

# TECHNICAL TEST

## BACKEND DEVELOPER

*REST API Development - PDF Management System*

Nama Kandidat	
Tanggal Tes	
Durasi	640 Menit (8 Jam)
Passing Grade	80%

### PETUNJUK PENGERJAAN

- Baca setiap task dengan teliti sebelum mulai coding
- Gunakan bahasa pemrograman Golang (tanpa framework diutamakan)
- Semua API harus dapat ditest menggunakan Postman
- Sertakan dokumentasi API (endpoint, method, request/response) di akhir penggerjaan
- Code harus clean, readable, dan mengikuti best practices
- Commit code ke repository (GitHub/GitLab) dan share link-nya

### TEKNOLOGI YANG DISARANKAN

Komponen	Pilihan
Bahasa Pemrograman	Golang
Database	PostgreSQL (Alternatif: MySQL atau SQLite)

## DESKRIPSI TASK

Buatlah REST API untuk sistem manajemen file PDF dengan fitur-fitur berikut:

No	Task	Method	Bobot
1	Generate Report PDF	POST	30%
2	Upload PDF	POST	20%
3	List File PDF	GET	20%
4	Delete File PDF	DELETE	15%
5	Dokumentasi API	-	15%

# TASK 1: GENERATE REPORT PDF

(Bobot: 30%)

Buat API untuk generate file PDF report dengan spesifikasi berikut:

Endpoint	POST /api/pdf/generate
----------	------------------------

## Requirement:

### 1. Header PDF:

- Logo/image di kiri atas (dapat di-set dari parameter atau default)
- Nama institusi/perusahaan di tengah
- Alamat dan kontak di bawah nama

### 2. Content PDF:

- Judul dokumen (dari parameter)
- Tanggal generate
- Isi konten dokumen (dari parameter, support text/paragraf)

### 3. Footer PDF:

- Nomor halaman (Page X of Y)
- Timestamp generate

### 4. File harus tersimpan di server dengan nama unik==

### 5. Record file harus tersimpan di database dengan status "CREATED"

## Contoh Request Body:

```
{  
  "title": "Laporan Kunjungan Pasien",  
  "institution_name": "RS Sehat Sentosa",  
  "address": "Jl. Kesehatan No. 123, Jakarta",  
  "phone": "(021) 123-4567",  
  "logo_url": "https://example.com/logo.png", // optional  
  "content": "Berikut adalah laporan kunjungan pasien bulan Januari 2025..."  
}
```

## Contoh Response (Success):

```
{  
  "success": true,  
  "message": "PDF generated successfully",  
  "data": {  
    "id": 1,  
    "filename": "report_20250127_abc123.pdf",  
    "filepath": "/uploads/pdf/report_20250127_abc123.pdf",  
    "status": "CREATED",  
    "created_at": "2025-01-27T10:30:00Z"  
  }  
}
```

## TASK 2: UPLOAD PDF

(Bobot: 20%)

Buat API untuk upload file PDF dari user dengan spesifikasi berikut:

<b>Endpoint</b>	POST /api/pdf/upload
<b>Content-Type</b>	multipart/form-data

### Requirement:

1. Hanya menerima file dengan ekstensi .pdf
2. Validasi MIME type harus application/pdf
3. Simpan file dengan nama unik (hindari overwrite)
4. Record file harus tersimpan dengan status "UPLOADED"
5. Return error message yang jelas jika validasi gagal

### Contoh Response (Success):

```
{  
  "success": true,  
  "message": "PDF uploaded successfully",  
  "data": {  
    "id": 2,  
    "original_name": "document.pdf",  
    "filename": "upload_20250127_xyz789.pdf",  
    "filepath": "/uploads/pdf/upload_20250127_xyz789.pdf",  
    "size": 1024567,  
    "status": "UPLOADED",  
    "created_at": "2025-01-27T10:35:00Z"  
  }  
}
```

### Contoh Response (Error - File Too Large):

```
{  
  "success": false,  
  "message": "File size exceeds maximum limit (10MB)",  
  "error_code": "FILE_TOO_LARGE"  
}
```

## TASK 3: LIST FILE PDF

(Bobot: 20%)

Buat API untuk menampilkan daftar semua file PDF dengan spesifikasi berikut:

<b>Endpoint</b>	GET /api/pdf/list
-----------------	-------------------

### Requirement:

1. Tampilkan semua file PDF yang ada di database
2. Setiap file harus memiliki status yang jelas:
  - CREATED - file yang di-generate dari API Generate
  - UPLOADED - file yang di-upload dari API Upload
  - DELETED - file yang sudah dihapus (soft delete)
3. Support filter berdasarkan status (opsional query parameter)
4. Support pagination (opsional)
5. Urutkan berdasarkan tanggal terbaru (default)

### Query Parameters (Opsional):

Parameter	Type	Description
status	string	Filter by status (CREATED/UPLOADED/DELETED)
page	integer	Nomor halaman (default: 1)
limit	integer	Jumlah data per halaman (default: 10)

### Contoh Response:

```
{  
  "success": true,  
  "data": [  
    {  
      "id": 1,  
      "filename": "report_20250127_abc123.pdf",  
      "original_name": null,  
      "size": 52436,  
      "status": "CREATED",  
      "created_at": "2025-01-27T10:30:00Z"  
    },  
    {  
      "id": 2,  
      "filename": "upload_20250127_xyz789.pdf",  
      "original_name": "document.pdf",  
      "size": 1024567,  
      "status": "UPLOADED",  
      "created_at": "2025-01-27T10:35:00Z"  
    }  
  ]  
}
```

```
        }
    ],
    "pagination": { "page": 1, "limit": 10, "total": 2 }
}
```

## TASK 4: DELETE FILE PDF

(Bobot: 15%)

Buat API untuk menghapus file PDF dengan spesifikasi berikut:

<b>Endpoint</b>	DELETE /api/pdf/{id}
-----------------	----------------------

### Requirement:

1. Implementasi Soft Delete (update status menjadi "DELETED", jangan hapus record)
2. File fisik di storage TIDAK dihapus (untuk keperluan audit/recovery)
3. Simpan timestamp deleted\_at
4. Return error jika file tidak ditemukan
5. Return error jika file sudah dalam status DELETED

### Contoh Response (Success):

```
{  
  "success": true,  
  "message": "PDF deleted successfully",  
  "data": {  
    "id": 1,  
    "filename": "report_20250127_abc123.pdf",  
    "status": "DELETED",  
    "deleted_at": "2025-01-27T11:00:00Z"  
  }  
}
```

## **TASK 5: DOKUMENTASI API**

*(Bobot: 15%)*

Buat dokumentasi API yang lengkap mencakup:

**1. README.md yang berisi:**

- Deskripsi project
- Tech stack yang digunakan
- Cara instalasi dan menjalankan project
- Struktur folder project

**2. API Documentation yang berisi:**

- List semua endpoint
- Request method, headers, body
- Response format (success & error)
- Contoh request/response

**3. Database Schema (ERD atau dokumentasi tabel)**

**4. Postman Collection (export dan sertakan di repository)**

## REFERENSI: DATABASE SCHEMA

Berikut adalah contoh struktur tabel yang dapat digunakan (boleh dimodifikasi sesuai kebutuhan):

Tabel: **pdf\_files**

Column	Type	Null	Description
<code>id</code>	BIGINT	NO	Primary Key, Auto Increment
<code>filename</code>	VARCHAR(255)	NO	Nama file yang tersimpan di server
<code>original_name</code>	VARCHAR(255)	YES	Nama file asli (untuk upload)
<code>filepath</code>	VARCHAR(500)	NO	Path lengkap file di server
<code>size</code>	BIGINT	YES	Ukuran file dalam bytes
<code>status</code>	ENUM	NO	CREATED, UPLOADED, DELETED
<code>created_at</code>	TIMESTAMP	NO	Waktu file dibuat/diupload
<code>updated_at</code>	TIMESTAMP	YES	Waktu terakhir diupdate
<code>deleted_at</code>	TIMESTAMP	YES	Waktu file dihapus (soft delete)

## DELIVERABLES

Setelah selesai mengerjakan, pastikan Anda menyerahkan:

No	Item	Keterangan
1	Link Repository	GitHub/GitLab repository (public atau invite penguji)
2	Source Code	Code yang clean dan well-structured
3	README.md	Dokumentasi cara instalasi dan menjalankan project
4	API Documentation	Dokumentasi endpoint API
5	Postman Collection	File .json collection untuk testing
6	Database Script (Opsional)	SQL script atau migration file
7	Live Demo (Interview)	localhost