

TUGAS 7 BACKEND LANJUTAN



MATA KULIAH BACKEND LANJUTAN (PRAKTIKUM)

Disusun Oleh:

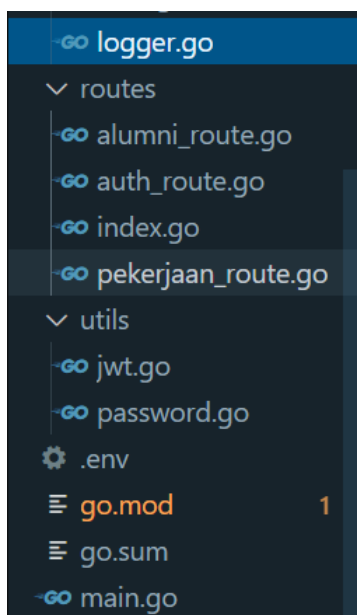
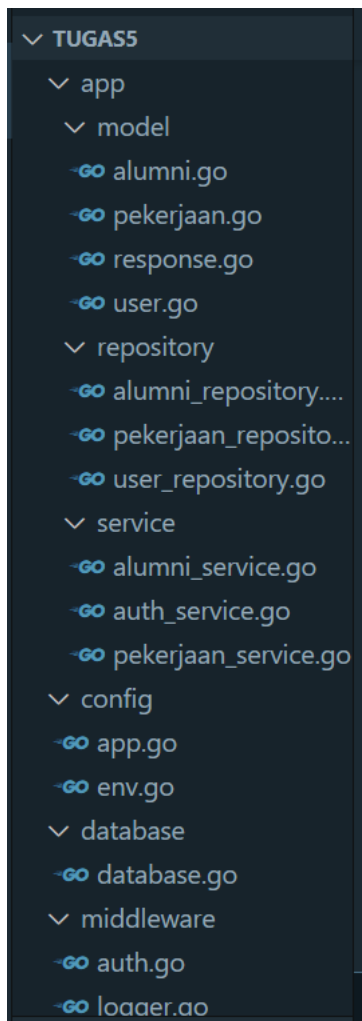
Elia Sari

(434231023)

Kelas Praktikum: C-8

**FAKULTAS VOKASI
UNIVERSITAS AIRLANGGA
AGUSTUS, 2025**

1. Struktur



2. Tabel + Kolom + Data

User:

Data Output Messages Notifications						
Showing rows: 1 to 4 Page No: 1 of 1						
	id [PK] integer	username character varying (50)	email character varying (100)	password_hash character varying (255)	role character varying (20)	created_at timestamp without time zone
1	1	admin	admin@university.com	\$2a\$10\$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi	admin	2025-09-13 13:25:37.405959
2	2	user1	user1@university.com	\$2a\$10\$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi	user	2025-09-13 13:25:37.405959
3	3	andi	andi@example.com	\$2a\$10\$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi	user	2025-09-25 11:55:39.357233
4	4	admin1	admin@example.com	\$2a\$10\$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi	admin	2025-09-25 11:55:39.357233

Alumni:

Showing rows: 1 to 19 Page No: 1												
	id [PK] integer	nim character	nama character varying (100)	jurusan character varying (100)	angkatan integer	tahun_lulus integer	email character varying (100)	no_telepon character varying (20)	alamat text	created_at timestamp without time zone	updated_at timestamp without time zone	user_id integer
10	16	A004	Rina Putri	Teknik Infor...	2018	2022	rina@email.com	081234567...	Jakarta	2025-09-18 12:57...	2025-09-18 12:5...	[null]
11	17	A005	Dedi Saputra	Sistem Infor...	2019	2023	dedi@email.com	081234567...	Bandung	2025-09-18 12:57...	2025-09-18 12:5...	[null]
12	18	A006	Maya Lestari	Teknik Komp...	2017	2021	maya@email.com	081234567...	Surabaya	2025-09-18 12:57...	2025-09-18 12:5...	[null]
13	19	A007	Fajar Prata...	Teknik Infor...	2018	2022	fajar@email.com	081234567...	Jakarta	2025-09-18 12:57...	2025-09-18 12:5...	[null]
14	20	A008	Nina Kusuma	Sistem Infor...	2019	2023	nina@email.com	081234567...	Bandung	2025-09-18 12:57...	2025-09-18 12:5...	[null]
15	21	A009	Riko Hidayat	Teknik Komp...	2017	2021	riko@email.com	081234567...	Surabaya	2025-09-18 12:57...	2025-09-18 12:5...	[null]
16	22	A010	Tina Mahar...	Teknik Infor...	2018	2022	tina@email.com	081234567...	Jakarta	2025-09-18 12:57...	2025-09-18 12:5...	[null]
17	23	A123...	lili	Informatika	2021	2025	lili@university.co...	081234567...	Jakarta	2025-09-25 11:48...	2025-09-25 11:4...	2
18	25	A543...	Andi Setiaw...	Teknik Infor...	2019	2023	andi.alumni@ex...	081234567...	Surabaya	2025-09-25 11:56...	2025-09-25 11:5...	3
19	24	A678...	Ibud	Sistem Infor...	2020	2024	ibud@university...	089876543...	Bandung	2025-09-25 11:48...	2025-09-25 11:4...	3

Pekerjaan_Alumni:

Showing rows: 1 to 20														Page No: 1		of 1	
	id [PK]	alumni_id	nama_perusahaan	posisi_jabatan	bidang_industri	lokasi_kerja	gaji_range	tanggal_mulai	tanggal_selesai	status	deskripsi_pekerjaan	created_at	updated_at	is_delete			
	integer	integer	character varying (100)	character varying (100)	character varying (100)	character varying (100)	character varying (100)	date	date	character varying (20)	text	timestamp without time zone	timestamp without time zone	timestamp without time zone			
11	20	16	CloudX	DevOps Engineer	Cloud	Bandung	15-22 juta	2023-02-...	[null]	aktif	Setup dan konfigurasi infrastruktur cloud.	2025-09-18 10:30:15	2025-09-18 10:30:15	[null]			
12	21	17	CyberSec	Security Analyst	Cybersecurity	Surabaya	12-18 juta	2023-03-...	[null]	aktif	Memantau dan menganalisis ancaman keamanan siber.	2025-09-18 11:45:22	2025-09-18 11:45:22	[null]			
13	22	18	SmartApps	Mobile Developer	Mobile	Jakarta	13-19 juta	2024-01-...	[null]	aktif	Membuat aplikasi mobile untuk platform Android dan iOS.	2025-09-18 12:00:00	2025-09-18 12:00:00	[null]			
14	23	19	DataBridge	BI Developer	Data Analytics	Bandung	11-17 juta	2023-04-...	[null]	aktif	Membangun dashboard dan laporan berbasis data.	2025-09-18 13:15:30	2025-09-18 13:15:30	[null]			
15	24	20	NextGen	Fullstack Developer	IT	Surabaya	14-20 juta	2023-05-...	[null]	resign	Mengembangkan aplikasi web fullstack.	2025-09-18 14:30:45	2025-09-18 14:30:45	[null]			
16	25	21	Innova	Project Manager	Management	Jakarta	18-25 juta	2023-06-...	[null]	aktif	Memimpin tim dalam menyelesaikan proyek.	2025-09-18 15:00:00	2025-09-18 15:00:00	[null]			
17	101	23	Google	Software Engineer	Teknologi	Jakarta	10-20 juta	2023-01-...	[null]	aktif	Ngoding dan mengembangkan produk baru.	2025-09-25 16:15:10	2025-09-25 16:15:10	[null]			
18	102	23	Tokopedia	QA Engineer	E-commerce	Jakarta	8-15 juta	2022-01-...	2022-06-...	aktif	Quality Assurance untuk platform e-commerce.	2025-09-25 17:30:00	2025-09-25 17:30:00	2025-09-25 12:19:53.539007			
19	104	25	Bukalapak	Backend Developer	E-commerce	Surabaya	9-14 juta	2023-06-...	[null]	aktif	Ngebangun sistem backend untuk marketplace.	2025-09-25 18:45:30	2025-09-25 18:45:30	[null]			
20	105	25	Traveloka	DevOps Engineer	Teknologi	Jakarta	12-18 juta	2022-01-...	2022-06-...	selesai	Ngurusin CI/CD pipeline.	2025-09-25 19:00:00	2025-09-25 19:00:00	2025-09-25 12:28:40.978112			

3. Penjelasan Kode

a. Model/[Alumni.go](#)

```
package model

import "time"

type Alumni struct {
    ID          int          `json:"id"`
    NIM         string       `json:"nim"`
    Nama        string       `json:"nama"`
    Jurusan     string       `json:"jurusan"`
    Angkatan    int          `json:"angkatan"`
    TahunLulus  int          `json:"tahun_lulus"`
    Email       string       `json:"email"`
    NoTelepon   string       `json:"no_telepon"`
    Alamat      string       `json:"alamat"`
    CreatedAt   time.Time    `json:"created_at"`
    UpdatedAt   time.Time    `json:"updated_at"`
    UserID      int          `json:"user_id"`
}
```

Struct Alumni ngegambarin tabel alumni. Field-nya nyimpen data dasar alumni, kayak NIM, nama, jurusan, angkatan, info kontak, sampai UserID yang nyambung ke akun user pemilik data. Ini dipakai buat validasi kepemilikan data pekerjaan di repository.

b. Model/[Pekerjaan.go](#)

```
package model

import "time"

type Pekerjaan struct {
    ID              int          `json:"id"`
    AlumniID        int          `json:"alumni_id"`
    NamaPerusahaan  string       `json:"nama_perusahaan"`
    PosisiJabatan   string       `json:"posisi_jabatan"`
    BidangIndustri  string       `json:"bidang_industri"`
    LokasiKerja     string       `json:"lokasi_kerja"`
    GajiRange       string       `json:"gaji_range"`
    TanggalMulaiKerja time.Time    `json:"tanggal_mulai_kerja"`
    TanggalSelesaiKerja *time.Time  `json:"tanggal_selesai_kerja"`
    StatusPekerjaan string       `json:"status_pekerjaan"`
    DeskripsiPekerjaan string      `json:"deskripsi_pekerjaan"`
    CreatedAt       time.Time    `json:"created_at"`
    UpdatedAt       time.Time    `json:"updated_at"`
    IsDelete        *time.Time  `json:"is_delete"`
}
```

Struct Pekerjaan ngegambarin tabel pekerjaan_alumni. Isinya detail pekerjaan alumni seperti nama perusahaan, posisi, industri, gaji, tanggal kerja, status, deskripsi, dan kolom is_delete bertipe waktu. is_delete jadi penanda kapan pekerjaan di-soft delete, bukan beneran dihapus dari database.

c. Repository/Pekerjaan_repository.go

```
func NewPekerjaanRepository(db *sql.DB) *PekerjaanRepository {
    return &PekerjaanRepository{DB: db}
}

func (r *PekerjaanRepository) SoftDelete(id int, userID int, isAdmin bool) error {
    now := time.Now()

    if isAdmin {
        _, err := r.DB.Exec(`UPDATE pekerjaan_alumni SET is_delete = $1 WHERE id = $2`, now, id)
        return err
    }

    res, err := r.DB.Exec(`
        UPDATE pekerjaan_alumni p
        SET is_delete = $1
        FROM alumni a
        WHERE p.alumni_id = a.id AND p.id = $2 AND a.user_id = $3
    `, now, id, userID)
    if err != nil {
        return err
    }

    rows, _ := res.RowsAffected()
    if rows == 0 {
        return sql.ErrNoRows
    }
    return nil
}
```

Repository ini yang langsung ngobrol sama database. Fungsi SoftDelete update kolom is_delete di tabel pekerjaan_alumni dengan waktu sekarang (time.Now()). Kalau user admin, update jalan langsung berdasarkan id. Kalau bukan admin, query update pakai join ke tabel alumni buat ngecek pekerjaan itu beneran milik user tersebut. Kalau baris yang diupdate = 0, repository balikin sql.ErrNoRows biar service bisa kasih error “tidak bisa hapus pekerjaan ini”.

d. Service/Pekerjaan_service.go

```
func SoftDeletePekerjaanService(c *fiber.Ctx, db *sql.DB) error {
    id, err := strconv.Atoi(c.Params("id"))
    if err != nil {
        return c.Status(400).JSON(fiber.Map{
            "error": "ID tidak valid",
            "success": false,
        })
    }

    userID := c.Locals("user_id").(int)
    role := c.Locals("role").(string)
    isAdmin := role == "admin"

    repo := repository.NewPekerjaanRepository(db)
    err = repo.SoftDelete(id, userID, isAdmin)
    if err != nil {
        if err == sql.ErrNoRows {
            return c.Status(403).JSON(fiber.Map{
                "error": "Tidak bisa hapus pekerjaan ini",
                "success": false,
            })
        }
        return c.Status(500).JSON(fiber.Map{
            "error": "Gagal soft delete pekerjaan: " + err.Error(),
            "success": false,
        })
    }

    return c.JSON(fiber.Map{
        "message": "Pekerjaan berhasil dihapus (soft delete)",
        "success": true,
    })
}
```

SoftDeletePekerjaanService yang jadi handler utama. Pertama dia ambil parameter id dari URL dan cek validitasnya. Terus ambil user_id dan role dari context (data user yang udah login). Role dicek apakah admin. Setelah itu dia bikin instance repository PekerjaanRepository dan panggil SoftDelete. Kalau ada error karena ID nggak ketemu atau user nggak punya hak, service balikin response JSON error. Kalau sukses, balikin pesan “pekerjaan berhasil dihapus (soft delete)”.

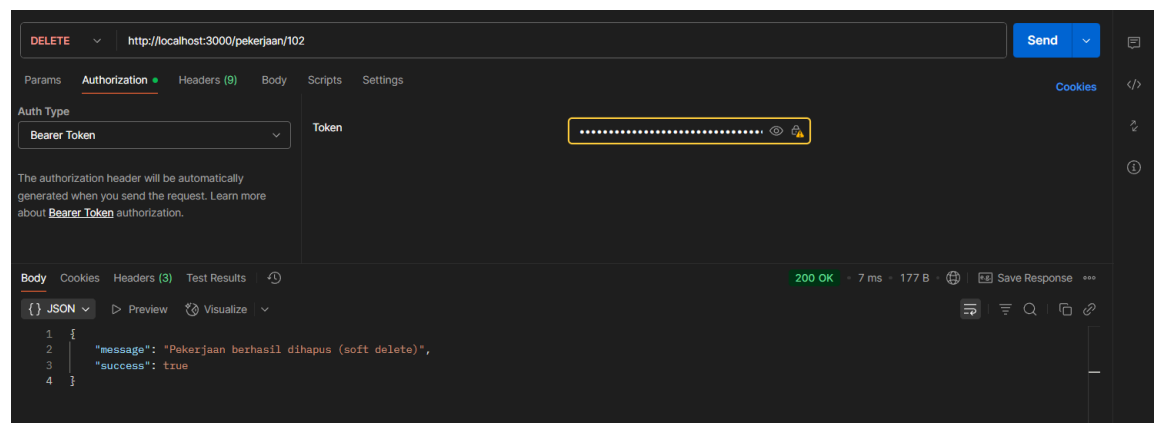
e. Route/Pekerjaan_r

```
pekerjaan.Delete("/:id", func(c *fiber.Ctx) error {  
    return service.SoftDeletePekerjaanService(c, db)  
})
```

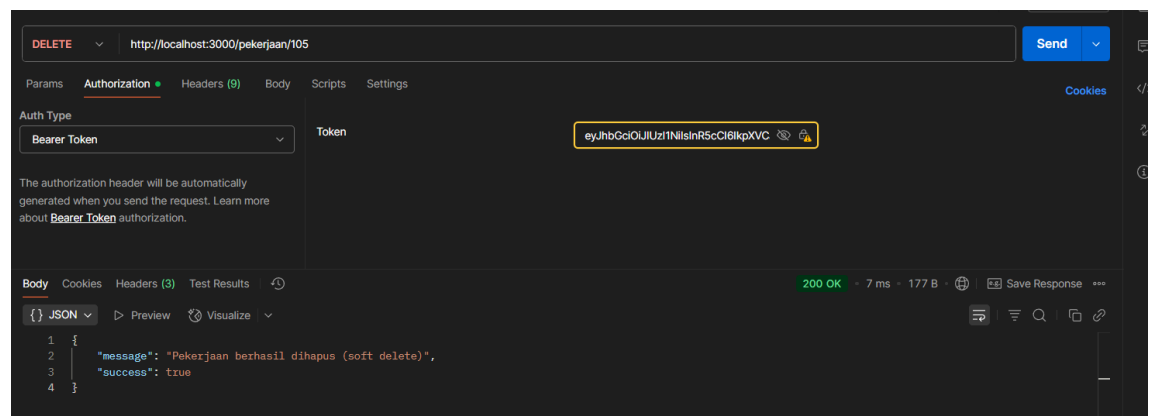
Di file ini dibuat endpoint DELETE /:id untuk resource pekerjaan. Kalau user nge-hit URL dengan metode DELETE dan nyantumin id pekerjaan, request akan diarahkan ke fungsi SoftDeletePekerjaanService. Fiber Ctx (c) dan koneksi database (db) dikirim ke service itu buat diproses.

4. Hasil

a. Delete by Admin



b. Delete by User dengan pekerjaannya



c. Delete by User yang bukan pekerjaannya

DELETE

http://localhost:3000/pekerjaan/102

Send

Params

Authorization

Headers (9)

Body

Scripts

Settings

Auth Type

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body

Cookies

Headers (3)

Test Results

403 Forbidden

8 ms

173 B

Save Response

JSON

Preview

Visualize

1

2

3

4

{

"error": "Tidak bisa hapus pekerjaan ini",

"success": false

}