jadual: `emr_audit_log`**

Column	Type	Description
<code>id</code>	bigint UNSIGNED PK	Primary key
<code>emr_id</code>	bigint UNSIGNED NOT NULL	FK → emr.id
<code>version</code>	int NOT NULL	Version number
<code>action</code>	enum('created','updated','deleted','finalized','amended') NOT NULL	Action performed
<code>changes</code>	json NULL	JSON field rekod changes (old vs new values)
<code>changed_by</code>	bigint UNSIGNED NOT NULL	FK → users.id
<code>changed_at</code>	datetime NOT NULL	Timestamp of change
<code>ip_address</code>	varchar(45) NULL	IP address of user
<code>user_agent</code>	text NULL	Browser/device info

**Indexes:**

- `idx_emr_id` on `emr_id`
- `idx_changed_by` on `changed_by`
- `idx_changed_at` on `changed_at`

**Foreign Keys:**

- `fk_audit_emr` FOREIGN KEY( `emr_id` ) REFERENCES `emr` ( `id` ) ON DELETE CASCADE

## 5.6 Model Eloquent

**Model: Emr**

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
```

```
use Illuminate\Database\Eloquent\SoftDeletes;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use App\Traits\HasAuditTrail;

class Emr extends Model
{
    use HasFactory, SoftDeletes, HasAuditTrail;

    protected $table = 'emr';

    protected $fillable = [
        'kod_emr', 'pesakit_id', 'temujanji_id', 'doktor_id', 'tarikh_konsultasi',
        'chief_complaint', 'history_present_illness', 'past_medical_history',
        'medications', 'allergies', 'family_history', 'social_history',
        'physical_examination', 'investigations', 'assessment', 'plan', 'notes',
        'status', 'finalized_at', 'template_used', 'version',
        'created_by', 'updated_by', 'deleted_by'
    ];

    protected $casts = [
        'tarikh_konsultasi' => 'datetime',
        'finalized_at' => 'datetime',
        'version' => 'integer',
    ];

    // Relationships
    public function pesakit()
    {
        return $this->belongsTo(Pesakit::class, 'pesakit_id');
    }

    public function doktor()
    {
        return $this->belongsTo(User::class, 'doktor_id');
    }

    public function temujanji()
    {
        return $this->belongsTo(TemuJanji::class, 'temujanji_id');
    }

    public function vitalSigns()
    {
        return $this->hasOne(VitalSigns::class, 'emr_id');
    }

    public function diagnoses()
    {
        return $this->hasMany(Diagnosis::class, 'emr_id')->orderBy('sequence');
    }

    public function attachments()
    {
        return $this->hasMany(EmrAttachment::class, 'emr_id');
    }
}
```

```
public function nursingNotes()
{
    return $this->hasMany(NursingNotes::class, 'emr_id')->orderBy('recorded_at', 'desc');
}

public function auditLogs()
{
    return $this->hasMany(EmrAuditLog::class, 'emr_id')->orderBy('changed_at', 'desc');
}

public function creator()
{
    return $this->belongsTo(User::class, 'created_by');
}

public function updater()
{
    return $this->belongsTo(User::class, 'updated_by');
}

// Scopes
public function scopeFinalized($query)
{
    return $query->where('status', 'finalized');
}

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

public function scopeByDoktor($query, $doktorId)
{
    return $query->where('doktor_id', $doktorId);
}

public function scopeByPesakit($query, $pesakitId)
{
    return $query->where('pesakit_id', $pesakitId);
}

// Accessors
public function getStatusBadgeAttribute()
{
    return config("emr.status_badges.{$this->status}", $this->status);
}

public function getPrimaryDiagnosisAttribute()
{
    return $this->diagnoses()->where('diagnosis_type', 'primary')->first();
}

// Methods
public function finalize()
{
    $this->update([

```

```

        'status' => 'finalized',
        'finalized_at' => now()
    ]);
}

public function canEdit(User $user)
{
    // Doktor boleh edit nota sendiri yang draft
    // Pengarah Perubatan boleh edit semua
    if ($user->hasRole('Pengarah Perubatan')) {
        return true;
    }

    return $this->doktor_id === $user->id && $this->status === 'draft';
}
}

```

#### Relationships:

- `belongsTo()` - Pesakit, User (doktor), Temujanji
- `hasOne()` - VitalSigns
- `hasMany()` - Diagnosis, EmrAttachment, NursingNotes, EmrAuditLog

**Factory**: Ya **Seeder**: Ya (untuk demo/testing)

---

## 5.7 Configuration File

File: `config/emr.php`

```

<?php

return [
    // Kod generation
    'kod_prefix' => 'EMR',
    'kod_format' => 'EMR-YYYYMMDD-9999',

    // Status
    'statuses' => ['draft', 'finalized', 'amended'],

    'status_labels' => [
        'draft' => 'Draft',
        'finalized' => 'Finalized',
        'amended' => 'Amended',
    ],
    'status_badges' => [
        'draft' => '<span class="badge badge-warning">Draft</span>',
        'finalized' => '<span class="badge badge-success">Finalized</span>',
        'amended' => '<span class="badge badge-info">Amended</span>',
    ],
    // Templates
    'templates' => [
        'general' => 'General Medicine',
        'pediatrik' => 'Pediatrik',
        'dermatologi' => 'Dermatologi',
    ]
];

```

```

'obstetrik' => 'Obstetrik & Ginekologi',
'orthopedik' => 'Orthopedik',
'ent' => 'ENT (Ear, Nose, Throat)',
],

// Vital signs ranges (for validation alerts)
'vertical_ranges' => [
  'bp_systolic' => ['min' => 50, 'max' => 300, 'normal' => [90, 120]],
  'bp_diastolic' => ['min' => 30, 'max' => 200, 'normal' => [60, 80]],
  'temperature' => ['min' => 30, 'max' => 45, 'normal' => [36.5, 37.5]],
  'pulse_rate' => ['min' => 30, 'max' => 200, 'normal' => [60, 100]],
  'respiratory_rate' => ['min' => 5, 'max' => 60, 'normal' => [12, 20]],
  'oxygen_saturation' => ['min' => 50, 'max' => 100, 'normal' => [95, 100]],
],
// BMI categories
'vertical_categories' => [
  ['max' => 18.5, 'label' => 'Underweight'],
  ['min' => 18.5, 'max' => 24.9, 'label' => 'Normal'],
  ['min' => 25, 'max' => 29.9, 'label' => 'Overweight'],
  ['min' => 30, 'label' => 'Obese'],
],
// File upload settings
'vertical_attachments' => [
  'allowed_types' => ['image/jpeg', 'image/png', 'application/pdf'],
  'max_size' => 10 * 1024 * 1024, // 10MB in bytes
  'storage_path' => 'emr-attachments',
],
// Auto-save settings
'vertical_auto_save_interval' => 30, // seconds

// Business rules
'vertical_backdate_limit_days' => 7, // Cannot create EMR older than 7 days
'vertical_finalization_deadline_hours' => 24, // Must finalize within 24 hours
'vertical_records_per_page' => 15,
// Integration settings
'vertical_auto_populate_prescription' => true,
'vertical_auto_generate_billing' => true,
'vertical_enable_lab_integration' => true,
];

```

## 5.8 Routes (Route Attributes)

File: `app\Http\Controllers\Admin\EmrController.php`

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\Emr;
use App\Http\Requests\StoreEmrRequest;

```

```

use App\Http\Requests\UpdateEmrRequest;
use App\Services\EmrService;
use App\Traits\HandlesApiResponses;
use Illuminate\Support\Facades\Log;
use Spatie\RouteAttributes\Attributes\Get;
use Spatie\RouteAttributes\Attributes\Post;
use Spatie\RouteAttributes\Attributes\Patch;
use Spatie\RouteAttributes\Attributes\Delete;
use Spatie\RouteAttributes\Attributes\Prefix;
use Spatie\RouteAttributes\Attributes\Middleware;

#[Prefix('admin/emr')]
#[Middleware(['web', 'auth'])]
class EmrController extends Controller
{
    use HandlesApiResponses;

    protected EmrService $emrService;

    public function __construct(EmrService $emrService)
    {
        $this->emrService = $emrService;
    }

    #[Get('/', name: 'admin.emr.index')]
    public function index()
    {
        $emrs = $this->emrService->getAllEmr();
        $stats = $this->emrService->getStats();
        return view('admin.emr.index', compact('emrs', 'stats'));
    }

    #[Get('/create', name: 'admin.emr.create')]
    public function create()
    {
        $templates = config('emr.templates');
        return view('admin.emr.create', compact('templates'));
    }

    #[Post('/', name: 'admin.emr.store')]
    public function store(StoreEmrRequest $request)
    {
        try {
            $emr = $this->emrService->create($request->validated());
            return $this->successRedirect('admin.emr.show', 'EMR berjaya dicipta', ['emr' => $emr->id]);
        } catch (\Exception $e) {
            Log::error('EMR creation failed', ['error' => $e->getMessage()]);
            return $this->errorRedirect('Gagal mencipta EMR');
        }
    }

    #[Get('/{emr}', name: 'admin.emr.show')]
    public function show(Emr $emr)
    {
        $emr->load(['pesakit', 'doktor', 'vitalSigns', 'diagnoses.icd10Code', 'attachments',

```

```

'nursingNotes', 'auditLogs']);
    return view('admin.emr.show', compact('emr'));
}

#[Get('/{emr}/edit', name: 'admin.emr.edit')]
public function edit(Emr $emr)
{
    // Check authorization
    if (!$emr->canEdit(auth()->user())) {
        return $this->errorRedirect('Anda tidak mempunyai kebenaran untuk edit EMR ini');
    }

    $templates = config('emr.templates');
    return view('admin.emr.edit', compact('emr', 'templates'));
}

#[Patch('/{emr}', name: 'admin.emr.update')]
public function update(UpdateEmrRequest $request, Emr $emr)
{
    try {
        $this->emrService->update($emr->id, $request->validated());
        return $this->successRedirect('admin.emr.show', 'EMR berjaya dikemaskini', ['emr' => $emr->id]);
    } catch (\Exception $e) {
        Log::error('EMR update failed', ['id' => $emr->id, 'error' => $e->getMessage()]);
        return $this->errorRedirect('Gagal mengemaskini EMR');
    }
}

#[Delete('/{emr}', name: 'admin.emr.destroy')]
public function destroy(Emr $emr)
{
    try {
        $this->emrService->delete($emr->id);
        return $this->successRedirect('admin.emr.index', 'EMR berjaya dihapus');
    } catch (\Exception $e) {
        Log::error('EMR deletion failed', ['id' => $emr->id, 'error' => $e->getMessage()]);
        return $this->errorRedirect('Gagal menghapus EMR');
    }
}

#[Post('/{emr}/finalize', name: 'admin.emr.finalize')]
public function finalize(Emr $emr)
{
    try {
        $this->emrService->finalize($emr->id);
        return $this->successRedirect('admin.emr.show', 'EMR berjaya di-finalize', ['emr' => $emr->id]);
    } catch (\Exception $e) {
        Log::error('EMR finalization failed', ['id' => $emr->id, 'error' => $e->getMessage()]);
        return $this->errorRedirect('Gagal finalize EMR');
    }
}

#[Get('/pesakit/{pesakit}/history', name: 'admin.emr.history')]
public function history($pesakitId)

```

```

{
    $history = $this->emrService->getMedicalHistory($pesakitId);
    return view('admin.emr.history', compact('history', 'pesakitId'));
}

#[Post('/{emr}/upload-attachment', name: 'admin.emr.upload')]
public function uploadAttachment(Request $request, Emr $emr)
{
    try {
        $attachment = $this->emrService->uploadAttachment($emr->id, $request->file('file'),
$request->description);
        return $this->successResponse('File berjaya diupload', $attachment);
    } catch (\Exception $e) {
        Log::error('EMR attachment upload failed', ['emr_id' => $emr->id, 'error' => $e-
>getMessage()]);
        return $this->errorResponse('Gagal upload file');
    }
}
}

```

**Route Middleware:** ['web', 'auth']

## 5.9 FormRequest Validation

File: app/Http/Requests/StoreEmrRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class StoreEmrRequest extends FormRequest
{
    public function authorize(): bool
    {
        return auth()->user()->hasPermissionTo('emr.create');
    }

    public function rules(): array
    {
        return [
            'pesakit_id' => 'required|exists:pesakit,id',
            'temujanji_id' => 'nullable|exists:temujanji,id',
            'tarikh_konsultasi' => 'required|date|after_or_equal:' . now()-
>subDays(config('emr.backdate_limit_days'))->format('Y-m-d'),
            'template_used' => 'nullable|in:' . implode(',', array_keys(config('emr.templates'))),

            // SOAP fields
            'chief_complaint' => 'required|string|max:5000',
            'history_present_illness' => 'nullable|string|max:5000',
            'past_medical_history' => 'nullable|string|max:5000',
            'medications' => 'nullable|string|max:2000',
            'allergies' => 'nullable|string|max:2000',
            'family_history' => 'nullable|string|max:2000',
            'social_history' => 'nullable|string|max:2000',
        ];
    }
}

```

```

    'physical_examination' => 'nullable|string|max:5000',
    'investigations' => 'nullable|string|max:5000',
    'assessment' => 'nullable|string|max:5000',
    'plan' => 'required|string|max:5000',
    'notes' => 'nullable|string|max:5000',

    // Vital signs
    'vital_signs.bp_systolic' => 'nullable|integer|min:50|max:300',
    'vital_signs.bp_diastolic' => 'nullable|integer|min:30|max:200',
    'vital_signs.temperature' => 'nullable|numeric|min:30|max:45',
    'vital_signs.pulse_rate' => 'nullable|integer|min:30|max:200',
    'vital_signs.respiratory_rate' => 'nullable|integer|min:5|max:60',
    'vital_signs.oxygen_saturation' => 'nullable|integer|min:50|max:100',
    'vital_signs.weight' => 'nullable|numeric|min:0.5|max:300',
    'vital_signs.height' => 'nullable|numeric|min:10|max:250',

    // Diagnoses
    'diagnoses' => 'required|array|min:1',
    'diagnoses.*.icd10_code_id' => 'required|exists:icd10_codes,id',
    'diagnoses.*.diagnosis_type' => 'required|in:primary,secondary',
    'diagnoses.*.notes' => 'nullable|string|max:1000',
};

}

public function messages(): array
{
    return [
        'chief_complaint.required' => 'Chief Complaint (Aduan Utama) adalah wajib',
        'plan.required' => 'Plan/Treatment adalah wajib',
        'diagnoses.required' => 'Minimal 1 diagnosis diperlukan',
        'diagnoses.*.icd10_code_id.exists' => 'ICD-10 code tidak valid',
        'tarikh_konsultasi.after_or_equal' => 'Tarikh konsultasi tidak boleh lebih dari ' .
config('emr.backdate_limit_days') . ' hari yang lalu',
    ];
}
}

```

## 6. Workflow dan User Flow

### 6.1 Workflow Utama: Konsultasi Pesakit

```

[Pesakit] → Check-in di Kaunter
    ↓
[Kerani] Buat/Update Temujanji
    ↓
[Jururawat] Capture Vital Signs
    ↓
[Sistem] Auto-create EMR draft dengan vital signs
    ↓
[Doktor] Buka EMR pesakit
[Doktor] Review Medical History (past EMRs)
    ↓
[Doktor] Start Consultation
[Doktor] Pilih Template (optional)
[Doktor] Rekod SOAP Notes (boleh guna voice-to-text)

```

```
↓  
- Subjective: Chief complaint, HPI, PMH, medications, allergies, etc  
- Objective: Physical exam, vitals (auto-populated)  
- Assessment: Clinical impression  
- Plan: Treatment plan  
↓  
[Doktor] Search & Add ICD-10 Diagnosis  
[Doktor] Mark Primary vs Secondary Diagnosis  
↓  
[Doktor] Attach Files (jika ada X-ray, lab results)  
↓  
[Doktor] Review & Finalize EMR  
↓  
[Sistem] Auto-populate Prescription ke Farmasi (based on diagnosis)  
[Sistem] Auto-generate Bill Items ke Billing (consultation + procedures)  
↓  
[Doktor] (Optional) Order Lab Tests → Link ke Makmal modul  
[Doktor] (Optional) Schedule Follow-up → Link ke Temujanji modul  
↓  
[Sistem] Save EMR dengan Audit Log  
[Sistem] Update Medical History Timeline
```

## 6.2 Workflow Voice-to-Text

```
[Doktor] → Click microphone icon pada field (contoh: Chief Complaint)  
↓  
[Browser] Request microphone permission (jika first time)  
↓  
[Doktor] Mulakan bercakap  
↓  
[Web Speech API] Convert speech to text real-time  
↓  
[Sistem] Display text dalam field (boleh edit manual)  
↓  
[Doktor] Click Stop atau Done  
↓  
[Sistem] Save text ke field
```

## 6.3 Workflow Audit Trail

```
[Doktor] → Edit EMR yang sudah finalized  
↓  
[Sistem] Create new version (version++)  
[Sistem] Log changes ke emr_audit_log:  
- emr_id  
- version  
- action: 'updated' atau 'amended'  
- changes: JSON with old vs new values  
- changed_by: user_id  
- changed_at: timestamp  
- ip_address  
- user_agent  
↓  
[Sistem] Update EMR status ke 'amended' (jika dari finalized)
```

↓  
[Pengarah Perubatan] Boleh view audit log & version history

## 6.4 State Flow

### EMR Status Flow:

[Draft] → [Finalized] → [Amended]

↓            ↓            ↓  
[Deleted]    [Deleted]    [Deleted]

- **Draft:** EMR baru dicipta, boleh edit freely
- **Finalized:** EMR telah siap, doktor confirm complete. Perlu audit trail untuk edit.
- **Amended:** EMR yang finalized telah di-edit (dengan audit trail)
- **Deleted:** Soft delete sahaja, data masih wujud dalam database

## 7. Keperluan UI/UX

### 7.1 Layout

- **Jenis Halaman:** Full page dengan sidebar navigation
- **Navigation:**
  - Main menu: "EMR / Konsultasi"
  - Submenu:
    - "Senarai EMR"
    - "Create New EMR"
    - "Medical History Search"

### 7.2 Bootstrap 5 + CoreUI Components

Senaraikan component yang digunakan:

- **Card** - Container untuk SOAP sections, Vital Signs, Diagnosis
- **Table** - Senarai EMR, Medical History timeline
- **Form** - SOAP notes input, Vital signs input, Diagnosis search
- **Modal** - Upload attachment, Add diagnosis, Voice-to-text controls
- **Badge** - Status EMR (Draft/Finalized/Amended), Diagnosis type (Primary/Secondary)
- **Button** - Save Draft, Finalize, Edit, Delete, Upload, Voice-to-text toggle
- **Accordion** - Medical History timeline (expand/collapse old EMRs)
- **Alert** - Validation errors, Success messages, Vital signs out of range warnings
- **Tabs** - SOAP sections (Subjective, Objective, Assessment, Plan)
- **Dropdown** - Template selection, ICD-10 category filter

### 7.3 Icons

- **Heroicons:**
  - `heroicon-o-document-text` - EMR icon
  - `heroicon-o-microphone` - Voice-to-text
  - `heroicon-o-paper-clip` - Attachments
  - `heroicon-o-clipboard-document-list` - Medical history
  - `heroicon-o-heart` - Vital signs
  - `heroicon-o-beaker` - Lab tests
  - `heroicon-o-calendar` - Follow-up appointment
  - `heroicon-o-magnifying-glass` - Search diagnosis

## 7.4 Responsive Design

- **Mobile Support:** Ya (untuk tablet usage by doctors)
- **Tablet Support:** Ya (primary device for doctors)
- **Breakpoints:** Standard Bootstrap breakpoints
  - Tablet (768px+): 2-column layout for SOAP sections
  - Mobile (<768px): Single column, stacked layout

## 7.5 Voice-to-Text UI

- Microphone icon button pada setiap textarea field
- Visual indicator bila recording (pulsing red dot)
- Real-time text display semasa speech recognition
- "Stop" button untuk hentikan recording
- Fallback ke manual typing jika browser tidak support Web Speech API

---

# 8. Keperluan Keselamatan

## 8.1 Authentication & Authorization

- **Authentication:** Laravel Breeze/Sanctum
- **Middleware:** auth untuk semua admin routes
- **Role-based Access:**
  - Doktor: Full access untuk EMR sendiri
  - Jururawat: View all EMR + Create nursing notes
  - Pengarah Perubatan: Full access semua EMR + Audit logs

## 8.2 Data Protection (PDPA Compliance)

- **Audit Trail:** Full audit trail untuk semua operasi (create, update, delete, finalize, amend)
- **Soft Delete:** Semua delete operation adalah soft delete
- **Consent:** Rekod consent pesakit untuk:
  - Penggunaan data perubatan
  - Sharing dengan third parties (insurance, lab)
  - Retention 7 tahun minimum
- **Data Encryption:**
  - Sensitive fields (allergies, medications) boleh encrypt at rest (optional)
  - HTTPS untuk data in transit
- **Access Logs:** Rekod siapa access EMR pesakit bila

## 8.3 Input Validation & Security

- **CSRF Protection:** Semua POST/PATCH/DELETE dilindungi CSRF token
- **SQL Injection Prevention:** Guna Eloquent ORM
- **XSS Prevention:** Guna Blade {{ }} escaping. Rich text editor (TinyMCE) sanitize HTML.
- **File Upload Security:**
  - Validate file type (whitelist: JPG, PNG, PDF sahaja)
  - Validate file size (max 10MB)
  - Scan uploaded files untuk malware (ClamAV integration atau VirusTotal API)
  - Store files outside web root, serve via controller dengan authentication check
  - Rename files untuk prevent directory traversal attacks

## 8.4 Voice-to-Text Security

- **Browser API:** Guna Web Speech API (client-side, tiada data send to external server)
- **Fallback:** Jika browser tidak support, disable microphone button gracefully
- **No Recording Storage:** Audio tidak disimpan, hanya text results

## 9. Keperluan Prestasi

### 9.1 Response Time

- **Halaman Senarai EMR:** < 2 saat
- **Halaman Create/Edit EMR:** < 1 saat
- **Submit EMR:** < 3 saat
- **Search ICD-10:** < 500ms (dengan autocomplete)
- **Medical History Load:** < 2 saat (with lazy loading untuk old records)
- **File Upload:** Depend on file size, show progress bar

### 9.2 Scalability

- **Database Indexing:** Index pada pesakit\_id, doktor\_id, tarikh\_konsultasi, status, ICD-10 code
- **Query Optimization:**
  - Eager loading untuk relationships (`with('pesakit', 'doktor', 'diagnoses.icd10Code')`)
  - Lazy loading untuk medical history (load on demand)
- **Caching:**
  - Cache ICD-10 codes list (update daily)
  - Cache templates (update on change)
  - Cache user permissions (clear on role change)
- **Pagination:** Limit 15 records per page untuk EMR list
- **File Storage:**
  - Guna S3 atau similar object storage untuk scalability
  - CDN untuk serve static files (attachments)

### 9.3 Concurrent Users

- **Expected Users:**
    - 20-30 doktor concurrent
    - 10-15 jururawat concurrent
    - Total ~50 concurrent users during peak hours
  - **Database Connections:** Connection pooling untuk handle concurrent requests
  - **Queue Jobs:** Queue file scanning dan notifications untuk offload main thread
- 

## 10. Keperluan Ujian

### 10.1 Unit Testing

Create tests in `tests/Unit/Emr/` :

- **Test:** EmrService - create EMR with valid data
- **Test:** EmrService - cannot create EMR with backdate > 7 days
- **Test:** EmrService - finalize EMR updates status and timestamp
- **Test:** VitalSignsService - BMI calculation is correct
- **Test:** VitalSignsService - BMI category assignment is correct
- **Test:** DiagnosisService - primary diagnosis is unique per EMR
- **Test:** EmrAuditService - audit log records all changes
- **Test:** EmrAuditService - version increment on update
- **Test:** Emr model - canEdit() returns correct permissions

### 10.2 Feature Testing

Create tests in `tests/Feature/EmrTest.php` :

- **Test:** Authenticated doktor can view EMR index page
- **Test:** Authenticated doktor can create EMR
- **Test:** Authenticated doktor can edit own draft EMR
- **Test:** Authenticated doktor cannot edit other doktor's EMR
- **Test:** Authenticated jururawat can view EMR but cannot edit
- **Test:** Authenticated jururawat can create nursing notes
- **Test:** Unauthenticated user cannot access EMR pages
- **Test:** Validation rules are enforced on EMR submit
- **Test:** Cannot create EMR without diagnosis
- **Test:** Cannot create EMR with invalid ICD-10 code
- **Test:** File upload validates file type and size
- **Test:** Finalize EMR changes status to 'finalized'
- **Test:** Edit finalized EMR creates audit log entry
- **Test:** Medical history displays chronologically

```

public function test_authenticated_doktor_can_view_emr_index()
{
    $user = User::factory()->create();
    $user->assignRole('Doktor');

    $response = $this->actingAs($user)
        ->get(route('admin.emr.index'));

    $response->assertStatus(200);
    $response->assertViewIs('admin.emr.index');
}

public function test_doktor_can_create_emr_with_diagnosis()
{
    $user = User::factory()->create();
    $user->assignRole('Doktor');
    $pesakit = Pesakit::factory()->create();
    $icd10 = Icd10Code::factory()->create();

    $data = [
        'pesakit_id' => $pesakit->id,
        'tarikh_konsultasi' => now()->format('Y-m-d H:i:s'),
        'chief_complaint' => 'Fever and cough',
        'plan' => 'Prescribe paracetamol and monitor',
        'diagnoses' => [
            ['icd10_code_id' => $icd10->id, 'diagnosis_type' => 'primary']
        ]
    ];

    $response = $this->actingAs($user)
        ->post(route('admin.emr.store'), $data);

    $response->assertRedirect();
    $this->assertDatabaseHas('emr', [
        'pesakit_id' => $pesakit->id,
        'chief_complaint' => 'Fever and cough'
    ]);
}

```

```
]);
}
```

## 10.3 Integration Testing

- **Test:** Integration dengan Farmasi - prescription auto-populated
- **Test:** Integration dengan Billing - bill items auto-generated
- **Test:** Integration dengan Makmal - lab orders created
- **Test:** Integration dengan Temujanji - follow-up scheduled
- **Test:** File upload to S3 successful
- **Test:** File scan for malware works (if integrated)

## 10.4 User Acceptance Testing (UAT)

**Scenario 1:** Doktor create EMR untuk pesakit baru dengan voice-to-text

- Doktor login
- Search pesakit
- Click "Create EMR"
- Jururawat sudah capture vital signs (auto-populated)
- Doktor pilih template "General Medicine"
- Doktor guna voice-to-text untuk chief complaint
- Doktor complete SOAP notes
- Doktor search dan add ICD-10 diagnosis
- Doktor finalize EMR
- Prescription auto-populate ke Farmasi
- Bill auto-generate ke Billing
- Expected Result: EMR created, status=finalized, audit log created, integrations triggered

**Scenario 2:** Jururawat view medical history dan add nursing notes

- Jururawat login
- Search pesakit
- View medical history timeline
- Expand previous EMR untuk review
- Add nursing notes untuk current visit
- Expected Result: Nursing notes saved, linked to current EMR

**Scenario 3:** Pengarah Perubatan audit EMR changes

- Pengarah Perubatan login
- Search EMR yang di-amend
- View audit log
- Compare version 1 vs version 2 (what changed)
- Expected Result: Audit log shows all changes with timestamps and user

---

# 11. Langkah Implementasi

## 11.1 Fasa 1: Setup & Database (Minggu 1)

- Create migrations untuk semua jadual (emr, vital\_signs, diagnoses, icd10\_codes, emr\_attachments, nursing\_notes, emr\_audit\_log)
- Seed ICD-10 codes database (import from standard ICD-10 dataset)
- Create Models dengan relationships
- Create configuration file config/emr.php
- Run migrations dan seed sample data

## 11.2 Fasa 2: Repository & Service Layer (Minggu 1-2)

- Create EmrRepository dengan methods: `create()`, `update()`, `delete()`, `search()`, `getMedicalHistory()`
- Create VitalSignsRepository
- Create DiagnosisRepository
- Create Icd10Repository dengan search functionality
- Create EmrService dengan business logic
- Create VitalSignsService dengan BMI calculation
- Create DiagnosisService
- Create MedicalHistoryService
- Create EmrAuditService untuk audit trail
- Create EmrException untuk custom errors

## 11.3 Fasa 3: FormRequest Validation (Minggu 2)

- Create `StoreEmrRequest` dengan comprehensive validation
- Create `UpdateEmrRequest`
- Create `StoreVitalSignsRequest` dengan range validation
- Create `StoreNursingNotesRequest`
- Test validation rules

## 11.4 Fasa 4: Controller & Routes (Minggu 2-3)

- Create EmrController dengan Route Attributes
- Implement CRUD methods
- Implement `finalize()` method
- Implement `history()` method untuk medical history
- Implement `uploadAttachment()` method
- Create VitalSignsController
- Create NursingNotesController
- Add error handling dengan `HandlesApiResponses` trait
- Clear route cache: `php artisan route:clear`

## 11.5 Fasa 5: Views & UI (Minggu 3-4)

- Create Blade template `index.blade.php` - Senarai EMR dengan search/filter
- Create Blade template `create.blade.php` - Form create EMR dengan SOAP sections
- Create Blade template `edit.blade.php` - Form edit EMR
- Create Blade template `show.blade.php` - View EMR detail dengan print option
- Create Blade template `history.blade.php` - Medical history timeline
- Create partial `soap-form.blade.php` - SOAP notes form dengan tabs
- Create partial `vital-signs.blade.php` - Vital signs input form dengan auto-BMI
- Create partial `diagnosis-search.blade.php` - ICD-10 search autocomplete
- Create partial `medical-timeline.blade.php` - Timeline accordion
- Integrate TinyMCE/Summernote untuk rich text editor
- Integrate Bootstrap 5 + CoreUI components
- Add responsive design
- Test UI di browser (desktop + tablet)

## 11.6 Fasa 6: Voice-to-Text Integration (Minggu 4)

- Implement Web Speech API JavaScript

- Add microphone icon buttons to textarea fields
- Add visual recording indicator
- Add real-time text display
- Add error handling untuk unsupported browsers
- Test voice-to-text across different browsers
- Add user guide tooltip untuk voice-to-text usage

### **11.7 Fasa 7: File Upload & Storage (Minggu 4-5)**

- Implement file upload functionality
- Setup Laravel Storage (S3 atau local)
- Add file type and size validation
- (Optional) Integrate malware scanning (ClamAV atau VirusTotal API)
- Implement secure file serving via controller
- Add file preview untuk images dan PDF
- Add file delete functionality
- Test file upload and download

### **11.8 Fasa 8: Integration dengan Modul Lain (Minggu 5-6)**

- Integration dengan Farmasi:
  - Auto-populate prescription berdasarkan diagnosis
  - Pass EMR data ke Farmasi modul
  - Test integration workflow
- Integration dengan Billing:
  - Auto-generate bill items (consultation fee + procedures)
  - Pass EMR data ke Billing modul
  - Test integration workflow
- Integration dengan Makmal:
  - Create lab order dari EMR
  - Display lab results dalam EMR
  - Test integration workflow
- Integration dengan Temujanji:
  - Schedule follow-up appointment
  - Link EMR to appointment
  - Test integration workflow

### **11.9 Fasa 9: Audit Trail & Version History (Minggu 6)**

- Implement HasAuditTrail trait
- Create EmrObserver untuk auto-log changes
- Implement version history tracking
- Create audit log viewer UI
- Add version comparison UI (diff view)
- Test audit trail functionality
- Test version history

### **11.10 Fasa 10: Testing (Minggu 6-7)**

- Write unit tests untuk Services dan Repositories
- Write feature tests untuk Controllers
- Write integration tests untuk modul integrations
- Perform manual UAT dengan doktor dan jururawat

- Fix bugs yang dijumpai semasa testing
- Performance testing (load testing untuk concurrent users)

## 11.11 Fasa 11: Deployment & Training (Minggu 7-8)

- Deploy ke production server
  - Setup cron jobs jika ada scheduled tasks
  - Configure S3 storage credentials
  - Training untuk Doktor:
    - Cara guna SOAP format
    - Voice-to-text feature
    - Search ICD-10 codes
    - Upload attachments
    - Medical history review
  - Training untuk Jururawat:
    - Capture vital signs
    - Add nursing notes
    - View EMR
  - Training untuk Pengarah Perubatan:
    - Audit trail review
    - Reporting dan analytics
  - Monitor error logs dan user feedback
  - Create user documentation (PDF guide)
- 

# 12. Kriteria Kejayaan

## 12.1 Metrics Utama

- **Masa Dokumentasi:** Berkurangan 50% dari manual (dari ~10 minit ke ~5 minit per konsultasi)
- **EMR Completion Rate:** > 95% EMR di-finalize dalam 24 jam
- **Voice-to-Text Usage:** > 60% doktor guna voice-to-text feature
- **Medical History Access:** Doktor access medical history dalam 90% konsultasi
- **Integration Success Rate:** > 98% auto-populate ke Farmasi/Billing berjaya
- **Audit Trail Compliance:** 100% changes direkod dalam audit log

## 12.2 User Satisfaction

- **Kepuasan Doktor:** > 4.0/5.0 (survey)
- **Kepuasan Jururawat:** > 4.0/5.0 (survey)
- **Pengarah Perubatan Satisfaction:** > 4.5/5.0 (survey untuk audit capabilities)

## 12.3 Technical Metrics

- **Uptime:** > 99.5%
- **Response Time:** < 2 saat untuk 95% requests
- **Bug Rate:** < 3 critical bugs per bulan selepas deployment
- **Data Accuracy:** 100% audit trail accuracy (no missing logs)

## 12.4 Clinical Quality Metrics

- **SOAP Format Compliance:** > 90% EMR guna SOAP format lengkap
  - **ICD-10 Coding Accuracy:** > 95% diagnosis ada ICD-10 code yang valid
  - **Medical Errors Reduction:** Target 30% reduction dalam medical errors related to missing history
- 

# 13. Risks & Mitigation

Risk	Likelihood	Impact	Mitigation
Doktor resist new system, prefer manual notes	High	High	Comprehensive training + voice-to-text untuk ease adoption + show time savings benefit
Voice-to-text accuracy rendah untuk medical terms	Medium	Medium	Provide fallback to manual typing + custom medical vocabulary training
Integration dengan Farmasi/Billing/Makmal fails	Medium	High	Thorough integration testing + fallback manual entry option + queue retry mechanism
File upload security vulnerability	Low	High	Strict file type validation + malware scanning + store outside web root + regular security audits
Performance degradation dengan large medical history	Medium	Medium	Lazy loading + pagination + database indexing + query optimization
PDPA compliance issues	Low	Critical	Full audit trail + soft delete only + proper consent management + regular compliance audits
ICD-10 codes database outdated	Low	Medium	Scheduled annual update from official WHO ICD-10 source
Network downtime prevents cloud file access	Low	Medium	Cache recently viewed files locally + offline mode consideration for future

## 14. Dependencies

### 14.1 External Packages

- **Package Name:** spatie/laravel-permission
  - **Version:** ^6.0
  - **Purpose:** Role-based access control (Doktor, Jururawat, Pengarah Perubatan)
- **Package Name:** tinymce/tinymce (atau) summernote/summernote
  - **Version:** Latest stable
  - **Purpose:** Rich text editor untuk SOAP notes
- **Package Name:** (Optional) league/flysystem-aws-s3-v3
  - **Version:** ^3.0
  - **Purpose:** S3 storage untuk attachments
- **Package Name:** (Optional) VirusTotal API Client
  - **Version:** Latest stable
  - **Purpose:** Malware scanning untuk uploaded files

### 14.2 Related Features/Modules

- **Bergantung Kepada:**
  - Modul Pendaftaran Pesakit (mesti wujud dahulu, untuk link pesakit\_id)
  - Modul Temujanji (optional, untuk link temujanji\_id)
  - User Management dengan Roles (Doktor, Jururawat, Pengarah Perubatan)
- **Memberi Impak Kepada:**
  - Modul Farmasi (terima prescription data dari EMR)

- Modul Billing (terima bill items dari EMR)
- Modul Makmal (terima lab orders dan send results ke EMR)
- Modul Temujanji (terima follow-up scheduling dari EMR)

### 14.3 Third-Party Integrations

- **Service:** AWS S3 atau compatible object storage
    - **Configuration Required:**
      - AWS\_ACCESS\_KEY\_ID
      - AWS\_SECRET\_ACCESS\_KEY
      - AWS\_DEFAULT\_REGION
      - AWS\_BUCKET
  - **Service:** (Optional) VirusTotal API
    - **Configuration Required:**
      - VIRUSTOTAL\_API\_KEY
  - **Service:** WHO ICD-10 Database
    - **Configuration Required:** Import ICD-10 codes dari official source (one-time setup, annual updates)
- 

## 15. Acceptance Criteria

### 15.1 Functional Acceptance

- Semua functional requirements (FR-01 hingga FR-15) dilaksanakan
- Semua user stories dapat diselesaikan dengan jayanya
- Doktor boleh create, view, edit, finalize EMR dengan SOAP format
- Voice-to-text berfungsi untuk all textarea fields
- ICD-10 search autocomplete berfungsi dengan accurate results
- Vital signs auto-calculate BMI dengan betul
- File upload validates type dan size, store dengan selamat
- Medical history display dalam chronological timeline
- Integration dengan Farmasi, Billing, Makmal berfungsi
- Audit trail rekod semua changes dengan accurate
- Role-based access berfungsi (Doktor vs Jururawat vs Pengarah)
- Search dan filter EMR berfungsi

### 15.2 Technical Acceptance

- Semua feature tests lulus (minimum 80% coverage)
- Semua unit tests lulus (minimum 70% coverage)
- Kod mengikut conventions dari `DEVELOPER_GUIDE.md` dan `.github/copilot-instructions.md`
- Kod diformat dengan `./vendor/bin/pint`
- Tiada N+1 query problems (guna eager loading)
- Route cache cleared selepas tambah routes
- Database migrations run successfully tanpa errors
- Seeder berfungsi untuk ICD-10 codes

### 15.3 Quality Acceptance

- Kod di-review oleh peer (technical lead)

- Manual testing selesai oleh QA team
- UAT selesai dengan doktor dan jururawat (feedback positif)
- Tiada console errors atau warnings di browser
- Responsive design berfungsi di tablet (primary device)
- Voice-to-text tested di multiple browsers (Chrome, Firefox, Safari)
- Accessibility considerations ditangani (keyboard navigation, screen reader friendly)
- Security audit selesai (file upload, XSS, SQL injection)

## 15.4 Documentation Acceptance

- PRD dikemaskini dengan implementation notes akhir
- DEVELOPER\_GUIDE.md dikemaskini dengan EMR module patterns
- API documentation untuk integration points (Farmasi, Billing, Makmal)
- User manual (PDF) untuk Doktor dan Jururawat
- Admin guide untuk Pengarah Perubatan (audit trail usage)
- Inline code comments untuk complex business logic

# 16. Lampiran

## 16.1 Contoh SOAP Format

### Subjective:

Chief Complaint: Demam dan batuk selama 3 hari

#### History of Present Illness:

Pesakit mengadu demam onset 3 hari yang lalu, grade: high ( $39^{\circ}\text{C}$ ), associated with productive cough dengan kahak kuning, tiada shortness of breath, tiada chest pain.

Past Medical History: Diabetes Mellitus Type 2 (on metformin)

Medications: Metformin 500mg BD

Allergies: Penicillin (rash)

Family History: Father - Hypertension, Mother - Diabetes

Social History: Non-smoker, no alcohol

### Objective:

#### Vital Signs:

- BP: 120/80 mmHg
- Temperature:  $38.5^{\circ}\text{C}$
- Pulse: 88 bpm
- Respiratory Rate: 18 /min
- SpO<sub>2</sub>: 97% on room air
- Weight: 70 kg, Height: 170 cm, BMI: 24.2 (Normal)

#### Physical Examination:

- General: Alert, conscious, mild distress
- Chest: Bilateral equal air entry, coarse crepitations at right lower zone
- CVS: S1 S2 normal, no murmurs

- Abdomen: Soft, non-tender

Investigations: Chest X-ray (attached) - Right lower lobe consolidation

#### Assessment:

1. Community-Acquired Pneumonia (Right lower lobe)
2. Diabetes Mellitus Type 2 - controlled

#### Plan:

1. Admit for IV antibiotics (Ceftriaxone 1g IV BD)
2. Antipyretic: Paracetamol 1g QID PRN
3. Bronchodilator: Salbutamol nebulizer 2.5mg QID
4. Monitor vitals Q6H
5. Repeat Chest X-ray in 3 days
6. Continue Metformin 500mg BD
7. Follow-up in OPD after 1 week post-discharge

## 16.2 Contoh ICD-10 Codes

Code	Description
A00.0	Cholera due to Vibrio cholerae O1, biovar cholerae
J18.9	Pneumonia, unspecified organism
E11.9	Type 2 diabetes mellitus without complications
I10	Essential (primary) hypertension
J06.9	Acute upper respiratory infection, unspecified

## 16.3 Database ER Diagram

```
[pesakit] 1---* [emr] *---1 [doktor/users]
  |
  |---1 [vital_signs]
  |
  |---* [diagnoses] *---1 [icd10_codes]
  |
  |---* [emr_attachments]
  |
  |---* [nursing_notes] *---1 [jururawat/users]
  |
  |---* [emr_audit_log]
```

## 16.4 References

- WHO ICD-10: <https://icd.who.int/browse10/2019/en>
- Web Speech API: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Speech\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API)
- PDPA Malaysia: <https://www.pdp.gov.my/>

## 16.5 Change Log

Tarikh	Penulis	Perubahan
13 Jan 2026	Product Team	PRD awal dicipta

## 16.6 Approval

- **Product Owner:** [Nama] - [Tarikh]
- **Tech Lead:** [Nama] - [Tarikh]
- **Pengarah Perubatan:** [Nama] - [Tarikh]
- **Stakeholders:** [Nama-nama] - [Tarikh]

---

**Status Implementasi:** Belum Bermula **Tarikh Selesai:** [Tarikh bila feature deploy ke production]

---

**Catatan:** Dokumen ini adalah living document dan akan dikemaskini mengikut keperluan semasa development.